A Novel Approach to Knowledge Discovery and Representation in Biological Databases

(Accepted version of Author's manuscript)

Jing Lu, Cuiqing Wang, Malcolm Keech

International Journal of Bioinformatics Research and Applications

http://www.inderscience.com/jhome.php?jcode=ijbra

A Novel Approach to Knowledge Discovery and Representation in Biological Databases

Jing Lu

Winchester Business School,
West Downs Campus,
University of Winchester,
Winchester, SO22 5HT, UK
Email: Jing.Lu@winchester.ac.uk

Cuiqing Wang

Faculty of Computer Science and Technology,
Shenyang University of Chemical Technology,
Shenyang, 110142, China
Email: Wangcuiqing@syuct.edu.cn

Malcolm Keech

ex-University of Bedfordshire,
Park Square, Luton, LU1 3JU, UK
Email: Malcolm.Keech@hotmail.com

Abstract: Extraction of motifs from biological sequences is among the frontier research issues in bioinformatics, with sequential patterns mining becoming one of the most important computational techniques in this area. A number of applications motivate the search for more structured patterns and concurrent protein motif mining is considered here. This paper builds on the concept of structural relation patterns and applies the Concurrent Sequential Patterns (ConSP) mining approach to biological databases. Specifically, an original method is presented using support vectors as the data structure for the extraction of novel patterns in protein sequences. Data modelling is pursued to represent the more interesting concurrent patterns visually. Experiments with real-world protein datasets from the UniProt and NCBI databases highlight the applicability of the ConSP methodology in protein data mining and modelling. The results show the potential for knowledge discovery in the field of protein structure identification. A pilot experiment extends the methodology to DNA sequences to indicate a future direction.

Keywords: Data analytics, bioinformatics, sequential patterns post-processing, structural relations, concurrent vector method, graphical modelling, biological databases, protein motif mining, DNA sequences, knowledge discovery

Reference to this paper should be made as follows: Lu, J., Wang, C.Q. and Keech, M. (201?) 'A novel approach to knowledge discovery and representation in biological databases', *Int. J. Bioinformatics Research and Applications, Vol.X, No.X, pp.X–Y.*

Biographical notes: Dr Jing Lu joined the University of Winchester in 2016 as a Senior Lecturer in Data Analytics. She is research-active in computer science and informatics with extensive experience of teaching in higher education, both as a Senior Lecturer/Associate Professor at Southampton Solent University and in China. Jing was awarded a PhD from the Department of Computing and Information Systems at the University of Bedfordshire in 2006 and became a Fellow of the HE Academy in 2009. Her research activities have extended across information systems and knowledge management through data mining and graph-based modelling to data analytics and knowledge discovery, including their application to real-world problems using database technologies. Jing's published research has featured in international journals covering business intelligence, data mining, modelling and management, and data warehousing as well as international conferences on artificial intelligence, data engineering, health and medical informatics, machine learning and data mining.

Cuiqing Wang is a Lecturer in Computer Science and Technology at the Shenyang University of Chemical Technology (SYUCT), China. She received a BSc in Computer Application (2000) from SYUCT and an MSc in Computer Science and Application (2006) from Northeastern University, China. Her research interests lie mainly in data mining and knowledge discovery.

Until recently Malcolm Keech was the Associate Dean of Creative Arts, Technologies & Science at the University of Bedfordshire, UK. Before joining the University in 1999, Malcolm had worked extensively in computing and IT development and management, both in the academic and industrial sectors. While his original academic background lies in mathematics (BA Oxford, MSc/PhD Manchester), his professional experience includes periods at London School of Economics, the Universities of London and Manchester, Florida State University, British Telecom and British Aerospace. He is a Fellow both of the Institute of Mathematics & its Applications and the Higher Education Academy.

1 Introduction

Pattern discovery has become a significant research area in bioinformatics, where the frequent occurrence of patterns in biosequences usually indicates that the sequences are biologically related and provide insight into biological process, cause of disease and evolution of life. One way of analysing the sequences is to group them into *families* with each family being a set of sequences which are believed to be related – evolutionarily, structurally or functionally [1]. A *motif* can be considered as a nucleotide or amino acid sequence pattern that is widespread and has a biological significance. Automatic extraction of motifs from biological sequences is therefore an important research problem in the study of molecular biology. Through the identification of protein sequence motifs

computationally, an unknown sequence can be classified into its predicted protein family for further biological analysis [2].

There are two main approaches for extracting knowledge from sequence data: one is to compare newly acquired data with already (manually) annotated data under the assumption that data similarity implies functional similarity. The second approach mines the data for frequently occurring patterns [3]. For the problem of mining patterns from biological datasets, the input is given as a set of related sequences and the goal is to find a set of patterns that are common to all or mostly all of the sequences in the dataset. Sequential patterns mining methods have been pursued in the literature to help determine the kinds of proteins that are likely to occur frequently in sequences [4]. The mining and modelling of frequent sequential patterns (such as motifs) that occur in many biosequences from a given database (i.e. DNA or protein sequences) could be essential to the interpretation and engineering of the biological data. The discovery of motifs in DNA and protein sequences is a much explored yet still developing research area [5,6,7,8,9].

With the successful development of efficient and scalable algorithms for mining frequent patterns from biological sequences, it is natural to extend the scope through post-processing and visualisation. There are various relationships among motifs and the study of this can lead to the discovery of significant but hidden patterns. Based on sequence databases, Structural Relation Patterns (SRP) mining aims to find and represent more complex information which can include concurrent patterns, exclusive patterns and iterative patterns [10]. All of these structures may have corresponding applications in motif identification and so the methodology was applied in the bioinformatics area [11], in particular for the analysis of protein sequences. The development of effective Concurrent Sequential Patterns (ConSP) mining techniques in the context of sequence motifs could help to determine those proteins that are likely to co-occur in target samples. Such data analytics and subsequent graphical modelling would facilitate the discovery of groups of proteins as well as the visualisation of interactions and relationships among them.

The remainder of this paper proceeds as follows: some related work is highlighted in section 2 to provide relevant background on sequential patterns mining from biological data as well as to establish the concept of concurrent sequential patterns in protein databases. In section 3, following a framework for protein data mining, the emphasis is on tailoring the ConSP mining approach to this domain – the Concurrent Vector method is presented with a worked example to illustrate its application. The corresponding approach to ConSP modelling and knowledge representation for protein data is described by the end of the same section. An experimental evaluation using well-known protein databases is given in section 4, which showcases the effectiveness of ConSP mining at generating new and interesting results as well as the potential of protein data modelling for visualisation. The work is then extended to DNA sequences through a pilot experiment, which indicates a future direction. The paper draws to a close by summarising and making brief conclusions.

2 Frequent Patterns Mining in Bioinformatics

To provide some background and further motivation, the nature of the biological data will be addressed first with the focus on mining patterns from protein sequences. This section will conclude with a discussion of post sequential patterns mining for structural features in this context.

2.1 Sequential Patterns Mining from Biological Databases

Biosequences are the primary sequences of DNA, RNA and protein molecules, and they represent the most basic type of biological information [12]. These sequences typically have a very small alphabet, i.e. 4 for DNA sequences and 20 for protein sequences, and many short patterns can occur in most sequences. The goal of pattern discovery in the latter domain is to find new and previously unknown patterns that are common to (or match) all or most of the sequences in the protein dataset.

A motif is an abstraction over a set of recurring patterns observed in a dataset and it captures the essential features shared by a set of similar or related objects. In the context of biological sequence analysis, a motif is a region or portion of protein or DNA/RNA sequences that has specific structure and significant function. Protein motifs have been divided into four categories: sequence motifs, sequence-structure motifs, structure motifs and structure-sequence motifs [13]. Sequence motifs are linear strings of amino acid residues with an implicit topological ordering and they are the most commonly encountered motif type in molecular biology. The discovery of sequence motifs can be formulated as the problem of finding short segments that occur frequently among a set of long protein sequences and therefore sequential patterns mining has become one of the important techniques in protein analysis.

Based on the traditional sequential patterns mining algorithms, Wang et al. proposed the SP-index algorithm to find the longest sequential patterns with gaps of arbitrary size [7]. The algorithm considers the characteristics of bioinformatics and contains two phases: the segment phase searches for frequent segments containing no gaps and generates base segments, which are then used to find the longest patterns in the pattern phase. The second phase can be time-consuming if there are many unnecessary segments in the first phase. To avoid the disadvantages of the SP-index algorithm, the Bit-Pattern-based (BP) approach has been proposed to mine sequential patterns in a protein database [5]. In this method, protein sequences are transformed into bit patterns first. Then by applying various bit operations (i.e. AND, OR, shifting and masking) on those bit patterns, the BP algorithm can find frequent segments and the longest sequential patterns. Biological datasets from the National Center for Biotechnology Information (http://www.ncbi.nlm.nih.gov) have been used to evaluate both the SP-index and BP approaches.

Motivated by the need to mine protein transmembrane helix features for protein sequence classification, the sequential patterns mining algorithm SPAM

has been modified to consider gaps and regular expression constraints [14]. The computational challenges for the motif extraction problem are two-fold: one is to design an efficient algorithm to enumerate the frequent motifs, and the other is to statistically validate the extracted motifs and report the significant ones [15]. EXMOTIF has been introduced to extract structured motifs within one or multiple biological sequences, which allow variable length gaps between simple motif components [9]. And a protein sequential patterns mining algorithm called BioPM uses a projection based on frequent prefixes to mine motifs, instead of considering all the possible sub-sequences [8].

Exarchos et al. presented a method based on protein sequence analysis and classification [6]. Specifically, sequential patterns mining was used for classification of proteins into folds highlighting the important role of data mining in bioinformatics. A group of protein sequences taken from the Protein Data Bank has been used to validate the proposed approach [16]. Gupta and Han reviewed applications of pattern discovery using sequential data mining in [4] which also covers predicting protein sequence functions, analysis of gene expression data, protein fold recognition and protein family detection.

Two of the key challenges in protein motif analysis are motif finding and representation [17]. Most of the above mining methods focus on the discovery of frequent sub-sequences; however, finding recurring *structural* features among protein sequences is important in bioinformatics. This motivates consideration of the authors' previous work on structural relation patterns mining and modelling [10].

2.2 Post-processing for Structural Relations in Protein Sequences

Structural Relation Patterns have been introduced to extend the search for complex patterns often hidden behind large sequences of data. In the context of frequent patterns mining, the focus of attention in the SRP family has been on mining concurrent sequential patterns, where the approach to graph-based modelling has proved to be illuminating.

Due to the characteristics of protein databases, where each protein is a linear sequence made up of smaller constituent molecules called amino acids, the following notation is required: let $\sum$ = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y} be a set of 20 items (e.g. amino acids). A sequence is an ordered list of items in this domain which can be written as $X = \langle x_1\ x_2\ \ldots\ x_p \rangle$, where $x_j \in \sum$ $(1 \le j \le p)$. X is contained in sequence $X' = \langle x_1'\ x_2'\ \ldots\ x_q' \rangle$ if $p \le q$ and $x_j = x_k'$ for all j, $1 \le j \le p$ and corresponding k, $1 \le k \le q$.

A protein sequence database (PSDB) is defined as a set $\{S_1, S_2, \ldots, S_n\}$, where each S_r $(1 \le r \le n)$ is a sequence of amino acid items. The *support* in PSDB of any given sequence S is the fraction/percentage of sequences in the database that contain S, i.e. $sup(S) = |\{S_r: S \subseteq S_r\}|/n$, where $|\ldots|$ denotes the number of sequences. Given a fraction *minsup* $(0 < minsup \le 1)$ as the minimum support threshold, *S* is called a *sequential pattern* in PSDB if $sup(S) \ge minsup$.

Example 1 Table 1 shows a sample protein sequence database PSDB = {<AHGAT>, <SASHTAT>, <HAHNAST>, <ANHATNT>} and the set of sequential patterns mined with a *minsup* of 50%. This will be used as a running worked example for illustration throughout the paper.

Table 1. Sequential patterns from a sample protein database

	Protein Seq	Sequential Patterns supported by each Sequence (SuppSP), *minsup*=50%
S_1	<AHGAT>	A,H,T,AA,AH,AT,HA,HT,AAT,AHA,AHT,HAT,**AHAT**
S_2	<SASHTAT>	A,H,T,S,AA,AH,AT,AS,HA,HT,TT,ST,AAT,AHA,AHT,ATT,**AST**,HAT,HTT,**AHAT**,AHTT
S_3	<HAHNAST>	A,H,T,S,N,AA,AH,AT,AS,AN,HA,HT,HN,ST,NA,NT,AAT,AHA,AHT,AHN,**AST**,ANA,ANT, HAT,HAN,HNT,NAT,**AHAT**,AHNT,ANAT,HANT
S_4	<ANHATNT>	A,H,T,N,AA,AH,AT,AN,HA,HT,HN,TT,NA,NT,AAT,AHA,AHT,AHN,ATT,ANA,ANT,HAT, HAN,HTT,HNT,NAT,**AHAT**, AHTT,AHNT,ANAT,HANT

Sequential patterns mining discovers a set of patterns from a given PSDB under a user-specified *minsup*. A sequential pattern is called a *maximal sequence* if it is not contained in any other sequential pattern. All patterns marked within the boxes in Table 1 are maximal sequences.

Most sequential patterns mining methods extract the complete set of sequential patterns and, in many cases, a large set of sequential patterns is neither intuitive nor necessarily very easy to understand or use. This motivates the search for ordering relationships which can better summarise the sequential patterns, leading to the advent of *concurrent* and *exclusive* patterns.

Let α, β be two of the sequential patterns mined from PSDB with minimum support threshold *minsup* and assume that α, β are not contained in each other. With regard to a particular data sequence $S \in$ PSDB, sequential patterns α and β have a *concurrent* relationship if and only if both of them have occurred in S, i.e. $(\alpha \angle S) \wedge (\beta \angle S)$ is true. This is represented by $[\alpha+\beta]_S$, where the notation '+' represents the concurrent relationship. The degree of concurrency can thus be defined below.

Definition 1 The *concurrence* of sequential patterns α and β is defined as the fraction of data sequences that contains both of the sequential patterns. This is denoted by $concurrence(\alpha,\beta) = |\{S_k:(\alpha \angle S_k) \wedge (\beta \angle S_k)\}|/n$, where $S_k \in$ PSDB, $1 \le k \le n$ and n is the total number of data sequences.

The user-specified minimum support threshold, *minsup* has been used as the frequency measurement for mining frequent itemsets and sequential patterns. Another fractional value, the minimum *concurrence* threshold, *mincon*

($0<mincon\leq minsup\leq 1$) is introduced below to check the concurrent relationships of sequential patterns.

Definition 2 Let *mincon* be the user-specified minimum concurrence. If concurrence(α,β)$\geq$*mincon* is satisfied, then α and β are called a *concurrent sequential patterns pair*. This is represented by ConSP = [$\alpha+\beta$], where there is no particular order, i.e. [$\alpha+\beta$] = [$\beta+\alpha$].

Consider PSDB = {<AHGAT>,<SASHTAT>,<HAHNAST>,<ANHATNT>} from Example 1 and assume a *mincon* of 50%. For the *maximal* sequential patterns pair AST and AHAT shown in bold in Table 1, according to Definition 1, *concurrence*(AST,AHAT) = 2/4 = 50%. Therefore, they constitute a concurrent sequential patterns pair given by ConSP = [AST+AHAT].

Extending the concurrent relationship to three sequential patterns α, β and γ motivates the notation $[\alpha+\beta+\gamma]_S$, equivalent to $(\alpha\angle S)\wedge(\beta\angle S)\wedge(\gamma\angle S)$. And, more generally, if $\{sp_1, sp_2, \ldots, sp_r\}$ is a set of r sequential patterns mined under *minsup*, then $[sp_1+sp_2+\ldots+sp_r]_S$ represents r sequential patterns which are concurrent with respect to data sequence S.

Furthermore, the concurrent sequential patterns represented by $ConSP_k = [a_1+a_2+\ldots+a_k]$ are contained in $ConSP_{k+m} = [b_1+b_2+\ldots+b_{k+m}]$ if $a_i\angle b_j$ for all i, $1\leq i\leq k$ and corresponding j, $1\leq j\leq(k+m)$. This is denoted by $ConSP_k\angle ConSP_{k+m}$. Concurrent sequential patterns are called *maximal ConSPs* if they are not contained in any other concurrent patterns.

Continuing with Example 1 and Table 1, data sequences S_3=<HAHNAST> and S_4=<ANHATNT> support all the sequential patterns AHAT, AHNT, ANAT and HANT, with *concurrence* (AHAT, AHNT, ANAT, HANT) = 2/4 = 50%. Therefore, they constitute another maximal concurrent pattern given by $ConSP_4$ = [AHAT+AHNT+ANAT+HANT].

Considering the same setting as above, sequential patterns α and β have an *exclusive* relationship if and only if one of them has occurred in S but not both, i.e. $((\alpha\angle S)\wedge\neg(\beta\angle S))\vee(\neg(\alpha\angle S)\wedge(\beta\angle S))$ is true. It is denoted by $[\alpha–\beta]_S$, where the notation '–' represents the exclusive relationship.

Definition 3 The *exclusion* of sequential patterns α, β is defined as the fraction of data sequences that contains just one of α or β, i.e. exclusion(α,β)= $|\{S_k:((\alpha\angle S_k)\wedge\neg(\beta\angle S_k))\vee(\neg(\alpha\angle S_k)\wedge(\beta\angle S_k))\}|/n$, where $S_k\in$PSDB and $1\leq k\leq n$.

Another fractional value, the minimum *exclusion* threshold ($0<minexc\leq minsup\leq 1$) is introduced below to check the exclusive relationships of sequential patterns.

Definition 4 Let *minexc* be the user-specified minimum exclusion. If *exclusion*(α,β)$\geq$*minexc* is satisfied, then α and β are called an *exclusive sequential patterns pair*, represented by ESP = [$\alpha–\beta$], where [$\alpha–\beta$] = [$\beta–\alpha$].

For the sequential patterns pair AH and AS shown double-underlined in Table 1, according to Definition 3 *exclusion*(AH,AS) = 2/4 = 50%, so they constitute exclusive sequential patterns.

Extending the exclusive relationship to three sequential patterns α, β and γ motivates the notation $[\alpha–\beta–\gamma]_S$, that is $((\alpha\angle S)\wedge\neg(\beta\angle S)\wedge\neg(\gamma\angle S))\vee(\neg(\alpha\angle S)\wedge(\beta\angle S)\wedge\neg(\gamma\angle S))\vee(\neg(\alpha\angle S)\wedge\neg(\beta\angle S)\wedge(\gamma\angle S))$. More generally, as before with concurrence, $[sp_1–sp_2–\ldots–sp_r]_S$ represents r sequential patterns which are exclusive with respect to data sequence S.

Another member of the structural relation patterns family, a sequential pattern *sp* is called an *iterative pattern* if it appears within the same data sequence at least n times ($n\geq 2$) and at most m times ($m\geq n$). The expression $<\{sp\}_n^m>$ denotes the iterative pattern, where m and n represent the upper and lower iteration bounds respectively of patterns appearing in a data sequence. If an iterative pattern has no upper iteration bound, then the parameter m is not required. Having introduced concurrent, exclusive and iterative patterns, the following unifies these patterns within an over-arching definition.

Definition 5 A *Structural Relation Pattern* (SRP) is a general designation of patterns consisting of sequential patterns, concurrent patterns, exclusive patterns, iterative patterns and their composition. That is, a concurrent pattern is an SRP. Similarly, an exclusive pattern is an SRP. Furthermore, the concurrent, exclusive or iterative combination of structural relation patterns constitutes new SRPs.

3 Concurrent Sequential Patterns Mining and Modelling

The original method to mine concurrent sequential patterns was proposed by using general sequence databases as the input, performing traditional sequential patterns mining and then post-processing the results. Protein data mining tailored to the context of bioinformatics was introduced in [11] and the corresponding Concurrent Vector method (ConVect) will be described here through its support vector/matrix-based ConSP mining approach, which uses the additional parameter *minlen* to specify the minimum length of sequential patterns sought. It aims to find potentially interesting ConSPs which are common to all (or mostly all) known protein sequences of a family. The ultimate focus is on visualising concurrent sequential patterns using graph-based modelling for knowledge representation.

3.1 Protein Data Mining Framework

Fig. 1 presents a framework for mining concurrent sequential patterns from protein sequences, which has four stages: Data Pre-processing, Sequential Patterns Mining, ConSP Mining and ConSP Modelling. First, depending on the nature of the biological database, pre-processing involves different types of task such as searching, retrieval and transformation. Second, frequently-occurring

patterns are sought through sequential patterns mining techniques which can be parameterised by *minsup* and *minlen*. The ConSP mining method then takes the results further by post-processing the sequential patterns under *mincon*≤*minsup*. Finally, the ConSP modelling stage aims to construct a graphical representation for the concurrent patterns.

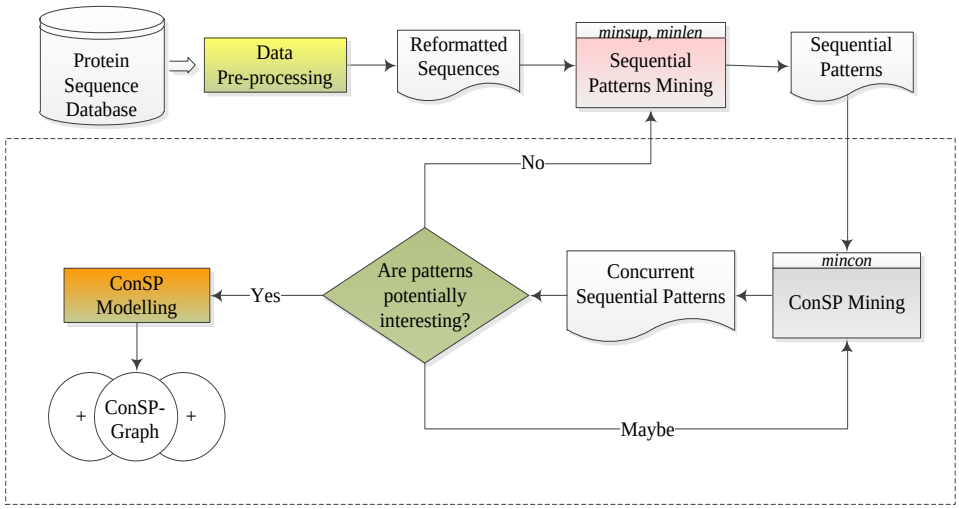

Fig. 1. Concurrent sequential patterns mining framework for protein databases.

Considering the boxed area in Fig. 1 from right to left, the initial cycle of concurrent pattern discovery is pursued through ConSP mining. The general approach here for choosing parameters is to apply a *mincon* threshold which is close to *minsup* while keeping *minlen* as long as possible, where the intention is to avoid a large number of sequential patterns too short for this context. If the patterns evaluated are perceived to be interesting, then ConSP modelling can proceed and the process completes for the parameters at this level; otherwise, another cycle commences based either on progressive ConSP mining or repeat sequential patterns mining. This corresponds respectively to incremental reduction of *mincon*, while the number of concurrent patterns remains countable, or a suitable change to the *minsup*/*minlen* combination.

3.2 ConVect Method and ConSP Generation

Given the sequence database PSDB = $\{S_1, S_2, \ldots, S_n\}$ and mined sequential patterns SP=$\{sp_1, sp_2, \ldots, sp_m\}$, a compact data structure or **support vector**, SeqVect(sp_i) is defined in [11] for every sequential pattern sp_i ($1 \le i \le m$) as:

$$\text{SeqVect}(sp_i) = [v_1 v_2 \ldots v_n],$$

where v_r=1 (1≤r≤n) if sequential pattern sp_i (1≤i≤m) is contained in data sequence S_r (1≤r≤n), i.e. $sp_i \angle S_r$; otherwise v_r=0. For example, for the PSDB = {<AHGAT>, <SASHTAT>, <HAHNAST>, <ANHATNT>} in Table 1 and sequential patterns AST and AHAT, SeqVect(AST)=[0110] and SeqVect(AHAT)=[1111].

Putting all of these vectors together forms columns of support vectors, SeqVect(SP) in two dimensions with sequential patterns sp_i (1≤i≤m) determining the binary entries v_{ri} corresponding to data sequences S_r (1≤r≤n) across the rows of a matrix. Considering the intersection of two support vectors for sp_i and sp_j (1≤i≤m, 1≤j≤m, i≠j) respectively, denoted by SeqVect(sp_i)∧SeqVect(sp_j) =$[v_1 v_2 ... v_n]$, v_r=1 (1≤r≤n) if both SeqVect(sp_i)=1 and SeqVect(sp_j)=1 in their r^{th} rows; otherwise v_r=0.

Counting the total number of entries in the resulting vector which have value "1" gives a measure of the support of both sequential patterns across PSDB, denoted by:

$$w=\text{Count}(\text{SeqVect}(sp_i)\wedge\text{SeqVect}(sp_j)).$$

If $w/n \geq mincon$, then sp_i and sp_j constitute a ConSP = [sp_i+sp_j].

The concept of a support vector and matrix can be extended therefore to a group of sequential patterns which will potentially make up a ConSP. The support vector of the concurrent sequential pattern $ConSP_s$ = [α_1+α_2+…+α_s] is defined as:

$$\text{ConVect}(\text{ConSP}_s) = [v_1 v_2 ... v_n],$$

where v_r=1 (1≤r≤n) if $ConSP_s \angle S_r$; otherwise v_r=0.

Similarly, putting all the support vectors for ConSPs together forms the columns of a **support matrix**, ConVect(ConSP) with respective concurrent sequential patterns determining the binary entries corresponding to data sequences across rows. If the intersection of ConSP = [sp_i+sp_j] with another sequential pattern sp_k (1≤k≤m, k≠i, k≠j) is considered, then:

$$\text{ConVect}(\text{ConSP})\wedge\text{SeqVect}(sp_k) = [v_1 v_2 ... v_n],$$

where v_r=1 (1≤r≤n) if both ConVect(ConSP)=1 and SeqVect(sp_k)=1 in their r^{th} rows; otherwise v_r=0.

Counting the total number of "1" entries in the resulting vector again gives:

$$w=\text{Count}(\text{ConVect}(\text{ConSP})\wedge\text{SeqVect}(sp_k)).$$

If $w/n \geq mincon$, then sp_k provides another concurrent sequential pattern which can be added to the existing result, such that ConSP = ConSP∪{sp_k} = [sp_i+sp_j+sp_k].

The Concurrent Vector method for mining ConSPs is divided into three phases in [11]. The approach will be described in detail here using the sample

protein database from Example 1 for illustration: PSDB = {<AHGAT>, <SASHTAT>, <HAHNAST>, <ANHATNT>}.

Phase 1 *Mining Sequential Patterns and Building Support Vectors*. First traditional sequential patterns mining discovers the frequently-occurring patterns from a given protein database under a user-specified *minsup*. For those with at least the specified minimum sequence length, *minlen*, the set of sub/super-sequences is determined for each pattern and the corresponding support matrix SeqVect(SP) formed across all patterns. From Table 1 there are 5 sequential patterns of minimum length 4 forming the columns and 4 protein data sequences forming rows, with results as shown in Table 2.

Table 2. A support matrix example

	AHAT	AHTT	AHNT	ANAT	HANT
S_1	1	0	0	0	0
S_2	1	1	0	0	0
S_3	1	0	1	1	1
S_4	1	1	1	1	1

Phase 2 *Concurrent Sequential Patterns Generation*. As follows:
A. Taking respective sequential patterns sp_i ($1 \le i \le m-1$) from SP in turn to provide the initial *seed-sequence*, the next sequential pattern sp_j ($i<j \le m$) is sought which can constitute an initial ConSP = $[sp_i+sp_j]$, so long as:

$$\text{Count}(\text{SeqVect}(sp_i) \wedge \text{SeqVect}(sp_j))/n \ge \textit{mincon}.$$

In this case, sp_i and sp_j can form a new seed-sequence for further ConSP checking.
B. Taking the next seed-sequence and corresponding ConSP as input, another sp_k is sought which contributes to forming an expanded ConSP$\cup\{sp_k\}$, if:

$$\text{Count}(\text{ConVect}(\text{ConSP}) \wedge \text{SeqVect}(sp_k))/n \ge \textit{mincon}.$$

In this eventuality, sp_k will be consolidated into the seed-sequence and so on.
C. Revert back to B to determine whether any further sequential patterns can be added to the intermediate ConSP in hand, until all combinations have been checked.

Example 2 Given SP={AHAT, AHTT, AHNT, ANAT, HANT} from Table 2.

1) Consider AHAT for the initial seed and take each of the next sequential patterns (i.e. AHTT, AHNT, ANAT and HANT) in turn to form the possible ConSPs.

1.1) in the case of AHTT:

Count(SeqVect(AHAT)∧SeqVect(AHTT))=
Count([1111]∧[0101])=Count([0101])=2;

i.e. concurrence(AHAT,AHTT)=2/4=50%≥*mincon*, so the first prospective ConSP_1=[AHAT+AHTT].

Using {AHAT, AHTT} as a seed-sequence and going through the rest of the sequential patterns in turn, i.e. AHNT, ANAT and HANT, none of these will satisfy the minimum concurrence condition. Therefore the ConSP result when taking AHAT as the initial seed remains the same, i.e. ConSP_1=[AHAT +AHTT].

1.2) in the case of AHNT:

Count(SeqVect(AHAT)∧SeqVect(AHNT))=
Count([1111]∧[0011])=Count([0011])=2;

i.e. concurrence(AHAT,AHNT)=2/4=50%≥*mincon*, so a second prospective ConSP_2=[AHAT+AHNT] can be used as a seed-sequence to go through the rest of the sequential patterns in turn, i.e. ANAT and HANT. For example, in the case of ANAT:

Count(ConVect(ConSP_2)∧SeqVect(ANAT))=
Count([0011]∧[0011])=Count([0011])=2.

Therefore, ANAT can be included to form an extended ConSP_2=[AHAT+AHNT+ANAT]. And considering the last sequential pattern HANT as a candidate for further potential concurrence gives:

Count(ConVect(ConSP_2)∧SeqVect(HANT))=
Count([0011]∧[0011])=Count([0011])=2.

So, the final ConSP_2=[AHAT+AHNT+ANAT+HANT] can be generated.

1.3) in the case of ANAT and HANT, by the same token the concurrent patterns ConSP_3=[AHAT+ANAT+HANT] and ConSP_4=[AHAT+HANT] can be deduced.

2) Following the above major iteration, taking AHTT as the next initial seed and considering each of the remaining sequential patterns to form an initial ConSP,

there are no other valid combinations to pair up concurrently under the given *mincon*.

3) Using AHNT next as the initial seed, the remaining sequential patterns are considered in turn and contribute to forming the fifth concurrent pattern ConSP_5=[AHNT+ANAT+HANT].

4) And when using the penultimate sequential pattern ANAT as the initial seed, an additional concurrent sequential pattern can be found as ConSP_6=[ANAT+HANT] and the process completes.

Therefore, in summary, the concurrent sequential patterns generated from this phase are:

ConSP_1=[AHAT+AHTT]
ConSP_2=[AHAT+AHNT+ANAT+HANT]
ConSP_3=[AHAT+ANAT+HANT]
ConSP_4=[AHAT+HANT]
ConSP_5=[AHNT+ANAT+HANT]
ConSP_6=[ANAT+HANT].

Phase 3 *Concurrent Sequential Patterns Optimisation*. Maximal ConSPs can be obtained by deleting those concurrent sequential patterns which are contained by other ConSPs, then deleting the sequential patterns in particular ConSPs (in turn) which are contained by other sequential patterns within the same ConSP.

For example, ConSP_3, ConSP_4, ConSP_5 and ConSP_6 from the above list are all contained within ConSP_2=[AHAT+AHNT+ANAT+HANT]. This means that the final maximal concurrent patterns are given by: ConSP_1=[AHAT+AHTT] and ConSP_2=[AHAT+AHNT+ANAT+HANT].

3.3 Graphical Representation

The use of graphical models in data mining that explore the inherent relationships among sequential patterns has already been presented for modelling concurrent sequential patterns [10]. It is adapted here to the bioinformatics domain to convey complex structural features, where every node of the graph represents an amino acid residue in a protein. Edges that connect pairs of residues indicate their adjacent relationship in the motifs.

The Concurrent Sequential Patterns Graph (ConSP-Graph) modelling approach is illustrated through a worked example: using ConSP-Graph construction to model the *highest level* $ConSP_4$=[AHAT+AHNT+ANAT +HANT] from Example 2.

Step 1. *Initialisation*. Determine the longest sequential patterns in $ConSP_4$ and in this example they are all the same length, therefore represent one of them by a directed graph G – e.g. AHAT – see Fig. 2(i).

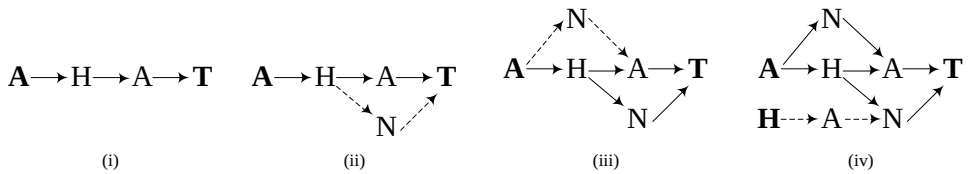

Fig. 2. Modelling of $ConSP_4$=[AHAT+AHNT+ANAT+HANT].

Step 2. Construction. For the next available sequential pattern, AHNT in $ConSP_4$, find any common prefix *preS* with G – this is AH; and find any common postfix *postS* – this is T. Taking out *preS* and *postS* from AHNT, the remaining part *elemS* = N can be represented by a directed graph G′. Add a directed edge from the last node of *preS* in G (i.e. H) to the first node of G′ (i.e. N). Also add a directed edge from the last node of G′ (i.e. N) to the first node of *postS* (i.e. T). The result of this step is the graph shown in Fig. 2(ii), where dotted lines represent new edges in the transitional model.

Step 3. Iteration. For the remaining sequential patterns in turn, i.e. ANAT and HANT, construct new graphs in a similar manner. Fig. 2(iii) shows the graph after adding sequential pattern ANAT while Fig. 2(iv) provides the ultimate representation of $ConSP_4$ from this method.

4 Experiments with Biological Databases

The empirical analysis of the Concurrent Vector method was performed on real-world datasets to test its effectiveness as well as to further illustrate the process of protein data mining. Data pre-processing precedes traditional sequential patterns mining in each case, whereupon ConSP mining is invoked using those SPs with a specified *minlen*. Results are summarised and a range of interesting graphs produced, which are arguably the most stimulating for the *higher level* ConSPs generated. A pilot experiment is then conducted to extend the methodology to DNA sequences.

4.1 Biological Data and Pre-processing

Biological databases can be broadly divided into sequence and structure databases. Sequence databases are applicable to both nucleic acid sequences and protein sequences, whereas structure databases are applicable to proteins only. Furthermore, protein sequence databases are classified as primary, composite and secondary depending on the content stored in them. Primary databases (e.g. SWISS-PROT and PIR) contain protein sequences as ‘raw’ data. Composite databases such as NCBI compile their sequence data from the primary sequence databases and filter them to retain only the non-redundant sequences. Secondary

databases (e.g. PROSITE and Pfam) contain the results of analysis from the primary sources, such as conserved and signature sequences.

The Universal Protein Resource (UniProt) is the world's most comprehensive catalogue of information on proteins. It is a central repository of protein sequence and function created by joining the information contained in SWISS-PROT, TrEMBL and PIR (UniProt Consortium, 2013). UniProtKB is the central hub for the collection of functional information on proteins with accurate, consistent and rich annotation. It consists of two sections: a section containing manually-annotated records with information extracted from the literature and curator-evaluated computational analysis (UniProtKB/Swiss-Prot), and a section with computationally-analysed records that await full manual annotation (UniProtKB/TrEMBL). Three datasets are extracted from this primary database for the experiments.

The National Center for Biotechnology Information accepts all submitted sequences and does not check whether the sequence is accurate or not – three datasets are extracted from this composite database for evaluation. Pfam is a secondary database of protein families, where families are sets of protein regions that share a significant degree of sequence similarity, thereby suggesting homology [18]. PROSITE consists of a large collection of biologically meaningful signatures described as patterns or profiles [19] which are common to all or nearly all of the members of the family – experiments using this secondary database are shown in [11].

It is necessary to transform the real-world biological sequences into a format suitable for sequential patterns mining (e.g. using PrefixSpan [20]). Table 3 shows a sample of protein sequences in the standard bioinformatics FASTA format, a text-based format for nucleotide sequences or peptide sequences, in which nucleotides or amino acids are represented using single-letter codes. A sequence in FASTA format begins with a single-line description, followed by lines of sequence data. The description line is distinguished from the sequence data by a greater-than (">") symbol in the first column. The word following the ">" symbol is the identifier of the sequence and the rest of the line is the description. The sequence ends if another line starting with a ">" appears; this indicates the start of another sequence. Pre-processing will eliminate the

Table 3. Sample protein sequences

>tr|T4ZID4|T4ZID4_PEPDI DNA polymerase III polC-type family protein (Fragment) OS=Peptoclostridium difficile CD127 GN=QEG_1158 PE=4 SV=1
MESIKEYLDKLEINNSGLGKQLKEVYINRVVYFKEDKIVYFYLTSKDIVSHELLDKFKEEL
>tr|T3HSB1|T3HSB1_PEPDC DNA polymerase III polC-type family protein (Fragment) OS=Peptoclostridium difficile (strain CD196) GN=QGC_1137 PE=4 SV=1
MESIKEYLDKLEINNSGLGKQLKEVYINRVVYFKEDKIVYFYLTSKDIVSHELLDKFKEELMYKLDYFK
>tr|G6RJ44|G6RJ44_STREE DNA polymerase III PolC-type domain protein OS=Streptococcus pneumoniae GA17971 GN=SPAR52_0014 PE=4 SV=1
MTRKEANKATALVGGIPEKGVTKHTNILVVGEQDWRVVGTDGLSSKMKKAQTLLEKGQDIEIMTENDFIRLLEE

description line and keep all the sequences as input for initial sequential patterns mining.

The Concurrent Vector method has been implemented in C++ and all experiments have been conducted on a 2.5GHz Intel Core i5 processor with 4GB main memory running under MS Windows 7.

4.2 Protein Sequences from Primary Databases

The first experiment concerns the SNAKE_TOXIN dataset from the UniProtKB database, a family of eukaryotic and viral DNA binding proteins consisting of cytotoxins, neurotoxins and venom peptides. This bio-dataset is not uncommon in sequential patterns mining experiments – it is dense and can generate many sequential patterns with a medium support threshold [21]. There are 400 reviewed sequences (i.e. manually annotated entries) with an average length of 76.

Table 4 shows the total number of SPs and ConSPs found on this dataset with various combinations of *minlen* and *mincon*=*minsup*. As *minsup* decreases, the number of patterns increases, sufficient that it becomes necessary to increase *minlen* to focus on a smaller number of more important patterns. It is worth noting that there is no ConSP when *minlen*=13 for *mincon*=100-95%.

Table 4. ConSP mining results summary for SNAKE_TOXIN dataset

mincon=	*minlen=9*		*minlen=10*		*minlen=11*		*minlen=12*	
minsup	∑SP	∑ConSP	∑SP	∑ConSP	∑SP	∑ConSP	∑SP	∑ConSP
100%	8	1	1	0	0	0	0	0
99%	106	16	29	7	4	1	0	0
98%	284	114	84	35	15	5	1	0
97%	661	704	198	106	39	20	4	0
96%	940	2623	357	467	87	57	10	9
95%	1396	8220	549	1416	173	207	35	9

Fig. 3 shows the graphical distribution of sequential patterns and ConSP mining results across the range of *minsup*, *mincon* and *minlen* values. There are too many patterns to be that useful in general as the thresholds decrease. This trend prevails even for *mincon*=*minsup*=90% – however, when *minlen*=13, there are 4 ConSPs found from the 18 sequential patterns with the highest level a single $ConSP_3$.

Fig. 4 shows the comparison of SP mining and ConSP mining in terms of the minimum length of sequential patterns across different *mincon*=*minsup* values. It is worth adding that, for example when *minlen*=11, the highest levels discovered are $ConSP_2$, $ConSP_4$, $ConSP_5$, $ConSP_{10}$ and $ConSP_{18}$ corresponding to incremental thresholds decreasing from 99% to 95%.

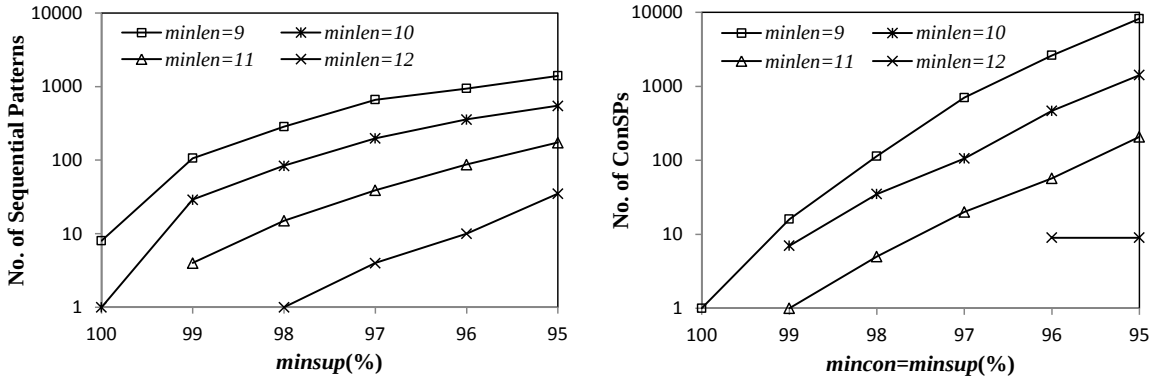

Fig. 3. Pattern distributions from mining SNAKE_TOXIN dataset.

As shown in Fig. 4(a)-(c), for each *minlen* setting, there is potentially a turning point where the total number of ConSPs exceeds the number of SPs. In the protein data mining domain, the purpose is not to find a large volume of patterns – it is more to discover those ConSPs which are common to all (i.e. *mincon*=100%) or nearly all (e.g. 90%≤*mincon*≤*minsup*<100%) of the members of the family – this broad strategy is applied in the rest of the experiments. Fig. 4(d) does not follow the trend shown from (a) to (c), i.e. under the setting *minlen*=12. However it is worth mentioning that, for the nine ConSPs when *mincon=minsup*=96%, the highest level is a $ConSP_3$; when *mincon=minsup* is

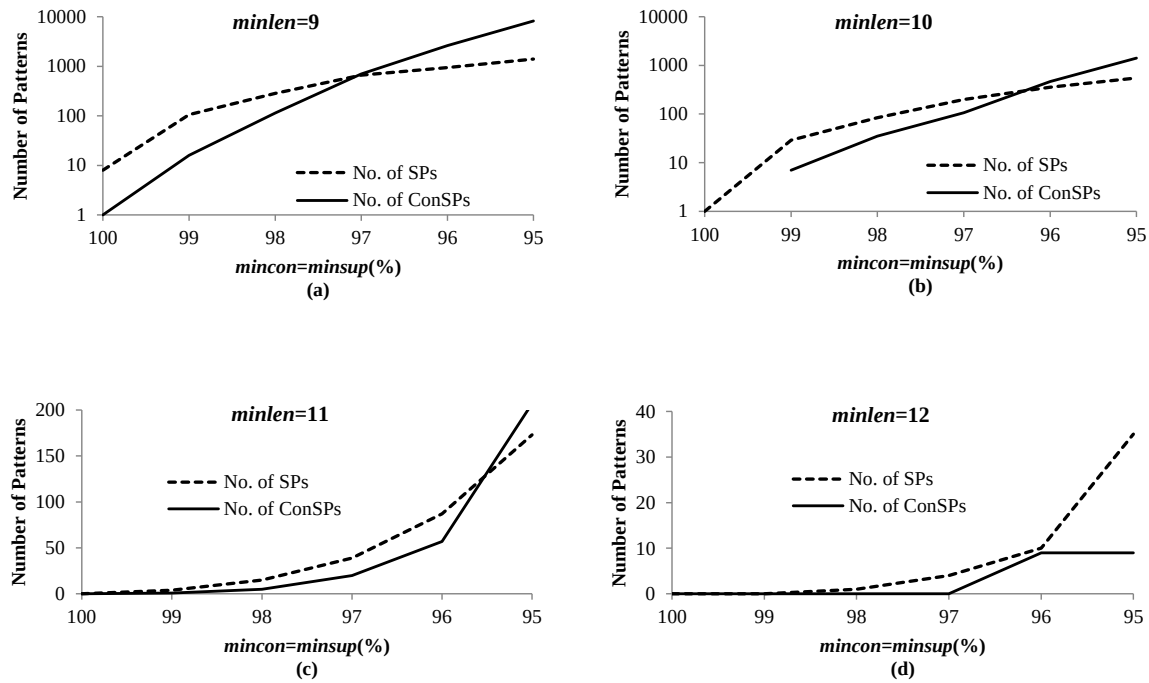

Fig. 4. Comparison between SPs and ConSPs from mining SNAKE_TOXIN dataset.

reduced to 95%, the highest level becomes a $ConSP_5$, reflecting the increased complexity of the nine concurrent patterns at this threshold.

The second dataset from UniProtKB is 'Elastase Pig' which contains 33 protein sequences with an average length of 274 – experimental results have been reported in Table 5. Due to the limited number of protein sequences here, some mining results are the same under the various settings. For example when *mincon=minsup* is between 100~98%, experiments have found two sequential patterns with *minlen*=4 and both of them are concurrent; when *mincon=minsup* is between 96~94%, one $ConSP_{19}$ is found from the 27 sequential patterns at the minimum sequence length of 7.

Table 5. ConSP mining results from 'Elastase Pig' dataset

mincon=minsup	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	4	2	1	1 $ConSP_2$
98%	4	2	1	1 $ConSP_2$
96%	7	27	1	1 $ConSP_{19}$
94%	7	27	1	1 $ConSP_{19}$
92%	10	6	1	1 $ConSP_6$
90%	11	7	2	1 $ConSP_4$

Reducing *mincon* and *minsup* down to 92%, while increasing *minlen* to 10, all 6 sequential patterns are concurrent to form a single $ConSP_6$. And, as *minsup* decreases to 90%, *minlen* can be increased further to yield a $ConSP_4$ at the highest level. A sample ConSP-Graph is illustrated in Fig. 5, where the concurrent protein sub-sequences in this family start either with amino acid "P" or "C" while converging on "P" at the end.

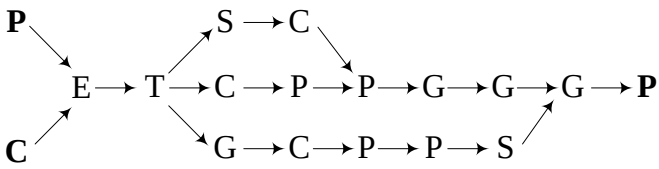

Fig. 5. An example of $ConSP_6$ modelling from 'Elastase Pig' (*mincon=minsup*=92%, *minlen*=10).

A third dataset is extracted from UniProtKB by using the condition Organism='Eubacterium', where there are 276 reviewed sequences with an average length of 290. Table 6 shows the total number of SPs and ConSPs found from this dataset with various combinations of *minsup*, *minlen* and *mincon*. Ostensibly, there are too many patterns to be that useful as the thresholds decrease to 95%.

Table 6. ConSP mining results summary for 'Eubacterium' dataset

minsup	*mincon*	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	100%	2	4	0	-
99%	99%	7	13	4	1 $ConSP_8$
98%	98%	8	425	22	1 $ConSP_{12}$
97%	97%	9	20	1	1 $ConSP_2$
96%	96%	9	274	355	1 $ConSP_7$
95%	95%	9	1293	4336	2 $ConSP_9$
95%	95%	10	9	0	-
	93%			19	1 $ConSP_4$
	91%			12	6 $ConSP_5$
	89%			1	1 $ConSP_9$

The shaded part of the table shows the comparison of SP mining and ConSP mining in terms of a minimum length of nine for sequential patterns across different *mincon*=*minsup* values. A turning point is first reached under *mincon*=*minsup*=96%, where the total number of ConSPs exceeds the number of SPs. Moreover, when the threshold is taken down to 95%, the number of patterns begins to increase exponentially when *minlen*=9.

Setting *minlen*=10 there is no ConSP at all when *mincon*=*minsup*=95%; however, concurrent patterns can be discovered when *mincon* is decreased, as can be seen in the lower section of Table 6. Furthermore, when *mincon*=89%, all 9 sequential patterns are concurrent and the single $ConSP_9$ can be modelled as in Fig. 6. It shows that the protein sub-sequences in this family share a common amino acid "M" at the start and all bar one sequence finish up with "K".

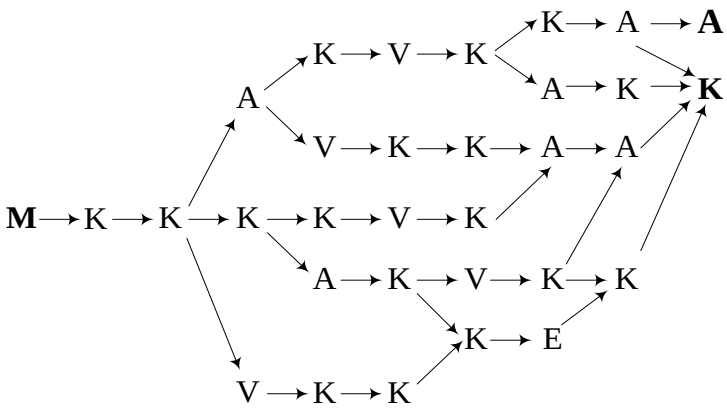

Fig. 6. An example of $ConSP_9$ modelling from 'Eubacterium' (*minsup*=95%, *mincon*=89%, *minlen*=10).

It is demonstrated here that appropriate increasing of *minlen* in conjunction with incremental reduction of *mincon*<*minsup* can connect all of the sequential patterns structurally at these relatively high threshold levels.

4.3 Protein Sequences from Composite Datasets

The first experiment from the NCBI Protein Database is initiated by searching for Protein Name='Integrin', where this dataset contains 726 protein sequences with an average length of 522. Integrins are transmembrane receptors that are the bridges for cell-cell and cell-extracellular matrix interactions – the experiments are reported in Table 7.

Table 7. ConSP mining results from 'Integrin' dataset

mincon=minsup	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	4	1	0	-
99%	6	173	108	1 $ConSP_{12}$
98%	8	244	140	1 $ConSP_{10}$
97%	10	379	377	1 $ConSP_{10}$
96%	11	344	282	1 $ConSP_{10}$
95%	12	720	380	1 $ConSP_{14}$

The experiments have been conducted based on decreasing *minsup* while optimising the minimum length of sequential patterns, *minlen*. Initially, when *minsup*=100% and *minlen*=4, there is one sequential pattern only. Setting *mincon=minsup*=99% and increasing *minlen* to 6, there are 173 sequential patterns and 108 ConSPs. Reducing *mincon* and *minsup* incrementally down to 95%, when *minlen*=12 there are 720 sequential patterns and 380 ConSPs, with the highest level a $ConSP_{14}$ – this is modelled by the ConSP-Graph in Fig. 7. It shows that the protein sub-sequences in this family share a common amino acid "N" at the start and all sequences finish up with "L".

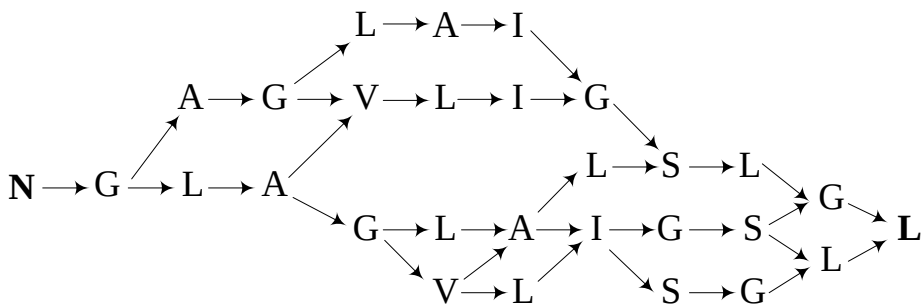

Fig. 7. An example of $ConSP_{14}$ modelling from 'Integrin' (*mincon=minsup*=95%, *minlen*=12).

Datasets can also be extracted from NCBI by using the conjunction of several conditions, e.g. (1) Search Databases='Protein', then (2) Organism='Human' with (3) sequence length ranging from 100 to 200. Table 8 shows the total number of sequences in the dataset under different protein length settings.

Table 8. Protein datasets under different settings

Dataset Name	Organism='Human'						
Protein Sequence Length	100	120	140	150	160	180	200
Total No. of Sequences	3850	2476	1101	678	848	762	728

Choosing the sequence length to be 150 here, then there are 678 protein sequences. The experiments conducted on this particular dataset are reported in Table 9 across a range of *minsup*/*minlen* values and various *mincon*.

Table 9. ConSP mining results from 'Human' dataset with sequence length=150

minsup	*mincon*	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	100%	4	9	1	1 $ConSP_9$
99%	99%	6	34	24	2 $ConSP_4$
98%	98%	7	37	15	15 $ConSP_2$
97%	97%	8	4	0	-
	96%			2	2 $ConSP_2$
96%	96%	8	44	12	12 $ConSP_2$
95%	95%	8	146	233	7 $ConSP_4$
	94%			1711	3 $ConSP_7$
	92%			8690	1 $ConSP_{13}$
	90%			9150	1 $ConSP_{14}$

For example, experiments with *mincon*=*minsup*=100% have found nine sequential patterns when *minlen*=4, therefore all of them make up the single $ConSP_9$. Reducing *minsup* and naturally increasing *minlen*, e.g. for *mincon*=*minsup*=99% and *minlen*=6, there are 34 sequential patterns and 24 ConSPs. Taking *minsup* down further to 95% and retaining *minlen*=8, the extent of patterns discovered increases and the turning point is reached for ConSPs. When *mincon* is now reduced to 94%, 1711 concurrent patterns have been found – while this is rather too large a number, one of the three highest level $ConSP_7$ at this parameter setting has been selected as representative and shown in Fig. 8.

Thereafter the number of ConSPs increases significantly and modelling is no longer pursued.

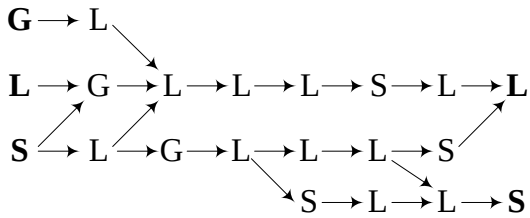

Fig. 8. An example of $ConSP_7$ modelling from 'Human' with sequence length of 150 (*minsup*=95%, *mincon*=94%, *minlen*=8).

The final dataset is obtained from NCBI using the search condition Protein Name='Cold Shock' and selecting the DNA Binding Domain Protein from the list. This dataset contains 3505 sequences with an average length of 102. Table 10 shows that there are no sequential patterns found under *mincon*=*minsup*=100%. Reducing *minsup* and increasing *minlen*, e.g. for *mincon*=*minsup*=99% and *minlen*=9, there are 99 sequential patterns and 104 ConSPs. Continuing the now customary reduction of thresholds leads to the shaded area in Table 10: there are 14 sequential patterns of length 15 and seven concurrent patterns can be discovered when *mincon*=*minsup*=94%, with the highest level a $ConSP_4$.

Table 10. ConSP mining results from 'Cold_Shock' dataset

minsup	*mincon*	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	100%	1	0	0	-
99%	99%	9	99	104	2 $ConSP_{10}$
98%	98%	11	174	238	2 $ConSP_6$
97%	97%	13	6	1	1 $ConSP_3$
96%	96%	14	7	1	1 $ConSP_3$
95%	95%	14	122	203	1 $ConSP_{14}$
95%	95%	15	2	0	-
	94%			1	1 $ConSP_2$
94%	94%	15	14	7	1 $ConSP_4$
	92%			30	6 $ConSP_7$
	90%			9	1 $ConSP_{12}$
	88%			1	1 $ConSP_{14}$

Reducing *mincon* incrementally down to 88%, all of the sequential patterns are then concurrent to form the $ConSP_{14}$. This is modelled by the ConSP-Graph in Fig. 9.

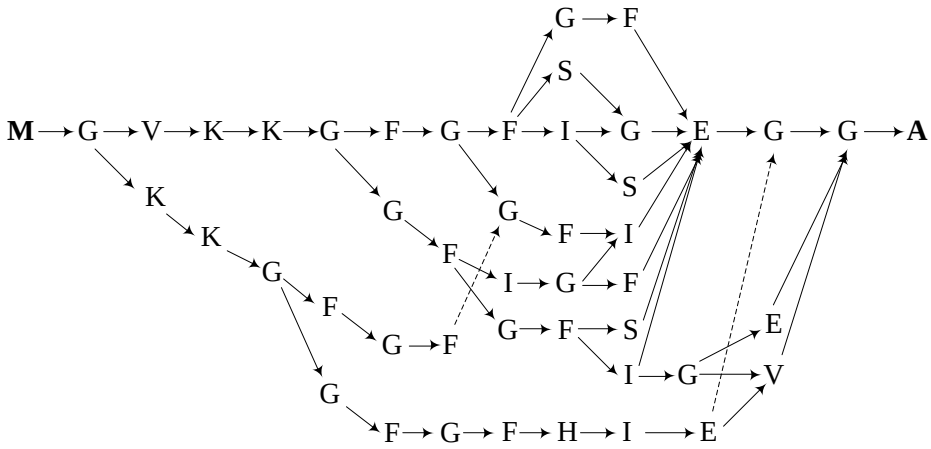

Fig. 9. An example of $ConSP_{14}$ modelling from 'Cold_Shock' (*minsup*=94%, *mincon*=88%, *minlen*=15).

It can be seen here that complex relationships are represented at relatively high threshold levels which could lead to protein structure identification of some biological significance.

4.4 Extension to DNA Sequences

It is also possible to consider using the ConSP methodology for discovery of motifs from DNA sequences and, on the theoretical side, the same framework could be applied. However, in the context of DNA, the average length of sequences is generally much greater than proteins with only four possible items A, C, G and T – representing the four nucleotide bases of a DNA strand. An example is given below before a pilot experiment is presented based on a breast cancer dataset.

Example 3 Given a small DNA sequence database DSDB={<GAGGAGA>, <AGATATGCTTAGAG>,<ACTGAGGTAGA>,<ATTGAGCTT>}. When *mincon=minsup*=100%, there are 11 sequential patterns but none of them are concurrent; decreasing to *mincon=minsup*=75%, there are 87 sequential patterns and two concurrent patterns, e.g. $ConSP_6$ = [ACTT+AGAGT+ATAGT+ATGAG +ATTAG+ATTGA]. Decreasing further to *mincon=minsup*=50%, there are three ConSPs from the 411 sequential patterns. For instance, $ConSP_7$=[ACTAGAG+ ACTTAGA+AGAGGAG+AGAGTAGA+ATAGGAG+ATAGTAGA+ATGAG AG].

The dataset used for the experiment comprises real-life DNA sequences extracted by using the advanced search technique provided by NCBI: (1) Search Databases='Nucleotide', (2) Organism='Human', (3) All Fields='Breast Cancer' with (4) sequence lengths from 60 to 70. There are 1156 sequences for this setting – a selection of results is reported in Table 11.

Table 11. ConSP mining results from 'Breast Cancer' DNA dataset with sequence lengths=60~70

minsup	*mincon*	*minlen*	∑SP	∑ConSP	Highest Level ConSP
100%	100%	5	9	1	1 $ConSP_9$
99%	99%	9	34	5	5 $ConSP_2$
98%	98%	10	54	6	6 $ConSP_2$
97%	97%	10	1162	2901	5 $ConSP_6$
97%	97%	11	4	0	-
96%	96%	11	182	22	22 $ConSP_2$
95%	95%	11	1398	1510	2 $ConSP_5$
95%	95%	12	0	0	-
94%	94%	12	33	0	-
	93%			10	10 $ConSP_2$
	92%			122	13 $ConSP_3$
	91%			375	3 $ConSP_5$
	90%			438	3 $ConSP_6$
92%	92%	12	1901	641	25 $ConSP_3$
90%	90%	13	84	0	-

Compared with the experiments for protein sequences, overall the highest level of ConSP found in DNA sequences appears relatively low. In Table 11, apart from when *mincon*=*minsup*=100% where all 9 sequential patterns make up one $ConSP_9$, most of the other results are at the $ConSP_2$ and $ConSP_3$ level. $ConSP_5$ and $ConSP_6$ can be found, but only when the total number of ConSPs is large, e.g. for *mincon*=*minsup*=97% (*minlen*=10) and *mincon*=*minsup*=95% (*minlen*=11).

However, when *mincon*=*minsup*=94% and *minlen*=12, there are 33 sequential patterns and no ConSPs. Incremental reduction of *mincon*<*minsup* generates more promising results, e.g. when *mincon*=90% there are 438 ConSPs, where the three highest level $ConSP_6$ have been modelled in Fig. 10.

There are only four nucleotide bases for DNA sequences and this causes more repetition of the items compared with ConSP mining in proteins. It is worth noting that only three are present here, with the "C" not featuring in the concurrent examples above. Nonetheless the ConSP-Graphs are interesting to an

extent and there is potential for further investigation of the Concurrent Vector method applied to DNA.

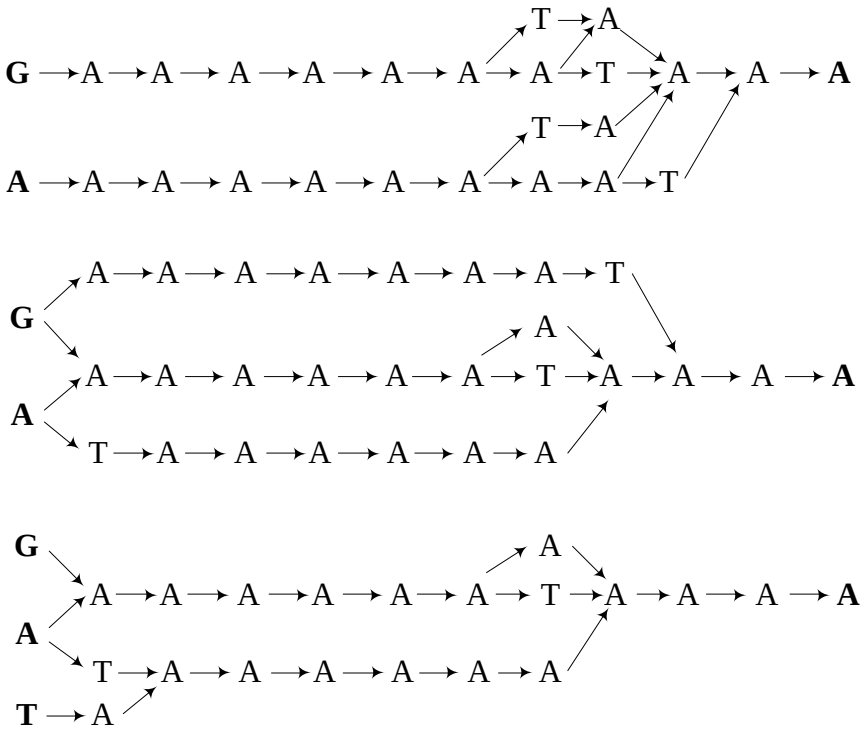

Fig. 10. Three examples of $ConSP_6$ modelling from 'Breast Cancer' DNA (*minsup*=94%, *mincon*=90%, *minlen*=12).

5 Conclusions

A novel protein mining framework has been proposed which incorporates traditional sequential patterns mining parameterised by *minsup* and *minlen* before checking for concurrent relationships within the *mincon* threshold. The Concurrent Vector method has been illustrated through worked examples and put through its paces with experiments which aim to discover sub-sequences common to all or nearly all of the proteins. This brings a need not only to operate at high percentage thresholds for *minsup*, but also to optimise *minlen* so that the number of patterns remains countable. Moreover, there is a corresponding requirement for *mincon* to be close enough to *minsup* while adding value to knowledge representation.

The experimental results from primary and composite biological databases demonstrate the feasibility and effectiveness of the data mining and modelling techniques, which identify connectivity and integration across real protein sub-sequences. In particular, higher level ConSPs can be discovered for suitable

minsup/*minlen* combinations by tuning *mincon*<*minsup* to generate a smaller number of more structured patterns. The visualisation of concurrent protein motifs may contribute towards further understanding of the inherent meaning attached to such biological data.

The methodology can be extended to DNA sequences and a pilot experiment has been conducted to illuminate the way forward. While there is more work to be done here, the applicability of the ConSP mining approach has been revealed and will be the subject of future research.

References

[1] A. Brazma, I. Jonassen, I. Eidhammer and D. Gilbert, Approaches to the automatic discovery of patterns in biosequences, *Journal of Computational Biology* **5** (1998), 279-305.

[2] C.M. Hsu, C.Y. Chen and B.J. Liu, WildSpan: mining structured motifs from protein sequences, *Algorithms for Molecular Biology* **6**(1) (2011), 1-16.

[3] S. Angelov and S. Inenaga, Composite pattern discovery for PCR application, string processing and information retrieval. In: *12th International Conference on String Processing and Information Retrieval*, SPIRE 2005, LNCS 3772, Springer, 2005, pp. 167-178.

[4] P. Kumar, P.R. Krishna and S.B. Raju, Pattern discovery using sequence data mining: applications and studies, IGI Global, Hershey, Pennsylvania, 2012.

[5] Y.I. Chang, C.C. Wu, J.R. Chen and Y.H. Jeng, Mining sequence motifs from protein databases based on a bit pattern approach, *International Journal of Innovative Computing, Information and Control* **8**(1) (2012), 647-657.

[6] T.P. Exarchos, C. Papaloukas, C. Lampros, and D.I. Fotiadis, Mining sequential patterns for protein fold recognition, *Journal of Biomedical Informatics* **41**(1) (2008), 165-179.

[7] K. Wang, Y. Xu and J.X. Yu, Scalable sequential pattern mining for biological sequences. In: *13th International Conference on Information and Knowledge Management*, ACM, 2004, pp. 178-187.

[8] Y. Xiong and Y.Y. Zhu, BioPM: an efficient algorithm for protein motif mining. In: *Bioinformatics and Biomedical Engineering*, Wuhan, China, IEEE, 2007, pp. 394-397.

[9] Y. Zhang and M.J. Zaki, Exmotif: efficient structured motif extraction, *Algorithms for Molecular Biology* **1** (2006), 21-39.

[10] J. Lu, M. Keech, W.R. Chen and C.Q. Wang, Concurrent sequential patterns mining and frequent partial orders modelling, *International Journal of Business Intelligence and Data Mining* **8**(2) (2013), 132-154.

[11] C.Q. Wang, J. Lu and M. Keech, Applications of concurrent sequential patterns in protein data mining. In: *10th International Conference on Machine Learning and Data Mining*, MLDM 2014, LNAI 8556, Springer International Publishing Switzerland, 2014, pp. 243-257.

[12] I. Jonassen, J.F. Collins and D.G. Higgins, Finding flexible patterns in unaligned protein sequences, *Protein Science* **4**(8) (1995), 1587-1595.

[13] D. Conklin, S. Fortier and J. Glasgo, Representation for discovery of protein motifs. In: *1st International Conference on Intelligent Systems for Molecular Biology*, Menlo Park, California, USA, 1993, AAAI/MIT Press, pp. 101-108.

[14] J. Ho, L. Lukov and S. Chawla, Sequential pattern mining with constraints on large protein databases. In: *12th International Conference on Management of Data*, 2005, pp. 89-100.
[15] G. Bruno and P. Garza, Temporal pattern mining for medical applications, *Data Mining: Foundations and Intelligent Paradigms* **25**(1) (2012), 9-18.
[16] H.M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T.N. Bhat and H. Weissig, The protein data bank, *Nucleic Acids Research* **28**(1) (2000), 235-242.
[17] G. Z. Dong and J. Pei, Sequence Data Mining. Advances in Database Systems, Springer US, 2010.
[18] M. Punta, P.C. Coggill, R.Y. Eberhardt, J. Mistry, J. Tate, C. Boursnell, N. Pang, K. Forslund and G. Ceric, The Pfam protein families database, *Nucleic Acids Research* **40** (D1) (2011), 290-301.
[19] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P.S. Langendijk-Genevaux, M. Pagni, and C.J. Sigrist, The PROSITE database. *Nucleic Acids Research* **34**(1) (2006), 227-230.
[20] PrefixSpan source code, http://www.pudn.com/downloads39/sourcecode/math/detail134610_en.html.
[21] J. Wang and J. Han, Efficient mining of frequent closed sequences. In: *20th International Conference on Data Engineering*, IEEE, 2004, pp. 79-90.