

On the Challenging Instances of the Planted Motif Problem*

Sudha Balla, Jaime Davila and Sanguthevar Rajasekaran

Dept. of Computer Sci. & Eng., University of Connecticut, Storrs CT 06269, USA

Abstract. A classic problem of motif discovery in DNA sequences, called the Planted (l, d) -Motif Problem has been widely studied over the past decade owing to its application in identifying vital signals such as transcription factor binding sites. Challenging instances of the problem are those that have been probabilistically proved as ‘difficult to be solved’ due to the existence of several motifs by random chance for such instances. In this work, we present an expected case analysis that helps understanding the real ‘challenging instances’ of the problem.

1 Introduction

The Planted (l, d) -Motif Problem (PMP) is one of the many variants of the problem of discovering *motifs* or patterns of biological significance in sequence data, described as follows:

Definition 1. Given n sequences $S = \{S_1, S_2, S_3, \dots, S_n\}$, $|S_i| = m, 1 \leq i \leq n$, from a fixed alphabet Σ , and also integers l and d , the goal is to identify a motif M of length l that has at least one occurrence M_i in $S_i, 1 \leq i \leq n$, such that the hamming distance $d(M, M_i) \leq d, 1 \leq i \leq n$.

PMP was formally described in [20]. In this paper a very specific instance PMP, calling where $n = 20, m = 600, l = 15, d = 4$ and $\Sigma = \{A, C, G, T\}$, was identified as *challenging*. Soon after its definition, several algorithms that adopt core techniques in computer science such as sorting, hashing, traversals of trees and graphs have been proposed for PMP. Algorithm WINNOWER attempts to identify large cliques in a multipartite graph G , constructed with the patterns of length l in the input sequences as its vertices. Two vertices u and v in G are connected by an edge iff u and v belong to two different sequences of the input and their hamming distance (i.e.) the number of substitutions needed to convert u to v and vice versa, $d(u, v) \leq 2d$. Algorithm WINNOWER treats all edges of G equally and does not distinguish edges that correspond to high and low similarities. Algorithm SP-STAR attempts to overcome this drawback by using a sum-of-pairs scoring function and a local improvement strategy to identify the best occurrences of the motif in the input set.

Buhler and Tompa [7] stated that there are instances of PMP that are more challenging than the $(15, 4)$ challenge problem and devised an algorithm called

* This research has been supported in part by the NSF Grant ITR-0326155.

PROJECTION to solve such instances. They concluded that WINNOWER and SP-STAR failed to solve the (14, 4), (16, 5) and (18, 6)-motif problems for the same values of n and m as above, while their algorithm PROJECTION succeeded in doing so. Algorithm PROJECTION uses the principle of random projections to arrive at better seeds of the input for an EM algorithm. It uses a hash function $h(x)$ constructed using k of the l positions chosen at random, and hashes all substrings of length l of the input sequences into buckets based on their value w.r.t. the k positions. It is based on an intuition that if $k < (l - d)$ a number of the n variants of M would hash into the same bucket. A probability weight matrix arrived from the substrings hashed on to highly enriched buckets is used as the initial seed to the EM algorithm. The work presented a probabilistic analysis of PMP to arrive at the difficult instances of PMP, such as the (9, 2), (11, 3), (13, 4), (15, 5), (17, 6)-motif problems and stated that these problems are inherently unsolvable by PROJECTION as the number of spurious motifs (patterns that appear by random chance in the input) for these instances is more than one (Table 2 of [7]). Algorithms MULTI-PROFILER [16], PatternBranching and ProfileBranching [21] also address PMP and were shown to perform well in practice for several instances on the problem on random and real biological data.

But there are earlier algorithms that have been proposed in the literature to identify motifs in a set of DNA sequences that could be binding sites for regulatory elements. Lawrence and Reilly [18] proposed an algorithm based on Expectation Maximization (EM) to identify such motifs. Bailey and Elkan's [1] contribution, algorithm MEME, was an extension of Lawrence and Reilly's work to discover multiple occurrences of a motif in a set of sequences and also to discover multiple planted motifs for a given input. Lawrence et al. [17] presented an algorithm based on Gibbs Sampling, called the GibbsDNA. Hertz and Stormo [15] devised a greedy algorithm CONSENSUS to identify functional relationships by aligning DNA, RNA or protein sequences. They used a log-likelihood scoring scheme to arrive at the information content of an alignment and the algorithm picked those alignments with highest information content. CONSENSUS successfully identified 19 of 24 sites of the DNA binding protein CRP-transcription factor in 18 DNA sequences of E-coli, each about 105 nt in length.

All the algorithms discussed above employ local search techniques and may not output the desired planted motif always. We refer to such algorithms as *approximate algorithms*. The performance of such approximate algorithms is measured using a factor called the *performance coefficient* in the literature. Let K be the number of actual residue positions (nl) of the input that correspond to the variants of motif M . Let P be the number of such residue positions predicted by an algorithm. *Performance Coefficient* (ρ) is defined as the ratio $(K \cap P)/(K \cup P)$. Algorithms that always output the correct answer are referred to as *exact algorithms*. While for approximate algorithms $0 < \rho < 1$, for exact algorithms $\rho = 1$. Table 1 gives the performance of several algorithms discussed above on the (15,4) instance of PMP.

There are several exact algorithms in the literature proposed for PMP in [4], [6], [12], [24], [25], [26], [27], [28]. Such algorithms are *exhaustive enumeration*

Table 1. Performance of Approximate Algorithms on (15, 4) instance of PMP

Algorithm	Year	ρ
GibbsDNA	1993	0.12
MEME	1995	0.10
CONSENSUS	1999	0.07
WINNOWER	2000	0.92
PROJECTION	2001	0.93
PatternBranching & ProfileBranching	2003	≈ 1.00 & 0.57

algorithms and as aptly stated in [7], they become impractical for the challenging instances of PMP. A salient exact algorithm called MITRA was proposed by Eskin and Pevzner [10] that adopts a mismatch tree data structure to represent the pattern space and performs a depth first search on the mismatch tree to identify the planted motif for a given input. MITRA was shown to be successful in identifying monads (simple planted motifs) and dyads (complex planted motifs that appear in pairs separated by a varying gap length in each input sequence) in synthetic and real biological data. The voting algorithm [8] adopts hashing techniques to identify planted motifs.

Exact algorithms that adopt sorting techniques were first presented in [23]. These algorithms solve the challenge instances (9, 2), (11, 3) and (13, 4) in time 1.43 seconds, 19.84 seconds and 228.94 seconds respectively, that were deemed inherently difficult to be solved computational methods in [7]. Algorithms PMSi and PMSP that adopt better pruning techniques while searching the motif space have been proposed in [9] and have solved the (15, 5) and (17, 6) instances of PMP in 35 minutes and 12 hours respectively. For a survey of algorithms for motif search see [22]. See also [2], [3].

There have also been contributions to PMP by researchers who have addressed related problems in [5], [4] (Substring Parsimony Problem), [13] (Closest String Problem), [11] (Common Approximate Substring Problem) and [19] (Consensus Patterns Problem).

In this work, we present an expected case analysis of PMP that arrives at the number of input sequences that need to be examined by an algorithm to distinguish strong motif candidates from spurious occurrences. Our analysis suggests a relook at what could be deemed as difficult instances of the problem.

2 Expected Case analysis of PMP

Consider any two occurrences of the motif M , say M_i and M_j occurring in sequences S_i and S_j respectively, $1 \leq i, j \leq n, i \neq j$. Clearly, $d(M_i, M_j) \leq 2d$. Let $Q = \{q_1, q_2, q_3, \dots, q_n\}$ be a set of substrings of the input such that $q_i \in S_i, |q_i| = l, d(q_i, q_j) \leq 2d, 1 \leq i, j \leq n$. Several algorithms proposed for PMP attempt to identify Q in order to discover M . The WINNOWER algorithm [20] identifies elements of Q by finding cliques in a graph. The algorithm of Gramm et al. [14] is based on a similar technique where the authors identify M as the

closest string of the elements of Q . We present an expected case analysis of this approach in order to arrive at the number sequences that need to be analyzed by an algorithm to distinguish strong planted motif candidates from spurious signals.

Let u and v be two random strings of length l . Let p be the probability that $d(u, v) \leq 2d$. Then, we have,

$$p = \sum_{k=0}^{2d} {}^l C_k (1/4)^{(l-k)} (3/4)^k.$$

Let u be an l -mer (substring of length l) in S_1 . Let $V_i = \{v_1, v_2, v_3, \dots, v_{|V_i|}\}$ be the set of l -mers in S_i , $2 \leq i \leq n$ such that for every $v \in V_i$, $d(u, v) \leq 2d$. The expected size of V_i , $2 \leq i \leq n$ is $O(pm)$ as there are $(m - l + 1)$ l -mers in each sequence.

Let T be a tree rooted at u with the following properties. The nodes at level i are l -mers that belong to S_i , $1 \leq i \leq n$. An internal node v in T is at a hamming distance of at most $2d$ from all its ancestors. The expected number of children of u in level 2 is $O(pm)$. Similarly, for every node in level 2, the expected number of children in level 3 is $O(p^2m)$. In general, the expected number of children at level i for a node in level $(i - 1)$ is $O(p^{(i-1)}m)$. Therefore, we can calculate the expected height h of T by solving for i in,

$$p^{(i-1)}m = 1 \Rightarrow i = 1 + \log_{(1/p)}m$$

Table 2 gives the p and the h values for several instances of PMP. The $E[l, d]$ values are based on the probabilistic analysis of [7], a value greater than one indicating those instances inherently unsolvable by algorithm PROJECTION.

Table 2. Challenging instances of PMP

l	d	p	h	$E[l, d]$
9	2	0.05	3.13	1.60
12	3	0.05	3.13	$3.19 * 10^{-7}$
15	4	0.05	3.13	$2.17 * 10^{-15}$
11	3	0.11	3.90	4.72
14	4	0.11	3.90	$4.20 * 10^{-7}$
17	5	0.11	3.90	$2 * 10^{-15}$
13	4	0.21	5.10	5.23
16	5	0.19	4.85	$2.33 * 10^{-7}$
19	6	0.17	4.61	$9.11 * 10^{-16}$
15	5	0.31	6.46	2.84
18	6	0.28	6.02	$7.11 * 10^{-8}$
21	7	0.26	5.75	$2.51 * 10^{-16}$
17	6	0.42	8.37	0.88
19	7	0.53	11.07	0.18
22	8	0.48	9.71	$2.02 * 10^{-9}$

We can calculate the expected number of nodes in T as,
 $= 1 \times O(pm) \times O(p^2m) \times \dots \times O(p^{(h-1)}m)$
 $= O((p^{(1+2+3+\dots+(h-1))})m^{(h-1)})$

$$= O(p^{(h(h-1)/2)} m^{(h-1)})$$

There are $O(m)$ such trees to be constructed and searched for finding all patterns of length l that qualify to be M for the input. The time required at every node of T is $O(hml)$. Thus, the expected runtime of the algorithm is $O(hm^{(1+(h/2))}l)$.

3 Discussion on the Challenging Instances

In this section we discuss the effect the analysis discussed above bears on the performance of several existing algorithms for PMP. From table 2 we see that the number of sequences (h) required by any algorithm to distinguish strong motif occurrences from those that are spurious increases with the increase in p value. For example, it is suffice to examine only three sequences of the input to identify strong signals for the (9, 2), (12, 3) and (15, 4) instances, while for the (19, 7) instance the number increases to 11. Several instances of the problem that have been deemed difficult in [7] share the same p value as those that are not. For example the (11, 3) challenging instance with $E[l, d] = 4.7$ has a p value of 0.11, the same as that of the (14, 4) and the (17, 5) instances with $E[l, d] = 4.2 * 10^{-7}$ and $2 * 10^{-15}$ respectively. Exact algorithms of [23], [8] and [9] have been able to identify all the motifs that qualify to be M for the given input for the (9, 2), (11, 3), (13, 4), (15, 5), (17, 6) instances where $E[l, d] > 1$. Therefore, categorizing instances with $E[l, d] > 1$ as ‘challenging’ pertains only to approaches that essentially converge towards only one M for a given input. Otherwise, as long as the $E[l, d] \leq c$, $c > 0$ being a small number such that the list of motifs output by an algorithm could be validated through biological experiments, the known exact approaches would be successful in identifying the motifs. But, instances with high h values are those that become ‘challenging’ due to impractical runtime for such algorithms.

The h values of table 2 also suggest the following. Algorithms of [23] and [8] generate the neighborhood of every l -mer in the input set to identify those common patterns that have occurred in the neighborhood of at least one l -mer in every input sequence. From the analysis, we can see that it is sufficient to generate a neighborhood of l -mers from h sequences of the input to identify such a set, say C , of common patterns. Subsequently, each element $M' \in C$ could be checked to see if $M' = M$. This would improve the practical performance of the above algorithms by a factor of the time spent in generating and processing the neighborhood of the $(n - h)$ sequences.

Algorithm WINNOWER of [20] eliminates spurious edges in a multipartite graph G of n parts, each consisting of m vertices, constructed from the l -mers of the input, based on the concept of *extendable cliques*. Initially, cliques of size $k = 2$ or $k = 3$ in G are considered. The above analysis suggests that for a given instance, the algorithm will not be able to eliminate several spurious edges in G until cliques of size $k = h$ are built. The success of WINNOWER in identifying (15, 4) instances is evident from the fact that $h = 3.13$ for this instance and hence several edges of G will be eliminated in the very early stages of the algorithm.

Also, constructing all the nC_k cliques by the algorithm need not be necessary to solve PMP. Instead, considering only cliques of size k with nodes from the first k parts of G is sufficient, which would achieve a significant speedup on the practical runtime of the algorithm.

In its basic form, algorithm PMSP of [9] identifies planted motifs as follows. Let u be an l -mer of S_1 . Let $B_i, 2 \leq i \leq n$, denote the set of l -mers of S_i such that for every element $v \in B_i, d(u, v) \leq 2d$. PMSP generates each neighbor x in the d -neighborhood (elements in the neighborhood are at a hamming distance of at most d) of every l -mer u of S_1 , and checks if $x = M$ by looking if there exists a $x' \in B_i, 2 \leq i \leq n$ such that, $d(x, x') \leq d$. An expected case runtime of $O(pnm^2l^d|\Sigma|^d(l/w))$, where w is the word length of the computer, is presented by the authors based on $|B_i|, 2 \leq i \leq n$. If we relook at the strategy of PMSP based on the analysis presented above, to check if any $x = M$, the number of B_i s the algorithm would scan in the expected case is h of table 2. Therefore, the expected case runtime of PMSP could be arrived as $O(phm^2l^d|\Sigma|^d(l/w) + zpnm)$, where z actual number of planted motifs for the given input.

4 Conclusion

In this paper, we presented an expected case analysis of an approach to discover planted (l, d) -motifs in a set of DNA sequences. The analysis helped in understanding factors that would deem certain instances of the problem as challenging instances to the algorithm that is employed to discover the motifs. Based on the analysis, we could also suggest strategies to improve the practical performance of several existing algorithms in the literature.

References

1. Bailey T. L., Elkan, C.: Fitting a mixture model by expectation maximization to discover motifs in biopolymers. Proc. Second International Conference on Intelligent Systems for Molecular Biology (1994) 28–36
2. Balla, S., Davila, J., Rajasekaran, S.: Approximation Algorithms for the Primer Selection, Planted Motif Search, and Related Problems. In T.E. Gonzalez, editor, Approximation Algorithms and Metaheuristics CRC Press (2006) 75–1.
3. Balla, S., and Rajasekaran, S., Sorting and FFT Based Techniques in the Discovery of Biopatterns, to appear in *Bioinformatics Algorithms: Techniques and Applications*, 2007.
4. Blanchette, M.: Algorithms for phylogenetic footprinting. Proc. Fifth Annual International Conference on Computational Molecular Biology (2001).
5. Blanchette, M., Schwikowski, B., Tompa, M.: An exact algorithm to identify motifs in orthologous sequences from multiple species. Proc. Eighth International Conference on Intelligent Systems for Molecular Biology (2000) 37–45
6. Brazma, A., Jonassen, I., Vilo, J., Ukkonen, E.: Predicting gene regulatory elements in silico on a genomic scale. Genome Research **15** (1998) 1202–1215.
7. Buhler, J. and Tompa, M.: Finding motifs using random projections. Proc. Fifth Annual International Conference on Computational Molecular Biology (RECOMB) (2001)

8. Chin, F. Y. L., Leung, H. C. M.: Voting Algorithms for Discovering Long Motifs. Proc. Third Asia Pacific Bioinformatics Conference (APBC) (2005) 261–271.
9. Davila, J., Balla, S., Rajasekaran, S.: Space and Time Efficient Algorithms For Planted Motif Search. Proc. 6th International Conference on Computational Science (ICCS 2006)/ 2nd International Workshop on Bioinformatics Research and Applications (IWBRA 2006) LNCS **3992** (2006) 822–829
10. Eskin, E., Pevzner, P. A.: Finding composite regulatory patterns in DNA sequences. *Bioinformatics* **S1** (2002) 354–363
11. Evans, P. A., Smith A. D., Wareham H. T.: On the complexity of finding common approximate substrings. *Theoretical Computer Science* **306** (2003) 407–430
12. Galas, D. J., Eggert, M., Waterman, M.S.: Rigorous pattern-recognition methods for DNA sequences: Analysis of promoter sequences from *Escherichia coli*. *Journal of Molecular Biology* **186(1)** (1985) 117–128
13. Gramm, J., Niedermeier, R., Rossmanith, P.: Fixed-parameter algorithms for Closest String and Related Problems. *Algorithmica* **37** (2003) 25–42
14. Gramm, J., Huffner, F., Niedermeier, R.: Closest Strings, Primer Design, and Motif Search. Poster in RECOMB (2002)
15. Hertz, G. Stormo, G.: Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics* **15** (1999) 563–577
16. Keich, U., Pevzner, P. A.: Finding motifs in the Twilight Zone. *Bioinformatics* **18** (2002) 1374–1381
17. Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., Wootton, J. C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science* **262** (1993) 208–214
18. Lawrence, C. E., Reilly, A. A.: An Expectation Maximization (EM) algorithm for the identification and characterization of common sites in unaligned bipolymer sequences. *Proteins: Structure, Function and Genetics* **7** (1990) 41–51.
19. Li, M., Ma, B., Wang, L.: Finding similar regions in many sequences. *Journal of Computer and System Sciences* **65** (2002) 73–96
20. Pevzner, P. A., Sze, S.-H.: Combinatorial approaches to finding subtle signals in DNA sequences. Proc. Eighth International Conference on Intelligent Systems in Molecular Biology (2000) 269–278
21. Price, A., Ramabhadran, S., Pevzner, P. A.: Finding subtle motifs by branching from sample strings. *Bioinformatics* **1(1)** (2003) 1–7
22. Rajasekaran, S., Algorithms for Motif Search, in *Handbook of Computational Molecular Biology*, edited by S. Aluru, Chapman & Hall/CRC, 2006, pp. 37-1–37-21.
23. Rajasekaran, S., Balla, S., Huang, C-H.: Exact Algorithms for Planted Motif Problems. *Journal of Computational Biology* **12(8)** (2005) 1117–1128
24. Sagot M. F.: Spelling approximate repeated or common motifs using a suffix tree. Springer-Verlag LNCS **1380** (1998) 111–127
25. Sinha, S., Tompa, M.: A statistical method for finding transcription factor binding sites. Proc. Eighth International Conference on Intelligent Systems for Molecular Biology (2000) 344–354
26. Staden, R.: Methods for discovering novel motifs in nucleic acid sequences. *Computer Applications in the Biosciences* **5(4)** (1989) 293–298
27. Tompa, M.: An exact method for finding short motifs in sequences, with application to the ribosome binding site problem. Proc. 7th Intl. Conf. on Intelligent Systems for Molecular Biology (ISMB) (1999) 262–271

28. van Helden, J., Andre, B., Collado-Vides, J.: Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology* **281(5)** (1998) 827–842