RC 20803 (92166) 21APR97
Bioinformatics / Data Mining 11 pages

IBM Research Report

Case Studies In Pattern Discovery Without Alignment
Results Using the *Teiresias* Algorithm

Isidore Rigoutsos Aris Floratos
IBM Research Division
T.J. Watson Research Center
P.O. Box 218
Yorktown Heights, NY 10598

Christos Ouzounis
EMBL Cambridge
The European Bioinformatics Institute
Hinxton Genome Campus
Cambridge CB10 1SD, England, UK

LIMITED DISTRIBUTION NOTICE
This report has been submitted for publication outside of IBM and will probably be copyrighted if accepted for publication. It has been issued as a Research Report for early dissemination of its contents. In view of the transfer of copyright to the outside publisher, its distribution outside of IBM prior to publication should be limited to peer communications and specific requests. After outside publication, requests should be filled only by reprints or legally obtained copies of the article (e.g. payment of royalties).

IBM Research Division
Almaden . TJ Watson . Tokyo . Zurich . Austin

Case Studies In Pattern Discovery Without Alignment
Results Using the *Teiresias* Algorithm

Isidore Rigoutsos[1,2] *Aris Floratos*[1,3] *Christos Ouzounis*[4]

February 1997

Abstract

Pattern discovery (an NP-hard problem), in contrast to pattern matching (a problem solvable in polynomial time), is a challenging problem. Using *Teiresias*, a newly developed algorithm, we are exploring a number of test cases in order to identify patterns of amino acids that appear across sequences (common motifs) as well as within individual sequences (internal repeats). The examples aim at: (a) the validation of the approach through the discovery of previously reported patterns; (b) the demonstration of the capability to automatically identify highly selective patterns particular to the sequences under consideration; and, (c) the discovery of previously unidentified patterns in these well-studied example cases.

1. Introduction

Computer analysis of biological sequences has been developed to a high level of sophistication, thanks to relevant developments in computer science (for example, use of hidden Markov models) and molecular biology (namely, the influx of genome sequence information).

One of the interesting problems arising in sequence analysis is the discovery of common sub-sequences in a family of related proteins (or genes) [11,5,3]. These sub-sequences are usually elements conserved during evolution due to an important structural or functional role.

Typically, pattern discovery begins with the comparison of a query sequence against the full database, the identification of homologous sequence and the generation of multiple alignments for the identification of the most conserved sequence motifs [12].

Such comparisons necessitate the definition and use of a similarity (or "distance" metric) [13]. The metric is defined for every pair of proteins and leads to a partition of any data set into "natural" clusters of sequences that exhibit sufficiently small pair-wise distances.

Sequences that are similar will result in small distance values and are expected to also share functional properties. Although 1D sequence similarity typically translates into functional similarity the opposite does not always hold. What differentiates the various methods available is the way in which they define the similarity metric.

One widely-used class of algorithms employs dynamic programming as the main approach for sequence comparison [14]. In this context, an *optimal* alignment is sought under some scoring scheme and allowing edit operations that include gap opening and elongation. Dynamic programming algorithms are global ("distance-based") [15], or local ("similarity-based") [16]. Another representation of similarity can be produced by the so-called "dot-plot" [17], which visually displays identical or similar residues across a matrix.

---

[1]IBM Thomas J.. Watson Research Center, Yorktown Heights, NY 10598, USA.

[2]Corresponding Author.

[3]Also: Courant Institute Of Mathematical Sciences, New York University, New York, NY 10012, USA.

[4]EMBL Cambridge, The European Bioinformatics Institute, Hinxton Genome Campus, Cambridge CB10 1SD, England, UK.

The second class of methods used to detect common patterns in sequences is pattern identification. In this discussion, the word pattern is used to refer to a specific set of amino acids (or nucleotides) with a given spatial arrangement, typically described as regular expressions [18].

An example of one such pattern is AF..H..RR where the dots are used to indicate that the respective positions could be occupied by "any" amino acid ("don't care" symbol). The pattern is said to match a protein at a given position *i*, if and only if the amino acids of the pattern all match the corresponding amino acids of the protein when placed at offset *i*. For example, AF..H..RR is said to match HWIRTAFLKHAARRIKWL at position 6.

In the early days of sequence comparison, some of the methods to compare two proteins were based on the concatenation of sequences and the identification of common repeats [21]. Pair-wise sequence comparison can thus be reduced to the problem of identifying repeats in the concatenated string of two sequences, while multiple sequence comparison is equivalent to the identification of common repeats in the concatenated string of the full set of sequences.

The assumption behind pattern discovery approaches is that a pattern that appears often enough is expected to play a role in defining the respective proteins' functional behavior, and/or evolutionary relationships.

A number of such algorithms have been steadily appearing in the bibliography the past few years; the type of patterns that each of them is able of handling varies and ranges from regular expressions to simple strings; a survey of several of these algorithms can be found in [1]. In most cases, the proposed methods rely on alignments.

The use of alignment in both classes of algorithms suffers from several drawbacks. First, from a computational standpoint, the problem of optimally aligning a set of strings has been shown to be NP-complete [4] thus making impractical the use of the approach with more than a few sequences. Consequently, most methods incorporate various heuristics in an effort to sidestep the computational complexity of the problem. In most cases (e.g. [2,3]) the overall alignment is built incrementally: initially, small subsets of sequences are aligned to produce a set of consensus sequences. The algorithm is repeated recursively on the generated consensus sequences until only one sequence remains.

A second drawback relates to the fact that occasionally two sequences may have an "intersection" (i.e. they share only weak, local similarities) but no substantial alignment can be produced.

Finally, domain swapping is a thorny problem for methods that rely on alignment. For example two sequences may contain three common domains, say a, b, and c but one of the sequences contains the three domains in the order a-b-c whereas the other contains them in the order c-b-a. In the absence of any other information there is no reason why the alignment stage would prefer one of the possible three alignments over the other two that are possible.

As mentioned above, not all pattern discovery algorithms rely on alignment in order to produce their results. Typically though, one or more of the following constraints is in effect: knowledge of the length or maximum length of the sought pattern, use of restrictive scoring schemes, a requirement for the presence of enough copies of the patterns to be discovered, requirement for the user to decide on a minimum length, some prior knowledge that the sequences under consideration are related, use of restricted size alphabets, maximum possible size for a pattern that is present, gap penalties usually arbitrarily chosen.

In this discussion we present results obtained using *Teiresias*, a novel pattern discovery algorithm [6]. The algorithm is a deterministic algorithm that has the ability when given a set of event streams in any alphabet of N+1 characters (N letters + space) to discover *all* the patterns occurring at least twice in *any relative arrangement* within the set of event streams.[5,6] What this means is that the algorithm will discover *anything* that appears at least twice (or any number of desired times that is greater than 2). By "anything" here we mean that we impose no

[5]Note the there can also be a single event stream at the input, in which case *Teiresias* will identify what is typically referred to as *internal repeats.*

[6]A pattern is identified to be any combination of characters (here, amino acids) and "don't " care characters with any spatial extent, which appears at least twice in the input under consideration.

restriction on the number of characters that will participate in the pattern; furthermore, we impose no restriction on the relative arrangement that the characters will have. In other words, a pattern is defined to be any combination of amino acids and "don't care" characters with any spatial extent, which appears at least twice in the input under consideration..

Let us illustrate the above with a specific example. Figure 1 shows an event stream of 2,000 characters from the English alphabet. In this stream, an adversary has hidden a phrase several times and has told us nothing as to the length or the composition of the phrase. The only thing that we have been told is that the phrase was hidden a minimum of two times.

```
>example 1
QOJPYBHHLSJUEXDFNYHLDSUXGTDLYHTUBLOCNDCXXIGUYQLQGLYFQJWGFPUTKVRYFNNJQXEQXKJYKAMNONRFQA
OYSGTOLWANADTLAFCIMFCMUPRMRATTEGEZSMZFANOTCTVYDAQPJXMSNQQZKUTFXLDYMXOCIGIZRMCYPKVTVBMN
LXYOHNDRCKFHEVGUQQEHQTQGFCXJDYACDJXKSSTFBAMUSBTJEQANMDICACCLXMCSUARASAGTEXVTVIEZEOAQWF
DAFPSITLOKANEDXOAJCWLNCVUBRCSAOTEIBHAUDOPHXFIZNBFMJRRHGYPPZGUNDKKUHJGSBHPIXYLBAXMPLOCG
DVKXTSGSDXDPNZHWIPAZHEDWEVFPJNSNSDBLHXRCJCGJXHTNIQCQKOYFFNKZMPNHGKCVGCBCOOACSQWXJFRDJY
CINMPLUTNSXGKMOBZVKDHTMXXWJGXVJGCYWLDDXLIOPDPBUXEJNBPZDKDPPVYXPHFTHYXRUPCXVVXIQDWMGDTM
WBMBDDXTPCZBXJPWMZJRMKEFFYADPSKTTSFANCDQXAICHJHCOUHRGBALTEOKCZZXETQMUOKFBAWPSATMWGANED
OIAMCVFMCCUZRWSAJTEBKBYUOEEVPWQXAIXXIESOKKLXJTZUDEUIFSHTHZYFVIFJLPYCLZKYTWRWTAWKYYPNRI
EAARSLIZNWWCFORKISDYYCCESGJTYWPYUWRWUEMLWXZQDLBZVJMIOOSQAIJDETDAHVRFPZKWOAJHALBMSBCMNI
PILEDYEBTXELJJWEFCEGTXGIHMJOOSCKFGIRRRYOOUZNNAJOZDKHLVTTOCYGWVILKQKDLGRSKRMIBUBHETTDDO
TWFGKTYEVNZVTTLCLZBCOKCQDTDPJJSVSZIZQRBFPYOKBCMWSPPAAPQPYJBLAKHWVNPYYPMPFIEKRNXINYIQKC
SZVYQHUWCZSNTFWXXNDNVLILVCRFOYDMRUPOAOPDFCSLVSRCV
```

Figure 1. An event stream of 2,000 characters which contains a hidden phrase of unknown length and composition at least two times. Each circle denotes the beginning of an occurrence of the hidden phrase.

A simple, straightforward approach would be to exhaustively generate first all strings of length l_{min} and check them against every possible location of the stream, then all strings of length l_{min}+1, all strings of length l_{min}+2, and so on. If the stream contained N=2000 characters, the minimum length was l_{min}=10, and $|\Sigma|$=27 (26 letters plus space), then ignoring boundary conditions, just the first iteration would require $N|\Sigma|^{l_{min}} \sim 5.4^{13}$ operations! For the shown stream, the embedded phrase has a length of 34, making such an approach prohibitive.

Teiresias can however determine the phase and the locations where it appears in a matter of seconds on an IBM Power-PC workstation using only a few Megabytes of main memory. The hidden phrase is "F.A..S.T...AN.D..A.C...C.U.R..A.TE" (where the dots are occupied by characters of the otherwise random stream) and appears a total of five times in the shown stream.

Even knowing that the hidden phrase begins at the locations marked with the circles (see Figure 1), it is not easy to discover what the phrase is or how long it is. Incidentally, "F.A..S.T...AN.D..A.C...C.U.R..A.TE" is the only collection of letters (and spaces), of *any* length that appears 4 or more times in the stream.

Although we have demonstrated the nature of the algorithm using a single stream of letters, the method directly extends to the case where the input is a collection of several sequences of letters. Also, the algorithm imposes no restriction on the size of the alphabet that the sequences can be built from. We have in fact tried the algorithm on input sequences built from an alphabet of 90 characters with the same high efficiency.

One of the things contributing to the algorithm's strength is the loose definition of what a pattern is. There is no restriction on the symbol composition of the pattern, its spatial extent, minimum length, maximum length or minimum number of occurrences.

Moreover, the algorithm does not make use of any alignment to discover the patterns: as will be shown in the results that follow, in many cases, the relative order of the discovered patterns will be different for different subsets

of the input set. For example, if *a*, *b*, and *c* denote three patterns, it may happen that several of the input sequences will contain the patterns in the order *a-b-c*, a second set will contain the patterns in the order *b-a-c* and a third set will contain the patterns in the order *c-b-a*. Any approach that relies on alignment is bound to perform poorly on such a data set; and this will be particularly true if the three occurring patterns contain only a few amino acids while at the same time spanning a large portion of the sequence (=weak patterns).

Another characteristic of the algorithm is that, although it is deterministic, it does not require any search of the corresponding search space; although counterintuitive, such a thing is possible and contributes to the algorithm's speed. And, of course, in this manner, the algorithm requires no clustering or classification steps and thus avoids carrying out any statistical evaluation of the input sequences.

The rest of the presentation is structured as follows. Sections 2 through 4 present and discuss results using three different input sets. Each section contains an analysis of the discovered patterns from a biological standpoint. We in essence showcase the power of the algorithm by identifying already known patterns as well as by finding new, previously unknown ones. We also show how the algorithm can be used to extract very weak and long internal repeats (correlations). In Section 5, we recapitulate the observations made in each of the previous sections and the presentation concludes with a discussion of the results.

2. Core Histones H3 and H4

Core histones have been suspected to be homologous for a long time [19], but it was only recently that detailed analyses established this evolutionary relationship [20]. All histones and the eukaryotic transcription factors CBF-A/C have been found to contain sequence patterns that are characteristic of the core histone fold [20]. These patterns contain, among other motifs, the pentapeptide KRKT[IV], towards the end of the C-terminal domain [20]. To demonstrate the power of the Teiresias algorithm, we have analyzed the core histone families H3 and H4 and compared the results to those determined previously.

For the purpose of carrying out the experiments, we generated an input set containing 19 sequences of the H3 family and 18 sequences of the H4 family. The PIR (Rel. 47) [24] names of the resulting set appear in the table below.

HSHU33	HSCH3	HSRK3	HSTR3	HSFI3	HSXL31	HSUR3M	HSUR3P
HSBO3	HSEAH3	HSKW3	HSPM3	HSBH3[1]	HSXL32	HSZP3	HSBY3
HSVK3L	HSDK34	HSMS34	HSHU4	HSBO4	HSPG4	HSRT4	HSCH4
HSTR4	HSXL4	HSUR4	HSUR4P	HSFF4	HSZM4	HSWT4	HSPM4
HSWT41	HSBY4	HSZP4	HSTE42	HSTE41			

Table 1. The PIR labels of the 37 sequences from the H3 and H4 families that were used in the experiments.

Processing the input set with *Teiresias* took place in a matter of a few minutes on an IBM Power-PC workstation and a large set of patterns was discovered. The discovered set comprised patterns that were common to both families as well as patterns that were common to only the members of a single family. In Figure 2, we are showing *all* discovered patterns which occurred in at least 34 (out of the possible 37) sequences. The patterns have been aligned with one representative member of the H3 and H4 family in order to clearly show the patterns' positions in the set. Notice that for several pairs of patterns, the relative order in the two families is not preserved (Figure 2).

The pattern T...I.....V..I.....R (see Figure 2) which is the only one appearing in all of the H3 and H4 family members of the input set can be found in 61 entries in the latest Non-Redundant Database (effective date 26 February 97 containing approximately over 62 million amino acids). Of those 61 entries, 51 are histones, whereas the remaining 10 proteins include the proteins: P15699 / AMDR_EMENI ACETAMIDASE REGULATORY PROTEIN, Q06943 / HMGZ_DROME HIGH MOBILITY GROUP PROTEIN Z, P36733 / VL1_HPV12 MAJOR CAPSID PROTEIN L1, P50790 / VL1_HPV24 MAJOR CAPSID PROTEIN L1, and Q06895 / VP4_ROTHY

OUTER CAPSID PROTEIN VP4; also included are four more hypothetical proteins, and a mutant protein sequence from PDB (1MAS-B). Ironically enough, HMG-Z belongs to the so-called *non-histone* chromosomal proteins and may represent a false positive. Yet, the pattern T...I.....V..I.....R is very specific, and identifies *all* histones in the Non-Redundant Database (many more than the original set on which *Teiresias* operated). Furthermore, the pattern is present in proteins which are either DNA-binding, viral, or hypothetical and thus in essence represents predictions of either evolutionary or functional similarity.

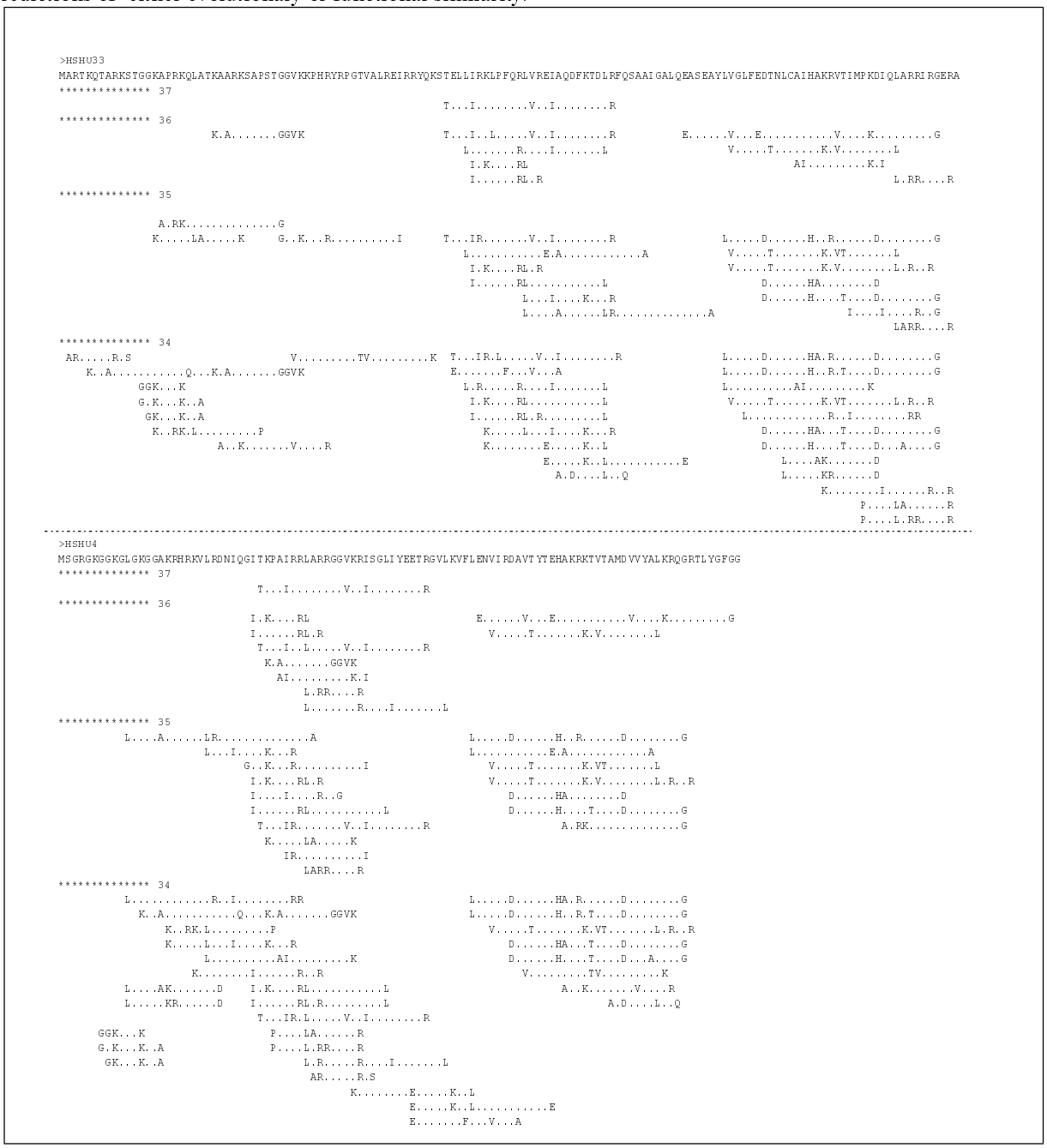

Figure 2. Several of the discovered patterns that are common to both the H3 and H4 families. The patterns are shown aligned with one protein from the H3 family (HSHU33) and one protein from the H4 family (HSHU4). Note that we show only the patterns that occurred in at least 34 sequences of the input set's 37 sequences (=19 H3 + 18 H4 members). It is interesting to observe that for several of the discovered patterns the relative order has not been preserved during evolution.

In Figures 3 and 4 we are showing only the patterns that were discovered as belonging to all of the members of the respective family: the patterns are aligned with the sequences HSHU33 and HSHU4 respectively. Notice that, in the case of the H3 family, there are no patterns shown aligned with the first half of HSHU33. The reason for this is the presence of the sequence HSHB3 which is an 80 amino acid fragment beginning with "STELLI."

A second comment relates to the first two patterns shown aligned with HSHU4. The second pattern, A.L.....IH, appears as if it can be combined with the first one to produce a "fuller" one; however, it cannot. Indeed one of the sequences, namely HSMS34, contains the pattern A.L.....IH in a region which only partially overlaps the one containing the first longer pattern thus forcing the algorithm to report them as two distinct pattern appearing in the same number of sequences.

The set of all the patterns that have been discovered in this set of core histones are reported in [8].

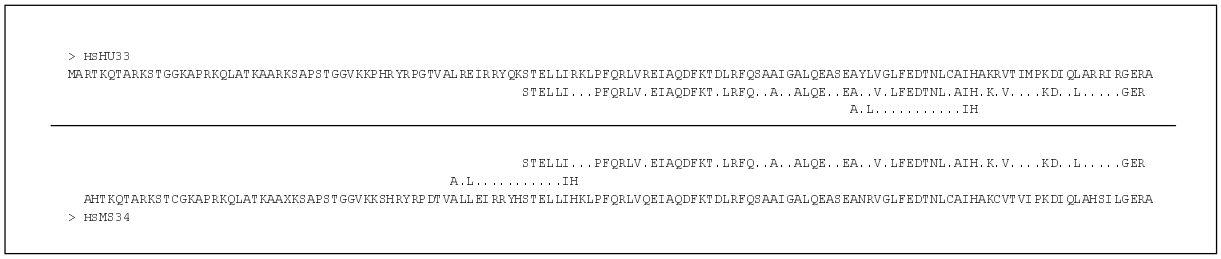

Figure 3. All of the discovered patterns that are common to *all* of the sequences of the input set that belong to the H3 family. The patterns are shown aligned with the sequences HSHU33 and HSMS34. Notice the location of the pattern A.L.....IH in these two member sequences (see also text regarding the presence of a fragment in the input set).

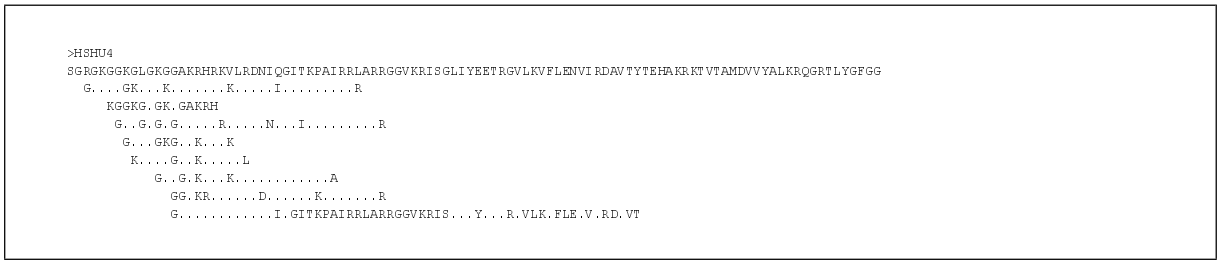

Figure 4. All of the discovered patterns that are common to *all* of the sequences of the input set that belong to the H4 family. The patterns are shown aligned with the sequence HSHU4 (see also text).

3. Leghemoglobins

Leghemoglobins are plant hemoglobins that show sequence similarity to other members of the large globin superfamily, especially to myo- and hemo-globins from animals [22]. All globins contain a conserved-histidine signature that represents the heme binding pocket [22].

BP1_CASGL	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			

Table 2. The SwissProt [25] labels of the 21 leghemoglobin sequences as they appear in the respective entry of PROSITE [23], Release 13.

We presented *Teiresias* with a set of leghemoblogins (Table 2), and the intention to determine whether the algorithm could identify the heme-binding-pocket signature successfully, or any other important pattern that has escaped attention. The algorithm took a few tens of seconds and generated a large number of patterns.

Given the generated patterns and assuming a simple-minded *iid* model for the presence of each amino acid in a given pattern, we used the amino acid frequencies observed naturally to attach a probability to each discovered pattern; the patterns were then ranked in order of increasing likelihood (i.e. most-*unlikely* first).

The pattern A.L.T.K.....W.....AY..L....K was ranked first and was determined that it uniquely identifies all leghemoglobins from the Non-Redundant Database, without a single spurious hit! The algorithm also determined the pattern corresponding to the heme-binding-pocket signature which was ranked second, after the C-terminal pattern 'A.L.T.K.....W.....AY..L....K'. The PROSITE [23] database lists the heme-binding motif as the family signature, and this has also been discovered by *Teiresias* by operating on the set of Table 2.

Figure 5 shows only those patterns that have been determined by the algorithm to be common to at least 20 of the 21 leghemoglobin sequences of the input set shown in; the patterns are shown aligned with HBPL_PARAD.

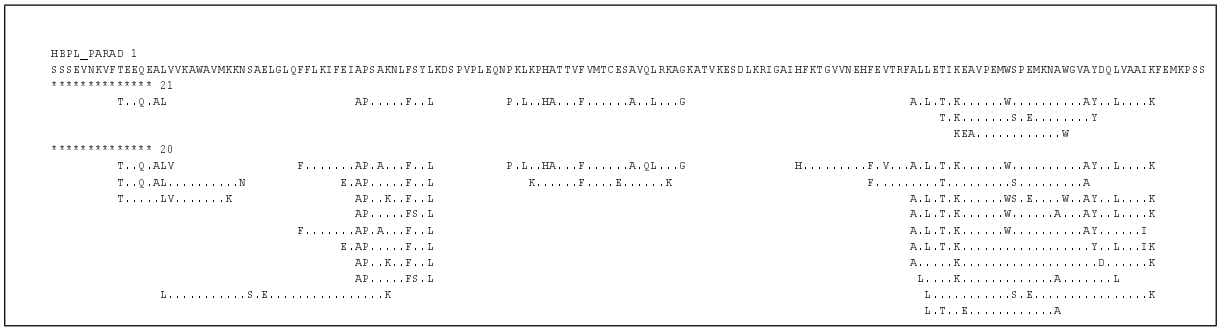

Figure 5. All of the discovered patterns that are common to at least 20 of the 21 sequences in the leghemoglobin input set (see Table 2).

The set of all the patterns that have been discovered in this set of leghemoglobins are reported in [9].

4. The Protein JQ0870

In this last section we will report on several results that were obtained using a single protein as an input. As mentioned previously, in cases like this, the *Teiresias* algorithm identifies and reports all internal repeats that are present in the sequence.

The PIR label of the sequence under consideration is JQ0870 and its amino acid synthesis (612 amino acids) is shown in Table 6.

>JQ0870 hypothetical 69.4K protein (iclR 3' region) - *E. Coli*
MEILPQRSDTLDKQELDLKSDRKEKEFPRIKLNGQCYFPGRPQNRIVCRHIAAQYINDIYQNVDYKPHQDDYSSAE
KFLTHFNKKCKNQTLALVSSRPEGRCVAACGDFGLVMKAYFDKMESNGISVMAAILLVDNHALTVRLRIKNTTEGC
THYVVSVYDPNVTNDKIRIMSETKENIKHYSLMDFMNVDYSLLKWSNDHVINQSVAIIPALPKEQLLMLKGSVDEI
TPPLSPATMNLLMAIGQNHQLTQLMIQLQKMPELHRTEMLTAYNSINLPGLYLAINYGNADIVETIFNSLSETGYE
GLLSKKNLMHILEAKDKNGFSGLFLAISRKDKNVVTSILNALPKLAATHHLDNEQVYKFLSAKNRTSSHVLYHVMA
NGDADMLKIVLNALPLLIRTCHLTKEQVLDLLKAKDFYGCPGLYLAMQNGHSDIVKVILEALPSLAQEINISASDI
VDLLTAKSLARDTGLFMAMQRGHMNVINTIFNALPTLFNTFKFDKKNMKPLLLANNSNEYPGLFSAIQHKQQNVVE
TVYLALSDHARLFGFTAEDIMDFWQHKAPQKYSAFELAFEFGHRVIAELILNTLNKMAESFTQKSISPYRTLNLCL
RRYA

Table 6. The amino acid sequence of JQ0870.

Quite a few internal repeats were identified in this sequence, including the ankyrin-like repeats that were originally identified in [10]. All discovered repeats can be found in [7]. We herein present two *very long* repeats contained

in JQ0870 that, to our best knowledge, have not been reported previously. Both of these repeats have a multiplicity of two.

The first of the two repeats contains 82 amino acids and spans 278 positions; its two copies overlap one another and share 36 amino acids. The second of the patterns contains 57 amino acids and spans 201 positions; this pattern's two copies share a mere 7 amino acids! An additional interesting observation is that both occurrences of the first pattern overlap both occurrences of the second pattern: in the region where *all four* instances intersect, the copies share 3 amino acids only! Figure 7 shows both copies of these two patterns together with their alignment with the sequence.

There is two additional comments that relate to JQ0870 and were derived by studying some of the discovered patterns that are reported in [7]:
(a) the sequence contains two copies of the hexapeptide "PGLYLA"; this hexapeptide appears in no known protein other than JQ0870 as a search on the Non-Redundant Database has shown;
(b) the sequence also contains two copies of the run "LY..M.NG..D" which as was the case with PGLYLA appears in no known protein of the Non-Redundant Database other than JQ0870.

```
>>> MEILPQRSDTLDKQELDLKSDRKEKEFPRIKLNGQCYFPGRPQNRIVCRHIAAQYINDIYQNVDYKPHQDDYSSAEKFLTHFNKKCKNQTLALVSSRPE
1-1
1-2
2-1
2-2
>>> GRCVAACGDFGLVMKAYFDKMESNGISVMAAILLVDNHALTVRLRIKNTTEGCTHYVVSVYDPNVTNDKIRIMSETKENIKHYSLMDFMNVDYSLLKWS
1-1
1-2
2-1
2-2
>>> NDHVINQSVAIIPALPKEQLLMLKGSVDEITPPLSPATMNLLMAIGQNHQLTQLMIQLQKMPELHRTEMLTAYNSINLPGLYLAINYGNADIVETIFNS
1-1                                                                    L.A.......GL.LAI.......V..I.N.
1-2
2-1                                                  L.I.............L.A......PGLYLA...G..DIV..I...
2-2
>>> LSETGYEGLLSKKNLMHILEAKDKNGFSGLFLAISRKDKNVVTSILNALPKLAATHHLDNEQVYKFLSAKNRTSSHVLYHVMANGDADMLKIVLNALPL
1-1 L........L........L.AK.......L.......D.......LNALP.L..T.HL..EQV...L.AK.......LY..M.NG..D..K..L.ALP.
1-2                   L.A.......GL.LAI.......V..I.N.L........L........L.AK.......L.......D.......LNALP.
2-1 L.....E...S.......L.AK......GLF.A..R...NV...I.NALP.L..T...D.........A.N......L.............V..AL..
2-2                                                                                                    L
>>> LIRTCHLTKEQVLDLLKAKDFYGCPGLYLAMQNGHSDIVKVILEALPSLAQEINISASDIVDLLTAKSLARDTGLFMAMQRGHMNVINTIFNALPTLFN
1-1 L............DLL.AK......GL..AMQ.GH......I..ALP.L.............LL.A.......GLF.A.Q....NV..T...AL.....
1-2 L..T.HL..EQV...L.AK.......LY..M.NG..D..K..L.ALP.L............DLL.AK......GL..AMQ.GH......I..ALP.L..
2-1 ..R....T.E...D....K.........LA...GH..I...IL..L...A.............L
2-2 .I.............L.A......PGLYLA...G..DIV..I...L.....E...S.......L.AK......GLF.A..R...NV...I.NALP.L..
>>> TFKFDKNMKPLLLANNSNEYPGLFSAIQHKQQNVVETVYLALSDHARLFGFTAEDIMDFWQHKAPQKYSAFELAFEFGHRVIAELILNTLNKMAESFT
1-1 .F.F...............Y.....A.........E.....L...A..F
1-2 ...........LL.A.......GLF.A.Q....NV..T...AL......F.F.................Y.....A.........E.....L...A..F
2-1
2-2 T...D.........A.N......L..............V..AL....R....T.E...D....K.........LA...GH..I...IL..L...A....
>>> QKSISPYRTLNLCLRRYA
1-1
1-2
2-1
2-2 .........L
```

Figure 7. Two of several very long internal repeats (1-1/1-2 and 2-1/2-2) discovered in sequence JQ0870; each of these repeats occurs twice in the sequence (see also text).

5. Conclusion

In the previous discussion, we presented results obtained with *Teiresias*, a new pattern discovery algorithm. The algorithm was applied to three distinct sets of inputs and the objectives were different each time.

The first input set consisted of a mix of several core histones 3 and 4 and the generated results demonstrated that the algorithm, in the absence of any biological context information, is able to derive results identical to those that have been obtained previously through different approaches.

The second set was designed in a such a way as to show whether the algorithm could be used to determine a known signature (the heme-binding pocket) of the set as well as any patterns that had previously escaped detection. *Teiresias* correctly discovered the common signature of the leghemoglobins as well as a new pattern that is also specific to leghemoglobins.

Finally, we showcased the algorithm by discovering two *very long* patterns that have not been previously observed. Also, we identified two short fragments that are unique to the protein used as the input and appear in no other known protein.

We believe that the presented results showcase the versatility of the *Teiresias* algorithm and its ability to correctly and efficiently discover patterns of significance contained in the input.

REFERENCES

[1] Brazma, A., I. Jonassen, I. Eidhammer, and D. Gilbert, "Approaches to the automatic discovery of patterns in biosequences", TR No 113, Department of Informatics, University of Bergen, Norway, 1995.

[2] Roytberg, M.A, "A Search for Common Patterns in many sequences", CABIOS, 8(1):57-64, 1992.

[3] Smith, R.F., T.F. Smith, "Automatic Generation of Primary Sequence Patterns from Sets of Related Protein Sequences", Proceedings National Academy Of Science (USA), pp. 118-122, 1990.

[4] Wang, L., and T. Jiang, "On the Complexity of Multiple Sequence Alignment", Journal of Computational Biology, 1(4): 337-348, 1994.

[5] Wang, J.T.L., T.G. Marr, D. Shasha, B.A. Shapiro and G.-W. Chirn, "Discovering Active Motifs In Sets Of Related Protein Sequences And Using Them For Classification," Nucleic Acids Research, 22(14):2769-2775.

[6] Rigoutsos, I. and A. Floratos, "*Teiresias*: An Algorithm For The Discovery Of Patterns In 1-Dimensional Event Streams." IBM Thomas J. Watson Research Center Technical Report. Spring 1997.

[7] Rigoutsos, I., and A. Floratos, "Preliminary Results On The Discovery Of Patterns Of Amino Acids In The Protein JQ0870." IBM Thomas J. Watson Research Center Technical Report. Spring 1997. In Preparation.

[8] Rigoutsos, I., A. Floratos, and C. Ouzounis, "Preliminary Results On The Discovery Of Patterns Of Amino Acids Common To Sequences Of Core Histones 3 and 4." IBM Thomas J. Watson Research Center Technical Report. Spring 1997.

[9] Rigoutsos, I., A. Floratos, and C. Ouzounis, "Preliminary Results On The Discovery Of Patterns Of Amino Acids Common To Sequences Of Leghemoglobins." IBM Thomas J. Watson Research Center Technical Report. Spring 1997.

[10] Neuwald, A.F. and P. Green, "Detecting Patterns In Protein Sequences," Journal Of Molecular Biology, 239, 698-712, 1994.

[11] Waterman M.S., D.J. Galas, R. Arratia, "Pattern Recognition In Several Sequences: Consensus And Alignment," Bulletin Of Mathematical Biology, 46, 515-527, 1984.

[12] Bork P, C. Ouzounis, and C. Sander, "From Genome Sequences To Protein Function," Current Opinion In Structural Biology, 4, 393-403, 1994.

[13] Smith T.F., M.S. Waterman, and W.M. Fitch, "Comparative Biosequence Metrics," Journal Of Molecular Evolution, 18, 38-46, 1981.

[14] Delcoigne A, and P. Hansen, "Sequence Comparison By Dynamic Programming," Biometrika, 62, 661-664, 1975.

[15] Needleman S.B., and C.D. Wunsch, "A General Method Applicable To The Search For Similarities In The Amino Acid Sequence Of Two Proteins," Journal Of Molecular Biology, 48, 443-453, 1970.

[16] Smith TF, and M.S. Waterman, "Identification Of Common Molecular Subsequences," Journal Of Molecular Biology, 147, 195-197, 1981.

[17] Gibbs A.J., and G.A. McIntyre, "The Diagram, A Method For Comparing Sequences," European Journal Of Biochemistry, 16, 1-11, 1970.

[18] Aho V.A., J.E. Hopcroft, and J.D. Ullman, "The Design And Analysis Of Computer Algorithms," Addison-Wesley, Menlo Park CA, 1974.

[19] Reeck GR, E. Swanson, and D.C. Teller, "The Evolution Of Histones," Journal Of Molecular Evolution, 10, 309-317, 1983.

[20] Ouzounis C.A., and N.C. Kyrpides, "Parallel Origins Of The Nucleosome Core And Eukaryotic Transcription From Archaea," Journal Of Molecular Evolution, 42, 234-239, 1996.

[21] Martinez H.M., "An Efficient Method For Finding Repeats In Molecular Sequences," Nucleic Acids Research, 11, 4629-4634, 1983.

[22] Kapp O.H., L. Moens, J. Vanfleteren, C.N. Trotman, T. Suzuki, and S.N. Vinogradov, "Alignment of 700 Globin Sequences: Extent Of Amino Acid Substitution And Its Correlation With Variation In Volume," Protein Science, 4, 2179-2190, 1995.

[23] Bairoch A, P. Bucher, and K. Hofmann, "The PROSITE database: its status in 1995," Nucleic Acids Research, 24, 189-196, 1996.

[24] Sidman K.E., D.G. George, W.C. Barker, and L.T. Hunt, "The Protein Identification Resource (PIR)," Nucleic Acids Research, 16(5):1869-71, 1988.

[25] Bairoch A., and Boeckmann B. "The SWISS-PROT Protein Sequence Data Bank: Recent Developments," Nucleic Acids Res. 21:3093-3096(1993).