Computers in Biology and Medicine ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Computers in Biology and Medicine

journal homepage: www.elsevier.com/locate/cbm

Mining frequent biological sequences based on bitmap without candidate sequence generation

Qian Wang [a,b,c,*], Darryl N Davis [c], Jiadong Ren [a,b]

[a] College of Information Science and Engineering, Yanshan University, Qianhuangdao, Hebei, China
[b] Computer Virtual Technology and System Integration Laboratory of Hebei Province, China
[c] Department of Computer Science, University of Hull, Hull, UK

ARTICLE INFO

Article history:
Received 28 September 2015
Accepted 22 December 2015

Keywords:
Biological sequence
Frequent pattern
Bitmap
Quicksort list

ABSTRACT

Biological sequences carry a lot of important genetic information of organisms. Furthermore, there is an inheritance law related to protein function and structure which is useful for applications such as disease prediction. Frequent sequence mining is a core technique for association rule discovery, but existing algorithms suffer from low efficiency or poor error rate because biological sequences differ from general sequences with more characteristics. In this paper, an algorithm for mining **F**requent **B**iological **S**equence based on **B**itmap, **FBSB**, is proposed. FBSB uses bitmaps as the simple data structure and transforms each row into a quicksort list QS-list for sequence growth. For the continuity and accuracy requirement of biological sequence mining, tested sequences used during the mining process of FBSB are real ones instead of generated candidates, and all the frequent sequences can be mined without any errors. Comparing with other algorithms, the experimental results show that FBSB can achieve a better performance on both run time and scalability.

Introduction

Biosequence mining can help people recognize interesting and important relationships between biological sequences for human genome research. There are usually different functions in the sequences, some of them are because of a special element, and some of them are the result of the interaction of a few elements. Biosequence mining is one of the key technologies to discover the single and mutual functions of the elements or the sequences. It can give reasonable prediction and guidance for making human nucleic acid, protein and other biological data. By identifying the protein-coding genes from DNA sequences, it can be found that some gene combination mode is related to drug allergy or appears frequently in some disease.

Bioinformatics refers to several subjects such as computer science, information science, and mathematics. Computer processing is a major part in the analyzing of biological data [1]. Even at the beginning of the biological research and development, many pattern mining algorithms for biosequence were proposed. Brazma [2] surveyed approaches to the pattern discovery in biosequences and placed these approaches within a formal framework that systematizes the types for algorithm comparison. It is found that most of the algorithms face efficiency problems when the pattern space grows rapidly.

Mining frequent patterns is an indispensable component in many data mining tasks such as association rule mining. These association rule mining algorithms can be partitioned into different categories. Apriori [3] and FP-growth [4] are two classic algorithms. Apriori uses candidate generate-test technique and FP-growth is based on a tree structure which records the sequence paths without candidate generation. The algorithms based on them can be classified in terms of whether there is candidate generation. Another classic algorithm is Eclat [5]. It is different because it expresses the dataset vertically. Algorithms can also be classified by horizontal or vertical dataset expression forms. BitTableFI [6] is BitTable-based and the BitTable is horizontally and vertically indexed. Although efficient bit wise operations are used, candidate generation and test ensure that BitTableFI suffers with high computational costs. Index-BitTableFI [7] is proposed to solve this problem. Index array and the correlative calculate method are applied in using BitTable horizontally. A breadth-first search strategy is used for quick identification of the co-occurrence items and depth-first search is for mining all the frequent itemsets in different levels. DBV-Miner [8] is presented by Bay et al. and developed for mining frequent closed itemsets. Dynamic Bit-Vector(DBV) and a lookup table are used, and the support of itemsets can be quickly computed by the intersection between two DBVs.

* Correspondence to: College of Information Science and Engineering, Yanshan University, Qianhuangdao, Hebei 066000, China. Tel.: +86 13731767789.
E-mail address: wanqianysu@163.com (Q. Wang).

http://dx.doi.org/10.1016/j.compbiomed.2015.12.016
0010-4825/

Bay et al. presented more algorithms based on various data structures, like lattice-based algorithms [9–11] and N-list based algorithm [12]. Sequential pattern mining is originally put forward by Agrawal et al. [13], who also presented three sequential mining algorithms. Afshar formally defined the maximal frequent sequence and proposed the corresponding algorithm MaxSequence [14]. MaxSequence needs to generate and test candidate maximal sequences, and a compressed prefix tree is used to store maximal frequent sequences. When the database is larger, the overhead for maintaining and storing the prefix tree compromises the scalability of the algorithm. A highly compressed lattice storage structure and a breadth-first approach are used by FMMSP [15], and maximal frequent sequences and closed frequent sequences are mined quickly without candidate generation.

Because of the nature of biology data, the above algorithms are not necessarily suitable for biological data mining. There are more algorithms designed for biosequences these days. TRFinder [16] algorithm seeks tandem repeats which can cause human disease and consist of two or more copies of nucleotide patterns. A probabilistic model and a statistical criteria collection are used to detect tandem repeats. REPuter [17] is based on a suffix tree and a sequence alignment technique for detecting various types of repeats in DNA, it circumscribes the wide scope of repeat analysis, but it is not efficient for mining frequent repeats. BioPM algorithm [18] was developed for protein sequence mining and it introduces the concept of multiple supports so as to improve performance and efficiency. The mMbioPM [19] algorithm optimizes the structure of hash lists to improve the efficiency of BioPM and reduces the run time. However, the efficiency of the BioPM algorithm and its improvements are not quite ideal because of the large scale of the projected database when the minimal support is lower. MSPM [20] is based on the prefix-tree structure, and it presents the concept of primary pattern which makes the degree of the prefix tree a constant. By avoiding too many short patterns generations, the scale of the prefix-tree will not be too large. Its efficiency is much better but it may miss frequent sequences. An index-based approach [21] proposes an interesting measure for meaningful biological information. Each leaf node in the tree structure is an array whose length is variable. The transaction ID and starting position of each sequence are stored in the arrays for memory reduction. CBFMM [22] mines nucleotide and protein sequence with a variety of FP-tree-based model definitions. Repetition detection in biological data is required for potential malfunction and disease identification. DPMine [23] is designed for colossal sequence discovery from biological dataset. It integrates a DPT+ tree, a doubleton data matrix and a one-dimensional array to find doubleton patterns which may further generate colossal sequences. For irrelevant regions in biological sequence evolution such as mutations, gap constraints need to be considered. DFSG [24] is designed for the sequences which are not conserved. For biological network analysis, graph simplification technology is used, aiming to reduce the graph size [25]. An overview is given in [26] to illustrate how the various frequent pattern mining algorithms can be used for human bioinformatic applications.

The FBSB algorithm, reported here, is proposed to mine frequent biosequences with a high efficiency and less memory space requirements. It first calculates the supports of the items for frequency mining and records the end position value of each 2-sequence to form a bitmap which is further used for frequent 2-sequence mining and pattern growth. A quicksort list is created for fast connecting sequences in the same biosequence. Two sequences in the quicksort with the position values next to each other in an ascending order can be connected easily by adding the last item of the second sequence to the first sequence, and the position value of the second sequence is used as the position value of the new sequence. The bitmap is updated and the storage space is released constantly. There is no candidate generation and every frequent sequence can be obtained. The algorithm can satisfy the requirements of biological data mining because of its good efficiency and high quality results. Experimental results show that FBSB is much better when compared with other methods.

The remainder of the paper is organized as follows. Section 2 introduces the problem definition about biological sequences and bitmap formation. Section 3 develops the FBSB algorithm and gives some examples. Section 4 presents the performance study of FBSB algorithm. Section 5 contains the concluding remarks.

Preliminaries and problem definitions

Problem definition

DNA and protein sequences are two typical types of biosequences. It should be noted that there are differences between biosequences and general sequences, so some definitions are given as follows.

Definition 1. Let Σ be an alphabet, a sequence $S=<s_1s_2\ldots s_m>$ with $s_i \in \Sigma (i=1,\ldots,m)$ is called a DNA sequence if $\Sigma=\{A,C,G,T\}$ consists of four nucleotides, or it is called a protein sequence if Σ consists of the 20 symbols for amino acids. A sequence can be called a k-sequence if it contains k nucleotides or amino acid symbols.

Example 1. Let $\Sigma=\{a,b,c\}$ and $S=<bacaab>$, S is a 6-sequence because there are six items in S.

Definition 2. Let sequence $S_1=<a_1,a_2,\ldots,a_m>$ and sequence $S_2=<b_1,b_2,\ldots,b_n>$ be two sequences $(m<n)$ on the alphabet Σ. S_1 is a subsequence of S_2 if there exist integers $i_1, i_2,\ldots,i_m$, such that $1\le i_1<i_2<\ldots<i_m\le n$ and $a_1=b_{i1}$, $a_2=b_{i2}$, …, $a_m=b_{im}$. It can also be said that S_2 is a super sequence of S_1.

Example 2. Let $S_1=<bacaab>$, $S_2=<aca>$, $S_3=<bab>$. S_2 is a subsequence of S_1, but S_3 is not a subsequence of S_1. This is different from general sequences; as the items of the subsequence must occur contiguously in the super sequence.

Definition 3. Given a database of biosequences D and a biosequence S. The support of the biosequence S in D, denoted as sup(S), is the number of the sequences in D which contains S as its subsequence.

Definition 4. Given a biosequence database $D=\{S_1,S_2,S_3,\ldots,S_{|D|}\}$ and a user-defined threshold ζ, where $|D|$ is the number of sequences in D. The minimal support min-sup can be calculated as min-sup$=\zeta^*|D|$. If the support of a biosequence S satisfies sup(S) $\ge$ min-sup, S is called a frequent sequence in D. If S is a k-sequence, it can be called a frequent k-sequence.

Property 1. For a sequence S, if S is an infrequent sequence, then any of its super sequences are also infrequent.

Bitmap of the sequence

For efficient mining process, all the position values of the 2-sequence occurrences are recorded. A bitmap is constructed to store the values and can be updated as the frequent sequences grow. A sequence may occur several times in the same database transaction, and all its occurrence positions should be put into a position array.

Definition 5. A bitmap is a two-dimensional table, where each row represents the ID of a sequence in the database and each column represents a sequence. A bitmap cell is denoted as $Pos_i(S)$,

Please cite this article as: Q. Wang, et al., Mining frequent biological sequences based on bitmap without candidate sequence generation, Comput. Biol. Med. (2016), http://dx.doi.org/10.1016/j.compbiomed.2015.12.016