Combinatorial Motif Analysis and Hypothesis Generation on a Genomic Scale

Yuh-Jyh Hu[1*], Suzanne Sandmeyer[2]
Calvin McLaughlin[2], Dennis Kibler[1]
Computer Science and Engineering Department*
Tatung University
Information and Computer Science Department[1]
Department of Biological Chemistry, College of Medicine[2]
University of California, Irvine
{yhu,kibler}@ics.uci.edu, {sbsandme,cal}@uci.edu
Keywords: motif, gene regulation, machine learning

Abstract

Motivation: Computer-assisted methods are essential for the analysis of biosequences. Gene activity is regulated in part by the binding of regulatory molecules (transcription factors) to combinations of short motifs. The goal of our analysis is the development of algorithms to identify regulatory motifs and to predict the activity of combinations of those motifs.

Approach: Our research begins with a new motif-finding method, using multiple objective functions and an improved stochastic iterative sampling strategy. Combinatorial motif analysis is accomplished by constructive induction that analyzes potential motif combinations. The hypothesis is generated by applying standard inductive learning algorithms.

Results: Tests using ten previously identified regulons from budding yeast and fourteen artificial families of sequences demonstrated the effectiveness of the new motif-finding method. Motif combination and classification approaches were used in the analysis of a sample DNA array data set derived from genomewide gene expression analysis.

Availability: Programs will be available as executable files upon request.

Contact: yhu@ics.uci.edu

*Now with Tatung University, Taipei, Taiwan. To whom correspondence should be addressed. Email: yhu@cse.ttu.edu.tw or yhu@ics.uci.edu

Introduction

Intensive study of the regulation of individual genes has provided us with a useful working model of gene regulation at the single gene level. The combination of DNA microarray technology (DeRisi *et. al.*, 1997; Wodicka *et. al.*, 1997) which can be used to monitor expression of many genes simultaneously, and computational approaches to data analysis are now providing an entre into a more global appreciation of gene regulation. Budding yeast *Saccharomyces cerevisiae* is a useful model system for test application of DNA arrays genomic analysis algorithms to gene regulation. Not only has the complete genome sequence of *Saccharomyces cerevisiae* been determined (Goffeau *et. al.*, 1996), but gene organization is relatively simple compared to the organization of metazoan organisms. Because microarray experiments provide comprehensive information about the levels of mRNA in the cell which are critical determinants of gene activity, it becomes theoretically possible to identify sets of genes that are similarly regulated under a given condition. This allows: (1) inferences about functions of genes that are not of known function which are co-regulated with genes of known function, (2) discovery of regulatory motifs and combinations of motifs, and (3) improved understanding of the biology of the cellular response to particular environmental stimuli. Computational approaches are described here which, coupled with the output of gene array experiments, can be used to identify regulatory motifs and combinations of those motifs that contribute to gene regulation.

Biologists have traditionally studied the regulation of genes selected for a particular type of activity or function. Although this approach has allowed the identification of regulatory proteins that affect the expression of those genes, it has tended to focus attention on specific sets of genes and inferences concerning the regulation of those genes that have by necessity been extended to less well understood genes. It is likely, given the number of genes for which no function is known (estimated at one third of the yeast genome), that regulatory proteins remain to be identified. In addition, due to the relatively intensive study of exemplary genes, certain aspects of regulation, including positional effects, multiplicity of regulatory motifs, orientation of motifs and the role of combinations of different motifs, although appreciated conceptually, have not been explored comprehensively. Emerging knowledge of genomewide gene activity, combined with the algorithms to infer motifs and to correlate activity and motifs, could broaden our understanding of gene regulation into under-explored areas.

We first present a new approach for the detection of potential regulatory motifs in sequences. This presentation expands upon work by others by combining multiple types of objective functions with an improved iterative sampling technique. It extends our previous report (Hu *et. al.*, 1999) describing motif identification by demonstrating the results of applying a constructive induction algorithm and a decision tree algorithm to the problem of identification of combinations of motifs. To demonstrate the effectiveness of the motif-finding algorithm, we compare its performance to the performance of other algorithms on ten families of yeast genes which share known regulatory motifs and on fourteen artificial test sets of sequences. Second, we report the experimental results of applying that algorithm to families of genes identified by clustering the genomewide gene expression results of a thermal stress time course.

System and Methods

All programs are written in ANSI C, including the Detecting Motifs from Sequences (DMS) software, the constructive induction system GALA (Hu and Kibler, 1996) and the decision tree learning program C4.5 (Quinlan, 1993). Programs have been tested on Sun SPARC workstations running UNIX, and on Intel machines running Windows 98, Windows NT and Linux[1].

Genes are grouped into families based on previously known collective behavior or based on temporal expression patterns determined by DNA array analysis. Our analysis of gene regulation focuses on the search for sequence motifs and combinations of motifs which are implicated in the regulation. This type of analysis suggests further biological tests on the genes through the inference from the hypotheses produced by the analysis.

[1]C4.5 requires minor modification to run under Windows platforms

There are four basic steps in our method.

- First, the genes are categorized into families according to their expression patterns. Some of the gene families used in this study were previously described, some are synthetic, and others were defined by cluster analysis of the results of a DNA array experiment. Cluster analysis was applied to identify genes, the RNAs of which were consistently positively and negatively regulated over the time course.

- Second, for any cluster of genes of interest, the control regions are extracted for each gene and a motif-finding algorithm is applied to the family to find significant motifs. Several methods have been developed for detection of patterns shared by a set of functionally related biosequences (van Helden *et. al.*, 1998; Hertz and Stormo, 1995; Hertz *et. al.*, 1990; Bailey and Elkan, 1995; Lawrence *et. al.*, 1993; Hu, 1998b; Hughey and Krogh, 1996; Eddy, 1995). We will introduce a new motif-finding algorithm called DMS, which will be applied in our gene regulation analysis.

  A particular challenge for finding regulatory motifs is that they can be quite short and thus not statistically distinctive *per se*. As genes can be regulated in many ways, redundancy, location and combinations have to be considered to distinguish regulatory motifs. Finding motifs alone is not sufficient for the analysis of gene regulation. The motifs found will serve as the building blocks for representational transformation in the next step. The change of representation is required to reveal regularity originally implicit in the sequence data.

- Third, based on the motifs found by DMS, the original sequences are transformed into a higher-level representation. It includes (1) the locations of the motifs, (2) the total number of repeats of each motif, (3) the number of repeats of each motif within a selected location range or upstream region, (4) the distance between motifs, and (5) combinatorial motifs as Boolean combinations.

  The objective of this step is to transform the raw string-based representation into one better suited for understanding and additional analysis. The new representation is used to reveal the regularity originally implicit in the raw string-based data. All the information described by the new representation can be either used as a whole or partially used for further analyses.

- Fourth, after the raw sequence data are transformed into the appropriate higher-level representation, a suitable standard inductive learning algorithm is applied to the data to generate hypotheses. There are many inductive learning algorithms available. Each has its own advantages and limitations. In our experiments, we applied a decision tree learning algorithm to construct hypotheses represented as decision trees. These hypotheses predict whether a gene will be expressed or not. The hypotheses are easily understood and provide a micro view of the families as they suggest reasons for different behaviors of the genes to complement the macro view of the genome-wide gene expression changes.

Algorithm

DMS: Detect Motifs from Sequences

The sequence segments, such as binding sites for a particular protein, are not necessarily accurately represented by a single sequence pattern because modest variations in the motif are important for controlling the differential binding of the protein to different regulatory regions. Consequently, the weight matrix was adopted for motif representation. By running an iterative sampling optimization process, DMS finds a user-specified number of motifs.

The weight matrix method approach has been used in various pattern-identification problems (Harr *et. al.*, 1983; Staden, 1984; Hertz *et. al.*, 1990; Lawrence *et. al.*, 1993). It is usually built from the base frequency of example biosequences. For example, in the seven-member NIT regulatory family (van Helden *et. al.*, 1998) a possible six-base motif matrix is illustrated in Figure 1. By dividing every element of the matrix by the number of sequences, we construct a normalized matrix illustrated in Figure 1.

Based on the normalized motif matrix, we can calculate the match score of any six-base sequence by dividing the sum of the value for each position by the width of the motif. For example, given a 6-base sequence, GATAAG, its match score is $\frac{0.86+1+1+1+1+1}{6}$. The success of these analyses confirms the fact that the frequencies of bases at positions within sites are related to the importance of the bases to the functioning within the sites (Stormo, 1988). The challenge is to find a matrix that best represents the motif.

We propose a new motif-finding algorithm, DMS. Unlike other approaches, DMS uses multiple types of objective functions, the motif consensus quality, the motif multiplicity significance and the motif coverage. The consensus quality guides the search for well-conserved motif candidates, the motif multiplicity significance reflects the value of multiple copies of a single motif, and the motif coverage addresses the importance of a motif's being commonly shared by a given family of sequences. The consensus quality of a matrix is derived from the entropy. The lower the entropy, the better conserved the motif. The entropy is calculated from the probability that each base occurs at each position in the motif, Pm_{base}. More precisely, the entropy for a particular column n in the matrix is given by:

$$E_n = -\sum_{i=b_1}^{b_4} Pm_i \log_2 Pm_i$$

where $b_1..b_4$ are the bases A, G, C, T. If the bases are uniformly distributed over a position, then the maximum value of 2 is obtained. If only a single base appears in a position, then the minimum value of 0 is obtained. Thus we define the consensus quality of column n as:

$$C_n = 2 - E_n$$

The final consensus quality of a matrix b, is defined as the average of all position quality.

$$con(b) = \frac{1}{W}\sum_{n=1}^{W} C_n$$

where W is the width of the motif. The multiplicity significance is derived from the measure of precision as defined in the information retrieval literature. It is simple and empirically effective. We define the multiplicity significance of a motif b as:

$$mul(b) = \frac{occ_S(b)}{occ_G(b)}$$

where $occ_S(b)$ is b's occurrences in a given family S, and $occ_G(b)$ is b's occurrences in genome. This measures the representativeness of a sequence in a family relative to the entire genome and, consequently, discounts sequence which are common everywhere, such as tandem repeats or poly A's.

The motif coverage is defined as the ratio of the number of the sequences containing b to the total number of sequences given.

A 0 7 0 7 7 0		A 0.00 1.00 0.00 1.00 1.00 0.00
G 6 0 0 0 0 7	normalized	G 0.86 0.00 0.00 0.00 0.00 1.00
C 1 0 0 0 0 0		C 0.14 0.00 0.00 0.00 0.00 0.00
T 0 0 7 0 0 0		T 0.00 0.00 1.00 0.00 0.00 0.00

Figure 1: A 6-base Motif Matrix Example

$$cov(b) = \frac{cont_S(b)}{|S|}$$

where $cont_S(b)$ is the number of sequences in S that contain b, and $|S|$ is the total number of sequences in S.

Given a set of N biosequences, DMS carries out an iterative improvement search that attempts to find a user-defined number, d, of matrices that maximize the consensus quality. These d matrices are motif candidates. DMS then ranks these motifs according to a merit measure based on the combination of the consensus quality, the multiplicity significance and the motif coverage. Given the d motifs, we first normalize the consensus quality, the multiplicity significance and the motif coverage of each motif b, using the maximum value, as defined below:

$$Con_{normal}(b) = \frac{con(b)}{MAX(con)}$$
$$Mul_{normal}(b) = \frac{mul(b)}{MAX(mul)}$$
$$Cov_{normal}(b) = \frac{cov(b)}{MAX(cov)}$$

where $MAX(con)$ is the maximum consensus quality of the d motifs, $MAX(mul)$, the maximum multiplicity significance of the d motifs, and $MAX(cov)$, the maximum motif coverage of the d motifs.

Combining all the objective functions introduced above, we propose the final merit measure of a motif b, $Merit(b)$, as defined below:

$$\frac{1}{\frac{1}{3}\left(\frac{1}{Con_{normal}(b)} + \frac{1}{Mul_{normal}(b)} + \frac{1}{Cov_{normal}(b)}\right)}$$

The value of merit is in the range between 0 and 1. It reflects the synergy of the consensus quality, the multiplicity significance and the motif coverage.

There are three steps in DMS which are detailed in the following subsections.

Translating Subsequences into Matrices

If we knew the motif location(s) in every sequence, we could generate a probability matrix corresponding to these positions. As these position are unknown, we take a different approach. We begin by allowing each subsequence of length W to be a candidate motif. Like most current algorithms, such as CONSENSUS, the Gibbs sampler, and MEME, etc., the length W is specified by the user. We convert this particular subsequence into a probability matrix in two steps, adopting an idea from (Bailey and Elkan, 1995). First we fix the probability of every base in the subsequence to some value $0 < X < 1$, and assign probabilities of the other bases according to $\frac{1-X}{4-1}$ (4 nucleic bases). Following Bailey and Elkan, we set X to 0.5. This gives us a set of seed probability matrices to be used as starting points for iterative improvement. For a given family of sequences, we can either exhaustively translate every subsequence into a matrix for analysis or we can select a random subset of the sequences and only generate candidate starting points from this subset. Because significant motifs are generally well conserved and thus occur in most sequences, this subsetting strategy is effective without empirically losing generality.

Filtering Possible Motif Occurrences

Rather than making the common assumption that each motif occurs only once per sequence, we allow for the possibility that a motif may occur multiple times in a single sequence. For each matrix and each sequence, we find the position that maximizes the match score and adds it to the list of potential motif positions. Then we set the threshold for deciding if a motif occurs at any position as the mean of the match scores. Finally we add to the list of motif positions any other position whose match score is greater than this threshold. Occurrence overlap is allowed. This process defines a set of potential motif positions.

Finding and Ranking Motif Candidates

After the likely motif positions are determined, DMS performs an iterative optimization procedure to find the motif probability matrix. Unlike current approaches, such as the Gibbs sampler, that search all possible positions within a sequence, DMS only considers the potential motif positions determined in the previous step. This strategy significantly constrains the search space. For initialization, DMS randomly selects a position from the set of potential motif positions that are determined in the previous step to form the initial probability matrix.

A sequence is then chosen at random for optimization. DMS optimizes the consensus quality of the matrix by checking every potential motif position within the selected sequence. For each position, we compute the consensus quality (as defined above) of the corresponding matrix. The position that achieves the maximum consensus quality is chosen to update the matrix. The process is repeated until no improvement is noted. In each optimization cycle, the order of sequences is randomly shuffled. The randomization in each trial cycle is important to remove implicit biases, such as the order of the sequences, that can be harmful in search algorithms (Hampson and Kibler, 1996). At this point, in each sequence, the subsequence that contributes to the last updated matrix is determined. We then compute the mean of the match scores of the subsequences that form the matrix, and use the mean as a threshold to select all subsequences with a match score over this threshold as possible motif occurrences in each sequence. DMS is not biased toward any predefined motif occurrences, e.g., 1 or more motif occurrences per sequences. The occurrence of a motif is determined by the match threshold as defined above. All these motif occurrences in sequences are used to form the final motif matrix, and it becomes a motif candidate.

The same procedure is performed on all matrices to produce the candidate motifs. Finally, DMS ranks the candidate motifs according to its merit measure.

A pseudo-code description of matrix optimization procedure is given in Figure 2.

Representational Transformation

Two types of additional motif information become available after DMS identifies motifs from the given family of sequences, the motif occurrences and the motif locations. Based on the information available, we can transform the original sequence data into a higher-level representation. Each sequence is transformed into a vector that contains the motif information associated with the sequence, including the number of motif repeats in the entire sequence and the number of motif repeats within a selected segment of the sequence. This vector representation was chosen because it is the most widely used representation for standard inductive learners in the machine learning community. It increases the applicability of machine learning techniques. Given only one family of sequences, the particular segment is selected based on the background knowledge, i.e. it is specified by the user. If two or more families of sequences are provided, the sequence segment can be either specified by the domain expert or determined by DMS. It is computationally prohibited to find the optimal segment that gains the maximum discrimination between families by checking all possibilities. Therefore, we divide the sequences into equal intervals. For each interval, we compute the information gain according to the number of motif repeats in that interval. DMS thus selects the interval that attains the highest information gain.

For example, assuming three motifs are found, M_1, M_2 and M_3, an original nucleic sequence can be represented as a vector, $(M_{total_1}, M_{total_2}, M_{total_3}, M_{[100-150bp]_1}, M_{[350-400bp]_2}, M_{[50-100bp]_3})$. The first three elements are the total number of repeats of M_1, M_2 and M_3. The fourth element presents the number of repeats of M_1 within the range of 100-150bp in the upstream region of the sequence. The last two elements present the number of repeats of M_2 and M_3 within the range of 350-400bp and 50-100bp, respectively. For instance, a sequence can be transformed into (5, 2, 3, 3, 1, 2). This means that the sequence has a total of 5 repeats of M_1, 2 repeats of M_2 and 3 repeats of M_3. There are 3 copies of M_1 located in 100-150bp upstream of the sequence, one copy of M_2 in 350-400bp upstream, and two copies of M_3 in 50-100bp upstream.

Given multiple families of genes, af-

```
Given: a set B  of biosequences
       a random subset S of B
       the width W of a motif
Return: a set C of ranked candidate motifs

Step 1. Translation
        For each subsequence b in S Do
          Translate b into candidate probability matrix m via:
              m(i,base) = .50 if base occurs in position i
                        = .17 otherwise

Step 2. Filter possible motif positions
        For each m in S Do
          For each sequence s in B Do
            Find Position p with highest match score in s
            Add p to Potential Positions
          Compute the mean of the highest match scores in B
          For each sequence in B Do
            Set Potential Positions to those with match score > mean

Step 3. Find and rank motif candidates
        Randomly choose a Potential Position in each sequence to
            initialize matrix M
        Repeat
          Randomly pick a sequence s in B
          Check if M's consensus quality can be improved by using a
             different Potential Position in s
          Update matrix M
        Until no improvement in M's consensus quality
        Compute the mean of match scores of subsequences contributing
to M
        For each sequence in B Do
          Select subsequences with match score > mean as motif
occurrences
        Form the final matrix FM with all occurrences in B
        Put FM in C
Sort all motif candidates in C according to merit
Return C
```

Figure 2: Pseudo-code of DMS

ter the transformation, the original sequence data is represented as sets of vectors. These vectors are used as the training examples for GALA (Hu and Kibler, 1996; Hu, 1998a) to further analyze motif combinations. From the point of view of GALA, each element of a vector is a primitive attribute. The purpose of applying GALA here is to find combinations of attributes as new attributes to improve the quality of the hypotheses that will be later generated by a standard inductive learning algorithm. GALA applies Boolean operators to construct new attributes represented as Boolean combinations. As a data-driven approach, it iteratively constructs and evaluates new attributes by analyzing the training examples. Short Boolean combinations are reasonably understandable. Comprehensibility allows domain experts to explore the new attributes either for further improvement or for justification.

Hypothesis Generation

The final hypothesis for different gene behaviors in the same environment is produced by the standard inductive learning algorithm C4.5 (Quinlan, 1993). The input to C4.5 is a set of vectors transformed from the original sequence data combined with the combinatorial motifs, i.e., Boolean combinations of motifs, generated by GALA. With the input as the training examples, C4.5 produces a classification hypothesis that could be used to explain why these families of genes behave differently under the same condition as well as suggesting additional biological questions.

Implementation

Finding Motifs in Real Regulons

Yeast metabolism has been widely studied, and in some cases the transcription factors involved in the regulation of members of a common pathway are known. Those families of co-regulated genes provide ideal data sets on which to test the systems designed to detect regulatory motifs.

From the study of the literature, van Helden *et. al.* defined ten families of genes that have known common regulatory site(s) or motif(s). There are many additional motifs involved in regulation generally, but the known ones in these regulons define ten learning tasks for comparing the various algorithms. These families are described in more detail in Table 1. The first column in Table 1 denotes the name of the regulatory family, column 2 shows the number of genes in that family, and column 3 presents the published motifs. It is assumed for this exercise that the regulation of a gene is determined by motifs in the upstream region. The 800bp upstream region was used for each gene, as this is the same sized region used by van Helden *et. al.* in their experiments.

Our objective was to test whether DMS can identify the published motifs. As the biological literature only publishes regulatory motifs in the IUPAC code, a methods was needed to construct a way to credit

Table 1: Ten Regulatory Families with Published Motifs

Family	Size	published motifs
NIT	7	GATAAG
MET	11	TCACGTG AAAACTGTGG
PHO	5	GCACGTGGG GCACGTTTT
PDR	7	TCCGCGGA
GAL	6	CGGNNNNNWNNNNNCCG
GCN	38	RRTGACTCTTT
INO	10	CATGTGAAWT
HAP	8	CCAAY
YAP	16	TTACTAA
TUP	25	KANWWWWATSYGGGGW

the algorithms that determined a probability matrix. The following procedure was used for determining a match. From each probability matrix we constructed a consensus pattern. If this consensus pattern matched the published motif in 80% of the positions of the motif, it was counted as a correct match. A base in the consensus sequence was allowed to match a disjunction of bases (as described by the IUPAC code) if the disjunction contained the base.

There are two parameters used by DMS. One is the motif width, and the other is the random subset size (see pseudo-code of DMS). To maintain consistency, for those families with more than ten members, we set the subset size to be 10; otherwise, we set the subset size to be equal to the family size. The motif width is set to that of the published motif in each family. Except for these two parameters, we did not tune DMS or modify the sequence data in any way, e.g. by prespecifying the expected number of motif matches/occurrences. To test the stability of DMS, we ran DMS on each family five times, using different random seeds. The results showed that DMS identified all the published motifs in all regulatory families in each run. By "identified" we mean that the published motifs are found and ranked in the top 40 motifs according to the merit measure as defined earlier. Most of the published motifs are ranked the top except for some weak motifs or short motifs, e.g., the motif in HAP family, CCAAY, is ranked the 34th, and the less conserved motif in PHO family, GCACGTTTT, is ranked the 18th.

Analysis

The motif-finding problem can be viewed as finding the patterns common to a given family of sequences. The difficulty of finding the biologically meaningful motifs is increased by the variability in (1) the bases at each position in the motif, (2) the lack of alignment of sites among the sequences, and (3) the multiplicity of motif occurrences within a given sequence. With these uncertain factors, the search space of motifs is generally computationally intractable. Moreover, as the characteristics of a given family are usually unknown in advance, and there is no universal measure of the significance of patterns, the biological meaning of a given motif must be verified empirically. The objective functions used by the current motif-finding algorithms are based on either heuristics or statistics which, at this time, are not equated with biological significance. Features considered likely to correlate with significance of computationally defined motifs include: (1) degree of conservation, (2) consistency of occurrence of the motif across the members of a family, and (3) distributional difference between the rest of the genome and the regulatory regions.

Systems using different measures of motif significance have been proposed. However, most of them cannot completely meet the criteria described above without certain assumptions. For example, the Gibbs sampler (Lawrence *et. al.*, 1993) and MEME (Bailey and Elkan, 1995) are both sensitive to the assumption of the expected number of motif matches. Weak assumptions cause breakdown of the systems. The reason for the failure is that the objective functions upon which the systems are based do not meet the criteria above without the assistance of a specific settings of program parameters. Relative information and likelihood are good measures of the motif consensus quality, but their values both vary with changes in the expected number of motif matches. Typically, the expected number of motif matches affects the performance of this type of measure. A new significance measure proposed by van Helden addresses the importance of motif multiplicity in terms of over-representation (van Helden *et. al.*, 1998). This measure avoids the burden of making assumptions of the expected number of motif matches, but it is based on the number of motif occurrences and does not consider the motif location. Therefore, it does not discriminate the difference between a high count motif located in one sequence and a high count motif distributed among many sequences.

DMS overcomes the above-mentioned limitations by combining multiple objective functions, the motif information content, multiplicity and coverage. Unlike some current approaches that require the setting of expected number of motif matches, DMS automatically computes the match threshold based on the mean match scores to determine the motif occurrences (see Figure 2). The information content is used to guide the search for conserved candidate motifs; the motif multiplicity addresses the value of copy number of a motif; and motif coverage reflects the importance of distribution of a motif among many members of a family. The synergy of these objective functions refines the predictions of the algorithm with respect to biological significance. First, the information content is used to measure the consensus quality. The higher information content (i.e., lower entropy), the better conserved is the pattern. Based on the information content, DMS can identify motif candidates of high consensus quality. Motifs with high information content are not necessarily significant. To mitigate this limitation, DMS applies other objective functions to measure the significance of motifs. Combined with information content, the motif multiplicity and the motif coverage help detect motifs not only of high consensus quality, but also of high representation and coverage in a given family. The strategy of using multiple complementary objective functions can alleviate the common limitations.

To verify the synergy of multiple objective functions, the merit measure was compared with two other objective functions, relative entropy[2] and the significance measure introduced by van Helden *et. al.*. We substituted relative entropy and van Helden's significance measure respectively for the merit measure in DMS, and checked the motif ranking according to the measure we used in DMS. In this way, we kept the same search strategy, and only varied the objective functions. The motif match threshold was also determined in the same way to maintain consistency. In Table 2, we show the rank of the published motifs in some real regulons according to the measure that was applied. Column 1 presents the families tested, and column 2 shows the published motifs or the seeded motifs. Columns 3-5 show the rank of the motif according to the measure applied. The rank of the motif given by the merit measure is generally more predictive than other single measures. This suggests that combining multiple complementary objective functions is a better ranking strategy for DMS. To further compare the significance of relative entropy with that of the consensus quality as defined earlier, we replaced the consensus quality with relative entropy in the merit measure, and re-ran DMS on the same families. The results showed no significant difference, i.e., the motif rankings were about the same. Note that we do not claim the merit measure is necessarily better than relative entropy or van Helden's significance measure in general. As we know the value of relative entropy or van Helden's measure is dependent on the number of motif matches, with appropriate settings of the expected number of motif matches, algorithms based on relative entropy, e.g., CONSENSUS, will also rank the motifs correctly (Stormo, Hertz, van Helden, 1999 personal communication). The above experimental results only suggest that for DMS, the merit measure is more predictive than relative entropy or van Helden's measure alone.

[2]Relative entropy (and its variant) has been widely used in several algorithms such as CONSENSUS and the Gibbs sampler. Bailey reported maximizing relative entropy is equivalent to maximizing likelihood ratio when the assumed probability distribution is multinomial (Bailey, 1993).

Table 2: Ranking of Motifs by Different Measures

Family	known motif	van Helden's sig	Relative Entropy	Merit of DMS
NIT	GATAAG	30	339	1
MET	TCACGTG	2	12	1
	AAAACTGTGG	5	6	1
PDR	TCCGCGGA	148	56	2
INO	CATGTGAAWT	3	13	1
HAP	CCAAY	149	97	34
YAP	TTACTAA	14	187	1

Analyses of Global Gene Expression

The advent of microarray technology makes it possible to simultaneously measure the activity of most genes (in this case defined as levels of corresponding mRNA) under various test conditions. These data can then be used to computationally define, by cluster analysis, new families of genes, which can be analyzed for common regulatory motifs. Implicit to this analysis is the assumption that genes which behave similarly are more likely to share common regulatory motifs.

In the experiment performed here, yeast gene expression was probed using the Affymetrix GeneChip System. The yeast YE6100 array interrogates over 6200 yeast genes, defined as ORFs longer than 100 codons, using multiple complementary 25-mers per gene. These are synthesized in situ on four silica wafers using a photolithographic process (Wodicka *et. al.*, 1996). The sequences of the oligonucleotide probes are designed to maximize specific hybridization to the target RNAs. The mixture of RNA is hybridized to the microarray and hybridized RNAs are then stained with streptavidin-phycoerythrin conjugate. The pattern of hybridization is detected as fluorescence using a Hewlett Packard scanning argon laser. Affymetrix proprietary software is used to process the image file, resulting in the values for each gene reflecting the absolute level of mRNAs in control and stressed cells.

Budding yeast cells were submitted to a thermal stress by shifting the culture from 23°C to 39°C. Samples were collected at 0, 5, 10, and 20 minutes and RNA was extracted. These RNA samples were analyzed as previously described (Wodicka *et. al.*, 1997) using an Affymetrix GeneChip machine and Affymetrix GeneChip 3.0 software. These results were output as values that displayed the absolute levels of RNA for approximately 6200 genes over the time course. Cluster analysis was performed in order to identify families of genes which behave similarly. These families were analyzed using DMS in order to identify potential common regulatory elements.

The heatshock response in yeast was chosen as the subject of the microarray experiment described here because it regulates a large number of genes, is effected by conserved families of proteins, and has been relatively intensively studied. Regulation of the stress response occurs at many levels includes transcriptional, translational and post-translational mechanisms. Based on the expression time course, 82 ORFs for which RNA levels increased and 39 ORFs for which RNA levels decreased were identified for further analyses. The 82 heat-shock genes were chosen to include only genes having a maximum expression level fold increase greater than four, and the 39 heat-stroke genes have a maximum expression level fold decrease greater than three. Because yeast genes tend to have transcriptional regulatory regions that are upstream of the ORF within 500bp, the 500bp upstream region for each gene was used for analysis. It should be noted that because of the fold thresholds chosen, there were genes which were regulated that were not included in the analysis.

DMS analysis was performed and motifs were detected based on the 82 genes which displayed an increase of greater than four folds in RNA levels during the heat shock. As we focus on short motifs in our current study, for DMS we set the motif width to be five, and the subset size to be twenty. According to the merit values, we selected the top fourteen interesting motifs for further analyses. In addition to the fourteen motifs, the background knowledge of the previously defined heat-shock element (Fernandes *et. al.*, 1994) was applied in the subsequent decision tree analysis for a total of fifteen motifs as listed in Figure 3. Including the heat shock element, there are two known motifs on the list. Motif 1 is a sub-motif (AGAA) of the heat-shock element (Fernandes *et. al.*, 1994), and motif 2 is the stress element (STRE) (Kobayashi and McEntee, 1993). The heat-shock element is an inverted repeat of TTC and GAA. As the gap between these two sub-motifs may vary, and a T (or A) is favored after C (or ahead of G), we added the sub-motif, TTCT, of the heat-shock element in our study.

Based on these selected motifs, we transformed each raw DNA sequence into a vector representation. Each vector indicates for each motif the number of total motif occurrences in a sequence, and the number of motif occurrences within a specific range in that sequence. Note that the motif occurrences are determined by DMS with the mean match score as the threshold as explained earlier. From the point of view of inductive learning, after the transformation, we have a set of pre-classified data, i.e., heat-shock genes and heat-stroke genes. We first applied GALA (Hu and Kibler, 1996; Hu, 1998a) to the new data set to analyze motif combinations, and used the inductive learning program C4.5 (Quinlan, 1993) to generate a hypothesis, represented as the decision tree shown in Figure 4. The output of GALA is a list of combinatorial motifs represented as Boolean combinations. These Boolean combinations will be used by C4.5 to construct a decision tree hypothesis. For example, the three nodes described as Boolean combinations in the decision tree shown in Figure 4 are part of the output of GALA. We modified the original output of C4.5 by directly putting in the motif Boolean combinations learned by GALA to increase readability. The decision tree describes features that are found and not found in genes that are positively or negatively regulated by the heat treatment. This description can be applied to other genes in order to predict their behavior under thermal stress.

There are three condition nodes in the hypothesis. Each node describes a condition with two outcomes, true or false. In each node, "$*$" means an "AND" and "$+$" means an "OR". For example, the root (i.e., node 1) describes a condition: (if there is one or more Motif 2 and Motif 14) OR (if there is one or more Motif 4 located in 100bp to 200bp upstream) OR (if there are two or more Motif 7 located in 200bp to 300bp upstream) OR (if there are two or more Motif 5 and one or more Motif 8 located in 400bp to 500bp upstream) OR (if there are three or more Motif 6) OR (if there is one or more Motif 2 located in 100bp to 200bp upstream). To classify a gene's behavior, the decision tree was traced from the top, i.e., node 1, to the bottom, i.e., a class. Note that when classifying a new gene, to keep the consistency, we used the same threshold as used by DMS during its search for motif occurrences to determine motif occurrences (refer to pseudo-code of DMS).

To verify the usefulness of the motif combinations generated by GALA, we performed two iterations of 10-fold cross validation by running C4.5 on the same data set with and without using the motif combinations generated by GALA. To perform one 10-fold cross validation, we first randomly shuffle the total data, i.e., the 82 heat-shock genes and the 39 heat-stroke genes, to remove the ordering bias. We then divide the data into ten equal-sized sets, i.e., each set contains 10% of the total data; the distribution of the heat-shock genes and the heat-stroke genes in each set will be at random. Each set of data will be iteratively used as the validation data to test the accuracy of the predictor, and the remaining nine sets of data will be used for training the predictor. The final predictive accuracy of the predictor is the average of the accuracy of the total ten runs of experiments. Our experimental results showed that motif combinations significantly improved the predictive accuracy by about 10% (80.85% with combinations compared with 70.84% without combinations) in paired t-test (confidence level $\gg 99\%$). We also tested the significance of the added sub-motif, TTCT, of the heat-shock element. We compared the predictive accuracy before and after removing this sub-motif, TTCT, based on the same 10-fold cross validation to keep the consistency. We found that the predictive accuracy significantly dropped by 2.69% (78.16% compared with 80.85%) in the paired t-test after we removed TTCT (confidence level $> 95\%$). These results suggest that appropriate background knowledge can improve the quality of the hypotheses.

```
              Motif 1 = TTCT                        Motif 2 = AGGGG
              A 0.000 0.000 0.000 0.000             A 1.000 0.000 0.195 0.012
0.000
              G 0.000 0.000 0.000 0.000             G 0.000 1.000 0.768 0.988
1.000
              C 0.000 0.000 1.000 0.000             C 0.000 0.000 0.012 0.000
0.000
              T 1.000 1.000 0.000 1.000             T 0.000 0.000 0.024 0.000
0.000

              Motif 3 = CCCTT                       Motif 4 = TCCCT
              A 0.000 0.085 0.000 0.000 0.000       A 0.000 0.000 0.000 0.000
0.000
              G 0.000 0.024 0.012 0.000 0.000       G 0.000 0.000 0.000 0.000
0.000
              C 1.000 0.890 0.988 0.000 0.000       C 0.061 1.000 0.866 1.000
0.000
              T 0.000 0.000 0.000 1.000 1.000       T 0.939 0.000 0.134 0.000
1.000

              Motif 5 = ACAAG                       Motif 6 = GAAGA
              A 0.732 0.000 1.000 1.000 0.000       A 0.000 1.000 0.780 0.000
0.000
              G 0.195 0.000 0.000 0.000 1.000       G 0.963 0.000 0.134 1.000
1.000
              C 0.012 1.000 0.000 0.000 0.000       C 0.037 0.000 0.000 0.000
0.000
              T 0.061 0.000 0.000 0.000 0.000       T 0.000 0.000 0.085 0.000
0.000

              Motif 7 = TATCA                       Motif 8 = TCTTG
              A 0.000 1.000 0.061 0.000 0.988       A 0.024 0.000 0.000 0.000
0.000
              G 0.000 0.000 0.000 0.000 0.000       G 0.037 0.000 0.000 0.000
1.000
              C 0.000 0.000 0.122 1.000 0.000       C 0.122 1.000 0.000 0.000
0.000
              T 1.000 0.000 0.817 0.000 0.012       T 0.817 0.000 1.000 1.000
0.000

              Motif 9 = TAAAG                       Motif 10 = GCAAA
              A 0.000 1.000 0.817 1.000 0.000       A 0.000 0.000 1.000 0.768
0.976
              G 0.000 0.000 0.000 0.000 1.000       G 1.000 0.000 0.000 0.000
0.000
              C 0.000 0.000 0.037 0.000 0.000       C 0.000 1.000 0.000 0.159
0.024
              T 1.000 0.000 0.146 0.000 0.000       T 0.000 0.000 0.000 0.073
0.000

              Motif 11 = CTTCT                      Motif 12 = AAGGA
              A 0.000 0.000 0.000 0.134 0.000       A 1.000 1.000 0.000 0.000
0.963
              G 0.000 0.000 0.000 0.000 0.000       G 0.000 0.000 1.000 0.890
0.037
              C 1.000 0.000 0.000 0.866 0.012       C 0.000 0.000 0.000 0.000
0.000
              T 0.000 1.000 1.000 0.000 0.988       T 0.000 0.000 0.000 0.110
0.000

              Motif 13 = CTTAT                      Motif 14 = AATCT
              A 0.000 0.000 0.000 0.963 0.000       A 1.000 1.000 0.012 0.000
0.000
              G 0.134 0.000 0.000 0.037 0.000       G 0.000 0.000 0.000 0.171
0.000
              C 0.866 0.000 0.000 0.000 0.000       C 0.000 0.000 0.037 0.829
0.000
              T 0.000 1.000 1.000 0.000 1.000       T 0.000 0.000 0.951 0.000
1.000

              Motif 15 = GGCAG
              A 0.000 0.000 0.366 0.878 0.000
              G 1.000 1.000 0.024 0.000 0.988
              C 0.000 0.000 0.610 0.000 0.012
              T 0.000 0.000 0.000 0.122 0.000
```

Figure 3: 15 Significant Motifs in 82 Heat-Shock Genes

```
NODE 1 : (M2=1)*(M14=1) + (M4 in [100,200] = 1) + (M7 in [200,300] = 2) +
         (M5=2)*(M8 in [400,500] = 1) + (M6=3) + (M2 in [100,200] = 1)
            === TRUE
                NODE 2 : (M1 in [0,100] = 3) + (M9=1)*(M10=1) + (M7=3) +
                         (M11 in [400,500] = 1) + (M9=1)*(M2 in [100,200] =
1)
                            === TRUE
                                Class: Heat-Shock  {76 heat-shock genes }
                                                   { 0 heat-stroke genes}
                            === FALSE
                                Class: Heat-Stroke { 1 heat-shock genes }
                                                   { 4 heat-stroke genes}
            === FALSE
                NODE 3 : ((M9<=0) + (M10 in [400,500] = 1))*
                         ((M3=1)*(M7=3)+(M3=1)*(M13 in [200,300]  = 1))
                            === TRUE
                                Class: Heat-Shock  { 5 heat-shock genes }
                                                   { 0 heat-stroke genes}
                            === FALSE
                                Class: Heat-Stroke { 0 heat-shock genes }
                                                   {35 heat-stroke genes}
```

Figure 4: The Hypothesis of Heat-Shock and Heat-Stroke Genes (represented by a Decision Tree)

Discussion

Computational tools for detecting subtle similarities and classifying sequences have become an essential component of the research process. Large databases of biological information create challenging data-mining problems and opportunities, each requiring new ideas. Conventional computer science algorithms have been useful, but are increasingly unable to address many of the most interesting sequence analysis problems. This is due to the inherent complexity of biological systems. Machine-learning approaches, on the other hand, are ideally suited for domains characterized by the presence of large amounts of data, noisy patterns, and the absence of general theories. The fundamental idea behind these approaches is to learn the theory automatically from the data, through a process of inference and model fitting. These methods provide a complementary approach to conventional methods. It is the confluence of all three factors—data, computers, and theoretical framework—that is fueling the machine-learning expansion, in bioinformatics and elsewhere (Baldi, 1998).

We propose using multiple objective functions to detect meaningful motifs in sequences. Our experimental results demonstrate the synergy of using information content and the multiplicity significance helps maintain the balance between the consensus quality and the over-representation of motifs. The strategy of using multiple complementary objective functions extends the power of current approaches.

Yeast genome-wide expression studies provide data sets which can now be used for the global analysis of gene expression. One example has been provided here where such a data set was used together with DMS to generate a ranked list of candidate regulatory motifs. Nevertheless, such regulatory motifs (including known regulatory sequences) are known to be relatively short and thus are not likely to be sufficient to fully specify the regulatory properties of sets of genes under particular conditions. From experiments on particular genes it is clear that combinations of motifs play an important role in gene regulation at the transcriptional level. In order to begin to address this problem computationally, a decision algorithm was used to develop an hypothesis which describes the regulated expression of a set of genes in terms of the candidate regulatory motif combinations. This description can now be tested and improved algorithms developed. This study and other recent studies suggest the potential power for biological prediction of computational analysis of experiments performed on a genomic scale.

Acknowledgements

We thank the UCI Genomics Group (S. Hampson, C. McLaughlin, H. Mangalam, R. Lathrop, G. W. Hatfield) for helpful discussions. We thank L. Yieh and S. Trinidad for assistance with the GeneChip experiments. Y-J Hu was in part supported by the Functional Genomics Program of the Chao Family Comprehensive Cancer Center.

References

Bailey, T. (1993) "Likelihood vs. Information in Aligning Biopolymer Sequences", UCSD Tech Report, CS93-318.

Bailey, T. and Elkan, C. (1995) "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization", Machine Learning, 21, p51-80.

Baldi, P. and Brunak, S. (1998) *Bioinformatics : the Machine Learning Approach*, Kluwer Academic Publishers.

DeRisi, J., Iyer, V. & Brown, P. (1997) "Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale", *Science*, 278, p680-696.

Eddy, S. (1995) "Multiple Alignment using Hidden Markov Models", in Proceedings of International Conference on Intelligent Systems for Molecular Biology, p114-120.

Fernandes, M. O'Brien, T. and Lis, J. T. (1994) "Structure and Regulation of Heat Shock Gene Promoters", in *The Biology of Heat Shock Proteins and Molecular Chaperones*, R. I. Morimoto, A. Tissieres and C. Georgopoulos, eds. (Cold Spring Harbor, NY: Cold Spring Laboratory Press), p375-393.

Goffeau, A., Barrell, B., Bussey, H., Davis, R., Dujon, B., Feldmann, H., Galibert, F., Hoheisel, J., Jacq, H., Murakami, Y., Philippsen, P., Tettelin, H. and Oliver, S. (1996) "Life with 6000 Genes", *Science*, 274(5287), p546, p563-567.

Hampson, S. and Kibler, D. (1996) "Large Plateaus and Plateau Search in Boolean Satisfiability Problems: When to Give Up Searching and Start Again", in DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 26, p437-455.

Harr, R., Haggstrom, M., Gustaffson, P. (1983) "Search Algorithm for Pattern Match Analysis of Nucleic Acid Sequences", Nucleic Acids Res., 11, p2943-2957.

Hertz, G., Hartzell III, G. and Stormo, G. (1990) "Identification of Consensus Patterns in Unaligned DNA Sequences Known to be Functionally Related", Computer Applications in Biosciences, Vol 6, No 2, p81-92.

Hertz, G. and Stormo, G. (1995) "Identification of Consensus Patterns in Unaligned DNA and Protein Sequences: A Large-Deviation Statistical Basis for Penalizing Gaps", in Proceedings of the 3rd International Conference on Bioinformatics and Genome Research, p201-216.

Hu, Y. and Kibler, D. (1996) "Generation of Attributes for Learning Algorithms", in Proceeding of the 13th National Conference on Artificial Intelligence, p806-811.

Hu, Y. (1998a) "Constructive Induction: Covering Attribute Spectrum", in *Feature Extraction, Construction and Selection: A Data Mining Perspective*, H. Liu and H. Motoda, eds. (Kluwer Academic Publisher), p257-272.

Hu, Y. (1998b) "Biopattern Discovery by Genetic Programming", in Proceedings of the 3rd Annual Genetic Programming Conference, p152-157.

Hu, Y., Sandmeyer, S. and Kibler. D. (1999) "Detecting Motifs from Sequences", in Proceedings of the 16th International Conference on Machine Learning.

Hughey, R. and Krogh, A. (1996) "Hidden Markov Models for Sequence Analysis: extension and analysis of the basic method", Computer Applications in Biosciences, Vol 12, No 2, p95-107.

Kobayashi, N. and McEntee, K. (1993) "Identification of cis and trans Components of a Novel Heat Shock Stress Saccharomyces cerevisiae", in Molecular and Cellular Biology, 13(1), p248-256.

Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A. and Wootton, J. (1993) "Detecting Subtle Sequence

Signals: A Gibbs Sampling Strategy for Multiple Alignments", SCIENCE, Vol 262, p208-214.

Lawrence, C. and Reilly, A. (1990) "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences", Protein: Structure Function and Genetics, 7, p41-51.

Quinlan, J. R. (1993) *C4.5 : Programs for Machine Learning*, Morgan Kaufmann, San Mateo, CA.

Staden, R. (1984) "Computer Methods to Locate Signals in Nucleic Acid Sequences", Nucleic Acids Res., 12, p505-519.

Stormo, G. (1988) "Computer Methods for Analyzing Sequence Recognition of Nucleic Acids", Annual Review of Biophysic and Biophysical Chemistry, 17, p241-263.

van Helden, J. V., Andre, B, and Collado-Vides, J. (1998) "Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies", Journal of Molecular Biology, 281, p827-842.

Wodicka, L., Dong, H., Mittmann, M., Ho, M. & Lockhart, D. (1997) "Genome-wide Expression Monitoring in *Saccharomyces cerevisiae*", Nature Biotechnology, 15, p1359-1367.