Available online at www.sciencedirect.com

ScienceDirect

Theoretical Computer Science 395 (2008) 158–170

Theoretical Computer Science

www.elsevier.com/locate/tcs

Detection of subtle variations as consensus motifs[☆]

Matteo Comin[a], Laxmi Parida[b,*]

[a] Department of Information Engineering, University of Padova, Italy
[b] IBM T. J. Watson Research Center, Yorktown Heights, NY 10598, USA

Abstract

We address the problem of detecting consensus motifs, that occur with subtle variations, across multiple sequences. These are usually functional domains in DNA sequences such as transcriptional binding factors or other regulatory sites. The problem in its generality has been considered difficult and various benchmark data serve as the litmus test for different computational methods. We present a method centered around unsupervised combinatorial pattern discovery. The parameters are chosen using a careful statistical analysis of consensus motifs. This method works well on the benchmark data and is general enough to be extended to a scenario where the variation in the consensus motif includes indels (along with mutations). We also present some results on detection of transcription binding factors in human DNA sequences.

Keywords: Pattern discovery; Subtle motifs; Consensus motifs; Transcription factors; Binding sites

1. Introduction

The problem of detecting common motifs across DNA sequences for locating regulatory sites, transcription binding factors or even drug target binding sites is of prime importance. The main difficulty is that these motifs have subtle variations at each occurrence. This problem has been of interest to both biologists and computer scientists. A satisfactory practical solution has been elusive although the problem is defined very precisely:

Problem 1 (*The Consensus Motif Problem*). Given t sequences s_i on an alphabet Σ, a length $l > 0$ and a distance $d \geq 0$, the task is to find all patterns p, of length l that occur in each s_i such that each occurrence p'_i on s_i has at most d mismatches with p.

The problem in this form made its first appearance in 1984 [19]. In this discussion, the alphabet Σ is $\{A, C, G, T\}$ and the problem is made difficult by the fact that each occurrence of the pattern p may differ in some d positions and the occurrence of the consensus pattern p may not have $d = 0$ in any of the sequences. In the seminal paper [19],

[☆] This paper is the extended journal version of [M. Comin, L. Parida, Subtle motif discovery for the detection of dna regulatory sites, in: Proceedings of Asia Pacific Bioinformatics Conference, APBC'07, 2007, pp. 95–104].

[*] Corresponding author.

E-mail addresses: ciompin@dei.unipd.it (M. Comin), parida@us.ibm.com (L. Parida).

0304-3975/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2008.01.017

Waterman and coauthors provide exact solutions to this problem by enumerating *neighborhood* patterns, i.e., patterns that are at most d Hamming distance from a candidate pattern. Sagot gives a good summary of the (computational) efforts in [17] and offers a solution that improves the time complexity of the earlier algorithms by the use of generalized suffix trees. These clever enumeration schemes, though exact, have a drawback that they run in time exponential in the pattern length.

This problem of detecting common subtle patterns across sequences is nevertheless of great interest and various statistical and machine learning approaches, which are inexact but more efficient, have been proposed [11,12,4,8,6]. One of the questions that can be asked to compare and test the efficacy of such methods of consensus motif detection systems is: *Given a set of sequences that harbor (with mutations) k motifs, what percentage of the k motifs does the system recover?* When k is large, all of the above approaches give good average-case performance under this criterion.

Yet another question to ask is: *Given a set of sequences that harbor (with mutations) ONE motif p, does the system recover p?* This is a rather difficult criterion to meet since these algorithms use some form of local search based on Gibbs sampling or expectation maximization or even clever heuristics. Hence it is not surprising that they may miss p. However, a question of this form is a biological reality. Consider the following, somewhat contrived, variation of Problem 1 which is an attempt at simplifying the computational problem.

Problem 2 (*The Planted (l, d)-Motif Problem*). Given t sequences s'_i on Σ, a pattern p of length l is embedded in s'_i, with exactly d errors (mutations), to obtain the sequence s_i of length n, for each $1 \le i \le t$. The task is to recover p, given s_i, $1 \le i \le t$ and the two numbers l and d.

Pevzner and Sze tantalized the community with the *challenge problem*, which was Problem 2 with parameters $n = 600$, $t = 20$, $l = 15$ and $d = 4$ [13]. A thrust of this paper also was the need for the deployment of combinatorial approaches to tackle this thorny problem. One of the algorithms they presented was an exact algorithm where the challenge problem was reduced to finding a t-sized clique in a t-partite graph with at most $n - l + 1$ vertices in each partition. Even the best known heuristics for clique finding problem failed to detect the clique corresponding to the signal. The second algorithm was based on enumerating possible patterns and checking their candidacy for being the subtle pattern using clever heuristics and an exhaustive search in a reduced space. A similar algorithm, with different heuristics was presented in [15,9,7].

One of the most effective algorithms, we found, was the one discussed by Buhler and Tompa [5]. The probabilistic algorithm uses a random projection h and hashes each input l-mer x into bucket $h(x)$. Any hash bucket with sufficiently many entries is explored as a potential embedded motif. This approach solved the challenge problem and some more. There has been a flurry of activity around this problem of subtle motifs [7,10]. For instance, Improbizer[1] uses expectation maximization to determine weight matrixes of DNA motifs that occur improbably often in the input data. See also [16] for some practical implementations of exact approaches.

Overview of our approach. We first clarify the different "motifs" used in this paper: Our central goal is to detect the *consensus* or the *embedded* or the *planted* motif in the given data sets which is also sometimes referred to as the *signal* in the data or the *subtle signal*. When a motif is not qualified with these terms, it refers to a sub-string that appears in multiple sequences, with possible wild cards (see Section 3.1.1 for the formal definition).

We propose an approach that uses unsupervised motif discovery to solve Problem 2. We show that this method works well for the more general Problem 1 as well. Recall that the signal ("subtle motifs") is embedded in t random sequences. The problem is compounded by the fact that although the consensus motif is solid (i.e., an l-mer without wild cards or dont-care characters), it is not necessarily contained in any of the t sequences. However, if we can obtain a correct alignment of the m sequences, then it is relatively easy to extract the consensus motif satisfying the (l, d) constraint. In other words, one of the difficulties of the problem is that the sequences are unaligned. The extent of similarity across the sequences is so little that any global alignment scheme cannot be employed. So we tackle this problem in two steps: First, we identify *potential signal (PS)* segments of interest in the input sequences. This is done by using the imprints of the discovered motifs on the input. Second, amongst these segments, we carry out an exhaustive comparison and alignment to extract the consensus motif. This delineation into two steps helps us also address a more realistic version of the problem that includes insertion and deletion in the consensus motif:

Problem 3 (*The Indel Consensus Motif Problem*). Given t sequences s_i on an alphabet Σ, a length $l > 0$ and a distance $d \ge 0$, the task is to find all patterns p, of length l that occur in each s_i such that each occurrence p'_i on s_i is at an edit distance (mutation, insertion, deletion) at most d from p.

The main focus of our method is in obtaining good quality PS segments and restricting the number of such segments to keep the problem tractable. The Type I error or false negative errors, in detecting PS segments, are reduced by using appropriate parameters for the discovery process based on a careful statistical analysis of consensus motifs which is discussed in Section 2. The Type II error or false positive errors are reduced by using irredundant motifs [3] and their statistical significance measures [2] discussed in Section 3.1. Loosely speaking, irredundancy helps to control the extent of over-counting of patterns and the pattern-statistics helps filter the true signal from the signal-like-background. In the scenario where indels (insertions and/or deletes) are permitted along with mutations, the unsupervised discovery process detects *extensible* motifs (instead of *rigid* motifs that have a fixed imprint length in all the occurrences). Also, the second step uses *gapped* alignments.

In general, it is hard to say how the other approaches can be modified to include indels. For exhaustive methods the introduction of indels would clearly increase the computing time considerably.

All non-exact methods are based on profiles or on k-mers, both of which are rigid. It is reasonable to say that, if the number of indels is much smaller than the size of the consensus, the chance of recovering the signal by such methods may be high. However, when the number of indels grow, it is unclear how these methods would work. Also, it is not immediately apparent to us how these methods can accommodate indels, since the rigidity in the profiles or k-mers is intrinsic to the method. On the other hand, our approach of using extensible motifs is one possible solution to overcome this bottleneck.

2. Statistics of consensus motifs

Here we make some calculations, under simplifying assumptions, to justify the unsupervised motif discovery approach to the problem. We consider the most general version of the problem which is formally stated as Problem 3 in the last section. Recall that this setting permits insertion and deletion as well as mutation in the embedded motif.

Given t sequences of length l each, a pattern satisfies *quorum* K if it occurs in $K' \geq K$ of the given t sequences. Further it is of *maximal* size h, if in each of the K' occurrences, the size cannot be increased without decreasing the number of occurrences K' (see Section 3.1.1 for a more rigorous definition).

For simplicity, the sequences are the same length l and all the t sequences are aligned and we will further assume that a pattern occurs at most once in each sequence.

Given a motif, let the embedded signal in each sequence be constructed with some d edit operations. Given one of these edit operations, we assume

(1) mutation (M), with probability of mutation given as q_M,
(2) deletion (X), with probability of deletion given as q_X and
(3) insertion (I), with probability of insertion given as q_I.

Since the only permissible edit operations are these three,

$$q_M + q_X + q_I = 1.$$

The model. We consider the following simplified model. Given a fixed pattern (or signal), p_{signal}, of length l, we construct t sequences from p_{signal}. To construct each sequence, d positions in the pattern p_{signal} are picked at random and an edit operation (mutation with probability q_M, deletion with probability q_X and insertion with probability q_I) is applied to produce the sequence. Then we study these t sequences.

In other words, given t sequences we assume that they are aligned. For example, the table below on the left shows exactly one edit applied to the signal motif and the table on the right shows the alignment of these embedded motifs.

Edits	signal = ACGTAC						
M	A	C	G	T	C	C	
X	A	G	T	A	C		
I	A	C	G	A	T	A	C
M	A	C	C	T	A	C	
M	G	C	G	T	A	C	

Alignment						
A	C	G	-	T	c	C
A	-	G	-	T	A	C
A	C	G	a	T	A	C
A	C	c	-	T	A	C
g	C	G	-	T	A	C

Assume that d out of the l positions are picked at random on the embedded motif for exactly one of the edit operations, insertion, deletion or mutation. l can be viewed as the size of the motif. Recall that we assume that the sequences are correctly aligned. Then if a position in the aligned sequence is a mismatch, then either it is due to

(1) a mutation (whose probability is q_M) or
(2) an insertion (whose probability is q_I).

Then the probability of this position to be a dot character (mismatch) is

$$\frac{d}{l}\left(q_M + q_I\right).$$

Next, the probability q of a position to be a solid character in a motif is:

$$q = 1 - \frac{d}{l}\left(q_M + q_I\right). \tag{1}$$

q for three scenarios is shown below.

		q_M	q_X	q_I	q
(1)	Exactly d mutations	1	0	0	$1 - d/l$
(2)	Exactly d edits	1/3	1/3	1/3	$1 - 2d/3l$
(3)	Exactly d edits with, equiprobable indel and mutation	1/2	1/4	1/4	$1 - 3d/4l$

When *no more than d' edit operations* are carried out on the embedded motif, it is usually interpreted as each collection of $0, 1, 2, \ldots, d'$ positions being picked with equal probability, and thus

$$d = d'/2$$

for Eq. (1).

Estimating the probability of occurrence of a motif. Recall that q is the probability of a position (character) in the input data to match a character in the pattern (signal). Let H be the number of solid characters and let the motif appear in at least K sequences. For instance in the following alignment, for the first 4 rows, i.e. $k = 4$, the pattern has $H = 3$ solid characters, namely A, T and C, shown in bold at the bottom row. In other words, in these four rows, the solid characters appear in *each* row of the aligned sequences.

Alignment						
A	C	G	-	T	c	C
A	-	G	-	T	A	C
A	C	G	a	T	A	C
A	C	c	-	T	A	C
g	C	G	-	T	A	C

Pattern							
A	C	G	-	T	c	C	←
A	-	G	-	T	A	C	←
A	C	G	a	T	A	C	←
A	C	c	-	T	A	C	←
g	C	G	-	T	A	C	
A				**T**		**C**	k

For a pattern p with some H solid characters, let p occur in some k sequences (and not in the remaining $(t - k)$ sequences). Then

(1) the probability of matches in the (aligned) H solid characters in the k rows is

$$q^{Hk},$$

and

(2) the probability of at least one mismatch in the (aligned) H positions in the remaining $(t - k)$ sequences, is

$$\left(1 - q^H\right)^{t-k}.$$

Thus the probability of occurrence of pattern p is given by

$$\left(1 - q^H\right)^{t-k} q^{Hk}. \tag{2}$$

If E_p denotes the event that p occurs in some fixed k sequences then for any two distinct events, i.e.,

$$p_1 \neq p_2,$$

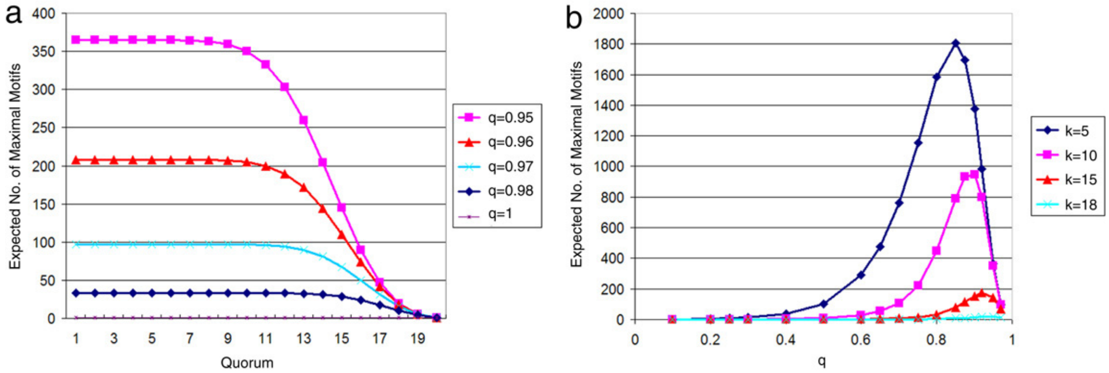

Fig. 1. For $t = 20, l = 20$, the expected number of maximal motifs $E[Z_{K,q}]$, is plotted against (a) quorum K shown along the X-axis (for different values of q), and, (b) against q shown along the X-axis (for different values of quorum K).

E_{p_1} and E_{p_2} are not necessarily mutually exclusive. However, if the pattern is *maximal*, i.e., H is the maximum number of solid characters seen in the k sequences, then for a fixed set of k sequences, there is at most one maximal pattern that occurs in these k sequences and not in the remaining $t - k$ sequences. Further, when the pattern is maximal there is a guarantee of mismatch in the remaining $(l - H)$ positions in all the k rows and the probability of this mismatch is given as

$$(1 - q^k)^{l-H}. \tag{3}$$

Thus to summarize, the probability of occurrence of some pattern with exactly H solid characters in exactly k sequences is given by

$$\left(1 - q^H\right)^{t-k} q^{Hk}(1 - q^k)^{l-H}. \tag{4}$$

Thus if $P_{maximal}(K, H, q)$ is the probability that some maximal pattern with H solid characters and quorum K occurs in the input data, then using Eq. (4),

$$P_{maximal}(K, H, q) = \sum_{k=K}^{t} \binom{t}{k} \left(1 - q^H\right)^{t-k} q^{Hk}(1 - q^k)^{l-H}. \tag{5}$$

Let $Z_{K,q}$ be a random variable denoting the number of maximal motifs with quorum K and q as defined above, and, $E[Z_{K,q}]$ denotes the expectation of $Z_{K,q}$. Using linearity of expectations (for a fixed t and l),

$$\begin{aligned} E[Z_{K,q}] &= \sum_{h=1}^{l} \binom{l}{h} P_{maximal}(K, h, q) \\ &= \sum_{h=1}^{l} \binom{l}{h} \left(\sum_{k=K}^{t} \binom{t}{k} \left(1 - q^h\right)^{t-k} q^{hk}(1 - q^k)^{l-h} \right). \end{aligned}$$

Now, it is rather straightforward to estimate $E[Z_{K,q}]$ given different values of q corresponding to different scenarios. Figs. 1 and 2 show some examples.

2.1. Rationale for using unsupervised motif discovery

A motif of length l that occurs across $t' \leq t$ sequences provides a local alignment of length l for the t' sequences which, in a sense, justifies the simplified scenario of the last section. The best case scenario, for our problem, is when the embedded motif m is identical in all t sequences and the discovery process detects this *single* maximal motif with quorum t. So the scenarios closer to the best case should have fewer (but important) maximal motifs. Fig. 1(a) shows the expected number of motifs with different values of q and quorum K. Notice that the expected number of motifs saturates for small values of K and falls dramatically as K increases. The saturation at lower values occurs since we

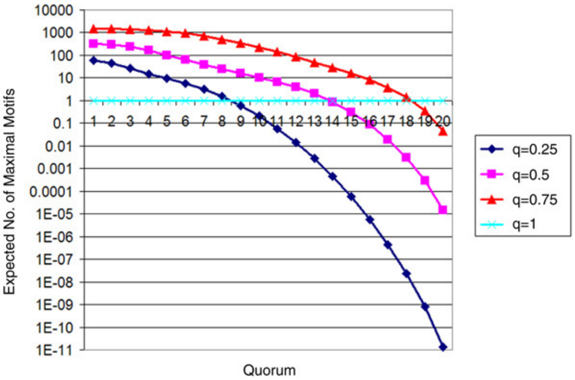

Fig. 2. For $t = 20$, $l = 20$, the expected number of maximal motifs $E[Z_{K,q}]$, is plotted against quorum K shown along the X-axis, for different values of q, in a logarithmic scale. Notice that when $q = 1$, the curve is a horizontal line at $y = 1$. Note that for DNA sequences, $q = 0.25$ corresponds to the random input case.

are seeking *maximal* motifs. Thus as q increases the saturation occurs at a higher value of K. Fig. 1(b) shows the variation of the expected number of maximal motifs with q which is unimodal, for different values of K. The value of q is determined by the given problem scenario and thus a large value of K is a good handle on controlling the number and "quality" of maximal motifs.

The *signal* is embedded in the *background* and it is important to exploit the characteristics that distinguishes one from the other. In our case, we assume that the background is random, in other words it is assumed to be randomly generated using an i.i.d. process. Under this condition, it is easy to see that $q = 1/4$. Thus we need to compare $E[Z_{K,q}]$ with $E[Z_{K,1/4}]$, the expectation for the random case. To compare these expectation curves, particularly around small values (close to 1 in the Y-axis), we study the plots of $\log(E[Z_{K,q}])$ against quorum K in Fig. 2.

For example, consider the case when $q = 0.75$; this is the approximate value of q for the *challenge problem* of Section 1. In Fig. 2, this is shown by the red curve and for large K, say $K \geq 16$, the expected number of motifs is small. Also, the corresponding expected numbers for the random case is extremely low, thus providing a strong contrast in the number of expected motifs. Hence the reasonable choice for the quorum parameter K is 16 or more, in the unsupervised discovery process.

Before we conclude this section, we must point out that in the case where the embedded motif is changed with insertions and/or deletions (indels), the q value is computed appropriately using Eq. (1) and the corresponding expectation curve in Fig. 2 is studied. However, the burden is heavier on the unsupervised discovery process and we use the extensible (or, variable-sized gaps) motif discovery capability in Varun [2] available at:
http://www.research.ibm.com/computationalgenomics.

3. SubtleVarun: Our approach

Here we present our approach, *SubtleVarun*, which detects the consensus motifs in two steps. We first locate regions in the sequence called *potential signal* (PS) segments. The statistical analysis of the previous section suggests that the detection of PS segments via unsupervised motif discovery is indeed possible. The two important parameters in the combinatorial discovery process are K and D: K is the quorum or the minimum number of sequences where the pattern must occur and D is the size of the gap between any two solid characters in the pattern. In the second step we carry out a local alignment of these short segments and extract the consensus motif.

3.1. (Step 1) Detecting PS segments

As seen in the last section, we expect to see more maximal patterns in the signal region than the background in an appropriate range of quorum K. We extract all common motifs across sequences using an unsupervised combinatorial motif discovery process. We use the system Varun [2] for this purpose. This allows us to discover motifs with "dont-cares" or wild-cards. The number of such characters is controlled by the parameter D in Varun, which is a bound on the number of "dont-cares" between any two solid characters in a pattern. Next, we simply count the number of motifs that cover a position i on the input. The first prediction of the PS segments are the positions (i's) with high counts.

This elementary rule works well for simple cases like Problem 2 with $n = 600$, $t = 20$, $l \leq 10$ and $d = 2$. Here the PS segments are predicted accurately. However, for $d > 2$ we found that it is difficult to distinguish the true from the false PS segments using this simple approach. To weed out these wrong PS segments, we explored other means of pruning the motifs using some combinatorial and statistical approaches. Firstly, we use the idea of *irredundant* or *basis* motifs [3], to avoid overcounting of patterns that cover the same region multiple times on the sequence. Secondly, we consider only those motifs that have a significant z-score and also, biased the motif count at a position i on the input with the probability of the occurrence of that motif. We briefly digress here, and give a short exposition, taken from [3,2], to keep the paper self-contained.

3.1.1. Irredundancy of motifs [3]

Let s be a sequence from an alphabet $\Sigma \cup \{.\}$, where '.' $\notin \Sigma$ denotes a don't care (*dot*, for short) and the rest are *solid* characters, we use σ to denote a singleton character. For characters e_1 and e_2, we write $e_1 \preceq e_2$ if and only if e_1 is a dot or $e_1 = e_2$. Allowing for spacers in a string is what makes it gapped. Such spacers are indicated by a dot character. Whenever defined, D will denote the maximum number of consecutive dots allowed in a string. A string m occurs at position l on s if for $1 \leq j \leq |m|$ the following holds:

$$m[j] \preceq s[l + j - 1].$$

For a sequence s and positive integer k, $k \leq |s|$, a string m is a *motif* of s with $|m| > 1$ and location list

$$\mathcal{L}_m = \{l_1, l_2, \ldots, l_p\},$$

if both of the following conditions hold:

(1) $m[1]$ and $m[|m|]$ are solid and
(2) $\mathcal{L}_m$ is the list of all and only the occurrences of m in s.

Given a motif m let $m[j_1], m[j_2], \ldots, m[j_l]$ be the l solid elements in the motif m. Then the *sub-motifs* of m are given as follows: for every j_i, j_t, the sub-motif $m[j_i \ldots j_t]$ is obtained by dropping all the elements before (to the left of) j_i and all elements after (to the right of) j_t in m. We also say that m is a *condensation* for any of its sub-motifs. For example, let

$$m_1 = x...yzw..x.x.$$

Then some sub-motifs of m_1 are

$$x...yzw..x.x, \quad yzw..x.x, \quad zw..x.x, \quad yzw..x, \quad yzw, \quad zw..x, \quad w..x.$$

We are interested in motifs for which any condensation would disrupt the list of occurrences. Formally, let m_1, $m_2, \ldots, m_k$ be the motifs in a string s. A motif m_i is *maximal in length* if there exists no m_l, $l \neq i$ with $|\mathcal{L}_{m_i}| = |\mathcal{L}_{m_l}|$ and m_i is a sub-motif of m_l. A motif m_i is *maximal in composition* if no dot character of m_i can be replaced by a solid character that appears in all the locations in $\mathcal{L}_{m_i}$. A motif is called *maximal* if it is maximal in composition and in length.

Requiring maximality in composition and length limits the number of motifs that may be usefully extracted and accounted for in a string. However, the notion of maximality alone does not suffice to bound the number of such motifs. It can be shown that there are strings that have an unusually large number of maximal motifs without conveying extra information about the input.

A maximal motif m is *irredundant* if m and the list $\mathcal{L}_m$ of its occurrences cannot be deduced by the union of a number of lists of other maximal motifs. Conversely, we call a motif m *redundant* if m (and its location list $\mathcal{L}_m$) can be deduced from the other motifs *without* knowing the input string s. More formally:

Definition 1 (*Redundant, Irredundant Motif*). A maximal motif m, with location list $\mathcal{L}_m$, is redundant if there exist some $p > 1$ maximal motifs m_j, $1 \leq j \leq p$, such that

$$\mathcal{L}_m = \mathcal{L}_{m_1} \cup \mathcal{L}_{m_2} \cdots \cup \mathcal{L}_{m_p},$$

(i.e., every occurrence of m on s is already implied by one of the motifs $m_1, m_2, \ldots, m_p$).
A maximal motif that is not redundant is called an irredundant motif.

Thus the set of irredundant motifs, denoted by $\mathcal{B}$ (also called the *basis* set), selects only those motifs that can describe the entire motif space. It also reduces the search space dimensionality from exponential to polynomial, without any loss of information.

For a given k, the basis is unique and a more detailed description can be found in [3]. In particular if n is the length of the input string and k is the minimum quorum one can prove that [3]

$$|\mathcal{B}| \leq n - 1 \text{ when } k = 2,$$

and that in general [14]

$$|\mathcal{B}| \leq \binom{n-1}{k-1}.$$

Extensible motifs. The motifs described above are also called *rigid* motifs. In other words, the length of the imprint of each occurrence of a motif is the same. However, we can define *extensible* motifs where this imprint length may change (in a controlled manner) at each occurrence [2]. In other words an extensible motif is a concatenation of rigid strings (each with possible dot characters) and the gap between the rigid sections is denoted by a dash ('-') character which represents up to D gaps in the imprint of each occurrence. In other words the number of dot characters corresponding to a dash character is

$$0, 1, 2, \ldots, D.$$

Note that there is no pre-determined bound on the number of dash characters in a motif since the motif is assumed to be maximal.

In [3], we show that given a string of length n and $k = 2$, each element of the basis corresponds to an autocorrelation of the string and since there are no more than n autocorrelations, the size of the basis is no more than n. However, when the motifs are extensible, this number does not hold.

Assume that a rigid segment of the motif must be of at least length r, then the number of autocorrelations is

$$O\left(n2^{\frac{n}{r}}\right).$$

It is unclear to us whether we can get a better bound on the size of the basis of extensible motifs. Nevertheless, the use of irredundancy is useful even in extensible motifs, since repetitive motifs can be filtered without leading to over counts.

3.1.2. *Statistical significance of motifs [2]*

When the alphabet size $|\Sigma| << n$, the chances of finding recurring motifs even in random sequences increase dramatically. Thus after the "combinatorial" elimination of candidate motifs, using quorum, density parameters (D) and irredundancy, we also use "statistical" elimination using z-scores.

We recall the following from [2]. An extensible motif is *degenerate* if it can possibly have multiple occurrences at a site i on the input s. Let m be an extensible non-degenerate motif generated by a stationary, *iid* source which emits ($\sigma \in \Sigma$) with probability p_σ. Let j_σ be the number of times σ appears in m and let e be the number of dash characters in m with gap sizes $\alpha_1, \alpha_2, \ldots \alpha_e$, then

$$p_m = \prod_{\sigma \in \Sigma} (p_\sigma)^{j_\sigma} \prod_{i=1}^{e} |\alpha_i|. \tag{6}$$

Further, let M^s denote a set of strings that has only the solid characters of at least s occurrences of m. For example, consider the motif a-b with realizations

$$a.b, \quad a..b \text{ and } a\ldots b.$$

Then

$$M^1 = \{a.b, a..b, a\ldots b\}$$

since m occurs once on each $m \in M^1$;

$$M^2 = \{a.bb, a..bb, a.b.b\}$$

since m occurs twice on each $m \in M^2$;

$$M^3 = \{a.bbb\}$$

since m occurs three times on $m \in M^3$.

Let m be a degenerate (possibly with multiple occurrences at a site) extensible motif, and let

$$p_{m^k} = \sum_{m' \in M^k} p_{m'},$$

then

$$p_m = \sum_{k=0}^{r-1} (-1)^k \left(p_{m^{k+1}} \right). \tag{7}$$

This follows directly from the inclusion–exclusion principle. Notice that for a degenerate motif, Eq. (6) is the zeroth-order approximation of Eq. (7). The first-order approximation is

$$p_m \approx p_{m^1} - p_{m^2}$$

and the second-order approximation is

$$p_m \approx p_{m^1} - p_{m^2} + p_{m^3}$$

and so on. Using Bonferroni's inequalities, a kth-order approximation of p_m is an over-estimate of p_m, if k is odd.

To summarize, if p_m be the probability of the motif m occurring at any location i on the input string s with $n = |s|$ and k_m is the observed number of times it occurs on s and if it can be assumed that the occurrence of a motif m at a site is an i.i.d. process, then the z-score is given as:

$$\frac{k_m - np_m}{\sqrt{np_m(1 - p_m)}}. \tag{8}$$

Note that the non-maximal motifs are no more surprising, in terms of their z-score, than the maximal motifs, thus allowing us to combine algorithmically maximality with z-scores. See [2] for a detailed justification and discussion.

3.1.3. Back to PS segment computation

We use Varun to discover irredundant motifs in the input data. In the right, the motif discovery parameters are K and D and $l = 15$, $d = 4$, $t = 20$, $n = 600$ and the value of q is

$$\frac{11}{15} \approx 0.73,$$

using Eq. (1). Column I shows the number of correct PS segments predicted using *all* motifs and column II shows the same using only *irredundant motifs*. In all the cases, there is an increase in the number of correctly detected positions for the latter.

K	D	I	II
20	2	2	3
19	2	0	1
20	3	3	4
19	3	1	2
20	4	2	5

We compute the z-score of each irredundant motif using the equation in the previous section and filter these motifs based on a cut-off threshold z-score. We further use a weighted count for each input position in the imprint of the motif m, where the weight is

$$\frac{1}{p_m}$$

and p_m is computed as in Eq. (7). Fig. 3 shows the results for a variety of settings comparing the use of statistical methods (both z-score and weighted counting), called Method II, with the one that does not use them, called Method I.

$l = 10, d = 2$

Motif params			Methods	
K	D	M	I	II
10	2	95	8	7
9	2	236	8	10
8	2	434	7	8

(a) $n = 100, t = 10$

Motif params			Methods	
K	D	M	I	II
20	2	281	10	10
19	2	459	12	13
18	2	588	18	18

(b) $n = 200, t = 20$

$t = 20, l = 15, d = 4$

Motif params			Methods	
K	D	M	I	II
20	2	539	2	4
19	2	647	6	7
18	2	837	12	12
20	3	952	5	6
19	3	1164	11	10
18	3	1582	13	13
20	4	1454	8	9
19	4	1832	9	10
18	4	2577	11	11

(c) $n = 300$

Motif params			Methods	
K	D	M	I	II
20	2	1588	2	2
19	2	3526	1	1
18	2	5456	1	1
16	2	7316	1	2
20	3	3348	4	4
19	3	7885	2	2
18	3	12444	1	1
17	3	15318	2	3
16	3	17017	0	1

(d) $n = 400$

Fig. 3. Number of PS segment positions predicted correctly using Methods I and II for different parameters. The motif discovery parameters are K and D and M is the total number of irredundant motifs discovered in the input. The values of q, obtained using Eq. (1), are as follows: (a) & (b) $q = 0.8$, (c) & (d) $q = 0.73$.

Notice that using Method II, we can restore all 10 positions of the $n = 200, t = 20, l = 10$ and $d = 2$ of Problem 2. In the experiments for $l = 15$ and $d = 4$, we can recover 4 positions correctly out of 20. We find that only in two cases Method I recovers more PS segment positions than Method II. However, in all the remaining 22 cases, Method II outperforms Method I.

Since it is very difficult to detect 100% of the PS segments correctly in this step alone, we use these partial PS segments in the next step to reconstruct the true signal.

3.2. (Step 2) Processing PS segments

In the previous step we identified the potential signal (PS) segments in the input. Next, we merge the information from each sequence by combining different PS segments. Assuming that the PS segment is predicted correctly, the planted motif is embedded in this segment. If the length of the consensus motif is known, say l, then the PS segment is constrained to be sub-string of length $2 \times l$. Thus given a candidate position i in sequence s, the signal is contained in the interval $s[i - l, i + l]$.

We next pick one PS segment from each sequence to "locally align" the segments across some t' sequences. We enumerate all the

$$\binom{t}{t'}$$

configurations here. Let the t' PS segments, each from a distinct sequence, be given as

$$(s_{i_1}[b_{i_1}, e_{i_1}], s_{i_2}[b_{i_2}, e_{i_2}], \ldots, s_{i_{t'}}[b_{i_{t'}}, e_{i_{t'}}]).$$

We make the assumption that the starting position x_{i_j} of the consensus motif in sequence s_{i_j} lies in the sub-string

$$s_{i_j}[b_{i_j}, e_{i_j}],$$

i.e., $b_{i_j} \le x_{i_j} \le e_{i_j}$. We are seeking all possible alignments of length l using these PS segments.

We use the following measure to evaluate an alignment. The *majority* string s_m, of length l, is simply the string obtained by using the majority base at each aligned position (column). The score f is the sum total of the aligned positions in all the t' segments that agree with s_m. For example, consider the aligned segments on the right where $t = 5, l = 8, d = 3$ and $t' = 5$. s_m is shown in bold and $f = 28$.

$$\begin{array}{rcccccccc}
(1) & \text{---}A & C & T & G & C & T & C & C\text{---} \\
(2) & \text{---}A & G & G & G & T & T & G & A\text{---} \\
(3) & \text{---}C & C & G & G & T & T & G & A\text{---} \\
(4) & \text{---}C & C & T & C & T & A & C & A\text{---} \\
(5) & \text{---}A & C & G & G & T & - & C & A\text{---} \\
s_m = & \mathbf{A} & \mathbf{C} & \mathbf{G} & \mathbf{G} & \mathbf{T} & \mathbf{T} & \mathbf{C} & \mathbf{A}
\end{array}$$

Since our first step is very tightly controlled, we found in practice that there are only a few candidate PS segments. Also, in the model that uses insertion and deletion (i.e., the length of the imprint of the occurrence of the consensus motif in each sequence is not necessarily l), we use the same score by keeping track of the alignment columns: deletions and insertions result in gaps in some sequences in the alignment (see sequence 5 in the above example). We consider all those alignments, whose score f exceeds a fixed threshold

$$\text{Thresh}_{t'}.$$

In all our experiments we have used $t' = 3$ and the threshold values are reported in the experiments.

Extracting the consensus motif across t sequences. At the previous step, we have multiple alignments, where each alignment is across some $t'(\leq t)$ sequences. From these we need to extract the consensus motif across all the t sequences. For each alignment, we designate the majority sub-string s_m (see last section) as the putative consensus motif. Then we scan all the t input strings for the occurrence of s_m with at most d errors which can be done in linear time. For each sequence, we pick the best occurrence, i.e., the one with the minimum edit distance from s_m. In practice, this step very quickly discards the erroneous consensus motifs and quickly converges to the one(s) satisfying the distance constraint of d.

4. Results

Let P be the set of all positions covered by the prediction and S be the same set for the embedded motif. The score of the prediction P, with respect to the embedded motif, can be given as (see [18]):

$$\text{score} = \frac{|P \cap S|}{|P \cup S|}.$$

The score is 1 if the prediction is 100% correct. However, even for values much smaller than 1, the embedded motif may be computed correctly. This measure is rather stringent and so we use yet another measure, the *solution coverage* (SC) score. This is defined as the number of sequences that contains at least one occurrence of the predicted motif whose distance from the prediction is within the problem constraint i.e., bounded by d. Again if the coverage is equal to the total number of sequences t, then the prediction can be considered 100% correct.

Results on benchmark synthetic data. We report our results in terms of these two measures in Fig. 4 averaged over eight random experiments.

Each experiment is defined by the four parameters n, t, l and d. In the unsupervised motif discovery process of the first step we use parameters $K = t = 20$ and $0 < D < 4$. The high K value was suggested by the statistical analysis in Section 2 and confirmed by our experiments in Section 3.1. In the second step we use $t' = 3$ based on our experiments reported in Fig. 3. In Fig. 4(a)–(c), we show the performance measures for various instances of Problem 2.

We compare our results with what we found as the best performing algorithm, PROJECTION [5]. In all cases our best results are similar, or slightly better, than PROJECTION as shown in Fig. 4. We observe that as we increase the number of gaps D, the *score* improves. In particular if $D = 0$ (i.e., solid motifs), the chances of success drops dramatically. We observe a similar tendency in Problem 3 as shown in Fig. 4(d) and (e). Although this version of the problem, with indels, should be harder, we find that the method gives surprisingly good results.

*Results on Human hm*01*r data.* We have tested the system on various real data sets and we give details of one such case—that of detecting transcription binding factors on human DNA sequences on the data set suggested by

K	D	N	Score	SC
20	1	2	0.066	10
20	2	2	0.415	12
20	3	4	0.95	20
20	4	3	0.94	20

(a) $l = 15, d = 4$
$Score_{PRJ} = 0.93$

K	D	N	Score	SC
20	0	0	0.02	10
20	1	1	0.49	11
20	2	1	0.8	20
20	3	1	0.93	20
20	4	2	0.91	20

(b) $l = 17, d = 5$
$Score_{PRJ} = 0.93$

K	D	N	Score	SC
20	1	2	0.75	11
20	2	2	0.95	20
20	3	4	0.95	20

(c) $l = 19, d = 6$
$Score_{PRJ} = 0.96$

K	D	N	Score	SC
20	0	1	0.05	5
20	1	3	0.75	20
20	2	3	0.81	20

(d) $l = 15$, 3 mutations & 1 indel

K	D	N	Score	SC
20	0	1	0.09	8
20	1	3	0.68	20
20	2	4	0.78	20

(e) $l = 15$, 2 mutations & 2 indels

Fig. 4. In all cases, $t = 20$, $n = 600$. The motif discovery parameters are K and D and we use $t' = 3$ and the values of $Thresh_{t'}$ are as follows: (a) 32 (b) 36 (c) 40 (d) & (e) 30. The results are averaged over 8 random problem instances. N is the total number of PS segments predicted correctly. See text for definitions of *Score* and *SC*. $Score_{PRJ}$ is the score for the PROJECTION algorithm by Tompa et al.

Tompa [18]. The details are as follows:

No	pos	Predictions										M	I
0	−101		*T*	*G*	*A*	*C*	*G*	*T*	*C*	*A*		–	1
1	−299		*T*	*G*	*C*	–	*G*	*T*	*C*	*A*		1	–
2	−71		*T*	*G*	*A*	*C*	*A*	*T*	*C*	*A*		1	1
3	−69	*A*	*T*	*G*	*A*	–	*G*	*T*	*C*	*A*	*G*	–	2
4	−527		*T*	*G*	*C*	*G*	*A*	*T*	*G*	*A*		2	1
6	−173		*T*	*G*	*A*	–	*C*	*T*	*A*	*A*		2	–
7	−1595		*T*	*G*	*A*	–	*A*	*T*	*G*	*A*		2	–
8	−221		*T*	*G*	*G*	–	*G*	*T*	*C*	*T*		2	–
9	−69		*T*	*G*	*A*	–	*C*	*T*	*G*	*C*		3	–
10	−105		*T*	*G*	*A*	–	*A*	*T*	*C*	*A*		1	–
12	−780		*T*	*G*	*C*	–	*G*	*T*	*C*	*A*		1	–
14	−1654	*A*	*T*	*G*	*A*	–	*A*	*T*	*C*	*A*		1	1
15	−69	*A*	*T*	*G*	*A*	–	*G*	*T*	*C*	*A*	*A*	–	2
16	−97		*T*	*G*	*A*	–	*G*	*T*	*A*	*A*		1	–
17	−1936	*A*	*T*	*G*	*A*	–	*A*	*T*	*C*	*A*		1	1
signal			**T**	**G**	**A**		**G**	**T**	**C**	**A**			

The parameters for this data set are $n = 2000$, $t = 18$. Note that we had to estimate l and d through a series of trials. l was estimated to be 7 and d to be 3. We use parameters $K = 18$ and $D = 1$ in the motif discovery process in Step 1 and use $t' = 3$ and $Thresh_{t'} = 12$ in Step 2. We identify the signal in 15 of the 18 sequences at positions given in the *pos* column. We miss the signal in only one sequence (sequence no 5) and the signal is absent in two other sequences (no 11 and 13). Overall our prediction includes 114 true positives out of 236 positions covered by this transcription factor. The remaining positions that are not covered by our prediction, excluding sequence no 5, are left and right extensions of the results reported in the previous table. However, by definition these extensions cannot be part of a planted motif.

We reconstruct the consensus sequence as

$$TGAGTCA$$

which is at most 3 edit distance away from the "embedded" signals. In the table M denotes number of mutations and I the number of insertions; no deletions were found.

Interestingly, we notice that the performance on the same data by no-indels methods are consistently poor. In particular out of 14 methods reported in [18], 5 report no prediction, 7 report predictions with no overlap, and only two methods, Improbizer [1] and MITRA [7], overlap the correct solution by respectively 5 and 9 positions (out of 236). For this data we can conclude that our prediction better approximates the real binding sites. Although more experiments would be needed for definitive conclusions, this is an encouraging fact.

5. Concluding remarks

The problem of detecting subtle consensus motifs is tricky and a purely combinatorial or a purely statistical approach has been unsatisfactory (see Section 1). It appears as it requires a delicate combination of the two methods. We have presented a method that uses unsupervised combinatorial pattern discovery, followed by a careful statistical refinement and processing. Since we use tried-and-tested tools such as pattern discovery, in the first step, and local alignment, in the second step, we have focussed more on choosing and combining appropriate parameters. Also, the extension of the method to handling a more general scenario such as inclusion of indels (insertion and/or deletion) in the embedded motif has been relatively straightforward. We achieved this by using extensible motifs in the pattern discovery process of the first step and gapped alignment in the second step. The results on benchmark data and some real DNA sequences have been very encouraging. We are looking at the yet harder instance of the problem which is the task of finding subtle motifs within the same sequence.

Acknowledgement

The first author's work was done during internship at IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA.

References

[1] W. Ao, J. Gaudet, W.J. Kent, S. Muttumu, S.E. Mango, Environmentally induced foregut remodeling by pha-4/foxa and daf-12/nhr, Science 305 (5691) (2004) 1743–1746.

[2] A. Apostolico, M. Comin, L. Parida, Conservative extraction of over-represented extensible motifs, ISMB (Supplement of Bioinformatics) 21 (2005) 9–18.

[3] A. Apostolico, L. Parida, Incremental paradigms for motif discovery, Journal of Computational Biology 11 (4) (2004) 15–25.

[4] T.L. Bailey, C. Elkan, Fitting a mixture model by expectation maximization to discover motifs in biopolymers, in: Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 1994, pp. 28–36.

[5] J. Buhler, M. Tompa, Finding motifs using random projections, Journal of Computational Biology 9 (2) (2002) 225–242.

[6] M. Comin, L. Parida, Subtle motif discovery for the detection of dna regulatory sites, in: Proceedings of Asia Pacific Bioinformatics Conference, APBC'07, 2007, pp. 95–104.

[7] Eleazar Eskin, Pavel Pevzner, Finding composite regulatory patterns in DNA sequences, Bioinformatics 18 (2002) 354–363.

[8] G.Z. Hertz, G.D. Stormo, Identifying DNA and protein patterns with statistically significant alignments of multiple sequences, Bioinformatics 15 (1999) 563–577.

[9] Keich, Pevzner, Finding motifs in the twilight zone, in: Annual International Conference on Computational Molecular Biology, April 2002, pp. 195–204.

[10] Uri Keich, Pavel Pevzner, Subtle motifs: Defining the limits of motif finding algorithms, Bioinformatics 18 (2002) 1382–1390.

[11] C.E. Lawrence, A.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 (1990) 41–51.

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

[13] P.A. Pevzner, S.-H. Sze, Combinatorial approaches to finding subtle signals in DNA sequences, in: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, AAAI Press, 2000, pp. 269–278.

[14] N. Pisanti, M. Crochemore, R. Grossi, M.-F. Sagot, Bases of motifs for generating repeated patterns with wild cards, IEEE/ACM Transaction on Computational Biology and Bioinformatics 2 (1) (2005) 40–50.

[15] Alkes Price, Sriram Ramabhadran, Pavel Pevzner, Finding subtle motifs by branching from sample strings, Bioinformatics (1) (2003) 149–155.

[16] S. Rajasekaran, S. Balla, C.-H Huang, Exact algorithms for planted motif problems, Journal of Computational Biology 12 (8) (2005) 1117–1128.

[17] M.F. Sagot, Spelling approximate repeated or common motifs using a suffix tree, in: Latin 98: Theoretical Informatics, in: Lecture Notes in Computer Science, vol. 1380, 1998, pp. 111–127.

[18] Martin Tompa, Nan Li, Timothy L. Bailey, George M. Church, Bart De Moor, Eleazar Eskin, Alexander V. Favorov, Martin C. Frith, Yutao Fu, W. James Kent, Vsevolod J. Makeev, Andrei A. Mironov, William Stafford Noble1, Giulio Pavesi, Graziano Pesole, Mireille Rgnier, Nicolas Simonis, Saurabh Sinha, Gert Thijs, Jacques van Helden, Mathias Vandenbogaert, Zhiping Weng, Christopher Workman, Chun Ye, Zhou Zhu, Assessing computational tools for the discovery of transcription factor binding sites, Nature Biotechnology 23 (2005) 137–144.

[19] M.S. Waterman, R. Aratia, D.J. Galas, Pattern recognition in several sequences: Consensus and alignment, Bulletin of Mathematical Biology 46 (4) (1984) 515–527.