Research Journal of Applied Sciences, Engineering and Technology 4(20): 4039-4042, 2012
ISSN: 2040-7467

Submitted: January 11, 2012 Accepted: April 20, 2012 Published: October 15, 2012

Using Pattern Mining Algorithms in Discovery Biological Data

[1]Morteza Poyan Rad, [1]Seyed Hossein Hosseiniasl and [2]Ali Abbaszadeh Sori
[1]Department of Computer Science, Babol Branch, Islamic Azad University, Babol, Iran
[2]Department of Computer, Qaemshahr Branch, Islamic azad University, Qaemshahr, 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 DNA sequences may exist other characters then not exist in alphabet. It is related to a function of the DNA that has been preserved in the evolutionary process of an organism. Discovery of DNA sequences dataset that contains gap is very hard job for algorithms. We present an algorithm that discovery this DNA sequences in datasets. Our algorithm used sequential pattern mining method for in problems.

Keywords: DNA sequences, gap, parallel mining, pattern mining, sequence mining

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_c ($1 \le c \le z$), called events (or itemsets) (Guozhu and Jian, 2007; Han and Kamber, 2006). Each event is a set represented as (y_1, y_2, ..., y_m) where,,,,,, y_k ($1 \le p \le n$) is an item. For briefness, the brackets are omitted if an element has only one item. A sequence dataset S is a set of sequences (Guozhu and Jian, 2007; Han and Kamber, 2006). The total number of items in a sequence is called the length of the sequence and a sequence with length l is called an l-sequence (Guozhu and Jian, 2007; Han and Kamber, 2006). A sequence β = <b_1, b_2 ... b_n> is called a subsequence of another sequence μ = <m_1, m_2 ... m_m>, represented as $\beta \subseteq \mu$, if there exist integers $1 \le c_1 \le ... \le c_n \le m$, such that $b_1 \subseteq m_{c1}$, $b_2 \subseteq m_{c2}$,..., $b_n \subseteq m_{cn}$. If α is a subsequence of μ, we utter that μ includes β (Guozhu and Jian, 2007; Han and Kamber, 2006). The support of a sequence β in a sequence dataset S, presented support (β), is the number of sequences in the dataset containing β given a minimum support threshold, min sup, the a group of sequential pattern, SP, is the group of all the subsequences whose support values are no less than min support (Guozhu and Jian, 2007; Han, 2006).

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. Pei *et al*. (2007), Agrawal and Srikant (1995) and De *et al*. (2009).

The DNA sequences are composed of 4 kinds of alphabets (A, G, T and C). The featured pattern, which includes gap, is discovered from frequent patterns extracted in the sequences. A DNA sequence with gap, signature or consensus pattern, is a short sequence that is embedded within the sequences of a same DNA family (Bork and Koonin, 1996). By identifying DNA sequence with gap, an unknown sequence can be quickly classified into its computationally predicted DNA family/families for further biological analysis.

PREVIOUS STUDIES ABOUT MOTIF DISCOVERY

In past years, many algorithms for finding DNA sequence have been proposed. DNA sequences discovery algorithms can be generally categorized into 3 types:

- String Alignment algorithms
- Exhaustive enumeration algorithms
- Heuristic methods

String alignment algorithms (Waterman *et al*., 1984; Needleman and Wunsch, 1970) find DNA sequences by minimizing a cost function which is related to the edit distances between sequences. Multiple alignments of sequences is a NP-hard problem and its computational time increases exponentially with the sequence size. Heuristic methods (Inge *et al*., 1995; Sagot and 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. Hajime *et al*., (2002), Isidore and Aris

Corresponding Author: Morteza Poyan Rad, Department of Computer Science, Babol Branch, Islamic Azad University, Babol, Iran

(1998), Chun and Jianyong (2008), Michael and Andy (2007) and Bill and Saman (2002). 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 (De *et al*., 2009; Chun and Jianyong, 2008; Bill and Saman, 2002). Our algorithm also can be suitable to find all of the frequent pattern, so our algorithm can find DNA sequences that contain gap. Our algorithm does faster and easier discovery DNA sequences contain gap. In this method it is not necessary to specify the average length of sequences.

The remainder of the study has been organized as follows: in Section 3, our algorithm is described in details. The experimental results are presented in Section 4 and conclusions are given in Section 5.

ALGORITHM

In here, we present a brief description of this algorithm. Let DB be a sequence dataset (Table 1). The algorithm starts with a scan of DB to identify the frequent 1-sequences. Then, a second scan of DB constructs the projected datasets for 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. (Han and Kamber, 2006; Jian *et al*., 2001). Performance studies (Pei *et al*., 2007; Wang and Han, 2004) have shown that the prefixspan algorithms is more efficient than the other algorithm (Agrawal and Srikant, 1995; Zaki, 2001; Yan *et al*., 2003).

After the projected datasets are built, algorithm searches each projected dataset and selects the sequential patterns. Our algorithm can generate frequent patterns including some gaps. For instance, "P = ATT***TTC*GGATT**T*T*CCC*GG" are considered to be a DNA sequence. Where,,,,,, * indicates one gap 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. 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

Table 1: Dataset for example

Seq_id	Sequence
MFKALRTIPVILNMNKDSKLCPN	1
MSPNPTNHTGKTLR	2

Input: The set sequence DNAs with Gap and the min support threshold min sup, M and N: the parameters of a gap constraint.
Output: The complete set of Pattern of protein
Method: Call DNAscan(<>,0,S,M,N)
Subroutine: DNAscan(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 $S|_a$ 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 $S|_{\alpha'}$, and call DNAscan (α', l+1, $S|_{\alpha'}$,M,N).

Fig. 1: Our algorithm

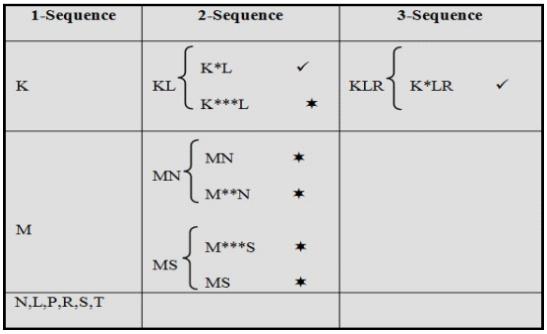

Fig. 2: Extraction of frequent pattern of our method

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, 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, Fig. 1 shows frequent patterns that are extracted in the two sequences, the number of gap is 3 (Table 1). Figure 2 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".

Table 2: Detail of used datasets

	#seq	Total.len	Ave. seq.len	Max. seq.len	Gaps number
DNAseq_1	135	35674	454	4773	2-6
Zince finger	497	355595	625	5136	3-6
DNAseq_2	385	315220	585	5029	2-8
Kringle	70	23385	334	3176	3-7

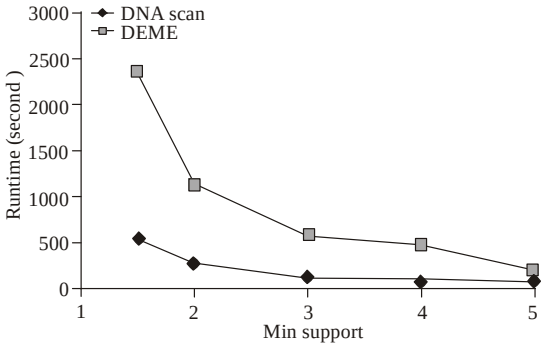

Fig. 3: Compare DNAscan and DEME

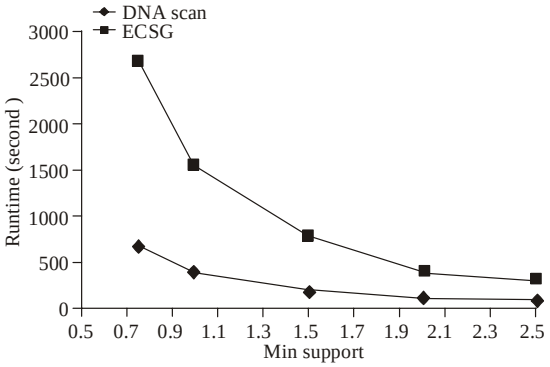

Fig. 4: Compare DNAscan and ECSG

Because of the different number of gap 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 gap 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 gap. Thus this algorithm extracts gap and number of them.

In Fig. 1 show Our Method for Discovery DNA sequences contain gap.

EVALUATION

All of our experiments were performed on a core 4CPU AMD (phenom X4 AMD) with using 4GB memory. Table 2 shows the detail of these data sets. These datasets are offered by NCBI.

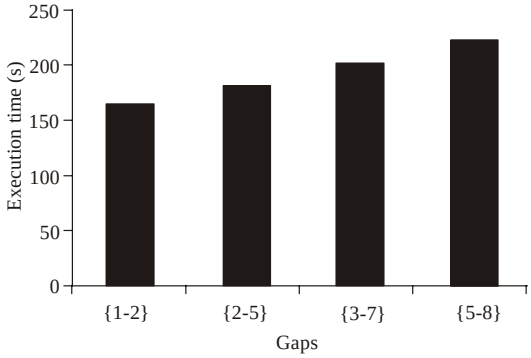

Fig. 5: Influence of changing gaps

We compare DNAscan with DEME algorithm (Redhead and Bailey, 2007) and ECSG algorithm (Chun and Jianyong, 2008) that result shown in Fig. 3 and 4. 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 Fig. 5. As the result our algorithm shows stable performance with the changing gaps number.

CONCLUSION

We suggest sequential pattern mining algorithm DNAscan for discovery DNA sequences contain gap. Our algorithm does faster and easier discovery DNA sequences contain gap. It can mine protein sequences with considering the gaps. The DNA sequence used by the verification experiment was a small-scale sequence. It will be necessary to verify the results by using a variety of DNA sequences in the future. In future, our algorithm can be designed to parallel, that speed of the algorithm higher takes.

REFERENCES

Agrawal, R. and R. Srikant, 1995. Mining sequential patterns. International Conference on Data Engineering (ICDE), IEEE Press, Taipei, Taiwan, pp: 3-14.

Bill, C.H.C. and K.H. Saman, 2002. Protein motif extraction with neuro-fuzzy optimization. J. Bioinformat., 18(8): 1084-1090.

Bork, P. and P. Koonin, 1996. Protein sequence motifs. Curr. Opin. Struct. Biol., 6: 366-376.

Chun, L. and W. Jianyong, 2008. efficiently mining closed subsequences with gap constraints. ACM T. Database Syst., 32: 313-322.

De, A., S. Giacometti, A. Pereira Jr. and W. Clemente, 2009. MILPRIT: A constraint based algorithm for mining temporal relational patterns. Int. J. Data Warehous. Min., 4(4): 42-61.

Guozhu, D. and P. Jian, 2007. Sequence data mining. Adv. Database Syst., 33: 15-45.

Hajime, K., K. Tomoki, M. Yasuma, K. Susumu and Y. Yukiko, 2002. Modified prefixspan method for motif discovery in sequence databases. PRICAI, Springer-Verlag, 2417: 482-491.

Han, M.K., 2006. Data Mining Concepts and Techniques. 2nd Edn., University of Illinois at Urbana-Champaign, Elsevierdirect, pp: 498-505.

Isidore, R. and F. Aris, 1998. Motif discovery without alignment or enumeration. Proceedings of Second Annual ACM International Conference on Computational Molecular Biology (RECOMB), pp: 221-227.

Inge, J., F.C. John and G.H. Desmond, 1995. Finding flexible patterns in unaligned protein sequences. Protein Sci., 4(8): 1587-1595.

Jian, P., H. Jiawei, M.A. Behzad and P. Helen, 2001. PrefixSpan: mining sequential patterns efficiently by prefix-projected pattern growth. International Conference on Data Engineering (ICDE), IEEE Computer Society Press, 16(11): 215-224.

Michael, L. and T. Andy, 2007. Regulatory Motif Discovery Using A Population Clustering Evolutionary Algorithm. IEEE/ACM T. Comput. Biol. Bioinformat., 4(3): 403-414.

Needleman, S. and C. Wunsch, 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Molecul. Biol., 48: 443-453.

Pei, J., J. Han and W. Wang, 2007. LINK"http://portal.acm.org/author_page.cfm?id=81452601906&coll=GUIDE&dl=GUIDE&trk=0&CFID=96302870&CFTOKEN=93775738"\t"_self" Constraint-based sequential pattern mining:The pattern-growth methods. J. Intell. Inform. Syst., 22(2): 133-160.

Redhead, E. and T.L. Bailey, 2007. Discriminative motif discovery in DNA and protein sequences using the DEME Algorithm. BMC Bioinformat., 8: 385.

Sagot, M and A. Viari, 1996. A double combinatorial approach to discovering patterns in biological sequences. Proceedings of the 7th Symposium on Combinatorial Pattern Matching, pp: 186-20.

Wang, J. and J. Han, 2004. BIDE efficient mining of frequent closed sequences. In ICDE, pp: 79-91.

Waterman, M., D. Galas and R. Arratia, 1984. Pattern recognition in several sequences: Consensus and alignment. Bull. Math. Biol., 46: 512-527.

Yan, X., J. Han and R. Afshar, 2003. Clospan: Mining closed sequential patterns in large datasets. In Intelligent Conferance Data Mining (SDM), pp: 166-177.

Zaki, M.J., 2001. Spade: An efficient algorithm for mining frequent sequences. Mach. Learn., 42(1-2): 31-60.