



Conservative extraction of over-represented extensible motifs

Alberto Apostolico^{1,2,*}, Matteo Comin² and Laxmi Parida³

¹Department of Computer Sciences, Purdue University, Computer Sciences Building, West Lafayette, IN 47907, USA, ²Dipartimento di Ingegneria dell'Informazione, Università di Padova, Padova, Italy and ³IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598, USA

Received on January 15, 2005; accepted on March 27, 2005

ABSTRACT

Motivation: The discovery of motifs in biosequences is frequently torn between the rigidity of the model on the one hand and the abundance of candidates on the other. In particular, the variety of motifs described by strings that include 'don't care' (dot) patterns escalates exponentially with the length of the motif, and this gets only worse if a dot is allowed to stretch up to some prescribed maximum length. This circumstance tends to generate daunting computational burdens, and often gives rise to tables that are impossible to visualize and digest. This is unfortunate, as it seems to preclude precisely those massive analyses that have become conceivable with the increasing availability of massive genomic and protein data. Although a part of the problem is endemic, another part of it seems rooted in the various characterizations offered for the notion of a motif, that are typically based either on syntax or on statistics alone. It seems worthwhile to consider alternatives that result from a prudent combination of these two aspects in the model.

Results: We introduce and study a notion of extensible motif in a sequence which tightly combines the structure of the motif pattern, as described by its syntactic specification, with the statistical measure of its occurrence count. We show that a combination of appropriate saturation conditions (expressed in terms of minimum number of dots compatible with a given list of occurrences) and the monotonicity of probabilistic scores over regions of constant frequency afford us significant parsimony in the generation and testing of candidate over-represented motifs.

The merits of the method are documented by the results obtained in implementation, which specifically targeted protein sequence families. In all cases tested, the motif reported in PROSITE as the most important in terms of functional/structural relevance emerges among the top 30 extensible motifs returned by our algorithm, often right at the top. Of equal importance seems the fact that the sets of all surprising motifs returned in each experiment are extracted

faster and come in much more manageable sizes than would be obtained in the absence of saturation constraints.

Availability: This software will be available for use with the suite of tools at www.research.ibm.com/bioinformatics

Contact: axa@dei.unipd.it

Please check contact details

1 INTRODUCTION AND SUMMARY

1.1 Preliminaries¹

The discovery of motifs in biosequences is attracting increasing interest due to the perceived multiple implication of motifs in biological structure and function. The approaches to motif discovery may be partitioned in two main classes. In the first class, the sample string is tested for occurrences of motifs in a family of a priori defined abstract models or templates. The second class of approaches assumes that the search may be limited to substrings in the sample or to some more or less controlled neighborhood of these substrings. The approaches in the first class are more rigorously justifiable, but often pose daunting computational burdens. Those in the second class tend to be computationally viable but rest on more shaky methodological grounds.

The characterizations offered for the notion of a motif could be partitioned roughly into statistical and syntactic. In a typical statistical characterization, a motif is a sequence of m positions such that at each position each character from (some subset of) the alphabet may occur with a given probability or weight. This is often described by a suitable matrix or profile, where columns correspond to positions and rows to alphabet characters (Hertz and Stormo, 1999; Lawrence *et al.*, 1993). The lineage of syntactic characterizations could be ascribed to the theory of error correcting codes: a motif is a pattern w of length m and an occurrence of it is any string at a distance of d , the distance being measured in terms of errors of a certain type. For example, we can have only substitutions in the Hamming variant, substitutions and indels in the Levenstein variant, and so on (Keich and Pevzner, 2002; Pevzner and

Could you please consider a change in the title for Section 1 since summary seems to imply an abstract

*To whom correspondence should be addressed.

¹The expert reader may skip this part.

Sze, 2000). Syntactic characterizations enable us to describe the model of a motif, a realization of it or both, as a string or simple regular expression over an extension of the input alphabet Σ , e.g. over $\Sigma \cup \{.\}$, where ‘.’ denotes the ‘don’t care’ (dot) character.

Irrespective of the particular model or representation chosen, the tenet of motif discovery equates over-representation of a motif with surprise and hence with interest. Thus, any motif discovery algorithm must ultimately weigh motifs against some threshold, based on a score that compares empirical and expected frequency, perhaps with some normalization. The departure of a pattern w from expectation is commonly measured by the so-called z -scores (Leung et al., 1996), which have the form

$$z(w) = \frac{f(w) - E(w)}{N(w)},$$

where $f(w) > 0$ represents a frequency, $E(w) > 0$ an expectation and $N(w) > 0$ is the expected value of some function of w . For a given z -score function, a set of patterns W and real positive threshold T , patterns such as $z(w) > T$ or $z(w) < -T$ are dubbed over-represented or under-represented, or simply surprising. The problem is that the number of patterns extracted in this way may escalate quite rapidly, a circumstance that seems to preclude precisely those massive analyses which have become conceivable with the increasing availability of whole genomes. Large-scale statistical tables may not only impose unbearable computational burden. They are also impractical to visualize and use—a circumstance that may defy the purpose of building them in the first place. A little reflection establishes how exponential build-up may take place. Assume that on the binary alphabet both *aabaab* and *abbabb* are asserted as reflections of candidate interesting motifs. We can give a concise description of this motif by writing *a.ba.b*, with ‘.’ denoting the dot, and then look for further occurrences of this motif. By this, however, we have immediately annexed also the spurious patterns *ababbb* and *abbaab*. A similar problem presents itself in the approaches that resort to profiles or weighed matrices mentioned earlier. In all these cases, the risk is having to tell Horatio that there are more things in his philosophy than are dreamed of in heaven and earth.² Despite setting aside computational aspects, tables that are too large at the outset risk to saturate the visual bandwidth of the user. In this spirit, approaches that limit from the start the number of patterns to be considered may ripe a more significant throughput, even in the comparison with exhaustive methods.

We regard the motif discovery process as distributed on two stages, where the first stage unearths motifs endowed with a certain set of properties and the second filters out the interesting ones. Since the redundancy builds up in the first stage,

it is there that we have to look for possible ways of reducing the unnecessary throughput. Since over-representation is measured by a score, one would have to find ways to neglect candidate motifs that cannot possibly make it to the top list, and ideally spot such motifs before they are even computed. Counterintuitive as it might look, we show that such a possibility may be offered by certain attributes of ‘saturation’ that combine in a unique way the syntactic structure and the list of occurrences or frequency for a motif. With solid words, for example, we know that in the worst case the number of distinct substrings in a string can be quadratic in the length of that string. Nevertheless, if we partition the substrings into buckets, by putting in the same bucket strings that have exactly the same set of occurrences, we only need a number of buckets linear in the textstring (Blumer et al., 1985). Similar linear bounds were established for special classes of rigid motifs containing ‘dots’ (Apostolico and Parida, 2004). When combined with intervals of score monotonicity, properties of this kind support the global detection of unusual words of any length in overall linear space (Apostolico et al., 2002). Some of these conservative scoring techniques were extended recently to rigid motifs with a prescribed maximum number of mismatches or dot (Apostolico and Pizzi, 2004).

1.2 Main results

In this paper, we introduce and study a characterization of extensible motifs in the definition of which structural or syntactic properties and occurrence statistics are solidly intertwined. We show that a prudent combination of saturation conditions (expressed in terms of minimum number of dots compatible with a given list of occurrences) and monotonicity of scores afford us significant parsimony in the generation and testing of candidate over-represented motifs. More specifically, we isolate as candidate surprising motifs only the members of an a priori well identified set of ‘maximally saturated’ patterns. By this set being identifiable a priori, we mean that the motifs in the set can be known before any score is computed. By neglecting the motifs other than those in our set, we would not be overlooking any surprising motif. In fact, we maintain that any such motif: (1) is embedded in one of the saturated ones and (2) does not achieve a larger score than the latter (hence, computing its score and publishing it explicitly would take more time and space but not add information). The results of this paper apply to extensible patterns a philosophy previously applied to rigid motifs described (1) by solid words (Apostolico et al., 2002) and (2) by words of some specified fixed length affected by a specified maximum number of errors (Apostolico and Pizzi, 2004). The transition from rigid to extensible motifs requires the orchestration of substantially novel concepts and tools, resulting in an algorithm for the extraction and weighing of extensible motifs, and a suite of software programs implementing the whole. The merits of the method are tested on families of protein sequences, as documented in the last part of the paper. In all cases tested, the

²‘There are more things in heaven and earth, Horatio, Than art dreamt of in your philosophy’—W. Shakespeare, *Hamlet*, I, v [76].

motif reported in PROSITE as the most important in terms of functional/structural relevance emerges either at the top or among the top ten or so of the (short) output list. Experiments related to the sensitivity and selectivity of the method are also reported.

1.3 Basic definitions and concepts

To proceed with a formal definition of the concepts highlighted above, let s be a sequence of sets of characters from an alphabet $\Sigma \cup \{.\}$, where ‘.’ $\notin \Sigma$ denotes a dot and the rest are solid characters. We use σ to denote a singleton character or a subset of Σ . For character (sets) e_1 and e_2 , we write $e_1 \leq e_2$ if and only if e_1 is a dot or $e_1 \subseteq e_2$. Allowing for spacers in a string is what makes it extensible. Such spacers are indicated by annotating the dot characters. Specifically, an annotated ‘.’ character is written as $^{\alpha}$ where α is a set of positive integers $\{\alpha_1, \alpha_2, \dots, \alpha_k\}$ or an interval $\alpha = [\alpha_l, \alpha_u]$, representing all integers between α_l and α_u including α_l and α_u . Whenever defined, d will denote the maximum number of consecutive dots allowed in a string. In such cases, for clarity of notation, we use the extensible wild card denoted by the dash symbol ‘-’ instead of the annotated dot character, $^{[1,d]}$ in the string. Note that ‘-’ $\notin \Sigma$. Thus, a string of the form $a.^{[1,d]}b$ will be simply written as $a - b$. A motif m is extensible if it contains at least one annotated dot, otherwise m is rigid. Given an extensible string m , a rigid string m' is a realization of m if each annotated dot $^{\alpha}$ is replaced by $l \in \alpha$ dots. The collection of all such rigid realizations of m is denoted by $R(m)$. A rigid string m occurs at position l on s if $m[j] \leq s[l + j - 1]$ holds for $1 \leq j \leq |m|$. A extensible string m occurs at position l in s if there exists a realization m' of m that occurs at l . Note that an extensible string m could possibly occur a multiple number of times at a location on a sequence s . All through the discussion, we are interested mostly in the (unique) first left-most possible occurrence at each location.

For a sequence s and positive integer k , $k \leq |s|$, a string (extensible or rigid) m is a motif of s with $|m| > 1$ and location list $L_m = (l_1, l_2, \dots, l_p)$, if both $m[1]$ and $m[|m|]$ are solid and L_m is the list of at all and only the occurrences of m in s . Given a motif m let $m[j_1], m[j_2], \dots, m[j_l]$ be the l solid elements in the motif m . Then the submotifs of m are given as follows: for every j_i, j_t , the submotif $m[j_i \dots j_t]$ 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 submotifs. We are interested in motifs for which any condensation would disrupt the list of occurrences. Formally, let m_1, m_2, \dots, 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 $|L_{m_l}| = |L_{m_i}|$ and m_i is a submotif 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 L_{m_i} . A motif m_i is maximal in extension if no annotated dot character of m_i can be replaced by a fixed length substring (without annotated dot characters)

that appears in all the locations in L_{m_i} . A maximal motif is maximal in composition, in extension and in length.

In the Section 2, we derive expressions for the probabilities and expected number of occurrence of a motif under simple probabilistic models. We further derive monotonicity properties that hold for related z -scores under the fairly acceptable assumption that the probability of a motif occurrence is < 0.5 . In Section 3 we discuss our algorithm, its implementation and usage. Section 4 contains results from preliminary experiments on protein families.

2 EXPECTATIONS AND SCORES

We begin by deriving some simple expressions for the the probability p_m of an extensible motif m under stationary, iid assumptions. Let m be an extensible motif generated by a stationary, iid source which emits $\sigma \in \Sigma$ with probability p_{σ} . Consider the set $R(m)$ of all possible realizations of m . Each realization is a string over $\Sigma \cup \{.\}$. For a specific realization \bar{m} , its probability $p_{\bar{m}}$ is given by

$$p_{\bar{m}} = \prod_{\sigma \in \Sigma} (p_{\sigma})^{j_{\sigma}}, \quad (1)$$

where j_{σ} is the number of times σ appears in \bar{m} . Thus, the dot has an implicit probability of 1.

An extensible motif is degenerate if it can possibly have multiple occurrences at a site i on the input s .

LEMMA 1. *Let m be an extensible non-degenerate motif generated by a stationary, iid source which emits ($\sigma \in \Sigma$) with a probability p_{σ} . Let j_{σ} be the number of times σ appears in m and let e be the number of annotated dots in m with annotations $\alpha_1, \alpha_2, \dots, \alpha_e$. Then*

$$p_m = \prod_{\sigma \in \Sigma} (p_{\sigma})^{j_{\sigma}} \prod_{i=1}^e |\alpha_i|. \quad (2)$$

PROOF. Since the motif is non-degenerate, by the definition of realization of a motif,

$$p_m = \sum_{\bar{m} \in R(m)} (p_{\bar{m}}).$$

Hence we need to compute $p_{\bar{m}}$ where \bar{m} is a rigid motif. Assume \bar{m} is a rigid motif with no dot characters. By the iid assumption, $p_{\bar{m}} = \prod_{\sigma \in \Sigma} (p_{\sigma})^{j_{\sigma}}$. Next, consider \bar{m} to be a rigid motif with possibly some dot characters. Again, clearly, $p_{\bar{m}} = \prod_{\sigma \in \Sigma} (p_{\sigma})^{j_{\sigma}}$. In other words, only the solid characters contribute non-trivially to the computation of $p_{\bar{m}}$. Hence, if m is not rigid,

$$p_m = |R(m)| \prod_{\sigma \in \Sigma} (p_{\sigma})^{j_{\sigma}}.$$

But $|R(m)| = \prod_{i=1}^e |\alpha_i|$, hence the result.

COROLLARY 1. *If m is a non-degenerate extensible motif where each $m[i]$ is a set of (homologous) characters, then*

$$p_m = \prod_{m[i] \neq \cdot, \cdot, \cdot} \left(\sum_{\sigma \in m[i]} p_\sigma \right) \prod_{i=1}^e |\alpha_i|. \quad (3)$$

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 $a.b$, $a..b$ and $a...b$. Then $M^1 = \{a.b, a..b, a...b\}$ since m occurs once on each $m \in M^1$; $M^2 = \{a.bb, a..bb, a.b.bb\}$ since m occurs twice on each $m \in M^2$; $M^3 = \{a.bbb\}$ since m occurs three times on $m \in M^3$.

COROLLARY 2. *Let m be a degenerate (possibly with multiple occurrences at a site) extensible motif, and let $p_{m^k} = \sum_{m' \in M^{k+1}} p_{m'}$; then*

$$p_m = \sum_{k=0}^{r-1} (-1)^k (p_{m^{k+1}}). \quad (4)$$

This follows directly from the inclusion–exclusion principle.

Notice that for a degenerate motif, Equation (2) is the 0-th order approximation of Equation (4). 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 k -th order approximation of p_m is an over-estimate of p_m , if k is odd.

Next, we obtain the form of p_m for a non-degenerate motif when input m is assumed to be generated by a Markov chain. For the derivation below, we assume that the Markov chain has order 1. For further discussion, we introduce the following definition.

DEFINITION 1. (cell $\langle \sigma_1, \sigma_2, \ell \rangle$, $C(m)$). *A substring \hat{m} , on m is a cell, that begins and ends in solid characters with only non-solid intervening characters: σ_1 at the start and σ_2 at the end position, and ℓ is the number of intervening unannotated dot characters. If the intervening character is the extensible character, then ℓ takes a value of -1 . For convenience, the cell is represented by the triplet $\langle \sigma_1, \sigma_2, \ell \rangle$. $C(m)$ is the collection of all such cells of m .*

For example, $C(ab..c.d-g) = \{ \langle a, b, 0 \rangle, \langle b, c, 2 \rangle, \langle c, d, 1 \rangle, \langle d, g, -1 \rangle \}$.

Let $p_{\sigma_1, \sigma_2}^{(k)}$ denote the probability of moving from σ_1 to σ_2 in k steps. Let s be a stationary, irreducible, aperiodic Markov chain of order 1 with state space Σ ($|\Sigma| < \infty$). Furthermore, π_σ is the equilibrium probability of $\sigma \in \Sigma$ and the $(|\Sigma| \times |\Sigma|)$ transition probability matrix $P[i, j]$ is defined as $p_{\sigma_i, \sigma_j}^{(1)}$. For a rigid motif \bar{m} , for each cell $\langle \sigma_1, \sigma_2, \ell \rangle \in C(\bar{m})$ is such that $\ell \geq 0$. It is easy to see that when $\ell \geq 0$, the cell represents the $(\ell + 1)$ step-transition probability given by $P^{\ell+1}$, i.e.

$p_{\sigma_1, \ell, \sigma_2} = P^\ell[\sigma_1, \sigma_2]$. Thus, for a rigid motif \bar{m} ,

$$p_{\bar{m}} = \pi_{\bar{m}[1]} \prod_{\langle \sigma_1, \sigma_2, \ell \rangle \in C(\bar{m})} P^\ell[\sigma_1, \sigma_2].$$

We are omitting further details, and from now on, let u and v be two motifs such that v is a condensation of u , and consider an arbitrary sequence of consecutive unit expansions—each consisting of inserting a character or character set at some position, or replacing a dot character with a solid character or character set—that transforms u into v . A score z is monotonic for u and v if the value of z is always either increasing or decreasing over any such expansion. The key observation here is that, under most probabilistic settings, the probability of a condensation v of u obeys $p_v \leq p_u$. This is almost immediate under iid distribution, as shown by the following theorem.

THEOREM 1. *Let v and u be possible degenerate extensible motifs under the iid model and let v be a condensation of u . Then, there is an integer $\hat{p} \leq 1$ such that $p_v = p_u \hat{p}$.*

PROOF. It is enough to consider the case of a unit condensation, i.e. where v has one more solid character than u . The claim holds trivially when the extra character is introduced as a prefix, infix, or suffix of u . In fact, in any such case the probability of the extra character multiplies each term of Expression 4, whence the whole probability as well. Consider next the case where the solid character in v substitutes a dot of u . We begin by describing an alternate way to compute p_u . With ℓ denoting the length of a longest string in $R(u)$, compute the set of all strings over Σ^ℓ and store them consecutively row-wise in a table. Compute for each row, the probability of the string in that row, which is the product of the probabilities of the individual characters (the sum of all row probabilities is 1). Consider now the realizations in $R(u)$ in succession. Check each realization against every row of the table; wherever the two match, mark the row if it had not been already marked. Let \mathbf{R} be the set of rows that are marked at the outset. Clearly, adding up the probabilities of the rows in \mathbf{R} yields p_u . Consider now the set of rows that would be similarly involved in the computation of p_v . This must be a subset of \mathbf{R} , whence $p_v \leq p_u$.

With Markov processes, the intuition at the basis is that if we split the transition probability into two consecutive segments we have: $P^\ell[\sigma_1, \sigma_2] = \sum_{\sigma_k \in \Sigma} P^{\ell_1}[\sigma_1, \sigma_k] \times P^{\ell_2}[\sigma_k, \sigma_2]$, where $\ell = \ell_1 + \ell_2$. Since all $P^\ell[\sigma_i, \sigma_j] \geq 0$, any specific character (or alphabet subset) acting as a bottleneck yields $P^\ell[\sigma_1, \sigma_2] \leq P^{\ell_1}[\sigma_1, \sigma_k] \times P^{\ell_2}[\sigma_k, \sigma_2]$. The following general property is derived in analogy with a similar one in Apostolico et al. (2002).

THEOREM 2. *If $f(u) = f(v) > 0$, $N(v) < N(u)$ and $E(v)/N(v) \leq E(u)/N(u)$, then*

$$\frac{f(v) - E(v)}{N(v)} > \frac{f(u) - E(u)}{N(u)}.$$

PROOF. Multiplying both terms by $N(v)/E(v)$ and using the assumption $f(v) = f(u) \geq 0$, after rearrangement we get

$$\frac{f(u)}{E(v)} \left(1 - \frac{N(v)}{N(u)}\right) > 1 - \frac{E(u)N(v)}{E(v)N(u)}.$$

Since $0 < N(v)/N(u) < 1$, the left-hand side is always positive. The right-hand side is always negative or zero.

When $N(u)$ is the square root of the variance, the z -score takes up the form

$$z(u) = \frac{f(u) - E(u)}{\sqrt{\text{Var}(u)}}.$$

In the Bernoulli model, for instance, this variance results in $\sqrt{np_u(1-p_u)}$. In our case, we let p_m be the probability of the motif m occurring at any location i on the input string s with $n = |s|$ and let k_m be the observed number of times it occurs on s . When it can be assumed that the occurrence of a motif m at a site is an iid process (Waterman 1995, Chapter 12), we have for large n and $k_m \ll n$,

$$\frac{k_m - np_m}{\sqrt{np_m(1-p_m)}} \rightarrow N(0, 1). \quad (5)$$

THEOREM 3. *Let u and v be motifs generated with respective probabilities p_u and $p_v = p_u \hat{p}$ according to an iid process. If $f(u) = f(v)$ and $p_u < 0.5$ then*

$$\frac{f(v) - E(v)}{\sqrt{E(v)(1-p_v)}} > \frac{f(u) - E(u)}{\sqrt{E(u)(1-p_u)}}.$$

PROOF. We show that the functions $N(u) = \sqrt{E(u)(1-p_u)}$ and $E(u)/N(u)$ satisfy the conditions of Theorem 2. First, we prove that $E(v) < E(u)$. Indeed, since $|v| - |u|/(n - |u| + 1) > 0$,

$$\frac{E(v)}{E(u)} = \frac{(n - |v| + 1)p_v}{(n - |u| + 1)p_u} = \left(1 - \frac{|v| - |u|}{n - |u| + 1}\right) \hat{p} < \hat{p} < 1.$$

Next, we study the ratio

$$\left(\frac{N(v)}{N(u)}\right)^2 = \left(1 - \frac{|v| - |u|}{n - |u| + 1}\right) \frac{p_v(1-p_v)}{p_u(1-p_u)} < \frac{p_v(1-p_v)}{p_u(1-p_u)}.$$

The concave product $p_u(1-p_u)$ reaches its maximum for $p_u = 0.5$. Since we assume $p_u < 0.5$, the rightmost term is smaller than one. The monotonicity of $N(u)$ is satisfied.

Finally, we prove that $E(u)/N(u)$ is also monotonic, i.e. $E(v)/N(v) \leq E(u)/N(u)$, which is equivalent to

$$\frac{E(v)}{E(u)} \frac{1-p_u}{1-p_v} \leq 1,$$

but $E(v)/E(u) < 1$ by hypothesis and $(1-p_u)/(1-p_v) < 1$ since $p_u > p_v$.

In conclusion, we can restrict our z -score computation to classes of maximal motifs, i.e. compute only the z -score for the maximally saturated motif among those in each class of motifs sharing the same list of occurrences.

3 ALGORITHMIC IMPLEMENTATION

The algorithm implementing the above criteria works by iterated pairwise combination of segments of maximal extensible motifs, followed by pruning of those pairings that are not found to be viable. The input is a string s of size n and two positive integers, K and D . The extensibility parameter D is interpreted in the sense that up to D (or 1 to D) a number of dot characters between two consecutive solid characters are allowed. The output is all-maximal extensible (with D spacers) patterns that occur at least K times in s . Incidentally, the algorithm can be adapted to extract rigid motifs as a special case. It suffices to interpret D as the maximum number of dot characters between two consecutive solid characters for this adaptation.

The algorithm works by converting the input into a sequence of possibly overlapping cells (see Definition 1). A maximal extensible pattern is a sequence of cells.

3.1 Initialization phase

The cell is the smallest extensible component of a maximal pattern and the string can be viewed as a sequence of overlapping cells. If no dot characters are allowed in the motifs, the cells are non-overlapping. The initialization phase has the following steps.

Step 1: Construct patterns that have exactly two solid characters separated by no more than D spaces or ‘.’ characters. This is done by scanning the string s from left to right. Furthermore, for each location we store the start and end positions of the pattern. For example, if $s = abz dab yxd$ and $K = 2, D = 2$, then all the patterns generated at this step are: $ab, a.z, a..d, bz, b.d, b..a, zd, z.a, z..b, da, d.b, d..y, a.y, a..x, by, b.x, b..d, yx, y.d, xd$, each with its occurrence list. Thus, $L_{ab} = \{(1, 2), (5, 6)\}$, $L_{a.z} = \{(1, 3)\}$ and so on.

Step 2: The extensible cells are constructed by combining all the cells with at least one dot character and the same start and end solid characters. The location list is updated to reflect the start and end positions of each occurrence. Continuing with the previous example, $b-d$ is generated at this step with $L_{b-d} = \{(2, 4), (6, 9)\}$. All cells m with $|L_m| < K$ are discarded. In the example, the only surviving cells are $ab, b-d$ with $L_{ab} = \{(1, 2), (5, 6)\}$ and $L_{b-d} = \{(2, 4), (6, 9)\}$.

3.2 Iteration phase

Let B be the collection of cells. If $m = \text{Extract}(B)$, then $m \in B$ and there does not exist $m' \in B$ such that $m' > m$ holds: $m_1 > m_2$ if one of the following holds. (1) m_1 has only solid characters and m_2 has at least one non-solid character. (2) m_2 has the ‘-’ character and m_1 does not. (3) m_1 and m_2 have $d_1, d_2 > 0$ dot characters and $d_1 < d_2$.

Furthermore, m_1 is \sim -compatible with m_2 if the last solid character of m_1 is the same as the first solid character of m_2 . Moreover, if m_1 is \sim -compatible with m_2 , $m = m_1 \sim m_2$ is the concatenation of m_1 and m_2 with an overlap at the common end and start character and $L'_m = \{((x, y), z) | ((x, l), z) \in L'_{m_1}, ((l, y), z) \in L'_{m_2}\}$. For example, if $m_1 = ab$ and $m_2 = b.d$ then m_1 is \sim -compatible with m_2 and $m_1 \sim m_2 = ab.d$. However, m_2 is not \sim -compatible with m_1 .

The procedure is best described by the pseudocode shown here. `NodeInconsistent(m)` is a routine that checks if the new motif m is non-maximal with respect to the earlier non-ancestral nodes by checking the location lists. Steps G:18–19 detect the suffix motifs of already detected maximal motifs. Result is the collection of all the maximal extensible patterns.

```

Main()
{
  Result ← {};
  B ← {mi | mi is a cell};
  For each m = Extract(B)
    Iterate(m, B, Result);
}
Iterate(m, B, Result)
{
G:1  m' ← m;
G:2  For each b = Extract(B) with
G:3    ((b  $\sim$ -compatible m') OR (m'  $\sim$ -compatible b))
G:4    If (m'  $\sim$ -compatible b)
G:5      mi ← m' ~ b;
G:6      If NodeInconsistent(mi) exit;
G:7      If (|Lm'| = |Lb|) B ← B - {b};
G:8      If (|Lm'| ≥ K)
G:9        m' ← mi;
G:10       Iterate(m', B, Result);
G:11      If (b  $\sim$ -compatible m')
G:12        mi ← b ~ m';
G:13        If NodeInconsistent(mi) exit;
G:14        If (|Lm'| = |Lb|) B ← B - {b};
G:15        If (|Lm'| ≥ K)
G:16          m' ← mi;
G:17          Iterate(m', B, Result);
G:18      For each r ∈ Result with Lr = Lm'
G:19        If (m' is not maximal w.r.t. r) return;
G:20      Result ← Result ∪ {m'};
}

```

The correctness follows from the observation that the above procedure essentially constructs the inexact suffix tree of Chattaraj and Parida (2005) implicitly, in a different order. A tight time complexity is more difficult to come by, however, if we consider M to be the number of extensible maximal motifs and S to be the size of the output, i.e. the sum of the sizes of the motifs and the sizes of the corresponding location lists, the time taken by the algorithm is $O(SM \log M)$.

In experiments of the kind described later in the paper, at 3-GHz clock, time ranged typically from few minutes to half an hour.

3.3 Varun implementation and usage

In this section we give some details of using Varun,³ an implementation of the discovery process of the extensible patterns with combinatorial and statistical pruning. This software will be available for use with the suite of tools at www.research.ibm.com/bioinformatics; all user-specific details appear here.

Since the pattern space can vary dramatically for different classes of inputs, a number of parameters have been introduced to allow the user exploit his specific domain knowledge maximally. One way of viewing this control is to prune the pattern space appropriately and various parameters are specified to meet this objective. There are essentially two classes of pruning parameters: (1) combinatorial pruning and (2) statistical pruning. To avoid clutter, we describe only a few of the critical pruning parameters here. Each parameter has a default value and it is not mandatory to specify all of them.

3.3.1 Combinatorial pruning Some of the combinatorial pruning parameters are

- (1) Pruning by occurrences.
 - (a) `-k<Num>`: Num is the quorum or the minimum number of times a pattern must occur in the input.
 - (b) `-c`: When this is specified the quorum k is in terms of the number of sequences where the pattern occurs at least once. For example, if this option is set and furthermore, `-k10` is specified, a valid pattern must occur in at least 10 distinct sequences. However if this option is not set, a valid pattern must have at least 10 occurrences, not necessarily in distinct sequences.
- (2) Pruning by composition.
 - (a) Using homology groups.
 - (i) `-b<File>`: File lists the symbol equivalences that define the homology groups. The default file is an empty file.
 - (ii) `n<Num>`: Num is the maximum number of bracketed elements (equivalence classes) in a pattern. For example, if `'-n2'` is specified, `[IL]...[LV]`, `L.[LV] - V` are valid patterns but not `[LV][IL][LV]..L`.
 - (b) `-R`: When this mode is specified, only rigid patterns are discovered.
 - (c) Extensibility: The following two parameters are used to prune the space of extensible patterns.

³A character from Indian mythology who is thousand eyed and sees all that happens in the world.

Figure 1 shows an example of the size of the pattern space for different parameter values.

- (i) $-D<Num>$: Num is the maximum number of consecutive dot characters (‘.’) in the realization of an extensible pattern. Note that a dot character and an extensible character are never consecutive in any valid pattern. For example, if ‘ $-D3$ ’ is specified, then $L...V$, LV , $L.L.V$ are valid patterns but not $L....L$. Furthermore, an extensible pattern of the form $L - V$ implies that there are 1–3 dot characters in the occurrences of this pattern between the bases L and V .
- (ii) $-d<Num>$: Num is the minimum number of non-extensible characters (including the dot character) between two consecutive extensible characters (‘-’). For example, if ‘ $-d4$ ’ is specified, then $L..H - L..H - L$ is a valid pattern but not $L...H - L.H - L$.

3.3.2 *Statistical pruning* In this parameter,

- (1) $-p<File>$: File lists the symbol probabilities used for the probabilistic analysis.
- (2) $-z<Val>$: Val is the minimum absolute value of z-score of the patterns.

3.4 Information display

- (1) *Displaying occurrence information.* The different modes of displaying the occurrence list of each valid pattern are as follows. (a) The occurrence list is not displayed (option $-L0$). (b) Only the start position of each occurrence is displayed (option $-L1$). (c) The start and end positions of each occurrence is displayed as $x_1 - x_2$ where x_1 is the starting position and x_2 the end position (option $-L4$).
- (2) *Displaying statistical information.* The different statistical information displayed for possible use are (Section 2) (a) the probability of occurrence of a pattern, (b) the observed number of occurrences and (c) the z-score. Figure 1 shows an example.

4 RESULTS FROM PRELIMINARY EXPERIMENTS

We tested Varun on six protein families by seeking the surprising motifs in each. Each family was picked at random from the PROSITE database.

- (1) *High potential iron–sulfur proteins (HiPIP) (id PS00596).* This is a specific class of high redox potential 4Fe–4S ferredoxins that function in anaerobic electron transport and occur in photosynthetic bacteria and in *Paracoccus denitrificans* (Breiter *et al.*, 1991).

Pattern	Probability	Occ.	Z-Score
[LIVP]-[LM]R.[GE][LIVP].GC	2.05647e-07	57	585.494
LR.[GE][LIVP].GC	2.53136e-07	63	582.758
L.[GE][LIVP].GC	4.77614e-06	70	148.626
R-[GE][LIVP].GC	6.33367e-06	66	121.48
L-[GE][LIVP].GC	1.43284e-05	83	101.21
G[LIVP][GE].GC	3.98344e-05	77	55.359
R-[LIVP].GC	4.68467e-05	65	42.6968
L-[LIVP].GC	0.00010598	112	48.3873

Fig. 1. A statistical summary of a small set of valid patterns on the coagulation factors 5/8 type C domain, also used in Figure 7.

Rank	z-score	Motif
1	1497,62	C-(6,7,8,9)[LIVM]...G[YWC]..[FYW]
2	978,872	P-(3,4,6,8,9)[LIVM]...G[YWC]..[FYW]
3	590,866	C-(6,7,8,9)[LIVM]...G[YWC]-(1,3,4,5,6,7)A
4	564,821	C-(6,7,8,9)[LIVM]...G[YWC]-(1,3,4,5,6,7)[A TD]
5	537,73	[LIVM]-(1,2,3,4,5,7,8,9)G[YWC]..[FYW]
6	385,2	[LIVM]-(1,2,3,4,5,7,8,9)G[YWC]..[FYW]
7	161,173	[LIVM]...G[YWC]-(2,4)[FYW]
8	156,184	[LIVM]-(1,2,3,4,5,6,7,8,9)G[YWC]
9	138,881	[LIVM]-(1,3,4,5,6)[LIVM]...G[YWC]-(1,3,4,5,6,7)A

Fig. 2. The functionally relevant motif is shown in bold for high potential iron–sulfur proteins (HiPIP) (id PS00596). Here 22 sequences of ~2500 bases were analyzed at $k = 22$, $D = 9$, $d = 4$.

Rank	z-score	Motif
1	7,60E+07	RA.T[LV].C.P-(2,3)G.HP....AC[ATD].L....[ASG]
2	21416,8	A..[LV].C.P-(2,3)G.HP-(1,2,4)[ASG].[ATD]
3	8105,33	A-(1,4)T...P-(2,3)G.HP...[ATD]-(3)L....[ASG]
4	5841,85	[ATD].T...P-(1,2,3)G.HP-(1,2,4)A.[ATD]
5	4707,62	P.[ASG]-(2,3,4)P....AC[ATD].L....[ASG]
6	4409,21	A..[LV]...P-(2,3)G.HP-(1,2,4)A.[ATD]
7	3086,17	P-(1,2,3)[ASG].P-(4)AC[ATD].L....[ASG]
8	3068,18	R..[ATD]...P-(2,3)G.HP-(1,2,4)[ASG].[ATD]
9	2615,98	[ASG][ATD]-(1,3,4)P....AC[ATD].L....[ASG]
10	2569,66	[ASG]-(1,2,3,4)P....AC[ATD].L....[ASG]
11	2145,6	G-(2,3)P....AC[ATD].L....[ASG]

Fig. 3. The functionally relevant motif is shown in bold for *Streptomyces subtilisin*-type inhibitors signature (id PS00999). Here 20 sequences of ~2500 bases were analyzed at $k = 20$, $D = 4$, $d = 4$.

Two of the cysteine residues of the motif shown in Figure 2 are involved in binding to the iron–sulfur cluster. This is the top ranking motif discovered by Varun out of the possible 273 extensible motifs.

- (2) *Streptomyces subtilisin*-type inhibitors (id PS00999). Bacteria of the *Streptomyces* family produce a family of proteinase inhibitors characterized by their strong activity toward subtilisin. They are collectively known as streptomyces subtilisin inhibitors (SSIs). Varun discovers this functionally significant motif as the top ranking one out of 470 extensible motifs (Fig. 3).
- (3) *Nickel-dependent hydrogenases (id PS00508)*. These are enzymes that catalyze the reversible activation of hydrogen and are further involved in the binding of

Rank	z-score	Motif
1	295840	[LIM]-(1,2,3,4)[STA][FY]DPC[LIM][ASG]C[ASG].H
2	2,86E+05	[LIM]-(1,2,3,4)[ASG][FY]DPC[LIM][ASG]C[ASG].H
3	155736	R-(1,4)[FY]DPC[LIM][ASG]C[ASG].H
4	78829	[LIM]-(1,2,3,4)[STA].DPC[LIM][ASG]C[ASG].H
5	76101,9	[LIM]-(1,2,3,4)[ASG].DPC[LIM][ASG]C[ASG].H
6	34205,6	[STA]-(1,4)DPC[LIM][ASG]C[ASG].H
7	30325,1	[LIM]-(1,2,3,4)[STA][FY]D.C[LIM][ASG]C..H
8	29276	[LIM]-(1,2,3,4)[ASG][FY]D.C[LIM][ASG]C..H
9	20527,3	[ASG]-(1,4)DPC[LIM][ASG]C[ASG].H
10	17503,4	[LIM]-(1,2,3,4)[ASG]..PC[LIM][ASG]C[ASG].H

Fig. 4. The functionally relevant motifs are shown in bold for Nickel-dependent hydrogenases (id PS00508). Here 22 sequences of ~23 000 bases were analyzed at $k = 22$, $D = 4$, $d = 3$.

Rank	z-score	Motif
1	2,84E+09	Y...L...C.[FYW]A...[STAH]R..P.FNE[STAH]K.I.F[STAH]M
2	8,28E+07	V(1,3,4)G...S.[STAH]...N...L...Q(4)[STAH]...L.[DN]...[FYW]..F...P...Q...A...I
3	5,55E+07	L(2,3)F...Q...[STAH][STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
4	4,27E+07	L(2,3)F...Q...[STAH][STAH][STAH]...S...[FYW]..F.R..PD..Q...A...I
5	4,23E+07	L...L...[STAH]...[STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
6	3,99E+07	LF(3)Q...[STAH][STAH]...[STAH]...[FYW]..F.R..PD..Q...A...I
7	3,38E+07	LF(3)Q...[STAH][STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
8	3,38E+07	LF...Q...[STAH](4)L.[DN]...[FYW]..F.R..PD..Q[STAH]A...I
9	3,29E+07	I(1)Q[STAH]...[STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
10	3,29E+07	IQ(4)[STAH]...L.[DN]...[FYW]..F.R..PD..Q[STAH]A...I
11	3,29E+07	IQ[STAH]...[STAH](4)L.[DN]...[FYW]..F.R..PD..Q...A...I
12	3,10E+07	L...Q(1,4)[STAH]...[STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
13	2,77E+07	L[FYW](3)Q[STAH]...[STAH]...L...[FYW]..F.R..PD..Q...A...I
14	2,58E+07	L(4)Q[STAH]...[STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
15	2,30E+07	S[STAH]S(2,4)L.[DN]...[FYW]..F.R..PD..Q[STAH]A...I
16	2,15E+07	L(1,3,4)C...[FYW]A...[STAH]R..P.F.E.K.I.F.M
17	1,40E+07	F(1)I.Q...[STAH][STAH](4)L[STAH]...[FYW]..F.R..PD..Q...A...I
18	1,37E+07	L(2,4)I...[STAH][STAH][STAH](3)LS...[FYW]..F.R..PD..Q...A...I
19	1,02E+07	L..I(1)Q...[STAH][STAH]...S...[FYW]..F.R..PD..Q...A...I
20	8,65E+06	I(1)Q...[STAH][STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
21	8,19E+06	S[STAH]-(1,2,3,4)L.[DN]...[FYW]..F.R..PD..Q[STAH]A...I
22	7,98E+06	Q(3)[STAH][STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
23	6,82E+06	F(3)Q...[STAH][STAH]...L[STAH]...[FYW]..F.R..PD..Q...A...I
24	5,66E+06	A[STAH][STAH](2,3)L.[DN]...[FYW]..F.R..PD..Q...A...I
25	5,57E+06	FI(3)[STAH]...[STAH]...L[STAH]...[FYW]..F.R..PD..Q...A...I
26	5,18E+06	LL(4)Q...[STAH]...L(1)[DN]...[FYW]..F.R..PD..Q...A...I
27	3,61E+06	LL(2)I...[STAH]...[STAH]...[FYW]..F.R..PD..Q...A...I
28	3,48E+06	[STAH][STAH]-(1,2,3)L.[DN]...[FYW]..F.R..PD..Q...A...I
29	3,17E+06	[STAH]...[STAH]...L.[DN]...[FYW]..F.R..PD..Q...A...I
30	2,47E+06	L...Q(4)[STAH][STAH]...S...[FYW]..F.R..PD..Q...A...I
31	2,43E+06	V(1,3)N.L...L(3)[STAH]...[STAH]...[STAH]...[FYW]..F...PD..Q...A...I
32	2,22E+06	[STAH][STAH][STAH]-(1,2,3)LS...[FYW]..F.R..PD..Q...A...I
33	2,06E+06	[STAH][STAH][STAH]...L...[FYW]..F.R..PD..Q...A...I
34	2,03E+06	Y...L...C...A...R..P.F.E.K.I(1,4)[FYW][STAH]
35	1,99E+06	IQ...[STAH](1)[STAH]...L.[DN]...[FYW]..F...PD..Q...A...I
36	1,99E+06	IQ(1)[STAH]...[STAH]...L.[DN]...[FYW]..F...PD..Q...A...I
38	1,97E+06	FI...[STAH](3)[STAH]...L.[DN]...[FYW]..F...PD..Q...A...I
40	1,97E+06	FI(3)[STAH]...[STAH]...L.[DN]...[FYW]..F...PD..Q...A...I
41	1,91E+06	[STAH][STAH]K(1,4)P.FNE[STAH]K.I.F[STAH]M
42	1,72E+06	CC[FYW].C.C...[FYW]-(2,4)[DN]..[STAH]C..C
43	1,57E+06	[STAH]-(1,3,4)[FYW]A...[STAH]R..P.F.E.K.I.F.M
44	1,49E+06	A-(1,3)[STAH]...L[STAH][DN]...[FYW]..F.R..PD..Q...A...I
45	1,36E+06	Q...[STAH][STAH](3)L[STAH]...[FYW]..F.R..PD..Q...A...I
46	1,32E+06	I(3)[STAH][STAH][STAH]...S...[FYW]..F.R..PD..Q...A...I
47	1,31E+06	[STAH][STAH]-(1,2,3,4)L.[DN]...[FYW]..F.R..PD..Q...A...I
48	1,24E+06	[STAH]...[STAH][STAH]-(1,3)LS...[FYW]..F.R..PD..Q...A...I
49	1,19E+06	[FYW]-(1,3,4)[STAH]...P.FNE[STAH]K.I.F[STAH]M
50	1,12E+06	L...[STAH](3)[STAH]...L[STAH]...[FYW]..F.R..PD..Q...A...I

Fig. 5. The functionally relevant motif is shown in bold for G-protein coupled receptors family 3 (id PS00980). This run involved 25 sequences of ~25 000 bases each at $k = 25$, $D = 4$, $d = 8$.

nickel. Again, this functionally significant motif is detected among the top three by Varun out of 4150 extensible motifs (Fig. 4).

- (4) *G-protein coupled receptor family 3 (id PS00980)*. Varun finds that the most important structural motif in this family is among the top 30 of the motifs out of 3508 extensible motifs (Fig. 5).

Rank	z-score	Motif
1	5,42E+06	C-(4,5)CCS..G[FYW]CG....[FYW]C
2	1,73E+06	C-(4,5)CCS..G[FYW]CG.....C
3	1,70E+06	C-(4,5)CCS..G.CG....[FYW]C
4	1,56E+06	CCS..G[FYW]CG....[FYW]C
5	544162	C-(4,5)CCS..G.CG....C
6	4,95E+05	CCS..G[FYW]CG....C
7	488261	CCS..G.CG....[FYW]C
8	155706	CCS..G.CG....C
9	104666	C-(4,5)C.S..[GASL][FYW]CG.....C
10	84133,4	C.....C-(3,4)[GASL][FYW]CG....[FYW]C
11	56078	C.....C-(3,4)G.CG....[FYW]C

Fig. 6. The functionally relevant motif is shown in bold for Chitin recognition (id PS00026). Here 53 sequences of ~13 823 bases were analyzed at $k = 53$, $D = 5$, $d = 10$.

Rank	z-score	Motif
1	969,563	P-(4,5,8,9,10)[LM]R.[GE][LIVP].GC
2	694,1	P-(4,5,8,9,10)[LM]R.[GE][LIVP].[GE]C
3	370,594	[LIVP]-(1,3,4,5,6,7,8,9,10)[LM]R.[GE].[GE]C
4	361,052	P-(4,5,8,9,10)[LM]R.[GE].[GE]C
5	261,519	[LIVP]-(1,3,4,5,6,7,8,9,10)[LM]R.[GE][LIVP].C
6	261,519	[LIVP]-(1,3,4,5,6,7,8,9,10)[LM]R..[LIVP].[GE]C
7	254,971	P-(4,5,8,9,10)[LM]R.[GE][LIVP].C
8	254,971	P-(4,5,8,9,10)[LM]R..[LIVP].[GE]C
9	249,763	[LIVP].....[LIVP]-(1,2,4,5,6,7,8,9,10)R.[GE]..GC

Fig. 7. The functionally relevant motif is shown in bold for Coagulation factors 5/8 type C domain (id PS01286). Here 40 sequences of ~80 290 bases were analyzed. Notice that in this case, the motifs have a fairly large gap size of 10 bases at $k = 40$, $D = 10$, $d = 10$.

- (5) *Chitin-binding type-1 domain (id PS00026)*. Varun finds that the most important structural motif in this family is one of the top two of the motifs out of 886 extensible motifs (Fig. 6).
- (6) *Coagulation factors 5/8 type C domain (FA58C) (id PS01286)*. Varun finds that the most important structural and functional motif in this family is one of the top two of the motifs out of 80290 extensible motifs (Fig. 7).

To summarize, we find that in almost all cases, the motif documented as the most important (as functionally/structurally relevant motif) in PROSITE is in the top extensible motifs returned by Varun as surprising. In the fourth set (Fig. 5) we find the PROSITE motif at position 42, shows that in some particular cases the patterns reported by Varun can be grouped together; in fact, the top scoring motifs are very close to each other in location and in composition. This reveals that a post processing step that clusters together the top patterns can only improve the goodness of the results. In all cases, the difference in the z -score between the top few and the rest is dramatic as can be seen in Figures 2–7 (Table 1). The differing values of the z -scores of each family is attributed to the different sizes

AQ: Please check placement of Table 1.

Table 1. Number of patterns in the experiment in Figure 7 with z -score ≥ 100.0 at various values of parameters D and d with quorum $k = 53$

	D			
	2	3	4	5
d				
3	121	196	370	1145
4	121	194	355	1008
5	114	182	326	891
8	112	178	313	758
10	112	178	313	727

of the the families (the number of members and the length of each member).

Next, we test the sensitivity and selectivity of Varun using the families as reported in PROSITE. Since most of the family sizes are small, we do these experiments along the lines of Wang *et al.* (1999, p. 46). The following six sets were selected randomly from each family: five sequences from each of the families, high potential iron–sulfur proteins, streptomycetes subtilisin-type inhibitors, nickel-dependent hydrogenases, G-protein coupled receptors family 3 and coagulation factors 5/8 type C domain, and eight sequences from the family of chitin-binding type-1 domain.

First, each family was contaminated with one of the sets that was drawn from a different family (e.g. the five sequences of G-protein was mixed with the family of the hydrogenases). Next, we contaminated each family with two sets from a different family and then subsequently three sets. In each of the experiments we found that the top ranked motifs were exactly as reported in Figures 2–7.

5 CONCLUSION AND FUTURE DIRECTIONS

The extensibility of a motif not only leads to a succinct description but also helps capture function and/or structure in a single pattern, which would be not possible through a rigid description (see case studies in Section 4). At the same time, with extensible motifs the number of candidates to be considered increases dramatically. Our characterization of a pattern rigidly conjugates structure and set of occurrences. This results in a definition of motif that lends itself to a natural notion of maximality, thereby embodying statistics and structure in one measure of surprise. This is unlike most previous approaches, that consider structure and statistics as separate features of a pattern. It leads here to a powerful syntactic mechanism for eliminating unimportant motifs before their score is computed. We show in this paper that for the class of over-represented motifs, the non-maximal motifs are not more surprising than the maximal motifs. The usefulness of the statistical measures resulting from this combination of ideas is demonstrated on a small set of families of proteins.

The results, though preliminary, look very promising. More advanced probabilistic frameworks are worthy of investigation. We are also currently working on the task of unsupervised discovery over the entire database to gauge suitable specificity and sensitivity parameters.

ACKNOWLEDGEMENTS

Work by A.A. was supported in part by the Italian Ministry of University and Research under the National Projects FIRB RBNE01KNFP, PRIN ‘Combinatorial and Algorithmic Methods for Pattern Discovery in Biosequences’ and by the Research Program of the University of Padova. Work was done by M.C. during his internship at IBM Thomas J. Watson Research Center. We are very grateful to Abhijit Chattaraj for his strong contributions to the code in the initial phase of the development.

REFERENCES

- Apostolico,A., Bock,M.E. and Lonardi,S. (2002) Monotony of surprise and large scale quest for unusual words. *J. Comput. Biol.*, **10**(3–4), 283–311.
- Apostolico,A. and Parida,L. (2004) Incremental paradigms for motif discovery. *J. Comput. Biol.*, **11**(1), 15–25.
- Apostolico,A. and Pizzi,C. (2004) Monotone scoring of patterns with mismatches. In *Proceedings of the 4th Workshop on Algorithms in Bioinformatics*, 17–21 September, Bergen, Norway. Lecture Notes in Computer Science, Vol. 3240, Springer, Berlin, pp. 87–98.
- Blumer,A., Blumer,J., Ehrenfeucht,A., Haussler,D., Chen,M.T. and Seiferas,J. (1985) The smallest automaton recognizing the subwords of a text. *Theoret. Comput. Sci.*, **40**, 31–55.
- Breiter,D.R., Meyer,T.E., Rayment,I. and Holden,H.M. (1991) The molecular structure of the high potential iron–sulfur protein isolated from *Ectothiorhodospira halophila* determined at 2.5-Å resolution. *J. Bio. Chem.*, **266**, 18660–18667.
- Chattaraj,A. and Parida,L. (2005) An inexact suffix tree based algorithm for extensible pattern discovery. *Theoret. Comput. Sci.*, **335**: 3–14.
- Hertz,G.Z. and Stormo,G.D. (1999) Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, **15**, 563–577.
- Keich,U. and Pevzner,P.A. (2002) Finding motifs in the twilight zone. In *Proceedings of the 6th Annual International Conference on Computational Molecular Biology*, April 2002, Washington, DC, pp. 195–204.
- Lawrence,C.E., Altschul,S.F., Boguski,M.S., Liu,J.S., Neuwald,A.F. and Wootton,J.C. (1993) Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, **262**, 208–214.
- Leung,M.Y., Marsh,G.M. and Speed,T.P. (1996) Over and under-representation of short DNA words in herpesvirus genomes. *J. Comput. Biol.*, **3**, 345–360.
- Pevzner,P.A. and Sze,S.-H. (2000) Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, AAAI Press, pp. 269–278.

- Taguchi,S., Kojima,S., Terabe,M., Miura,K.I. and Momose,H. (1994) Comparative studies on the primary structures and inhibitory properties of subtilisin–trypsin inhibitors from streptomyces. *Eur. J. Biochem.*, **220**, 911–918.
- Volbeda,A., Charon,M.H., Piras,C., Hatchikian,E.C., Frey,M. and Fontecilla-Camps,J.C. (1995) Crystal structure of the nickel–iron hydrogenase from *Desulfovibrio giges*. *Nature*, **373**, 580–587.
- Wang,J.T.L., Shapiro,B.A. and Shasha,D. (1999) *Pattern Discovery in Biomolecular Data*. Oxford University Press, Oxford.
- Waterman,M.S. (1995) *An Introduction to Computational Biology: Maps, Sequences and Genomes*. Chapman Hall, New York.