Using Sequential Pattern Mining in Protein Sequences Discovery with Gap

Khadijeh Jahanian

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.

Abstract: Sequences are one of the most important types of data. Recently, mining and analysis of sequence data has been studied in several fields. In a protein sequence may exist other characters than not exist in alphabet. It is related to a function of the protein that has been preserved in the evolutionary process of an organism. Discovery of protein sequences are hard job for algorithms. We present an algorithm that discovery this protein sequences in datasets. Our algorithm used sequential pattern mining method for in problems.

Key words: Sequence mining; Gap; Protein sequences;

INTRODUCTION

Allow \( I = \{a_1, a_2, ..., a_n\} \) be a set of items. A sequence \( s \) is a set of variable, represented as \(<B_1, B_2, ..., B_l>\), where \( B_i \ (1 \leq c \leq 2) \), called events (or item sets) \((\{1\}, \{2\})\). Each event is a set represented as \((y_1, y_2, ..., y_m)\) where \( y_k \ (1 \leq p \leq n) \) is an item. For simplicity, the brackets are omitted if an element has only one item. A sequence dataset \( S \) is a set of sequences \((\{1\}, \{2\})\). The total number of elements in the sequence set is called the length of the sequence and a sequence with length \( l \) is called an \( l \)-sequence \((\{1\}, \{2\})\). A sequence \( b = <b_1, b_2 > \) is called a subsequence of another sequence \( \mu = <m_1, m_2, ..., m_n> \), represented as \( \beta \subseteq \mu \), if there exist integers \( 1 \leq c_1 \leq ... \leq c_n \leq m \), such that \( b_1 \leq m_{c_1}, b_2 \leq m_{c_2}, ..., b_2 \leq m_{c_n} \). If \( \alpha \) is a subsequence of \( \mu \), we utter that \( \mu \) includes \( \beta \) \((\{1\}, \{2\})\). The support of a sequence \( \beta \) in a sequence dataset \( S \), presented \( \text{support}(\beta) \), is the number of sequences in the dataset containing \( \beta \) given a minimum support threshold, \( \text{min sup} \), the a group of sequential pattern, \( \text{SP} \), is the group of all the subsequences whose support values are no less than \( \text{min sup} \) \((\{1\}, \{2\})\).

The objective of sequential pattern mining is to discover frequent subsequences in a dataset. Sequential pattern mining has multiple applications, comprising the finding frequent units in DNA sequences, the inspection of scientific or curative processes and analysis of web log registered acts. Several sequential pattern mining algorithms have been proposed so far. \((\{3\}, \{4\}, \{7\})\)

A motif is a featured pattern in amino acid sequences. It is related to a function of the protein that has been preserved in the evolutionary process of an organism. The amino acid sequence is composed of 20 kinds of alphabets. The featured pattern, which includes wild cards, is discovered from frequent patterns extracted in the sequences. A Protein sequence motif, signature or consensus pattern, is a short sequence that is embedded within the sequences of a same protein family (Bork & Koonin, 1996). By identifying protein sequence motifs, an unknown sequence can be quickly classified into its computationally predicted protein family/families for further biological analysis.

2. Previous Studies About Motif Discovery:

In past years, many algorithms for finding protein sequence motifs have been proposed. Sequence motif discovery algorithms can be generally categorized into 3 types: 1) String Alignment algorithms, 2) Exhaustive enumeration algorithms, and 3) Heuristic methods. String alignment algorithms (Waterman et al, 1984; Delcoigne & Hansen, 1975; Needleman & Wunsch, 1970) find sequence motifs by minimizing a cost function which is related to the edit distances between sequences. Multiple alignment of sequences is a NP-hard problem and its computational time increases exponentially with the sequence size. Heuristic methods ((Jonassen & Higgins, 1995), ( Sagot & Viari, 1996)) can have a better performance but are usually less flexible.

The existing pattern extraction algorithms, which include multiple alignment, statistical method, present some problems. (Kitakami & et al, 2002), (Rigoutsos & Floratos, 1998), (Bateman & et al, 1999), (Wang & Parthasarathy, 2004), (Sonnhamer & et al, 1997), (Mitchell & et al, 1990), (Bill & et al, 2002), (Li & et al, 2008), (Lones & et al, 2007), (Syed & et al, 2010). These algorithms are neither functional nor fast for the discovery of motifs from large-scale amino-acid sequences.

There are a lot of algorithms for solving this problem but none of them are as useful as our algorithm. (Wang & Parthasarathy, 2004), (Chang & Halgamuge, 2002), (De Amo & et al, 2009), (Li & Wang, 2008), (Altschul, 1998). In figure 1 shown Previous Studies About Motif Discovery. Our represented method for motifs extraction is suitable than explained methods. Our algorithm also can be suitable to find all of the frequent pattern. In this method it is not necessary to specify the average length of sequences.

Corresponding Author: Khadijeh Jahanian, Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran.
Mob: +989355357675
3. Algorithm:

In here, we present a brief description of this algorithm. Let DB be a sequence dataset. The algorithm starts with a scan of DB to identify the frequent 1-sequences. Let i be a sequence, a projection i of DB, denoted as P(i, DB), is a set of subsequences, which are made up of the sequences in DB containing i after deleting the events appearing before the first occurrences of i within each sequence for instance, Table 1 shows a sequence dataset. With the support threshold as 2 the projected dataset for sequence AB is P(AB, DB) = {C, CB, C, BCA} as you see sequences of AB routes be deleted and remain subsequence constitute this set. Performance studies have shown that the prefixspan algorithms is more efficient than the other algorithm. After the projected datasets are built, algorithm searches each projected dataset and selects the sequential patterns. Our algorithm can generate frequent patterns including some wild cards. For instance, "P=ATT***TTC*GGATT**T*CCC*GG" are considered to be a protein sequence. where * indicates one wild card symbol. it can be every symbol. the other method considered TT***T and TT**T as a TTT pattern, but in our suggested algorithm they known as TT2T and TT3T. The our method generates (k+1)-length frequent patterns from each k-length frequent pattern in a set of sequences. The last character of each (k+1)-length frequent pattern is found from one of the characters that exist among the next position of a k-length frequent pattern and the last position in the sequences.

First of all, the our method extracts the 1-length frequent patterns with considering their min support. this method extracts the 2-length frequent patterns from one 1-length frequent pattern. In fact, this method extracts the (k+1)-length frequent pattern from k-length frequent pattern. For instance, figure 2 shows frequent patterns that are extracted in the two sequences. The number of wild cards is 3 (table 1). figure 1 show how this represented algorithm work. this method extracts 1-length frequent patterns, "K, L, M, N, P, R, S, T" that support min support. Next, this method extracts 2-length frequent patterns from 1-length frequent patterns, When the 1-length frequent pattern is "K" the 2-length frequent patterns are "K*L". Because of the different number of wild cards in the two sequences, the 2-length frequent pattern,"K1L," is extracted. Next, this method extracts 3-length frequent patterns from the 2-length frequent pattern, "K*L". The extracted 3-length frequent pattern is "K*LR" when the 1-length frequent pattern is "M" the 2-length frequent patterns are "MN". Because of the different number of wild cards in the two sequences, the 2-length frequent pattern, "MN" is not extracted. The 2-length frequent pattern "MS" is also not extracted because of the different number of wild cards. thus this algorithm extract wild cards and number of them.
Table 1: Dataset for example.

<table>
<thead>
<tr>
<th>seq_id</th>
<th>sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>MFKALRTIPVILNMNKDSKLCPN</td>
</tr>
<tr>
<td>2</td>
<td>MSPNPTNHTGKTLR</td>
</tr>
</tbody>
</table>

Fig. 2: Extraction of frequent pattern of Mprefixspan method.

In figure 3 show Our Method for Discovery Motif.

| Input: the set sequence Proteins with Wild Card, and the min support threshold min_sup, M and N: the parameters of a gap constraint. |
| Output: The complete set of motifs |
| Method: Call Motifscan(<>,0,S,M,N) |
| Subroutine: Motifscan(α, l, S|α,M,N) |
| Parameters: |
| • α: 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: |
| a) b can be assembled to the last element of α to form a sequential pattern; |
| b) <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’, and call Motifscan (α’, l+1, S|α’,M,N). |

Fig. 3: Our Algorithm (Motifscan).

4. Evaluation:
All of our experiments were performed on a core 4CPU AMD (phenom X4 AMD) with using 4GB memory. Table 3 shows the detail of these data sets. These datasets are offered by NCBI.

Table 3: Detail of used datasets.

<table>
<thead>
<tr>
<th></th>
<th>#seq</th>
<th>Total.len</th>
<th>Ave.seq.len</th>
<th>Max.seq.len</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sequence_1</td>
<td>110</td>
<td>33385</td>
<td>435</td>
<td>3872</td>
</tr>
<tr>
<td>Zince finger</td>
<td>467</td>
<td>245595</td>
<td>525</td>
<td>4036</td>
</tr>
<tr>
<td>Sequence_2</td>
<td>320</td>
<td>205220</td>
<td>495</td>
<td>4025</td>
</tr>
</tbody>
</table>
The performance ratio of each parameter is as follows:
1) Zince finger data set, min_sup = 2, min support ratio 40, and wild cards number 2-7.
2) Sequence_1 data set, min_sup = 2, min support ratio 40, and wild cards number 2-5.
3) Sequence_2 data set, min_sup = 2, min support ratio 40, and wild cards number 2-7.

We compare Motifscan with ECSG algorithm (Wang & et al, 2007) that result shown in figure 4. also we tested the influence of changing min support threshold on the performance of our algorithm. The results are shown in figure 5. our algorithm shows stable performance with different support threshold. also we tested the influence of changing wild cards number on the performance of our algorithm. The results are shown in figure 6. As the result our algorithm shows stable performance with the changing wild cards number.

Fig. 4: Compare Motifscan and ECSG.

Fig. 5: Efficacy of changing min support.

Fig. 6: Influence of changing wild cards.
5. Conclusions:

In this paper, we suggest sequential pattern mining algorithm motifscan for discovery motif. It can mine motif sequences with considering the wild cards. The amino acid sequence used by the verification experiment was a small-scale sequence. It will be necessary to verify the results by using a variety of amino acid sequences in the future.

REFERENCES


Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Helen Pinto, 2001. PrefixSpan: Mining Sequential Patterns Efficiently by Prefix-Projected Pattern Growth. in International Conference on Data Engineering (ICDE), IEEE Computer Society Press, 16(11): 215-224.

