Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151, available online at http://www.idealibrary.com on IDEAL®

The Emergence of Pattern Discovery Techniques in Computational Biology

Isidore Rigoutsos,[1] Aris Floratos, Laxmi Parida, Yuan Gao, and Daniel Platt

Bioinformatics and Pattern Discovery Group, IBM TJ Watson Research Center, Yorktown Heights, New York 10598

Received November 5, 1999; accepted April 3, 2000; published online September 28, 2000

In the past few years, pattern discovery has been emerging as a generic tool of choice for tackling problems from the computational biology domain. In this presentation, and after defining the problem in its generality, we review some of the algorithms that have appeared in the literature and describe several applications of pattern discovery to problems from computational biology. © 2000 Academic Press

1. INTRODUCTION

Recent years have witnessed an emergence of pattern discovery methodologies for solving numerous tasks which arise in computational biology. Known also as "data mining" approaches, they represent a novel approach for extracting useful information from databases containing various types of biological information.

Initially, dynamic programming techniques were applied to the analysis of biological sequences and to the determination of sequence similarity between a query sequence and one or more biological sequences (DNA, proteins, and fragments) from a collection. Subsequent studies of such sequence similarity revealed conserved functional and structural signals, thus making the argument for the usefulness of such approaches. Research effort spanning almost two decades gave rise to a number of useful algorithms and an abundance of interesting scientific results [2, 58, 63, 77].

Almost in parallel, researchers began looking into other approaches in an effort to develop concise consensus sequences that captured and represented regions of similarity across several sequences presumed to be related. A large number of early methods relied on *multiple string alignment* [19, 25, 44, 70] as the method of choice for discovering these regions [22, 54, 58, 60, 80, 89, 91]. The related sequences could be transformed to one another through permissible *edit* operations (e.g., mutations, insertions, deletions) each of which had an associated cost. Alignment-based methods had the drawback of imposing an alignment of all the sequences in the processed input; additionally, they worked best only if the involved parameters were optimized for the set being considered [4].

Pattern discovery techniques were introduced to alleviate the problems associated with multiple sequence alignment and algorithms have been steadily appearing in the bibliography [46, 59, 65, 66, 71, 78, 79, 83, 86]. Essentially, these algorithms seek to determine one or more patterns that represented one or more blocks of related sequences. In some cases, these algorithms are used to compute the cardinality and the boundaries of conserved blocks within groups of related sequences [41, 42, 50], build profiles [16, 36], build HMMs [48, 81], or generate regular expressions that characterized and described sequence regions of interest [5, 8, 32]. Additional applications included the use of pattern discovery techniques to solve other NP-hard problems [34] such as multiple sequence alignment [56, 61, 62, 75], the determination of tandem repeats in DNA stretches [82], etc.

More recently, discovery techniques began being applied to other problems from computational biology that departed from the traditional sequence analysis work. These problems include text mining, structure characterization and prediction, promoter signal detection, gene expression analysis, and others [9, 10, 17, 40, 45, 67, 74].

In this paper, we present a moderately detailed discussion of related work that appeared in recent years as well as describe some of the algorithms and applications in whose development we have been involved. Clearly, we cannot provide a comprehensive coverage of all related work in this field; consequently, and whenever possible, we will refer the reader to review articles.

2. THE GENERAL PROBLEM

What we are typically presented with is a database D composed of one or more records. The records can have the same or different arbitrary lengths and can be thought of as streams of possible "*events*" chosen from a set E of

[1] To whom correspondence should be addressed. E-mail: rigoutso@us.ibm.com.

1096-7176/00 $35.00

permissible events. Examples event sets E include but are not limited to the following: letters from a fixed alphabet (e.g., the 26 letters of the English language, 4 nucleotides, 20 amino acids), words from an allowed vocabulary (e.g., words from a natural language), categorical values from a finite collection (e.g., {red, green, blue, white, black} or {pop, jazz, R&B, hip-hop}), integer numbers from a set of bounded cardinality (e.g., $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$), real numbers from a finite interval (e.g. $[-5, +7]$), etc.

Frequently, there also exists an evaluation function F whose *domain* is the set $E \times E$ and *range* the set $\mathbb{R}$ of real numbers. F can be thought of as a function that quantifies the quality of similarity between any two events from the set E. This function can be a distance metric in the strict sense[2] but several functions have also appeared in the literature that do not satisfy the commutative law. Obviously, the exact form of the function F is dependent on the problem domain and the set of permissible events [21, 35, 43, 76].

Given the database D, the event set E, and the evaluation function F, the task at hand is to determine *interesting* combinations of events which are contained in D. However, neither is the subset of events participating in the interesting combinations known nor the exact form of these combinations. Ideally, the task is to be carried out in the absence of domain-specific knowledge and this makes the notion of *interesting* poorly defined.

One way to address these problems is to recast the notion of interesting in terms of the number of times some combination of events appears. Thus, a combination of events will be considered interesting if and only if it appears a minimum number of times in the processed input database. This is called the "threshold" or minimum "support."

Even with this definition in place, determining interesting combinations in the input is not a straightforward task. Additional needed elements include the definition of the *nature* of allowed patterns and their minimum allowed *density*.

The nature of the allowed patterns is defined with the help of regular expressions whose complexity has traditionally been used as a key differentiator among the various algorithms that appeared in the literature. Typically, and unless the nature of sought patterns is extremely simple, detecting all patterns that exist in an input of length N is an NP-hard problem.[3] Examples of regular expressions that reflect the types of patterns that are captured by previously proposed algorithms include E^+, i.e., one or more *consecutive* events; $E(E \cup \{.\})^* E$, i.e., two or more events separated by an arbitrary number of events or wild-card characters; $E([EE^*E] \cup \{.\} \cup E)^* E \cup E$, i.e., a single event or two or more events separated by any combination involving a single event, a wild-card, or a choice among two or more possible events. The character "." in these expressions is the wild-card or don't care character and is used to denote a position occupied by an event that remains unspecified; for example, K...[ILMV]..H is used to denote an instance of event K followed by *any three* permissible events followed by any one of I, L, M, or V followed by any two permissible events and terminating in H.

As for the minimum allowed density, this is typically defined as a minimum allowed value for the ratio of the positions that are occupied by non-wild cards over the actual span of an instance of the pattern. The concept of density can be either implicit to the algorithm or explicit in which case it can be deduced from the values of the parameters passed to the algorithm. In earlier work [65, 66], we introduced the concept of the $\langle L, W \rangle$ pattern and this is the definition that we will also use in this discussion. A pattern P will be called an $\langle L, W \rangle$ pattern (with $L \leqslant W$) if and only if any L consecutive literals (i.e., non-wild cards) span at most W positions.

A final requirement for defining the set of possible solutions is that of the minimum required support: an allowed combination of events, i.e., a pattern P will not be contained in the set of reported solutions unless there exist at least K instances of it in the database D.

To recapitulate, we can define the problem of discovering "interesting combinations" of events as follows: given a database D comprising one or more variable-length streams of events from an event set E, an evaluation function $F: E \times E \to \mathbb{R}$ defined on the event set, and parameters $\{L, W, K\}$, find all $\langle L, W \rangle$ patterns that have at least K instances in D.

The algorithms that have appeared in the literature and have attempted to tackle this problem fall into one of two broad categories. The first category includes algorithms which operate by *enumerating* the solution space; i.e., they *hypothesize* each possible pattern in turn (for example in order of increasing span) and *verify* whether its support exceeds the predefined user threshold. Clearly, the difficulty in enumerating the patterns–solutions increases with the number of positions that the pattern spans; consequently, some discovery algorithms impose restrictions on the maximum length that the discovered patterns can have.

The algorithms in the second category begin with the observation that if the input D contains patterns whose instances span many positions and satisfy the support threshold K, then *fragments* of these longer patterns must also appear K or more times in D. Thus, the algorithms in this category begin by collecting all these seed patterns, a

[2] For F to be a distance metric, it must satisfy the following four conditions: $F(e, e) = 0$; $F(e_1, e_2) \geqslant 0$; $F(e_1, e_2) = F(e_2 e_1)$; and $F(e_1, e_3) \leqslant F(e_1, e_2) + F(e_2, e_3)$.

[3] A reduction form the *longest common subsequence* problem [53] can be used to prove NP-hardness.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

relatively straightforward task, and concentrate on building the final patterns–solutions out of their component seed patterns. Several of the more efficient algorithms follow this approach.

Independent of how the patterns–solutions contained in D are obtained, all algorithms should ideally report all $\langle L, W \rangle$ patterns that have at least K instances in D. In [65, 66] we introduced an additional property that reported patterns may have the "maximality" with respect to *length* and *composition*. A pattern P that is maximal and is reported as occurring K' times in the database D cannot be made more specific either by prepending/appending an event combination to it or by dereferencing any of the wild cards it contains without simultaneously decreasing its support. Another way of describing this maximality property is that any reported pattern P that is claimed to occur K' times in D is as long and as dense as it can be without violating the density constraint that the parameters L and W dictate.

In an effort to control the size of generated output, some of the proposed methods introduce heuristics; others restrict the type of allowed patterns. Finally, a third group employs a measure of information content or importance curbing the reporting to only those of the patterns that exceed a threshold [46, 59, 71, 86]. In all these cases, the performance improves at the expense of reporting an incomplete set of the results.

3. SOME OF THE ALGORITHMIC APPROACHES

We next highlight some representative algorithms among those that have appeared over the past few years. The list is by no means exhaustive and for a discussion from a theoretical standpoint the interested reader should refer to [15]. It should also be noted that any of the algorithms that will be mentioned below can be modified, at least in principle, with moderate effort to address problems outside the immediate computational biology context for which they were developed; we will thus be using the term "event" to refer interchangeably to any of the terms "character," "alphabet symbol," "amino acid," and "nucleotide."

One characteristic of the earlier algorithms was their reliance on first determining a multiple sequence alignment for the input streams and subsequently constructing a consensus sequence from it. Those of the consensus sequence fragments whose support exceeded threshold were reported as the discovered patterns. The natural mapping of the biological operations of mutation, insertion, and deletion to string editing operations made the use of multiple sequence alignment the subtask of choice.

Probably the earliest instance of a pattern discovery recipe is the one appearing in [80] as part of a multiple sequence method: the recipe called for the determination of all substrings exceeding a minimum length and appearing in all of the input streams. To achieve discrimination power, the minimum length had to be substantial at the expense of missing shorter common substrings.

In [79], all *pairings* of input streams are formed and scored. Each pair of similar input streams is then replaced by a pattern that captures the alignment of the pair's members in preparation for the next iteration. All successive iterations of the procedure are carried out on pairs of generated *patterns* until only one pattern (or none) remains. Upon termination, a binary dendrogram can be built with each internal node corresponding to a pattern present in all of the input sequences at the leaves of the subtree rooted at this node.

At about the same time, the work in [78] introduced the MOTIF algorithm for carrying out pattern discovery. MOTIF operates by exhaustively enumerating all L-tuples ($L = 3$) of amino acids that appear in the input set and the distance between the first and last amino acid of the triplet was bounded from above ($W = 21$). Those of the L-tuples with instances exceeding threshold are used as *anchor regions* to induce local alignments and further expand the patterns. A variation of this method can be found in [83].

The method described in [69] begins by selecting a *basic* stream which is then compared with all other streams in the input and any similar segments that appear in at least K streams are determined. Obviously, the quality of the results is dependent on the choice of the *basic* stream.

MOTIF is also the starting point for the ASSET method discussed in [59]. In addition to allowing positions in the L-tuples to be occupied by at most two possible events ("ambiguously" defined positions), the method permits the discovery of rigid patterns that are of arbitrary length. The number of positions that can be occupied by "double characters" should be kept to a minimum to avoid performance degradation, whereas a double filtering (minimum required support and statistical importance) stage furthers speeds things up.

This last approach is combined in [46] with a depth-first search strategy leading to an even more powerful algorithm. The resulting method, PRATT, allows not only for ambiguous positions with more than two possible events but also for flexible gaps. A user-defined, minimum required support as well as parameters controlling the maximum pattern length, the maximum number of components, the maximum number of ambiguous components, and the nature of allowed ambiguous components are used to prune the search tree.

A similar algorithm is described in [71] that also allows for ambiguous positions but permits only rigid gaps. When the cardinality of the event set E is large (e.g., $|E| = 20$), one needs to specify the groups of events that can occupy an

ambiguous position in advance together with a limit on the number of a group's instances in any pattern. To avoid generating redundant results, and when two patterns match the same stream of events, the more specific pattern is the one that is explored further.

DISCOVER is different than the above in that it hypothesizes potential patterns and then verifies them [86, 87]. The algorithm seeks patterns that are the concatenation of several components each of which is a string over the event set E; two successive strings can potentially appear at variable distances from one another. The algorithm employs a *generalized* suffix tree, each internal node of which keeps track of how many of the input sequences contain the string that labels the path from the root of the tree to the node. After determining which of the strings have support that exceeds a predetermined support threshold, the algorithm hypothesizes patterns comprising m string components and verifies which of those combinations satisfy the minimum support requirement. To improve performance, the algorithm makes use of the support that a pattern under consideration obtains from a random *fraction* of the input database and probabilistically decides whether its support by the entire database will exceed the predetermined threshold. Note that valid patterns may be discarded during this step; if a pattern makes it through this step, its support will be recomputed for the entire input database and if it is above threshold then the pattern will be reported. The performance of the approach is acceptable if the starting collection of candidate components is small and the examined patterns do not comprise too many of such components. It is also possible that the final list includes redundant, overlapping, and nonmaximal patterns.

More recently, EMOTIF [60] was proposed as an extension of the work in [91]. The algorithm begins with a multiple sequence alignment (e.g., of the type encountered in the BLOCKS database [41]) and a collection C of sets of events from the power set $2^{|\mathbf{E}|}$; the assumption is that the events of each such set can replace one another without an adverse effect on the function of the protein. This must be taken into account during the pattern discovery process. Beginning with the empty set, patterns are built by considering each of the columns of the aligned input in turn: the pattern that is being built will expand to include the next unvisited column as long as the resulting pattern has support that exceeds the preselected threshold. If the amino acids occupying a column are unrelated (i.e., not captured by any of the sets in C), then a wild-card symbol is introduced for the column; on the other hand, if they can be described equally well by more than one set in C, then the most specific of these sets is used to represent the column. Finally, if the column under consideration is occupied by a single amino acid, then the latter is used to represent the column. Once the pattern is built, EMOTIF can be used to search and enumerate the resulting space in an exhaustive manner: for each of the possible choices within column i that is supported by sufficiently many sequences (i.e., exceeds the predetermined threshold), the respective sequences become propagated to column-$(i+1)$ and the process repeats. Use of either maximization of statistical significance or minimization of entropy allows the ranking of the patterns.

In work carried out in our group, we developed a two-phase algorithm called TEIRESIAS [27, 28, 65, 66]. During a first *scanning phase*, the algorithm compiles a complete collection of *elementary* patterns (elementary L-tuples of events that span not more than W positions in any event stream—clearly, $L \leqslant W$). During the *convolution phase*, the algorithm recursively combines them into increasingly longer patterns of decreasing support that have the property of being maximal with respect to both length and composition. The extent of the reported patterns is of course bounded only by the size of the processed database.

Figure 1 depicts the convolution phase with two arbitrary patterns being combined into a longer one. A *left* and a *right* pattern from the pool of active patterns[4] each of which carries a *position* list indicating where exactly it appears in the processed database may be combined if and only if the left pattern ends in a suffix of n non-wild card events that is identical to a prefix of the right pattern: in the presence of the suffix/prefix agreement, the position lists of the two patterns are also examined for agreement and the pattern resulting from the convolution is given a position list that is the intersection of the position lists of the component patterns.[5] The position lists of the two component patterns are updated and returned to the pool of active patterns if their support continues to exceed threshold. The order in which convolutions are to be performed is decided through the use of two partial orderings, *prefix*-wise and *suffix*-wise,[6] and is essential in avoiding the generation of redundant patterns. Finally, the algorithm guarantees that *all* maximal patterns with support exceeding the predetermined threshold are reported.

In its original description, TEIRESIAS discovered patterns of the type $E(E \cup \{.\})^* E$, i.e., patterns comprising

[4] At any time, the pool of active patterns comprises patterns that are not maximal, have not yet been reported, and whose support still exceeds threshold—initially the pool comprises only the elementary patterns.

[5] To guarantee completeness of results, n must be equal to $L-1$.

[6] Let σ_1 and σ_2 be two events from E and x, y two strings from $(\Sigma \cup \{.\})^*$. Then the prefix-wise partial ordering is defined as follows: (a) $\sigma_1 x <_{\text{pf}} \varnothing$, $\varnothing <_{\text{pf}} \cdot x_1$, $\sigma_1 x <_{\text{pf}} \cdot y$; (b) $\sigma_1 x <_{\text{pf}} \sigma_2 y$ if and only if $x <_{\text{pf}} y$ and; (c) $\cdot x <_{\text{pf}} \cdot y$ if and only if $x <_{\text{pf}} y$. The suffix-wise partial ordering is defined analogously as (a) $x\sigma_1 <_{\text{sf}} \varnothing$, $\varnothing <_{\text{sf}} x\cdot$, $x\sigma_1 <_{\text{sf}} y\cdot$; (b) $x\sigma_1 <_{\text{sf}} y\sigma_2$ if and only if $x <_{\text{sf}} y$ and; (c) $x\cdot <_{sf} y\cdot$ if and only if $x <_{\text{sf}} y$.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

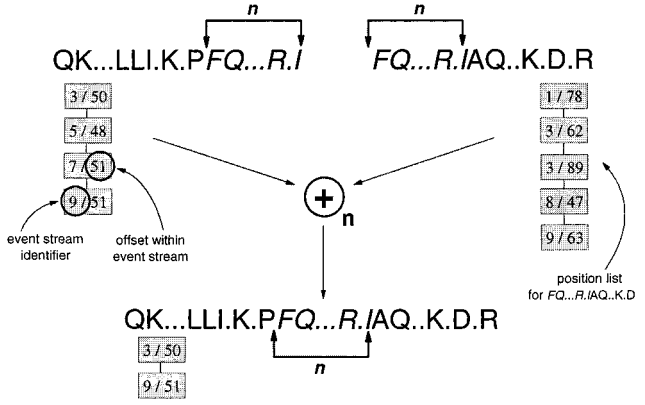


FIG. 1. The convolution phase of TEIRESIAS.

either individual events or the don't care character. However, it is frequently desirable to carry out the discovery process by permitting events from a small collection to substitute for one another. For example, one may wish to allow for the positively charged arginine to replace the positively charged lysine and vice versa or allow for any of the hydrophobic amino acids leucine, methionine, isoleucine, and valine to indistinguishably replace one another. We have thus further developed the algorithm to allow for the discovery of patterns in the presence of a collection C of sets of events from the power set $2^{|E|}$, in a manner similar to [46, 60, 79]. The patterns that the algorithm can now accommodate belong to the set captured by the regular expression $(E \cup [EE^*E])(E \cup [EE^*E] \cup \{.\})^*(E \cup [EE^*E]) \cup E$. Note that the sets in the collection C need not form a partition of E. Example collections for the case of pattern discovery in biological sequences include (a) $C = \{\{A,G\}, \{C\}, \{D,E\}, \{F,Y\}, \{H\}, \{I,L,M,V\}, \{K,R\}, \{N,Q\}, \{P\}, \{S,T\}, \{W\}\}$; (b) $C = \{\{A,G,C,F,Y, H,I,L,M,V,N,Q,P,S,T,W\}, \{D,E\}, \{K,R,H\}\}$; (c) $C = \{\{A,G\}, \{A,G,P,S,T\}, \{C\}, \{D,E\}, \{D,E,Q,N\}, \{K,R,H\}, \{I,L,M,V\}, \{F,Y,W\}\}$; and others. The actual choice of the collection to use is dependent on the specifics of the problem that is being addressed.

Finally, SPLASH [18] uses MOTIF as its starting point and combines it with the "maximality" constraint to generate its results. Because of its MOTIF-like origin, this approach depends on the presence of *identically* conserved amino acids that will serve as "anchor points" around which larger patterns involving homologous substitutions can be built.

4. APPLICATIONS OF PATTERN DISCOVERY

We will next describe several applications of pattern discovery that our group and others have explored. Those of the applications which have wider applicability will be discussed in more detail.

4.1.1. *Single and Composite Descriptors*

Probably the first application of pattern discovery in computational biology was that of determining sequence features that have been conserved through evolution and their eventual association with other functional or structural features.

In that regard, the PROSITE database [8] represents the earliest concerted effort to compile a comprehensive collection of individual patterns that characterize small, curated collections of proteins and protein fragments. What one is after is the discovery of a combination of amino acids that is present in all of the members of the protein collection under consideration. This combination is given in the form of a regular expression that is a characteristic descriptor for the collection. The expression should be sensitive enough to identify new, previously unknown members of the collection but also specific enough so as to not generate erroneous "hits."

The various pattern discovery algorithms which have been proposed over the years have used PROSITE entries as benchmark tests. Of particular interest are those entries that correspond to protein families as well as functional or structural domains that exhibit extreme sequence divergence: in these cases, weight matrices are given instead of a single descriptor; these matrices allow the detection of members of the class. Finding a succinct, descriptive pattern to replace a weight matrix is thus always welcome. The regular expression

[KR]....[ILMV]....L...[AG]...T....[ILMV]L.....[AG]...

[AG]..[ST].[FY][ILMV].[AG]

is an example of such a pattern. It was generated by running TEIRESIAS on the sequences members of PROSITE entry PS00347 (Rel. 15.0). The latter contains 10 sequences (of which one is a short fragment) that are poly(ADP-ribose) polymerases (PARPs). PARP is a eukaryotic enzyme that catalyzes the covalent attachment of ADP-ribose units from NAD(+) to various nuclear acceptor proteins. PARPs contain an N-terminus domain that binds specifically single-stranded DNA in a zinc-dependent manner. PROSITE reports both a pattern and a matrix for PS00347: the pattern is present in only 8 of the 10 sequences, whereas the, matrix allows the detection of only 9 of the 10 sequences. The pattern we are reporting here is present in *all* 10 of the sequences in PS00347 and is specific enough to be used as a predicate for identifying additional members of this collection. It is worth pointing out that the motif we report here

comprises positions that are primarily chemically conserved; indeed, there are only three locations (L, T, L) where the respective amino acid is identically conserved.

As the databases grew in size, additional diverse and important families of proteins became known and single descriptors proved limiting for sufficiently characterizing these families. Composite-descriptor approaches came to alleviate some of the problems afflicting the single-descriptor methods and representative such examples abound [5, 6, 30, 31, 32, 33, 37, 43, 72, 81]. The driving observation behind composite descriptors is that many proteins are composed of a small number of conserved elements each of which could be represented by a descriptor. A collection could then be represented by some subset of the patterns corresponding to the identifiable conserved regions.

4.1.2. *Discovery of Tandem Repeats in DNA Sequences*

A tandem repeat is a collection of multiple instances of a combination of nucleotides with the instances appearing back to back (i.e., tandemly) in genomic DNA. Each instance is a slightly modified form of the same basic unit. The nature and extent of the basic unit as well as the number of copies that make up a tandem repeat can vary substantially both across repeats and across organisms.

The problem of tandem repeat discovery can be defined as the determination of all tandem repeats present in the input under consideration, the extent of the basic unit from which the instances are derived, the number of instances corresponding to the repeat, and their location in the processed input.

There is a substantial body of work on the study of tandem repeats due to their impact on chromosome pairing and the resulting highly polymorphic repeat clusters. References [51, 64] contain detailed discussions on the various biological aspects relating to tandem repeats.

Early algorithmic work was carried out in the presence of rather restrictive assumptions: for example, all instances of the tandem repeat were required to be *exact* copies of the basic unit, or *approximate* copies of the basic unit could be allowed but only repeats comprising *two* copies of the basic unit were sought [47, 49, 57].

More recently effective algorithms were proposed which relax these assumptions: they begin with the identification of tandem-repeat *seeds* or *suspicious* patterns which they subsequently treat as candidates to be either verified or rejected through examination of the respective neighborhoods in the processed input [12, 13, 38]. For a detailed review, from a computer science standpoint, of previously reported algorithms, the reader can refer to [39].

It is fair to say that among the previously proposed algorithms, those which produce exhaustive results can only be effective when the size of the processed genomic input is relatively small, e.g., several hundred thousand nucleotides. On the other hand, those algorithms that employ heuristics can handle larger data sets at the expense of missing some of the tandem repeats that are present in the input.

With this in mind, we designed an algorithm which permits the discovery of tandem repeats when the copies of the basic unit are approximate [82]. At the same time, our algorithm imposes no limitation on the extent of the basic unit or the number of copies and is efficient with large inputs. If one thinks of the problem of tandem repeat identification as a pattern discovery problem, the basic observation that drives this approach is that a region containing a tandem repeat will give rise to a large number of patterns and their position lists will contain offsets that are very close to one another.

On a given input, and after deciding the amount of variation that the basic unit can undergo,[7] we use the TEIRESIAS algorithm to determine *all* patterns that appear two or more times. The algorithm's guarantees of discovering all patterns that are maximal in composition and length and whose support is above a certain threshold are naturally suitable for this. We define a function on the position lists of the discovered patterns which when evaluated at a given location returns a nonnegative value. Using this function, we can zoom in to those of the locations where the evaluation function exceeds an experimentally determined threshold and further improve the localization while discarding false positives. We finally report the properties of the corresponding tandem repeat: extent of basic unit, number of copies, absolute positions in the processed input, and a score for the Clustal-w alignment [85] of all the instances.

4.1.3. *Multiple Sequence Alignment*

In this section, we concern ourselves with the task of aligning salient sequence features that are potentially present in a given collection of proteins and protein fragments. The sequences of the collection are presumed to be homologous (i.e., ancestrally related).[8] As we have already mentioned, the multiple sequence alignments of sequences found early use in the discovery of patterns in biological sequences.

We define the multiple sequence alignment problem as follows: given a collection of N sequences, insert spaces

[7] This can be effectively decided by an appropriate choice of the values for the parameters L and W.

[8] A variant of the problem attempts to align such inputs by focusing on the alignment of functional or structural features that are shared by the sequences in the collection of interest; obviously, this variant depends on domain-specific knowledge that may or may not be available in the general case.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

(gaps) into the sequences or at the beginning/end of them so that the resulting sequences all have the same length. The quality of the resulting alignment directly depends on the scoring function that is used to reward agreements, penalize mismatches and gaps, and bring out the *best commonality* of these sequences.

In addition to their use in pattern discovery and the determination of conserved sequence features in proteins and DNA, algorithms for multiple sequence alignment find important uses in the representation of protein families and superfamilies, the deduction of evolutionary history directly from biological sequences, gene cloning, and shotgun sequence assembly [39]. Not surprisingly, there has been considerable research work that both examined the problem *per se* and studied the impact of scoring functions on the obtained results; references [3, 19, 39, 44, 56, 88] address many of these points and the interested reader should refer there for more details.

Multiple sequence alignment can be formulated in the context of dynamic programming but the resulting space requirements make such an approach applicable to sets containing only very few, relatively short sequences. Consequently, a large body of research work explored alternative schemes that try to generate an optimal alignment of N sequences through an iterative approach based on comparisons between pairs of sequences [25].

In the context of our work on the applications of pattern discovery, we designed MUSCA, a two-phase algorithm for computing the multiple sequence alignment for a set of N sequences. During the first phase, we discover patterns that are common among a subset of the N sequences. We use these patterns during the second phase to produce the multiple sequence alignment. The conceptual underpinnings of this method can be traced to earlier work [75] where the multiple alignment is driven by pairwise comparisons of the processed input sequences. What distinguishes our work is that it naturally disengages itself from the dependence on the order in which the input sequences are considered and that its starting point is a K-wise alignment (with $K \geqslant 2$).

To begin, MUSCA uses TEIRESIAS to generate the patterns that are present in the input sequences to be aligned. The patterns are then mapped to vertices of a directed graph. If two patterns p_i and p_j do not occur simultaneously in any sequence, then there is no edge connecting the corresponding vertices of the graph. An edge will connect the vertices corresponding to p_i and p_j with direction from p_i to p_j if p_i occurs before p_j in all the sequences where they both appear. Edges are labeled depending on whether and p_i and p_j (a) are pairwise *in*compatible, (b) have overlapping instances, or (c) are pairwise compatible but do not overlap. All vertices that are joined by incompatible edges as well as those participating in inconsistent cycles comprise the *basic infeasible* sets. After labeling the vertices of the reduced graph with the help of a simple cost function, we use a greedy algorithm to obtain a solution to a weighted set-cover problem that essentially identifies the minimum number of patterns/vertices that must be removed. The resulting graph is used to determine blocks that involve overlapping feasible patterns. We obtain the final alignment by properly aligning the blocks and padding up the existing gaps.The resulting algorithm works very efficiently with large inputs that contain long sequences and a detailed description of it together with results on several benchmark datasets can be found in [61].

In Fig. 2, we are showing a small example-alignment generated by MUSCA. The input set was seven proteins involved in transcription regulation that contain the MADS domain; the proteins are from *Arabipdopsis thaliana*, *Brassica napus*, *Dianthus caryophyllus*, *Petunia hybrida*, *Antirrhinum majus*, and *Nicotiana tabacum* and were subselected from the members of the prosite entry PS00350 [8].

4.1.4. *Bio-Dictionary/Homology Searching/Functional and Structural Annotation*

The main characteristic of the early applications of pattern discovery in the analysis of biological sequences was that the processed datasets were curated collections of sequences. The implicit assumptions made when compiling such collections were that (a) the sequences/members of the collection are indeed related, and (b) they form a *single* set (as opposed to being the union of several smaller sets).

Such approaches make the discovery of conserved functional and structural signals that cross family boundaries difficult. Methods for sequence homology helped in that respect but as the databases grew in size the statistical thresholds were raised accordingly, effectively burying otherwise valid telltale homologies.

Ideally, one would like to carry out unsupervised discovery by treating the largest possible database as *an indivisible entity*; doing so should in principle permit the discovery of many more signals that are still conserved at the sequence level. If we treat proteins as the biological analog of sentences in natural languages, then any recurrent functional and structural signals whose traces remain at the level of the amino acid sequence should be observable as patterns-words that are being reused. Figure 3 should help appreciate the beneficial impact of a big database on the completeness of the results that one obtains through unsupervised pattern discovery. In this simple example, we have removed the "spaces" between consecutive words and the assumption is that the reader/database-miner does not have knowledge of the English language. "Coherent" combinations of English letters will not be observed until there

FIG. 2. An example multiple-sequence alignment produced by MUSCA. The shown proteins are involved in transcription regulation. Those amino acids that participated in the patterns that were used to induce the alignment are shown capitalized. Dashes represent inserted gaps. See also text.

is enough knowledge base to support them. The small database shown in the figure begins with two streams and reaches its final size through four more instalments. The number of coherent letter combinations that can be discovered increases after each instalment (indicated by rectangles in the figure). However, as can be seen, there are still parts of the database that cannot be accounted for.

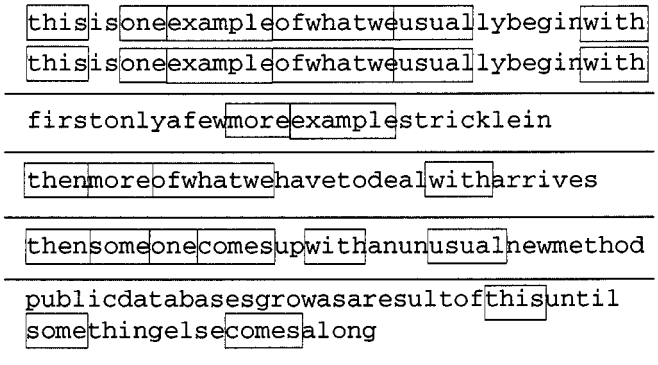


FIG. 3. The impact of having access to a large input database. See text for an explanation.

Although highly desirable, it was not until recently that unsupervised pattern discovery on a large public database was achieved [67]: using TEIRESIAS, we were able to process the February 10, 1999 release of the GenPept database[9] as a whole and generate a collection of almost 27 million patterns that covered 98.12% of all amino acid positions of the processed input. The database contained a little over 387,000 proteins totalling approximately 120 million amino acids. Due to its obvious analogy to natural languages, we have been referring to the compiled collection of patterns as the "Bio-Dictionary." The pattern entries of the Bio-Dictionary are called "seqlets."

The Bio-Dictionary essentially contains compact descriptors for almost all of the sequence space of natural proteins, to the extent that this space is uniformly sampled by the current contents of GenPept. Additionally, by associating each of the seqlets with probability estimates (through a second order Markov model) and currently available functional ("Feature Table, FT" and "Description, DE" entries)

[9] The impact on performance would of course be different from algorithm to algorithm.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

information from Swiss–Prot [7] and structural annotation from the Protein Data Bank (PDB) [1], we obtain a complete characterization of each discovered pattern. In Fig. 4, we are showing an augmented such entry. However, it should also be noted that there are Bio-Dictionary seqlets with no instances in the PDB or which capture previously unobserved signals for which there is no functional annotation in Swiss–Prot: consequently these entries cannot be fully characterized until more information becomes available.

The availability of the Bio-Dictionary has opened up new opportunities in the study and analysis of proteins. First, recall that, by definition, each seqlet of the Bio-Dictionary is representative of a protein fragment which appears with modifications across more than one input sequences. As such, the seqlets denote and capture local homologies that are shared by the respective sequences. When presented with a query sequence for which we wish to determine the existence of any similarities between it and sequences in GenPept, we simply need to identify which of the seqlets are present in the query: the sought similarities will correspond to the region of the query that a seqlet covers and the respective regions in the sequences that contain it. Frequently, more than one seqlet will corroborate the same local alignment and this needs to be taken into account. As with traditional homology-searching algorithms, scoring schemes can be imposed and used to rank the results. This approach replaces an expensive search of a query against the entire database by a much faster search against a dictionary of patterns without compromising sensitivity; as the contents of the Bio-Dictionary begin to saturate, the computational savings of such searches are bound to become substantial. A detailed discussion of this approach can be found in [26].

Another straightforward use is in the definition of protein families and domains using entries of the Bio-Dictionary. Some of the patterns are specific enough to be used as family predicates and for carrying out functional annotation of proteins and protein fragments. Others correspond to recurrent functional/structural elements.

In an analogous manner, those seqlets that have instances in sequences of the PDB can be associated with structural fragments: these structural fragments are the three-dimensional realizations of the sequence substrings that the pattern captures. If there is more than one instance in the PDB, an alignment of the fragments and a computation of the respective RMS error provide an estimate of how well the pattern can characterizes a local three-dimensional structure. As in the case of homology searching, multiple patterns can corroborate the same local three-dimensional structure.

The association of sequence patterns with three-dimensional structure through supervised learning methods and the subsequent exploitation of such associations have been pursued for a number of years with increasing degrees of success [17, 40, 45, 74]. What differentiates the Bio-Dictionary approach is that treating the input database as a whole allows for the exhaustive discovery of structural signals which cross family boundaries and could have otherwise remained unobserved.

Given a query, we can identify the seqlets that are present in it and subselect only those with instances in the PDB: for those regions of the query (possibly the *entire* query) that are covered by such patterns we can in principle attempt to predict the three-dimensional structure by piecing together the structural fragments that correspond to successive, overlapping patterns. However, of course, the problem is not that simple and many more things need to be taken into account; we are currently in the process of studying this problem in more detail.

Additional applications of the Bio-Dictionary together with a detailed description of how it was generated can be found in [67]. Details on Bio-Dictionaries that have been compiled from complete genomes can be found in [68].

4.2. *Association Discovery*

The discussion so far involved situations where the length of the various streams in the database under consideration was variable and unbounded. Let us now consider a database comprising streams each of which is of fixed length. Without loss of generality, we can also assume that the event set is a *mixture* of both categorical values and numerical events. With this in mind we can alternatively view each of the streams as a record composed of a fixed number of fields, each of the fields representing the answer to a multivalued question. The following is an example of a five-stream database with each stream comprising six distinct events:

S_1 =	rent	R&B	romance	BSc	\$30K	4 dr. sedan
S_2 =	rent	rock	fiction	MSc	\$50K	2 dr. hatchback
S_3 =	own	jazz	science-fiction	PhD	\$70K	sports util. veh.
S_4 =	own	jazz	romance	PhD	\$70K	4 dr. hatchback
S_5 =	rent	R&B	fiction	MSc	\$30K	2 dr. sedan

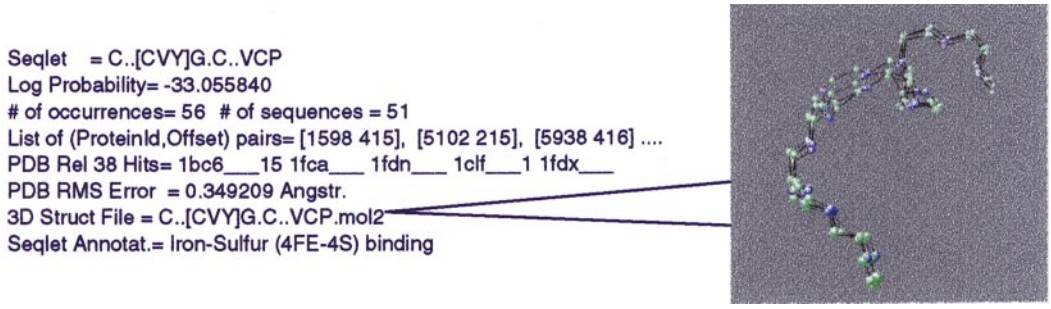
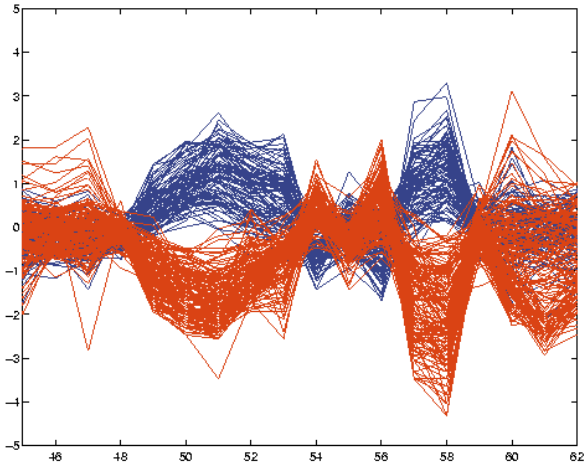


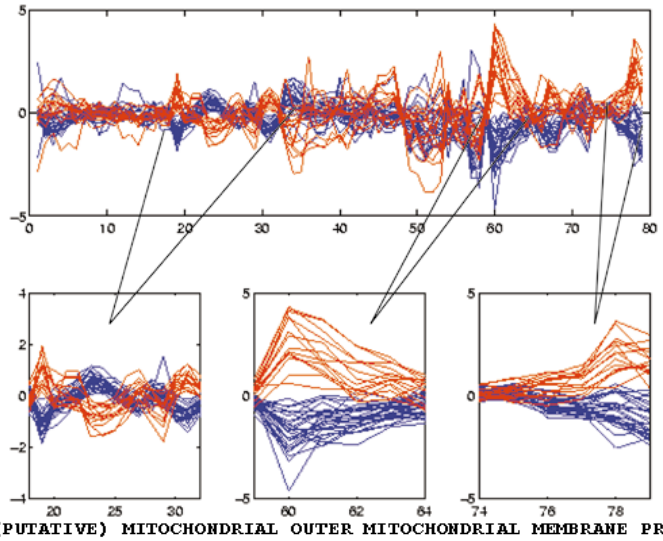
FIG. 4. An example augmented entry from the Bio-Dictionary. It consists of the actual seqlet, its estimated log probability, information about all instances of the seqlet in the processed release of GenPept in the form of (protein id, offsets) pairs, the identities of those entries in the PDB that contain an instance of the seqlet and the RMS error obtained from aligning the respective structural fragments, a functional "meaning" for the seqlet, and potentially other information.



r-YIL136W	b-YOL069W	b-YOR176W	b-YOL044W	b-YGR264C	r-YHR193C	b-YKL004W	b-YDR439W	r-YPR102C	r-YDR418W
b-YDR173C	b-YDR356W	b-YOR278W	r-YDR265W	r-YGL135W	r-YMR064W	r-YPL057C	r-YIL134W	r-YEL054C	r-YHL001W
r-YGR008C	b-YLR085C	r-YBR248C	r-YOR363C	r-YBL092W	b-YJR064W	r-YHR190W	r-YBR132C	r-YKL006W	r-YNL069C
b-YLR393W	r-YLR172C	r-YCL030C	b-YDR538W	r-YOL127W	r-YAL003W	b-YLR450W	b-YJL219W	r-YIL133C	r-YNL301C
b-YPR122W	r-YAL015C	b-YER068W	r-YDR123C	r-YGL030W	b-YKL048C	r-YCR048W	b-YIL171W	r-YOL120C	r-YBL027W
b-YFL053W	b-YGL163C	b-YPR141C	r-YKL198C	r-YLR448W	r-YGR061C	b-YOR237W	b-YJL214W	r-YMR242C	r-YPL079W
b-YLR102C	b-YML032C	b-YGL192W	r-YJR059W	r-YPL198W	r-YJL130C	r-YIL101C	b-YIL170W	r-YLR344W	r-YER117W
r-YAL040C	b-YEL037C	b-YNL012W	b-YOR261C	r-YGL147C	r-YMR271C	r-YBR067C	b-YOL156W	r-YHR010W	r-YGR148C
r-YMR199W	b-YLR288C	b-YOL064C	b-YDL147W	r-YGR214W	b-YBR037C	b-YDR148C	r-YER145C	r-YDL075W	r-YGR034W
b-YDL017W	b-YHR164C	r-YDR502C	b-YGR270W	r-YOL040C	r-YBR167C	b-YGR099W	b-YMR056C	r-YPL143W	r-YDR471W
b-YJL013C	r-YKL017C	r-YCR050C	b-YDR059C	r-YGL123W	r-YFL002C	b-YML043C	r-YBR222C	r-YPR043W	r-YOR234C
b-YGR049W	b-YLR274W	r-YNL131W	b-YDR054C	r-YHR203C	r-YOR001W	b-YOR290C	r-YOR348C	r-YDL191W	r-YIL052C
r-YIL035C	r-YGR180C	b-YLR115W	b-YDL064W	r-YOR096W	r-YOR028C	r-YGR063C	r-YMR301C	r-YLR185W	r-YDL136W
b-YER176W	b-YKR010C	b-YIL030C	r-YBL067C	r-YNL096C	r-YGL019W	r-YGL043W	r-YDR135C	r-YIL148W	r-YHR141C
r-YGR282C	b-YPL152W	r-YGR178C	r-YBR058C	r-YBL072C	b-YKL196C	b-YFL036W	b-YIL075C	r-YNL162W	r-YDL083C
r-YLR286C	b-YOR034C	b-YGR013W	r-YDR155C	r-YER102W	r-YMR183C	r-YDR397C	b-YML091C	r-YOR293W	r-YNL302C
b-YBR195C	r-YJR019C	b-YPR178W	r-YML074C	r-YPL081W	b-YKL122C	r-YPL037C	r-YCL017C	r-YCR031C	r-YJL136C
b-YOL012C	r-YGL035C	r-YBL074C	b-YCL043C	r-YEL050C	r-YFL048C	b-YDL150W	r-YHR163W	r-YOL121C	r-YPR132W
b-YLL022C	r-YLR142W	b-YPL253C	r-YDR483W	r-YHR147C	b-YDL226C	r-YPR187W	r-YER090W	r-YKR057W	r-YLR333C
r-YDR174W	r-YJL101C	r-YMR308C	b-YKL201C	r-YJR113C	b-YKR101W	b-YER148W	b-YJL024C	r-YGR118W	r-YER131W
b-YKL049C	r-YBR145W	b-YER107C	r-YML115C	r-YDR385W	b-YOL068C	b-YPL122C	r-YBL017C	r-YGR027C	r-YHR021C
r-YBL097W	r-YOR344C	b-YOL067C	b-YPL069C	r-YOR133W	b-YJL076W	b-YDR362C	b-YPL120W	r-YGL189C	r-YDL061C
b-YDR515W	b-YOL056W	b-YBL056W	r-YGR147C	r-YLR069C	b-YKR019C	b-YPR104C	b-YOL109W	r-YKL156W	r-YBR084C-A
b-YFL037W	r-YGR087C	r-YOR079C	r-YNL247W	b-YJL102W	r-YLR260W	r-YMR039C	r-YKR082W	r-YLR388W	
b-YNL126W	r-YAL038W	r-YOR095C	r-YOR168W	r-YGR094W	r-YBR036C	b-YHR119W	b-YER105C	r-YGR085C	

FIG. 7. An example of a relationship among 248 genes that can be discovered by applying the TEIRESIAS pattern discovery algorithm on the signs of the derivatives of the raw expression ratios of [24] from *S. cerevisiae*, after they have been quantized. The identities of these 248 genes are also shown. Each gene's identifier is prefixed by either "b" or "r" depending on whether the gene belongs in the "blue" or the "red" group. See also text.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151



r-YIL136W	OM45	(PUTATIVE) MITOCHONDRIAL OUTER	MITOCHONDRIAL MEMBRANE PROTEIN
r-YML004C	GLO1	AMINO ACID METABOLISM	GLYOXALASE I
r-YGR008C	STF2	ATP SYNTHESIS	ATPASE
b-YGR152C	RSR1	BUD SITE SELECTION	GTP-BINDING PROTEIN, RAS SUPERFAMILY
r-YLR178C	TFS1	CELL CYCLE	SUPPRESSES CDC25 MUTATIONS
b-YDR184C	ATC1	CELL POLARITY	MEMBER OF BUD6P COMPLEX
b-YKL114C	APN1	DNA REPAIR	APURINIC/APYRIMIDINIC ENDONUCLEASE
r-YBL075C	SSA3	ER AND MITOCHONDRIAL TRA	CYTOSOLIC HSP70
r-YFR053C	HXK1	GLYCOLYSIS	HEXOKINASE I
r-YLL026W	HSP104	HEAT SHOCK RESPONSE /THE	HEAT SHOCK PROTEIN
b-YCL027W	FUS1	MATING; CELL FUSION	SH3 DOMAIN PROTEIN
b-YGR155W	CYS4	METHIONINE BIOSYNTHESIS	CYSTATHIONINE BETA-SYNTHAS
b-YJL208C	NUC1	MITOCHONDRIAL METABOLISM	ENDONUCLEASE
b-YNL112W	DBP2	MRNA DECAY	RNA HELICASE
b-YGL120C	PRP43	MRNA SPLICING	SPLICEOSOME DISASSEMBLY FACTOR; RNA HELICASE
b-YGR159C	NSR1	NUCLEAR PROTEIN TARGETIN	NLS-BINDING PROTEIN
r-YGR088W	CTT1	OXIDATIVE STRESS RESPONS	CATALASE T
r-YBR286W	APE3	PROTEIN DEGRADATION	VACUOLAR AMINOPEPTIDASE Y
r-YOR027W	STI1	PROTEIN FOLDING	COMPONENT OF HSP70-HSP90 COMPLEXES
r-YLR259C	HSP60	PROTEIN FOLDING	MITOCHONDRIAL CHAPERONIN
b-YER001W	MNN1	PROTEIN GLYCOSYLATION	ALPHA-1,3-MANNOSYLTRANSFERASE
b-YDR341C	NONE	PROTEIN SYNTHESIS	ARGININE-TRNA SYNTHETASE
b-YGR264C	MES1	PROTEIN SYNTHESIS	METHIONYL TRNA SYNTHETASE
b-YOR293W	RPS10A	PROTEIN SYNTHESIS	RIBOSOMAL PROTEIN S10A
b-YDR399W	HPT1	PURINE BIOSYNTHESIS	HYPOXANTHINE GUANINE PHOSPHORIBOSYL TRANSFERASE
b-YBL039C	URA7	PYRIMIDINE BIOSYNTHESIS	CTP SYNTHASE 1
r-YKL035W	UGP1	PYRIMIDINE METABOLISM	UGP1, UDP-GLUCOSE PYROPHOSPHORYLASE
b-YGL171W	ROK1	RRNA PROCESSING	RNA HELICASE
b-YJR002W	MPP10	RRNA PROCESSING	U3 SNORNP PROTEIN
b-YPL211W	NIP7	RRNA PROCESSING	UNKNOWN
b-YPL266W	DIM1	RRNA PROCESSING, 18S	DIMETHYLADENOSINE TRANSFERASE
r-YPR149W	NCE102	SECRETION, NON-CLASSICAL	UNKNOWN
b-YMR127C	SAS2	SILENCING	ZINC-FINGER PROTEIN
b-YCL054W	NONE	SILENCING (PUTATIVE)	UNKNOWN
b-YOR340C	RPA43	TRANSCRIPTION	RNA POLYMERASE I 36 KD SUBUNIT
b-YNL248C	RPA49	TRANSCRIPTION	RNA POLYMERASE I 46 KD SUBUNIT
b-YJL148W	RPA34	TRANSCRIPTION	RNA POLYMERASE I SUBUNIT
b-YDL150W	RPC53	TRANSCRIPTION	RNA POLYMERASE III 47 KD SUBUNIT
b-YPR187W	RPO26	TRANSCRIPTION	SHARED SUBUNIT OF RNA POLYMERASES I, II, AND III
r-YGR194C	NONE	XYLULOSE UTILIZATION	XYLULOKINASE

FIG. 8. An example of a relationship among 40 genes that can be discovered by applying the TEIRESIAS pattern discovery algorithm on the signs of the derivatives of the raw expression ratios of [24] from *S. cerevisiae*, after they have been quantized. The identities of these 40 genes are also shown. Notice that the expression profile agreement extends over three *distinct* time intervals. See also text.

There are numerous contexts where the database to be processed is such as the one shown here and any patterns that can be discovered in it have actual commercial value. Examples of patterns that can be found in this mini-database include "rent R&B..$30K" and "own jazz · PhD $70K." Considerations such as maximality of patterns, minimum allowed density, and minimum support carry in a natural manner.

It should be noted that pretty much all of the pattern discovery algorithms that we have discussed can in principle be modified (with varying degrees of difficulty) to handle inputs of this type.[10] In fact, whether the database is presented as above, or as

S_1 =	1	3	6	9	12	15
S_2 =	1	4	7	10	13	16
S_3 =	2	5	8	11	14	17
S_4 =	2	5	6	11	14	18
S_5 =	1	3	7	10	14	19

or as

$S_1 = e_1\ e_3\ e_6\ e_9\ e_{12}\ e_{15}$
$S_2 = e_1\ e_4\ e_7\ e_{10}\ e_{13}\ e_{16}$
$S_3 = e_2\ e_5\ e_8\ e_{11}\ e_{14}\ e_{17}$
$S_4 = e_2\ e_5\ e_6\ e_{11}\ e_{14}\ e_{18}$
$S_5 = e_1\ e_3\ e_7\ e_{10}\ e_{14}\ e_{19}$

with $E = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}, e_{11}, e_{12}, e_{13}, e_{14}, e_{15}, e_{16}, e_{17}, e_{18}, e_{19}\}$, the amount of information remains the same. Patterns discovered on an input of this type (fixed length) are typically thought of as "associations," and the problem of discovering these patterns is referred to as "association discovery."

We can define the problem of association discovery as follows: given a database of N records each of which comprises two or more fields, determine the set of all possible associations that involve at least two fields and which are supported by at least K records, with $2 \leqslant K \leqslant N$.

The problem of finding *maximal* associations is computationally very demanding. Using brute force, one can tackle the problem in one of two ways:

(1) Form all subsets of two or more records and examine all their fields in order to identify the maximal intersection (equivalently: maximal association). If the dataset contains N records, then all 2^N possible groupings (i.e., the *power set* of the input) need to be considered, only a subset of which leads to intersections involving K or more records: the approach quickly becomes very prohibitive as the number N of records increases.

(2) Enumerate each of the $(|E|+1) * (|E|+1) * \cdots * (|E|+1)$ combinations of values (i.e., traverse all points of the underlying lattice) and for each of them identify that subset of records that contains the combination under consideration.[11] As before, the approach quickly becomes prohibitive as the number of fields, F, increases.

Depending on the specifics of the database under consideration, choice (1) may be more efficient than choice (2), or vice versa. In both cases, the computation quickly becomes prohibitive as the number of records and the number of fields increase. Since we are interested in associations that are supported by a minimum of K records, unpromising paths can be pruned early leading to substantial performance gains but the problem still remains computationally demanding.

Finally, it should be noted that the input may be quantized and classes of equivalent symbols that correspond to neighboring quantization bins can be established before carrying out association discovery.

4.2.1. *Gene Expression Analysis*

In the past several years, considerable research effort has been invested in the analysis of gene expression data. In such studies one seeks to first establish and then exploit significant relationships among individual genes. Typical applications include the study of preferential gene expression in specific tissues, the study of transcription differences that are responsible for the transition from normal to abnormal cell behavior, the study of gene transcription changes as the result of a cell's natural development or its response to environmental changes and signals, etc.

[10] The impact on performance would of course be different from algorithm to algorithm.

[11] Notice that here we have assumed the availability of the "don't care" character.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

Technological advances during the past decade led to the advent of DNA array technology and its natural application to the study of gene expression. One can identify two basic types of DNA arrays: the first type is known as the "DNA chip" and comprises a large number of oligonucleotides each of which is synthesized *in situ* on solid support using known sequence information [29]; the second type is known as the "microarray" and consists of a collection of DNA targets, usually PCR products from cDNA or genomic clones arrayed on solid support [73]. Independently of the type of the array the remaining steps are essentially the same: mRNA that has been purified from experimental samples from the cell line or tissue to be studied is labeled with a fluorescent dye (e.g., "red") whereas a reference sample is labeled with a different fluorescent dye (e.g., "green"). Both labeled samples are then allowed to hybridize (i.e., base-pair) with the targets of the used DNA array. After hybridization has completed, the DNA array is washed to remove any unbound mRNA probe and scanned with the help of a laser that is used to excite the molecules of the fluorescent dye. The emitted light is captured by an appropriate detector such as a confocal microscope and the light's intensity is recorded for each of the two dyes: those locations of the DNA array that have more bound probe will fluoresce more strongly. For each target of the DNA array, and for the corresponding probe, the ratio of concentrations between the studied cell line or tissue and the reference sample can be computed as the ratio of the fluorescence intensities for each of the two dyes. These intensity ratios are typically log-transformed so that inductions and repressions of identical magnitude will give rise to values with opposite signs, whereas probes whose expression levels have remained unchanged will generate log ratios with value 0.

Clearly, the comparative studies that can be afforded by this new and exciting technology are limited only by the experimenters' imagination. Nonetheless, all of the possible applications can essentially be assigned to one of two categories depending on the nature of the data being generated.

The first category includes experiments which attempt to capture and characterize the microscopic state of a cell (or tissue, plant, etc.) and associate it with an observable, static, macroscopic state. For example, given a collection of M corn plants for which we also know the values of observable properties such as rate of drydown, stalk lodging, root lodging, percentage of dropped ears, plant height and ear height, we determine the level of expression for each one of N genes of interest. The (i, j)th entry of the resulting M by N matrix is the level of expression of the jth gene in the ith plant. What is sought is one or more associations of *some subset of the N genes* (and their respective expression levels) with *some subset of the observable plant properties*. Such associations essentially hypothesize a causal relationship between genotype and phenotype that can be corroborated or refuted through further study.

In the second category, we have dynamic studies that attempt to track the induction or repression of various genes as a function of time; such changes can be part of a cell's natural development or the response to environmental changes or signals. A representative example is the aggregation of data from experiments on the budding yeast *S. cerevisiae* that are reported and discussed in [24]: each one of the M genes in *S. cerevisiae* is tracked for a total of N time steps and its expression level recorded; among others, these N time steps include observations during the course of natural cell stages such as the mitotic cell division cycle, as well as responses to high- and low-temperature shocks. The (i, j)th entry of the resulting M by N matrix is the (log-transformed) expression of the ith gene as measured at the jth time step. What is sought in this case is the identification of genes that have similar (resp. opposite) expression profiles. Such profiles can be indicative of the status of cellular processes. Also, profiles that are shared by genes of known function and uncharacterized or novel genes may be helpful in elucidating the function of the latter.

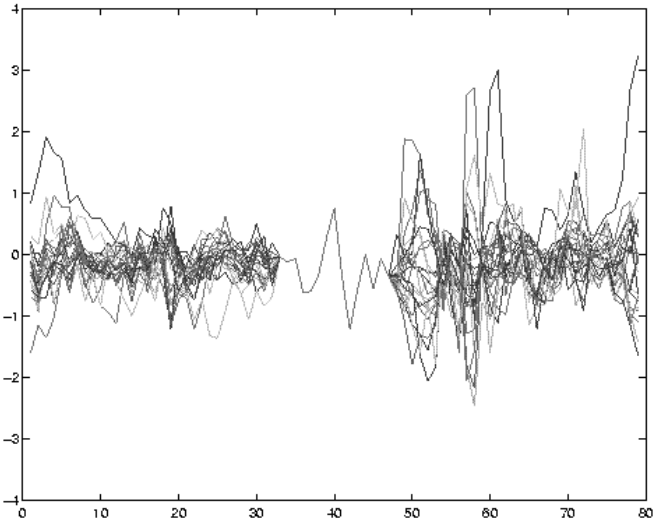
In the general case, one can think of the input as being an M by N matrix of real numbers with the task at hand being the identification of relationships involving *subsets* of the rows and *subsets* of the columns of this matrix. Establishing such relationships is important because they can be turned into hypotheses about information flow from/to the corresponding genes or can be combined to generate putative network models involving these as well as additional genes [23].

Most of the analyses that have appeared in the literature and which attempt to generate such hypotheses treat the columns of the data matrix as points in a high-dimensional space and apply traditional clustering/partitioning techniques [11, 23, 24, 52, 55, 84, 90]. The results obtained so far using clustering methodologies have been very encouraging. However, it should also be pointed out that in the general case and for a given collection of studied genes, clustering methodologies can capture only a *subset* of the relationships that potentially exist among the studied genes. Indeed, the typical clustering algorithm allows a given gene to participate in a single cluster. However, this is rather limiting since a given gene may simultaneously belong to multiple clusters for different reasons. Additionally, clustering schemes seek to form clusters whose members are *globally similar*: for example, by applying clustering to gene expression data from dynamic studies we will form clusters of genes whose expression profiles agree (within threshold) across all (or most of the) N time steps. Any agreements in the expression profiles of a subset of the M genes that span

only a fraction of the N time steps are thus likely to remain unnoticed.

It should be clear that the computational questions from both of the above categories can be recast in the context of association discovery with the M by N expression level/ratio matrix being essentially similar to the matrices we examined in the previous section. This formulation of the problem as well as the above considerations regarding the use of clustering approaches makes gene expression analysis a natural application for pattern discovery. Pattern discovery essentially subsumes clustering-based methodologies since it can permit a given gene to participate in more than one cluster: for an input matrix involving M rows, a pattern discovery-based approach will form and report *all* the subsets from the power set 2^M whose support exceeds the predetermined threshold.

In what follows, we will showcase the applicability of pattern discovery using the budding yeast *S. cerevisiae* data from [24]. This input represents an aggregation of expression data at 79 time points from time courses during the following processes: the cell division cycle after synchronization by alpha factor arrest (18 time points), centrifugal elutriation (14 time points) and with a temperature-sensitive cdc15 mutant (15 time points), sporulation (11



YNL330C RPD3	CHROMATIN STRUCTURE	HISTONE DEACETYLASE
YOL012C HTA3	CHROMATIN STRUCTURE	HISTONE-RELATED
YNL312W RFA2	DNA REPAIR	REPLICATION FACTOR A 36 KD SUBUNIT
YNL290W RFC3	DNA REPLICATION	REPLICATION FACTOR C 40 KD SUBUNIT
YOL018C TLG2	ENDOCYTOSIS	TRANS-GOLGI NETWORK T-SNARE
YNR017W MAS6	MITOCHONDRIAL PROTEIN TA	INNER MEMBRANE TRANSLOCASE COMPONENT
YNL286W CUS2	MRNA SPLICING, PUTATIVE	UNKNOWN
YNL316C PHA2	PHENYLALANINE BIOSYNTHES	PREPHENATE DEHYDRATASE
YNL306W NONE	PROTEIN SYNTHESIS	RIBOSOMAL PROTEIN, MITOCHONDRIAL SMALL SUBUNIT
YNR045W PET494	PROTEIN SYNTHESIS	COX3 TRANSLATIONAL ACTIVATOR
YNL284C MRPL10	PROTEIN SYNTHESIS	RIBOSOMAL PROTEIN, MITOCHONDRIAL L10
YNR015W SMM1	PROTEIN SYNTHESIS, MITOC	UNKNOWN
YNL332W THI12	PYRIMIDINE BIOSYNTHESIS	UNKNOWN
YNL282W POP3	RRNA AND TRNA PROCESSING	RNASE P AND RNASE MRP SUBUNIT
YNR049C MSO1	SECRETION	UNKNOWN; INTERACTS WITH SEC1P
YNR019W ARE2	STEROL METABOLISM	ACYL-COA STEROL ACYLTRANSFERASE
YNR043W MVD1	STEROL METABOLISM	MEVALONATE PYROPHOSPHATE DECARBOXYLASE
YNR001C CIT1	TCA CYCLE	CITRATE SYNTHASE
YNR023W SNF12	TRANSCRIPTION	COMPONENT OF SWI/SNF GLOBAL ACTIVATOR COMPLEX
YNL268W LYP1	TRANSPORT	LYSINE PERMEASE
YOL020W TAT2	TRANSPORT	TRYPTOPHAN PERMEASE
YNL292W PUS4	TRNA PROCESSING	PSEUDOURIDINE SYNTHASE
YNR041C COQ2	UBIQUINONE BIOSYNTHESIS	PARA-HYDROXYBENZOATE POLYPRENYLTRANSFERASE

FIG. 5. An example of a relationship among 23 genes that can be discovered by applying the TEIRESIAS pattern discovery algorithm on the raw expression ratios of [24] from *S. cerevisiae*. The identities and functional information of these 23 genes are also shown. See text for more details.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

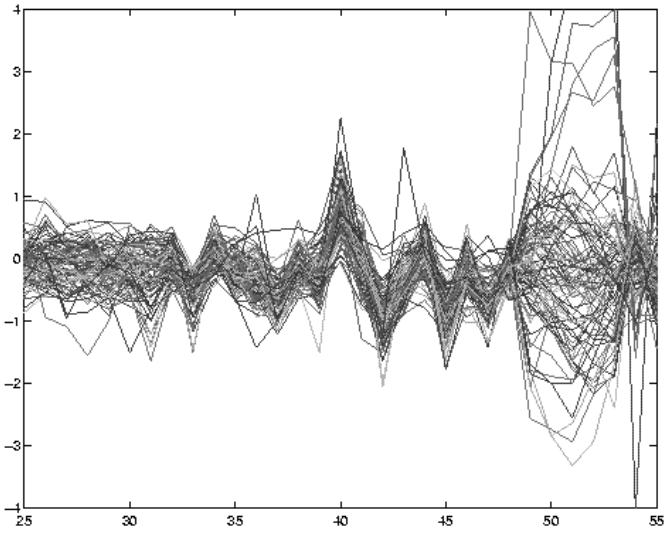
time points in total), heat shock (6 time points), shock by reducing agents (4 time points), cold shock (4 time points), and the diauxic shift (7 time points). A characteristic of this dataset is that it only contains the 2467 genes from *S. cerevisiae* which are currently functionally annotated. The actual input table of ratio values was obtained from the website http://rana.stanford.edu/clustering.

We first applied TEIRESIAS on the raw input, i.e., the matrix of ratios. The algorithm generated several million patterns which involved 10 or more genes. Essentially all the gene groups that are reported in [24] were discovered as well as many more additional ones involving a subset of the 79 time steps and genes of different functions. One such example can be seen in Fig. 5. It corresponds to 23 genes whose expression profiles coincide during time steps 33 through 47 inclusive but agree nowhere else; as can also be seen from the figure, these genes span several functional categories.

Instead of applying pattern discovery directly on the raw expression ratios, one can instead relax the requirement that the ratios agree in value and instead use as an alternative representation for the input the *sign of the input's derivatives*. For each of the 2467 genes of the input the expression ratio values were quantized using three quantization steps for every unit of expression ratio; the quantized values, which essentially smoothed the input array, were then replaced by the signs of their derivatives. We essentially rewrote the original input using a three-symbol alphabet: + if the expression level increased with respect to the previous time point, − if it decreased, and 0 if it remained unchanged. We applied TEIRESIAS on this rewritten input and generated numerous patterns that involved subsets of the 2467 genes. As can be seen from Fig. 6, this new representation scheme allows us to discover a relationship involving four times as many genes as before and whose profiles coincide during time steps 33 through 47 inclusive; notice that the expression profile of these 92 genes follows the shape of the expression profile of the 23 genes from Fig. 5. As before, the 92 genes span several distinct functional categories.

However, there is much more that can be achieved by applying pattern discovery on such inputs. In particular, discovering groups of genes that are *inversely* regulated is now a straightforward task as is evidenced by Fig. 7. In this figure, we are showing the expression profiles of 248 genes which again cross functional boundaries; the genes comprise two groups of 94 (= blue) and 154 (= red) genes, respectively, with the two groups being inversely regulated during time steps 48 through 59 inclusive.

Finally, in Fig. 8, we are showing the expression ratio profiles for 40 genes that again cross functional boundaries. The 40 genes belong to two groups comprising 26 (= blue) and 14 (= red) genes, respectively, with the two groups being inversely regulated in the time intervals 19 through 32 inclusive, 59 through 64 inclusive, and 74 through 79 inclusive. The expression profiles of these genes disagree at all remaining time points.



YER069W	YJL194W	YGR007W	YGL137W
YDL181W	YOR074C	YCL004W	YML049C
YNR058W	YMR234W	YDL147W	YDL153C
YHR208W	YER073W	YPL003W	YKR101W
YER114C	YER054C	YDL143W	YMR219W
YKL092C	YGL035C	YDL141W	YBR036C
YMR100W	YLR134W	YOL110W	YPR001W
YBR131W	YDL205C	YOR017W	YNL009W
YNL229C	YER068W	YEL054C	YDL165W
YDR118W	YJR094C	YDL083C	YKL139W
YGL134W	YHR042W	YJL136C	YBR193C
YDL179W	YBR272C	YJR123W	YMR227C
YNL289W	YCR028C	YPL090C	YDR311W
YDL155W	YDL217C	YDR337W	YHR050W
YAR019C	YFR028C	YJR064W	YNL318C
YGL229C	YDR301W	YDR441C	YKL217W
YJL099W	YJR093C	YFL058W	YFR029W
YPR052C	YMR240C	YBR037C	YDR160W
YHR107C	YMR268C	YHR089C	YHL016C
YJR065C	YDR473C	YCL001W	YMR261C
YDR369C	YHR206W	YEL022W	YOR125C
YER162C	YDR481C	YDL195W	YPL065W
YBR278W	YDR123C	YOL062C	YDR495C

FIG. 6. An example of a relationship among 92 genes that can be discovered by applying the TEIRESIAS pattern discovery algorithm on the signs of the derivatives of the raw expression ratios of [24] from *S. cerevisiae*, after they have been quantized. The identities of these 92 genes are also shown. See also text.

These examples demonstrate the kind of results that can be obtained by applying pattern discovery methodologies on gene expression data. Although we have showcased this on dynamic data (expression ratios at distinct time steps), the approach is equally well applicable to what we referred to above as static inputs.

4.3. *Other Extensions*

As evidenced by research papers in recent computational biology conferences [14, 20], there is increasing interest in the development of machine learning methods for extracting useful information from textual databases that contain information about biological sequences.

The issues and consideration are the same as in the case of the other databases that we have already discussed in this presentation, thus making this problem naturally suitable for treatment by pattern discovery methods. The completeness of the reported results and the guarantee of maximality in composition and length are essential ingredients here as well.

To showcase the power that a pattern discovery approach can bring to solving this problem, we applied TEIRESIAS to a well-known piece of literature, namely "A Tale of Two Cities" by Charles Dickens. After mapping all nouns to their singular form and the verbs to the corresponding infinitive, we compiled the text's vocabulary which contained 7768 words. We then replaced each paragraph of the original text by a stream of integers corresponding to the vocabulary index of the words that were replaced. TEIRESIAS subsequently processed the integer streams searching for patterns that had a minimum allowed density of $L = 4$ integers/words that were not more than $W = 8$ positions apart. In Fig. 9, we are showing some of the discovered patterns: for each pattern, we are also showing the number of its instances followed by the number of paragraphs that contained these instances. As before, the wild cards are placeholders for events (in this case, words) that could not be dereferenced.

Finally, in closing, and in addition to all of the mentioned applications of pattern discovery in molecular biology, we should also mention its application to the discovery of regulatory elements in genomic DNA. So far, there has been a small number of publications that address this topic from a data-mining perspective [9, 10]. However, given the successful results reported therein, we anticipate an increase in the number of practitioners that will pursue this line of research in the foreseeable future.

```
4  4  THE . . . . ONE THOUSAND SEVEN HUNDRED AND
4  4  OF LIBERTY EQUALITY FRATERNITY OR DEATH
4  3  THE GOOD REPUBLICAN BRUTUS OF ANTIQUITY
[...]
3  3   I BE THE RESURRECTION AND THE LIFE SAITH THE LORD HE THAT BELIEVETH IN ME THOUGH HE BE
DEAD
 YET SHALL HE LIVE AND WHOSOEVER LIVETH AND BELIEVETH IN ME SHALL NEVER DIE
3  3  THE . OF THE GOOD REPUBLICAN BRUTUS OF ANTIQUITY
3  3  REPUBLIC ONE AND INDIVISIBLE OF LIBERTY EQUALITY FRATERNITY OR DEATH
3  3  NIGHT IN . ONE THOUSAND SEVEN HUNDRED AND
3  3  THE . OF . . ONE THOUSAND SEVEN HUNDRED AND
3  3  CATCH HOLD OF . THROAT AND CHOKE . FOR HALF A GUINEA
3  3  MADAME DEFARGE . HER WAY ALONG THE STREET
3  2  THE . MY HUSBAND MY FATHER AND MY BROTHER
[...]
2  2  FOR YOUR OWN ADVANTAGE YOU WILL . . THE
2  2  FOULON WHO TELL . . . THAT . MIGHT GRASS
2  2  OF THE . . . WOUND ABOUT HER HEAD . . . HAVE
2  2  STILL MADAME DEFARGE PURSUE HER WAY ALONG THE STREET COME NEAR AND NEAR
2  2  CARTON STILL DRINKING THE PUNCH REJOIN WHY SHALL I
2  2  FOR THE LOVE OF HEAVEN OF JUSTICE OF GENEROSITY OF THE HONOUR OF YOUR NOBLE NAME
2  2  A WHITE-HAIRED MAN SAT ON A LOW BENCH STOOPING FORWARD AND VERY BUSY MAKE SHOE
2  2  LEAN AGAINST THE DOOR-POST KNITTING AND SAW NOTHING
2  2  I'D CATCH HOLD OF YOUR THROAT AND CHOKE YOU FOR HALF A GUINEA
2  2  THE OLD SYDNEY CARTON OF OLD SHREWSBURY SCHOOL
2  2  BUT NOT STRAIGHT HAVE A PECULIAR INCLINATION TOWARDS THE LEFT CHEEK
2  2  WITH HIS . . . THE . OF HIS . . HIS . . THE
2  2  TWO HUNDRED AND FIFTY PARTICULAR FRIEND
2  2  THE . OF . . FROM WHICH HE HAVE . . . . TO
2  2  PRECIPITATE HIMSELF OVER THE HILL-SIDE . . AS
2  2  LOOK BACK LOOK BACK AND SEE IF WE BE PURSUE
2  2  CHAPTER . THE . CHAPTER . THE . CHAPTER
2  2  AND . . . THE . MONSIEUR THE MARQUIS . HIS
2  2  CHAPTER I . . CHAPTER II THE . CHAPTER III THE
2  2  THE . OF . . CRUNCHER SITTING ON . STOOL IN
2  2  IN THE HOLLOW . THE . AND . THE . . THE
2  2  FROM THE STREET . . HIGH WALL AND . STRONG GATE
2  2  REPUBLIC ONE AND INDIVISIBLE LIBERTY EQUALITY FRATERNITY OR DEATH
[...]
```

FIG. 9. Some of the patterns that are present in "A Tale of Two Cities" by Charles Dickens. Only *elementary* patterns involving a minimum of four words such that the first and last words were not more than seven words apart were sought. Preceding each pattern is the number of occurrences (first number column) and the number of paragraphs containing those occurrences (second number column).

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

5. DISCUSSION

In this paper, we discussed the increasing use of pattern discovery techniques in addressing problems from computational biology. We began with a historical perspective on some of the algorithms that have been proposed over the years, the type of patterns that they handled, and the properties they exhibited. We then proceeded with the presentation of several important computational biology applications, the types of approaches that have been used so far in tackling them, and some of the newer approaches that revolve around the use of pattern discovery ideas. This by no means represents an exhaustive coverage of all the methods that have appeared in the literature and should instead be considered a rather short treatise on the possibilities that are opening up for people in the computational biology domain as a result of the availability of pattern discovery-tools. Our intent was to briefly describe some of the more prominent applications of the basic techniques in the hope that the readers will borrow ideas from these methods and devise new ones.

Note added in proof. Access to Web-based versions of some of the described algorithms, as well as downloadable executable code and Bio-Dictionaries for several complete genomes, is provided through the Bioinformatics and Pattern Discovery Group's Web page at http://www.research.ibm.com/bioinformatics/.

REFERENCES

1. Abola, E. E., Sussman, J. L., Prilusky, J., and Manning, N. O. (1997). Protein data bank archives of three-dimensional macromolecular structures, *Methods Enzymol.* **277**, 556–571.
2. Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool, *J. Mol. Biol.* **5**(3), 403–410.
3. Al-t-sch, S. F. (1989). Gap costs for multiple sequence alignment, *J. Theor. Biol.* **138**(3), 297–309.
4. Argos, P., and Vingron, M. (1990). Sensitivity comparison of protein amino acid sequences, *Methods Enzymol.* **183**, 352–365.
5. Attwood, T. K., Beck, M. E., Flower, D. R., Scordis, P., and Selley, J. N. (1998). The PRINTS protein fingerprint database in its fifth year, *Nucleic Acids Res.* **26**(1), 304–308.
6. Bailey, T. L., and Elkan, C. (1995). The value of prior knowledge in discovering motis with MEME, *in* "Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB '95)," Menlo Park, California, AAAI Press.
7. Bairoch, A., and Apweiler, R. (1998). The SWISS-PROT protein sequence data bank and its supplement TrEMBL in 1998, *Nucleic Acids Res.* **26**(1), 38–42.
8. Bairoch, A., Bucher, P., and Hofmann, K. (1996). The PROSITE database: Its status in 1995, *Nucleic Acids Res.* **24**, 189–196.
9. Brazma, A., Jonassen, I., Vilo, J., and Ukkonen, E. (1998). Predicting gene regulatory elements in silico on a genomic scale, *Genome Res.* **8**(11), 1202–1215.
10. Brazma, A., Vilo, J., Ukkonen, E., and Valtonen, K. (1997). Data mining for regulatory elements in yeast genome, *in* "Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology (ISMB '97)," pp. 65–74.
11. Ben-Dor, A., and Yakhini, Z. (1999). Clustering gene expression patterns, *in* "Proceedings of the Third Annual ACM International Conference on Computational Molecular Biology (RECOMB '99)," Lyon, France.
12. Benson, G., and Waterman, M. S. (1994). A method for fast database search for all k-nucleotide repeats, *in* "Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology (ISMB '94)," pp. 83–98.
13. Benson, G. (1998). An algorithm for finding tandem repeats of unspecified pattern size, *in* "Proceedings of the Second Annual ACM International Conference on Computational Molecular Biology (RECOMB '98)," New York, NY.
14. Blaschke, C., Andrade, M. A., Ouzounis, C., and Valencia, A. (1999). Automatic extraction of biological information from scientific text: Protein–protein interactions, *in* "Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB '99)," Menlo Park, California, AAAI Press.
15. Brazma, A., Jonassen, I., Eidhammer, I., and Gilbert, D. (1998). Approaches to the automatic discovery of patterns in biosequences, *J. Comput. Biol.* **5**(2), 279–305.
16. Bucher, P., and Bairoch, A. (1994). A generalized profile syntax for biomolecular sequence comparison methods, *in* "Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology (ISMB '94)".
17. Bystroff, C., and Baker, D. (1998). Prediction of local structure in proteins using a library of sequence-structure motifs, *J. Mol. Biol.* **281**(3), 565–577.
18. Califano, A. SPLASH: structural pattern localization analysis by sequential histograms, *Bioinformatics*, in press.
19. Carrillo, H., and Lipman, D. (1988). The multiple sequence alignment problem in biology, *SIAM J. Appl. Math.*, 1073–1082.
20. Craven, M., and Kumlien, J. (1999). Constructing biological knowledge bases by extracting information from text sources, *in* "Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB '99)," Menlo Park, California, AAAI Press.
21. Dayhoff, M. (1978). Atlas of protein sequence and structure, *Nat. Biomed. Res. Found.* **5**.
22. Delcoigne, A., and Hansen, P. (1975). Sequence comparison by dynamic programming, *Biometrika* **62**, 661–664.
23. D'haeseleer, P., Wen, X., Fuhrman, S., and Somogyi, R. (1997). Mining the gene expression matrix: Infering gene relationships form large scale gene expression data, *in* "Proceedings of the International Workshop on Information Processing in Cells and Tissues," pp. 203–212.
24. Eisen, M. B., Spellmann, P. T., Brown, P. O., and Botstein, D. (1998). Cluster analysis and display of genome-wide expression patterns, *Proc. Natl. Acad. Sci. USA* **95**(25), 14863–14868.
25. Feng, D., and Doolittle, R. (1987). Progressive sequence alignment as prerequisite to correct phylogenetic trees, *J. Mol. Evol.* **25**, 351–360.
26. Floratos, A., Rigoutsos, I., Parida, L., Stolovitzky, G., and Gao, Y. (1999). Sequence homology detection through large-scale pattern discovery, *in* "Proceedings of the Third Annual ACM International Conference on Computational Molecular Biology (RECOMB '99)," Lyon, France.
27. Floratos, A. (1999). "Applications of pattern discovery in computational biology," Ph.D. thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York Univ.
28. Floratos, A., and Rigoutsos, I. (1998). On the time complexity of the TEIRESIAS algorithm, IBM Technical Report RC 21161 (94582). IBM TJ Watson Research Center.

29. Fodor, S. P., Read, J. L., Pirrung, M. C., Stryer, L., Lu, A. T., and Solas, D. (1991). Light-directed, spatially addressable parallel chemical synthesis, *Science* **251**(4995), 767–673.
30. Gao, Y., Mathee, K., Narasimhan, G., and Wang, X. (1999). Motif detection in protein sequences, *in* "Proceedings SPIRE '99," Cancun, Mexico.
31. Gao, Y. (1997). "Detection of helix-turn-helix motifs in proteins," M.Sc. thesis, Department of Mathematical Sciences, University Memphis, Memphis, TN.
32. Gao, Y., Rigoutsos, I., Floratos, A., Parida, L., and Narasimhan, G. Unsupervised building and exploitation of composite decriptors for collections of proteins and protein fragments, submitted.
33. Gao, Y., Yang, M., Wang, X., Mathee, K., and Narasimhan, G. (1997). Detection of HTH motifs via data mining. International Conference on Bioinformatics, Nov. 6–9, Atlanta.
34. Garey, M. R., and Johnson, D. S. (1979). "Computers and Intracctability: A Guide to the Theory of NP-Completeness."
35. George, D. G., Barker, W. C., and Hunt, L. T. (1990). Mutation data matrix and its uses, *in* "Methods in Enzymology" (R. F. Doolittle, Ed.), Vol. 183, p. 338, Academic Press, New York.
36. Gribskov, M., McLachlan, M., and Eisenberg, D. (1987). Profile analysis: Detection of distantly related proteins, *Proc. Natl. Acad. Sci. USA* **84**, 4355–4358.
37. Grundy, W. N., Bailey, T. L., Elkan, C. P., and Baker, M. E. (1997). Meta-MEME: Motif-based hidden Markov models of protein families, *CABIOS* **13**, 397–406.
38. Guan, X., and Uberbacher, E. C. (1996). A first look-up algorithm for detecting repetitive DNA sequences, *in* "Proceedings of the Pacific Symposium on Biocomputing," pp. 718–719.
39. Gusfield, D. (1997). "Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology," Cambridge Univ. Press.
40. Han, K. F., Bystroff, C., and Baker, D. (1997). Three-dimensional structures and contexts associated with recurrent amino acid sequence patterns, *Protein Sci.* **6**(7), 1587–1590.
41. Henikoff, S., and Henikoff, J. (1991). Automatic assembly of protein blocks for database searching, *Nucleic Acids Res.* **19**, (23) 6565–6572.
42. Henikoff, S., and Henikoff, J. (1994). Protein family classification based on searching a database of Blocks, *Genomics* **19**, 97–107.
43. Henikoff, S., and Henikoff, J. (1992). Amino acid substitution matrices form protein blocks, *Proc. Natl. Acad. Sci. USA* **89**, 100915–100919.
44. Hirosawa, M., Totoki, Y., Hishida, M., and Ishikawa, M. (1995). Comprehensive study on iterative algorithms of multiple sequence alignment, *CABIOS*.
45. Jonassen, I., Eidhammer, I., and Taylor, W. R. (1999). Discovery of local packing motifs in protein structures, *Proteins Struct. Funct. Genet.* **34**(2), 206–219.
46. Jonassen, I., Collins, J. F., and Higgins, D. G. (1995). Finding flexible patterns in unaligned protein sequences, *Protein Sci.* 1587–1595.
47. Karp, R. M., Miller, R. E., and Rosenberg, A. L. (1972). Rapid identification of repeated patterns in strings, trees and arrays, *in* "Proceedings of the ACM Symposium on the Theory of Computing," pp. 125–136.
48. Krogh, A., Brown, M., Mian, I., Sjoelander, K., and Haussler, D. (1994). Hidden Markov model in computational biology: Applications to protein modeling, *J. Mol. Biol.* **235**, 1501–1531.
49. Landau, G., and Schmidt, J. (1993). An algorithm for approximate tandem repeats, *in* "Proceedings of the Fourth Symposium on Combinatorial Pattern Matching," Springer LNCS 684, pp. 120–133.
50. Lawrence, C., and Reilly, A. (1990). An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences, *Proteins Struct. Funct. Genet.* **7**, 41–51.
51. Lewin, B. (1997). "Genes VI," Oxford Univ. Press, New York.
52. Liang, S., Fuhrman, S., and Somogyi, R. (1998). REVEAL, a general reverse engineering algorithm for inference of genetic network architectures, *in* "Proceedings of the Pacific Symposium on Biocomputing," Vol. 3, pp. 18–29.
53. Maier, D. (1978). The complexity of some problems on subsequences and supersequences, *J. ACM* 322–336.
54. Martinez, M. (1988). A flexible multiple sequence alignement program, *Nucleic Acids Res.* 1683–1691.
55. Michaels, G., Carr, D. B., Wen, X., Fuhrman, S., Askenazi, M., and Somogyi, R. (1998). Cluster analysis and data visualization of large-scale gene expression data, *in* "Proceedings of the Pacific Symposium on Biocomputing," Vol. 3, pp. 42–53.
56. Miller, W., Zhang, Z., and He, B. (1996). Local multiple alignment via subgraph enumeration, *Disc. Appl. Math.* **71**, 337–365.
57. Milosavljevic, A., and Jurka, J. (1993). Discovering simple DNA sequences by the algorithmic significance method, *Comp. Appl. Biosc.* **9**, 407–411.
58. Needlman, 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.
59. Neuwald, A. F., and Green, P. (1994). Detecting patterns in protein sequences, *J. Mol. Biol.* 698–712.
60. Neville-Manning, C. G., Sethi, K. S., Wu, D., and Brutlag, D. L. (1997). Enumerating and ranking discrete motifs, *in* "Proceedings of the Fifth International Conference on Intelligent Systems for Molecular Biology," Chalkidiki, Greece.
61. Parida, L., Floratos, A., and Rigoutsos, I. (1999). An approximation algorithm for alignment of multiple sequences using motif discovery, *J. Combinat. Optimiz.* **3**, 247–275.
62. Parida, L., Floratos, A., and Rigoutsos, I. (1998). MUSCA: An algorithm for constrained aligment of multiple data sequences, *in* "Proceedings of the Ninth Workshop on Genome Informatics," Tokyo, Japan.
63. Pearson, W. R., and Lipman, D. J. (1988). Improved tools for biological sequence comparison, *Proc. Natl. Acad. Sci. USA* **85**(8), 2444–2448.
64. Primrose, S. B. (1995). "Principles of Genome Analysis," Blackwell Science, Cambridge, MA.
65. Rigoutsos, I., and Floratos, A. (1998). Motif discovery without alignment or enumeration, *in* "Proceedings of the 2nd annual ACM International Conference on Computational Molecular Biology (RECOMB)," New York, NY.
66. Rigoutsos, I., and Floratos, A. (1998). Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm, *Bioinformatics* **14**(1), 55–67.
67. Rigoutsos, I., Floratos, A., Ouzounis, C., Gao, Y., and Parida, L. (1999). Dictionary building via unsupervised hierarchical motif discovery in the sequence space of natural proteins, *Proteins Struct. Funct. Genet.* **37**(2).
68. Rigoutsos, I., Gao, Y., Floratos, A., and Parida, L. (1999). Building dictionaries of 1D and 3D motifs by mining the unaligned 1D sequences of 17 archaeal and bacterial genomes, *in* "Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology (ISMB '99)," Menlo Park, California, AAAI Press.
69. Roytberg, M. A. (1992). A search for common patterns in many sequences, *CABIOS* 57–64.
70. Sagot, M.-F., Viari, A., and Soldano, H. (1995). Multiple sequence comparison: A peptide matching approach, *in* "Proceedings of the Sixth Symposium on Combinatorial Pattern Matching," pp. 366–385.
71. Sagot, M.-F., and Viari, A. (1996). A double combinatorial approach to discovering patterns in biological sequences, *in* "Proceedings of the Seventh Symposium on Combinatorial Pattern Matching," pp. 186–208.

Metabolic Engineering **2**, 159–177 (2000)
doi:10.1006/mben.2000.0151

72. Saqi, M. A., and Sternberg, M. J. (1994). Identification of sequence motifs from a set of proteins with related function, *Protein Eng*. **7**(2), 165–171.
73. Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray, *Science* **270**(5235), 467–470.
74. Simons, K. T., Ruczinski, I., Kooperberg, C., Fox, B. A., Bystroff, C., and Baker, D. (1999). Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins, *Proteins Struct. Funct. Genet*. **34**(1), 82–95.
75. Smith, R. F., and Smith, T. F. (1992). Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap renalties for use in comparative protein modelling, *Protein Eng*. **1**, 35–41.
76. Smith, T. F., Waterman, M. S., and Fitch, W. M. (1981). Comparative biosequence metrics, *J. Mol. Evol*. **18**, 38–46.
77. Smith, T. F., and Waterman, M. S. (1981). Identification of common molecular subsequences, *J. Mol. Biol*. **147**, 195–197.
78. Smith, H. O., Annau, T. M., and Chandrasegaran, S. (1990). Finding sequence motifs in groups of functionally related proteins, *Proc. Natl. Acad. Sci. USA*, 826–830.
79. Smith, R. F., and Smith, T. F. (1990). Automatic generation of primary sequence patterns from sets of related protein sequences, *Nucleic Acids Res*., 118–122.
80. Sobel, E., and Martinez, M. (1986). A multiple sequence alignment program, *Nucleic Acids Res*., 363–374.
81. Sonnhammer, E. L., Eddy, S. R., and Durbin, R. (1997). Pfam: A comprehensive database of protein domain families based on seed alignments, *Proteins Struct. Funct. Genet*. **28**(3), 405–420.
82. Stolovitzky, G., Gao, Y., Floratos, A., and Rigoutsos, I. (1999). Tandem repeat detection using pattern discovery with application to the identification of yeast satellites. IBM Technical Report RC 21508 (96944), IBM TJ Watson Research Center.
83. Suyama, M., Nishioka, T., and Jun'ichi, O. (1995). Searching for common sequence patterns among distantly related proteins, *Protein Eng*. 1075–1080.
84. Tamayo, P., Slonim, D., Mesirov, I., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E. S., and Golub, T. R. (1999). Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation, *Proc. Natl. Acad. Sci. USA* **96**(6), 2907–2912.
85. Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, *Nucleic Acids Res.* **22**(22), 4673–4680.
86. 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*., 2769–2775.
87. Wang, J., Chirn, G., Marr, T. G., Shapiro, B. A., Shasha, D., and Zhang, K. (1994). Combinatorial pattern discovery for scientific data: Some preleminary results, *in* "Proceedings of the ACM SIGMOD Conference on Management of Data," pp. 115–124.
88. Wang, I., and Jiang, T. (1994). On the complexity of multiple sequence alignment, *J. Comput. Biol*. 337–348.
89. Waterman, M. S., Galas, D. J., and Arratia, R. (1984). Pattern recognition in several sequences: Consensus and alignment, *Bull. Math. Biol*. **46**, 515–527.
90. Wen, X., Fuhrman, S., Michaels, G. S., Carr, D. B., Smith, S., Barker, L., and Somogyi, R. (1998). Large scale temporal gene expression mapping of CNS development, *Proc. Natl. Acad. USA* **95**, 334–339.
91. Wu, T. D., and Brutlag, D. L. (1995). Identification of protein motifs using conserved amino acid properties and partitioning techniques, *in* "Proceedings of the Third International Conference on Intelligent Systems for Molecular Biology (ISMB '95)," pp. 402–410.