Sequence analysis

An efficient, versatile and scalable pattern growth approach to mine frequent patterns in unaligned protein sequences

Kai Ye1,*, Walter A. Kosters2 and Adriaan P. IJzerman1

1Division of Medicinal Chemistry, Leiden/Amsterdam Center for Drug Research and 2Leiden Institute of Advanced Computer Science, Leiden University, Leiden, The Netherlands

Received on October 24, 2006; revised on December 12, 2006; accepted on December 27, 2006

Advance Access publication January 19, 2007

Associate Editor: Limsoon Wong

ABSTRACT

Motivation: Pattern discovery in protein sequences is often based on multiple sequence alignments (MSA). The procedure can be computationally intensive and often requires manual adjustment, which may be particularly difficult for a set of deviating sequences. In contrast, two algorithms, PRATT2 (http://www.ebi.ac.uk/pratt/) and TEIRESIAS (http://cbcsv.watson.ibm.com/) are used to directly identify frequent patterns from unaligned biological sequences without an attempt to align them. Here we propose a new algorithm with more efficiency and more functionality than both PRATT2 and TEIRESIAS, and discuss some of its applications to G protein-coupled receptors, a protein family of important drug targets.

Results: In this study, we designed and implemented six algorithms to mine three different pattern types from either one or two datasets using a pattern growth approach. We compared our approach to PRATT2 and TEIRESIAS in efficiency, completeness and the diversity of pattern types. Compared to PRATT2, our approach is faster, capable of processing large datasets and able to identify the so-called type III patterns. Our approach is comparable to TEIRESIAS in the discovery of the so-called type I patterns but has additional functionality such as mining the so-called type II and type III patterns and finding discriminating patterns between two datasets.

Availability: The source code for pattern growth algorithms and their pseudo-code are available at http://www.liacs.nl/home/kosters/pg/

Contact: k.ye@lacdr.leidenuniv.nl

1 INTRODUCTION

One of the crucial topics in the analysis of biological data is the discovery of frequent patterns in a set of DNA or protein sequences. These patterns usually hint at shared biological functions. For example, some patterns may be essential for the proteins to fold correctly while other patterns may form a certain micro-environment to recognize a small-molecule ligand or another protein partner. Various algorithms have been designed to identify such patterns either by overlaying protein structures (Copley et al., 2001; Lupas et al., 2001; Russell et al., 1998) or by mining in sequences. For the latter, most pattern discovery approaches either use aligned sequences as an input or create a multiple sequence alignment (MSA) in an early stage of the analysis such as PRINTS (Attwood et al., 2003), PROSITE (Hulo et al., 2006) and Pfam (Baldi and Chauvin, 1994; Baldi et al., 1994; Bateman et al., 1994; Shigeta et al., 2003). In addition to MSA, some algorithms such as correlation-based approaches (Kuipers et al., 1997), evolutionary trace (Lichtarge et al., 1996) and two-entropies analysis (Ye et al., 2006) even include a phylogeny to uncover further information about functional sites. Construction of such MSAs, however, requires parameterization, is computationally intensive, often requires manual adjustment, and can be particularly difficult for a set of deviating sequences.

In contrast, TEIRESIAS (Rigoutsos and Floratos, 1998) and PRATT2 (Jonassen, 1995; Jonassen et al., 1997) are used to directly identify frequent patterns from biological sequences without aligning them. The combinatorial pattern discovery algorithm TEIRESIAS identifies patterns in two stages. In the first stage of scanning the elementary patterns, TEIRESIAS identifies all elementary patterns of length at most W, with at least L non-wildcard items (W ≥ L). The elementary patterns must show up in at least K sequences of the input, the so-called support of the pattern. In the second stage of elementary pattern convolution, all elementary patterns are combined into larger patterns until no more patterns emerge. Although TEIRESIAS reports all patterns which fulfill the predefined parameters and does not score and rank patterns, it provides utilities in its web server to filter out some patterns based on properties such as the number of non-wildcard items in the patterns. Using a pattern graph, PRATT2 searches for conserved patterns showing up in at least k out of n sequences. It uses the (n − k + 1) shortest sequences to construct a pattern graph and then uses the pattern graph to mine patterns. The identified patterns are scored and only the top 50 patterns are reported by default. PRATT2 can also identify patterns with limited flexibility in the number of consecutive wildcards. In the additional refinement stage, it replaces some wildcards with ambiguous residues if it can find any.

In this study, we followed the ideas of pattern growth that are emerging in computer science (Pei et al., 2004) and designed six algorithms to provide a complete solution of pattern discovery in unaligned protein sequences. In computer science, quite a few studies have contributed to the efficient mining of
A, T and D placed in that order in an eight residues peptide A*T*D with a window of eight matches protein sequences with predefined length (window). For example, a type III pattern which have the residues placed in the right order within a pattern set efficiently.

We grew patterns continuously in a so-called projected database. The projected databases kept on shrinking by eliminating sequences which did not support the current pattern. None of the present extensions in sequential pattern mining allows finding the number of wildcards between frequent items so that they cannot yield biologically meaningful PROSITE-like patterns. In this study, we used the principles of wildcard constraints to mine PROSITE-like patterns and frequent items and their sequential order but also the number of wildcards between frequent items carries essential information.

The computation of \( S_d \) from \( S_a \) requires, for each \( a \)-location, the check whether or not the residue on its right side (if any) equals the newly appended item (b or x). In case of equality this gives an \( a' \)-location in \( S_a \). Sequences without \( a' \)-locations are deleted.

The main call is \( PGOne (\Lambda, S_d) \), where \( \Lambda \) is the empty pattern; note that each residue in database \( S_a \) is a \( A \)-location, including the position before each sequence. This call creates a projected database that marks all occurrences of a single residue \( b \), reports all frequent patterns that begin with this \( b \), and then proceeds to the next amino acid.

We also adapted our algorithms to mine two datasets at the same time to identify one of the three pattern types to discriminate between the two datasets.

2 METHODS

2.1 Algorithms

A pattern is an ordered series of items, where each item is either one of 20 residues or three special wildcard characters, \( x \), \( x(i, j) \) or \( * \) in which \( x \) matches any of 20 residues, \( x(i, j) \) matches \( i \) to \( j \) (\( 0 \leq i \leq j \)) residues, \( * \) matches any combination of residues with undefined length. The support of a pattern in a database is the ratio of sequences that satisfy the pattern over all sequences. Note that if a pattern occurs twice or more in a sequence, it is only counted once.

Since the algorithms for mining the three types of patterns from one dataset are very similar, we describe the algorithm for type I patterns from one dataset (PGOneI, PG stands for pattern growth) in detail. The descriptions of the other two, PGOneII and PGOneIII are available at http://www.liacs.nl/home/kosters/pg/. A method similar to PGOneIII, in which a length constraint is introduced into the sequential pattern mining process, was also described by Pei et al. (2002).

The inputs of the algorithm are dataset \( S \) which consists of a series of non-empty sequences, a minimum support threshold \( \text{min}_\text{support} \), the minimal number of non-wildcard items in the patterns to be reported \( \text{min}_\text{non}_\text{wc} \) and the maximal number of consecutive wildcards \( x \max\_\text{wc}_f \).

The output of the algorithm consists of all patterns that have support larger than or equal to \( \text{min}_\text{support} \), and that satisfy the other input parameters. The supports are also given. These patterns are called frequent.

The algorithm is as follows. Here \( a \) is a pattern and \( S_a \) is the so-called projected database that contains all sequences that satisfy the pattern \( a \), where the last element of each occurrence of \( a \) is marked (the so-called \( a \)-locations). Note that \( S_a \), the number of sequences in \( S_a \), is the support of \( a \). Let \( \text{non}_\text{wc} (a) \) be the number of non-wildcard items in \( a \), \( \text{curr}_\text{wc}_f (a) \) the number of consecutive wildcards at its end and \( \text{last} (a) \) its last item.

\[
\begin{align*}
\text{PGOneI} & (a, S_a) \\
& \text{if } |S_a| \geq \text{min}_\text{support} \\
& \text{if } \text{non}_\text{wc} (a) \geq \text{min}_\text{non}_\text{wc} \text{ and } \text{last} (a) \neq x \text{ then} \\
& \text{report } a, |S_a| \\
& \text{for each residue } b \text{ do} \\
& \quad d' := a \text{ with } b \text{ appended to it} \\
& \quad \text{PGOneI} (a', S_a) \\
& \text{if } a \neq \Lambda \text{ and } \text{curr}_\text{wc}_f (a) < \max\_\text{wc}_f \text{ then} \\
& \quad d' := a \text{ with } x \text{ appended to it} \\
& \quad \text{PGOneI} (a', S_a)
\end{align*}
\]

We compared our program with PRATT2 and TEIRESIAS in identifying the type I patterns since all three are able to find this type of patterns. After that we compared our program with PRATT2 in finding type II patterns. We also investigated the performance of our program in mining the type III patterns which are novel in protein sequence analysis. In addition, we adapted our algorithms to mine two datasets at the same time. This allowed us to identify one of the above pattern types to discriminate between the two datasets.

From one dataset, we can identify three different types of patterns. Type I is a pattern with a defined number of wildcards such as A*xT. Type II is a PROSITE-like pattern (Hulo et al., 2006). For example, Ax(3,4)DF is a pattern with three residues and one limited flexible wildcard region. It matches any protein sequence \( A \) followed by three or four residues and ending with DF. Type III patterns match protein sequences which have the residues placed in the right order within a predefined length (window). For example, a type III pattern A*T*D with a window of eight matches protein sequences with A, T and D placed in that order in an eight residues peptide (window).

The main call is \( PGOne (\Lambda, S_d) \), where \( \Lambda \) is the empty pattern; note that each residue in database \( S_a \) is a \( A \)-location, including the position before each sequence. This call creates a projected database that marks all occurrences of a single residue \( b \), reports all frequent patterns that begin with this \( b \), and then proceeds to the next amino acid.

We also adapted our algorithms to mine two datasets at the same time to identify one of the three pattern types to discriminate between the two datasets. Two sequence datasets (one defined as positive dataset and the other as negative) are required. These two datasets were recursively mined at the same time and the support difference was
evaluated and compared with a predefined \( \text{min\_support\_diff} \). Only when the number of non-wildcards is no less than the predefined minimal number of non-wildcards \( \text{min\_wc\_eva} \), the evaluation of the support difference starts since the supports are high in both datasets in the beginning of the mining process. Only those patterns with support difference no less than \( \text{min\_support\_diff} \) between the positive dataset and negative dataset are reported.

2.2 Comparison with PRATT2 and TEIRESIAS

In life sciences, TEIRESIAS and PRATT2 are the two state-of-the-art algorithms that identify patterns from a set of unaligned protein sequences. Therefore, we compared our PGOneI algorithm with these two programs with respect to the identification of type I patterns from one dataset. In addition, PGOneII was compared with PRATT2 (current version pratt2.1) in identifying type II patterns in one dataset. The PRATT2 algorithm developed by Jonassen et al. (1995) was downloaded from http://iubio.bio.indiana.edu/soft/molbio/pattern/. The TEIRESIAS algorithm was obtained from http://cbsrv.watson.ibm.com/. The implementations of our algorithms, PRATT2 and TEIRESIAS were compiled/installed under Red Hat Enterprise Linux WS 3.0. All tests were performed on the same PC with Intel Pentium 4 CPU 3.20 GHz, 1.00 GB of RAM and a Maxline III 7L250S0 250 GB Hard Drive.

2.3 Datasets

G protein-coupled receptors (GPCRs) are important drug targets. They are membrane-bound, serpentine-like, proteins with seven transmembrane (TM), alpha-helical, domains. A subdivision in three classes has been proposed of which class A (rhodopsin-like) is by far the largest. We used class A GPCRs as our test set for at least two reasons. First, class A GPCRs contain thousands of divergent protein sequences, so that we can study the properties of our algorithms in various sizes of real sequences. Second, class A GPCRs includes two big divergent datasets (olfactory receptors and non-olfactory receptors) which facilitate demonstration of our algorithms in two datasets.

Protein sequences of class A GPCRs were extracted from the GPCRDB (March 2005 release (9.0); http://www.gpcr.org/7tm/). We removed orphan receptors and split the sequences into two datasets which happen to be of similar size: olfactory (2034) and non-olfactory (2027). This large number of proteins and two subsets with similar sizes facilitate benchmarking of our various implementations with different functionalities. It also helped us to compare our algorithms with both PRATT2 and TEIRESIAS which also aim to discover patterns from unaligned protein sequences. To illustrate the impact of dataset size on the performance of our algorithms and the above two programs, we randomly extracted protein sequences from the non-olfactory dataset to form a series of datasets with different sizes (20, 50, 100, 200, 300, 400, 500, 600, 1000, 2000).

2.4 Reference type I patterns

In order to evaluate pattern discovery by PRATT2, TEIRESIAS and PGOneI from one dataset, we defined reference type I patterns though MSA as a ‘golden standard’. We extracted the MSA for the TM domains of all non-olfactory proteins from the GPCRDB and calculated the Shannon entropy for each position. We considered the 32 positions with an entropy value <1.5 as the globally conserved positions based on our previous study (Ye et al., 2006). Then we combined the most frequent residue at the defined conserved positions in each helix as a pattern.

The reference patterns (32 defined positions) are GxxGNxxV for helix 1, FLxxLxxADL for helix 2, SxxxLxxISxDRY for helix 3, FxxPxxxxxxxY for helix 5, FxxCWxPF for helix 6, LxxxNSxxNPxxY for helix 7.

Fig. 1. Overview of six implementations of the pattern growth approach.

3 RESULTS

3.1 Implementations of the pattern growth algorithm

Six programs were implemented to identify frequent patterns using the principle of pattern growth. Three of them, PGOneI, PGOneII and PGOneIII aim to mine frequent patterns from one dataset while the other three, PGTwoI, PGTwoII and PGTwoIII work at two datasets at the same time to find patterns that discriminate members of a positive dataset from those of a negative dataset. Figure 1 depicts an overview of our solutions to the problem of mining frequent patterns from unaligned protein sequences. ‘PG’ stands for ‘Pattern Growth’ since we use a pattern growth approach.

3.2 Characteristics of PGOneI

Since PGOneI, TEIRESIAS and PRATT2 are all able to find type I patterns from one dataset, we compared these three algorithms in efficiency and completeness of pattern discovery.

We first examined the efficiency when we varied the size of input, the support and the maximum number of consecutive wildcards. For the size of input, a series of random datasets from the non-olfactory protein set with various sizes were used as inputs and the runtimes were recorded. The three programs were used to find type I patterns from one dataset in two runs for different supports (0.3 for low support and 0.7 for high support). The maximal number of consecutive wildcards in the patterns was three. The refinement step of PRATT2 was turned off to speed it up.

As shown in Figure 2A, both TEIRESIAS and PGOneI succeeded in all tests and the runtimes are almost linearly correlated to the size of the input. PGOneI is slightly more efficient than TEIRESIAS. The runtime of PRATT2 increases dramatically when the size of the input increases. PRATT2 fails when the input contains 600 proteins independent of the levels of the support. For all three programs, the runtime for searching patterns with high support is of course much shorter than for those with low support.

To test how support affects the efficiency of the three algorithms, we used a random set with 200 proteins of non-olfactory receptors as input. As shown in Figure 2B, the performance of PGOneI is comparable to TEIRESIAS and much better than PRATT2.
We also addressed how the number of consecutive wildcards affects the efficiency. We recorded the runtime of the three algorithms in search of patterns with a different maximum number of consecutive wildcards from a random set with 200 proteins of non-olfactory receptors. Figure 2C shows again that PGOneI is comparable to TEIRESIAS and much better than PRATT2.

To evaluate the completeness of PGOneI, PRATT2 and TEIRESIAS in finding type I patterns, we ran the three programs using the dataset with 500 non-olfactory proteins as input since PRATT2 failed at an input of 600 proteins. The three programs were configured to find type I patterns with the maximum number of consecutive wildcards equal to three. The identified patterns were required to be present in at least 150 proteins. TEIRESIAS reported all patterns which satisfy the requirement. PGOneI yielded the same result as TEIRESIAS when we had it report all patterns (data not shown). However, since the probability of having a pattern with two non-wildcards in any protein of say 400 residues is close to one, a lot of meaningless two non-wildcard patterns were diluting the pattern pool. As shown in Table 1, if we let PGOneI only report patterns with at least 3 non-wildcard residues, PGOneI yielded only about 1/3 of the patterns without missing any important patterns compared to TEIRESIAS. PRATT2 identified less functional positions because it ranks patterns by its scoring function and reports only the top ones (by default, PRATT2 reports the top 50 patterns and it is not able to report all patterns satisfying the parameters even when the maximal number of patterns to report is set big enough). Thus the mining results of PRATT2 are incomplete and important patterns may be missing. None of the three algorithms found all predefined 32 positions (see Section 2.4) for two reasons. First, the predefined maximum number of consecutive wildcards was equal to three, prohibiting these three algorithms to find the pattern FxxPxxxxxxY. Second, a combination of several most frequent residues may not be frequent enough to be detected, although the most frequent residue on an individual position is.

### 3.3 Characteristics of PGOneII

We implemented PGOneII to mine type II patterns from one dataset. Since TEIRESIAS cannot find type II patterns, we only compared PGOneII with PRATT2.

We first used the dataset with 200 non-olfactory proteins as input to examine how the maximum flexibility affects the runtime of both PGOneII and PRATT2 in mining patterns with different support. The support values of 0.3 and 0.7 mean that the identified patterns must show up in at least 30 and 70% of proteins in the dataset, respectively. The maximum number of consecutive wildcards in the patterns was three. The refinement step of PRATT2 was turned off to speed it up. We varied the maximum flexibility from 0 to 2. When the maximum flexibility is 0, no flexibility in the number of wildcards is allowed; if 1, patterns such as Ax(2,3)T and Wx(0,1)S are allowed. As shown in Figure 3, PGOneII is more efficient than PRATT2, especially in mining patterns with low support.

Then we compared PGOneII with PRATT2 on the different sizes of inputs, different supports and various maximum numbers of consecutive wildcards. The results were similar to the comparison between PGOneI and PRATT2 shown in Figure 2 (data not shown). For example, PRATT2 failed again before the size of the datasets reaches 600 proteins while PGOneII was successful in all runs.

We used the random set of 500 proteins (non-olfactory receptors) to compare the completeness of pattern discovery between PRATT2 and PGOneII. The maximum number of consecutive wildcards was set at 3, support 0.7 and the maximum flexibility 1. PGOneII identified 1387 patterns. The maximum number of patterns to be reported by PRATT2 was set at 5000 but PRATT2 reported only 1111 patterns after it had scanned a total of 361 962 patterns. The patterns identified by PGOneII included all patterns found by PRATT2. PRATT2 failed to find some patterns such as Sx(2,3)Nx(0,1)PxxY which

---

**Table 1. Completeness of pattern discovery for PRATT2, TEIRESIAS and PGOneI**

<table>
<thead>
<tr>
<th>Program</th>
<th>Size of report</th>
<th>Number of positions identified out of 32</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRATT2</td>
<td>1000</td>
<td>23</td>
</tr>
<tr>
<td>TEIRESIAS</td>
<td>2742</td>
<td>28</td>
</tr>
<tr>
<td>PGOneI</td>
<td>927</td>
<td>28</td>
</tr>
</tbody>
</table>

---

K.Ye et al.
was identified by PGOneII and is well known to be important for class A GPCRs.

3.4 Characteristics of PGOneIII
Since neither TEIRESIAS nor PRATT2 is able to find type III patterns, we examined the properties of PGOneIII without comparison. We only show how various windows affect the runtime as the size of the input and the support affect PGOneIII in the same way as PGOneI.

First, we examined how the size of the input affects the runtime. The runtime was again linear with the size of the input, when we used a window of eight, no matter whether patterns with high or low support were searched for.

To understand how runtime changes when PGOneIII searches patterns with different support, we mined the dataset with 200 non-olfactory proteins for patterns having support from 0.1 to 1.0 in a window of eight. Runtime increases very fast when support drops from 1.0 to 0.1. When support is 1.0, most patterns will not grow long since the pattern and its offspring pattern will not be pruned anymore once it fails in any of the proteins in the input. When support drops towards 0, more and more branches of the pattern tree will be searched and eventually PGOneIII will dump every fragment of proteins with a slide window eight as a pattern when support is close to 0.0.

When we increase the window, PGOneIII will search patterns that cover longer fragments in the proteins. We mined the dataset of 200 non-olfactory proteins for patterns having a window from 5 to 15. As shown in Figure 4, runtime rose much slower in the search for patterns with high support than with low support.

3.5 Classification with PGTwoI, PGTwoII and PGTwoIII
We adapted the pattern growth approach to mine two datasets at the same time and find those patterns (one of the three types) with high support in the predefined positive dataset and low support in the negative one.

We found a series of motifs which distinguish non-olfactory receptors from olfactory receptors. As shown in Table 2, all these patterns are related and indicate that the well-known motif of a cluster of aromatic residues in helix 6 is unique for the non-olfactory receptors (Visiers et al., 2002).

We also examined which patterns distinguish olfactory from non-olfactory receptors and compared our findings with previous studies on this topic (Mombaerts, 1999). Among other findings, we learned that the sequence of ‘FSTCSSH’, reported to occur in TM6 of a number of olfactory receptors, could be updated as two related patterns ‘TCxxHxxxV’ and ‘KxxxTCxxH’, which show up in >80% of olfactory receptors while <0.2% of non-olfactory receptors have these. We also found a frequent pattern of ‘YxxxxxGN’ in TM1 which was not included in Mombaerts’s study.

4 DISCUSSION
4.1 Reduce search space by pattern growth
A seemingly straightforward algorithm to find patterns with length \( L \) in a dataset of \( n \) unaligned protein sequences is to generate and test each pattern candidate. Let us estimate,
however, how many pattern candidates are needed to find all type I patterns with a length of 10. Since at each position, both 20 residues and the wildcard X may show up, the total number of pattern candidates to be generated would be $2^{20} \sim 10^{13}$. Moreover, for each pattern candidate, the entire dataset would have to be scanned to record the number of protein sequences having the pattern candidate. Thus, a more sophisticated method is warranted.

In this study, we present a pattern growth solution to largely reduce the number of pattern candidates and the cost of scanning the dataset. We grow the pattern by adding elements to its ‘right-hand’ side. Once the pattern is no longer frequent, we stop growing it and its extension will not be tested. For example, if pattern AxxDxxxG is not frequent, we will not test patterns such as AxxDxxxGA, AxxDxxxGS and AxxDxxxGx. In this way, we significantly reduce the number of pattern candidates to be tested while all patterns that satisfy the requirement will be reported.

In addition, we avoid scanning the entire dataset for each pattern candidate. Obviously, if one protein does not contain a certain pattern, it will not contain the extension of that pattern either. For each pattern that is defined as frequent in the mining process, we create a projected database to store the index of the protein sequences which contain that pattern. The extensions of that pattern will only be tested on the proteins in the projected database.

4.2 Mining various pattern types by pattern growth

The pattern growth approach not only provided us with an efficient solution in pattern discovery but also enabled us to find various types of patterns.

In this study, we carefully controlled the ways to grow the pattern resulting in three different types of patterns. We expect that the pattern growth approach may find many other pattern types proposed in the future because of the excellent adaptability to various gap constraints.

The three pattern types in this study target protein regions with different levels of conservation. The type I pattern may capture the conserved positions in a region with secondary structure such as functional motifs in a helix or a beta sheet. The type I pattern is also the major outcome of MSA-based approaches.

In some cases, small insertions/deletions may happen in the region with secondary structure yielding type II patterns. For example, proline and/or double glycine are known to introduce kinks in a helix. To maintain the overall structure, an insertion/deletion may then be necessary, of which there are many examples in class A GPCRs. It is more difficult for MSA-based methods to find type II patterns compared to type I patterns.

We also introduce a type III pattern which may target protein fragments without a defined secondary structure. MSA-based methods are not suitable to find type III patterns. Novel as this new type of patterns is for protein sequences, we are faced with the question of their meaning. Type I and II patterns can be found and interpreted by overlaying homologous protein structures. However, type III patterns may not be found in this way since we cannot overlay regions that do not have a well-defined structure. For example, the third intracellular loop of GPCRs has various lengths in different receptors and is believed to be involved in G protein coupling. Unfortunately, in the only crystal structure of class A GPCRs, bovine rhodopsin, the structure of that loop is not well defined and varies in different studies. Given the fact that we have limited other tools, if any, to find and interpret the biologically important residues in a protein region without a defined secondary structure, our PGOneIII algorithm may become particularly valuable.

4.3 Mining two datasets at the same time

We also adapted our pattern growth approach to mine two datasets at the same time to find patterns that distinguish members in the positive dataset from those in the negative one. Of course this can be done by comparing mining results from the two datasets. However, our pattern growth approach provides us with an efficient way to find the complete patterns that satisfy the requirement.

Users may get a better idea about what patterns may be associated with certain properties from mining two datasets. Subsequently, these patterns can be used for classification or to predict new members which share the same properties.

4.4 With MSA versus without MSA

Pattern discovery algorithms using aligned protein sequences have been developed for decades. However, ambiguity in an MSA would jeopardize such pattern analysis. In this study, we proposed a new algorithm to identify patterns directly from unaligned protein sequences. Thus we circumvented the potential problems caused by creating an MSA. From the results in this study, we argue that for a set of proteins with low sequence identity our approach is much better than the approaches based on MSA for several reasons. First, the computation time to build an MSA (hours or even days for a protein family as big as GPCRs) is saved. Second, for proteins with low sequence identity, MSA is not reliable, which will harm the pattern discovery. For example, some functionally important positions may escape detection because they are in or even close to gap regions. On the other hand, when protein sequences are highly homologous, the approaches based on MSA are better because our approach will slow down to search deep in the pattern tree and the computation is expensive and largely meaningless since millions of patterns may be generated from the highly homologous sequences. In addition, MSA becomes reliable for pattern discovery in such cases.

In short, MSA-based approaches may find type I and type II patterns, whereas our approach finds all three types of patterns.

4.5 Pattern growth versus TEIRESIAS

TEIRESIAS uses a very different way to efficiently mine type I patterns from one dataset. It first finds frequent short patterns and then glues them together to form a larger pattern. Similar to our approach, TEIRESIAS reports all patterns that satisfy the parameters entered by the user. Although the algorithms are different, PGOneI and TEIRESIAS behave very similar: both succeed in all the tests, the runtime is linear with the size
of the input and is affected by the support (or more precisely the number of patterns to find).

Although TEIRESIAS is efficient in mining type I patterns from one dataset, it is not able to find type II and type III patterns.

4.6 Pattern growth versus PRATT2

PRATT2 uses a different strategy to mine type I and II patterns from one dataset. It uses a portion of sequences to create a pattern graph, then tries to find (or optimize) patterns according to its scoring function. Creating and searching a pattern graph is expensive in terms of runtime and memory. That is why it runs slow and failed when the input contained 600 or more proteins in our tests. This failure in processing large datasets significantly reduces the application of PRATT2 in the life sciences, in view of the ever increasing amount of data.

PRATT2 seems to find ‘best’ rather than all patterns that satisfy the requirement because of the nature of its algorithm. Unfortunately, PRATT2 ranks the patterns by its own scoring function and outputs only the top 50 patterns by default. Even if the maximal number of patterns to be reported is set big enough, PRATT2 yields only part of all patterns. This policy looks friendly at first glance but it does not provide all solutions.

5 CONCLUSIONS

In this study, we designed and implemented six algorithms to mine three different pattern types from either one or two datasets using a pattern growth approach. We compared our approach to PRATT2 and TEIRESIAS in efficiency, completeness and the diversity of pattern types. Compared to PRATT2, our approach is faster, capable of processing large datasets and able to identify the type III patterns. Our approach is comparable to TEIRESIAS in discovery of type I patterns but has additional functionality such as mining type II and type III patterns and finding discriminating patterns from two datasets.

ACKNOWLEDGEMENTS

We acknowledge valuable discussion with Edgar de Graaf and Jeroen Kazius, and thank Margot Beukers for critical review and comments. This work was financially supported by an NWO-Bioinformatics Breakthrough Grant (050-71-041), The Netherlands.

Conflict of Interest: none declared.

REFERENCES

In Proceedings of the 20th International Conference on Very Large Data Bases (VLDB’94), pp. 487-499.
Attwood,T.K. et al. (2003) PRINTS and its automatic supplement, prePRINTS.
Protein Sci., 4, 1587–1595.
Omostiru,O. et al. (1996) An evolutionary trace method defines binding surfaces common to protein families.
Lupas,A.N. et al. (2001) On the evolution of protein folds: are similar motifs in different protein folds the result of convergence, insertion, or relics of an ancient peptide world?
Science, 286, 707–711.
Pei,J. et al. (2002) Mining sequential patterns with constraints in large databases.
In Proceedings of the 11th ACM International Conference on Information and Knowledge Management (CIKM’02), pp. 18–25.
Bioinformatics, 14, 55–67.
Protein Eng., 11, 1–9.
Bioinformatics, 19, 667–668.
Proteins, 63, 1018–1030.