Biological Sequence Data Mining

Yuh-Jyh Hu

Computer and Information Science Department
National Chiao-Tung University
1001 Ta Shueh Rd., Hsinchu, Taiwan
yhu@cis.nctu.edu.tw

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/motif (i.e. short sequence) finding, in 2000 Pevzner and Sze reported that most current motif finding algorithms are incapable of detecting the target signals in their so-called Challenge Problem. In this paper, we show that using an iterative-restart design, our new algorithm can correctly find the targets. 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, we extend the original problem to an even more challenging one. We address the issue of combinatorial signals with gaps of variable lengths. To demonstrate the efficacy of our algorithm, we tested it on a series of the original and the new challenge problems, and compared it with some representative motif-finding algorithms. In addition, to verify its feasibility in real-world applications, we also tested it on several regulatory families of yeast genes with known motifs. The purpose of this paper is two-fold. One is to introduce an improved biological data mining algorithm that is capable of dealing with more variable regulatory signals in DNA sequences. The other is to propose a new research direction for the general KDD community.

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 [1][2]. Once the Human Genome Project is completed, it can be expected that related experiments will be carried out soon. The tough computational challenges resulting from large-scale genomic experiments lie in the specificity and complexity of the biological processes, e.g., how we identify the genes directly involved in diseases, how these genes function, and how these genes are regulated, etc. Answers to the questions above are absolutely related

L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp. 228–240, 2001.

to the future of health care and genomic medicine that will lead to personalized therapy. The success of the future health care will definitely affect the entire human race in terms of life quality and even life span. Though the content of this paper is focused on one specific biological problem, another important objective of this paper is to draw the attention of the general KDD community to a new research area which needs considerable efforts and novel techniques from a wide variety of research fields, including KDD.

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 (aka motifs or patterns), i.e. transcription factor binding sites, that are shared by the control regions of these genes.

It has been 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. In any case these characteristics make the problem computationally difficult. For example, a typical problem would be: given 30 DNA sequences, each of length 800, find a common pattern of length 8. Let us simplify the problem, as many algorithms do, and assume the pattern occurs exactly once in each sequence. This means that there are approximately 800^{30} potential locations for a motif candidate. 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 fundamental but important step in the study of genomics [3-9]. Despite that there already exist many various algorithms, this problem is nevertheless far from being resolved [10]. They found several widely used motif-finding algorithms failed on the Challenge Problem as follows.

Let $S = \{s_1, ..., s_t\}$ 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)-signal.

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 [3], CONSENSUS [4] and the Gibbs sampler [5] 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 4 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.

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 [11-13] or use other less expressive representations than the weight matrix, e.g., regular expression-like languages or the IUPAC code [14][15]. To alleviate the limitations of current approaches, we introduce a new algorithm called MERMAID, 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

There are three main interrelated computational issues: the representation of a pattern, the definition of the objective function, and the search strategy. While we examine the algorithms on computational grounds, the final, gold-standard is how well the algorithm does at predicting motifs.

2.1 Representation

As the primary DNA sequences are described by a double-stranded string of nucleic bases {A,C,G,T}, the most basic pattern representation is the exact base string. Due to the complexity and flexibility of the motif binding mechanism, there is rarely any motif that can be exactly described by a string of nucleic bases. To obtain more flexibility, the IUPAC code was designed, which extends the expressiveness of the simple base string representation by including all disjunctions of nucleotides. In this language there is a new symbol for each possible disjunction, e.g. W represents A or T.

A more informative pattern representation is a probability matrix in which each element reflects the importance of the base at a particular position. Such matrices can be easily translated into the IUPAC code, while the converse is not true. These matrices are often transformed from the observed occurrence

frequencies. For example, in the NIT regulatory family [6] which contains 7 members, a possible 6-base motif matrix is illustrated in Fig. 1. The normalized matrix is also shown in this figure.

```
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 to    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
```

Fig. 1. A 6-base Motif Matrix Example

2.2 Objective Function

The purpose of an objective function is to approximate the biological meanings of the patterns in terms of a mathematical function. The objective function are heuristics. Once the objective function is determined, the goal is to find those patterns with high objective function value. Different objective functions have been derived from the background knowledge, such as the secondary structures of homologous proteins, the relation between the energetic interactions among residues and the residue frequencies, etc [17][18]. Objective functions based on the information content or its variants were proposed [4][5]. Others evaluate the quality of the pattern by its likelihood or by some other measures of statistical significance [3][13].

Even though there are many different objective functions currently used, it is still unclear what is the most appropriate object function or the best representation for patterns that will correspond to biological significant motifs. More likely, additional knowledge will need to be incorporated to improve motif characterization. In the final analysis, the various algorithms can only produce candidate motifs that will require biological experiments to verify.

2.3 Search Strategy

If one adopts the exact string representation, then one can exhaustively check every possible candidate. However this approach is only able to identify short known motifs or partial long motifs [13]. Therefore, the primary representation used is a probability matrix [3][4][5][7]. Once one accepts a probability matrix as the representation, then there is no possibility for an exhaustive search. Initial approaches started with hill-climbing strategies, but these typically fell into local optimum. Standard approaches to repairing hill-climbing, such as beam and stochastic search, were tried next[4]. The current approaches involve a mixture of sampling and stochastic iterative improvement. This avoids the computational explosion and maintains or improves the ability to find motifs [3][5][7].

3 MERMAID: Matrix-Based Enumeration and Ranking of Motifs with gAps by an Iterative-Restart Design

According to the objective function they apply, most current approaches based on greedy or stochastic hill-climbing algorithms optimize the probability matrix with all positions within a sequence [4][5]. 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 can begin by allowing each substring of length l to be a candidate signal. We then convert this particular substring into a probability matrix, adopting an idea from [3]. 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.

Nevertheless, when the target signal is very subtle, e.g., (15,4)-signal, the way that we only consider the selected potential signal positions becomes biased. 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 optimization process may thus converge to other variant patterns than the correct signal.

When dealing with subtle signals, a stochastic approach is not guaranteed to find the correct target signal owing 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 is not 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 again. We can iteratively restart the optimization procedure with 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 proposed some extension to SP-STAR to deal with gapped signals [10], 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[9][14]. Therefore, a natural extension to the Challenge Problem is to find combinatorial (l, d)-signals. A combinatorial (l, d)-signal 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 is a two-phase approach. We find signal component candidates in the first phase. In the second phase, we use the component candidates to form and verify signal combinations [16]. This approach is effective when the signal components are significant enough *per se* so they can be identified in the first phase for later combination check. In cases that the signal components gain significance only in combinations, the former approach may overlook the interaction between components and thus fail to find meaningful combinations. To avoid this limitation, an alternative approach is to find combinatorial signals directly. We developed MERMAID (Matrix-based Enumeration and Ranking of Motifs with gAps by an Iterative-restart Design) to deal with subtle combinatorial signals.

The main process flow of MERMAID is divided into four steps. Given a biosequence family, it first translates substring combinations into matrices. We convert this particular substring into a probability matrix in two steps, adopting an idea from [3]. First we fix the probability of every base in the substring 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. We also tried setting X to 0.6. The result showed no significant difference. Each matrix represents a component of a combinatorial motif. This step gives us a set of seed probability matrices to be used as starting points for iterative improvement. Second, it filters the potential motif positions in the family of sequences. Note that each single motif is derived from a substring combination. Thus, besides the matrices, MERMAID also keeps the locations of substrings for all potential motifs to deal with the flexible gaps. Third, given the set of potential motif positions that include the location of each motif component (i.e. substring), it performs an iterative stochastic optimization procedure to find motif candidates. Finally, it ranks and reports these candidates according to the motif significance that is based on the combination of different types of quality measures, including consus [4], multiplicity [6][13] and coverage [7]. The consus quality is derived from the relative entropy, which is used to measure how well a motif is conserved. The multiplicity is defined as the ratio of the number of motif occurrences in the family to that in the whole genome. This measures the representativeness of a motif in a family relative to the entire genome, and consequently, discounts motifs which are common everywhere, such as tandem repeats or poly A's. We define motif coverage as the ratio of the number of the sequences containing the motif to the total number of biosequences in the family. This reflects the

importance of a motif's being commonly shared by functionally related family members. Due to limited space, please refer to [16] for more details.

A pseudocode description of the iterative-restart optimization procedure in MERMAID is given in Fig. 2. Let n be the sequence length. The pseudocode (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. Pseudocode (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.

4 Experimental Results

One of the goals of this paper is to demonstrate that enhanced by applying an iterative restart strategy, our new motif detection algorithm is able to find 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 300-400bp. Their system called SP-STAR breaks at length 800 to 900, and their other algorithm named WINNOWER performs well through the whole range of lengths till 1000bp. Using the same data generator to create data samples (thanks to Sze for providing the program), we demonstrate our new algorithm is competitive with other systems. We tested MERMAID over eight samples, as Pevzner and Sze did, each containing 20 i.i.d. sequences of length 1000bp. The comparison of performance of the various algorithms is shown in Table 1. The numbers in Table 1 present the *performance coefficients* as defined in [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 $|K \cap P|/|K \cup P|$.

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

```
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


  (1)For each substring combo s in S Do
  (2)  Set s to ss as a seed

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

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

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

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

 (22)Sort all motif candidates in C according to significance
 (23)Return C
```

Fig. 2. Pseudocode of MERMAID

Table 1. Comparison of performance for (15,4)-signals in 20 i.i.d. sequences of length 1000bp

CONSENSUS	Gibbs	MEME	MEME (w/ iterative restart)	oligonucleotide analysis	WINNOWER (clique size is 3)	SP-STAR	MERMAID
0.06	0.11	0.02	0.09	0.00	0.88	0.23	0.75

Table 2. Performance of MERMAID for (6,1)-X(m,n)-(6,1)-signal in 20 i.i.d. sequences of length 1000bp

g = 3	g = 5	g = 7	g = 9
0.91	0.88	0.90	0.56

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 as a seed 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. The result is also presented in Table 1.

Table 1 indicates that MERMAID outperforms CONSENSUS, the Gibbs sampler and MEME (with or w/o iterative restart) by a significant scale. Note that the performance coefficients of WINNOWER and SP-STAR reported in (Pevzner and Sze, 2000) 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. We also tested MERMAID on ten real regulons collected by van Helden *et. al.* [6] to verify its usefulness in finding motifs in real-world domains. MERMAID successfully identified all the known motifs in each regulon.

For the extended Challenge Problem, we tested MERMAID on $(6,1)$-$X(m,n)$-$(6,1)$-signals in a set of 20 sequences of length 1000bp, where m and n were varied to form a gap ranging from three to nine bases. The experimental results are presented in Table 2, in which g presents the gap range. It shows that the performance of MERMAID is quite stable till the gap length reaches nine.

In addition to the artificial problem, we also tested MERMAID on several real regulons [13] 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 the brackets presents the rank of the signal found by MERMAID. Converting the matrices found into the IUPAC codes, we compared them with the published motifs, and found they have significant similarity. The known motifs in the regulatory families are all

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

ranked in the top ten. The experimental results indicate MERMAID, which was originally developed to deal with variable gaps, also performs well on real domains where motifs have fixed gaps.

5 Conclusion and Future Work

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 have demonstrated that simply attaching a iterative restart strategy with MEME shows little improvement.

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 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. MERMAID was developed to deal with subtle combinatorial signals. Our experiments showed MERMAID successfully detected combinatorial signals composed of proximal components as well as the known motifs with gaps in many real regulons of yeast genes.

For the future work, we aim to improve MERMAID in two directions. One is efficiency and the other is flexibility. 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. A wider range of gap length produces a

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	CGGn4ATGGAA	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	TTCCA-X(1)-CGGAA [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

larger search space for motif-finding algorithms, and in such cases, it is computationally prohibited to enumerate all possibilities exhaustively. We thus plan to apply a second stochastic sampling process to search through the space of variable gaps, and incorporate domain knowledge when available to constrain the search space.

References

[1] DeRisi, J., Iyer, V. and Brown, P., "Exploring the Metabolic and Genetic Control of Gene Expression on a Genomic Scale", *Science*, Vol 278, (1997) pp. 680-696.

[2] Wodicak, L., Dong, H., Mittmann, M., Ho, M. and Lockhart, D., "Genome-wide Expression Monitoring in Saccharomyces cerevisiae", *Nature Biotechnology*, Vol 15, (1997) pp. 1359-1367.

[3] Bailey, T. and Elkan, C., "Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization", *Machine Learning*, 21, (1995) pp. 51-80.

[4] Hertz, G., Hartzell III, G. and Stormo, G., "Identification of Consensus Patterns in Unaligned DNA Sequences Known to be Functionally Related", *Computer Applications in Biosciences*, Vol 6, No 2, (1990) pp. 81-92.

[5] Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A. and Wootton, J., "Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignments", *Science*, Vol 262, (1993) pp. 208-214.

[6] van Helden, J., Andre, B, and Collado-Vides, J., "Extracting Regulatory Sites from the Upstream Region of Yeast Genes by Computational Analysis of Oligonucleotide Frequencies", *Journal of Molecular Biology*, 281, (1998) pp. 827-842.

[7] Hu, Y., Sandmeyer, S. and Kibler, D., "Detecting Motifs from Sequences", in Proceedings of the 16th International Conference on Machine Learning, (1999) pp. 181-190.

[8] Gelfand, M., Koonin, E. and Mironov, A., "Prediction of Transcription Regulatory Sites in Archaea by a Comparative Genomic Approach", *Nucleic Acids Research*, Vol 28(3), (2000), pp. 695-705.

[9] Li, M., Ma, B. and Wang, L. "Finding Similar Regions in Many Strings", in Proceedings of the 31st ACM Annual Symposium on Theory of Computing, (1999) pp. 473-482.

[10] Pevzner, P. and Sze, S. "Combinatorial Approaches to Finding Subtle Signals in DNA Sequences", in Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, (2000).

[11] Rocke, E. and Tompa, M. "An Algorithm for Finding Novel Gapped Motifs in DNA Sequences", in *RECOMM-98*, (1998) pp. 228-233.

[12] Sinha, S. and Tompa, M. "A Statistical Method for Finding Transcription Binding Sites", in Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology, (2000).

[13] van Helden, J., Rios, A. F. and Collado-Vides, J., "Discovering Regulatory Elements in Non-coding Sequences by Analysis of Spaced Dyads", *Nucleic Acids Research*, Vol 28, (2000) pp. 1808-1818.

[14] Bairoch, A. "PROSITE: a dictionary of sites and patterns in proteins", Nucleic Acids Research, 20, (1992) pp. 2013-2018.

[15] Jonassen, I. "Methods for Finding Motifs in Sets of Related Biosequences", Dept. of Informatics, Univ. of Bergen, Norway, PhD thesis, 1996.

[16] Hu, Y., Sandmeyer, S., McLaughlin, C. and Kibler, D., "Combinatorial Motif Analysis and Hypothesis Generation on A Genomic Scale", *Bioinformatics*, Vol 16, (2000) pp. 222-232.
[17] Stormo, G. "Computer Methods for Analyzing Sequence Recognition of Nucleic Acids", Annual Review of Biophysic and Biophysical Chemistry, 17, (1988) p241-263.
[18] Lawrence, C. and Reilly, A. "An Expectation Maximization (EM) Algorithm for the Identification and Characterization of Common Sites in Unaligned Biopolymer Sequences", Protein: Structure Function and Genetics, 7, (1990) p41-51.