# Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm 

Isidore Rigoutsos ${ }^{1}$ and Aris Floratos ${ }^{1,2}$<br>${ }^{1}$ Bioinformatics and Pattern Discovery Group, Computational Biology Center, IBM Thomas J. Watson Research Center, PO Box 704, York Town Heights, NY 10598 and ${ }^{2}$ Courant Institute of Mathematical Sciences, New York University, New York, NY 10012, USA

Received on November 11, 1997; revised and accepted on December 15, 1997


#### Abstract

Motivation: The discovery of motifs in biological sequences is an important problem. Results: This paper presents a new algorithm for the discovery of rigid patterns (motifs) in biological sequences. Our method is combinatorial in nature and able to produce all patterns that appear in at least a (user-defined) minimum number of sequences, yet it manages to be very efficient by avoiding the enumeration of the entire pattern space. Furthermore, the reported patterns are maximal: any reported pattern cannot be made more specific and still keep on appearing at the exact same positions within the input sequences. The effectiveness of the proposed approach is showcased on a number of test cases which aim to: (i) validate the approach through the discovery of previously reported patterns; (ii) demonstrate the capability to identify automatically highly selective patterns particular to the sequences under consideration. Finally, experimental analysis indicates that the algorithm is output sensitive, i.e. its running time is quasi-linear to the size of the generated output.


Contact: rigoutso@us.ibm.com

## Introduction

One of the problems arising in the analysis of biological sequences is the discovery of sequence similarity in the primary structure of related proteins or genes. Such similarity usually corresponds to residues conserved during evolution due to an important structural or functional role.
Several methods have been proposed for dealing with this problem. One widely used class of algorithms (Needleman and Wunsch, 1970; Delcoigne and Hansen, 1975; Waterman et al., 1984; Sobel and Martinez, 1986; Martinez, 1988; Smith and Smith, 1990; Wu and Brutlag, 1995; NevilleManning et al., 1997) employs global string alignment (Carrillo and Lipman, 1988); in this context, edit operations (e.g. mutations, insertions, deletions), along with their associated costs, are used for transforming one sequence to another. What is sought is a minimum-cost consensus sequence that
highlights the regions of similarity among the input sequences. A detailed survey of several multiple-string alignment algorithms can be found in Hirosawa et al. (1995).

Alignment algorithms suffer from several inherent drawbacks. First, the task of optimally aligning a set of strings is computationally very expensive [it is known to be an NPhard problem (Aho et al., 1974; Wang and Jiang, 1994)]. Second, alignment of entire sequences can reveal only global similarities (Smith et al., 1990; Wang et al., 1994); if the sequences under comparison are distantly related or if the relative order of their similar regions varies among sequences (domain swapping), it is quite possible that no substantial alignment can be produced. In such situations, traces of evolutionary relationships might only be detectable as similarities over short stretches of residues.

One way to overcome the difficulty that alignment algorithms have in identifying local similarities is to focus on the discovery of patterns shared by the input sequences. A pattern is a formal way to define the notion of local similarity. As an example, consider the alphabet of the 20 amino acids; in this context 'A.CH' is a valid pattern, describing all oligopeptides that start with an alanine, have an arbitrary amino acid in the second position and end with a cysteine followed by a histidine. A protein matches ' $A . C H$ ' if it contains at least one peptide stretch that is described by this pattern. The assumption behind pattern discovery approaches is that a pattern that appears often enough in a set of biological sequences is expected to play a role in defining the respective sequences' functional behavior and/or evolutionary relationships.

A number of pattern-discovery algorithms have been steadily appearing in the literature over the past few years (Smith and Waterman, 1981; Martinez, 1983; Smith et al., 1990; Roytberg, 1992; Benson and Waterman, 1994; Neuwald and Green, 1994; Wang et al., 1994; Jonassen et al., 1995; Sagot et al., 1995; Suyama et al., 1995; Guan and Uberbacher, 1996; Sagot and Viari, 1996). The pattern types that these algorithms are able to handle range from simple strings to quite general regular expressions. A survey of
several of these algorithms can be found in Brazma et al. (1995). Most of the proposed approaches proceed by enumerating the solution space, i.e. they generate all (or most of) the possible patterns and then verify, for each one, that it has sufficient support (i.e. it appears in sufficiently many input sequences-the exact number is usually provided by the user). Several considerations also arise here. First, unless the nature of patterns sought is extremely simple, the problem of detecting all existing patterns is NP-hard (Garey and Johnson, 1979); typically, a reduction from the longest common subsequence problem (Maier, 1978) can be used to prove this. Possible remedies include the use of heuristics (Neuwald and Green, 1994; Wang et al., 1994; Jonassen et al., 1995; Sagot and Viari, 1996) which offer enhanced performance (but frequently at the expense of sacrificing the completeness of the results), and/or the structural restriction of the patterns sought (e.g. most algorithms restrict the maximum length that a pattern can have). A second problem, usually overlooked, has to do with the quality of the reported results. In order to help the human expert (who is usually the recipient of the output) make sense of the results, it is desirable to produce only maximal patterns. For example, if ' $A . C E$ ' is a pattern appearing in a given number of positions within the input, then it makes no sense reporting the pattern ' $A . . E$ ' if this second pattern appears at the exact same positions and nowhere else. In other words, a reported pattern must be as specific as possible.
In this discussion, we present TEIRESIAS, a novel algorithm for the discovery of patterns in biological sequences. Our algorithm is capable of detecting and reporting all existing patterns in a set of input sequences without enumerating the entire solution space and without using pairwise alignments. These properties allow for enhanced performance. Furthermore, the patterns reported are guaranteed to be maximal. The utility of the algorithm is demonstrated on several examples and its performance is evaluated experimentally through its application to a wide range of synthetic data.

## Methods

TEIRESIAS operates in two phases: scanning and convolution. During the scanning phase, elementary patterns with sufficient support are identified. These elementary patterns constitute the building blocks for the convolution phase. They are combined into progressively larger and larger patterns until all the existing, maximal patterns have been generated. Furthermore, the order in which the convolutions are performed makes it easy to identify and discard non-maximal patterns.

## Terminology and problem statement

Let $\Sigma$ be the alphabet of residues at hand (e.g. the set of all amino acids). A pattern is defined to be any string of the form:

$$
\begin{equation*}
\sum\left(\sum\left\{\prime^{\prime} .^{\prime}\right\} * \sum\right. \tag{1}
\end{equation*}
$$

i.e. any string that begins and ends with a residue, and contains an arbitrary combination of residues and '.' characters. The '.' (referred to as the 'don't care character') is used to denote a position that can be occupied by an arbitrary residue. The following is an example of a valid pattern over the alphabet of amino acids:

## 'A.CH..E'

A pattern $P$ is really a regular expression and as such it defines a language $\boldsymbol{G}(P)$. The elements of the language are all the strings that can be obtained from $P$ by substituting each don't care by an arbitrary residue from $\Sigma$. For the pattern 'A.CH..E', for example, the following peptides are elements of $\boldsymbol{G}(‘ A . C H . . E ’)$ :

## ADCHFFE, ALCHESE, AGCHADE

For any pattern $P$, any substring of $P$ that is itself a pattern is called a subpattern of $P$. For example, ' $H . . E$ ' is a subpattern of the pattern 'A.CH..E'. A pattern $P$ is called an $<L, W\rangle$ pattern (with $L \leq W$ ) if every subpattern of $P$ with length $W$ or more contains at least $L$ residues. Notice that every $\langle L, W\rangle$ pattern is also an $<L, W+1>$ pattern, an $<L, W+2>$ pattern, and so on.

A string of residues over $\Sigma$ is said to match a given pattern $P$ if it contains at least one substring (i.e. a block of consecutive residues) that belongs in $\boldsymbol{G}(P)$. For example, each sequence in the following set matches the pattern 'A.CH..E' (the boldfaced strings indicate the particular matching substrings in each sequence):

## $S=\{$ LFAADCHFFEDTR, LKLALCHESESDR, AFAGCHADELFT $\}$

Given a pattern $P$ and a set of sequences $S=\left\{s_{1}, s_{2}, \ldots, s_{n}\right\}$, we define the offset list of $P$ with respect to $S$ (or simply the offset list of $P$, when $S$ is unambiguously implied) to be the following set:

$$
L_{S}(P)=\left\{(i, j) \mid \text { sequence } s_{i} \text { matches } P \text { at offset } j\right\}
$$

For the particular $S$ given above, for example, the offset list of the pattern 'A.CH..E' is:

$$
L_{S}\left({ }^{\prime} A . C H . . E^{\prime}\right)=\{(1,4),(2,4),(3,3)\}
$$

A pattern $P^{\prime}$ is said to be more specific than a pattern $P$ if $P^{\prime}$ can be obtained from $P$ by changing one or more don't care characters to residues, or by appending an arbitrary string of residues and don't cares to the left or/and right of $P$.

The following patterns are all more specific than the pattern 'A.CH..E':
'AFCH..E', 'A.CHL.E.K', 'SA.CH..E', 'SA.CH..EF..KL'
Notice that if a pattern $P^{\prime}$ is more specific than a pattern $P$, then for every set $S$ of input sequences:

$$
\left|L_{S}\left(P^{\prime}\right)\right| \leq\left|L_{S}(P)\right|
$$

Given a set of sequences $S$, a pattern $P$ is called maximal with respect to $S$ if there exists no pattern $P$ which is more specific than $P$ and such that $|L S(P)|=|L S(P)|$. In other words, if $P$ is a maximal pattern, then it cannot be made more specific without simultaneously reducing the size of its offset list.

Using the above definitions, we can succinctly describe the problem addressed by TEIRESIAS as follows:
Problem definition: Given a set $S=\left\{s_{1}, s_{2}, \ldots, s_{n}\right\}$ of input sequences and parameters $L, W, K$, find all maximal $<L, W>$ patterns that have support at least $K$ (i.e. they appear in at least $K$ distinct sequences in $S$ ).
In what follows, the letters $L, W$ and $K$ are used to denote the parameters specified in the problem definition given above, and are assumed to have values which have been provided by the user. Furthermore, when the term 'pattern' is used without further qualification, it always implies an $<L$, $W>$ pattern.

## The algorithm

TEIRESIAS begins by scanning the sequences in the input set $S$ and locating all elementary patterns with support at least $K$. An elementary pattern is just a $\langle L, W>$ pattern containing exactly $L$ residues. Figure 1 gives an example of the scanning process for a particular set $S$ of input sequences and particular values for the parameters $L$ and $W$.
The result of the scanning process described above is a set containing all the elementary patterns (and their associated offset lists) which satisfy the minimum support requirement. This set is the input to the convolution phase. To understand how the convolution works, it helps to realize that the scanning phase of the algorithm breaks up all the patterns that exist in the input into smaller pieces. The task, then, of the convolution phase is to put these pieces back together (in a time/space-efficient way) in order to recover the original patterns.
The key observation behind the convolution phase is that an original pattern $P$ can be reconstructed by piecing together pairs $A, B$ of intermediate patterns such that a suffix of $A$ is the same as a prefix of $B$. For example, consider again the set of sequences $S$ given in Figure 1. This set contains the pattern 'F.ASTS'. It is possible to reconstruct this pattern in the following way (see also Figure 2):


Fig. 1. The scanning process builds the set of all $\langle L, W\rangle$ elementary patterns with support at least $K$. Each elementary pattern in the set has an associated offset list. For the set $S$ here, the only $<3,4>$ patterns that appear in all the three input sequences are the following: 'F.AS', 'AST', 'AS.S', 'STS' and 'A.TS'.


Fig. 2. Reconstructing the maximal pattern 'F.ASTS' from the elementary patterns gathered in the scanning phase.

1. Combine together the elementary patterns ' $F . A S$ ' and ' $A S T$ ' into the pattern ' $F . A S T$ ' (observe that ' $A S$ ' is both a suffix of the pattern ' $F . A S$ ' and a prefix of ' $A S T$ ').
2. Combine the newly generated pattern ' $F . A S T$ ' with the elementary pattern 'STS' to get 'F.ASTS' (again, 'ST' is a suffix of ' $F . A S T$ ' and a prefix of 'STS').

Figure 2 depicts these two steps graphically.
To make the above description more precise, we need the following definitions: given any pattern $P$ with at least $L$ residues, let $\operatorname{prefix}(P)$ be the (uniquely defined) subpattern of $P$ that has exactly $(L-1)$ residues and is a prefix of $P$. For example, if $L=3$, then

$$
\operatorname{prefix}\left({ }^{`} F . A S T S '\right)=‘ F . A^{\prime}, \operatorname{prefix}\left({ }^{‘} A S T '\right)=‘ A S '
$$

Similarly, let $\operatorname{suffix}(P)$ denote the suffix subpattern of $P$ with exactly $(L-1)$ residues. Again, for $L=3$ :

$$
\operatorname{suffix}\left({ }^{`} F . A . . . S^{\prime}\right)=‘ A . . S ', \text { suffix }\left({ }^{`} A S T S '\right)=‘ T S '
$$

We can now describe a new binary operation, referred to herein as convolution (denoted by $\oplus$ ), between any pair of
$P=A S D . \ldots F$
$Q=S E \cdot E R F \cdot D G$
Prefix Wise Less

| Step 1: Align the patterns along their left-most columns | Step 2: Locate the first column containing both a don't care and a residue character (this column is marked with an $\mathbf{x}$ below). | Step 3: The pattern with the residue character in the $\mathbf{x}$-marked columr is the "smallest". |
| :---: | :---: | :---: |
| P: ASD...F | P: ASD...F | $\mathrm{P}<_{p f} \mathrm{Q}$ |
|  | \|x |  |
| Q: SE.ERF.DG | Q: SE.ERF.DG |  |

Suffix Wise Less

| Step 1: Align the patterns | Step 2: Locate the first column | Step 3: The pattern with the |
| :---: | :---: | :---: |
| along their right-most columns | containing both a don't care and a residue character (this column is marked with an $\mathbf{x}$ below). | residue character in the x -marked columr is the "smallest". |
| P: ASD...F | $\begin{array}{r} P: \quad \text { ASD } \ldots F \\ \mathbf{x} \end{array}$ | $Q<{ }_{s f} P$ |
| Q: SE.ERF.DG | Q: SE.ERF.DG |  |

Fig. 3. Deciding which of two patterns is prefix-wise (or suffix-wise) less than the other.
patterns. Let $P, Q$ be arbitrary patterns with at least $L$ residues each; the convolution of $P$ with $Q$ is a new pattern $R$ defined as follows:

$$
R=P \oplus Q \begin{cases}\frac{P Q^{\prime}}{\phi} & \begin{array}{l}
\text { if suffix }(P)=\operatorname{prefix}(Q)_{3} \\
\text { otherwise }
\end{array}\end{cases}
$$

where $Q^{\prime}$ is a string such that $Q=\operatorname{prefix}(Q) Q^{\prime}$ [i.e. $Q^{\prime}$ is what remains of $Q$ after the $\operatorname{prefix}(Q)$ is thrown away] and $\phi$ denotes the empty string. The patterns $P, Q$ are called convolvable if $P \oplus Q \neq \phi$. Here are some examples for the case $L=3$ :
$' D F . A . T \bigoplus^{\prime} A . T S E '=$ 'DF.A.TSE', 'AS.TF' ${ }^{\prime}$ 'T.FDE' $=\phi$
If two patterns $P, Q$ are convolvable and $R=P \oplus Q$, then the offset list $L_{S}(R)$ of the resulting pattern $R$ is the subset of $L_{S}(P)$ defined as:
$L_{S}(R)=\left\{(i, j) \in L_{S}(P) \mid \exists(i, k) \in L_{S}(Q)\right.$ such that $k-j=|P|-\mid$ suffix $\left.(P) \mid\right\}$,
where $|P|$ denotes the number of characters (counting both residues and don't cares) in the pattern $P$.
Opting to use convolution as the main tool for reconstructing the original patterns turns out to be very efficient: in the naive 'all-against-all' approach, when an intermediate pattern $P$ is extended it has to be compared with every possible pattern $Q$ in order to see whether $P$ and $Q$ can be pasted together into a larger pattern. We, on the other hand, focus only on patterns $Q$ such that $P$ and $Q$ are convolvable (the exact details are given below). As a result, in extending the pattern $P$ only a small number of patterns $Q$ are actually considered; furthermore, this number becomes smaller as the value of parameter $L$ becomes larger. The challenge now is, while
using the convolution, to still be able to (i) generate all the patterns and (ii) manage to identify and discard quickly patterns that are not maximal.

In order to achieve the above goals, two partial orderings on the universe of patterns are introduced. These orderings will be used to guide the way the convolutions are performed. Using them, we can guarantee that (i) all patterns are generated and (ii) a maximal pattern $P$ is generated before any non-maximal pattern subsumed by $P$. This way a nonmaximal pattern can be detected with minimal effort, just by comparing it against all patterns reported up to that point (comparisons can be made very efficiently using the appropriate hashing scheme to keep track of the maximal patterns). Instead of giving a rigorous mathematical definition, we describe these orderings by example.

Let $P, Q$ be two arbitrary patterns. To decide if $P$ is prefixwise less than $Q$ (we write this as $P<_{p f} Q$ ), we use the following procedure. First, the two patterns are aligned so that their left-most residues are in the same column. Second, the columns of the alignment are examined starting at the left and proceeding to the right. We stop when we see a column (if any exists) in which one of the aligned characters is a residue and the other is a don't care character. If the residue comes from the pattern $P$, then $P$ is prefix-wise less than $Q$ (see Figure 3 for a concrete example).

In exactly the same way, we can determine whether a pattern $P$ is suffix-wise less than $Q$ (we write $P<_{s f} Q$ ), only now the alignment is done so that the right-most residues of the two patterns are aligned together, and the examination of the columns starts from the right and proceeds to the left.

We can now describe the operation of TEIRESIAS. The result patterns are generated and reported in stages. Each stage will construct all the maximal patterns that have a given support $K^{\prime}$ (where $K^{\prime} \geq K$ ). A stack is used at each stage. At the beginning of the stage, the stack is initialized with all the elementary patterns $P$ having support at least $K$. Furthermore, the entries of the stack are ordered according to the 'prefix-wise less than' partial ordering defined above. This means that if $P<_{p f} Q$, then the elementary pattern $P$ is closer to the top of stack than the elementary pattern $Q$.
The algorithm always works with the pattern $P$ which is found at the top of stack; we call this pattern the current top. First, it will extend this pattern to the 'right' (suffix-wise) by looking at patterns $Q$ in the stack which are convolvable with $P$, i.e. $P \oplus Q \neq \phi$. If more than one such $Q$ exists, then each of them will be convolved with $P$, one at a time. Specifically, if $Q_{1}, Q_{2}$ are two such patterns and $Q_{1}<p f Q_{2}$, then the convolution of $P$ with $Q_{1}$ is tried before that with $Q_{2}$. Let $R$ be the pattern resulting from such a convolution. If $R$ is matched by fewer than $K$ distinct input sequences (this can easily be checked by examining the offset list of $R$ ), then $R$ is discarded, the current top remains unchanged, and the next convolution is tried out. Otherwise, $R$ is placed at the top of stack, thus becoming the new current top, and the procedure starts over again, this time with the new current top.
After the pattern $P$ at the top of the stack can no longer be extended to the suffix direction, the same process is applied trying now to extend $P$ to the left (prefix-wise). This time we are looking for patterns $Q$ such that $Q \oplus P \neq \phi$ (again, if more than one such pattern exists, the partial ordering is $<_{s f}$ used in order to resolve the order in which the convolutions will be performed).

When extension in both directions has been completed, the current top is popped from the stack and is checked for maximality. If found to be maximal, it is reported. The process starts all over with the new top of stack. The processing of the current stage terminates when no more patterns remain in the stack.
It can be shown rigorously that the procedure described above: (i) finishes; (ii) produces all the maximal $\langle L, W\rangle$ patterns satisfying the minimum support requirement; and (iii) reports no pattern that is not maximal.

## Results

In this section, we demonstrate the usefulness of the algorithm by applying it to a number of test cases. We consider two data sets: core histones (H3 and H4) and leghemoglobins. In the first case, TEIRESIAS is used in order to discover similarities across the (weakly-related) core histone families H 3 and H 4 . For the leghemoglobins, the algorithm is used for the determination of PROSITE-like patterns that are characteristic of the family. We also evaluate the performance of

TEIRESIAS through a series of experiments using carefully designed synthetic data; we start with an original sequence, mutate it extensively and generate sets of sequences with varying degrees of similarity. The behavior of the algorithm is then studied as a function of that similarity.

## Core histones H3 and H4

Core histones have been the object of extensive study due to their central role in the packaging of DNA within the cell. These small proteins are rich in positively charged amino acids that help them bind to the negatively charged DNA double helix (Watson et al., 1987). The four core histones (H2A, H2B, H3 and H4) bind together into an octameric construct (reminiscent of a cylindrical wedge) that provides the substrate for 146-bp-long DNA segments to wrap around, thus creating the nucleosome complexes within the cell chromatin. Examination of this octamer through crystallographic methods (Arents and Moudrianakis, 1993) has revealed the existence of the core histone motif, a particular structural construct found to be shared by all the core histone proteins. As it turns out, this motif plays a vital role in the assembly of histone pairs into dimers, which are further combined into the core histone octamer.

Apart from being a deciding factor in the internal architecture of the core histone octamer, the existence of the motif in all of the core histones has also provided evidence towards the validation of a long-standing speculation (Reeck et al., 1983), namely the alleged evolution of the core histones from a common ancestral protein. Recent work (Baxevanis, 1995; Ouzounis and Kyrpides, 1996), based on the alignment of core histone proteins along their common motif, has indeed established this evolutionary relationship: all histones (as well as the eukaryotic transcription factors CBF-A/C) have been found to contain sequence patterns that are characteristic of the core histone fold (Ouzounis and Kyrpides, 1996).

In order to demonstrate its utility, we have used TEIRESIAS to analyze the core histone families H3 and H4. Our intention was to see whether we could find evidence of the common evolutionary origin of these two families in the form of patterns preserved in both H3 and H4 proteins. This is a challenging task, as there is virtually no observable similarity across the two protein families (while, within family limits, proteins are highly conserved).

For the purpose of carrying out the experiments, we selected representative subsets from both families, making sure to include proteins from a wide range of organisms. We used subsets instead of the entire subfamilies in order also to check the predictive power of the patterns found, i.e. their ability to match other histone sequences not in the original input set. The input set comprised 20 proteins ( 13 from the H3 family and seven from the H4 family) and is shown in

Table 1. The parameter settings used were $L=3$ (in the later section on performance, we discuss at some length how the parameter $L$ is chosen), $W=35$ and $K=7$. We also tried larger values for $W$, but no more substantial patterns were discovered.

Table 1. The SwissProt labels of the 20 sequences from the H 3 and H 4 families that were used in the experiments ( 13 sequences come from the H3 family and seven from the H 4 family)

| H33_HUMAN | H32_BOVIN | H32_XENLA | H3_PSAMI | H3_STRPU |
| :--- | :--- | :--- | :--- | :--- |
| H31_HUMAN | H3_ENCAL | H3_CAEEL | H3_PEA | H31_SCHPO |
| H3_YEAST | H34_CAIMO | H34_MOUSE |  |  |
| H4_HUMAN | H4_CAEEL | H4_WHEAT | H4_YEAST | H4_SCHPO |
| H41_TETPY | H42_TETPY |  |  |  |

Processing the input set with TEIRESIAS required only a few seconds on an IBM Power-PC workstation, and a large set of patterns was discovered. This set contained patterns that were common to both families, as well as patterns that were common to only the members of a single family. In Figure 4, we present only a small part of the output produced by TEIRESIAS; more specifically, we show all of the discovered patterns having four or more amino acids and occurring in at least 19 out of the 20 sequences (the patterns have been aligned with one representative member of the H 3 and H4 families in order to show clearly the patterns' positions in the set). Interestingly enough, there are quite a few patterns appearing in all the proteins in our input set. In order to verify the extent to which these patterns do indeed indicate evolutionary relationship (and are not found just by chance), we searched SwissProt (Release 34) for sequences matching the most descriptive among these patterns, i.e. those with the largest number of non-don't care positions (Table 2).
The three patterns we used for the search turned out to be very specific to the H 3 and H 4 protein families; they are able to pick almost all the members of these two families. They are also very selective as they generated no false positives. In that same table, we also show the results of searching the given patterns in the non-redundant database of the core histone sequences maintained at the National Center for Biotechnology Information (NCBI) (Baxevanis et al., 1995; Baxevanis and Landsman, 1997). The high sensitivity of the patterns chosen is also clearly demonstrated by that experiment: the patterns are found to belong in just about all the histone proteins in the non-redundant database.
In Figure 5, we show a small part of the results produced by TEIRESIAS for the H3 (13 sequences) and H4 (seven sequences) subsets [the set of all the patterns that have been discovered in this set of core histones is reported in Rigoutsos et al. (1997a)]. In each case, only patterns found to belong in all the members of the respective subset are shown. The fig-
ure depicts clearly the extensive degree of amino acid conservation within subfamily boundaries.

Table 2. For each pattern, the numbers of H3 and H4 members found to contain the pattern are shown. The searches were performed over SwissProt Release 34 (containing 33 histones 3 and 20 histones 4 ) and the non-redundant data base at NCBI (containing 81 histones 3 and 59 histones 4). It is interesting to note that in searching SwissProt, not a single false positive was generated. Only the pattern 'E......V...E...........V....K.........G' was matched by proteins that were not clearly annotated as histones, namely the proteins CENA_BOVIN, CENA_HUMAN and YB21_CAEEL. All three of these proteins, however, are hypothetical H3-like proteins: the first two play the role of histones in the assembly of centromeres, while the last one is a hypothetical histone 3 protein in chromosome X (Bairoch and Apweiler, 1997)

| Patterns | SwissProt | NCBI |  |  |
| :--- | :--- | :--- | :--- | :--- |
|  | H3 (33) | H4 (20) | H3(81) | H4 (59) |
| G......................T...I........V..I........R | 27 | 19 | 79 | 58 |
| K.A.......GGVK | 24 | 20 | 77 | 59 |
| E.........E...............K.........G | 27 | 19 | 76 | 58 |

Looking at the alignment of the discovered patterns along the H33_HUMAN protein, one notices that the following two H3 family patterns
$P_{1}=$ 'A.TKQTA.KST..KAPRKQL..KAA.K.AP..GGVKK.H...P.TVAL.EI........L'
$P_{2}=$ 'STELLI...PFQRLV.EIAQDFKT.LRFQ..A..ALQE..EA..V.LFEDTNL.AIH.K. V....KD..L.....GER'
fit almost perfectly the two structural domains reported for the H3 family in the PRODOM database, namely the domains with accession numbers 687 (pattern $\mathrm{P}_{1}$ ) and 521 (pattern $\mathrm{P}_{2}$ ). The first of these two domains is reported in PRODOM to appear in 29 sequences, all of which are annotated as histones 3, while the second domain is found in 36 proteins of which 30 are in the H3 family while the remaining six (SwissProt names: YB21_CAEEL, CSE4_YEAST, CENA_BOVIN, CENA_ HUMAN, YMH3_CAEEL, YL82_CAEEL) are H3-like proteins.

Verifying observations. In order to verify the correspondence between patterns and PRODOM domains observed in Figure 5, we performed the following experiment. Each pattern was slid over all sequences in the SwissProt database. At every offset, we counted all characters of the sequence that matched a non-don't care character of the pattern (this is equivalent to generating a scoring matrix from the pattern where every amino acid position of the patterns contributes a weight of one, while every don't care character contributes nothing). The final score assigned to a sequence was the maximum among all the scores of its offsets. As it turns out, even our simple scoring strategy is quite effective; for both patterns the highest scoring sequences ( 29 for $\mathrm{P}_{1}$ and 36 for $\mathrm{P}_{2}$ ) were exactly those found to be in the corresponding PRODOM domains.

## $>$ H33 <br> _HUMAN

ARTKQTARKSTGGKAPRKQLATKAARKSAPSTGGVKKPHRYRPGTVALREIRRYQKSTELLIRKLPEQRLVREIAODEKTDLRFOSAAIGALOEASEAYLVGLEEDINLCAIHAKRVTIMPKDIOLARRIBGERA
Patterns shared by $\mathbf{2 0 / 2 0}$ sequences

$>H 4$ _HUMAN
SGRGKGGKGLGKGGAKRHRKVLRDNIQGITKPAIRRLARRGGVKRISGLIYEETRGVLKVFLENVIRDAVTYTEHAKRKTVTAMDVVYALKROGRTLYGFGG


Fig. 4. Several of the discovered patterns that are common to both the H 3 and H 4 families. The patterns are shown aligned with one protein from the H3 family (H33_HUMAN) and one protein from the H4 family (H4_HUMAN). The underlined portion of each protein marks the location of the core histone fold on that protein. Only the patterns that occurred in at least 19 sequences of the input set's 20 sequences are shown. It is interesting to observe that for several of the discovered patterns the relative order has not been preserved during evolution.

Table 3. The SwissProt labels of the 22 leghemoglobin sequences as they appear in the respective entry of PROSITE Release 13.2 (Bairoch et al., 1996)

| BP1_CASGL6 | HBP2_CASGL | HBPL_PARAD | HBPL_TRETO | LGB1_LUPLU | LGB1_MEDSA |
| :--- | :--- | :--- | :--- | :--- | :--- |
| LGB1_MEDTR | LGB1_PEA | LGB1_SOYBN | LGB1_VICFA | LGB2_LUPLU | LGB2_MEDTR |
| LGB2_SESRO | LGB2_SOYBN | LGB3_MEDSA | LGB3_SESRO | LGB3_SOYBN | LGB4_MEDSA |
| LGBA_PHAVU | LGBA_SOYBN | LGB_PSOTE | HBP_CANLI |  |  |

## Leghemoglobins

Leghemoglobins are plant hemoglobins that show sequence similarity to other members of the large globin superfamily, especially to myo- and hemoglobins from animals (Kapp et al., 1995). All globins contain a conserved histidine signa-
ture that represents the heme-binding pocket (Kapp et al., 1995).

We presented TEIRESIAS with the set of leghemoblogins shown in Table 3. This is exactly the set of SwissProt proteins comprising the plant globin family of the PROSITE (Release 13.2) entry with accession number PS00208. Our intention


Fig. 5. Discovered patterns that are common to all of the sequences of the (a) H 3 and (b) H 4 families. For each family, the patterns are shown aligned with the corresponding human protein. In the H33_HUMAN protein, we have also marked the two structural domains of the H3 histone family as they are reported in the PRODOM database (http://protein.toulouse.inra.fr/prodom/prodom.html): the first domain (PRODOM \#687) covers the underlined part of the sequence, while the second domain (PRODOM \#521) stretches over the doubly underlined part. As can be seen in (a) above, the two substantial patterns found for the H3 family fit these two domains almost perfectly.
was to determine whether the algorithm could identify the heme-binding pocket signature successfully, or any other important pattern. We used the parameter settings $L=3, W=35$ and $K=10$ (as with the core histones, larger values of $W$ did not introduce any more interesting patterns).
Figure 6 shows those patterns that have been determined by the algorithm to be common to at least 21 of the 22 leghemoglobin sequences of the input set; the patterns are shown aligned with the sequence LGBA_SOYBN [the set of all the patterns that have been discovered in this set of leghemoglobins is reported in Rigoutsos et al. (1997b)]. Among the patterns belonging to all the input sequences, we find 'P.L..HA...F......A..L...G'. This is, in effect, the leghemoglobin signature reported in the PROSITE database (to be exact, the pattern reported there is '[SN]P.L..HA...F') and it describes the heme-binding pocket. Another pattern identified by TEIRESIAS and appearing in all the input leghemoglobin sequences is 'A.L.T.K......W..........AY..L....K': furthermore, a search of SwissProt reveals that it is specific to leghemoglobins and creates no false positives. As it turns out, this particular pattern spans the last two helices in the tertiary structure of the leghemoglobin.

Determining descriptive power. In order to check the descriptive power of the above two patterns in greater detail, we used the aggregate database of proteins at NCBI [including entries from, among others, the SwissProt (Bairoch and Ap-
weiler, 1997), EMBL (Rodriguez-Tomé et al., 1996), GenBank (Benson et al., 1997) and PIR (Sidman et al., 1988) databases] to extract a large number of leghemoglobin sequences. The sequences were obtained using BLAST to search for proteins similar to the leghemoglobin LGBA_SOYBN; from the search results, only proteins with a probability of $<0.005$ were retained for further processing. The sequences that survived this first screening were then aligned and divided into groups, each group comprising of proteins differing in at most two positions. Finally, we constructed a non-redundant set of leghemoglobin sequences by selecting one representative from each group; we assumed that all sequences within a group really code for the same protein and that the small differences are the result of sequencing errors. The result of the process described above was a set containing 60 proteins: the 22 SwissProt proteins listed in Table 3 and 38 other leghemoglobins (listed in Table 4).

We then proceeded to search the two patterns in this larger leghemoglobin database: both patterns turn out to be very sensitive (Table 5). Each pattern matches every sequence either exactly or within one edit operation (mutation, insertion or deletion), with the exception of one case which requires two mutations. The only sequences missed are those which are fragments and do not contain the region corresponding to the particular pattern at hand.
VLGFTEKODALVSSSEEAFKANIPOYSVVFYTSILEKKAPAAKDLESELANGVDPTNPKLTGHAEKLEALVRDSAGOLKASGTVYADAALGSVHAOKAVTDPOFVVVKEALLKTIKAAVGDKWSDELSRAWEVAYDELAAAIKKA

|  | BL00208A |  | BL00208B |  | BL00208C |
| :---: | :---: | :---: | :---: | :---: | :---: |
| Patterns | shared by 22/22 | sequences |  |  |  |
| т.K..... | ..s.e. . . . . . ${ }^{\text {y }}$ | AP.....F..L | P.L. .HA. . F......A..L. . ${ }^{\text {g }}$ | T. K . | .... ${ }^{\text {y }}$ |
| т.K.... | . .s.......... Y $^{\text {S }}$ |  | P.L. . . . . . . . . . . A..L | т.K. | . . . Y |
| T.K..... | ..s.E |  | L. . . A. . . . . . . . . ${ }^{\text {. }}$. . . . .G | т.K. |  |
| T. Q.AL |  |  |  | T.K. | .... ${ }^{\text {y }}$. L |
|  |  |  |  | A.L.T.K. | . AY...L. |
|  |  |  |  | KEA...... |  |
|  |  |  |  | F...K... | . . . . L. |
|  |  |  |  | V.K........ | ..... L. |
| Patterns shared by $21 / 22$ sequences |  |  |  | A.L.T. |  |
| T. Q.AL. | .....n F. | ...AP.A...F..L | P.L..HA...F......A.QL...G | H........F.V...A.L.T.K. | . .AY..L. |
| т.K.... | . . . . . A. . . $Y$ | E.AP.....F. L |  | , | A.... Y |
| T.....lv. | . . . . . . ${ }^{\text {k }}$ | AP.. K.FF. . L |  | A.L.T.K | .w. AY. .L. |
| T. Q.AL.. | . | AP.....FS.L |  | A.L.t.K. | ....AY. . $L$ |
| T. Q.ALV | S.... | . AP......F. ${ }^{\text {L }}$ |  | A.L.T.K | A. . AY. . |
|  |  |  |  | A.L.T.K. | ....... ${ }^{\text {. . }}$ L. |
|  |  |  |  | т.K. | . . . Y. . ${ }^{\text {L }}$ |
|  |  |  |  | VVK............ | …........ |
|  |  |  |  | T.K. | ....... |
|  |  |  |  | K......F....E...... ${ }^{\text {K }}$ |  |
|  |  |  |  | H..................... | A. . . AY |
|  |  |  |  | F....EA..... |  |
|  |  |  |  | F......... T. | . A |
|  |  |  |  | $\text { v...A...... } \mathrm{K} .$ | . L. |
|  |  |  |  | V.E......K. |  |
|  |  |  |  | A..... ${ }_{\text {K }}$ |  |
|  |  |  |  | $\begin{aligned} & \text { A. . T. } \mathrm{K} \\ & \text { A. . } \mathrm{T} \cdot \mathrm{~K} \end{aligned}$ |  |
|  |  |  |  | A........ |  |
|  |  |  |  | T.K. | . . . . . . . |
|  |  |  |  | L.... |  |
|  |  |  |  | v.K....... | . . . . . . L |
|  |  |  |  | V............. | A........... |
|  |  |  |  | A.L. ${ }^{\text {a }}$ |  |
|  |  |  |  |  | ...... L |

Fig. 6. All of the discovered patterns that are common to at least 21 of the 22 sequences in the leghemoglobin input set (shown aligned with the LGBA_SOYBN protein). The underlined regions of the protein correspond to the three blocks of the leghemoglobin sequence family, as they are reported in the BLOCKS database (Henikoff and Henikoff, 1991, 1994). The corresponding block names are BL00208A, BL00208B and BL00208C.

Table 4. The non-SwissProt proteins included in the non-redundant set of leghemoglobins. Each sequence is represented by its locus and a letter code, indicating the database it came from. The following letter codes are used: E (for EMBL), P (for PIR), G (for GenBank) and O (other)

| OSU76028 | $\mathbf{G}$ | OSU76029 | $\mathbf{G}$ | ALFLEGHEMA | $\mathbf{G}$ | ALFLEGHEMB | $\mathbf{G}$ | SESLBDRLA | $\mathbf{G}$ |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- | :--- |
| SESLBDRLB | $\mathbf{G}$ | VUU33205 | $\mathbf{G}$ | VUU33206 | $\mathbf{G}$ | VFALBA | $\mathbf{G}$ | VFALBB |  |
| VFALBC | $\mathbf{G}$ | PHVLBA | $\mathbf{G}$ | SOYLBGII | $\mathbf{G}$ | GMU4713 | $\mathbf{G}$ |  |  |
| S21371 | $\mathbf{P}$ | S21372 | $\mathbf{P}$ | S21373 | $\mathbf{P}$ | S21374 | $\mathbf{P}$ | S21375 |  |
|  |  |  |  |  |  |  |  |  |  |
| S46502 | $\mathbf{P}$ | S08507 | $\mathbf{P}$ | S08508 | $\mathbf{P}$ | S01020 | $\mathbf{P}$ | S42046 |  |
| GPDRNL | $\mathbf{P}$ | GPFJ2 | $\mathbf{P}$ | GPSYC2 | $\mathbf{P}$ | A20801 | $\mathbf{P}$ | A54493 | P |
| VFLBBMR | $\mathbf{E}$ | VFLBKMR | $\mathbf{E}$ | VFLB1 | $\mathbf{E}$ | VFLB29MR | $\mathbf{E}$ | VFLB49MR | $\mathbf{E}$ |
| MSLEGH12 | $\mathbf{E}$ |  |  |  |  |  |  |  |  |
| AB004549 | $\mathbf{O}$ | $1102189 B$ | $\mathbf{O}$ | $711674 A$ | $\mathbf{O}$ |  |  |  |  |

Table 5. Results of searching the two patterns discovered by TEIRESIAS in the non-redundant leghemoglobin database (containing 60 proteins). The term 'edit operation' denotes either a mutation or an insertion or a deletion

| Pattern | No. of sequences <br> matched <br> exactly | No. of sequences <br> matched within <br> 1 edit operation | No. of sequences <br> matched within <br> 2 edit operations | No. of sequences <br> unmatched due to <br> fragmental data |
| :--- | :--- | :--- | :--- | :--- |
| P.L..HA...F.....A..L...G | 51 | 1 | 1 | 7 |
| A.L.T.K......W.........AY..L....K | 48 | 9 | 0 | 3 |

On conserved regions. The results obtained by TEIRESIAS for the leghemoglobin family can be used to highlight another application for the algorithm, namely the derivation of information regarding the conserved regions in a family of sequences. The alignment of patterns shown in Figure 6 reveals four such regions: two of them near the N -terminus of LGBA_SOYBN, one in the middle of the sequence, corresponding to the heme-binding domain, and one close to the C-terminus. These regions are also identified in the BLOCKS database (Henikoff and Henikoff, 1991, 1994) as the blocks BL00208A, BL00208B and BL00208C of the plant globin family. The location of these blocks on LGBA_SOYBN is shown in Figure 6; the correspondence between our four regions and the BLOCKS results is clear. The only difference is that we find two separate areas corresponding to what is reported as the single block BL00208A in the BLOCKS database. We are able to identify the aforementioned regions by assigning a cost to each amino acid position within an input sequence. This cost indicates, roughly, the degree of conservation of that position by taking into account the number of patterns where that position appears preserved. Looking at Figure 6, for example, it is clear that the rightmost of the four regions shows a heavier concentration of patterns. This is an indication that the amino acids in that region have been preserved relatively well during evolution; in Kapp et al. (1995), the authors reach the same conclusion by studying the multiple alignment of many globin sequences. It is important to note that the precision of our approach relies on the fact that our algorithm can generate all the maximal patterns that the input set contains. A detailed description of our method can be found in I.Rigoutsos and A.Floratos (in preparation).

## Performance

In the course of experimenting with various input sets, we have observed that perhaps the single most important factor which affects the performance of the algorithm is the amount of similarity between the input sequences. In order to evaluate better the relationship between performance and input composition, we carried out a number of experiments. The starting point for all of them was a random sequence $P$ of 400 amino acids; the sequence was generated by the random protein generator at http://www.expasy.ch/sprot/randseq. html using the amino acid frequencies from the most recent version of SwissProt. We then fixed a percentage $X$ and used the original protein $P$ in order to obtain 20 derivative proteins, each one of them having a pairwise similarity of about $X \%$ to $P$. The derivative proteins were obtained through successive applications of the appropriate PAM matrix (George et al., 1990) on $P$.

We obtained six such input sets of proteins: for $X$ being 40, $50,60,70,80$ and $90 \%$. For each set, we ran TEIRESIAS
using a different minimum support (i.e. value of $K$ ) every time. Furthermore, for every choice of the minimum support, we used several different values for parameter $W$ of the algorithm. Parameter $L$ was set to 3 ; the reason for setting $L=3$ is that this is the smallest value for which the benefits of convolution become apparent as the prefixes and suffixes used are non-trivial. Larger values of the parameter $L$ affect the performance of the algorithm by decreasing the convolution time while increasing the scanning time. We have found this to be a really favorable trade-off only when there is a subset of at least $K$ input sequences ( $K$ being the minimum support) exhibiting extensive degree of similarity. In all other cases, there seems to be no substantial gain. For each execution of the algorithm, we kept track of two things: (i) the running time and (ii) the total number of maximal patterns reported by the algorithm. Figures 7 and 8 provide a graphical representation of the results for the cases $W=10$ and $W=15$. The measurements obtained for other values of $W$ are similar.

The algorithm required just a few seconds when the minimum support was set at about the total number of sequences in the input. Furthermore, the variation in the similarity among the different input sets did not have any observable effect. The reason for this particular behavior is that, independent of the amount of similarity, there are not too many patterns that belong to all (or almost all) the input sequences. When we allow for smaller minimum support values, the degree of similarity in the input becomes an important factor. More specifically, the higher the similarity, the longer it takes to produce the final patterns. Examining the curve corresponding to the $X=90 \%$ similarity case, one can see that even small decreases in the minimum support create a very large number of additional patterns. This is a direct consequence of the fact that as the degree of similarity in the input increases, so does the number of distinct patterns that are contained in the input. In fact, it is not hard to construct inputs where the number of maximal patterns is exponential to the size of the input. So, the increase in the running time is in a sense unavoidable: simply reporting the output becomes a daunting task, let alone generating it.

For problems like this, where part of the complexity stems from the sheer size of the solution, the best that one can hope for is an algorithm which is output sensitive, i.e. its running time is almost linear on the actual size of the output. The experimental results shown here indicate that TEIRESIAS does indeed exhibit such desirable behavior; as is evident from Figures 7 and 8, the rate of increase in the running time of the algorithm mirrors almost exactly the corresponding rate of increase in the number of patterns in the output. Furthermore, additional experiments with varying input sizes (ranging up to 1000 sequences) produced curves almost identical to the ones shown below. In conclusion, the size of the output seems to be the only factor affecting the performance of the algorithm; the running time remains reasonable

$$
W=10
$$



Fig. 7. For every similarity level (ranging from 40 to $90 \%$ ), we run TEIRESIAS on the input set corresponding to that level. For every such input set (containing 20 sequences), we performed a number of experiments, each time allowing a different value for the minimum support (the values ranged from 20 down to 12). For each execution of the algorithm, we recorded the running time (right graph above) and the total number of patterns reported by TEIRESIAS (left graph). In every run, the value of parameter $W$ was set to 10 .

$$
W=15
$$



Fig. 8. Same as Figure 7, for parameter $W$ set to 15.
even for very large input sets as long as they contain a moderate number of patterns.

## Discussion

In this work, we presented and discussed TEIRESIAS, a novel algorithm for the discovery of rigid patterns in un-
aligned biological sequences. In order to evaluate the utility of the algorithm, we have applied it to two distinct sets of inputs. In both cases, we demonstrated that the algorithm, in the absence of any context information, is able to derive results of proven biological significance. Furthermore, we briefly discussed how the complete set of patterns that the algorithm generates can be exploited towards the automatic
discovery of preserved regions along the proteins of the input set. Finally, we described a set of experiments demonstrating the performance of the algorithm.
We believe that what distinguishes TEIRESIAS from the existing methods of discovering local similarities in biological sequences is the combined effect of the following two features: (i) it finds all the maximal patterns with (a userspecified) minimum support; (ii) its performance scales quasi-linearly with the size of the output.
The enhanced performance achieved by our algorithm stems from the utilization of the convolution operation; instead of starting with a seed pattern and extending it by just one position at a time (as several other methods do), the convolution operation permits the extension of a pattern by several positions in a single step. Furthermore, our ordering of the intermediate patterns when performing the convolutions provides additional performance benefits, by avoiding the generation of non-maximal patterns. The speed gains achieved afford one the ability to look for patterns with very small supports. This is particularly useful when the composition of the input is not uniform, i.e. when it is comprised of sequences that do not necessarily all belong to a single group. This was the case with the core histones; although there were a few weak patterns shared by all the proteins, when we allowed the support to be very small, larger patterns that distinguished the H3 from the H4 members appeared. In this particular example, the ability to find patterns with small support permitted a finer-grain analysis of the sequences comprising the input. Independent research (E.BornbergBauer, E.Rivals and M.Vingron, Of Coils and Zippers, unpublished data, 1997) has also validated the usefulness of TEIRESIAS for this kind of analysis.
Another property that differentiates TEIRESIAS from existing work is the kind of structural restriction the user is allowed to impose on the patterns to search for. In all the algorithms that we are aware of, the speed of the pattern-discovery process can be controlled by bounding (in one manner or another) the length of the reported patterns. This, however, has the drawback that long patterns will escape attention or be broken into multiple non-maximal and overlapping pieces. In our case, only the parameter $W$ (which, in essence, indicates the maximum number of don't care characters between two successive residues in a pattern) needs to be set. It thus becomes possible to discover patterns of arbitrary length (e.g. the case of core histones; Figure 5) as long as preserved positions are not more than $W$ residues away.
Finally, TEIRESIAS is guaranteed to report all the maximal patterns meeting the structural restrictions set by the user. Other approaches restrict the search space by incorporating a probabilistic or information-theoretic model of importance in order to decide what patterns to look for and report. We are of the opinion that the assignment of a measure of importance to the patterns should be disjoint from the dis-
covery process. This way we can guarantee that all the existing patterns are indeed reported. The task of choosing which of them to keep ought to be a post-discovery, problem-specific consideration, and dependent on the particular reason the patterns were sought in the first place. For example, when identifying conserved regions along the input sequences (as in the case of the leghemoglobins), we need the entire set of existing patterns. On the other hand, when considering issues of pattern specificity/sensitivity, then what qualifies as 'important' might also be dependent on factors other than the input (e.g. the size and composition of the data base in which the patterns are to be searched for).

We should point out that, in its current implementation, the algorithm does not handle flexible gaps. For example, if this functionality were part of the algorithm, we would be able to see that the two leghemoglobin patterns 'P.L..HA...F......A..L...G' and 'A.L.T.K......W..........AY..L....K' are in fact the two pieces of one larger, flexible pattern; in that larger pattern, the two pieces appear separated by a variable number of don't cares which ranges from 27 to 31 positions. We are currently extending our method so that the detection of flexible gaps becomes an integral part of the algorithm. We have also been working on allowing groups of residues (rather than a sole residue) to occupy a single position within a pattern (as is the case, for example, with the patterns used in PROSITE).

## References

Aho,V.A., Hopcroft,J.E. and Ullman,J.D. (1974) The Design and Analysis of Computer Algorithms. Addison-Wesley, Menlo Park, CA.
Arents,G. and Moudrianakis,E.N. (1993) Topography of the histone octamer surface: repeating structural motifs utilized in the docking of nucleosomal DNA. Proc. Natl Acad. Sci. USA, 90, 10489-10493.
Bairoch,A. and Apweiler,R. (1997) The SWISS-PROT protein sequence data bank and its supplement TREMBL. Nucleic Acids Res., 25, 31-36.
Bairoch,A., Bucher,P. and Hofmann,K. (1996) The PROSITE database: its status in 1995. Nucleic Acids Res., 24, 189-196.
Baxevanis,A.D. and Landsman,D. (1997) Histone and histone fold sequences and structures: a database. Nucleic Acids Res., 25, 272-273.
Baxevanis,A.D., Arents,G., Moudrianakis,E.N. and Landsman,D. (1995) A variety of DNA-binding and multimeric proteins contain the histone fold motif. Nucleic Acids Res., 23, 2685-2691.
Benson,D.A., Boguski,M.S., Lipman,D.J. and Ostell,J. (1997) GenBank. Nucleic Acids Res., 25, 1-6.
Benson,G. and Waterman,M.S. (1994) A method for fast database search for all k-nucleotide repeats. In Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology, pp. 83-98.

Brazma,A., Jonassen,I., Eidhammer,I. and Gilbert,D. (1995) Approaches to the Automatic Discovery of Patterns in Biosequences. Technical Report, Department of Informatics, University of Bergen, Norway.
Carrillo,H. and Lipman,D. (1988) The multiple sequence alignment problem in biology. SIAM J. Appl. Math., 48, 1073-1082.
Delcoigne,A. and Hansen,P. (1975) Sequence comparison by dynamic programming. Biometrika, 62, 661-664.
Garey,M.R. and Johnson,D.S. (1979) Computers and Intractability: $a$ Guide to the Theory of NP-Completeness. Freeman, San Francisco.
George,D.G., Barker,W.C. and Hunt,L.T. (1990) Mutation data matrix and its uses. In Doolittle, R.F. (ed) Methods in Enzymology, Vol. 183, Molecular Evolution: Computer Analysis of Protein and Nucleic Acid Sequences. Academic Press, San Diego, CA, pp. 333-351.
Guan,X. and Uberbacher,E.C. (1996) A fast look-up algorithm for detecting repetitive DNA sequences. In Hunter, L. and Klein, T.E. (eds). Proceedings of the Pacific Symposium on Biocomputing. World Scientific, Singapore, pp. 718-719.
Henikoff,S. and Henikoff,J. (1991) Automatic assembly of protein blocks for database searching. Nucleic Acids Res., 19, 6565-6572.
Henikoff,S. and Henikoff,J. (1994) Protein family classification based on searching a database of blocks. Genomics, 19, 97-107.
Hirosawa et al. (1995) Comprehensive study on iterative algorithms of multiple sequence alignment. Comput. Applic. Biosci, 11, 13-18.
Jonassen,I., Collins,J.F. and Higgins,D.G. (1995) Finding flexible patterns in unaligned protein sequences. Protein Sci., 4, 1587-1595.
Kapp,O.H., Moens,L., Vanfleteren,J., Trotman,C.N., Suzuki,T. and Vinogradov,S.N. (1995) Alignment of 700 globin sequences: extent of amino acid substitution and its correlation with variation in volume. Protein Sci., 4, 2179-2190.
Maier,D. (1978) The complexity of some problems on subsequences and supersequences. J. ACM, 25, 322-336.
Martinez,M. (1983) An efficient method for finding repeats in molecular sequences. Nucleic Acids Res., 11, 4629-4634.
Martinez,M. (1988) A flexible multiple sequence alignment program. Nucleic Acids Res., 16, 1683-1691.
Needleman,S.B. and Wunsch,C.D. (1970) A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol., 48, 443-453.
Neuwald,A.F. and Green,P. (1994) Detecting patterns in protein sequences. J. Mol. Biol., 239, 698-712.
Neville-Manning,C.G., Sethi,K.S., Wu,D. and Brutlag,D.L. (1997) Enumerating and ranking discrete motifs. In Proceedings of Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA, pp. 202-209.
Ouzounis,C.A. and Kyrpides,N.C. (1996) Parallel origins of the nucleosome core and eukaryotic transcription from Archaea. J. Mol. Evol., 42, 234-239.
Reeck,G.R., Swanson,E. and Teller,D.C. (1983) The evolution of histones. J. Mol. Evol., 10, 309-317.
Rigoutsos,I. and Floratos,A. (1997) Discovery of Conserved regions in Biological Sequences Using Pattern Interlaying. IBM Thomas J.Watson Research Center Technical Report, in preparation.

Rigoutsos,I., Floratos,A. and Ouzounis,C. (1997a) Preliminary Results on the Discovery of Patterns of Amino Acids Common to Sequences of Core Histones 3 \& 4. IBM Thomas J.Watson Research Center Research Report RC20804.
Rigoutsos,I., Floratos,A. and Ouzounis,C. (1997b) Preliminary Results on the Discovery of Patterns of Amino Acids Common to Sequences of Leghemoglobins. IBM Thomas J.Watson Research Center Research Report RC20806.
Rodriguez-Tomé,P., Stoehr,P.J., Cameron,G.N. and Flores,T.P. (1996) The European Bioinformatics Institute (EBI) databases. Nucleic Acids Res., 24, 6-13.
Roytberg,M.A. (1992) A search for common patterns in many sequences. Comput. Applic. Biosci., 8, 57-64.
Sagot,M.F. and Viari,A. (1996) A double combinatorial approach to discovering patterns in biological sequences. In Proceedings of the 7th Symposium on Combinatorial Pattern Matching. Springer, pp. 186-208.
Sagot,M.-F., Viari,A. and Soldano,H. (1995) Multiple sequence comparison: a peptide matching approach. In Proceedings of the 6th Symposium on Combinatorial Pattern Matching. Springer, pp. 366-385.
Sidman,K.E., George,D.G., Barker,W.C. and Hunt,L.T. (1988) The Protein Identification Resource (PIR). Nucleic Acids Res., 16, 1869-1871.
Smith,H.O., Annau,T.M. and Chandrasegaran,S. (1990) Finding sequence motifs in groups of functionally related proteins. Proc. Natl Acad. Sci. USA, 87, pp. 826-830.
Smith,R.F. and Smith,T.F. (1990) Automatic generation of primary sequence patterns from sets of related protein sequences. Nucleic Acids Res., 18, 118-122.
Smith,T.F. and Waterman,M.S. (1981) Identification of common molecular subsequences. J. Mol. Biol., 147, 195-197.
Sobel,E. and Martinez,M. (1986) A multiple sequence alignment program. Nucleic Acids Res., 14, 363-374.
Suyama,M., Nishioka,T. and Jun'ichi,O. (1995) Searching for common sequence patterns among distantly related proteins. Protein Eng., 8, 1075-1080.
Wang,J., Marr,T.G., Shasha,D., Shapiro,B.A. and Chirn,G. (1994) Discovering active motifs in sets of related protein sequences and using them for classification. Nucleic Acids Res., 22, 2769-2775.
Wang,L. and Jiang,T. (1994) On the complexity of multiple sequence alignment. J. Comput. Biol., 1, 337-348.
Waterman,M.S., Galas,D.J. and Arratia,R. (1984) Pattern recognition in several sequences: consensus and alignment. Bull. Math. Biol., 46, 515-527.
Watson,J.D., Hopkins,N.H., Roberts,J.W., Steitz,J. and Weiner,A.M. (1987) Molecular Biology of the Gene, 4th edn. The Benjamin/Cummings Publishing Co., Menlo Park, CA.
Wu,T.D. and Brutlag,D.L. (1995) Identification of protein motifs using conserved amino acid properties and partitioning techniques. In Proceedings of the 3rd International Conference on Intelligent Systems for Molecular Biology. AAAI Press, Menlo Park, CA, pp. 402-410.

