ELSEVIER

Computer Methods and Programs in Biomedicine 70 (2003) 11–20

Computer Methods and Programs in Biomedicine

www.elsevier.com/locate/cmpb

Finding subtle motifs with variable gaps in unaligned DNA sequences

Yuh-Jyh Hu *

Computer and Information Science Department, National Chiao-Tung University, 1001 Ta Shueh Road, Hsinchu, Taiwan, ROC

Received 2 May 2001; received in revised form 21 September 2001; accepted 4 October 2001

Abstract

Biologists have determined that the control and regulation of gene expression is primarily determined by relatively short sequences in the region surrounding a gene. These sequences vary in length, position, redundancy, orientation, and bases. Finding these short sequences is a fundamental problem in molecular biology with important applications. Though there exist many different approaches to signal (i.e. short sequence) finding, some new study shows that this problem still leaves plenty of room for improvement. In 2000, Pevzner and Sze proposed the Challenge Problem of motif detection. They reported that most current motif finding algorithms are incapable of detecting the target motifs in their Challenge Problem. In this paper, we show that using an iterative-restart design, our new algorithm can correctly find the target motifs. Furthermore, taking into account the fact that some transcription factors form a dimer or even more complex structures, and transcription process can sometimes involve multiple factors with variable spacers in between, we extend the original problem to an even more challenging one by addressing the issue of combinatorial signals with gaps of variable lengths. To demonstrate the effectiveness of our algorithm, we tested it on a series of the new challenge problem as well as real regulons, and compared it with some current representative motif-finding algorithms. © 2002 Elsevier Science Ireland Ltd. All rights reserved.

Keywords: Gene regulation; Subtle signals; Motif detection; Gaps; Transcription factors

1. Introduction

Multiple various genome projects have generated an explosive amount of biosequence data; however, our biological knowledge has not been able to increase in the same pace of the growth of biological data. This imbalance has stimulated the development of many new methods and devices to address issues such as annotation of new genes. One of the most promising new designs is the microarray gene chip technology which allows direct measurement of the expression level change of each gene in a genome in parallel [1,2]. Biologists can easily isolate co-regulated genes according to their gene expression level change. This will not only increase the efficiency of experiments on gene expression, but also provide a better macro view of gene behavior on a genomic scale.

* Tel.: + 886-3-573-1795.

E-mail address: yhu@cis.nctu.edu.tw (Y.-J. Hu).

PII: S0169-2607(01)00198-5

A cluster of co-regulated genes isolated by gene expression measurements can only show which genes in a cell have similar reaction to a stimulus. What biologists further want to understand is the mechanism that is responsible for the coordinated responses. The cellular response to a stimulus is controlled by the action of transcription factors. A transcription factor, which itself is a special protein, recognizes a specific DNA sequence. It binds to this regulatory site to interact with RNA polymerase, and thus to activate or repress the expression of a selected set of target genes. Given a family of genes characterized by their common response to a perturbation, the problem we try to solve is to find these regulatory signals (also known as motifs or patterns), i.e. transcription factor binding sites that are shared by the control regions of these genes.

The motif finding problem can be formulated as follows: given a sample of sequences defined over a set of symbols (e.g. A, G, C and T in the case of DNA sequences), and unknown patterns (motifs) implanted at various locations in the sequences, how can we find the unknown patterns? According to motif representations, motif significance measures and motif search strategies, many different approaches to this problem have been developed [3–9]. Though these algorithms have been proved effective in many different real domains, a new study reported that several representative motif-finding algorithms are unable to detect the subtle motifs in some particular form, and this was introduced as the Challenge Problem of motif finding [10]. Due to the fact that transcription factors may form a dimer or more complex structures, and some transcription initiations may require the binding of two or more transcription factors at the same time, we further extend the Challenge Problem by addressing the issue of combinatorial signals with gaps of variable lengths. Most of the current approaches can only find motifs consisting of continuous bases without gaps. Some methods have been proposed to deal with motifs or alignments with gaps, but they either limit the focus on fixed-gaps [11,12] or use other less expressive representations than the weight matrix, e.g. regular expression-like languages or the IUPAC code [13–16]. To alleviate the limitations of current approaches, we introduce a new algorithm called MERMAID (Matrix-based Enumeration and Ranking of Motifs with gAps by an Iterative-restart Design), which adopts the matrix for motif representation, and is capable of dealing with gaps of variable lengths. This presentation expands upon work by others by combining multiple types of motif significance measures with an improved iterative sampling technique. We demonstrate its effectiveness in both the original and the extended Challenge Problems, and compare its performance with that of several other major motif finding algorithms. To verify its feasibility in real-world applications, we also tested MERMAID on many families of yeast genes that share known regulatory motifs.

2. Background

The identification of sequence motifs is a fundamental but important approach for suggesting good candidates for biologically functional regions that may be responsible for gene regulation. Fundamentally gene regulation is determined by chemical reactions which are, in turn, controlled by the shape and electrostatic charges of the molecules involved. One such instance of this is the interaction between regulatory proteins and their target binding sites. The significance of this is that this can lead to a coordination of regulation via a combination of motifs. Unfortunately this information is not typically available. We expect that the local shape of a binding or receptor site will be primarily determined by the bases involved, acknowledging the fact that non-local base changes can affect local shape.

The analysis of non-coding regions in genomes in order to understand the control mechanism is a difficult problem. 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. Research on finding subtle regulatory signals has been around for many years, and still draws a lot of attention because it is one of the most crucial steps in the study of genomics. Emerging knowledge of genome-wide 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.

Despite the fact that there already exist many various algorithms, this problem is nevertheless far from being resolved according to Pevzner and Sze [10]. They found several widely used motif-finding algorithms failed on the Challenge Problem defined as what follows.

Let $S = \{s_1, s_2, \ldots, s_t\}$ be a sample of t n-letter sequences.

Each sequence contains an (l, d)-signal, i.e. a signal of length l with d mismatches. The problem is how to find the correct (l, d)-signals.

In their experiments, they implanted a (15, 4)-signal in a sample of 20 sequences. To verify the effect of the sequence length, they varied n from 100 to 1000. The experimental results showed that as the sequence length increased, the performance of MEME [5], CONSENSUS [3] and the Gibbs sampler [4] decreased dramatically. There are two causes to their failures. First, the algorithms may lodge in local optima. The increase of the sequence length can incur more local optima, and further aggravates the problem. Second, they rely on the hope that the instances of the target signal appearing in the sample will reveal the signal itself. However, in the Challenge Problem, there are no exact signal occurrences in the sample, only variant instances with four mismatches instead. Pevzner and Sze proposed WINNOWER and SP-STAR to solve the Challenge Problem, but the applicability of WINNOWER is limited by its complexity and the performance of SP-STAR drops significantly like others as the sequence length increases [10].

3. Design considerations

Most current approaches based on greedy or stochastic hill-climbing algorithms optimize the weight matrix with all positions within a sequence [3,4]. This is not only inefficient, but may also increase the chance of getting trapped in local optima in case of subtle signals contained in long sequences due to a greater number of similar random patterns coexisting in the sequences. To avoid this drawback, we begin by allowing each substring of length l to be a candidate signal. We convert this particular substring into a probability matrix, adopting an idea from Ref. [5]. This gives us a set of seed probability matrices to be used as starting points for iterative improvement. We use the seed probability matrix as a reference to locate the potential signal positions with match scores above some threshold. The optimization procedure only checks these potential positions instead of all possible locations in a sequence. By directing the attention to the patterns same as or close to the substring that is considered a motif candidate, we can significantly constrain the search space during the iterative improvement process.

However, when the target signal is very subtle, e.g. (15, 4)-signal, the bias that we only consider the selected potential signal positions becomes harmful. This bias is based on the assumption that the instances of the target signal existing in the sample have sufficient regularity so that we can finally derive the correct target signal from these instances through optimization. Unfortunately, this optimistic assumption does not hold if the regularity represented by the signal instances is inadequate to distinguish themselves from similar random patterns. As a consequence, the chance of mistaking random patterns for real signal instances gets higher. The algorithm may thus be misled to other variant patterns than the correct signal.

When dealing with subtle signals, it is not guaranteed stochastic optimization can find the correct target signal due to the influence of similar random patterns. However, the pattern it converges to must be close to the target itself because the random patterns must carry some resemblance to the target signal; otherwise, they would not be selected to participate in the optimization process. Suppose the target signal is the most conserved pattern in the sample as usually expected and we use one signal instance as the seed for optimization. No matter what pattern it finally converges to, this pattern is at least closer to the target signal than the substring (i.e. the signal instance in the sample) used as the seed even if it may not be the same as the target. Since the converged pattern is closer to the target signal, one way to further refine this pattern is to reuse it as a seed, and run through the optimization process again. We can iteratively restart the optimization procedure, using the refined pattern as a new seed, until no improvement is shown. With this iterative restart strategy, we expect to successfully detect subtle signals like (l, d)-signals in the Challenge Problem.

Pevzner and Sze introduced some extension to SP-STAR to deal with gapped signals, but their method typically addressed the fixed-gap issue only. However, in some real domains, motifs may contain gaps of variable lengths, and simultaneous and proximal binding of two or more transcription factors may be required to initiate transcription. Therefore, a natural extension to the Challenge Problem proposed by Pevzner and Sze is to find combinatorial (l, d)-signals. A combinatorial (l, d)-signal may consist of multiple (l, d)-signals as its components, and the length of gap between two components may vary within a given range. For example, a (l, d)–X(m, n)–(l, d)-signal is one that has two (l, d)-signals with a gap of variable lengths between m and n bases. Note that the signal length and the number of mutations may be different in various components.

There are generally two approaches to finding combinatorial signals. The first one is a two-step approach. We first find signal component candidates. In the second step, we use the component candidates to form signal combinations and verify their significance [17]. This approach is effective only if the signal components by themselves are significant enough so they can be isolated in the first step for later combination check. In cases that the signal components gain significance only in combinations, the earlier approach may overlook the interaction between components and thus fail to find their combinations. To avoid this limitation, an alternative approach is to find combinatorial signals directly. Based on the design consideration mentioned above, we developed MERMAID to deal with subtle combinatorial signals. MERMAID enumerates all possible substring combinations with different gap lengths within a given range. It constructs a probability matrix for each signal component, and then applies an iterative-restart procedure to optimize the matrices. Provided that the gap range is relatively small, the time complexity of MERMAID does not increase dramatically.

4. System description

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 in MERMAID. Given a sample of N biosequences, MERMAID first converts a substring combination into an initial matrix combination, and then carries out an iterative improvement search to optimize the consensus quality of the matrix combination. To avoid the lodge on local optimum, it then applies an iterative restart strategy to refine the matrices. The same process is repeated for all substring combinations in the sample to produce a user-defined number, d, of matrix combinations that maximize motif significance that is based on the combination of multiple types of motif quality measures, including consensus [3], multiplicity [9] and coverage [6]. These d matrix combinations are combinatorial motif candidates, and can be later ranked according to its significance.

Following Ref. [18], the consensus quality of a matrix is derived from the entropy. The entropy is calculated from the probability that each base occurs at each position in the motif, Pm. More precisely, the entropy for a particular column n in the matrix is given by:

$$E(n) = -\sum_{i=b1}^{b4} Pmi \cdot \log_2 Pmi$$

where $b1,\ldots,b4$ are the bases A, G, C, and 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, where w is the width of the motif.

$$\mathrm{Con}(b) = \frac{1}{w}\sum_{1}^{w} C(n)$$

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

$$\mathrm{Mul}(b) = \frac{\mathrm{occ}_S(b)}{\mathrm{occ}_G(b)}$$

where $\mathrm{occ}_S(b)$ is b's occurrences in a given family S, and $\mathrm{occ}_G(b)$ is b's occurrences in genome.

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

$$\mathrm{Cov}(b) = \frac{\mathrm{cont}_S(b)}{|S|}$$

where $\mathrm{cont}_S(b)$ is the number of sequences in S that contain b, and $|S|$ is the total number of sequences in S.

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:

$$\mathrm{Con}_{\mathrm{norm}}(b) = \frac{\mathrm{Con}(b)}{\mathrm{MAX(Con)}}$$

$$\mathrm{Mul}_{\mathrm{norm}}(b) = \frac{\mathrm{Mul}(b)}{\mathrm{MAX(Mul)}}$$

$$\mathrm{Cov}_{\mathrm{norm}}(b) = \frac{\mathrm{Cov}(b)}{\mathrm{MAX(Cov)}}$$

where MAX(Con) is the maximum consensus quality of the d motifs, MAX(Mul) is the maximum multiplicity significance of the d motifs, and MAX(Cov), is the maximum motif coverage of the d motifs.

It is important to evaluate all objective functions above in conjunction because it may be easy to optimize any single one separately. However, for a motif to be significant we demand that it be conserved as well have good coverage and high multiplicity. In order to quantify this with a single measure, we borrow the idea of F-measure, a weighted combination [21], and propose the final merit measure of a motif b as defined below:

$$\mathrm{Merit}(b) = \frac{1}{\frac{1}{3}\left(\frac{1}{\mathrm{Con}_{\mathrm{norm}}(b)} + \frac{1}{\mathrm{Mul}_{\mathrm{norm}}(b)} + \frac{1}{\mathrm{Cov}_{\mathrm{norm}}(b)}\right)}$$

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.

The main process flow of MERMAID is divided into four steps. First, it translates substring combinations into matrices. Each matrix represents a component of a combinatorial motif. Second, it filters the potential motif positions in the sample of sequences. Third, given the set of potential motif positions, it performs an iterative stochastic optimization procedure to find motif candidates. Finally, it ranks and reports these candidates based on the motif significance.

A pseudo-code description of the iterative-restart optimization procedure in MERMAID is given in Fig. 1. Let n be the sequence length. The pseudo-codes (4)–(9) scan the entire sample against each matrix m to find the highest match scoring substring combination in each sequence, locate the potential positions of the combinatorial motif, and form an initial matrix combination M. These totally take $O(n \cdot G^{N-1} \cdot |S|)$ operations, where G is the maximum gap range and N is the total number of motif components. Let p be the maximum number of potential positions in a sequence, p typically $\ll n$. The inner repeat-loop (10)–(14) takes $(p \cdot L)$ operations to check different positions, where L is a constant for the cycle limit. Pseudo-codes (15)–(19), which scan the entire sample against matrix M to isolate signal repeats, and form the final probability matrix FM, also take $O(n \cdot G^{N-1} \cdot |S|)$ operations. From above, the outer repeat-loop (3)–(21) totally takes $O(L(2n \cdot G^{N-1} \cdot |S| + pL) = O(n \cdot G^{N-1} \cdot |S|)$. Now considering the outer for-loop (1)–(21) and (22)–(23), we conclude the whole procedure is bounded by $O(n\cdot G^{N-1}\cdot|S|\cdot n\cdot G^{N-1}\cdot|S|) = O((n\cdot G^{N-1}\cdot|S|)^2)$. When G and N are relatively small, $O((n\cdot G^{N-1}\cdot|S|)^2) = O((n\cdot|S|)^2)$, which is the same as MEME and SP-STAR, but lower than WINNOWER's $O((n\cdot|S|)^{k+1})$, where k is the clique size, $k \geq 2$ in general.

5. Status report

One of the goals of this paper is to demonstrate that by applying a simple iterative restart strategy, our motif detection algorithm is capable of finding subtle signals, e.g., (15, 4)-signal. Based on its definition, we reproduced the Challenge Problem, and used it to compare our new algorithm with others.

Pevzner and Sze's study [10] showed that for a (15, 4)-signal, CONSENSUS, the Gibbs sampler and MEME start to break at sequence length 300–400 bp. Their system called SP-STAR breaks at length 800–900, and their other algorithm named WINNOWER performs well through the whole range of lengths till 1000 bp. Using the same data generator to create data samples (thanks to Sze for providing the program), we demonstrate that MERMAID is competitive with others. Moreover, in order to show that it is the synergy of the iterative restart strategy and the optimization procedure combined with the multiple objective functions in MERMAID that helps find the subtle signals, we implanted in the sample the motif found by MEME with minimum mismatches to the target signal at a random position. We then reran MEME. We repeated the above process, and checked whether this iterative restart strategy alone could improve MEME's performance. The reason we tested MEME is that MERMAID adopts the same motif enumeration method as MEME. Since MEME exhaustively tests every substring in the sample, the implanted substring will be used in the next run. We only implanted the motif closest to the real signal (i.e. minimum mismatches) to ensure that the base distribution in the sample was nearly unchanged. Though we did not actually re-code MEME, this approximate simulation could still effectively reflect its performance.

```
Given: a set of biosequences, S
       the total width of a combinatorial motif, W (excluding gaps)
       the maximal gap range, G
       the number of components in a combinatorial motif, N
       the cycle limit, L
Return: a set of ranked motif candidates, C

For each substring combo s in S Do
   Set s to ss as a seed

   Repeat
      Translate each substring in ss into candidate probability matrix m via:
          m(i,base) = .50 if base occurs in position i
                    = .17 otherwise

      Find highest match scoring substring combo in each sequence in S
      Compute the mean of the highest match scores in S
      For each sequence in S Do
         Set Potential Positions to those with match score >= mean

      Randomly choose a Potential Position in each sequence
          to initialize matrix combo M
      Repeat
         Randomly pick a sequence s in S
         Check if M's significance can be improved by using a
             different Potential Position in s
         Update matrix combo M
      Until (no improvement in M's consensus) or (reach the cycle limit L)

      Compute the mean of match scores of substring combo contributing to M
      For each sequence s in S Do
         Isolate motif repeats to those with match score >= mean
      Form the final matrix combo FM with all repeats in S
      Convert matrix combo FM into string combo ss as a new seed
   Until (no improvement in FM's merit) or (reach the cycle limit L)
   Put FM in C

Sort all motif candidates in C according to merit
Return C
```

Fig. 1. Pseudo-code of MERMAID.

To keep the consistency, we followed Pevzner and Sze's test methodology as mentioned above. We tested each algorithm on eight random samples. Each sample contains 20 i.i.d. sequences, each of 1000 bp. Each sequence contains one (15, 4)-signal at random position. The objective of this experiment is to demonstrate the performance of various algorithms in detecting the implanted signals. The numbers in Table 1 present the *performance coefficients* as defined in Ref. [10] averaged over eight samples. Let K be the set of known signal positions in a sample, and let P be the set of predicted positions. The *performance coefficient* is defined as $P \cap K / P \cup K$.

Table 1 indicates that MERMAID outperforms CONSENSUS, the Gibbs sampler and MEME (with or without iterative restart) by a significant scale. Note that the performance coefficients of WINNOWER and SP-STAR reported in Ref. [10]

Table 1
Comparison of performances in detecting (15, 4)-signals

CONSENSUS	Gibbs sampler	MEME	MEME (w/iterative restart)	Oligonucleotide analysis (van Helden)	WINNOWER	SP-STAR	MERMAID
0.06	0.11	0.02	0.09	0.00	0.88	0.23	0.75

are included only for reference because we did not have access to these two systems at the time. However, this indirect evidence may suggest that MERMAID performs better than SP-STAR, and is expected to be comparable with WINNOWER. Also note that though the performance of MEME with iterative restart is a bit better than the original MEME, yet the result is not significant. This not only shows that iterative restart alone may not improve the performance, but also provides the evidence that the success of MERMAID is attributed to the synergy of the iterative restart strategy and the optimization procedure combined with the multiple objective functions.

We also tested MERMAID on ten real regulons collected by van Helden et al. [9] to verify its usefulness in finding motifs in real-world domains. MERMAID successfully identified all the known motifs in each regulon.

For motifs with gaps of variable lengths, we first tested MERMAID on (6, 1)–X(m, n)–(6,1)-signals in a set of 20 sequences, each of length 1000 bp, where m and n present the lower and the upper bound of the gap between two (6, 1)-signals. Without losing the generality, we fixed the lower bound, m, at 1, and varied the upper bound, n, from three to nine in each experiment. For example, in the first experiment, we set n to be 3, which means each 1000 bp sequence contains a (6, 1)–X(1, 3)–(6, 1)-signal, i.e. two (6, 1)-signals with a gap of one to three bases at random in between. The purpose of these experiments is to verify the variance tolerance of MERMAID to the gap range. As the gap range increases, the search space of the target signals increases, and thus makes it harder for MERMAID to locate the right motifs.

The experimental results are presented in Table 2. It shows that the performance coefficient of MERMAID is quite stable till n reaches 7, then breaks at 9. For comparison, we also tested CONSENSUS, Gibbs sampler, MEME and oligonucleotide analysis on the same data sets. The results show the performance coefficient of each of the above algorithms is near zero in all gap ranges.

In addition to the artificial problem, we also tested MERMAID on several real regulons [11] in which the known binding sites have fixed gaps. The summary of the regulons is presented in Table 3, and we show the results in Table 4. In the fourth column of Table 4, the number within each bracket means the rank of the signal found by MERMAID. The experimental results indicate MERMAID, which was originally developed to deal with variable gaps, performs well on real domains where motifs have fixed gaps. The motifs converted from the weight matrices discovered by MERMAID are all very similar to the known motifs.

6. Lessons learned

The difficulty of finding the biologically meaningful motifs results from the variability in (1) the bases at each position in the motif, (2) the location of the motif in the sequence and (3) the multiplicity of motif occurrences within a given

Table 2
Performance of MERMAID on (6, 1)–X(1, n)–(6, 1)-signal

$n=3$	$n=5$	$n=7$	$n=9$
0.91	0.88	0.90	0.56

Table 3
Summary of regulons used in the experiments

Family	Genes
GAL4	GAL1, GAL2, GAL7, GAL80, MEL1, GCY1
CAT8	ACR1, lCL1, MLS1, PCK1, FBP1
HAP1	CYB2, CYC1, CYC7, CTT1, CYT1, ERG11, HEM13 HMG1, ROX1
LEU3	GDH1, lLV1, LEU1, LEU2, LEU4
LYS	LYS1, LYS2, LYS4, LYS9, LYS20, LYS21
PPR1	URA1, URA3, URA4
PUT3	PUT1, PUT2

sequence. In addition, the short length of many biologically significant motifs and the fact that motifs gain biological significance only in combinations make them difficult to determine [11,19]. Though many protein–DNA-binding domains establish contact with a limited number of adjacent nucleotides, a good number of transcription factors bind to a pair of or more relatively short conserved nucleotide sequences separated by non-conserved regions.

Various approaches have been developed to identify shared motifs from functionally related biosequences. Each method is based on a different model for the motifs. We review some of the methods for the detection of motifs, and compare them to our new approach. These methods were selected because they are well developed. They are freely available over the Internet, and represent a spectrum of different approaches as shown in Table 5.

Despite the fact that these approaches only find motifs of continuous nucleotides, several algorithms adopting the ideas from these approaches are further developed to detect gapped motifs [7,11,12]. Unlike current gapped motif detection algorithms that can only deal with fixed gaps, MERMAID was developed to identify subtle combinatorial signals with variable spacers in between. Our experiments showed that MERMAID not only effectively detected combinatorial signals composed of proximal components in artificial domains, but also successfully identified the known motifs with gaps in real regulons. The ability of MERMAID to extract single motifs with variable spacers can be further generalized to identify motif combinations. As gene expression often requires the binding of multiple transcription factors to specific DNA sequences, with the generalized capability MERMAID can discover the potential interaction between transcription factors.

7. Future plans

In this paper we have described a new subtle signal detection algorithm called MERMAID, which iteratively restart a multi-strategy optimization procedure combined with complementary objective functions to find motifs. The experimental results show that the system performs significantly better than most current algorithms in the Challenge Problem. To argue the success of MERMAID is attributed to the synergy of iterative restart and other components in the system, i.e. optimization procedures and objective functions, we used MEME as an example to demonstrate that simply attaching an iterative restart strategy to an arbitrary motif finding algorithm shows little improvement.

For future work, we aim to improve MERMAID in two directions. One is efficiency and the other is generality. First, the optimization process in MERMAID for a single candidate is independent of each other. Therefore, MERMAID can be easily implemented on a parallel or distributed system to improve its efficiency. Second, MERMAID only performs well on combinatorial signals with gaps within a relatively tight range. Despite that higher consensus quality of each signal component allows for wider gap ranges, MERMAID has difficulties finding very subtle combinatorial signals whose components reveal little locality. A wider range of gap length produces a larger search space for motif-finding algorithms, and in such cases, it is computationally prohibited to enumerate all possibilities exhaustively. Thus we plan to apply another stochastic sampling technique to search through the space, and incorporate domain knowledge when available to constrain the search space.

Table 4
Summary of MERMAID's analysis results in regulons

Family	Known motifs	Dyad-analysis by van Helden et al.	MERMAID
GAL4	CGGRnnRCYnYnCnCCG	TCGGAn9TCCGA TCGGAn8CGCCGA CCGGAn9TCCGA	CGG–X(11)–CCG [1] A 0.0 0.0 0.0–X(11)–0.0 0.0 0.0 G 0.0 0.9 1.0–X(11)–0.0 0.0 1.0 C 1.0 0.1 0.0–X(11)–0.9 1.0 0.0 T 0.0 0.0 0.0–X(11)–0.1 0.0 0.0
CAT8	CGGnnnnnnGGA	CGCn4ATGGAA	CGG–X(6)–GGA [1] A 0.0 0.0 0.0–X(6)–0.0 0.0 1.0 G 0.0 1.0 1.0–X(6)–1.0 1.0 0.0 C 1.0 0.0 0.0–X(6)–0.0 0.0 0.0 T 0.0 0.0 0.0–X(6)–0.0 0.0 0.0
HAP1	CGGnnnTAnCGG	GGAn5CGGC	CGG–X(6)–CGG [10] A 0.0 0.0 0.0–X(6)–0.0 0.0 0.0 G 0.3 0.8 0.9–X(6)–0.0 1.0 1.0 C 0.6 0.0 0.1–X(6)–1.0 0.0 0.0 T 0.1 0.2 0.0–X(6)–0.0 0.0 0.0
LEU3	RCCGGnnCCGGY	ACCGGCGCCGGT	GCCGG–X(2)–CCGGC [3] A 0.1 0.0 0.0 0.0 0.0–X(2)–0.1 0.0 0.0 0.0 0.4 G 0.8 0.0 0.0 0.8 0.9–X(2)–0.0 0.2 1.0 1.0 0.0 C 0.0 1.0 0.9 0.2 0.0–X(2)–0.9 0.8 0.0 0.0 0.6 T 0.1 0.0 0.1 0.0 0.1–X(2)–0.0 0.0 0.0 0.0 0.0
LYS	WWWTCCRnYGGAWWW	AAATTCCG	TTCCR–X(1)–YGGAA [10] A 0.0 0.0 0.1 0.0 0.5–X(1)–0.0 0.0 0.0 0.9 1.0 G 0.0 0.1 0.0 0.0 0.5–X(1)–0.1 1.0 1.0 0.1 0.0 C 0.0 0.1 0.9 1.0 0.0–X(1)–0.6 0.0 0.0 0.0 0.0 T 1.0 0.8 0.0 0.0 0.0–X(1)–0.3 0.0 0.0 0.0 0.0
PPR1	WYCGGnnWWYKCCGAW	CGGn6CCG	TTCGG–X(2)–AACCCCGAG [4] A 0.0 0.0 0.0 0.0 0.0–X(2)–1.0 0.7 0.0 0.0 0.0 0.0 0.0 0.4 0.0 G 0.0 0.0 0.0 1.0 1.0–X(2)–0.0 0.3 0.3 0.0 0.0 0.0 0.7 0.3 0.7 C 0.3 0.0 1.0 0.0 0.0–X(2)–0.0 0.0 0.7 0.7 1.0 0.7 0.3 0.3 0.0 T 0.7 1.0 0.0 0.0 0.0–X(2)–0.0 0.0 0.0 0.3 0.0 0.3 0.0 0.0 0.3
PUT3	YCGGnAnGCGnAnnnCCGA CGGnAnGCnAnnnCCGA	CGGn10CCG	TCGG–X(10,11)–CCGA [1] A 0.0 0.0 0.0 0.0–X(10,11)–0.0 0.0 0.0 1.0 G 0.0 0.0 1.0 1.0–X(10,11)–0.0 0.0 1.0 0.0 C 0.0 1.0 0.0 0.0–X(10,11)–1.0 1.0 0.0 0.0 T 1.0 0.0 0.0 0.0–X(10,11)–0.0 0.0 0.0 0.0

Table 5
Characteristics of some representative motif-finding algorithms

Algorithm	Search strategy	Objective function	Representation
CONSENSUS	Beam search	Information content	Frequency matrix
Gibbs Sampler	Stochastic hill-climbing	Ratio of pattern probability to background probability	Probabilistic matrix
MEME	EM [20] variant	Likelihood	Probabilistic matrix
van Helden et al.	Exhaustive	Statistical significance assuming binomial distribution	Base string

References

[1] J. DeRisi, V. Iyer, P. Brown, Exploring the metabolic and genetic control of gene expression on a genomic scale, Science 278 (1997) 680–696.

[2] L. Wodicak, H. Dong, M. Mittmann, M. Ho, D. Lockhart, Genome-wide expression monitoring in *Saccharomyces cerevisiae*, Nat. Biotechnol. 15 (1997) 1359–1367.

[3] G. Hertz, G. Hartzell III, G. Stormo, Identification of consensus patterns in unaligned DNA sequences known to be functionally related', Comput. Appl. Biosci. 6 (2) (1990) 81–92.

[4] C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald, J. Wootton, Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignments, Science 262 (1993) 208–214.

[5] T. Bailey, C. Elkan, Unsupervised learning of multiple motifs in biopolymers using expectation maximization, Mach. Learning 21 (1995) 51–80.

[6] Y. Hu, S. Sandmeyer, D. Kibler, Detecting motifs from sequences, Proceedings of the 16th International Conference on Machine Learning, 1999, pp. 181–190.

[7] M. Li, B. Ma, L. Wang, Finding similar regions in many strings, Proceedings of the 31st ACM Annual Symposium on Theory of Computing, 1999, pp. 473–482.

[8] M. Gelfand, E. Koonin, A. Mironov, Prediction of transcription regulatory sites in Archaea by a comparative genomic approach, Nucleic Acids Res. 28 (3) (2000) 695–705.

[9] J. van Helden, B Andre, J. Collado-Vides, Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies, J. Mol. Biol. 281 (1998) 827–842.

[10] P. Pevzner, S. Sze, Combinatorial approaches to finding subtle signals in DNA sequences, Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, 2000.

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

[12] E. Rocke, M. Tompa, An algorithm for finding novel gapped motifs in DNA sequences, RECOMM-98, 1998, pp. 228–233.

[13] A. Bairoch, PROSITE: a dictionary of sites and patterns in proteins, Nucleic Acids Res. 20 (1992) 2013–2018.

[14] I. Jonassen, PhD Thesis, Department of Informatics, University of Bergen, Norway, 1996.

[15] Y. Hu, Biopattern discovery by genetic programming, Proceedings of the 3rd Annual Genetic Programming Conference, 1998, pp. 152–157.

[16] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D.J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs, Nucleic Acids Res. 25 (17) (1997) 3389–3402.

[17] Y. Hu, S. Sandmeyer, C. McLaughlin, D. Kibler, Combinatorial motif analysis and hypothesis generation on a genomic scale, Bioinformatics 16 (2000) 222–232.

[18] Y. Hu, A framework for genomic gene expression analysis, Proceedings of the International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences, 2000, pp. 165–175.

[19] Y. Hu, An integrated approach for genome-wide gene expression analysis, Computer Methods and Programs in Biomedicine, vol. 65, no. 3, Elsevier Science, Amsterdam, 2001, pp. 163–174.

[20] C. Lawrence, A. Reilly, An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences, Protein: Struct. Funct. Genet. 7 (1990) 41–51.

[21] D. Lewis, W.A. Gale, A sequential algorithm for training text classifier, Proceedings of the 17th Annual International ACM-SIGIR Conference on Research and Development in Information Retrieval, 1994, pp. 3–12.