AN EFFICIENT ALGORITHM FOR STRING MOTIF DISCOVERY*

FRANCIS Y.L. CHIN AND HENRY C.M. LEUNG†

Department of Computer Science, The University of Hong Kong, Pokfulam
Hong Kong, China

Finding common patterns, motifs, in a set of DNA sequences is an important problem in bioinformatics. One common representation of motifs is a string with symbols A, C, G, T and N where N stands for the wildcard symbol. In this paper, we introduce a more general motif discovery problem without any weaknesses of the Planted (l,d)-Motif Problem and also a set of control sequences as an additional input. The existing algorithms using brute force approach for solving similar problem take $O(n(t+f)l5^l)$ times where t and f are the number of input sequences and control sequences respectively, n is the length of each sequence and l is the length of the motif. We propose an efficient algorithm, called VAS, which has an expected running time $O(nfl(nt)^k(4^{k-1}+1/4^{k-1})^l)$ using $O((nt)^k(4^{k-1}+1/4^{k-1})^l)$ space for any integer k. In particular when $k = 3$, the time and space complexities are $O(nlf(nt)^3(1.0625)^l)$ and $O((nt)^3(1.0625)^l)$ respectively. This algorithm makes use of voting and graph representation for better time and space complexities. This technique can also be used to improve the performances of some existing algorithms.

1 Introduction

A *genome* is a DNA sequence consisting of four types of nucleotides (symbols): A, C, G and T. During the *gene expression* process, some substrings of the genome, called *genes*, are decoded to produce proteins. In order to start the gene expression process, a molecule called *transcription factor* binds to a *binding site*, represented by a short substring, in the *promoter region* of the gene. Genes seldom work alone. One kind of transcription factor may bind to the binding sites of several genes, allowing the genes to be decoded together. Such binding sites should then have the same length and similar patterns. Finding the common pattern, motif, of the binding sites from a set of sequences representing the promoter regions is an important problem for understanding how gene expression works.

A motif is usually represented by a string [3,4,7,8,12,14-23] or a matrix [1,2,5,6,9-11,13]. When a motif is represented by a $4 \times l$ probability matrix M, where l is the length of the binding sites, the i-th column of M represents the occurrence probabilities of A, C, G and T at the i-th position of a binding site. Although many real biological motifs can be better represented by a matrix, most existing algorithms [1,2,5,6,9,10,13] cannot guarantee finding the optimal matrix-represented motif from a given set of sequences and those algorithms that can may take a prohibitively long time to do so when l is large [11].

When a length-l string P is used to represent the motif, all binding sites are length-l strings similar to P with at most d point substitutions. In other words, the Hamming distance between the motif and each binding site is at most d. Since there are a finite

* The research was supported in parts by the RGC grant HKU 7135/04E
† email: {chin, cmleung2}@cs.hku.hk

number of length-l strings (4^l possible strings), many algorithms [4,8,12,15,17-19,21-23] can guarantee finding the best string motif. The drawback is that some real biological motifs cannot be represented by strings.

Pevzner and Sze [17] define a precise version of motif discovery problem using string representation which has been considered in [4,12,14,15,18,19].

> **Planted (l,d)-Motif Problem:** Suppose there is a fixed but unknown nucleotide sequence P (the motif) of length l. Given t length-n nucleotide sequences and each sequence contains a planted variant of P, we want to determine P without knowing the positions of the planted variants. A variant is a substring derivable from P with at most d point substitutions.

Many algorithms have been developed to solve this problem [4,14,15,18]. However, this problem makes various assumptions and has the following weaknesses.

1. Because of experimental error and noise, some input sequences may not contain any variants of P. On the other hand, some promoter regions may contain more than one binding site [5,10,11,13,20].
2. Although the binding sites of a transcription factor may be different from each other, in real biological data, there are some conserved positions where all binding sites have the same nucleotides. The Planted (l,d)-Motif Problem does not exploit this property, making the defined problem more difficult than the actual problem [11,23].
3. It is difficult for biologists to determine the parameter d without any knowledge about the motif and the binding sites.
4. There are some patterns which are not motifs, but occur frequently in some parts of the genome, inside and outside the promoter regions. Algorithms for solving the Planted (l,d)-Motif Problem may mistakenly find these patterns as motifs [2].

Many algorithms [12,18,20-22] have been studied to solve the motif discovery problem without these weaknesses. Sinha and Tompa [21] modified the planted (l,d)-motif problem to overcome the first three weaknesses. In their model, a sequence may contain zero or more variants of the motif which is represented by a length-l sequence, called *pattern*, consisting of symbols {A, C, G, T, N}. A *variant* of a pattern P is a substring exactly the same as P except that each wildcard symbol N is replaced by A, C, G, or T where d is the number of wildcard symbols N in P. Patterns with different d values are compared by their z-scores (the number of standard deviations by which the number of variants of a pattern in the input sequences exceeds its expected number). Patterns with higher z-scores are more likely to be the correct motifs and the optimal motif is the pattern with the highest z-score.

During experiments, biologists usually get a set of sequences (control set) that do not contain many binding sites of the transcription factors as a by-product [2]. Takusagaw and Gifford [2,22] also considered motifs with wildcard symbols but with a set of control sequences as an additional input to overcome the last weakness. Patterns with relatively more variants in the input sequences than in the control set are likely to be the correct motifs.

There are a number of algorithms [8,20] which can find motifs with wildcard symbols, a similar problem model as given in [21,22]. However, all these algorithms find the motif by brute force. Since there are 5^l possible motifs (patterns), the running times of these algorithms increase exponentially with l.

In this paper, we give the first algorithm which solves the motif discovery problem without any of the above weaknesses, an extension of the motif discovery problem stated in [22]. Our algorithm called VAS (stands for Voting Algorithm from sets of Substrings) uses a new technique based on voting to find the maximum clique of a graph constructed from the set of sequences. Algorithm VAS can effectively find the optimal motif, in a few seconds/minutes instead of hours/days before. This new technique, when applied to some existing algorithms, should be able to greatly improve their performances too. In [17], a graph is constructed for solving the Planted (l,d)-Motif Problem. Each length-l substring in the input sequence is represented by a vertex, and an edge between two vertices exists if the two corresponding substrings differ by no more than $2d$ point substitutions. The Planted (l,d)-Motif Problem can be reduced to the finding of the maximum clique of the constructed graph, which takes $O((nt)^{(t-1)+2.376})$ time when d is large and $O((nt)^2)$ space. In [4], a voting algorithm was introduced for finding the motif. This algorithm, through some heuristics, can solve the Planted (l,d)-Motif Problem for large l and d with a high probability. However, its time and space complexity are $O(nt(3l)^d)$ and ($O(n(3l)^d)$ if we want to find the motifs with 100% certainty.

Algorithm VAS, based on voting from k similar substrings (when k = 2, the two similar substrings form an edge), has the merits of voting and graph representation and has better expected time and space complexities. For example, when k = 2, the expected time complexity and space complexity of VAS are $O(nlf(nt)^2(1.25)^l)$ and $O((nt)^2(1.25)^l)$ respectively; when k = 3, VAS takes $O(nlf(nt)^3(1.0625)^l)$ time and $O((nt)^3(1.0625)^l)$ space, where n is the length of the input sequences, t is the number of the input sequences, f is the number of the sequences in the control set and k can be any positive integer. Experimental results show that VAS has good performances on both simulated data and real biological data.

This paper is organized as follows. In Section 2, we define the extended motif discovery problem with control set. In Section 3, we describe VAS for solving this extended motif discovery problem. Experimental results on both simulated data and real biological data are shown in Section 4 followed by a conclusion in Section 5.

2 The Motif Discovery Problem

In order to address the weaknesses of the Planted (l,d)-Motif Problem, we define the motif discovery problem without any of these weaknesses as long as *P has relatively more variants in the set of input sequence T than in the set of control sequences F*. Before proceeding further, we have to formally define the meaning of "*P* has relatively more variants in *T* than in *F* " in the above problem definition.

Let t and f be the number of length-n sequences in T and F respectively. Barash et al. [2] determined whether a pattern P is the motif by considering the "random selection null hypothesis" that sequences in T are randomly selected from all the $t + f$ sequences. P is the motif if this hypothesis is false. However, they assumed that each sequence contains only one binding site and did not consider sequences with zero or multiple binding sites. Similar approach has also been used in [22]. In this paper, we verify whether P has relatively more variants in T than in F using a similar hypothesis as Barash et al. with the extra assumption that each sequence may contain zero or more variants of the motif.

Similar to [3,10], we break down the sequences in T and F into $\alpha = t(n - l + 1)$ and $\beta = f(n - l + 1)$ length-l substrings. Assume that T and F contain k_t and k_f variants of a motif with pattern P, consider the null hypothesis that the sequences in T are constructed by combining α substrings randomly selected from the $\alpha + \beta$ substrings without replacement (we may not be able to combine α substrings to construct t length-n sequences). Given a pattern P with $k_t + k_f$ variants in set T and F respectively, the probability that k_t of them are in set T is

$$P_{hyper}(k_t \mid k_t + k_f, \alpha, \beta) = \frac{\binom{\alpha}{k_t}\binom{\beta}{k_f}}{\binom{\alpha+\beta}{k_t+k_f}}$$

followed from the hyper-geometric probability distribution. The p-value that the null hypothesis is true can be calculated by summing up the tail of the probability distribution for $k_t' > k_t$.

$$p-value = P(k_t, k_f, \alpha, \beta) = \sum_{k_t' = k_t}^{\min\{M, k_t + k_f\}} P_{hyper}(k_t' \mid k_t + k_f, \alpha, \beta)$$

A pattern P with a small p-value means the null hypothesis is unlikely, i.e. P is likely to be the motif. Based on what we have discussed, we give the formal definition of the extended motif discovery problem without any of the above weaknesses as follows:

> **Extended Motif Problem with Control Set:** Suppose there is a fixed but unknown pattern P (the motif) of length l with symbols A, C, G, T and N. Given k_t variants of P in the t length-n nucleotide sequences in T and k_f variants of P in the f length-n nucleotide sequences in F, where $k_t/t >> k_f/f$ in the sense that P has a small p-value,we want to determine P with knowledge of the motif length l only.

In practice, there might be a few patterns with small p-values. Our algorithm will find the optimal motif which is the pattern with the smallest p-values. Note that the input of d is not necessary in the above problem definition because the correct pattern P should include the knowledge of d. Our algorithm will exhaust all values of d to find the pattern P with the smallest p-value.

3 Algorithm

Since there are 5^l possible length-l patterns and checking which pattern has the smallest p-value by brute force takes $O(5^l nl(t + f))$ time which can be extremely long for large l. Existing algorithms for solving the planted motif problem, like WINNOVER [17], PROJECTION [3] and SPELLER [19], cannot be extended to solve the extended motif discovery problem easily because they either do not guarantee finding the motifs or need a long running time when d is large.

Algorithm 1: VAS when $k = 1$.

```
1   Create a hash table V with zero at each entry          {V stores the number votes for each pattern}
2   min_p ← 1                                              {min_p is the minimum p-value}
3   For each length-l substring S in T
4       For d ← 0 to l
5           For each pattern P with exactly d symbol Ns such that S is a variant of P
6               V(H(P)) ←V(H(P)) + 1                       {H(P) is the hash value of P}
7   sort the patterns in V in non-increasing order of the number of votes V(H(P))
8   For each pattern P
9       k_t ←V(H(P))
10      If P(k_t, 0, α, β) < min_p
11          count the number of variants k_f of P in F
12          If P(k_t, k_f, α, β) < min_p
13              min_p ←P(k_t, k_f, α, β)
14              motif ← P
15      Else output motif
```

We might apply the basic idea of the Voting algorithm [4] to solve the extended motif discovery problem (Algorithm 1). For each length-l substring S in the input sequence, one vote is given to patterns P such that S is a variant of P. Note that all patterns with different d values, $0 \le d \le l$, have been considered in the algorithm. Since there are $t(n - l + 1)$ length-l substrings in T, and a length-l substring can be a variant of $\binom{l}{d}$ possible patterns with exactly d symbol Ns, the time needed for the algorithm is

$$O\left(n(t-l+1)l\sum_{d=0}^{l}\binom{l}{d}\right) = O(ntl2^l)$$

Since exactly $t(n - l + 1)2^l$ votes will be issued, there will be at most $n(t - l + 1)2^l$ entries in the hash table V. The time needed to count the number of variants of a pattern in F is nlf. Therefore the time needed for verifying each entry in V is at most $nlft(n - l + 1)2^l = O(nlf(nt)2^l)$. The total running time of the algorithm is $O(ntl2^l + nlf(nt)2^l) = O(nlf(nt)2^l)$. The memory needed for storing the hash table V is $O(nt2^l)$.

Although the base number of the exponent is reduced from 5 to 2, the time and space complexities of this direct voting algorithm are still very large. The space complexity remains impractical for large l.

The planted motif problem can also be viewed as a maximum clique problem [17]. Even though the maximal clique problem is NP-complete, this approach has the advantage that the space complexity is at most $O((nt)^2)$ as there are $t(n - l + 1)$ substrings (vertices) in T. In order to reduce the time and space complexities, we modify the Voting algorithm to vote patterns by a set of substrings instead of a single substring.

Normally the hidden motif should have at least two variants in T. One vote will be given to those patterns P such that length-l substrings S and S' in T are variants of P. Thus, a pattern with k variants in T should get exactly $\binom{k}{2}$ votes. Intuitively, the time and space complexities can be reduced because the hash table V does not needed to handle those patterns with only one variant in T. The expected time complexity and space complexity can be calculated as follows.

Assume the occurrence probabilities of A, C, G and T are 0.25. The probability that S differs from S' in i positions is $\binom{l}{i} 0.25^{l-i} 0.75^{i}$ and the number of patterns P such that both S and S' are variants of P is

$$\sum_{d=i}^{l} \binom{l-i}{d-i} = \sum_{j=0}^{l-i} \binom{l-i}{j} = 2^{l-i}$$

So the expected number of patterns voted by each pair of substrings is

$$\sum_{i=0}^{l} \binom{l}{i} \left(\frac{1}{4}\right)^{l-i} \left(\frac{3}{4}\right)^{i} 2^{l-i} = \sum_{i=0}^{l} \binom{l}{i} \left(\frac{2}{4}\right)^{l-i} \left(\frac{3}{4}\right)^{i} = \left(\frac{5}{4}\right)^{l}$$

Since there are $O((nt)^2)$ pairs of substrings in T, the time complexity of the algorithm (including checking the patterns in F) is

$$O\left(l(nt)^2 \left(\frac{5}{4}\right)^{l} + nlf\,(nt)^2 \left(\frac{5}{4}\right)^{l} \right) = O\left(nlf\,(nt)^2 \left(\frac{5}{4}\right)^{l} \right)$$

And the space complexity of the algorithm is

$$O\left((nt)^2 \left(\frac{5}{4}\right)^{l} \right)$$

In order not to miss motifs with only one variant in T, we check whether each length-l substring in T can be the motif. This checking step takes $O(nlf(nt))$ time and $O(1)$ space which do not affect the time and space complexities of VAS.

With this approach, the time and the space complexities are reduced by a factor of

$$nt2^{l} / (nt)^2 \left(\frac{5}{4}\right)^{l} = \frac{1}{nt} \left(\frac{8}{5}\right)^{l}$$

if we vote on patterns by pairs of substrings instead of single substrings. The algorithm can be speeded up if and only if $nt < (8/5)^l$. Therefore, voting by pairs of substrings is

beneficial when the size of the input sequences is small or the length of the motif is long. Similar improvement can be performed by giving votes to patterns of k substrings. The expected time complexity and space complexity for voting from k substrings are $O(nlf(nt)^k(4^{k-1}+1/4^{k-1})^l)$ and $O((nt)^k(4^{k-1}+1/4^{k-1})^l)$ respectively. In practice, VAS has the best time complexity when $k = 2$ or 3 depending on the size of the input sequences and the length of the motif.

4 Experimental Results

We have implemented VAS in C++ and used it to find motifs in both simulated and real biological data. In this section, we describe the performance of VAS and compare it with some existing motif discovery algorithms. All experiments were taken on a 2.4GHz P4 CPU with 1 GB memory.

Table 1. Successful rate and running time of VAS.

l	d	$b = 10$		$b = 20$		$b = 30$		$b = 40$	
		Success rate	Time	Success rate	Time	Success rate	Time	Success rate	Time
8	1	100%	13.7s	100%	13.7s	100%	13.8s	100%	13.8s
	2	74%	13.8s	76%	13.7s	100%	13.7s	100%	13.8s
10	3	100%	17.1s	100%	17.0s	100%	16.7s	100%	16.8s
	4	62%	17.1s	54%	16.7s	100%	16.9s	100%	16.7s
12	5	100%	23.1s	100%	23.3s	100%	23.3s	100%	23.4s
	6	44%	23.0s	84%	23.1s	100%	23.2s	100%	23.3s
14	7	100%	37.8s	100%	37.6s	100%	37.7s	100%	37.6s
	8	68%	37.7s	96%	37.7s	100%	37.7s	100%	37.7s
16	9	100%	67.1s	100%	67.1s	100%	67.1s	100%	67.1s
	10	100%	67.0s	82%	67.2s	100%	67.2s	100%	67.0s
18	11	100%	132s	100%	132s	100%	131s	100%	132s
	12	100%	132s	100%	132s	100%	131s	100%	132s
20	13	100%	256s	100%	256s	100%	255s	100%	256s
	14	100%	255s	100%	256s	100%	256s	100%	256s

4.1. *Simulated Data*

The simulated data were generated as follows. All input instances contain $t = 20$ length-600 sequences in T and $f = 20$ length-600 sequences in F. Each nucleotide of these sequences was generated independently with the same occurrence probability 0.25. Then a length-l motif M with d Ns was picked randomly from all possible patterns and b variants of M were planted in the sequences in T at random positions. The motif length l and the sequences in T and F were inputted to VAS for finding the motifs. For each set of parameter l, d and b, we ran 50 test cases. Table 1 shows the successful rate and the average running time of VAS when $k = 2$ (votes are given by pairs of substrings in T).

Since the number of votes given by each pair of substrings in T is almost independent of the number of planted variants in T and the number of Ns in the motif pattern, the running time of VAS is independent of these factors as shown in Table 1. Algorithm VAS may not find the motif when d, the number of Ns in the motifs, is relatively large (e.g. (8,2), (10,4), (12,6)) and the number of planted variants in T is small ($b = 10$ or $b = 20$).

It is because random patterns *P* might have more variants in *T* and less variants in *F* than the motif *M* in these cases. Since VAS cannot distinguish *M* from these random patterns *P*, VAS fails to find the motif. However, when the number of non-N symbols in *M* is reasonably large (> 6), VAS can find the motif *M* successfully with high probability.

Common motif discovery algorithms like PROJECTION [3] and VOTING [4] are developed for solving planted motif problem without control set. In order to compare the performance of these algorithms with VAS, we reduce the values of *d* for these algorithms such that they can theoretically find the motif [3] and plant exactly one variant in each sequence in *T*. Table 2 shows the results of these algorithms.

Table 2. Successful rate and running time of brute force algorithm, PROJECTION, VOTING. And VAS

l	*d*	Brute Force		PROJECTION		VOTING		VAS	
		Success rate	Time	Success rate	Time	Success rate	Time	Success rate	Time
8	1	100%	268s	94%	18s	100%	<1s	100%	13.8s
10	2	100%	72min	98%	77s	100%	0.7s	100%	16.7s
12	3	-	-	88%	371s	100%	28.4s	100%	23.4s
14	4	-	-	76%	650s	100%	412s	100%	37.7s
16	5	-	-	82%	20min	100%	31min	100%	67.0s
18	6	-	-	88%	34min	-	-	100%	132s
20	7	-	-	86%	48min	-	-	100%	256s

Although brute force algorithms can find motif when *l* is small, they fail to find the motif when $l > 10$. Voting algorithm [4] (we use the basic voting algorithm without heuristic search) has a better performance than the brute force algorithms because its running time increases exponentially with *d* instead of *l*. The running time of PROJECTION does not increase sharply with *l* because it performs heuristic search for finding the motifs. However, it does not guarantee that the motifs can be found all the time and has a success rate less than 100%. When compared with these algorithms, VAS has the best performance in both accuracy and running time.

4.2. *Real Biological Data*

SCPD [24] contains different transcription factors for yeast. For each set of genes regulated by the same transcription factor, we chose the 600 base pairs in the upstream of the genes as the input sequences *T*. 100 random sequences in the upstream of yeast's genes were picked randomly as the set of control sequences *F*. The lengths of the motifs were same as those of the published motifs. For PROJECTION and the Voting algorithm, we tested all possible *d* from 0 to *l*. Experimental results are showed in Table 3. Some motifs with many wildcard symbols (e.g. GAL4) cannot be represented properly by the planted motif problem and can be found by VAS only. Since PROJECTION and the voting algorithm do not consider the set of control sequences, they fail to find motifs when relatively less variants are in *T* (e.g. ACE2). On the other hand, VAS can find the motifs in these cases with the help of the control sequences. Note that we have not shown all the experimental results because PROJECTION, the Voting algorithm and VAS have the same performance on the rest transcription factors in the SCPD.

Table 3. Experimental results on real biological data

Name	Published Pattern	PROJECTION	VOTING	VAS
CuRE	TTTGCTC	TTTGCTC	TTTGCTC	TTTGCTC
GATA	CTTATC	CTTATC	CTTATC	TTATCG
ACE2	GCTGGT	-	-	GCTGGT
AP1	TTANTAA	-	TTACTAA	TTANTAA
GAL4	$CGGN_{11}CCG$	-	-	CGGNGNNCTNTNGNCCG
ROX	YYNATTGTTY	-	-	TCCATTGTTC

Symbol Y means C or T. N_{11} means 11 Ns.

5 Discussion

In this paper, we have introduced VAS for solving the extended motif discovery problem with control set using $O(nlf(nt)^k(4^{k-1}+1/4^{k-1})^l)$ time and $O((nt)^k(4^{k-1}+1/4^{k-1})^l)$ space for any positive integer k. Not only VAS can solve the motif discovery problem with least assumptions, experimental results show that VAS has the best performance than existing algorithms in both speed and accuracy. Since VAS can find the number of variants of every length-l patterns in T in short running time, the new technique used in VAS can also be applied to find string motifs for other motif discovery algorithms for those problems without control set F [12] or based on other hypotheses [20]. For example, if the input does not contain any control sequences, we cannot use the hyper-geometric distribution for the evaluation of p-values by. In this case, we may have to evaluate the p-values based on the background occurrence probabilities of the nucleotides. The extension of this work will have similar performance as VAS and will be included in the full paper.

VAS works well on the extended motif discovery problem because it is easy to find the set of patterns to be voted by a substring in T. This task may become difficult when the definition of variants is changed. In the future, we will investigate how to use VAS to solve motif discovery problems with other definitions of variants, for example, motif with IUPAC symbols.

6 References

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

2 Y. Barash, G. Bejerano and N. Friedman. A simple hyper-geometric approach for discovering putative transcription factor binding sites. *WABI*, p278-293, 2001.

3 J. Buhler and M. Tompa. Finding motifs using random projections. *RECOMB*, p69-76, 2001.

4 F. Chin and H. Leung. Voting Algorithms for Discovering Long Motifs. *APBC*, p261-271, 2005.

5 F. Chin, H. Leung, S.M. Yiu, T.W. Lam, R. Rosenfeld, W.W. Tsang, D. Smith and Y. Jiang. Finding Motifs for Insufficient Number of Sequences with Strong Binding to Transcription Factor. *RECOMB*, p125-132, 2004

6 G. Z. Hertz and G. D. Stormo. 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, 1995

7 U. Keich and P. Pevzner. Finding motifs in the twilight zone. *RECOMB*, p195-204, 2002

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

9 C. Lawrence, S. Altschul, M. Boguski, J. Liu, A. Neuwald and J. Wootton. Detecting subtule sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262:208-214, 1993

10 C. Lawrence and A. Reilly. An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins: Structure, Function and Genetics*, 7:41-51, 1990

11 H. Leung and F. Chin. Finding Exact Optimal Motif in Matrix Representation by Partitioning. *Bioinformatics*, 21(supp 2):ii86-ii92, 2005

12 H. Leung and F. Chin. Generalized Planted (*l*,*d*)-Motif Problem with Negative Set**.** *WABI*, p264-275, 2005

13 H. Leung, F. Chin, S.M. Yiu, R. Rosenfeld and W.W. Tsang. Finding Motifs with Insufficient Number of Strong Binding Sites. *Jour. Comp. Biol.*, 2005 (will appear)

14 M. Li, B. Ma, and L. Wang. Finding similar regions in many strings. *Journal of Computer and System Sciences*, 65:73-96,2002

15 S. Liang. cWINNOWER Algorithm for Finding Fuzzy DNA Motifs. *Computer Society Bioinformatics Conference*, p260-265, 2003

16 G. Pesole, N. Prunella, S. Liuni, M. Attimonelli, and C. Saccone. Wordup: an efficient algorithm for discovering statistically significant patterns in dna sequences. *Nucl. Acids Res.*, 20(11):2871-2875,1992.

17 P. Pevzner and S.H. Sze. Combinatorial approaches to finding subtle signals in dna sequences. *In Proc. of the Eighth International Conference on Intelligent Systems for Molecular Biology*, p269-278, 2000.

18 S. Rajasekaran, S. Balla and C.H. Huang. Exact algorithms for planted motif challenge problem. *APBC*, p249-259, 2005.

19 M.F. Sagot. Spelling approximate repeated or common motifs using a suffix tree. *In C.L. Lucchesi and A.V. Moura editors, Latin'98: Theoretical Informatics, volume 1380 of Lecture Notes in Computer Science*, p111-127, 1998.

20 S. Sinha. Discriminative motifs. *In Proc. of the Sixth Annual International Conference on Computational Biology*, p291-298, 2002

21 S. Sinha and M. Tompa. A statistical method for finding transcription factor binding sites. *In Proc. of the Eighth International Conference on Intelligent Systems for Molecular Biology*, p344-354, 2000.

22 K.T. Takusagawa and D.K. Gifford. Negative information for motif discovery. *PSB*, p360-371, 2004

23 M. Tompa. An exact method for finding short motifs in sequences with application to the ribosome binding site problem. *In Proc. of the 7th International Conference on Intelligent Systems for Molecular Biology*, p262-271, 1999.

24 J. Zhu and M. Zhang. SCPD: a promoter database of the yeast Saccharomyces cerevisiae. *Bioinformatics* 15:563-577, 1999. http://cgsigma.cshl.org/jian/