COMBINATORIAL MOTIF ANALYSIS IN YEAST GENE PROMOTERS: THE BENEFITS OF A BIOLOGICAL CONSIDERATION OF MOTIFS

A Thesis

by

KEVIN L. CHILDS

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

December 2004

Major Subject: Computer Science

COMBINATORIAL MOTIF ANALYSIS IN YEAST GENE PROMOTERS: THE BENEFITS OF A BIOLOGICAL CONSIDERATION OF MOTIFS

A Thesis

by

KEVIN L. CHILDS

Submitted to Texas A&M University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Approved as to style and content by:

Thomas R. Ioerger (Chair of Committee)	Sing-Hoi Sze (Member)
Terry L. Thomas (Member)	Valerie E. Taylor (Head of Department)

December 2004

Major Subject: Computer Science

ABSTRACT

Combinatorial Motif Analysis in Yeast Gene Promoters: The Benefits of a Biological Consideration of Motifs. (December 2004)

Kevin L. Childs, B.S., University of Michigan; Ph.D., Texas A&M University

Chair of Advisory Committee: Dr. Thomas R. Ioerger

There are three main categories of algorithms for identifying small transcription regulatory sequences in the promoters of genes, phylogenetic comparison, expectation maximization and combinatorial. For convenience, the combinatorial methods typically define motifs in terms of a canonical sequence and a set of sequences that have a small number of differences compared to the canonical sequence. Such motifs are referred to as (*l, d*)-motifs where *l* is the length of the motif and *d* indicates how many mismatches are allowed between an instance of the motif and the canonical motif sequence. There are limits to the complexity of the patterns of motifs that can be found by combinatorial methods. For some values of *l* and *d,* there will exist many sets of random words in a cluster of gene promoters that appear to form an (*l, d*)-motif. For these motifs, it will be impossible to distinguish biological motifs from randomly generated motifs. A better formalization of motifs is the (*l, f, d*)-motif that is derived from a biological consideration of motifs. The motivation for (*l, f, d*)-motifs comes from an examination of known transcription factor binding sites where typically a few positions in the motif are invariant. It is shown that there exist (*l, f, d*)-motifs that can be found in the promoters of gene clusters that would not be recognizable from random sequences if they were described as (l, d)-motifs. The inclusion of the f-value in the definition of motifs suggests that the sequence space that is occupied by a motif will consist of a several clusters of closely related sequences. An algorithm, CM, has been developed that identifies small sets of overabundant sequences in the promoters from a cluster of genes and then combines these simple sets of sequences to form complex (l, f, d)-motif models. A dataset from a yeast gene expression experiment is analyzed with CM. Known biological motifs and novel motifs are identified by CM. The performance of CM is compared to that of a popular expectation maximization algorithm, AlginACE, and to that from a simple combinatorial motif finding program.

TABLE OF CONTENTS

LIST OF FIGURES

FIGURE Page

LIST OF TABLES

INTRODUCTION

A focus of biological research has been to understand the mechanisms of control of gene expression. Although gene expression can be regulated at many levels, much biological research examines the control of gene expression by protein transcription factors that bind to specific, short deoxyribose nucleic acid (DNA) sequences that are found in the promoter regions of genes. The DNA sequences to which the transcription factors bind are called motifs. Transcription factors aid in the transcription of ribose nucleic acid (RNA) from a gene. Because the transcription factor binding sequences that are present in a promoter are so important for determining a gene's expression, biologists are interested in identifying motif sequences. Initially, it was difficult to identify transcription factor binding motifs. A biologist would have to either sequentially mutate the nucleotides in a gene's promoter and examine the effect of the mutations on the gene's expression or do footprinting experiments to discover the region of a gene's promoter that is bound by a DNA-binding protein. In this way, regions of the promoter that are important for transcription factor binding could be found. With the advent of genomic scale DNA sequencing projects and large-scale gene expression experiments, it has become possible to find transcription factor motifs in a more wholesale fashion. Working under the assumption that genes that have similar expression patterns are likely to be regulated by the same transcription factors, it is possible to compare the promoter regions of co-regulated genes in order to find common sequences. This comparison cannot be done easily by eye because the binding sites for transcription factors are degenerate. Although a set of gene

This thesis follows the style and format of the journal *Bioinformatics*.

promoters may all contain the motif for a particular transcription factor, the particular instance of that motif may be different in each promoter. The problem of identifying an unknown transcription factor motif in the promoters of a set of similarly regulated genes has been greatly aided by computational methods.

Nonetheless, the motif-finding problem is nontrivial even when using computational analysis. Promoters are often several hundred nucleotides long. Most motifs are rather short at 6 to 20 nucleotides. Typically, some percentage of the positions in a given instance of a motif will vary from the canonical sequence of the motif. The sets of gene promoters that are used for motif analysis often range from 20 to 120 sequences each with a length between 500 to 1000 nucleotides. The problem is to find an unknown instance of the unknown motif in some or all of the promoters of the co-regulated genes when each instance of the motif may contain an unknown percentage of degenerated nucleotides at unknown positions when compared to the canonical motif sequence.

This problem inspired a computer science problem called Closest String, and it can be stated in more formal terms (Li *et al.*, 2002). Given a set of strings, S, of a common length, l, find a string, s, such that $d(s, s_i) \leq d$ for all s_i in S, where $d(s, s_i)$ is the Hamming distance between two strings and d is some number less than or equal to l. The Hamming distance between two strings is the minimum number of positions in one string that must be changed in order to generate the second string. More similar to motif finding is the Closest Substring problem. Given a set of strings, S, each of length greater than or equal to l, find a string, s, such that $d(s, s_i) \leq d$ for some substring, s_i, of length l from each string in S. The Closest String problem, and by extension the Closest Substring problem, is NP-hard (Frances and Litman, 1997). However, for small numbers of strings and small values of l

and *d*, these problems are tractable, and they can be solved in reasonable amounts of time. With motif finding, the parameters *d*, *l* and the number of strings to analyze result in a problem that is believed to be not always solvable for probabilistic reasons explained below.

Motif finding algorithms fall into one of several categories. There are Expectation Maximization (EM) techniques that iteratively build a probability matrix model of the motif (Bailey and Elkan, 1995; Lawrence *et al.*, 1993). Initially, random words from each promoter are chosen to generate the motif model. Then the word that was used to build the model is removed from the model, and another word from that promoter with a good score based on the model is chosen to replace it. The model is then rebuilt, and the process is repeated many times. This heuristic performs a local search and can settle upon an incorrect solution. However, the algorithm is typically run numerous times, and the best motif model is chosen as the solution. This general method works reasonably well and is used by many researchers. Phylogenetic comparison is another class of motif finding that has recently become possible for reasonably closely related organisms with sequenced genomes (Cliften *et al.*, 2003; Kellis *et al.*, 2003; McCue *et al.*, 2002). It is possible to compare the promoter regions of homologous genes from such pairs of organisms to find regulatory sequences. If the genomes that are being compared have diverged to just the right degree, it is easy to find evolutionarily conserved motifs. This method mainly suffers from the lack of fully sequenced genomes. Another common group of motif algorithms consists of enumerative methods. These algorithms count the number of instances of putative motifs of a specific length and score these words based on some statistical criteria. Several of these algorithms are discussed below in more detail.

One of the early enumerative methods was described by Galas *et al.* (1985), who used *Escherichia coli* gene promoters that were pre-aligned relative to the transcription start site. By scanning the multiple alignment of promoters for columns of words of some small fixed length, Galas *et al.* sought a word with a small Hamming distance from the other words in the column and then identified the word with the best Hamming score from all of the sequences in the column as the consensus representation of a motif. Thus, Galas *et al.*'s motif finding closely resembles the Closest Substring solution. This method could not handle motifs with lengths longer than 7 nucleotides and was very dependent on having an appropriate pre-alignment of the motifs within the promoters.

A more general, enumerative motif-finding algorithm has been provided by Brazma *et al.* (1998) who used exhaustive enumeration to place all promoter subsequences of a given size and given degree of degeneracy into a modified suffix tree. Such a tree contains as many levels as the maximum size of the words being examined. The branches from each node in the tree represent letters for each nucleotide or combination of nucleotides according to the IUPAC specification (A, C, G, T, W = A or T, S = G or C, R = A or G, Y = C or T, K = G or T, M = A or C, B = C or G or T, D = A or G or T, H = A or C or T, V = A or C or G, N = A or C or G or T). Nodes represent words that are spelled out by following the branches from the root to a given node. Sequences are placed in the tree by following the branches in a depth-first manner. Each node and leaf represents the word that is created by the labels of the branches taken to get to that node or leaf. Frequency counts are kept at every node and leaf of the number of different words that have been used to visit that node or leaf. Brazma *et al.* made such trees for the subsequences of promoters from a group of co-regulated genes and for the subsequences of promoters from all other genes of yeast

(*Saccharomyces cerevisiae*). For individual word patterns, they compared the counts of matches to the pattern by the subsequences from the promoters of co-regulated genes to the counts of matches to the pattern by the subsequences from the promoters of other genes. In this way they were able to find motifs that were over-represented in the promoters of groups of co-regulated genes compared to the promoters of other genes. This method is an improvement over the work by Galas *et al.* (1985) because it does not require that the promoters be aligned in any way, and the motif patterns found by Brazma *et al.* can occur at widely scattered positions throughout the promoter.

Sagot (1998) and Vanet *et al.* (2000) have described a slightly different use of more compact suffix trees for clustering related promoter subsequences. They use suffix trees in which the nodes only have four branches, one representing each nucleotide (A, C, G and T). In order to allow matches between all words that are separated by some maximum Hamming distance, *d*, the suffix trees are built by associating a specific maximum number of allowable mismatches with the insertion of a word into the tree. As the tree is being traversed in order to match the word to the leaves of the tree, if a branch does not match the current position in the word, the branch may be followed if at least one mismatch may still be allowed, and the number of subsequently allowed mismatches are decreased by one. This type of suffix tree uses less space than the ones made by Brazma *et al.* (1998), but each leaf will contain more instances of words from the promoter sequences. Interestingly, the word that is represented by each leaf of such a suffix tree is the solution to the Closest String problem for the set of words that provided matches to that leaf and a distance value of *d*. Vanet *et al.* used this technique to find motifs in the promoters of the bacteria *Helicobacter pylori*. They defined motifs as the leaf words of suffix tree leaves that contained a statistically significant number of matches. Significance was based on the expected number of words that should match a given leaf word given the GC content of *H. pylori* promoter sequences. This algorithm produces naïve motifs because it asserts that all sequences that have a Hamming distance of *d* from the motif defined by a leaf word are equally likely to be instances of the motif. Sets of instances of actual motifs are more likely to show more haphazard variation among the positions in the motif.

An novel combinatorial motif finding algorithm is the graph-based method of Pevzner and Sze (2000). They describe the motif-finding problem in terms of finding the maximal clique in a graph. A clique is a subset of vertices in a graph such that every vertex in the subset is connected to every other vertex by an edge. Each *l*-length word in each of the promoters under consideration is treated as a vertex in a graph. Vertices are connected by an edge if the Hamming distance between their words is 2*d* or less. Any such clique implies that there is a Closest String solution with distance of *d* to each of the words of the clique. The Closest String solution to a clique that contains a vertex from each promoter is a solution to the motif problem. Finding a maximal clique in a graph is an NP-hard problem. A *k*-clique is a clique where the vertices belong to distinct subsets (*i.e.* promoters) and where the clique has vertices from at least *k* subsets. Finding *k*-cliques is similar to motif finding, and it is also an NP-hard problem. To make the search easier, Pevzner and Sze first efficiently discard all vertices that are not possibly members of a *k*-clique. The remaining vertices are greatly reduced, and it is then easier to find the *k*-clique solution if it exists.

In order to bring standardization to motif finding, Pevzner and Sze (2000) described a motif finding challenge problem, Given a canonical motif sequence, *m*, of length *l*, place in each of 20 promoter sequences of length 600 a random neighbor of *m* that differs at no more than *d* positions from *m*. Such sets of motifs are referred to as (*l, d*)-motifs. Pevzner and Sze pointed out that many probabilistic motif-finding algorithms such as CONSENSUS (Hertz and Stromo, 1999), Gibbs sampler (Lawrence *et al.*, 1993) and MEME (Bailey and Elkan, 1995) have difficulty solving this problem for (15, 4)-motifs which they called the challenge problem. However, Pevzner and Sze's clique-based algorithm is able to find (15, 4)-motifs.

Buhler and Tompa (2002) also answered the challenge problem with a probabilistic enumerative approach to motif finding that involves random projection. The basic idea behind Buhler and Tompa's method is that given a set of instances of a motif there will be a several subsets of these sequences that do not vary from each other for some small number of positions, *p*. Their algorithm examines every *l*-length word in every promoter at *p* randomly chosen positions and sorts the words into bins based on the characters at the *p* positions. The contents of any bin that contain a statistically significant number of words are then used to seed an EM-based motif finding algorithm such as MEME (Bailey and Elkan, 1995) or Gibbs sampler (Lawrence *et al.*, 1993). Significance is based on the expected number of random words that would match the given projection under an appropriate background model. This random projection-based selection, followed by EM analysis, works well in finding (14, 4), (16, 5) and (18, 6) planted motifs which are more difficult than Pevzner and Sze's (2000) original challenge problem. However, there are (*l, d*)-motifs that Buhler and Tompa suggested could not be easily solved by any algorithm.

Buhler and Tompa (2000) published a formula that can be used to calculate the expected number of sets of instances of (*l, d*)-motifs that can be found in a each promoter of a set of promoters:

$$p_d = \sum_{i=0}^{d} \binom{l}{i} (0.75)^i (0.25)^{l-i} \qquad (1)$$

$$E(l,d) = 4^l \left(1 - (1 - p_d)^{n-l+1}\right)^t \qquad (2)$$

where p_d is the probability of a random *l*-length word having a distance of *d* or less from some unknown canonical motif, *E*(*l, d*) is the expected number of sets of (*l,* d)-motifs that will be found by chance in each of *t* promoters each of length *n*. High *E(l, d)* values therefore indicate (*l,* d)-motifs that are difficult to distinguish from common random sets of sequences that appear to be (*l,* d)-motifs, and low *E(l, d)* values represent (*l,* d)-motifs that are rarely occur by chance and that are likely to be biologically significant.

Table 1 shows the expected number of sets of subsequences from a set of 20 randomly generated promoters that will meet the specified (*l, d*)-motif criteria (*i.e.* one subsequence or word from each promoter that has a Hamming distance of *d* or less from some canonical motif sequence). For some values of *l* and *d*, the (*l, d*)-motif problem will be essentially impossible since there will exist many solutions that occur by chance. For example, in a set of 20 promoters, each of length 800, 55 sets of (9, 2)-motifs are expected to be found. However, in that same set of 20 promoters, a (9, 1)-motif is very rare (Table 1). Thus, if a (9, 1)-motif is found in a set of 20 promoters, it is unlikely to have occurred by chance, and a researcher may have confidence that it is biologically significant. However, if a (9, 2)-motif is found, there is some possibility that it has occurred by chance, and a researcher may be less willing to believe that it is biologically relevant without additional evidence. There is a distinct transition point from uncommon (*l, d*)-motifs to common (*l, d*)-motifs for each motif length shown in Table 1, (9, 1) to (9, 2), (10, 2) to

Table 1. Expected number of sets of (*l, d*)-motifs in 20 promoters each 800 nucleotides in length.

	Distance (*d*) Between Instances of (*l, d*)-Motif					
Motif Length (*l*)	0	1	2	3	4	5
9	1.02e-45	3.99e-17	55.37	260311	262144	262144
10	3.70e-57	1.98e-27	9.42e-06	290459	1.04e+06	1.04e+06
11	1.32e-68	5.29e-38	1.45e-14	207.20	3.99e+06	4.19e+06
12	4.69e-80	1.06e-48	3.49e-24	5.22e-05	1.59e+06	1.67e+07
13	1.66e-91	1.82e-59	3.90e-34	2.21e-13	315.42	5.32e+07
14	5.90e-103	2.75e-70	2.95e-44	1.64e-22	7.66e-05	4.68e+06
15	2.09e-114	3.77e-81	1.73e-54	5.50e-32	6.14e-13	237.60

(10, 3), (11, 2) to (11, 3), (12, 3) to (12, 4), (13, 3) to (13, 4), (14, 4) to (14, 5) and (15, 4) to (15, 5). These transition points are seemingly the dividing points between (*l, d*)-motifs that can be found and (*l, d*)-motifs that cannot be found by combinatorial methods. This sharp division between easy-to-find and difficult-to-find (*l, d*)-motifs is due to the degree of difference that may occur between instances of (*l, d*)-motifs. With a (12, 3)-motif, two instances of the motif may have a Hamming distance of 6 (50%), but with a (12, 4)-motif, instances have a Hamming distance of as much as 8 (66%). This also means that any two instances of a (12, 3)-motif must share at least 50% of their sequence, but two instances of a (12, 4)-motif may only have 33% of their sequence in common. As the amount of common sequence between two instances of some (*l, d*)-motif goes down, the probability that two random sequences could be instances of that motif goes up, and thus, random solutions to the (*l, d*)-motif become more probable.

This thesis will describe a combinatorial motif-finding method that expands the sensitivity of the combinatorial class of motif-finding algorithms. The method that will be described here is capable of discovering motifs that are theoretically unidentifiable by other combinatorial methods. The increased sensitivity of this new method is due to two innovations. First, the computational definition of a motif is modified to account for the biological nature of motifs. The motif-space occupied by true motif instances is more restricted than the motif-space that is suggested by the (*l, d*)-motif model. This change suggests that biological motifs are easier to find than (*l, d*)-motifs. Second, the motif-finding problem is decomposed into the problem of finding smaller subsets of instances of simple motifs and then recombining the subsets to form a larger set of instances of a more complex motif. Decomposition of a complex problem into smaller solvable problems is a classic computer science technique that is suggested in this instance by the biological definition of motifs.

MOTIVATION

The methods of Galas *et al.* (1985), Sagot (1998), Vanet *et al.* (2000), Pevzner and Sze (2000) and Buhler and Tompa (2002) all use the (*l, d*)-motif definition either directly or indirectly. This is a biologically unintuitive notion of motifs. This definition is very convenient for formulating motif-finding algorithms in computer science terms, but it overstates the likely difficulty of finding motifs in biological sequences. The reason for this assertion is that biological motifs are not completely free to vary at every position in the motif. While even biologists abstract DNA sequences to strings of A's, C's, G's and T's, DNA is a long, linear molecule. The physical characteristics of DNA vary along its length depending on the sequence of nucleotides that make up the molecule. In the context of a DNA polymer, a motif is bound by its protein transcription factor by means of ionic and hydrogen bonding. Some of these bonds will be essential, and others may be helpful for transcription factor binding but not necessary. Additionally, there will be some positions in a motif where a given nucleotide may sterically interfere with the binding of a transcription factor. Thus, in a biological context, variation at some positions in a canonical motif sequence will be completely or somewhat constrained, but the (*l, d*)-motif allows for no such constraints. However, any position in an (*l, d*)-motif can vary, and any sequence that is a distance of *d* from the canonical (*l, d*)-motif is considered an instance of the motif even if that sequence contains a non-canonical nucleotide at a position of the motif that must remain fixed for biological functionality. This is a failing of the (*l, d*)-motif model. Figure 1 represents this presumed relationship between this biological notion of a motif and the (*l, d*)-motif definition. For (*l, d*)-motifs, the canonical motif is found in the center of the circle and is the Closest String solution for the set of sequences that are contained within the circle. The center motif may also be a canonical motif for the biologically constrained motifs in the gray ellipses, but not every motif that is a distance of *d* from the canonical motif will be an instance of the biologically constrained set of motifs. Importantly, each distinct subset of the biological motif space may have a distinct *d'* Closest String solution where *d'* ≤ *d*, and these solutions may be different from, although still related to, the canonical motif for the (*l, d*) space.

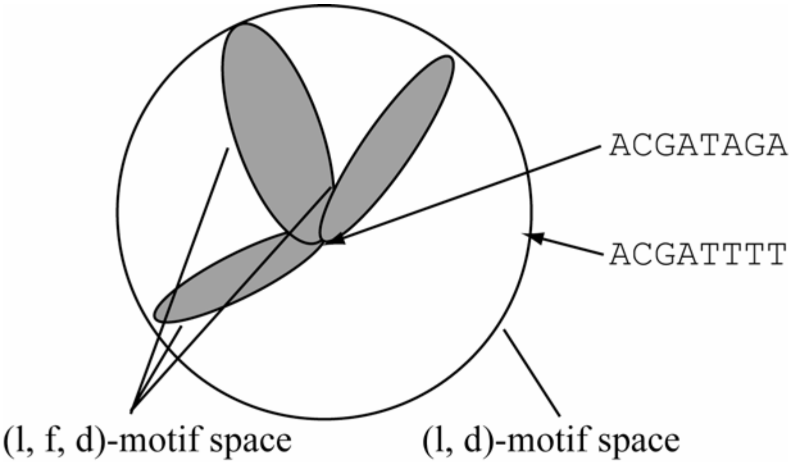

Figure. 1. A two-dimensional representation of *l*-dimensional sequence space. The position of the canonical sequence for this motif, ACGATAGA, is shown to reside in the center of the space. The sequence, ACGATTTT, has a hamming distance of 3 from the canonical motif, and this sequence is at the edge of this hypothetical (8, 3)-motif.

There are several new ideas about motif space that are suggested by the biological motif definition above. First, the (*l, d*)-motif problem may be much harder than finding real biological motifs. For example, if a few positions within a real motif are essentially invariant, then the (*l, d*)-motif problem may be recast as an (*l, f, d*)-motif problem where *f* indicates the number of relatively invariant positions within the motif. Table 2 shows how small values of *f* decrease the likelihood of finding such a motif by chance in a set of promoter sequences. While it is common to find a (9, 0, 2)-motif [(9, 2) in (*l, d*)-motif terms] by chance, it is much less likely to find a (9, 2, 2)- or (9, 3, 2)-motif by chance. In fact, while it is common to find (9, 0, 2)-, (10, 0, 3)-, (11, 0, 3)-, (12, 0, 4)-, (13, 0, 4)-, (14, 0, 5)- and (15, 0, 5)-motifs by chance in a set of 20 randomly generated promoters (Table 2), it is uncommon to find (9, 2, 2)-, (10, 4, 3)-, (11, 2, 3)-, (12, 4, 4)-, (13, 2, 4)-, (14, 3, 5)- and (15, 1, 5)-motifs by chance in those same promoters. Although Buhler and Tompa (2002) assert that some (*l, d*)-motifs cannot be found algorithmically, for similarly difficult values of *l* and *d*, the data shown in Table 2 suggests that there are (*l, f, d*)-motifs that can be found computationally. Second, another idea suggested by the biological motif definition is that the sequence space that is occupied by a motif is not uniformly filled (Figure 1). Sequence space filled by an (*l, d*)-motif could theoretically be evenly occupied by real and potential motifs. Biological motifs are likely to be subsets of regions of an (*l, d*)-motif space (Figure 1). Portions of a biological motif space will be constrained by the fixed positions in the motif. This suggests that instances of motifs will be concentrated in the allowable regions of biological motif space and that it may be possible to identify subsets of the instances of (*l, f, d*)-motifs. Because the maximum distance between motif instances in these subsets can be less than *d,* these subsets may represent easier motif finding targets. This suggests that (*l, f, d*)-motifs can be reconstructed by combining the subsets that define the full motif.

The expectation values given in Table 2 were calculated using Equation 2, but the probability value was determined using a modified version of Equation 1:

$$p_d = \sum_{i=0}^{d} \binom{l-f}{i} (0.75)^i (0.25)^{l-i} \qquad (3)$$

where the variables are interpreted as they were in Equation 1, but *f* is the number of fixed positions in the motif. The effect of adding some number of fixed positions is to reduce the probability that a random subsequence from a promoter will have a distance of *d* or less from a given canonical motif that has *f* specific, fixed positions. A reduction in p_d will cause a corresponding decrease in the calculated expectation value, as seen in Table 2.

Table 2. Expected number of sets of (*l*, *f*, *d*)-motifs in 20 promoters each 800 nucleotides in length.

l and *d* Values	Number of Fixed (*f*) Positions in (*l, f, d*)-Motif					
	0	1	2	3	4	5
9, 2	55.37	3.10	0.077	0.00067	1.50e-06	4.79e-10
10, 3	290459	53517.8	2607.73	20.92	0.016	7.49e-07
11, 3	207.2	3.96	0.027	5.66e-05	2.84e-08	2.44e-12
12, 4	1.59e+06	103226	1269.09	2.22	0.00044	7.79e-09
13, 4	315.42	2.15	0.0048	3.18e-06	5.47e-10	1.98e-14
14, 5	4.68e+06	92430.8	301.36	0.14	1.04e-05	9.62e-11
15, 5	237.60	0.61	0.00052	1.39e-07	1.08e-11	2.14e-16

MOTIF-FINDING PROGRAM OVERVIEW

The program used here is referred to as CM, and it is based upon identifying small, overabundant groups of related words in a single cluster of co-regulated genes and then grouping these small sets of words in a deterministic manner to form a putative motif model. The rules for combining words allow the formation of potentially complex motif models that are combinatorially difficult to find (Tables 1 and 2). CM then examines motif models to determine their *l, f* and *d* parameters, and an expectation value, which is a measure of how often a motif with similar parameters occurs by chance, is calculated for the model. CM outputs motif models as a list of instances of the motif along with statistics such as the expectation value. A flow chart that outlines the steps taken by CM is shown in Figure 2.

The first step in CM is to read from disk all of the promoters from the genes in the cluster that is to be analyzed as well as all of the promoters from the other genes in the experiment. These sets of promoters will be referred to as in-class promoters and out-class promoters, respectively. The out-class promoters consist of the promoters from the genes in the expression experiment that were used for clustering but that are not in the cluster that is being analyzed.

CM decomposes all promoters into their constituent *l*-length words and stores them in suffix trees (Sagot, 1998). The words from out-class promoters are stored in an out-class suffix tree with a depth of *l*. The words from in-class promoters are stored in a separate in-class suffix tree with a depth of *l*. However, the words from the in-class promoters are placed in the in-class suffix tree with some allowed number of misspellings, *d*. For example with one allowed misspelling ($d = 1$), the word AGT would be stored in

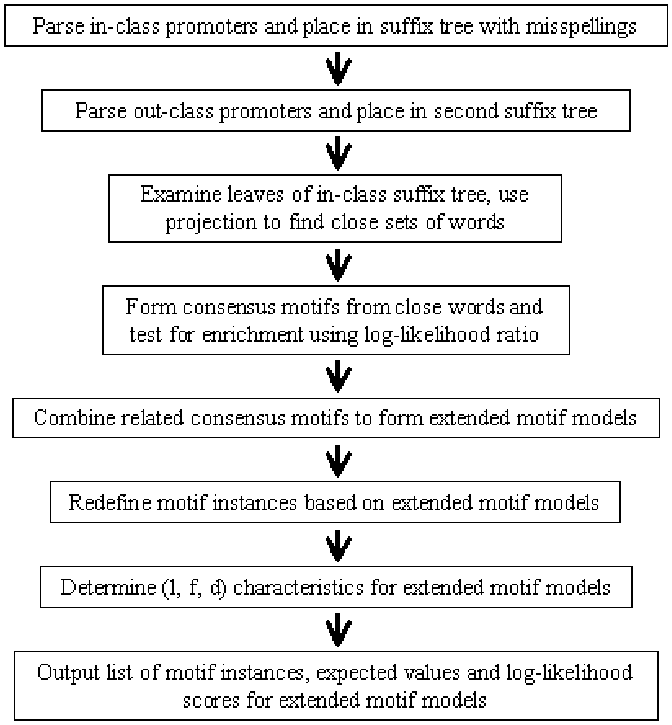
Parse in-class promoters and place in suffix tree with misspellings
Parse out-class promoters and place in second suffix tree
Examine leaves of in-class suffix tree, use projection to find close sets of words
Form consensus motifs from close words and test for enrichment using log-likelihood ratio
Combine related consensus motifs to form extended motif models
Redefine motif instances based on extended motif models
Determine (l, f, d) characteristics for extended motif models
Output list of motif instances, expected values and log-likelihood scores for extended motif models

Figure. 2. Flow chart of steps taken by CM.

suffix tree leaves that correspond to the words AGT, CGT, GGT, TGT, AAT, ACT, ATT, AGA, AGC and AGG. The effect of storing the in-class promoter words with some degree of misspelling in a suffix tree is to simultaneously discover all sets of words in the in-class promoters that have a distance from each other of 2*d*. The set of words found in a leaf form a clique of words, with between-word distances of 2*d* or less. Each leaf of the suffix tree is represented by a leaf-word, which is the word that would be placed in that leaf without any misspellings. Leaf-words represent the central word of the clique of words that have been placed in the leaf, and leaf words have a Hamming distance to any other word in the clique of *d* or less.

Sagot (1998) and Vanet *et al.* (2000) have used suffix trees created in this fashion to find cliques of related words, and they allowed these cliques to describe (*l, d*)-motifs. Figure 1 illustrates the relationship of a clique of words stored in a suffix tree leaf. The canonical leaf-word for that clique is at the center of that diagram. However, considering the biological definition of motifs, such cliques may be too general.

In order to ensure that too many words are not used to define a motif, CM uses projection to identify subsets of related words from within the words found at a leaf. Whereas Buhler and Tompa (2002) used projection with a few fixed positions to identify possibly related words from a set of promoters, CM uses projections to mask a few positions that vary within the clique of words stored in suffix tree leaves. In CM, the number of positions that are masked is typically the same as the number of misspellings that were allowed when forming the in-class suffix tree. The projection technique described by Buhler and Tompa is technically the same as that used by CM, but sets of words identified in each case are related to each other by different degrees of similarity (*i.e.* sets of words that match with some small degree of misspelling). In CM, the goal is to find subsets of related words within a clique of words found in a suffix tree leaf.

For illustration, Table 3 shows a clique of words found in a hypothetical suffix tree leaf. Three subsets of words that were found using projection are shown along with the positions that were masked. While the set of words found in a suffix tree leaf form a clique with a maximal distance of $2d = 6$ between members of the clique, a subset of words found by projection within a leaf form a clique with a maximal distance of $d = 3$ (the number of projected positions) between each member.

After using projection to identify subsets of words from within a leaf of the in-class suffix tree, CM forms a consensus motif from each subset using the IUPAC nucleotide degeneracy codes and then attempts to calculate whether or not the consensus motif is over-represented within the in-class promoters. Table 3 also shows the consensus motifs formed for the three subsets identified from the hypothetical set of words found in a suffix tree leaf. The reason to form consensus motifs is to both simplify and generalize the subsets of words identified within suffix tree leaves. CM then determines how many out-class promoters contain words that match each consensus motif. The percent of in-class motifs that contain words that were used to form each consensus motif and the percent of out-class motifs that match each consensus motif are used to calculate a log-likelihood score for each consensus motif. The log-likelihood score formula,

$$\log\left(\frac{p(\text{consensus motif} \mid \text{in-class cluster})}{p(\text{consensus motif} \mid \text{other clusters})}\right), \qquad (4)$$

does not calculate probabilities, but uses actual counts of in-class and out-class words. If the log-likelihood score is less than a user-defined cutoff, the consensus motif is discarded. All combinatorially-based motif-finding algorithms use some type of measure to determine whether or not a putative motif is enriched within a particular cluster of promoters. Many use background models to estimate the degree of uniqueness of a particular motif. Because CM stores the out-class promoters in a suffix tree, it is possible to quickly traverse the tree to directly measure the number of out-class matches to a consensus motif. Log-likelihood ratios usually are used to compare the probabilities of two similar events within two different populations, and probabilities are usually used to estimate the occurrence of the event within the population. However, because the in-class and out-class populations are

Table 3. The hypothetical words stored in a leaf of a suffix tree formed by CM and subsets found by projection.

Words in Leaf	Projection 1	Projection 2	Projection 3
	* * *	** *	***
AGTAGTG	AGTAGTG	AGTCGAG	AGTTGTG
AGTAGCG	AGTAGCG	ACTCGTG	AGTTAGG
AGTAGGG	AGTAGGG	GCTCGAG	AGTAAGG
ATTAGGG	ATTAGGG	CGTCGTG	AGTTGGG
AATCGCG	AATCGCG	CCTCGAG	AGTGAAG
AATCGAG	AATCGAG	AGTCGTG	AGTCGTG
AGTCGAG	AGTCGAG	CCTCGTG	AGTCAAG
AGTCGAG	AGTCGAG	AGTCGAG	AGTCGAG
ACTCGTG			
GCTCGAG	ADTMGNG	VSTCGWG	AGTNRDG
CGTCGTG			
CCTCGAG			
AGTCGTG			
CCTCGTG			
AGTTGTG			
AGTTAGG			
AGTAAGG			
AGTTGGG			
AGTGAAG			
AGTCGTG			
AGTCAAG			
AGTCAAG			

The canonical leaf-word for this set of words is AGTCGAG. The positions used for projection are shown with asterisks. Derived "consensus motifs" are shown below each projection subset.

completely known, CM uses the actual percent occurrences of the motifs instead of estimates to calculate log-likelihoods.

At this stage CM has a collection of consensus motifs, each of which represents a small clique of words that are enriched within the in-class promoters. As discussed in the previous section, each of these small cliques represents a possible subset of instances of a true motif. The task is to combine these small cliques in a sensible manner to form larger, more complex motifs. The graph-based nature of cliques suggests a natural strategy for combining small motif cliques into larger motif cliques. The mathematical rationale behind this strategy will be explored in the following section. There are three rules for combining motif cliques that will be briefly stated here. Unless it is stated otherwise, the cliques mentioned here have a maximal distance between elements of d, as happens in the special instance of the cliques that CM identifies by projection.

Proposition 1. If two cliques each have a common vertex, then the two cliques may be combined to form a new clique such that the maximal distance between elements in the new clique will be $2d$ or less.

Proposition 2. If two cliques do not contain a common element but they do contain vertices that have a distance of e or less from each other, then the two cliques may be combined to form a new clique such that the maximal distance between elements in the new clique will be $(2d + e)$ or less.

Proposition 3. If two cliques (with elements of length l) do not contain a common element but they do contain elements such that the suffix of an element from the first clique is a prefix of an element from the second clique with m mismatches between the prefix and the suffix and the length of the suffix/prefix is x, then the two cliques may be combined to form a new clique such the maximal distance between elements of the new clique will be $(2d + 2(l - x) + m)$ or less.

This last rule for combining cliques is more complex than the first two rules because it requires actually redefining the vertices and edges, and thus the words, that comprise each clique. For example, if word X = ACGTAAGGA and word Y = CGTAAGGAT are elements of two different cliques, then the promoters from which these words were derived would have to be examined to find the nucleotide immediately to the right of X and the nucleotide immediately to the left of Y. This information would be used to appropriately extend each of these words, and all of the other words from these two cliques would be treated similarly.

In practice, CM forms super-cliques by making use of two timesaving steps. First, when generating larger cliques, CM does not use all vertices of cliques for making comparisons with other cliques. A consensus motif that represents a small clique is treated as an element of that clique. CM only compares consensus motifs when forming larger cliques. This is a convenience that does not lose generality because the information content of an individual word is conserved in its consensus motif. Second, CM does not explicitly extend consensus motifs and the word instances represented by them as the super-cliques are formed. This would be cumbersome and would not aid in clique generation. Instead as large cliques are formed, information about the relative positions of consensus motifs to each other is maintained. If two growing cliques are merged, the positional data for the consensus motifs of one clique is updated relative to the second clique.

The last several steps performed by CM simply deal with generating the final motif models and calculating some statistics that can be used to assess each motif model's significance. First, the in-class and out-class scores for each consensus motif in a super-clique are combined. Redundant matches with any single promoter are only counted once. Log-likelihood ratios are calculated for each super-clique. Next, the actual in-class words that comprise the small clique subsets are extended according to the relative positions of their consensus motifs within the super-cliques. Duplicate words that arise from identical positions within a given promoter are reduced to a single instance. However, multiple unique words from a single promoter are retained and used to define the motif. A list of in-class promoter words is formed for each super-clique. After trimming the ends of uninformative positions, it is this set of words that comprises a motif model. There are numerous ways to represent motif models, but using a simple list of instances of the motif guarantees no loss of information. Finally, each motif model is examined in terms of the (*l, f, d*)-motif definition described in the previous section. The values of *l, f* and *d* are empirically determined. Using these parameters, along with the size of the promoter cluster that is being analyzed and the number of in-class promoters that contain an instance of the presumptive motif, CM calculates an expectation value for the motif model. The equation for finding the expectation value is based on Equation 2 from Buhler and Tompa (2002):

$$E(l,f,d) = \sum_{s=S}^{s=t} \left(4^{l} \left(1-(1-p_{d})^{n-l+1}\right)^{s} \left((1-p_{d})^{n-l+1}\right)^{t-s} \binom{t}{s} \right) \qquad (5)$$

where p_d is found from Equation 3, *t* is the total number of promoters in the cluster, *S* is the number of promoters that contain at least one instance of the presumptive motif and the other variables are the same as in Equation 2. This expectation value can be interpreted as the number of (*l, f, d*)-motifs that one would expect to find in *s* out of *t* randomly generated promoters of length *n* when *l, f, d, n, s* and *t* have the values calculated for the given motif model. CM outputs the words that comprise the motif model, the in-class/out-class log-likelihood ratio and the expected values for the motif-model. Three different expectation values are computed for each model. The first uses a loose *f*-value that is defined as the number of positions in the model where more than 95% of the characters in that position are identical. The loose *f*-value is designed to account for cases where a few incorrect instances have been included in a motif definition. The loose *f*-value discounts these cases.

The second uses a strict *f*-value that only counts a position to be fixed if every character in that position is identical. The use of a strict *f*-value results in a conservative estimate of expected values. Additionally, the use of these two types of *f*-values allows an estimation of the expected value for a motif model with some number of "nearly-fixed" positions. In such a case, the expected value would fall between the expected values calculated using the strict and loose *f*-values. The third expectation value uses an *f*-value of zero and calculates an expected value for an (*l*, *d*)-motif model.

CLIQUE-COMBINING RULES

The clique-combining rules mentioned in the previous section are an integral part of CM and warrant a fuller explanation. While CM makes use of these rules for creating large cliques, it is possible that incorrect or nonsensical cliques will result. Additionally, even though these rules could be used to characterize enlarged cliques as they are formed, CM waits until all cliques have been formed and empirically determines each clique's *l*, *f*, and *d* parameters. Nonetheless, a discussion of these rules will make clear the fact that small, statistically significant cliques can be merged into large, statistically significant cliques that would be difficult or impossible to find by other combinatoric methods.

Although they were stated in the previous section, the clique combining rules will be more explicitly described here along with short proofs. The proofs of these rules are based on the Triangle Inequality Theorem, $|x|-|y| \le |x+y| \le |x|+|y|$. For the rules and proofs below, cliques *A* and *B* are defined as follows. Clique *A* is a clique with a between member distance of *d* or less. The length of the words that are represented by vertices in *A* is l_a. Clique *B* is a clique with a between member distance of *d'* or less. The length of the words that are represented by vertices in *B* is l_b.

Proposition 1. Given two cliques, *A* and *B*, such that neither is a proper subset of the other, and given motifs *a* and *b* that are elements of *A* and *B*, respectively, if *A* and *B* share a common vertex *c* then *a* and *b* must have a Hamming distance of *d* + *d'* or less.

Proof 1. The distance between *a* and *c* is *d* or less because *a* and *c* are members of clique *A*. The distance between *b* and *c* is *d'* or less because *b* and *c* are members of clique *B*. By the Triangle Inequality Theorem, the distance between *a* and *b* must be *d* + *d'* or less.

Rule 1. Given two cliques, *A* and *B*, such that neither is a proper subset of the other, and given motifs *a* and *b* that are elements of *A* and *B*, respectively, if *A* and *B* share a common vertex *c* then combine cliques *A* and *B* to form a new clique with a maximum between member Hamming distance of $d + d'$ or less.

Proposition 2. Given two cliques, *A* and *B*, such that neither is a proper subset of the other, and given motifs *a* and *b* that are elements of *A* and *B*, respectively, if *A* has an element *x* and *B* has an element *y* such that *x* and *y* have a Hamming distance of *e*, then *a* and *b* must have a Hamming distance of $d + d' + e$ or less.

Proof 2. The distance between *a* and *x* is *d* or less because *a* and *x* are members of clique *A*. It is stated that the distance between *x* and *y* is *e*. By the Triangle Inequality Theorem, the distance between *a* and *y* must be $d + e$ or less. The distance between *b* and *y* is *d'* or less because *b* and *y* are members of clique *B*. By the Triangle Inequality Theorem, the distance between *a* and *b* must be $d + d' + e$ or less.

Rule 2. Given two cliques, *A* and *B*, such that neither is a proper subset of the other, and given motifs *a* and *b* that are elements of *A* and *B*, respectively, if *A* has an element *x* and *B* has an element *y* such that *x* and *y* have a Hamming distance of *e*, then combine cliques *A* and *B* to form a new clique with a maximum between member Hamming distance of $d + d' + e$ or less.

Lemma. Given two *l*-length words, *p* and *q*, and that have a Hamming distance of *d*. If *p* and *q* are both extended to the right by *e* characters to form words *p'* and *q'*, then the Hamming between *p'* and *q'* will be $d + e$ or less. This lemma also holds if *p* and *q* are extended to the left by *e* characters.

Proof of Lemma. The *l*-length prefix of *p'* is *p*, and the *l*-length prefix of *q'* is *q*.

The Hamming distance between the prefixes is d. The e-length suffices of p' and q' can only have a maximum Hamming distance of e. Since the concatenation of l-length prefix of p' and the e-length suffix of p' is p' and since the concatenation of l-length prefix of q' and the e-length suffix of q' is q', the Hamming distance of p' and q' is $d + e$ or less.

Proposition 3. Given two homogenous cliques, A and B, such that neither is a proper subset of the other, and given motifs a and b that are elements of A and B, respectively, if the words that define a and b overlap by z characters with m mismatches, then the cliques A and B can be redefined as cliques A' and B' so that the distance between any element of A' and any element of B' will be $d + d' + (l_a – z) + (l_b – z) + m$ or less.

Proof 3. Without loss of generality, assume that a contains a suffix of b and that b contains a prefix of a. Also, let x be an element of A, and let y be an element of B. The distance between a and x is d or less because a and x are vertices in A. Therefore, the distance between the z-length prefix of x and the z-length prefix of a is also d or less. It is stated that the distance between the z-length prefix of a and the z-length suffix of b is m. By the Triangle Inequality Theorem, the distance between the z-length prefix of x and the z-length suffix of b is $d + m$ or less. The distance between b and y is d' or less because b and y are vertices in B. Therefore, the distance between the z-length suffix of y and the z-length suffix of b is also d' or less. By the Triangle Inequality Theorem, the distance between the z-length prefix of x and the z-length suffix of y is $d + d' + m$ or less. By the lemma, if the z-length prefix of x and the z-length suffix of y are extended $(l_a – z)$ characters to the right and $(l_b – z)$ characters to the left to form vertices x' and y', then the Hamming distance between x' and y' will be $d + d' + m + (l_a – z) + (l_b – z)$ or less. Because x and y can be any of the vertices in A and B, respectively, each vertex in both A

and B could be similarly extended to form cliques A' and B' such that the distance between any element of A' and any element of B' will be $d + d' + m + (l_a - z) + (l_b - z)$ or less.

Rule 3. Given two homogenous cliques, A and B, such that neither is a proper subset of the other, and given motifs a and b that are elements of A and B, respectively, if the words that define a and b overlap by z characters with m mismatches, then the redefined cliques A and B by extending the word definitions of the vertices of the cliques in an appropriate manner and combine the newly defined vertices into a single clique so that the Hamming distance between any members of the new clique will be $d + d' + (l_a - z) + (l_b - z) + m$ or less.

ALGORITHM PERFORMANCE WITH YEAST GENE PROMOTERS

While it would not be difficult to simulate an (*l, f, d*)-motif problem, it is a more stringent test of the algorithm to analyze real biological data. Because the yeast genome is fully sequenced and because so much motif analysis has been performed in yeast, data derived from experiments with yeast were used for testing CM. Cho *et al.* (1998) performed a gene expression time course study throughout the yeast mitotic cycle. Tavazoie *et al*. (1999) used the expression data from Cho *et al*. but completely reanalyzed those data. After variance normalizing the expression levels from each gene, the 3000 most variable genes were used for clustering by Tavazoie *et al*. Genes were grouped by *k*-means clustering to form a total of 30 gene clusters. Several of these clusters are heavily enriched with genes that are related to a particular yeast physiology or cellular function, but many of these clusters are not dominated by related genes. This is a weakness of all gene expression studies and will have an effect on all motif finding algorithms. Tavazoie *et al*. used an EM-based algorithm, AlignACE (Hughes *et al.*, 2000; Roth *et al*., 1998), to find transcription factor binding motifs within each of these clusters. The promoters from genes in the clusters from Tavazoie *et al.* have been used for testing CM. Of the 3000 genes clustered by Tavazoie *et al*., only 2842 were used in this study. The difference is likely due to the different annotation versions for the yeast genome. These data have allowed me to use the motifs found by Tavazoie *et al.* as positive controls in my work and to make direct comparison between the results from CM and AlignACE.

Each of the 30 clusters defined by Tavazoie *et al.* (1999) was analyzed by CM using each of four different sets of parameters that described the basic word length and the number of allowed mismatches, (8, 1), (10, 1), (10, 2) or (12, 2). It was empirically determined that the log-likelihood cutoff value had to be adjusted to reduce the number of consensus motifs that are generated by CM. If too many consensus motifs are identified, the motif-clique combining will generate motif models that are comprised of spurious consensus motifs mixed with consensus motifs that represent true motifs. The log-likelihood cutoff values used were 0.7, 1.1 and 1.4 for consensus motifs of lengths 8, 10 and 12, respectively.

As an example of the way that CM combines consensus motifs to form complex motifs, Table 4 shows the set of consensus motifs that were used to form one motif model. All of the consensus motifs in Table 4 represent (10, 1)-motifs. Each has a log-likelihood score of 1.1 or greater and is over-represented by this measure in the in-class gene promoters. The consensus motif, AYTGCGTTTG, is the base motif for this super-clique, and this consensus motif represents the canonical motif for a clique of six 10-character words. The members of this clique have a between member distance of two or less. Each of the cliques represented by the consensus motifs in Table 4 are (10, 1)- or (10, 9, 1)-motifs in (*l, d*) and (*l, f, d*) terms, respectively. The offset scores for each of the non-base consensus motifs in Table 4 describe the positions of those consensus motifs relative to the base motif. Figure 3 shows the multiple sequence alignment of this set of consensus motifs. Based on the “footprint” of this multiple sequence alignment, CM reexamined the promoters of the in-class genes that had matches to these consensus motifs in order to find longer instances for this expanded motif model. The longer instances from the in-class promoters were aligned, and any end positions that CM judged to be uninformative were trimmed from the set of instances. In this case, all of the end positions were judged to be useful. The appropriate 12-character instances from the matching in-class promoters were

Table 4. Set of related consensus motifs that were combined by CM to form a single motif model.

Consensus Motif	Offset	Log-likelihood Score	Number of In-Class Matches	Number of Out-Class Matches
AYTGCGTTTG	0	1.28	6	12
MATTGCGTTT	1	1.24	6	13
ATKGCGTTTG	0	1.21	6	14
ATTGCGTTKG	0	1.18	6	15
ATTGCGTTTK	0	1.18	6	15
TCGCGTTTDT	-1	1.15	6	16

The in-class genes for this cluster number 73, and the out-class genes number 2769.

```
 CAAACGCART
  AAACGCAATK
 CAAACGCMAT
 CMAACGCAAT
 MAAACGCAAT
AHAAACGCGA
************
```

Figure. 3. Multiple sequence alignment of the consensus motifs from Table 4. The *'s indicate the width of the "footprint" analyzed in the promoters that had matches to these consensus motifs. These consensus motifs were used to develop a full motif model that is shown in Figure 4B.

output by CM (data not shown). While the motif models that are output by CM are in the form of a list of word instances, these sets of words have been used to generate logo plots (Crooks et al., 2004; Schneider and Stephens, 1990; http://weblogo.berkeley.edu) for convenient representation of the motif models in this thesis. The 12-character instances for this motif model were used to create a logo plot (Figure 4B).

The consensus motifs described above that were used to form the motif model shown in Figure 4B were combined using the clique combining rules from the previous section. Only Rules 1 and 2 were used for this set of consensus motifs. If the multiple

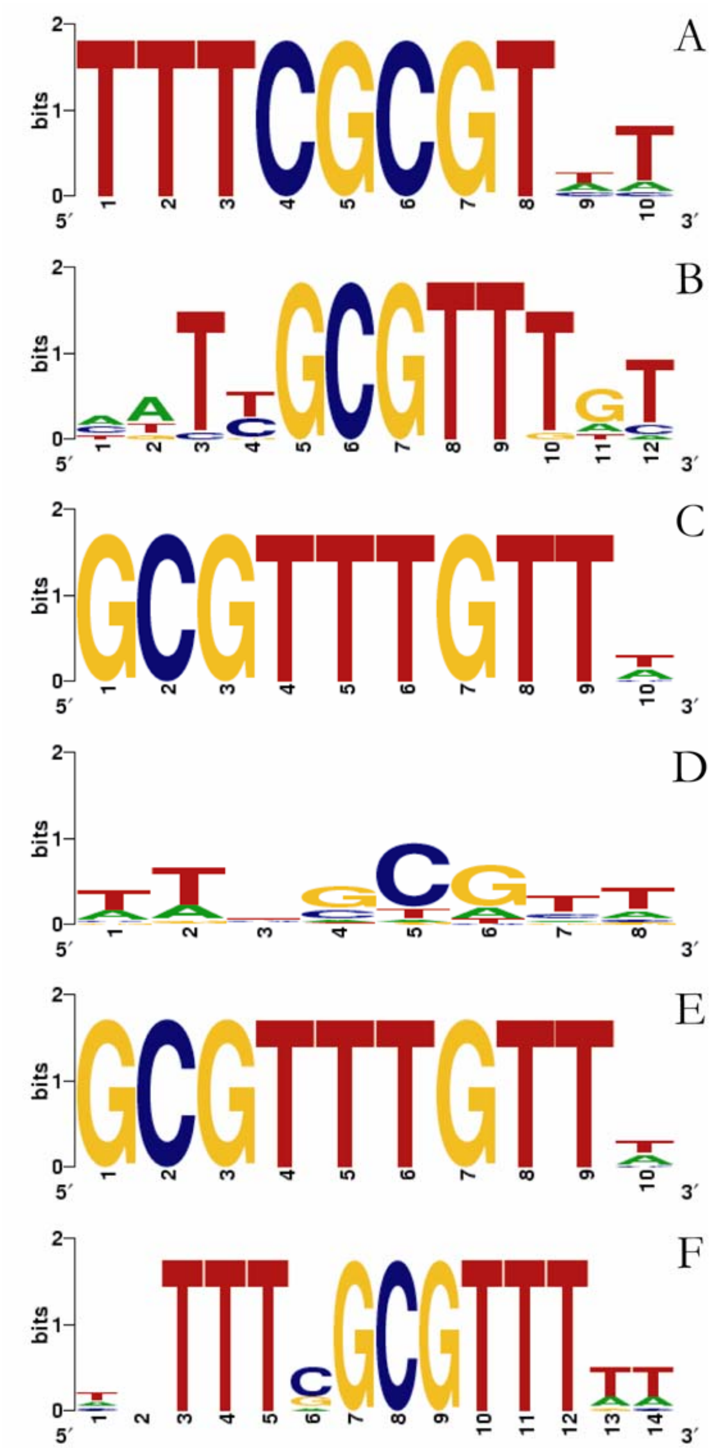

Figure 4. True motif models derived from a cluster of centrosome organizing genes. The gene cluster used to find these motif models is from Tavazoie *et al*. (1999). See Table 5 for parameters used to find these models. These models are representative of a motif that is bound by an unnamed transcription binding factor that Tavazoie et al. called M14a.

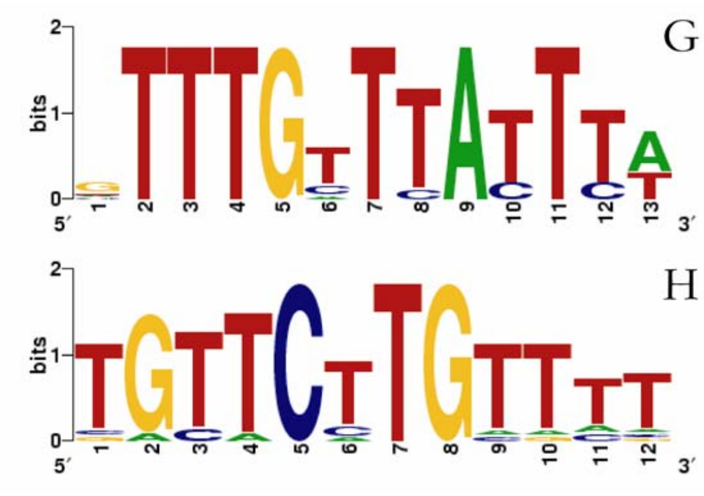

Figure 4 continued.

Table 5. Statistical details of motif models found by CM that correspond to known motifs.

Motif Model	Consensus Motif Word Length	Allowed Mis-matches	In-Class Hits	Out-Class Hits	In-Class Size	Out-Class Size	Log-likeli-hood Ratio	Expected (*l*, *d*)	Expected (*l*, *f*, *d*), Strict *f*	Type of (*l*, *f*, *d*)-Motif, Strict *f*	Expected (*l*, *f*, *d*), Loose *f*	Type of (*l*, *f*, *d*)-Motif, Loose *f*
4A	10	1	13	33	73	2769	1.17	0.013	1.6e-10	10, 8, 1	1.6e-10	10, 8, 1
4B	10	1	14	42	73	2769	1.10	1.6e+07	4.3e+02	12, 5, 3	0.00061	12, 7, 3
4C	10	1	8	24	73	2769	1.10	3e+02	8e-05	10, 9, 1	8e-05	10, 9, 1
4D	10	2	63	545	73	2769	0.64	6.6e+04	6.6e+04	8, 0, 4	6.6e+04	8, 0, 4
4E	10	2	8	24	73	2769	1.10	3e+02	8e-05	10, 9, 1	8e-05	10, 9, 1
4F	12	2	9	29	73	2769	1.07	1.1e+05	5.9e-08	14, 9, 3	5.9e-08	14, 9, 3
4G	12	2	10	20	73	2769	1.28	1e+07	0.032	13, 7, 3	0.032	13, 7, 3
4H	12	2	12	58	73	2769	0.89	23	0.047	12, 3, 2	0.00015	12, 5, 2
5A	10	1	57	105	160	2682	0.96	6.7e+07	6.7e+07	13, 1, 5	5.9e+07	13, 4, 5
5B	10	2	79	326	160	2682	0.61	6.7e+07	6.7e+07	13, 1, 5	6.7e+07	13, 2, 5
6	12	2	50	159	98	2744	0.95	2.7e+08	2.7e+08	14, 0, 6	2.7e+08	14, 2, 6
7A	8	1	88	134	183	2659	0.98	1e-05	1.6e-41	8, 6, 1	1.6e-41	8, 6, 1
7B	10	1	115	183	183	2659	0.96	6.6e+04	6.6e+04	8, 2, 3	6.6e+04	8, 4, 3
7C	10	2	149	523	183	2659	0.62	4.1e+03	4.1e+03	6, 0, 3	4.1e+03	6, 0, 3
7D	12	2	24	9	183	2659	1.59	3	9.7e-22	12, 8, 2	9.7e-22	12, 8, 2
8A	8	1	15	67	58	2784	1.03	1.5e+06	0.16	8, 7, 1	0.16	8, 7, 1
8B	10	1	32	184	58	2784	0.92	5.2e+06	9.2e+05	9, 2, 2	5.7e-10	9, 6, 2
8C	12	2	25	95	58	2784	1.10	2.9e+12	5.8e+12	19, 2, 8	5.8e+12	19, 2, 8
9A	10	1	12	67	58	2784	0.93	4.1e+07	2.6	10, 7, 2	2.6	10, 7, 2
9B	12	2	11	24	58	2784	1.34	1e+10	0.94	14, 7, 4	0.94	14, 7, 4

alignment of the consensus motifs in Figure 3 and if the motif combining rules from the previous section are considered, it can be seen with only a little difficulty that after calculating the extended motif instances, the new super-clique represents a motif of length 12 with a *d*-value of 11 or less. A clique of this length with a *d*-value of 11 would be accompanied by thousands of other similarly loosely related cliques. In fact, the motif model derived from these combined consensus motifs is not a (12, 11)-motif. CM examined the instances that comprise the motif model and found that the maximum distance between any two instances of this motif model is only six. Thus, the maximum distance between the canonical central motif of this clique and any instance is only three. Therefore, the motif model in Figure 4B is a (12, 3)-motif. This example demonstrates that CM is able to generate complex motifs from simple, easy to find motif-cliques.

CM also outputs additional statistical information about derived motif models. Table 5 shows data for the motif that is shown in Figure 4B (see the entry for Motif Model 3B) as well as the other motifs that will be described below. These data include the (*l, d*)-parameters (Consensus Motif Word Length and Allowed Mismatches) for the consensus motifs that were used to form the motif model. CM outputs the in-class and out-class sizes as well as the number of in-class and out-class promoters with matches to the motif model. The log-likelihood score for the overall model is also provided. The (*l, f, d*)-characteristics are determined twice for each motif model. Finally, CM calculates the expected number of (*l, d*)- and (*l, f, d*)-motifs that should be found by chance in the given number of matching promoters in a cluster of the given size. For example, for the motif shown in Figure 4B (Motif Model 3B), there were 14 gene promoters that matched the model from a cluster comprised of 73 genes. The number of (12, 3)-motifs that are expected to be found in this many promoters in a cluster of this size is 5.47e+13. With a strictly calculated *f*-value of 5, the expected number of (12, 5, 3)-motifs that should be found by is 8.37e+3. However, with a more loosely defined *f*-value of 7, the expected number of (12, 7, 3)-motifs that should be found is 1.60e-3.

CM produced a large number of motif models, and not all of those models will be presented here. All of the motif models generated by CM that are related to biological motifs found by Tavazoie *et al*. (1999) are shown in Figures 4 to 9. Statistics related to these confirmed motif models are shown in Table 5. CM also found many motifs that have not been determined to be related to true biological motifs but that were found in a relatively large percentage (> ~15%) of the gene promoters within the gene cluster from which they were derived and that are apparently interesting based on their statistical characteristics (percent of cluster represented, expected numbers and log-likelihood ratios). Statistics for this second set of motif models are shown in Table 6, and the logo plots for these motif models are presented in Figures 10 to 26. A third group of motif models were generated by CM. While this third group of models often have impressive expected-number values and good log-likelihood ratios, each model usually represents putative motifs from a small percentage of the genes from a cluster (< 15%). Descriptions of this third group of models are not given here, but their possible importance will be discussed below.

Some insight may be gained into the types of motif models that CM is able to find in biological sequences. All of the yeast motifs that were found by both Tavazoie *et al.* (1999) and CM are relatively GC rich compared to those motifs that were not found by CM. The Tavazoie *et al*. motifs M1a, M3a, M4, M14b and SCB were not recognized by

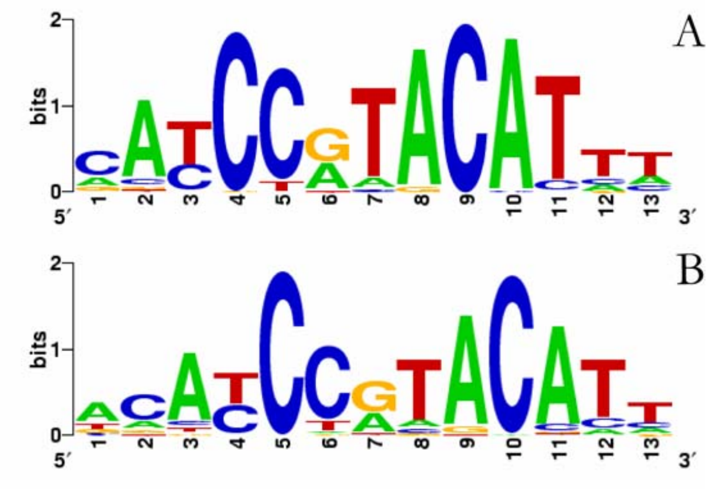

Figure 5. Known motif models derived from a cluster of ribosome function genes (Tavazoie *et al*., 1999). See Table 5 for parameters used to find these models. Models 5A and 5B are representative of a motif that is bound by the Rap1 transcription binding factor.

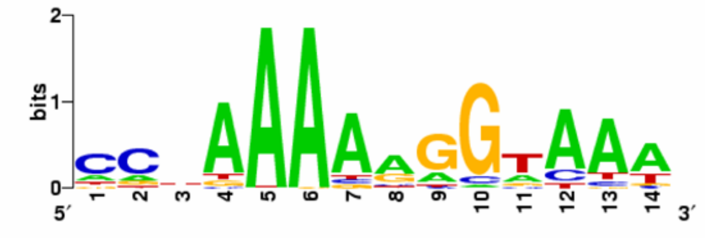

Figure 6. A known motif model derived from a cluster of budding and cell polarity genes (Tavazoie *et al*., 1999). See Table 5 for parameters used to find this model. This model is representative of a motif that is bound by the ECB transcription binding factor.

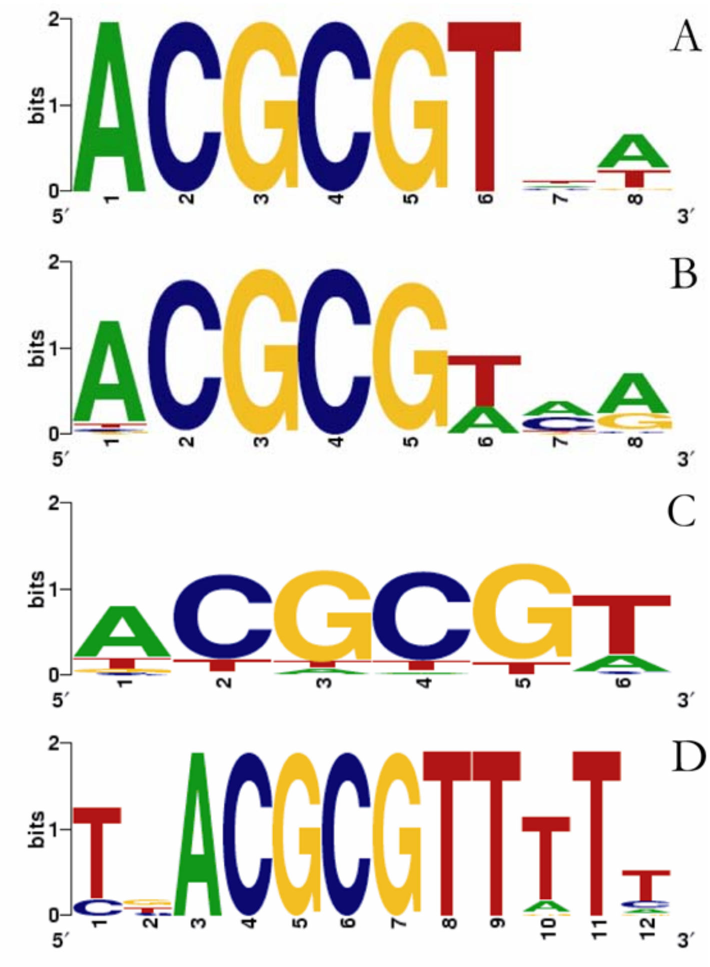

Figure 7. Known motif models derived from a cluster of replication and DNA synthesis genes (Tavazoie *et al*., 1999). See Table 5 for parameters used to find these models. These models are representative of a motif that is bound by the MCB transcription binding factor.

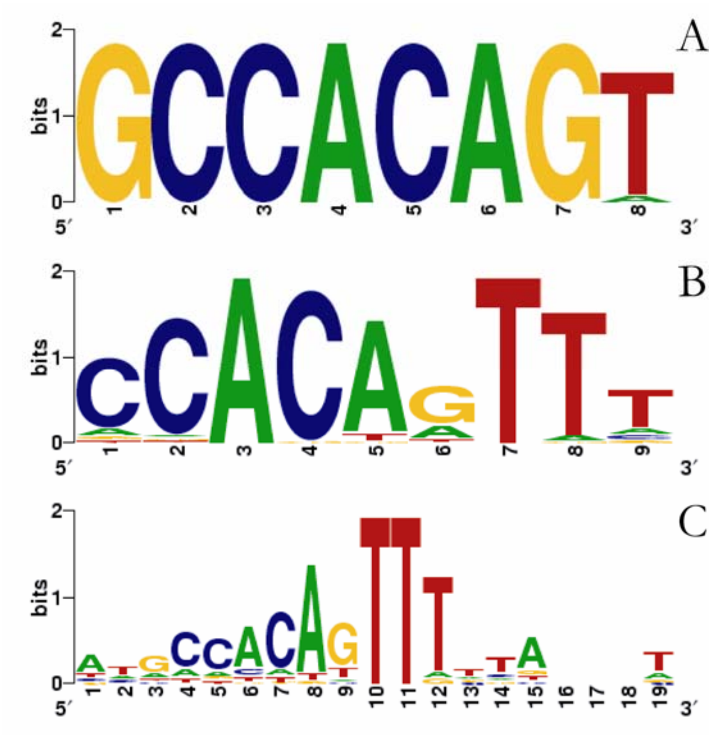

Figure 8. Motif models representing the Met31/32p binding site that were derived from a cluster of methionine and sulfur metabolism genes (Tavazoie *et al*., 1999). See Table 5 for parameters used to find these models.

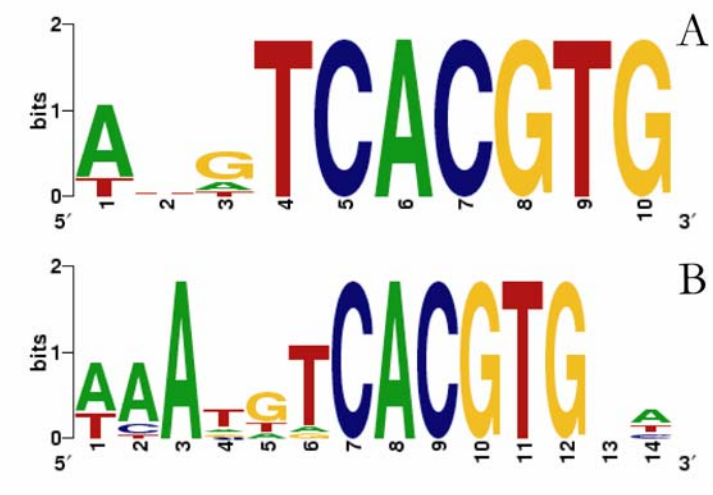

Figure 9. Motif models representing the Cbf1p binding site that were derived from a cluster of methionine and sulfur metabolism genes (Tavazoie *et al*., 1999). See Table 5 for parameters used to find these models.

Table 6. Statistical details of motif models found by CM that do not correspond to known motifs.

Motif Model	Consensus Motif Word Length	Allowed Mis-matches	In-Class Hits	Out-Class Hits	In-Class Size	Out-Class Size	Log-likeli-hood Ratio	Expected (*l*, *d*)	Expected (*l*, *f*, *d*), Strict *f*	Type of (*l*, *f*, *d*)-Motif, Strict *f*	Expected (*l*, *f*, *d*), Loose *f*	Type of (*l*, *f*, *d*)-Motif, Loose *f*
10A	10	2	12	16	160	2682	1.10	5.9e+04	0.013	10, 8, 1	0.013	10, 8, 1
10B	10	2	13	21	160	2682	1.02	1.5e+04	0.00079	10, 8, 1	0.00079	10, 8, 1
11A	10	2	15	44	73	2769	1.11	5.3e+07	6.7e+06	10, 3, 2	2.4e+04	10, 5, 2
11B	10	2	9	19	73	2769	1.25	3.3e+03	0.33	10, 7, 1	0.33	10, 7, 1
11C	12	2	12	58	73	2769	0.89	1.4e+03	2.9	12, 3, 2	0.0094	12, 5, 2
12A	8	1	20	29	183	2659	1.00	4.6e+06	0.11	9, 7, 1	6.4e-06	9, 8, 1
12B	12	2	15	8	183	2659	1.44	1.3	1e-15	13, 9, 2	1e-15	13, 9, 2
13A	10	1	21	73	84	2758	0.98	4.4e+09	7.7e+09	14, 1, 5	7.5e-05	14, 7, 5
13B	10	1	10	27	84	2758	1.08	6.4e+07	6.5e+05	10, 6, 2	6	10, 8, 2
13C	10	1	10	30	84	2758	1.04	1.7e+08	1e+09	12, 4, 4	1.1e+09	12, 5, 4
13D	10	2	18	47	84	2758	1.10	1.7e+08	9.7e+08	12, 4, 4	9.7e+08	12, 4, 4
14A	10	2	36	116	96	2746	0.95	2.6e+05	2.6e+05	9, 1, 4	3.4e+05	9, 3, 4
14B	10	2	15	51	96	2746	0.92	7.3e+06	6.1e+07	10, 4, 3	6.1e+07	10, 4, 3
14C	10	2	20	56	96	2746	1.01	1.6e+08	2.5e+08	11, 2, 3	2.5e+08	11, 3, 3
14D	10	2	23	77	96	2746	0.93	2.6e+05	2.6e+05	9, 1, 4	2.6e+05	9, 1, 4
14E	10	2	15	37	96	2746	1.06	7.3e+07	5.5e+05	10, 5, 2	5.5e+05	10, 5, 2
15A	10	2	15	39	77	2765	1.14	7.2e+06	7.1e+06	9, 5, 2	2.3e+05	9, 6, 2
15B	10	2	16	65	77	2765	0.95	5.5e+07	5.6e+05	10, 4, 2	5.6e+05	10, 4, 2
15C	10	2	24	102	77	2765	0.93	2.6e+05	2.6e+05	9, 1, 4	2.6e+05	9, 1, 4
15D	10	2	16	56	77	2765	1.01	5e+06	3.3e+07	11, 2, 4	8.3e+07	11, 3, 4
16A	12	2	17	49	70	2772	1.14	1.2e+08	9e+05	11, 5, 3	0.017	11, 7, 3
16B	12	2	16	44	70	2772	1.16	1.5e+08	7.6e+08	12, 4, 4	2.2e+05	12, 6, 4
17A	10	2	17	30	83	2759	1.28	6e+07	37	10, 6, 2	37	10, 6, 2
17B	10	2	16	38	83	2759	1.15	6.5e+06	5.9e+07	10, 5, 3	5.9e+07	10, 5, 3
17C	10	2	16	49	83	2759	1.04	6.5e+06	3.7e+07	10, 3, 3	5.3e+07	10, 4, 3
17D	10	2	21	62	83	2759	1.05	1e+06	5.3e+06	10, 3, 4	5.3e+06	10, 3, 4
17E	10	2	23	61	83	2759	1.10	1e+06	1.4e+06	10, 2, 4	1.4e+06	10, 2, 4
17F	10	2	19	49	83	2759	1.11	7.8e+06	2.9e+06	9, 5, 2	2.9e+06	9, 5, 2

Table 6 continued.

Motif Model	Consensus Motif Word Length	Allowed Mismatches	In-Class Hits	Out-Class Hits	In-Class Size	Out-Class Size	Log-likelihood Ratio	Expected (*l*, *d*)	Expected (*l*, *f*, *d*), Strict *f*	Type of (*l*, *f*, *d*)-Motif, Strict *f*	Expected (*l*, *f*, *d*), Loose *f*	Type of (*l*, *f*, *d*)-Motif, Loose *f*
17G	10	2	17	42	83	2759	1.13	1.4e+08	6.2e+06	11, 5, 3	1.1e+04	11, 6, 3
17H	10	2	16	41	83	2759	1.11	6.5e+06	5.9e+07	10, 5, 3	5.9e+07	10, 5, 3
18A	12	2	19	62	61	2781	1.15	1.3e+08	4e+08	12, 2, 4	3.7e+06	12, 5, 4
18B	12	2	17	83	61	2781	0.97	3e+09	0.00013	14, 6, 4	0.00013	14, 6, 4
18C	12	2	20	67	61	2781	1.13	1.2e+08	23	12, 4, 3	23	12, 4, 3
18D	12	2	17	56	61	2781	1.14	3e+09	0.6	14, 5, 4	0.6	14, 5, 4
18E	12	2	18	56	61	2781	1.17	1.3e+08	3.8e+08	12, 4, 4	8.5e+06	12, 5, 4
19A	10	2	17	36	82	2760	1.20	1.4e+08	1.2e+08	11, 4, 3	5.5e+06	11, 5, 3
19B	10	2	22	112	82	2760	0.84	2.7e+05	9.7e+06	9, 4, 3	9.7e+06	9, 4, 3
20A	10	1	12	35	63	2779	1.18	3.1e+06	45	11, 5, 2	45	11, 5, 2
20B	10	1	13	46	63	2779	1.10	6.8e+11	7.3e+11	17, 1, 6	4e+08	17, 4, 6
21A	10	2	17	72	82	2760	0.90	6.4e+06	5.2e+07	10, 4, 3	5.4e+07	10, 5, 3
21B	10	2	16	45	82	2760	1.08	7.7e+06	7.4e+06	9, 5, 2	7.4e+06	9, 5, 2
21C	10	2	22	59	82	2760	1.10	1e+06	5.3e+06	10, 3, 4	2.1e+07	10, 4, 4
22	10	2	15	101	73	2769	0.75	1.2e+08	1.1e+08	11, 4, 3	1.1e+08	11, 4, 3
23A	10	1	16	25	45	2797	0.90	1.7e+07	1.1e+08	12, 3, 5	3e+08	12, 4, 5
23B	10	1	13	61	45	2797	1.12	1e+08	4.3e+08	12, 3, 4	2.4	12, 7, 4
24A	10	1	16	82	63	2779	0.93	3.4e+07	8.2e+02	10, 5, 2	2.4	10, 6, 2
24B	10	2	14	50	63	2779	1.10	1.3e+08	5.9e+08	12, 3, 4	7e+08	12, 4, 4
24C	10	2	19	91	63	2779	0.96	2.1e+05	1.9e+06	8, 4, 2	1.9e+06	8, 4, 2
24D	12	2	24	63	63	2779	1.23	1.3e+08	4.2e+08	12, 2, 4	6e+07	12, 4, 4
25A	10	1	17	89	64	2778	0.92	2e+09	6.3e+08	13, 3, 4	9.5	13, 6, 4
25B	12	2	22	106	64	2778	0.95	1.4e+08	2.6e+08	12, 1, 4	1.7e+08	12, 4, 4
25C	12	2	24	53	64	2778	1.29	1.4e+08	4.2e+08	12, 2, 4	5.4e+08	12, 3, 4
26A	10	1	12	57	49	2793	1.08	2.7e+05	47	11, 4, 2	0.026	11, 6, 2
26B	10	1	18	50	49	2793	1.11	2.7e+05	4.4e+06	9, 4, 3	4.4e+06	9, 4, 3

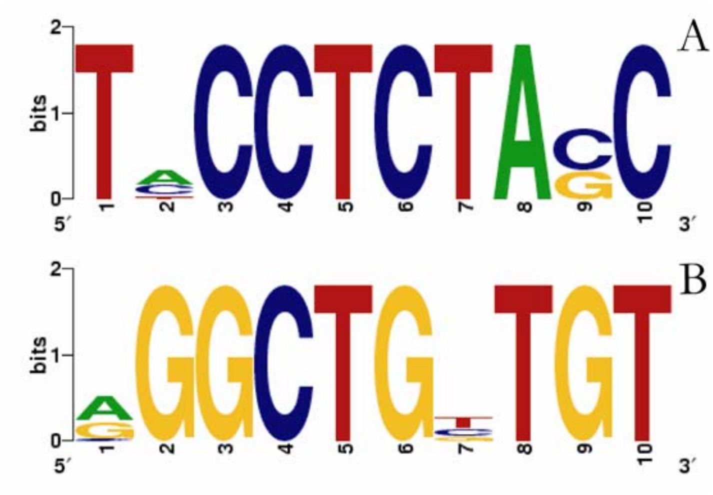

Figure 10. Putative motif models derived from a cluster of ribosome function genes (Tavazoie *et al*., 1999). See Table 6 for parameters used to find this model. These models are not known to represent true motifs bound by a particular transcription factor.

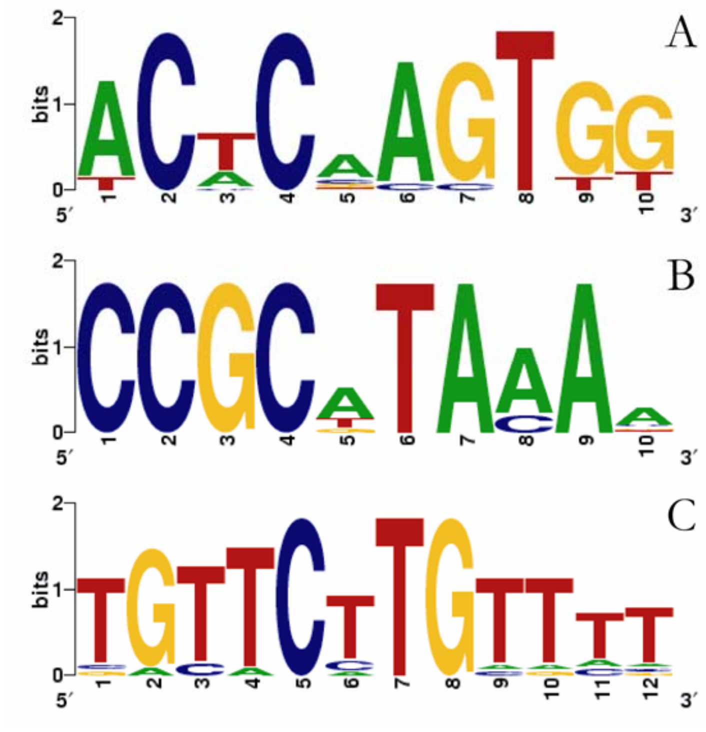

Figure 11. Putative motif models derived from a cluster of centrosome organizing genes (Tavazoie *et al*., 1999). See Table 6 for parameters used to find this model. These models are not known to represent true motifs bound by a particular transcription factor.

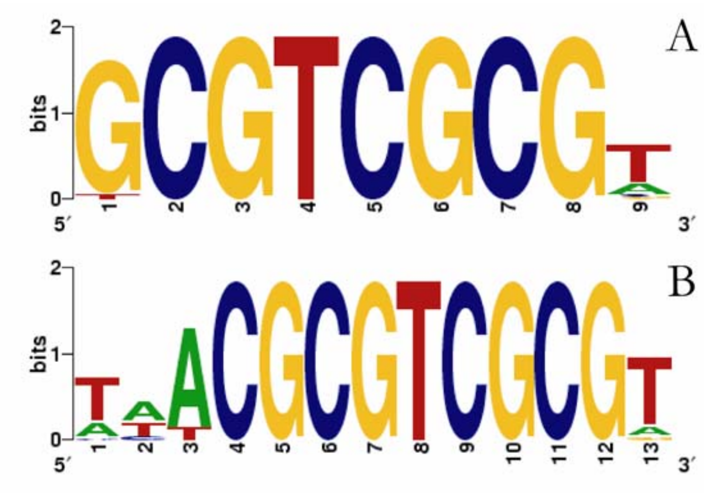

Figure 12. Putative motif models derived from a DNA replication and synthesis genes (Tavazoie *et al*., 1999). See Table 6 for parameters used to find these models. Note that motif models 12A and 12B are variants of the MCB transcription binding sites shown in Figure 7.

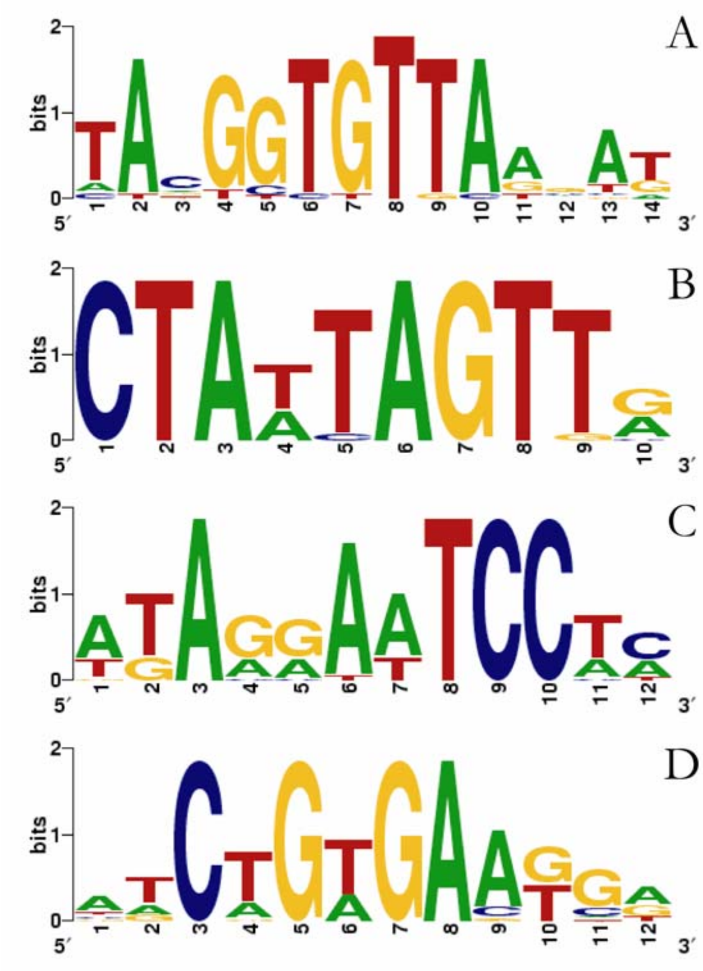

Figure 13. Putative motif models derived from a functionally ill-defined cluster 10 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

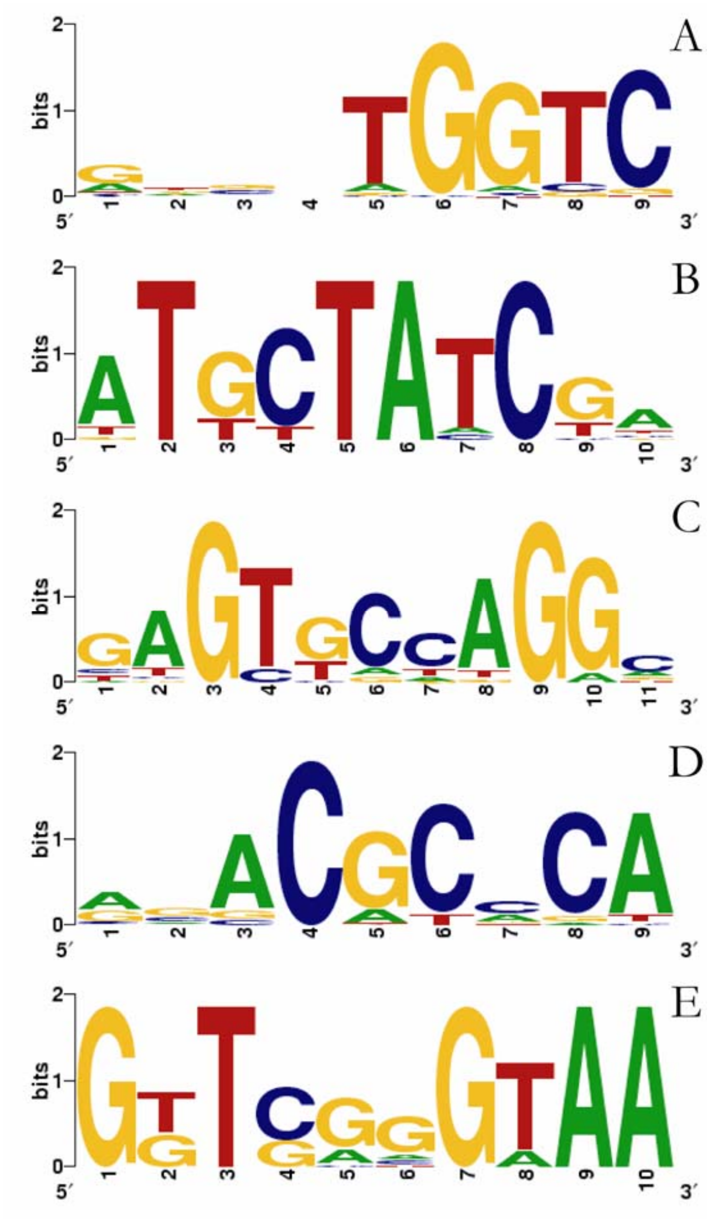

Figure 14. Putative motif models derived from a functionally ill-defined cluster 16 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

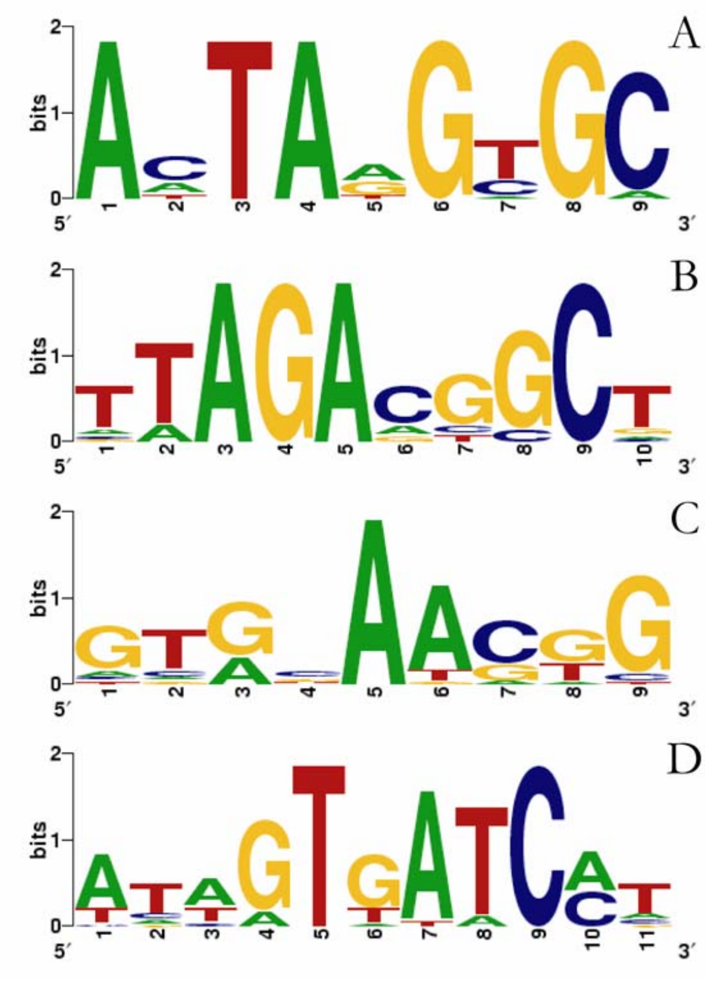

Figure 15. Putative motif models derived from a functionally ill-defined cluster 17 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

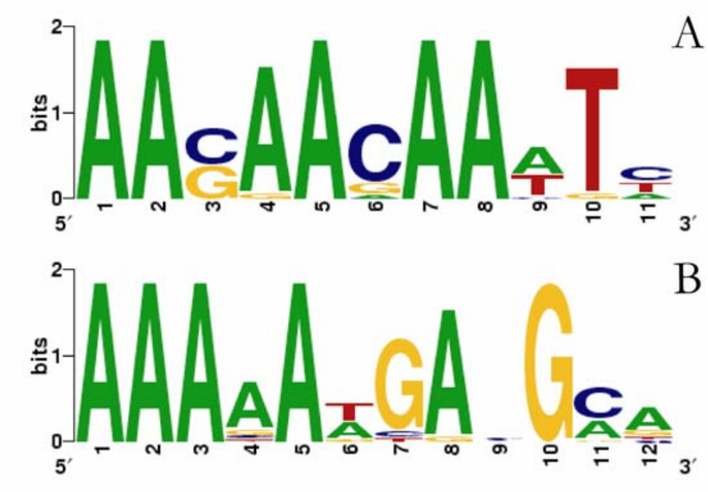

Figure 16. Putative motif models derived from a functionally ill-defined cluster 19 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

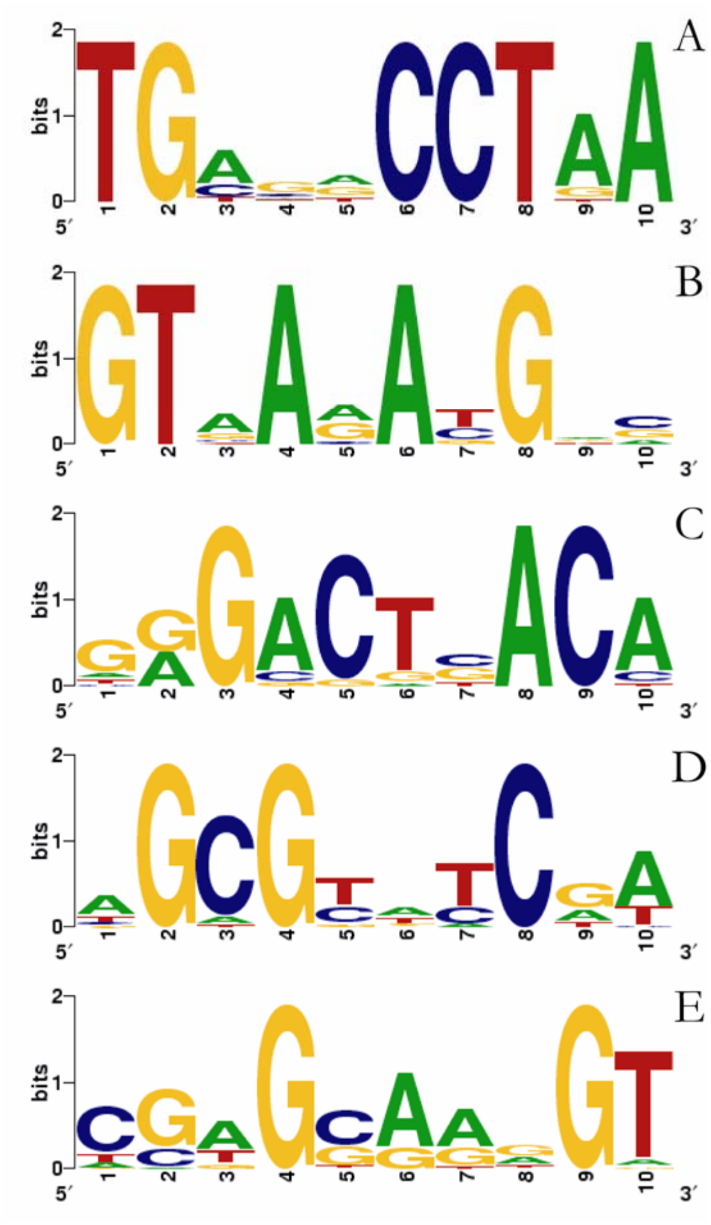

Figure 17. Putative motif models derived from a functionally ill-defined cluster 20 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

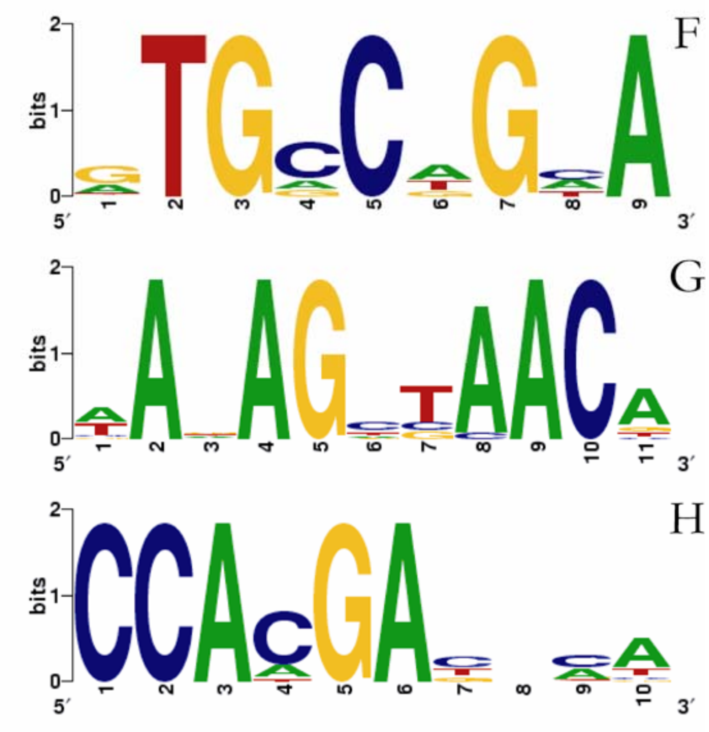

Figure 17 continued.

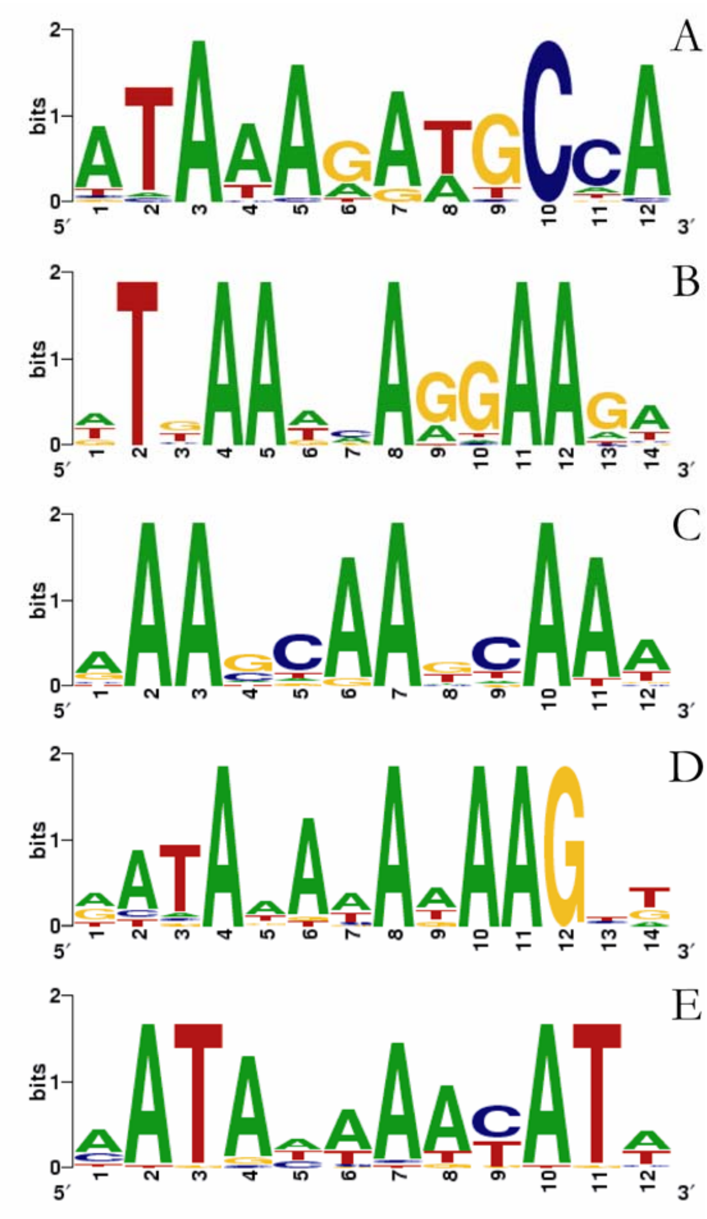

Figure 18. Putative motif models derived from a functionally ill-defined cluster 21 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

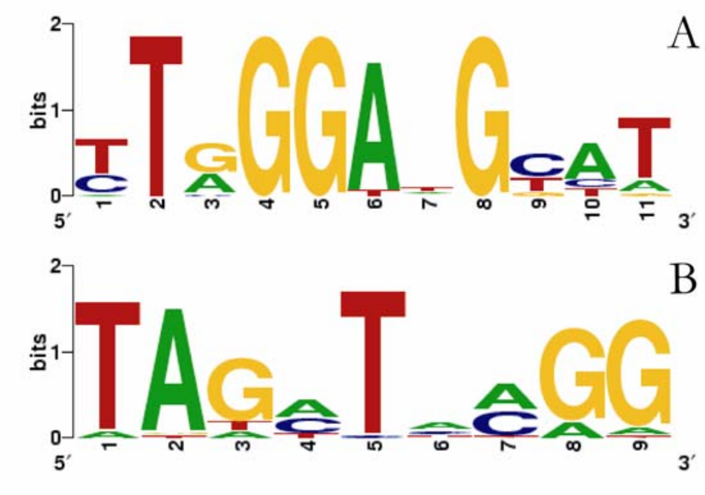

Figure 19. Putative motif models derived from a functionally ill-defined cluster 22 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

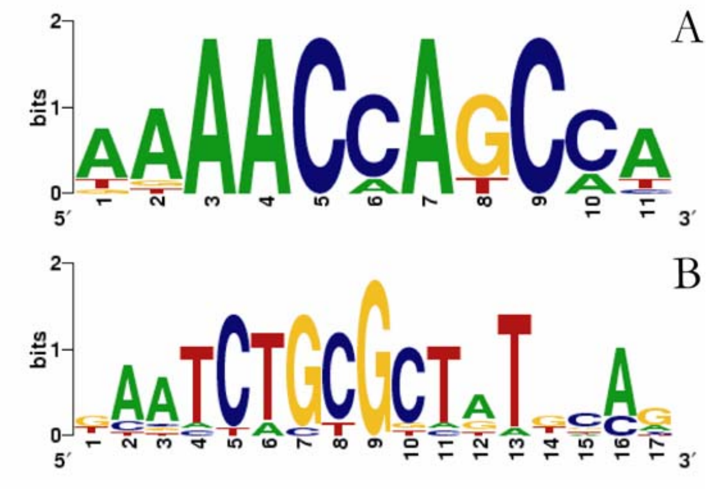

Figure 20. Putative motif models derived from a functionally ill-defined cluster 23 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

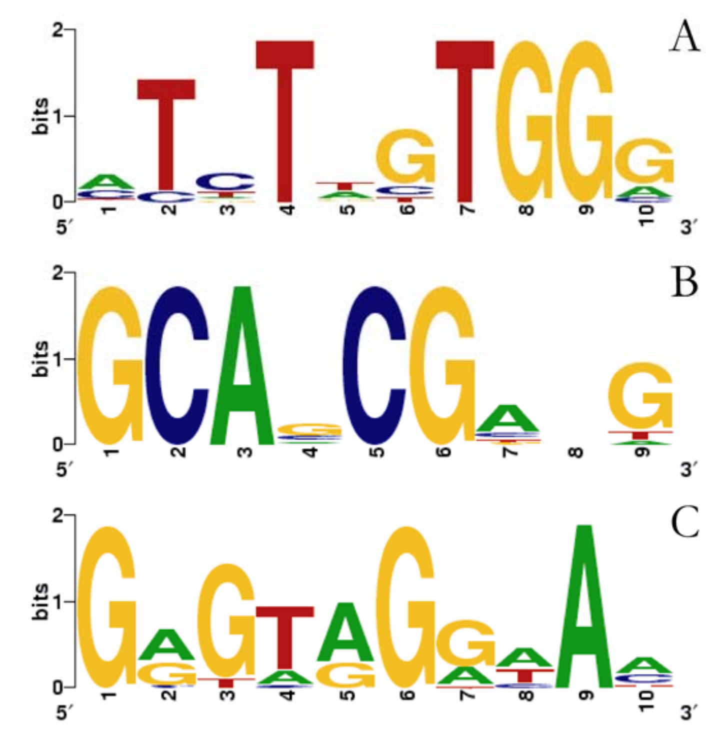

Figure 21. Putative motif models derived from a functionally ill-defined cluster 24 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

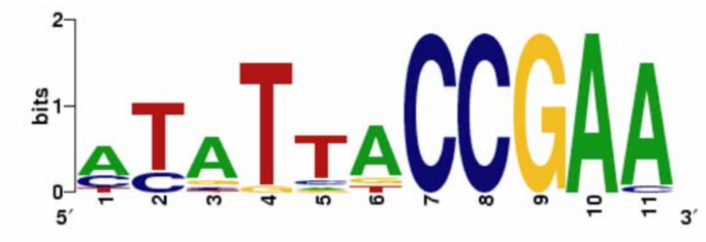

Figure 22. Putative motif model derived from a functionally ill-defined cluster 25 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find this model. This model is not known to represent true motifs bound by a particular transcription.

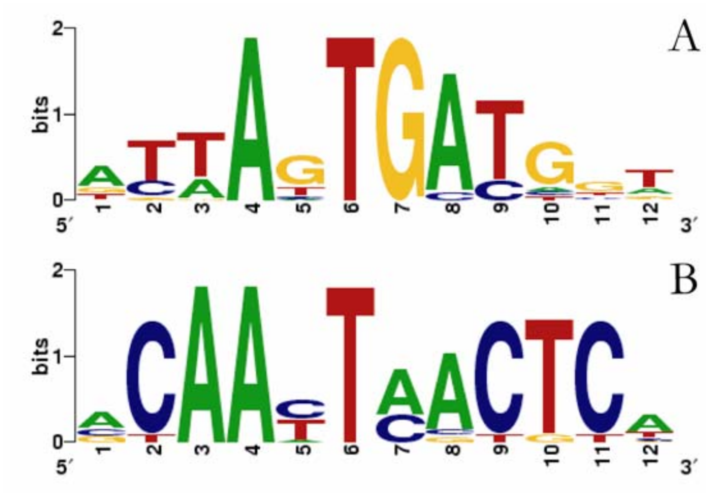

Figure 23. Putative motif models derived from a functionally ill-defined cluster 26 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

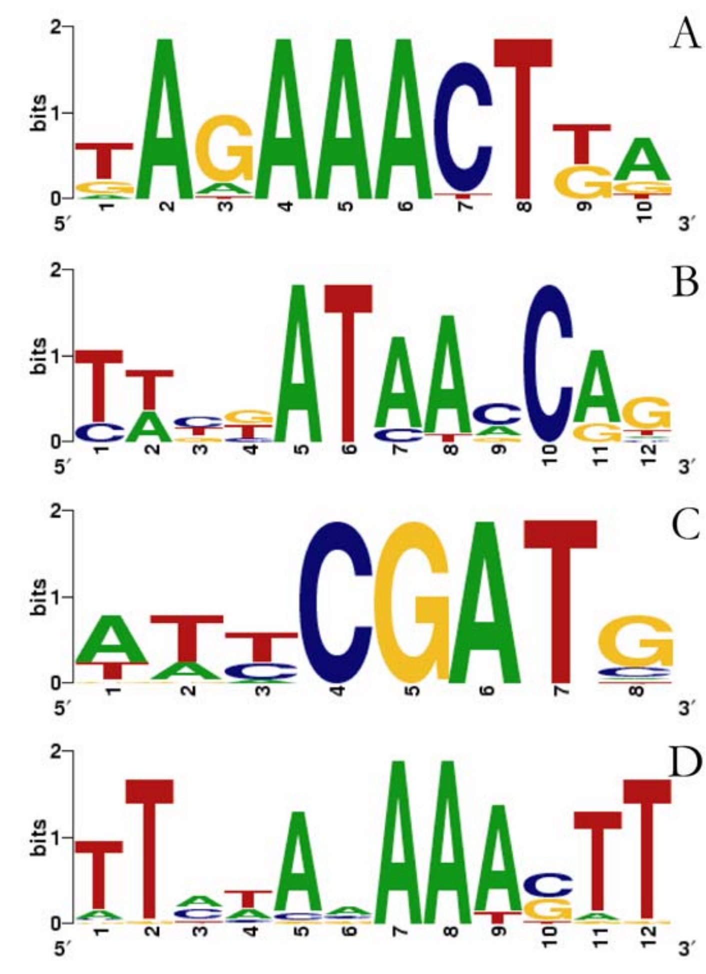

Figure 24. Putative motif models derived from a functionally ill-defined cluster 27 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

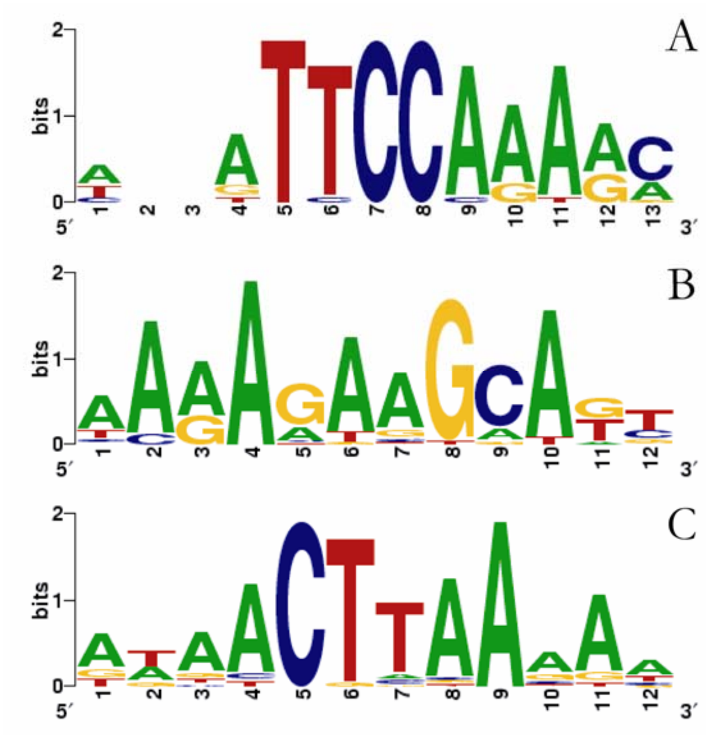

Figure 25. Putative motif models derived from a functionally ill-defined cluster 28 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

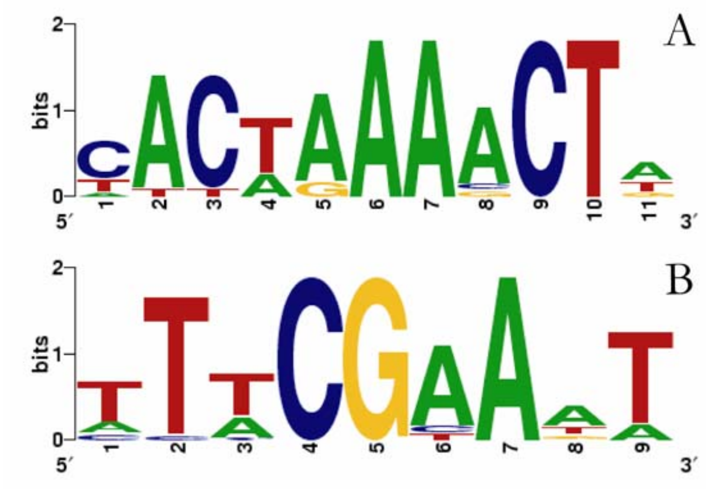

Figure 26. Putative motif models derived from a functionally ill-defined cluster 29 from Tavazoie *et al*. (1999). See Table 6 for parameters used to find these models. These models are not known to represent true motifs bound by a particular transcription factor.

CM, and they are all AT rich. None of the known motifs that CM identified are AT rich. The yeast genome has an overall AT/GC ratio of 62:38. It may be that AT rich motifs are hidden in motif space amongst the relatively higher AT background promoter sequence, and that CM has a limited sensitivity to this level of background noise. However, the Tavazoie *et al*. motifs M3b and STRE, which are GC rich, were also missed by CM, and all of the missed motifs may represent a class of motif that CM will often fail to identify for unknown reasons.

The models generated by CM were usually too specific or too general relative to the motifs found by Tavazoie *et al*. (1999). Of course, this assumes that the Tavazoie *et al*. motifs approximately represent the true number of instances of each of the motifs. Only in a few cases did CM make a motif model that had about as many in-class matches as did the corresponding model from Tavazoie *et al*. (data not shown). Most models matched many fewer or many more promoters than did the Tavazoie *et al*. motifs, and this observation reveals two weaknesses of CM. When CM only finds a few specific consensus motifs, the number of matching promoters is also low, but when CM finds a relatively large number of consensus motifs using more permissive parameters, (10, 2) or (12, 2), there are also a few incorrect consensus motifs that are combined with the biologically related consensus motifs. This over-generalization leads to matches to promoters that do not contain a biological instance of the motif. It is impossible to reduce this inappropriate motif combining because the motif combining parameters that were used for these analyses were as strict as possible. The only way to prevent incorrect combinations would be to not allow any combinations.

The different motif length and mismatch parameters that were tested greatly affect the sensitivity of CM to find biological motifs and the types of models that CM produces (Table 5). Shorter initial consensus motif lengths tend to produce shorter final models. Most of the longest models were generated when the consensus motif length was 12. Not surprisingly, the more complex models (those that were long and had many variable positions) were found when two consensus motif mismatches were allowed. If consensus motif mismatch values were larger than 1 for the 8-character consensus motifs or larger than 2 for the 10-character consensus motifs, the number of consensus motifs with good log-likelihood scores became too large, and consensus motif grouping resulted in combined sets of clearly unrelated sequences (data not shown). Even with the motif lengths and mismatch values of (8, 1), (10, 1), (10, 2) and (12, 2), occasional super-cliques were formed that consisted of distinct subsets of consensus motifs with a few consensus motifs that formed an intersection between the subsets (data not shown).

While the motif models in Figures 4 to 9 are known to represent biological motifs based on the data from Tavazoie *et al*. (1999), CM output many additional motif models. Figures 10 to 26 show motif models that were not identified by Tavazoie *et al*. as being biological motifs, but they were derived from a relatively large percentage (> ~15%) of the gene promoters from the cluster in which they were found (Table 6). In this way, these putative motifs are similar to the known biological motifs that were characterized by CM (Table 5). As with all motif models output by CM, these putative motifs are also enriched within their gene cluster, as evidenced by their log-likelihood values. While many of these putative motifs do not have exceptionally impressive expected values even when using *f*-values, some of the biological motifs from Tavazoie *et al*. also did not have small expected values. The rareness of a motif model can be measured by its expectation value , but a poor expected value alone should not be used to discount the likelihood that a motif model represents a biologically significant set of sequences.

CM found one biological motif that is an apparent variant of a motif that Tavazoie *et al*. (1999) identified. Figures 12A and 12B show an alternate binding site for MCB. This alternate site is a tandem repeat of the primary version of this motif (Figure 7). The reason this motif was missed by Tavazoie *et al*. may have to do with the way that their motif-finding algorithm, AlignACE, masks instances of motif models that it has identified as it iteratively searches for motifs. This alternate MCB binding site has also been noted by Kellis *et al*. (2003).

Other evidence exists that suggests that some of the motifs models in Figures 10 to 26 are related to biological motifs. The TRANSFAC database (http://www.gene-regulation.com) is a database of transcription factors and known motif sequences. This database was examined to determine if any additional CM motifs were representative of biological motifs. The TRANSFAC database contains motifs in two formats, probability matrices and individual binding site sequences. There are many more examples in the individual sequence portion of the database, and these were searched with partial consensus sequences from the CM motif models. Matches to the database were common, but only rarely were these matches absolutely convincing. Many matches were partial, and given the nature of the transcription factor binding sites in the database, partial matches to CM motif models are rarely convincing. The binding-sequences in the TRANSFAC database are often derived from transcription factor footprinting experiments. Such studies often result in long sequences that represent the "footprint" of the transcription factor over the gene promoter even if only a small subsequence within the footprint is required for transcription factor binding. It is that small sequence that represents a motif. Knowing if a match between a CM motif and a long footprinting-derived sequence is significant or not is impossible without conducting laboratory experiments. Nonetheless, a few interesting results can be mentioned. These matches were first recognized during comparisons with the TRANSFAC database, but then original publications were identified. The yeast transcription factor REB1 has been variously described as having a consensus binding site of CCGGGTA or CGGGTRR (Chasman *et al*., 1990; Morrow *et al*., 1989). The consensus sequence of the CM motif 14E is GTTCGGGTAA (Figure 13E), and this is a perfect match to the second consensus binding sequence for REB1. None of genes that are known to be regulated by REB1 are in the cluster in which motif 14E was found. The yeast transcription factor SWI5 has a consensus binding site of ACCAGC (Dorhmann *et al*., 1996), and the CM motif 20A has consensus sequence AAAACCAGCCA (Figure 20A). Of the three genes with known binding sites for SWI5, only CTS1 is in the gene cluster in which motif 20A was found. There may be other biological motifs represented in the CM motifs shown in Figures 10 to 26, but a conservative analysis only yielded the two comparisons given above. Although most of the genes that contributed sequences that have been interpreted as being instances of REB1 and CTS1 have not been previously identified as containing these binding sites, this may be due to the TRANSFAC database being out of date or to the difficulty of finding literature that reports the existence of these binding sites in these genes.

Although CM found many of the biological motifs that were identified by Tavazoie *et al*. (1999), there were some biological motifs that were described by Tavazoie *et al*. but that were not recognized by CM. The converse also seems to be true. There are some apparent biological motifs that CM found that were not found by Tavazoie *et al*. who only published motifs that were known or that they confirmed to be biological. To double check, AlignACE, the program used by Tavazoie *et al*., was used to reanalyze the gene clusters from Tavazoie *et al*., and CM motifs 14E and 20A (Figures 14 and 20) are not found in the AlignACE results (data not shown). A few of the CM motif models in Figures 10 to 26 are similar to motif models that were described by AlignACE. However, most motif models in Figures 10 to 26 are not similar to motifs found by AlignACE. These results suggest that there are different types of patterns that can be recognized by these two algorithms. AlignACE and CM try to find motif models in different ways. AlignACE uses Expectation Maximization to form a probability-based motif model that represents a locally maximal maximum-likelihood score. CM exhaustively enumerates word instances and groups sets of cluster-enriched words that are separated by some small distance in motif space. It should not be surprising that such different strategies would each create different sets of erroneous motif models. What might be surprising is that each of these algorithms may have trouble identifying a common set of biological motifs. Each algorithm likely has a bias against some types of biological motifs. However, the characteristics of the troublesome motifs that make identification by the programs difficult are not known and represent our ignorance of the true nature of transcription factor binding sites.

In total, CM only generated a few hundred motif models for all 30 gene clusters examined and all parameter sets used. The number of (*l, d*)-motif models that should be expected by chance is greater than the actual number of motif models identified by CM. Importantly, when the expectation values were calculated using the empirically derived *f*-

values, the expected number of motifs generally decreased dramatically. Motifs models with small expected values are unusual and may be biologically significant. In fact, most of the biologically verified motif models found by CM do have expected values that indicate that they are rare. Based on the simple (*l, d*)-expected values, motif models similar to those found by CM would be expected to occur hundreds, thousands or hundreds of thousands of times in the yeast gene clusters examined here. There are likely two explanations for the relatively small number of motif models created by CM. The notion of finding small closely related sets of words in motif space and combining them with nearby sets of related words to form motifs is the same idea that leads to the inclusion of the *f*-value in the expectation calculations. It was shown that large *f*-values will be found in rare motifs (Table 2). CM is designed to find motifs that will have some number of fixed positions and should be expected to describe motifs that are rare. However, not all of the motifs found by CM have small expected values. The reason that CM does not generate hundreds of these models is probably due to the selection of consensus motifs that are enriched in a particular gene cluster. The use of consensus motifs that represent enriched sequences is likely to lead to only the formation of unusually common full motif models. A simple change in the log-likelihood cutoff used by CM would greatly affect the number of consensus motifs that CM considers for motif combining and would also affect the number of motif models output by CM.

The purpose for using CM to analyze the promoters of the genes in the clusters defined by Tavazoie *et al*. (1999) was to test CM with biological data and to be able to compare the results from CM's analysis to the results produced by the EM algorithm, AlignACE, used by Tavazoie *et al*. The results are encouraging. CM does find biological motifs in the promoters of yeast genes. Although CM did not find all of the motifs that were identified by Tavazoie *et al*., there were a few biological motifs that were recognized by CM that were not found by AlignACE. Additionally, there were many additional motif models generated by CM that have significant expectation values and log-likelihood scores and that would be good candidates for additional investigation by wet-lab scientists.

The use of biological data to test CM served another purpose as well. CM is designed around the notion of fixed-positions in transcription factor binding sites. Therefore, the recognition of biological motifs, many with fixed positions, by CM is a confirmation of this theoretical underpinning of the algorithm. It is possible to model transcription factor binding motifs as related subsets of sequences surrounding a canonical motif sequence in the manner that is depicted in Figure 1.

Finally, CM uses a motif-combining procedure that is based on clique-combining rules, and these analyses confirmed that in a biological context simple, easy to identify motifs may be combined to form complex, apparently difficult to find motifs. Many of the motif models, both biological and putative, generated by CM have (*l, d*)-motif expectation values that are very large, but when these motif models are describe in (*l, f, d*)-terms, their expectation values are often very small and significant. Buhler and Tompa (2002) suggested that there is a limit to the types of motifs that may be discovered by combinatorial methods. CM is capable of finding motifs that Buhler and Tompa suggest should not be identifiable.

COMPARISON WITH AN (*L, D*)-MOTIF-FINDING ALGORITHM

CM is a combinatorial motif-finding algorithm. It is based on enumeration and is similar to the methods of Brazma *et al*. (1998), Sagot (1998) and Vanet *et al*. (2001). While CM has been shown to perform comparably to the EM-based AlignACE motif-finding algorithm, it is also of interest to compare the results of CM and these other combinatorial methods that identify (*l, d*)-motifs.

To make this comparison, CM was modified into a simple combinatorial program similar to that described by Sagot (1998) and Vanet *et al*. (2001). This program will be referred to as naïve-CM. All words from the in-class and out-class promoters were stored in suffix trees with some small number of allowed misspellings. The result of storage in this type of data structure is that every leaf in the tree corresponds to a Closest String solution to the words that have matched that leaf. Thus, the words that match a leaf form an (*l, d*)-motif. After filling the in-class and out-class suffix trees, each leaf from the in-class tree was compared to the corresponding leaf of the out-class tree. A log-likelihood score was calculated based on the number of genes that contributed matching words to the leaves, and the total number of in-class and out-class genes. In order to reduce the number of spurious results, the (*l, d*)-motifs with log-likelihood scores above a set threshold were sorted by numbers of in-class hits. The sorted results were output as a list of words and associated statistics. Tables 7 to 12 show results from this type of combinatorial analysis of three of the clusters from Tavazoie *et al*. (1999) when using words of length ten and allowing one or two misspellings when making comparisons to leaf words. Thus, these results describe putative (10, 1)- and (10, 2)-motifs.

Table 7. Statistical details of motif models found in a cluster of ribosome function genes (160 genes total) by naïve-CM that finds (10, 1)-motifs. The 34 (10, 1)-motifs presented here are all of the motifs found with a log-likelihood greater than 0.7.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likeli-hood Ratio	Expected (*l*, *d*)
ACATCCGTAC	35	100	0.77	9.4e-16
GTACGGATGT	35	100	0.77	9.4e-16
ACCCAGACAT	32	107	0.7	1.4e-12
ATGTCTGGGT	32	107	0.7	1.4e-12
ACCCGTACAT	30	98	0.71	1.5e-10
ATGTACGGGT	30	98	0.71	1.5e-10
CACCCGTACA	30	90	0.75	1.5e-10
TGTACGGGTG	30	90	0.75	1.5e-10
ACACCCGTAC	27	78	0.76	1.2e-07
GTACGGGTGT	27	78	0.76	1.2e-07
AATGTACGGG	25	73	0.76	8.6e-06
CCCGTACATT	25	73	0.76	8.6e-06
GTACGGATGG	20	54	0.79	0.15
CCATCCGTAC	20	54	0.79	0.15
TGTACGGACG	17	51	0.75	28
CGTCCGTACA	17	51	0.75	28
CACCGGTACA	16	50	0.73	1.4e+02
TGTACCGGTG	16	50	0.73	1.4e+02
CATCCGTACG	15	46	0.74	6.6e+02
CGTACGGATG	15	46	0.74	6.6e+02
GCAACGCGAG	14	44	0.73	2.9e+03
CTCGCGTTGC	14	44	0.73	2.9e+03
CTTTCGGGCG	12	40	0.7	4.4e+04
GCGGGTACTG	12	39	0.71	4.4e+04
GGGCGCGAAA	12	40	0.7	4.4e+04
CACCGGGACA	12	40	0.7	4.4e+04
CAGTACCCGC	12	39	0.71	4.4e+04
AGACGCGAGT	12	33	0.79	4.4e+04
TGGGCGCGAA	12	29	0.84	4.4e+04
ACTCGCGTCT	12	33	0.79	4.4e+04
CGCCCGAAAG	12	40	0.7	4.4e+04
TGTCCCGGTG	12	40	0.7	4.4e+04
TTCGCGCCCA	12	29	0.84	4.4e+04
TTTCGCGCCC	12	40	0.7	4.4e+04

Table 8. Statistical details of motif models found in a cluster of DNA replication and synthesis genes (183 genes total) by naïve-CM that finds (10, 1)-motifs. Only 35 of 447 (10, 1)-motifs with a log-likelihood greater than 0.7 are shown here.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likeli-hood Ratio	Expected (*l*, *d*)
ACGCGTAAAA	52	135	0.75	9.6e-33
TTTACGCGTT	52	134	0.75	9.6e-33
TTTTACGCGT	52	135	0.75	9.6e-33
AACGCGTAAA	52	134	0.75	9.6e-33
AAACGCGTCA	48	84	0.92	6.4e-28
TGACGCGTTT	48	84	0.92	6.4e-28
AAACGCGTTA	42	121	0.7	4.8e-21
TAACGCGTTT	42	121	0.7	4.8e-21
GACGCGTTTT	41	91	0.82	6e-20
AAAACGCGTC	41	91	0.82	6e-20
TTTGACGCGT	40	96	0.78	7.4e-19
ACGCGTCAAA	40	96	0.78	7.4e-19
TTGACGCGTT	38	93	0.77	1e-16
ACGCGTAATA	38	85	0.81	1e-16
TATTACGCGT	38	85	0.81	1e-16
AACGCGTCAA	38	93	0.77	1e-16
TTTACGCGTC	37	73	0.87	1.1e-15
CAAAACGCGT	37	103	0.72	1.1e-15
ACGCGTTTTG	37	103	0.72	1.1e-15
GACGCGTAAA	37	73	0.87	1.1e-15
ATTTACGCGT	36	88	0.77	1.2e-14
CACGCGTTTT	36	100	0.72	1.2e-14
TCACGCGTTT	36	67	0.89	1.2e-14
ACGCGTAAAT	36	88	0.77	1.2e-14
AAAACGCGTG	36	100	0.72	1.2e-14
AAACGCGTGA	36	67	0.89	1.2e-14
TAACGCGTAA	35	68	0.87	1.2e-13
TTACGCGTTA	35	68	0.87	1.2e-13
TGACGCGTAA	34	55	0.95	1.2e-12
TTACGCGTCA	34	55	0.95	1.2e-12
ATGACGCGTT	33	55	0.94	1.2e-11
AGACGCGTAA	33	65	0.87	1.2e-11
GACGCGTAAT	33	63	0.88	1.2e-11
AATAACGCGT	33	95	0.7	1.2e-11
AACGCGTCAT	33	55	0.94	1.2e-11

Table 9. Statistical details of motif models found in a cluster of centrosome organization genes (73 genes total) by naïve-CM that finds (10, 1)-motifs. Only 35 of 232 (10, 1)-motifs with a log-likelihood greater than 0.7 are shown here.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likeli-hood Ratio	Expected (*l*, *d*)
CCGCGTTTGT	11	67	0.79	55
ACAAACGCGG	11	67	0.79	55
TATCCCTCAC	10	75	0.7	4.2e+02
CCGGTAATAG	10	74	0.71	4.2e+02
CTATTACCGG	10	74	0.71	4.2e+02
GTGAGGGATA	10	75	0.7	4.2e+02
TTCGCGTATC	9	66	0.71	2.8e+03
GTTGGCGCTA	9	68	0.7	2.8e+03
GTGTTACCGT	9	67	0.71	2.8e+03
GTCGCGTTTG	9	67	0.71	2.8e+03
CGGCTCTATC	9	65	0.72	2.8e+03
TCAAACGCGC	9	68	0.7	2.8e+03
TCCGCGTTTG	9	67	0.71	2.8e+03
GATACGCGAA	9	66	0.71	2.8e+03
GCGCGTTTGA	9	68	0.7	2.8e+03
CAAACGCGGA	9	67	0.71	2.8e+03
CAAACGCGAC	9	67	0.71	2.8e+03
TAGCGCCAAC	9	68	0.7	2.8e+03
GATAGAGCCG	9	65	0.72	2.8e+03
ACGGTAACAC	9	67	0.71	2.8e+03
GGCCTATCAC	8	53	0.76	1.7e+04
CAGCATGGCC	8	54	0.75	1.7e+04
TTCGCGCTGC	8	52	0.77	1.7e+04
CACTCTTAGC	8	55	0.74	1.7e+04
TGACTGAGGC	8	51	0.77	1.7e+04
TCTAGCTGGG	8	56	0.73	1.7e+04
GCAGCGCGAA	8	52	0.77	1.7e+04
ACGCGAAAGC	8	60	0.7	1.7e+04
GTAAAGGCCC	8	56	0.73	1.7e+04
GTGATAGGCC	8	53	0.76	1.7e+04
ATAGGACACG	8	51	0.77	1.7e+04
CCTATCACAG	8	55	0.74	1.7e+04
GCTAAGAGTG	8	55	0.74	1.7e+04
GCCTCAGTCA	8	51	0.77	1.7e+04
GGGCCTTTAC	8	56	0.73	1.7e+04

Table 10. Statistical details of motif models found in a cluster of ribosome function genes (160 genes total) by naïve-CM that finds (10, 2)-motifs. Only 35 of 1562 (10, 2)-motifs with a log-likelihood greater than 0.2 are shown here.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likeli-hood Ratio	Expected (*l*, *d*)
CATCGGTACA	94	993	0.2	2.5e-08
TGTACGGGTG	94	868	0.26	2.5e-08
CACCCGTACA	94	868	0.26	2.5e-08
TGTACCGATG	94	993	0.2	2.5e-08
CCCAGCCATT	91	960	0.2	1.1e-06
AATGGCTGGG	91	960	0.2	1.1e-06
GTACGGGTGT	88	781	0.28	4.1e-05
ACACCCGTAC	88	781	0.28	4.1e-05
GGTACGGATG	87	848	0.24	0.00013
TGTTCGGGTG	87	889	0.21	0.00013
CATCCGTACC	87	848	0.24	0.00013
CACCCGAACA	87	889	0.21	0.00013
CATCCGTCCA	84	846	0.22	0.0034
TGGACGGATG	84	846	0.22	0.0034
GTTCGGGTGT	82	764	0.26	0.027
CACCCGCACA	82	812	0.23	0.027
TGTGCGGGTG	82	812	0.23	0.027
ACACCCGAAC	82	764	0.26	0.027
CAGCCGTACA	82	838	0.21	0.027
TGTACGGCTG	82	838	0.21	0.027
ACCCAGGCAT	79	822	0.21	0.49
ATGCCTGGGT	79	822	0.21	0.49
GCATCCGTAC	78	787	0.22	1.2
CGTCCGTACA	78	716	0.26	1.2
GTACGGATGC	78	787	0.22	1.2
AGTACGGGTG	78	734	0.25	1.2
TGTACGGACG	78	716	0.26	1.2
CACCCGTACT	78	734	0.25	1.2
ACTGTCTGGG	77	793	0.21	3
ATGTCCGGGT	77	794	0.21	3
ACCCGGACAT	77	794	0.21	3
ACCCGTACAC	77	749	0.24	3
CTGTACGGAG	77	740	0.24	3
GTGTACGGGT	77	749	0.24	3
CTCCGTACAG	77	740	0.24	3

Table 11. Statistical details of motif models found in a cluster of DNA replication and synthesis genes (183 genes total) by naïve-CM that finds (10, 2)-motifs. Only 35 of 2588 (10, 2)-motifs with a log-likelihood greater than 0.2 are shown here.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likelihood Ratio	Expected (*l*, *d*)
AACGCGTAAA	150	1332	0.21	2.4e-44
TTTACGCGTT	150	1332	0.21	2.4e-44
TAAAACGCGT	137	1215	0.21	1.2e-31
ACGCGTTTTA	137	1215	0.21	1.2e-31
AACGCGTCAA	135	1042	0.27	7.1e-30
TTGACGCGTT	135	1042	0.27	7.1e-30
AAACGCGTTA	134	1184	0.22	5.1e-29
TAACGCGTTT	134	1184	0.22	5.1e-29
ACGCGTAAAT	130	1150	0.22	1e-25
ATTTACGCGT	130	1150	0.22	1e-25
AACGCGAAAC	129	1156	0.21	6.6e-25
GTTTCGCGTT	129	1156	0.21	6.6e-25
AATTACGCGT	128	1046	0.25	4e-24
ACGCGTAATT	128	1046	0.25	4e-24
TGACGCGTTT	128	1087	0.23	4e-24
AAACGCGTCA	128	1087	0.23	4e-24
ACGCGTTTTG	127	1129	0.21	2.4e-23
CGCGTAAATT	127	1125	0.21	2.4e-23
AATTTACGCG	127	1125	0.21	2.4e-23
CAAAACGCGT	127	1129	0.21	2.4e-23
ATATAACGCG	126	1123	0.21	1.4e-22
CGCGTTATAT	126	1123	0.21	1.4e-22
GACGCGTAAA	125	970	0.27	8e-22
TACGCGTAAA	125	1103	0.22	8e-22
AGACGCGTAA	125	915	0.3	8e-22
TTACGCGTCT	125	915	0.3	8e-22
TTTACGCGTC	125	970	0.27	8e-22
TTTACGCGTA	125	1103	0.22	8e-22
TTACGCGTTG	124	965	0.27	4.4e-21
CAACGCGTAA	124	965	0.27	4.4e-21
ACGCGTTTAA	123	1088	0.22	2.4e-20
ATAACGCGTT	123	1028	0.24	2.4e-20
AACGCGTTAA	123	1057	0.23	2.4e-20
AACGCGTTAT	123	1028	0.24	2.4e-20
ACGCGTTAAA	123	1079	0.22	2.4e-20

Table 12. Statistical details of motif models found in a cluster of centrosome organization genes (73 genes total) by naïve-CM that finds (10, 2)-motifs. Only 35 of 5827 (10, 2)-motifs with a log-likelihood greater than 0.2 are shown here.

Leaf Word Sequence	In-Class Hits	Out-Class Hits	Log-likeli-hood Ratio	Expected (*l*, *d*)
AAATACCCGC	52	1228	0.21	4.8e-07
GCGGGTATTT	52	1228	0.21	4.8e-07
AATGCGCGAA	48	1094	0.22	0.00052
TTCGCGCATT	48	1094	0.22	0.00052
AATACGCGCA	48	1056	0.24	0.00052
TGCGCGTATT	48	1056	0.24	0.00052
CGCGTTTGTA	47	1091	0.21	0.0026
GTAGCGCAAT	47	1046	0.23	0.0026
TACAAACGCG	47	1091	0.21	0.0026
ATTGCGCTAC	47	1046	0.23	0.0026
GATACGCGAA	45	996	0.23	0.052
AGACGCGTAA	45	995	0.23	0.052
TTACGCGTCT	45	995	0.23	0.052
CGCCTTGGTT	45	1028	0.22	0.052
TTCGCGTATC	45	996	0.23	0.052
AACCAAGGCG	45	1028	0.22	0.052
TCGCGTTTGG	44	967	0.24	0.21
GAAACGCGGA	44	1015	0.22	0.21
CCAAACGCGA	44	967	0.24	0.21
ACAAACGCGC	44	921	0.26	0.21
TCCGCGTTTC	44	1015	0.22	0.21
GCGCGTTTGT	44	921	0.26	0.21
ATCAGCACGT	43	1029	0.2	0.83
CAAAGCGCGA	43	1009	0.21	0.83
ACGTGCTGAT	43	1029	0.2	0.83
TGTTACGCGT	43	973	0.22	0.83
CGCGTTTGGT	43	931	0.24	0.83
ACCAAACGCG	43	931	0.24	0.83
GACGTAACCA	43	1023	0.2	0.83
TAACAACGCG	43	1005	0.21	0.83
TGGTTACGTC	43	1023	0.2	0.83
TTCGCGCTTG	43	1021	0.2	0.83
TCGCGCTTTG	43	1009	0.21	0.83
ACGCGTAACA	43	973	0.22	0.83
TAGACCACGA	43	936	0.24	0.83

Naïve-CM analysis of yeast gene promoters does produce (*l, d*)-motifs that represent biological motifs, and this was expected based on the work of Brazma *et al*. (1998), Sagot (1998) and Vanet *et al*. (2001). Most of the (10, 1)-motifs found in the analysis of the ribosome function genes were variations on a single biological motif, RAP1 (Table 7). Some of the motifs generated by naïve-CM from the ribosome function gene cluster are not easily recognized as being related to known biological motifs. All of the motifs identified in the cluster of DNA replication and synthesis genes are related to the biological motif MCB (Table 8). The expected values for the most of the motifs are very good, although some of the motifs from the ribosome function gene cluster do have poor expected value scores, and the motifs that are not apparently related to known biological motifs also have poor expected values. The motifs from the centrosome organization genes include examples that are related to a known biological motif from these genes, M14a, (Table 9, Tavazoie *et al*. (1999)), but there are many other motifs that were produced by naïve-CM for this cluster that are not apparently related to known biological motifs. Additionally, none of the motifs produced by this analysis have very good expectation values. This is because the best motifs are derived from matches to only 15% of the promoters of these genes. In all three (10, 1)-motif analyses performed here, only a few of the identified (*l, d*)-motifs had log-likelihood ratios above 0.8. For each naïve-CM run, there were some motifs with better log-likelihood scores, but they have many fewer matches to the in-class genes. This is an indication that the (*l, d*)-motif models are too general and that they match many non-biological motifs in the out-class gene promoters.

The (10, 2)-motif analyses all performed similarly relative to the (10, 1)-motif examinations of the same clusters (Tables 10 to 12). Motifs related to the same three biological motifs, RAB1, MCB and M14a, were found when using the (10, 2) parameters. Additionally, each (10, 2)-motif model represented many more matches to the in-class genes, but each also has many hundreds of matches to out-class genes. Thus, the log-likelihood ratios for all of the (10, 2)-motifs shown in Tables 10 to 12 are between 0.2 and 0.3. In all of the motif models generated for the (10, 2) analyses, no motif had a log-likelihood above 0.5. These motif models are very generalized and it is difficult to imagine that they closely represent the true biological motifs.

A comparison between the results of CM and naïve-CM suggests benefits of using CM over a simpler combinatorial approach to motif-finding. Although CM can be slower than naïve-CM when there are many consensus motifs to combine, CM makes more specific models than naïve-CM. The log-likelihood ratios for all of the motif models for CM-derived motif models are generally higher than those for naïve-CM motifs, and these ratios are a good measure of specificity. The motifs made by CM typically match fewer promoters than the motifs found with naïve-CM, and this too is a measure of specificity. While matching too few true motif instances is undesirable, incorrectly matching promoters is also bad. CM results in more conservative motif models that are less likely to have been derived from non-motif sequences. The motif models identified by naïve-CM will only be as complex as the initial parameters that are used to find the models. If words of length ten are placed in the naïve-CM suffix trees with one allowed misspelling, only (10, 1)-motifs will be identified. CM has the ability to combine initially identified consensus motifs into more complex motif models. CM also trims uninformative positions from the ends of motif models. Thus, the motifs generated by CM are more informative than those made by naïve-CM. The clique combining rules used by CM could be added to naïve-CM, but the results would likely not be better than the results from CM. When motifs are combined to form more complex motifs, they are inherently forming more generalized motif models. If the initial motifs before combining are more general in naïve-CM than in CM, then the combining naïve-CM motifs will result in even more generalized and less informative complex motifs.

SUMMARY

As the sequence of complete genomes has become available and with the widespread use of gene expression experiments, there has been renewed interest in computational methods for identifying regulatory motifs in the promoters of commonly expressed genes. There are three main classes of motif finding techniques, phylogenetic comparison, expectation maximization methods and combinatorial. This thesis has reported research aimed at improving on the performance of combinatorial motif-finding algorithms.

For purposes of conveniently describing transcription factor binding motifs, combinatorial motif analysis techniques typically generalize motifs as a construct referred to as an (*l, d*)-motif. This notion of a motif consists of a set of related sequences of length *l* that each have *d* or fewer differences with a canonical sequence. In *l*-dimensional sequence space, an (*l, d*)-motif will be a hypersphere with a diameter of *d* and with the canonical sequence at the center of the hypersphere. This definition of a motif is very convenient for computational purposes, but it is an artificial notion that can be improved upon after consideration of biological motifs.

Biological motifs are sequences that are constrained by the physical parameters of the proteins that bind them. Sets of sequences to which transcription factors are known to bind often have a common consensus sequence as in the (*l, d*)-motif definition, but not all positions in the canonical sequence are equally likely to vary in individual instances of the motif. Some positions are often invariant among all instances of a given motif. This suggests the definition of (*l, f, d*)-motifs where the *f*-value refers to the number of invariant positions in the motif. In *l*-dimensional sequence space, an (l, f, d)-motif will consist of small clusters of related sequences that intersect or overlap the central canonical motif sequence. Unlike the hypersphere of an (*l, d*)-motif, not all portions of the *d*-radius hypersphere around the central canonical sequence will represent possible motif instances of an (*l, f, d*)-motif (Figure 1).

Buhler and Tompa (2002) have argued that for a given size cluster of genes there are some values of *l* and *d* for which (*l, d*)-motifs will be unidentifiable because random solutions to the (*l, d*)-motif problem become common. It has been shown in this thesis that using the combinatorial reasoning of Buhler and Tompa, there are values of *l*, *f* and *d* where (*l, d*)-motifs should not be identifiable but where (*l, f, d*)-motifs can be recognized.

A motif-finding algorithm, CM, was written that makes use of the biological definition of (*l, f, d*)-motifs. CM is a combinatorial algorithm that finds simple (*l, d*)-motifs that correspond to the small clusters of related sequences that reside in multidimensional sequences space near the canonical sequence of an unknown motif. These small sets of sequences are identified using projection (Buhler and Tompa, 2002) and are treated as simple (*l, d*)-motifs. CM calculates a log-likelihood ratio for every (l, d)-motif that is found by projection and discards those that have log-likelihood scores that do not indicate enrichment within the gene cluster. CM then treats these simple (*l, d*)-motifs as cliques and combines them with other cliques using a series of clique-combining rules which possibly result in the formation of more complex cliques. As a result of clique combining, CM often identifies (*l, f, d*)-motifs that should not be identifiable by regular (*l, d*)-motif finding algorithms.

CM was used to analyze gene promoter sequences from clusters of genes derived from a yeast gene expression experiment (Cho *et al*., 1998; Tavazoie *et al*., 1999). These data have also been analyzed by AlignACE, an expectation maximization algorithm. CM found many of the biological motifs in these gene clusters that had been identified by AlignACE. There were some biological motifs that CM was not able to recognize. There were also a few biological motifs identified by CM that were missed by AlignACE. CM produced many other motif-models that have not been identified as being truly biological, and these motif-models should be treated as hypotheses. If these motif-models could be tested in a biology lab, those with the lowest expected values should be tested first. Many of the biological motifs found by CM have low expected values, and this suggests that such motifs are more likely to be biologically relevant.

In order to compare CM with simpler combinatorial methods that are not designed to find (*l, f, d*)-motifs, an (*l, d*)-motif finding program, naïve-CM, was written. Naïve-CM was only tested on a few of the gene clusters that had been analyzed by CM, but biological motifs were identified. However, the motif models produced by naïve-CM are relatively simple and usually over-generalized. Additionally, naïve-CM is not able to find very complex motifs because random (*l, d*)-motif solutions quickly become common as the *l* and *d* values are made more challenging. Thus, CM usually produced more specific motif-models than did naïve-CM, and CM was easily able to generate complex (*l, f, d*)-motifs that naïve-CM would not be able to recognize.

REFERENCES

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

Brazma, A., Jonassen, I., Vilo, J. and Ukkonen, E. (1998) Predicting gene regulatory elements *in silico* on a genomic scale. *Genome Res*., **8**, 1202-1215.

Buhler, J. and Tompa, M. (2002) Finding motifs using random projections. *J. Comp. Biology,* **9**, 225-242.

Chasman, D.I., Lue, N.F., Buchman, A.R., LaPointe, J.W., Lorch, Y. and Kornberg, R.D. (1990) A yeast protein that influences the chromatin structure of UASG and functions as a powerful auxiliary gene activator. *Genes Dev*., **4**, 503-514.

Cho, R.J., Campbell, M.J., Winzeler, E.A., Steinmetz, L., Conway, A., Wodicka, L., Wolfsberg, T.G., Gabrielian, A.E., Landsman, D., Lockhart, D.J. and Davis, R.W. (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol. Cell*., **2**, 65-73.

Cliften, P., Sudarsanam, P., Desikan, A., Fulton, L., Fulton, B., Majors, J., Waterston, R., Cohen, B.A. and Johnston, M. (2003) Finding functional features in *Saccharomyces* genomes by phylogenetic footprinting. *Science,* **301**, 71-76.

Crooks, G.E., Hon, G., Chandonia, J.M. and Brenner, S.E. (2004) WebLogo: A sequence logo generator. *Genome Res*., **14**, 1188-1190.

Dorhmann, P.R., Voth, W.P. and Stillman, D.J. (1996) Role of negative regulation in promoter specificity of the homologous transcription activators Ace2p and Swi5p. *Mol. Cell. Biol*., **16**, 1746-1758.

Frances, M. and Litman, A. (1997) On covering problems of codes. *Theoret. Comput. Syst*., **30**, 113–119.

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.

Hertz, G.Z. and Stormo, G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics,* **15**, 563-577.

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

Kellis, M., Patterson, N., Endrizzi, M., Birren, B. and Lander, E.S. (2003) Sequencing and comparison of yeast species to identify genes and regulatory elements. *Nature,* **423**, 241-254.

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

Li, M., Ma, B. and Wang, L. (2002) On the closest string and substring problems. *J. ACM,* **49**, 157-171.

McCue, L.A., Thompson, W., Carmack, C.S. and Lawrence, C.E. (2002) Factors influencing the identification of transcription factor binding sites by cross-species comparison. *Genome Res.*, **12**, 1523-1532.

Morrow, C., McCaw, P.S. and Baltimore, D. (1989) A new DNA binding and dimerization motif in immunoglobulin enhancer binding, daughterless, MyoD and Myc proteins. *Cell,* **65**, 777-783.

Pevzner, P. and Sze, S.-H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. *Proc 8th Intl. Conf. on Intel. Systems Mol. Biology (ISMB)*, Menlo Park, California, pp. 56-64, AAAI Press.

Roth, F.R., Hughes, J.D., Estep, P.E. and Church, G.M. (1998) Finding DNA regulatory motifs within unaligned non-coding sequences clustered by whole-genome mRNA quantitation. *Nature Biotech.*, **16**, 939-945.

Sagot, M.-F. (1998) Spelling approximate repeated or common motifs using a suffix tree. *"LATIN '98, Theoretical Informatics", Lecture Notes in Computer Sciences*, Springer-Verlag, New York, New York, pp. 111-127.

Schneider, T.D. and Stephens, R.M. (1990) Sequence logos: A new way to display consensus sequences. *Nuc. Acid. Res.*, **18**, 6097-6100.

Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J. and Church, G.M. (1999) Systematic determination of genetic network architecture. *Nature Genetics,* **22**, 281-285.

Vanet, A., Marsan, L., Labigne, A. and Sagot, M.-F. (2000) Inferring regulatory elements from a whole genome. An analysis of *Helicobacter pylori* σ^{80} family of promoter signals. *J. Mol. Biol.*, **297**, 335-353.

VITA

Kevin L. Childs was born in Detroit, MI, on 14 April 1965. He received his B.S. degree in botany from the University of Michigan in 1987 and his Ph.D. in plant physiology from Texas A&M University in 1993. Kevin may be reached by contacting the Department of Computer Science, Texas A&M University, College Station, TX 77843.