Closed Sequential Pattern Mining in High Dimensional Sequences

Meng Han[1,2], Zhihai Wang[1], Jidong Yuan[1]
[1]School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, 100044, P.R. China
[2]School of Computer Science and Engineering
Beifang University of Nationalities, Yinchuan, 750021, P.R. China
Email: zhhwang@bjtu.edu.com

***Abstract*—High dimensional sequences, such as biological sequences, are characterized by a small number of transactions, and a large number of items in each transaction. Mining sequential patterns in the sequences need to consider different forms of patterns, such as contiguous patterns, local patterns which appear more than one time in a special sequence, and so on. Mining closed patterns might lead to not only a more compact complete result set, but also better efficiency. In this paper, a novel algorithm based on BIDE (BI-Directional Extension) and multi-support is presented for high dimensional sequences specifically. It mainly mines three types of closed sequential patterns which are sequential patterns, local sequential patterns and total sequential patterns. Thorough experimental performances on biological sequences have demonstrated that the proposed algorithm could reduce memory consumption and generate more compact patterns.**

***Index Terms*—sequential pattern mining; high dimensional sequence; closed pattern; biological sequence; data mining**

I. INTRODUCTION

Sequential pattern mining is an important way to discover patterns that occur in many sequences in a given database. Previous methods mining complete set of patterns, which is huge for effective utilization. We need a compact but high quality set of patterns, such as closed patterns and maximal patterns [1, 2]. For a fixed minimum support threshold, the set of closed frequent patterns contains the complete information regarding to its corresponding frequent patterns; whereas the set of max-patterns, though more compact, usually does not contain the complete support information regarding to its corresponding frequent patterns.

In recent years, some methods focus on multi-support, such as BioPM [3], WildSpan [4]. But these algorithms mine the complete patterns. In this paper, a novel algorithm based on BIDE(BI-Directional Extension) [5] and multi-support is provided specifically for high dimensional sequence. It uses to mine three kinds of interesting closed patterns.

The rest of this article is organized as follows. Section 2 reviews BIDE algorithm. In section 3, some concepts are defined, and an improvement of BIDE algorithm: M-bBIDE is proposed. Section 4 shows the experimental results of biological sequential pattern mining and some interesting patterns are provided. Finally, the conclusion is provided in section 5.

II. THE BIDE ALGORITHM

The BIDE [5] is a competitive algorithm for mining closed sequential patterns. It used a sequence closure checking scheme to avoid the maintenance of closed candidate sequence. The proposed back scan pruning method could prune the search space more aggressively than the methods used in CloSpan [6].

The BIDE algorithm is used to mining discontinuous closed patterns. It is not suitable to mining biological datasets. In this paper, we need contiguous patterns. So, the following patterns in this paper are contiguous patterns.

Definition 1 (Closed Sequential Pattern) [1] A pattern X is a closed sequential pattern in a data set S if X is frequent in S and there exists no super pattern Y such that Y has the same support as X in S.

TABLE I.

BIOLOGICAL SEQUENCES

sequence id	sequence
10	(g)(a)(g)(g)(a)(g)(a)
20	(a)(g)(a)(t)(a)(t)(g)(c)(t)(t)(a)(g)(a)(g)
30	(a)(c)(t)(g)(a)(g)(g)(t)(a)(g)(a)
40	(a)(t)(t)(g)(a)(g)(c)(t)(t)

For example, Table Ⅰ shows the input high dimensional biological sequence database S. Suppose the minimum support is 50%, denoted as min_sup=50% (0.5), so the subsequences occurrence frequency in the set of sequences is no less than 2 (4*0.5). The set of items in the database S is {g, t, c, a}, and the *sequence_id* are {10, 20, 30, 40}. There are 7 items in sequence 10, 14 items in sequence 20, 11 items in sequence 30 and 9 items in sequence 40. Since these 4 sequences contain subsequence x=(g)(a)(g), x is a length-3 pattern, and its support is 4, denoted as $support(x)$=4(100%).

TABLE II.

PROJECTED DATABASE AND SEQUENTIAL PATTERNS

prefix	projected database	sequential pattern
g	10: (a)(g)(g)(a)(g)(a)	(g),

doi:10.4304/jsw.8.6.1368-1373

	10: (g)(a)(g)(a) 10: (a)(g)(a) 10: (a) 20: (a)(t)(a)(t)(g)(c)(t)(t)(a)(g)(a)(g) 20: (c)(t)(t)(a)(g)(a)(g) 20: (a)(g) 30: (a)(g)(g)(t)(a)(g)(a) 30: (g)(t)(a)(g)(a) 30: (t)(a)(g)(a) 30: (a) 40: (a)(g)(c)(t)(t) 40: (c)(t)(t)	(g)(c), (g)(a), (g)(g), (g)(c)(t), (g)(a)(g), (g)(c)(t)(t), (g)(a)(g)(g)
t	20: (a)(t)(g)(c)(t)(t)(a)(g)(a)(g) 20: (g)(c)(t)(t)(a)(g)(a)(g) 20: (t)(a)(g)(a)(g) 20: (a)(g)(a)(g) 30: (g)(a)(g)(g)(t)(a)(g)(a) 30: (a)(g)(a) 40: (t)(g)(a)(g)(c)(t)(t) 40: (t)	(t), (t)(a), (t)(t), (t)(g), (t)(a)(g), (t)(g)(a), (t)(a)(g)(a), (t)(g)(a)(g)
c	20: (t)(t)(a)(g)(a)(g) 30: (t)(g)(a)(g)(g)(t)(a)(g)(a) 40: (t)(t)	(c), (c)(t), (c)(t)(t)
a	10: (g)(g)(a)(g)(a) 10: (g)(a) 20: (g)(a)(t)(a)(t)(g)(c)(t)(t)(a)(g)(a)(g) 20: (t)(a)(t)(g)(c)(t)(t)(a)(g)(a)(g) 20: (t)(g)(c)(t)(t)(a)(g)(a)(g) 20: (g)(a)(g) 20: (g) 30: (c)(t)(g)(a)(g)(g)(t)(a)(g)(a) 30: (g)(g)(t)(a)(g)(a) 30: (g)(a) 40: (t)(t)(g)(a)(g)(c)(t)(t) 40: (g)(c)(t)(t)	(a), (a)(t), (a)(g), (a)(g)(a), (a)(g)(g)

When *min_sup* is 0.5, complete sequential patterns of database *S* are shown in Table Ⅱ. The first column is prefix and the second column is the corresponding projected database of prefix. It is clear that, there are 24 complete sequential patterns, 4 length-1 patterns, 9 length-2 patterns, 7 length-3 patterns and 4 length-4 patterns.

When *min_sup*=0.5, Table Ⅲ shows the complete sequential patterns, maximal sequential patterns and closed sequential patterns of dataset *S*. It is clear that 24 complete patterns can be compressed into 5 maximal sequential patterns or 9 closed sequential patterns. Therefore, 9 closed patterns are more compressed than 24, and they contain the complete support information regarding to its corresponding frequent patterns. The maximal patterns are more compact than closed patterns, but it does not contain the complete information regarding to its corresponding frequent patterns. For example, the support of maximal pattern (g)(a)(g)(g) is 2 and support of (g)(a)(g) is 4. The support of these two pattern is different, but pattern (g)(a)(g) is compressed through mining maximal pattern.

TABLE III.
SEQUENTIAL PATTERNS AND SUPPORTS

sequential pattern	maximal pattern	closed pattern
(g), (g)(c), (g)(a), (g)(g),(g)(c)(t), (g)(a)(g),(g)(c)(t)(t), (g)(a)(g)(g)	(g)(c)(t)(t), (g)(a)(g)(g)	(g)(a)(g), (g)(c)(t)(t), (g)(a)(g)(g)
(t), (t)(a), (t)(t), (t)(g),(t)(a)(g), (t)(g)(a),(t)(a)(g)(a), (t)(g)(a)(g)	(t)(a)(g)(a), (t)(g)(a)(g)	(t)(g), (t)(a)(g)(a), (t)(g)(a)(g)
(c), (c)(t), (c)(t)(t),		(c)(t)
(a), (a)(t), (a)(g), (a)(g)(a), (a)(g)(g)	(a)(t)	(a)(t), (a)(g)(a)

III. THE NOVEL ALGORITHM

In this chapter, we provide a novel algorithm, called M-bBIDE (**M**ulti-support **BIDE** for **b**iological datasets). M-bBIDE is an improvement of the algorithm BIDE devoted to mining contiguous closed sequential patterns in biological data set. Meanwhile, it mines some interesting patterns based on multi-support. At first, we propose some definitions.

Definition 2 (Support) [7] The support of a subsequence *X* in a dataset *S* is the number of tuples in the dataset containing *X*, denoted as $support(X)=|\{<sequence\_id, s> | (<sequence\_id, s> \in S) \wedge (X \subseteq S)\}|$.

Definition 3 (Local Support) [3] The local support of a subsequence *X* in a dataset *S* is the number of tuples in a specific sequence *Y* containing *X*, denoted as $local\_support(X, Y)=|\{<location\_id, Y> | (Y \subseteq S) \wedge (X \subseteq Y)\}\}|$.

Definition 4 (Total Support) The total support of a subsequence *X* in a dataset *S* is the total number of tuples in *S*, denoted as $total\_support(X)=\sum_Y local\_support(X, Y)$.

Definition 5 (Local Sequential Pattern) [3] Local sequential pattern is a subsequence whose occurrence frequency in one specific sequence is no less than local minimum support (*local_min_sup*(*sequence_id*)).

Definition 6 (Total Sequential Pattern) Total sequential pattern is a subsequence whose occurrence frequency in dataset *S* is no less than total minimum support (*total_min_sup*).

There are three kinds of patterns in this paper. The first one is sequential pattern, which is a subsequence whose occurrence frequency in the set of sequences is no less than *min_sup* [7]. The second one is local sequential pattern, which is a subsequence whose occurrence frequency in one specific sequence (suppose $sequence\_id=s_i$) is no less than $local\_min\_sup(s_i)$. The third one is total sequential pattern, which is a subsequence whose occurrence frequency in dataset *S* is no less *total_min_sup*.

TABLE IV.
CLOSED SEQUENTIAL PATTERNS AND SUPPORTS

closed pattern	support	total support	location <*sequence_id, location_id*>
(t)(g)	3	3	<40, {2}> <30, {2}> <20, {5}>
(c)(t)	3	3	<40, {6}> <30, {1}>, <20, {7}>
(g)(a)(g)	4	5	<40, {3}> <30, {3}> <20, {11} <10, {0, 3}>
(a)(g)(a)	3	4	<30, {8}> <20, {0, 10}>

			<10, {4}>

For example, the database *S* is shown in Table Ⅰ and supposed the *min_sup* is 0.75. We get 4 closed sequential patterns as shown in the first column of Table Ⅳ. The first column is closed sequential pattern, denoted as *X*, the second column is *support*(*X*), the third column is *total_support*(*X*) and the last column is the location of *X*. The *support*((a)(g)(a))=3 means that it appears in 3 sequences: *sequence_id*={10, 20, 30}. While *total_support* ((a)(g)(a))=4 means it appears 4 times in dataset *S*. It appears 1 time in the location 4 of sequence 10(denoted as <*sequence_id*, *location_id*>=<10, {4}>); 2 times in sequence 20, <*sequence_id*, *location_id*>=<20, {0, 10}> and 1 time in sequence 30, <*sequence_id*, *location_id*>=<30, {8}>. If the *total_min_sup* number is 4, then we get 2 total sequential patterns: (g)(a)(g) and (a)(g)(a).

Table Ⅴ shows the frequent subsequences which appear in sequence 10. There are 2 subsequences, (g)(a)(g) and (a)(g)(a). Pattern (g)(a)(g) appears 2 times in *location_id*={0, 3}, therefore *local_support*((g)(a)(g), 10)=2. Pattern (a)(g)(a) appears 1 time in *location_id*={4} and the *local_support* ((a)(g)(a), 10)=1. If *local_min_sup*(10) is 2, then we get 1 local sequential pattern (g)(a)(g) in sequence 10.

TABLE V.

LOCAL CLOSED SEQUENTIAL PATTERNS IN SEUQNCE 10

local pattern	**local support**	**location <sequence_id, location_id>**
(g)(a)(g)	2	<10, {0, 3}>
(a)(g)(a)	1	<10, {4}>

Algorithm M-bBIDE is shown in algorithm 1. The input parameters are sequence database and three minimum supports. M-bBIDE is an improvement of BIDE algorithm, which change it more suitable for mining patterns in biological sequences and mining different interesting pattern based on multi-support.

Algorithm 1 (*M-bBIDE*)

Input: an input biological sequence databases *S*, minimum support thresholds

support_thresholds ={*min_sup*, *local_min_sup*, *total_min_sup*}

Output: the three sets of frequent closed sequences

Method 1:

Call *M-bBIDE*(*S*)

(1) Scan *S*, find length-1 frequent patterns *α*.

(2) For each *α* do

(2.1) Scan *S* again, find the location information <*α*, *sequence_id*, *transaction_id*> to *PrefixLocation*|*α*.

(2.2) *S*|*α*=pseudo projected database(*S*).

(2.3) If(!*ForwardScan*(*α*)), call *bide*(*α*, *S*|*α*, *PrefixLocation*|*α*, *support_thresholds*).

Method 2:

Call *bide*(*α*, *S*|*α*, *PrefixLocation*|*α*, *support_thresholds*)

(1) Scan *S*|*α* once, find each frequent item, *b*.

(2) For each *b*, append it to *α* to form a new prefix *α'*.

(2.1) According to *PrefixLocation*|*α*, find the location information <*α'*, *sequence_id*, *transaction_id*> to *PrefixLocation*|*α'*.

(2.2) *S*|*α'*=pseudo projected database(*S*|*α*).

(2.3) If(!*ForwardScan*(*α'*)), then call *bide*(*α'*, *S*|'*α*, *PrefixLocation*|*α'*, *support_thresholds*).

(2.4) If(!*ForwardScan* (*α'*)&&!*BackwardExtension*(*α'*))

Then output *α'* which satisfies the *support_ thresholds*.

In algorithm 1, the method *ForwardScan*() is used to check whether it exists item before prefix to meet the same support. Return true if we should stop to explore this prefix. The method *BackwardExtension*() return true if there is a backward-extension [5].

IV. PERFORMANCE EVALUATION

In this chapter, we provide experimental results to compare the performance of three algorithms: PrefixSpan, MM-PrefixSpan and M-bBIDE. PrefixSpan mines complete contiguous patterns by PrefixSpan algorithm, MM-PrefixSpan algorithm mines maximal contiguous patterns by PrefixSpan based on multi-support, and M-bBIDE mines closed contiguous patterns based on multi-support. In our performance study, we select 7 biological sequences, which meet the condition of "Homo sapiens" and "cancer" from NCBI. There are four kinds of cancers as shown in Table Ⅵ. The first 4 lines are sequence about colon cancer, line 5 is about stomach cancer, line 6 is about colorectal cancer and the last line is about ovarian cancer. The sequence lengths are shown in Table Ⅵ and the average length of these sequences is 2384.

TABLE VI.

BIOLOGICAL SEQUENCES

sequence_id	**sequence title**	**length of sequence**
U14658	Mutation in the DNA mismatch repair gene homologue hPMS2 is associated with hereditary nonpolyposis colon cancer	2697
U03911	The human mutator gene homolog MSH2 and its association with hereditary nonpolyposis colon cancer	3080
U07418	Mutation of a mutL homolog in hereditary colon cancer	2503
U07343	Mutation in the DNA mismatch repair gene homologue hMLH1 is associated with hereditary non-polyposis colon cancer	2484
U27467	A novel Bcl-2 related gene, Bfl-1, is overexpressed in stomach cancer and preferentially expressed in bone marrow	737
U04045	Mutations of a mutS homolog in hereditary nonpolyposis colorectal cancer	2947
U34880	A cDNA from the ovarian cancer critical region of deletion on chromosome 17p13.3	2234

All experiments were conducted on a 3.0GHz AMD PC with 2.0GB main memory, running Microsoft

Windows 7. Three algorithms, PrefixSpan, MM-PrefixSpan and M-bBIDE, were implemented by us using JDK 1.6 and Eclipse SDK 3.7.0.

First, we analyze the results about multi-support sequential patterns. When the *min_sup*=100% there are 623 complete patterns and 267 closed patterns. The distributions of frequent patterns are shown in figure 1. When the *min_sup*=50%, there are 3593 complete patterns and 2103 closed patterns, as shown in figure 2. It is clear that the number of closed patterns is much less than the number of complete patterns and. About 45% patterns are compressed.

When the *min_sup*=100%, we get 2 length-7 closed patterns: (c)(t)(c)(g)(t)(a)(g) and (c)(c)(g)(t)(t)(a)(a). Table Ⅶ shows that the supports and total supports of them are all 7. It means that these two patterns appear once in each sequence. The locations of them are shown in the last column of Table Ⅶ. Value <U14658, 224> means that pattern (c)(t)(c)(g)(t)(a)(g) appears in *location_id*={224} of U14658.

Supposed number of sequences meeting *min_sup* is 7 and number of subsequences meeting *total_min_sup* is 50, we can get the total sequential patterns as shown in Table Ⅷ. There are 7 total patterns, 5 lentgh-4 patterns and 2 length-5 patterns. It is clear that pattern (t)(a)(t)(t) appears 76 times in 7 biological sequences. In details, it appears 19 times in sequence U14658, 15 times in sequence U03911, 14 times in sequence U07418, 14 times in sequence U07343, 6 times in sequence U27467, 7 times in sequence U04045 and 1 time in U34880. Because (t)(a)(t)(t) appears 14 times in U07343 and U07418, the relationship between these two sequences may be more closer than with others.

From the results of M-bBIDE, we can also get the local patterns in one sequence. For example, when *min_sup*=100% and *local_min_sup*(U14658)=13, there are 8 local patterns in U14658 as shown in Table Ⅸ. Such as pattern (t)(a)(t)(t) appears 19 times and its locations are in *location_id*={354, 401, 772, 911, 1294, 1751, 1900, 1952, 1563, 2036, 2570, 2651, 2835, 2846, 2866, 2876, 2936, 2946, 2978}.

Figure 3 shows the processing time of the three algorithms: PrefixSpan, MM-PrefixSpan and M-bBIDE, at different support thresholds. The *min_sup* is from 0.5 to 1. It is clear that the runtime of M-bBIDE is more than PrefixSpan and MM-PrefixSpan. The reason is the consumption of backward and forward extension checks on every prefix.

The memory usage of the three algorithms at different support thresholds is shown in figure 4. It is clear that the memory usage of M-bBIDE is lower than others because of the pruning strategy. Figure 5 shows the number of sequence patterns of the three algorithms at different support thresholds. It is clear that mining maximal and closed sequential pattern compress the result of complete sequential patterns. The number of closed patterns is higher than maximal patterns. Closed pattern contains the complete support information regarding to its corresponding frequent patterns, while maximal pattern does not contain it. Therefore mining closed patterns meet the user's requirements.

The relation of complete frequent pattern to maximal frequent pattern and closed frequent pattern is shown in figure 6. The number of sequential pattern is larger than number of closed sequential pattern, and number of closed pattern is larger than number of maximal sequential pattern.

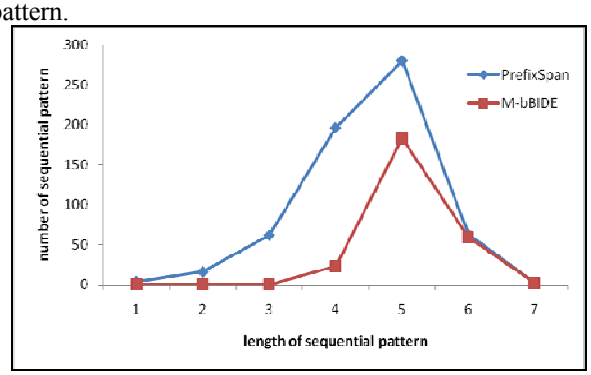

Figure 1. Numbers of frequent patterns when *min_sup* is 100%

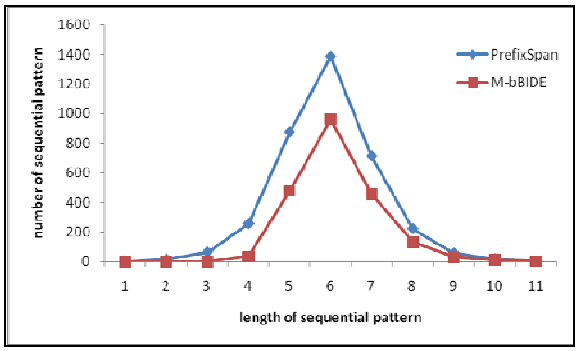

Figure 2. Numbers of frequent patterns when *min_sup* is 50%

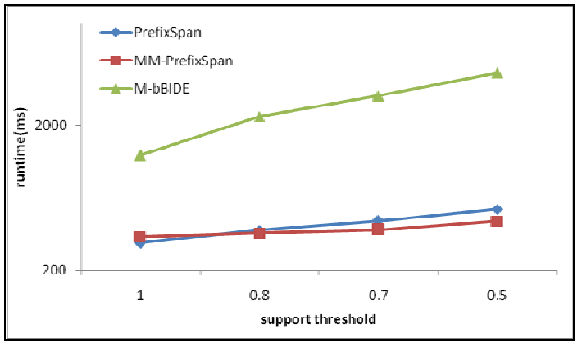

Figure 3. Runtime of three algorithms on biological sequences

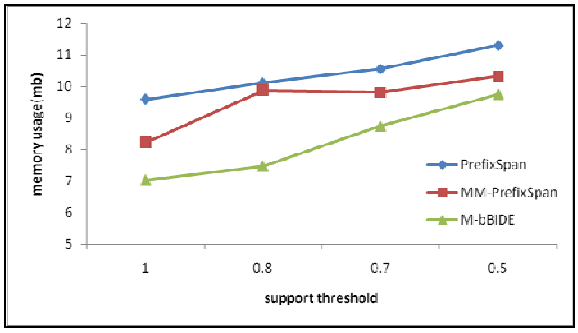

Figure 4. Memory usage of three algorithms on biological sequences

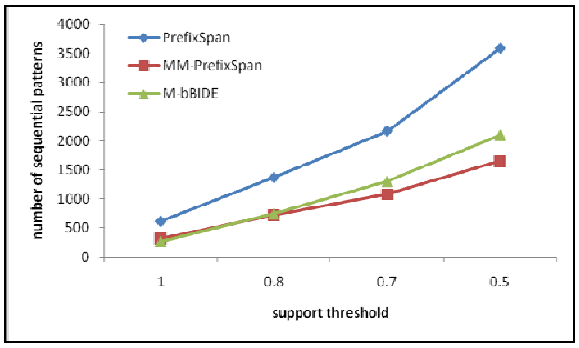

Figure 5. Number of frequent patterns on biological sequences

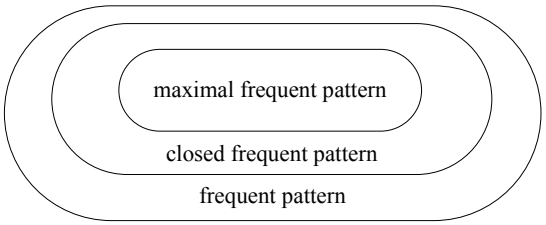

Figure 6. Relationship between sequence subsets

TABLE VII.

THE SUPPORTS AND LOCATIONS OF TWO LENGTH-7 PATTERNS

closed pattern	**support**	**total support**	**location** ***<sequence_id, transaction_id>***
(c)(t)(c)(g)(t)(a)(g)	7	7	<U14658, 224> <U03911, 289> <U07418, 1208> <U07343, 1228> <U27467, 906> <U04045, 81> <U34880, 1490>
(c)(c)(g)(t)(t)(a)(a)	7	7	<U14658, 1171> <U03911, 1236> <U07418, 1723> <U07343, 1743> <U27467, 2309> <U04045, 622> <U34880, 789>

TABLE VIII.

PATTERNS WHOSE TOTAL SUPPORTS ARE MORE THAN 50

closed pattern	**support**	**total support**
(g)(c)(c)(t)	7	72
(t)(a)(c)(t)	7	52
(t)(a)(t)(t)	7	76
(t)(t)(a)(t)	7	55
(a)(a)(a)(a)(a)	7	54
(a)(a)(t)(c)	7	62
(a)(g)(a)(t)(g)	7	50

TABLE IX.

PART OF LOACL PATTERNS IN SEQUENCE U14658

closed pattern	**local support**	**location** ***<sequence_id, transaction_id>***
(t)(a)(t)(t)	19	<U14658, {2036, 2651, 401, 2946, 2978, 911, 772, 1294, 1900, 2835, 1952, 1563, 354, 2866, 1751, 2936, 2876, 2846, 2570}>
(t)(t)(a)(c)	13	<U14658, {2002, 1128, 2950, 1565, 1203, 1321, 2797, 603, 1140, 2362, 2689, 1214, 1654}>
(t)(t)(a)(t)	13	<U14658, {713, 1252, 3053, 308, 2444, 1951, 2863, 2650, 2977, 1897, 1293, 840, 2873}>
(a)(a)(a)(a)(a)	13	<U14658, {685, 744, 1975, 684, 1611, 2985, 683, 2645, 2644, 2807, 1267, 1591, 1695}>
(a)(a)(t)(c)	16	<U14658, {821, 2810, 2777, 789, 1431, 2237, 2578, 381, 558, 2376, 1236, 259, 222, 2397, 2026, 3063}>
(a)(a)(g)(a)	13	<U14658, {1135, 1936, 955, 2224, 2319, 1978, 2970, 2932, 1207, 1473, 397, 1105, 271}>
(a)(t)(g)(a)	13	<U14658, {1939, 1711, 1399, 2773, 2825, 1681, 2356, 249, 2245, 900, 759, 1455, 1273}>
(a)(g)(t)(t)	13	<U14658, {305, 2475, 1304, 826, 2859, 865, 800, 2560, 3071, 1348, 1180, 391, 1555}>

V. CONCLUSION

Contiguous and long patterns are remarkable significance for biological data analysis. Although BIDE is a fast algorithm for mining closed sequential patterns, it not suitable for discovering patterns on biological data. Further considering the characters of biological sequence, an improvement of the BIDE is provided in this paper. This novel algorithm, called M-bBIDE is based on the multi-support to discover three types of contiguous closed pattern in biological datasets.

There are three kinds of patterns: sequential pattern, local sequential pattern and total sequential pattern. They correspond to three subsequence supports: *support*, *local_support* and *total_support*. The *support*(X) (X is a subsequence) is the number of tuples in the dataset containing X. If *support*(X) $\geqslant$ *min_sup*, then X is a sequential pattern. The *local_support*(X, Y) (Y is a biological sequence) is the number of tuples in sequence Y containing X. If *local_support*(X, Y)$\geqslant$*local_min_sup*(Y), then X is a local sequential pattern in sequence Y. The *total_support* (X) is the sum number of *local_support*(X, Y). If *total_support*(X)$\geqslant$*total_min_sup*, then X is a total sequential pattern.

There are many interesting issues that need to be studied, such as mining high dimensional sequential patterns with constraints [8-10], mining closed gapped subsequences [6, 11, 12], mining multiple patterns [13] and so on.

ACKNOWLEDGMENT

This work was supported by the National Nature Science Foundation of Ningxia (No. NZ12214), Fundamental Research Funds for the Central Universities (No. 2012YJS024), and the National Nature Science Foundation of China (No. 71061001).

REFERENCES

[1] Han, J. W., Cheng, H., Xin, D., and Yan, X. F.. Frequent pattern mining: current status and future directions. Data Mining and Knowledge Discovery, Vol. 15, 2007, pp. 55-86.

[2] Tang, K. M., Dai, C. Y., and Chen, L.. A novel strategy for mining frequent closed itemsets in data streams. Journal of Computers, Vol. 7, No. 7, 2012, pp. 1564-1572.

[3] Xiong, Y., and Zhu, Y. Y.. BioPM: An efficient algorithm for protein motif mining. In: Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering, 2007, pp. 394-397.

[4] Hsu, C. M., Chen, C. Y., and Liu, B. J.. WildSpan: Mining structured motifs from protein sequences. Algorithms for Molecular Biology, Vol. 6, Issue 1, 2011, pp. 1-16.

[5] Wang, J., and Han, J. et al. Frequent closed sequence mining without candidate maintenance. IEEE Transactions on Knowledge and Data Engineering, Volume 19, Issue 8, 2007, pp. 1042-1056.

[6] Chen, Y. C., Peng, W. C., and Lee, S. Y.. CEMiner – an efficient algorithm for mining closed patterns from time interval-based data. In: Proceedings of the 11th IEEE International Conference on Data Mining(Taiwan, ICDM2011), 2011, pp. 121-130.

[7] Pei, J., J. W., and Wang, J. Y. et al. Mining sequential patterns by pattern-growth: The prefixspan approach. IEEE Transactions On Knowledge And Data Engineering, Volume 16, 2004, pp. 1-17.

[8] Ferreira, P. G., and Azevedo, P. J.. Protein sequence pattern mining with constraints. Knowledge Discovery in Databases, Vol. 3721, 2005, pp. 96-107.

[9] He, D., Zhu, X. G., Wu, X. D.. Mining approximate repeating patterns from sequence data with gap constraints. Computational Intelligence, Vol. 27, Issue 3, 2011, pp. 336-362.

[10] Wang, K., Xu, Y., Yu, J . X.. Scalable sequential pattern mining for biological sequences. In: Proceedings of the thirteenth ACM international conference on Information and knowledge management(New York), 2004, pp. 178-187.

[11] Lavanya, B., and Murugan, A.. A DNA based approach to find closed repetitive gapped subsequences from a sequence database. International Journal of Computer Applications, Vol. 29, No 5, 2011, pp. 45-49.

[12] Fournier-Viger, P., Nkambou, R., and Nguifo, E. M.. A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems. In: Proceedings of the 7th Mexican International Conference on Artificial Intelligence, Vol. 5317, 2008, pp. 765-778.

[13] Yang, S. Y. , Chao, C. M. , and Chen, P. Z. . Incremental mining of across-streams sequential patterns in multiple data streams. Journal of Computers, Vol. 6, No. 3, 2011, pp. 449-457.

Meng Han is currently a candidate of Ph.D. student in Beijing Jiaotong University (Beijing, China), and also a lecturer in Beifang University of Nationalities (Yinchuan, China). Her research interests include data mining and machine learning.

Zhihai Wang received his PhD in Computer Science from Hefei University of Technology in 1998. He is now a professor in School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China. He has published dozens of papers in international conferences and journals. His research interest includes data mining and artificial intelligence.

Jidong Yuan is currently a candidate of Ph.D. student in Beijing Jiaotong University (Beijing, China). His research interests include machine learning and data mining.