Mining Sequential Pattern By Prefix-Projected Pattern Growth For Protein Sequences

Mya Kyar Phyu, Khaing Moe San
Computer University (Lashio)
myakyarphyu87@gmail.com

Abstract

Data mining involves the use of sophisticated data analysis tools to discover previously unknown, valid patterns and relationships in large data sets. Sequential pattern mining is to find all frequent sequential pattern with a user specified minimum support threshold. PrefixSpan is a pattern growth method, particularly popular in biomedical fields. PrefixSpan is the most promosing of pattern growth method and is based on recursively constructing the pattern and the search to projected database. At each step, algorithm looks for the frequent sequences with prefix α, in the corresponding projected database. In prefixspan algorithm, no candidate sequence needs to be generated. The search space is reduced at each step allowing for better performance, in the presence of small support threshold. In this paper, the patterns of protein sequences can be discovered to analyse the structure of amino acids which are building blocks of protein sequences by using prefixspan approach.

1. Introduction

Bioinformatics is a promising young field that applies computer technology in molecular biology and develops algorithm and method to manage and analyze biological data. DNA and protein sequences are sequential biological data exists in huge volumes as well, it is important to develop effective method to compare and discover biosequence patterns [3]. PrefixSpan mines the complete set of pattern without candidate subsequence generation.PrefixSpan algorithm are relevant some sequential nature, like events in the case of temporal information, or amino-acid sequences for problems in bioinformatics. Frequent pattern analysis has been a focused of study in data mining, and a lot of algorithms and methods have been developed for mining frequent patterns, sequential patterns and structural patterns. DNA or protein sequential patterns usually allow a nontrivial number of insertion, deletion, and mutations. Frequent sequential pattern discovery has been an active research area for years. Finding sequential patterns, many biological data analysis tasks need to find frequent structured patterns, such as frequent protein or chemical compound structures from the data. In this paper, we systematically explore a pattern-growth approach for efficient mining of sequential patterns in large sequence database. Sequence databases are recursively projected into a set of smaller projected databases based on the current sequential pattern(s), and sequential patterns are grown in each projected databases by exploring only locally frequent fragements. The sequential pattern mining problem was first introduced by Agrawal and Srikant in [1]. Dhany Saputra, Dayang R. A. Rambli and Oi Mean Foong [2] proposed an improvement of pattern growth-based PrefixSpan algorithm, called I-PrefixSpan. The general idea of I-PrefixSpan is to use sufficient data structure for Seq-Tree framework and separator database to reduce the execution time and memory usage.

2. Sequential Pattern Mining: concepts and primitives

Sequential pattern mining is the mining of frequently occurring ordered events or subsequences as patterns. Sequential patterns are useful for shelf placement and promotions. This industry, as well as telecommuni-cations and other businesses, may also used sequential patterns for targeted marketing, customer retention, and many other tasks [5]. Other areas in which sequential patterns can be applied include Web access pattern analysis, weather prediction, production processes, DNA sequence, protein sequence and network intrusion detections [6]. The sequential pattern mining problem was first introduced as follows: “Given a set of sequences, where each sequence consists of a list of events or elements) and each event consists of a set of items, and given a user-specified minimum support threshold of min-sup, sequential pattern mining finds all frequent subsequences, that is, the subsequences whose occurrence frequency in the set of sequences is no less than min-sup."

3. Pattern growth based approaches

Pattern-growth method is used for mining sequential patterns efficiently. The new methods are non-Apriori and apply a divide-and-conquer, pattern-growth principle. Pattern Growth method consists of two approaches: FreeSpan (for frequent pattern-projected Sequential pattern mining) and PrefixSpan (for Prefix-projected Sequential pattern mining). They mine the complete set of sequential patterns but greatly reduce the efforts of candidate subsequence generation.

The general idea is that sequence databases are recursively projected into a set of smaller projected databases and sequential patterns are grown in each projected databases by exploring only local frequent fragments. Pattern growth is a method of frequent-pattern, mining that does not require candidate generation. Instead of repeatedly scanning the entire database and generating and testing large sets of candidate sequences, divide and conquer strategy: recursively project a sequence database into a set of smaller databases and then mine each projected database to find frequent patterns. The algorithm builds prefix patterns, which it concatenates with suffix patterns to find frequent patterns, avoiding candidate generation. Pattern Growth method is a more recent approach to deal with sequential pattern mining problems. Pattern-growth methods try to grow longer patterns from shorter ones. Accordingly, they divide the search space and focus only on the subspace potentially supporting further pattern growth at a time. Thus, their search spaces are focused and are confined by projected databases. A projected database for a sequential pattern α contains all and only the necessary information for mining sequential patterns that can be grown from α. As mining proceeds to long sequential patterns, projected databases become smaller and smaller.

3.1 Prefix-projected sequential pattern mining

PrefixSpan is a recently proposed method for mining frequent sequential patterns. PrefixSpan use prefix projection to mine the complete set of frequent sequential patterns. PrefixSpan is a pattern growth method, particularly popular in biomedical applications. For more detail, information are described how prefixspan works. Given sequence database S, and minimum support of ξ . PrefixSpan performs the following step to mine the frequent sequential patterns.

Step 1: Scan S once, and find all the frequent items. These frequent items are frequent sequential patterns with length one.

Step 2: According to the length-1 frequent sequence found in the first step, the complete set of frequent sequential patterns can be divided into different subsets, one subset for each length-1 frequent sequence.

Step 3: Each subset, in second step, can be mined by constructing its corresponding postfix projected database, and mining it recursively projected database.

PrefixSpan use recursion to mine the frequent sequences. First frequent items in the database are found and then for each frequent item, a projected database is created. This process is repeated for each projected database, until the projected database contains no frequent items. By using prefixspan algorithm, example of finding sequential patterns in a sequence database is shown in following Table[1].

Table 1. A sequence database

Sequence-id	Sequence
1	<a(abc)(ac)d(cf)>
2	<(ad)c(bc)(ae)>
3	<(ef)(ab)(df)cb>
4	<eg(af)cbc>

A sequence <a(abc)(ac)d(cf)> has five elements(a),(abc),(ac),(d) and (cf), where items a and c appear more than once respectively in different elements. It is also a 9 sequences since there are 9 instances appearing in that sequences. Item a happens three times in this sequences,so it contributes 3 to the length of the sequence. However, the whole sequence<a(abc)(ac)d(cf)> contributes only one to the support of <a>. Also, sequence <a(bc)df> is a subsequence of <a(abc)(ac)d(cf)>. Since both sequences 10 and 30 contain subsequence s=<(ab)c>,s is a sequential pattern of length 3(i.e., 3-pattern). For the sequence database S in Table 1 with min-sup 2, sequential patterns in S can be mined by a prefix-projection method in the following Steps.

Step 1: Find length-1 sequential patterns. Scan S once to find all frequent items in sequences. Each of theses

frequent items is a length-1 sequential pattern. They are <a>:4,<b>:4,<c>:4,<d>:3,<e>:3 and <f>:3, where <pattern>:count represents the pattern and its associated support count.
Step 2: Divide search space. The complete set of sequential patterns can be partitioned into the following six subsets according to the six prefixes(1) the ones having prefix <a>;….; and (6) the ones having prefix<f>.
Step 3: Find subsets of sequential patterns. The subsets of sequential patterns can be mined by constructing the corresponding set of projected databases and mining each recursively. The projected databases as well as sequential patterns found in them while thwe mining process is explained as follows:
(a). Find sequential patterns with prefix <a>. Only the sequences containing <a> should be collected. Moreover,in a sequence containing<a>, only the subsequence prefixed with the first occurrence of <a> should be considered. For example, in sequence<(ef)(ab)(df)cb>, only the subsequence <(-b)(df)cb> should be considered for mining sequential patterns prefixed with <a>.Notice that (-b) means that the last element in the prefix, which is a, together with b, from one element. The sequences in S containing <a> are projected with regards to <a> to form the <a>projected database, which consists of four suffix sequences:<(abc)(ac)d(cf)>,<(d)c(bc)(ae)>,<(-b)(df)cb>, and <(f)cbc>. By scanning the <a> projected database once, its locally frequent items are a:2,b:4,b:2,c:4,d:2, and f:2. Thus, all the length-2 sequential patterns prefixed with <a> are found, and they are:<aa>:2,<ab>:4,<(ab)>:2,<ac>:4,<ad>:2 , and <af>:2. Recursively, all sequential patterns with prefix <a> can be partitioned into six subsets:1) those prefixed with <aa>,2) those with <ab>,…., and, finally,3) those with <af>. These subsets can be mined by constructing respective projected database and mining each recursively as follows. Similarily, we can find sequential patterns having prefix <b>,<c>,<d>,<e> and <f>, respectively, by constructing <b>-,<c>-,<d>-,<e>-, and <f>- projected databases and mining them respectively. The set of sequential patterns is the collection of patterns found in the above recursive mining process [4].

3.2 PrefixSpan algorithm

PrefixSpan is the most promosing of pattern growth methods and is based on recursively constructing the pattern, and the search to projected databases. An α-projected database is the set of subsequence in the database. At each step, algorithm looks for the frequent sequences with prefix α ,in the corresponding projected database.

```
    Input: A sequence database S, and the minimum support threshold minsup.
    Output:      The complete set of sequential patterns
    Method:      Call PrefixSpan(<>,0,S)
Subroutine: PrefixSpan (α , l, S |α)
    Parameters: α:a sequential pattern; l: the length of α ; S|α: the  α-projected database , if  α ≠ <>, Otherwise the sequence database S.
    Method:
(1). Scan  S |α once, find the set of frequent items b such that:
    (i). b can be assembled to the last element of α to form a  sequential pattern; or
    (ii).<b> can be appended   to α  to form a sequential pattern.
(2). For each frequent item b, append it to α to form a sequential pattern α' , and output α'.
(3). For each α' , construct α' -projected database S |α'.
(4). Call PrefixSpan(α' ,l+1, S |α').
```

4. Implementation of protein sequence pattern mining

This system finds all sequential patterns using PrefixSpan algorithm. Firstly, user has to input the minimum threshold and the protein sequence database which he wants to find the sequential pattern. Then, the system creates projected databases recursively and finds sequential patterns again and again by comparing support count of sequential pattern and minimum support count. Finally, the system produces all sequential patterns which support count are greater than or equal to the minimum support count. And the user can use the final output of the system to do desired diagnoses and research work.

4.1 System Design

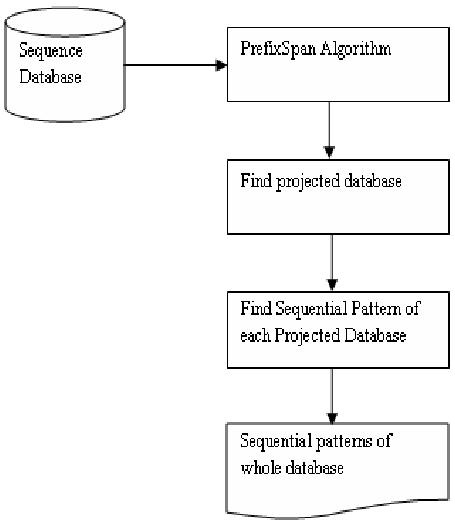
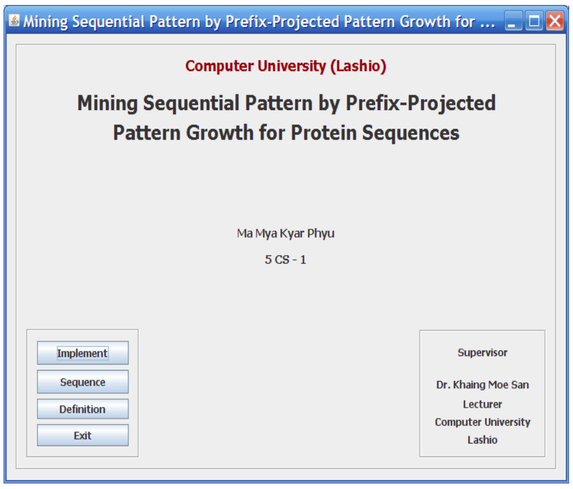

Sequence Database
PrefixSpan Algorithm
Find projected database
Find Sequential Pattern of each Projected Database
Sequential patterns of whole database

Figure 1. Overview of the system

The system flow diagram of the system is shown in figure 1. Firstly, the system partitions the protein sequence database into (n) search space with (n) prefixes. For each search space, the system finds local projected databases. For each local projected database, the system finds the sequential patterns corresponding to the respective prefixes which support count is greater than minimum support count. After sequential patterns for all local projected database of a search space has been got, the system executes the same process for another search space until all search spaces are completed.

In this section, the detail implementation of the system is presented. The main interface of the system is shown in figure 2. There are three main parts in this system. The 'Definition' part shows the actual name and the 3-letter-symbols of each of 20 amino acids. They are represented by one-letter symbols in this system. The 'Sequence' part can be used to add new protein sequences into the sequence database and, to delete sequences from the sequence database. The 'Implement' part is the most important part of the system. In this part, the sequential patterns of the desired sequences, which are stored in sequence database, can be found by the desired value of minimum support.


Mining Sequential Pattern by Prefix-Projected Pattern Growth for ...

Computer University (Lashio)

Mining Sequential Pattern by Prefix-Projected Pattern Growth for Protein Sequences

Ma Mya Kyar Phyu

5 CS - 1

Supervisor

Dr. Khaing Moe San

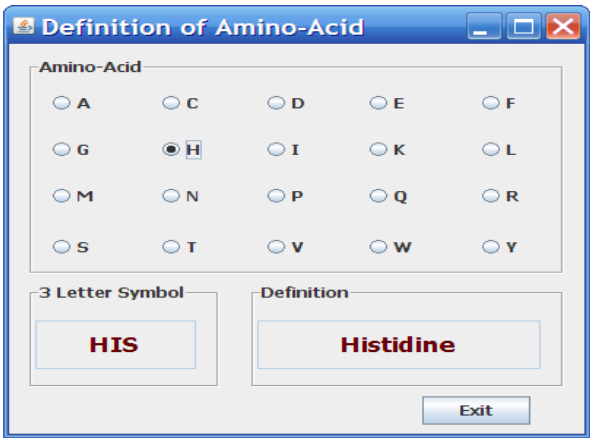
Lecturer

Computer University

Lashio

Figure 2. Main interface of the system

In this part, one-letter-symbols for each of the 20 amino acids are shown. The definition of amino acids commonly found in protein sequences are A-Alanine, V-Valine, L-Leucine, I-Isoleucine, F-Phenylalanine, P-Proline, M-Mehionine, D- Aspartic Acid, E-Glutamic Acid, K- Lysine, R- Arginine, S- Serine, T-Threonine, C- Cysteine, N- Asparagine, Q-Glutamine, H- Histidine, Y- Tyrosine, W-Tryptophun, G- Glycine. The frame for this part is shown in figure 3.


Definition of Amino-Acid

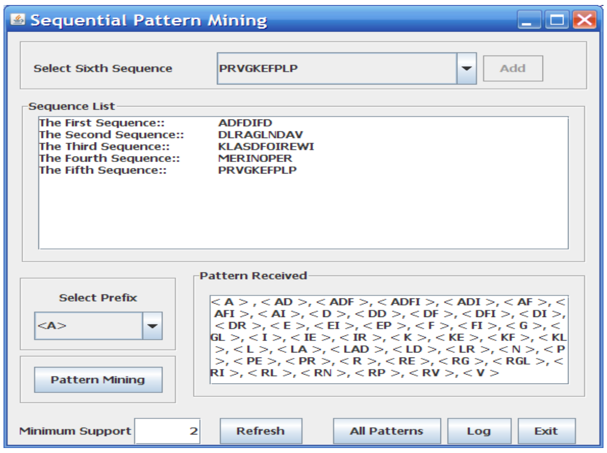
Amino-Acid

A	C	D	E	F
G	[✓ H]	I	K	L
M	N	P	Q	R
S	T	V	W	Y

3 Letter Symbol: HIS

Definition: Histidine

Figure 3. Definition Part of the System


Sequential Pattern Mining

Select Sixth Sequence PRVGKEFPLP Add

Sequence List

```
The First Sequence::       ADFDIFD
The Second Sequence::      DLRAGLNDAV
The Third Sequence::       KLASDFOIREWI
The Fourth Sequence::      MERINOPER
The Fifth Sequence::       PRVGKEFPLP
```

Select Prefix <A>

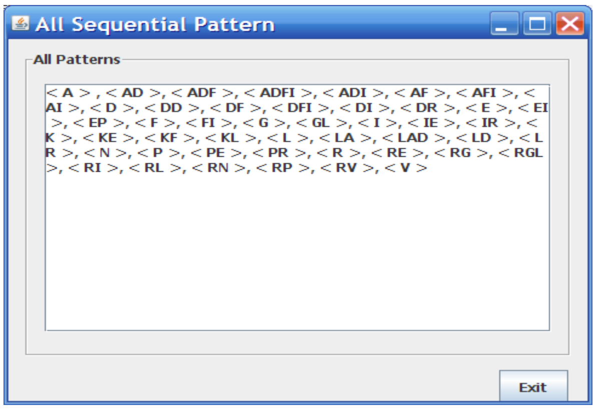
Pattern Received

```
< A > , < AD >, < ADF >, < ADFI >, < ADI >, < AF >, <
AFI >, < AI >, < D >, < DD >, < DF >, < DFI >, < DI >,
< DR >, < E >, < EI >, < EP >, < F >, < FI >, < G >, <
GL >, < I >, < IE >, < IR >, < K >, < KE >, < KF >, < KL
>, < L >, < LA >, < LAD >, < LD >, < LR >, < N >, < P
>, < PE >, < PR >, < R >, < RE >, < RG >, < RGL >, <
RI >, < RL >, < RN >, < RP >, < RV >, < V >
```

Minimum Support 2

Figure 4. Sequential Patterns for Input Sequences and Result Sequences

In this part, the user can find the sequential patterns of desired sequences in the sequence database by giving the desired value of minimum support. If all the necessary input values are given to the system, it will calculate all sequential patterns and show the result to the user .The frame for this part is shown in figure 4.


All Sequential Pattern

All Patterns

```
< A > , < AD >, < ADF >, < ADFI >, < ADI >, < AF >, < AFI >, <
AI >, < D >, < DD >, < DF >, < DFI >, < DI >, < DR >, < E >, < EI
 >, < EP >, < F >, < FI >, < G >, < GL >, < I >, < IE >, < IR >, <
K >, < KE >, < KF >, < KL >, < L >, < LA >, < LAD >, < LD >, < L
R >, < N >, < P >, < PE >, < PR >, < R >, < RE >, < RG >, < RGL
>, < RI >, < RL >, < RN >, < RP >, < RV >, < V >
```

Figure 5. All Sequential Patterns for All Prefixes

If the user wants to see the all sequential patterns, he can press the ‘All Patterns’ button and the system will show all of the patterns for all prefixes as shown in figure 5.

5. Experimental Results

This system implements prefixspan algorithm for generating frequent sequential patterns of protein sequence datasets. One of the crucial topics in the analysis of biological data is the discovery of frequent patterns in a set of protein sequences. These patterns usually hint at shared biological functions. 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. Protein sequences are essential biological data and exist in huge volumes. Discovering patterns from protein sequence require detail analysis of each protein sequences.

Protein sequences are composed of 20 amino acids. The definition of amino acids commonly found in protein sequences are A-Alanine, V-Valine, L-Leucine, I-Isoleucine, F-Phenylalanine, P-Proline, M-Mehionine, D- Aspartic Acid, E- Glutamic Acid, K-Lysine, R- Arginine, S- Serine, T- Threonine, C-Cysteine, N- Asparagine, Q- Glutamine, H-Histidine, Y- Tyrosine, W- Tryptophun, G- Glycine.

The prefixspan recursively projects a sequence database into a set of smaller projected sequence database by exploring only locally frequent fragments. In this paper, prefixspan algorithm is presented for generating frequent sequential patterns using protein sequences.

6. Conclusions

The system creates projected databases recursively and finds sequential patterns again and again by comparing support count of sequential pattern and minimum support count. This system will be very useful in bioinformatics areas for the research of biological sequences and patterns. The system produces all sequential patterns which support count are greater than or equal to the minimum support count. The search space is reduced at each step allowing for better performance, in the presence of small support threshold.

7. References

[1]Agrawal and Srikant. Mining sequential patterns.Mar.1995.

[2]Dhany Saputra, Dayang R. A. Rambli and Oi Mean Foong. Mining sequential patterns using I-PrefixSpan.

[3] H.Lu, J. Han, and L. Feng, "Stock Movement and n-Dimensional Inter-Transaction Association Rules" Proc. 1998 SIGMOD Wordshop Research Issues on Data Mining and Konwledge Discovery (DMKD' 98). Vol, 12, june 1998

[4] Joshua Ho, Lior Lukov and Sanjay ChawlaIn. Time efficient sequential pattern mining algorithm, Pex-SPAM.

[5] J.Han, G. Dong, and Y. Yin, "Efficient Mining of Partial Periodic Patterns in Time Series Database", Apr. 1998.

[6] R. Srikant and R. Agrawal " Mining Sequential Patterns: Generalizations and Performance Improvements," Mar. 1996.