Frequent Contiguous Pattern Mining Algorithms for Biological Data Sequences

S. Rajasekaran
R & D Centre
Bharathiar University
Coimbatore, India

L.Arockiam, Ph.D
Department of Computer Science
St. Joseph's College
Tiruchirapalli, India

ABSTRACT
Transaction sequences in market-basket analysis have large set of alphabets with small length, whereas bio-sequences have small set of alphabets of long length with gap. There is the difference in pattern finding algorithms of these two sequences. The chances of repeatedly occurring small patterns are high in bio-sequences than in the transaction sequences. These repeatedly occurring small patterns are called as Frequent Contiguous Patterns (FCP). The challenging task in pattern finding of bio-sequences is to find FCP. FCP gives clues for genetic discovery, functional analysis and also helps to assemble a whole genome of species. Most of the existing FCP algorithms are all based on Apriori method. They require repeated scanning of the database and large number of intermediate tables to produce the results. So, these algorithms require large space and high computational time. In this paper, we are analyzing few of the currently available FCP algorithms with their advantages and disadvantages.

General Terms
Bio-Sequences Pattern Mining, Bio-Sequences Algorithms

Keywords
Frequent Contiguous Pattern, Apriori, Scalable Pattern Mining, Surprising Bio-Patterns, Spanning Tree

1. INTRODUCTION
Frequent Patterns are patterns that are repeatedly occurring in a database. The adjacent repeated patterns are called as Frequent Contiguous Patterns. Frequent pattern mining helps to search the recurring relationships in a given data set. Frequent patterns are patterns of item sets, sub sequences or sub structures. Identifying patterns is helpful to find association, correlation and many relationships among data. It is an important overwhelming task and a good topic in research. Support and confidence are the two measures of interestingness. Patterns can be represented as association rules and the association rules are said to be strong if it satisfies both a minimum support threshold and a minimum confidence threshold. These threshold values are determined by the users or the field experts.

Let I = $\{i_1, i_2, \ldots i_n\}$ is an item set and D is the transaction database and T is a transaction associated with a transaction id. A & B are set of items and a transaction T contains A, if $A \subseteq T$.

The definition of support and confidence for the rule $A \Rightarrow B$ are

Support $(A \Rightarrow B) = P (A \cup B)$
Confidence $(A \Rightarrow B) = P (B/A)$
Confidence $(A \Rightarrow B) =$ Support $(A \cup B)$ / Support (A) [1]

The association rules are formed by Support and confidence [2]. Thus frequent pattern mining provides the solution for association rules formation. There are many kinds of frequent patterns mining, such as sequential patterns mining, structured patterns mining and scalable patterns mining.

The Sequential Pattern Mining involves frequent subsequences in a sequence data set. The Structured pattern mining is the most general form of frequent pattern mining. Structured pattern mining looks for frequent sub structures in a structured data set such as trees, graphs, sequences, sets of single items, or combination of above structures. Single item sets are the simplest structures in which each item element may contain recursive subsequences, sub trees or sub graphs.

The Apriori [3] based models such as GSP (Generalized Sequential Pattern) [4], SPADE (Sequential Pattern Discovery using Equivalent classes) [5] and PrefixSpan [6] are scalable pattern mining algorithms. These models are applicable to Transaction sequences in market-basket analysis. As web click stream sequences and bio-sequences are of long length with gaps, above models are not useful for analyzing these sequences. In web click streams, to predict next click gap is needed. Gaps in bio-sequences help to find approximate patterns for insertions, deletions and mutations. These types of sequences such as web click stream, bio-sequences can be viewed as constraint relaxation or enforcement.

1.1 Sequences patterns in biological data
Various computer algorithms and methods are developed to manage and analyze the huge volume of biological data. These algorithms help to compare and align biological sequences and predict bio-sequence patterns. DNA and Protein sequences consist of long linear chain of chemical components. DNA sequences contain four nucleotides namely Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) and protein sequences contains 20 amino acids. A gene sequence is a sequence of nucleotides arranged in a specific order. A genome is the complete set of genes of an organism.

All living organisms are related by evolution. So there exist more similarities in nucleotides and protein sequences of species. The process of lining up sequences is termed as sequence alignment, which helps to achieve highest level of similarities between the sequences. Alignment primarily identifies similar sequences with long conserved subsequences as between two or more biological sequences. If the two sequences share the common ancestor then they are homologous. The degree of similarity helps to find the possibility of homology between two sequences. Sequence similarity helps to determine the relative position of multiple species in an evolution tree called as phylogenetic tree.

A collection of data sequences is called sequence database (SDB). These data sequences contain repeatedly occurring of fixed number of data items. For instance a DNA SDB contains sequences made of repeatedly occurring four fixed

characters A, G, T and C only in any order. SDB applications require FCP to identify frequently occurring common patterns existing in the sequences of the SDB.

For instance to extract motif or regulatory regions from genomic SDBs, FCP is needed. In biological SDB applications, FCP helps to identify regulatory regions (a part of DNA sequences which are responsible for regulating the genes.

The rest of this paper is organized as follows. Section 2 analyses and discusses currently available methods to find FCP with their merits and demerits with example. Section 3 concludes with future direction of our research [7-8].

2. VARIOUS METHODS FOR FREQUENT CONTIGUOUS PATTERNS

2.1 SP-Index (Segment to Position Index) algorithm

SP-Index algorithm [9] finds FCP in two phases. The first phase is called as segment phase and the second phase is called as pattern phase. This method considers bio-sequence pattern is of the form $X_1 * X_2 \ldots X_k$ where X_i is a short region of consecutive items and "*" denotes the gap length. [10-11]

Initially, segment phase finds all the base segment patterns which satisfy minimum support. Subsequently, pattern phase form a root directory for the base patterns. Then construct the SP-Trees, which connect base patterns in root directory with its consecutive positions.

For Example:

Table 1. A sample Biological DB

SequenceID	Sequence
10	AGATCAG
20	AGTATCA
30	GAATCTA

Table 2. Phase I: Segment Phase

Base Segments	Position Lists
B1: AG	(10:1,6) (20:1)
B2: AT	(10:3), (20:4), (30:3)
B3: TC	(10:4), (20:5), (30:4)
B4: CA	(10:5), (20:6)

Segment phase creates a root directory consist of base patterns, B1=AG, B2=AT, B3=TC and B4=CA which satisfy minimum support, min_sup = 2⁄3 and minimum length, min_len = 2. The root directory contains these base patterns with their sequenceID and positionID. The base pattern B1=AG occurs at sequenceID "10" at the position "1" and "6" and sequenceID "20" at the positon "1". Similarly, sequenceID and positionID of the base patterns B2, B3 and B4 are provided in Table 2.

Root Directory

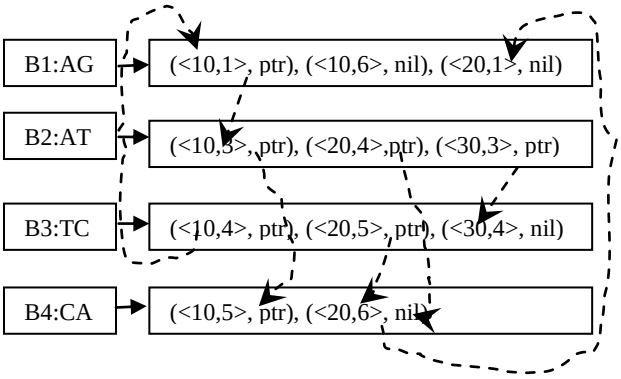


Fig 1: Pattern Phase - SP-Index Tree

Pattern phase construct SP-Tree which connects base patterns B1, B2, B3 and B4 in root directory with its consecutive positions. Here, sequenceID "10" contain "AT" at 3rd & 4th position and "TC" at 4th & 5th, combine these two produce a new pattern of length 4 "ATCA" .

2.1.1 Advantages and disadvantages:

This method avoids the repeated scanning of the DB. The segment phase produces a position list for base segment which reduces the searching space for patterns. This method didn't generate all possible patterns instead it finds existing pattern in DB and generates the remaining patterns based on these existing base patterns. Thus this method reduces time required for pattern finding and searching space for patterns. It also quickly generates the new patterns.

The disadvantage of this method is the construction of root directory for all existing base patterns and the creation of SP – Trees needs high time and space complexity. Similarly, the traversal of whole SP-Trees to find the patterns also needs high time and space complexity.

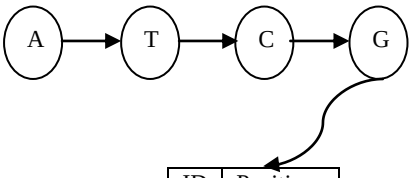
2.2 Surprising contiguous pattern mining algorithm

This algorithm [12] describes about interesting/surprising patterns which are not frequent, may still be informative in computational biology and bioinformatics. This algorithm gave new measurements called "minimum information gain threshold" and "minimum confidence threshold" [13] which are based on the probability occurrence of characters in the database.

This algorithm contains two steps. In the first step, it generates the Index based spanning tree for fixed pattern length by searching the database. The leaf node contains the sequenceID and respective sequence positions of the pattern in the spanning tree. [14-16]

Table 3. A sample Biological DB

SequenceID	Sequence
10	ATCGGCGTGATCG
20	GATCGCCTATCG
30	CTCTCATTG



ID	Positions
10	1,10
20	2,9

Fig 2: Index based spanning tree for the patterns ATCG

In the second step the leaf node of the spanning tree is checked with the threshold measurement and produces the interesting pattern.

2.2.1 Advantages and Disadvantages:

This algorithm gives a new view to pattern discovery and its measurement. This algorithm avoids repeated scanning of DB. The search space for the pattern is also reduced with respect to the "information gain threshold" and "confidence threshold".

The construction of spanning tree for fixed length needs large space and high time complexity. This algorithm also has repeated spanning tree traversal to find the interesting patterns with respect to "information gain threshold" and "confidence threshold" which increases time complexity.

2.3 Location based FCP

The algorithms to find FCP like Apriori, GSP etc. require repeated scanning of DB and large number of intermediate results. The approach adopted by Location based FCP algorithm [17] uses an intelligent way of sorting and joining techniques than the existing algorithms which improves the time complexity. This algorithm contains three steps namely (1) Finding fixed length pattern table with position information (2) Sort the patterns in the position table (3) Joining the patterns by Apriori rule and generating the new maximum length patterns.

The sorting technique used by this method is based on its last occurring position which is a less time consuming process than the alphabetical order sort by the traditional approaches. This type of sort also expedites pattern joining process. Pattern joining process produces next higher length pattern. This algorithm applies Apriori rule to join length-2 patterns and produces higher length patterns.

Table 4. Sample Biological DB

SequenceID	Sequence
50	GAGTGCTTAATCG

Table 5. Pattern table with position information

ID	Pattern	Position_Information Before Sorting (Seq_ID, Start Position)	Pattern	Position_Information After Sorting (Seq_ID, Start Position)
1	GAGT	(50,1)	ATCG	(50,10)
2	AGTG	(50,2)	AATC	(50,9)
3	GTGC	(50,3)	TAAT	(50,8)
4	TGCT	(50.4)	TTAA	(50,7)
5	GCTT	(50,5)	CTTA	(50,6)
6	CTTA	(50,6)	GCTT	(50,5)
7	TTAA	(50,7)	TGCT	(50,4)
8	TAAT	(50,8)	GTGC	(50,3)
9	AATC	(50,9)	AGTG	(50,2)
10	ATCG	(50,10)	GATC	(50,1)

By combining the positions (50, 9), (50, 10), we get the newly generated pattern of length 5 is AATCG.

2.3.1 Advantages and Disadvantages:

The sorting and joining can produce new patterns quickly than the traditional methods. Repeated scanning of DB is also not required.

But the intermediate tables for sorting and joining require large memory spaces. So, it can be further optimized by using some other parameters and techniques.

2.4 Fast Contiguous FCP using Position Information

This method [18] contains three steps to find FCP, namely

(i) Generates fixed length spanning tree with sequenceID and position information
(ii) Creates the hash table for patterns starting with A,T,C and G with their start-index and End-Index positions
(iii) Performs joining operation to produce higher length patterns by using Binary search and hash table

Table 6. Sample Biological DB

SequenceID	Sequence
10	CTGCGCTGTTCAC
20	TCGATCCTTCTGC
30	GATCGATGCTAC
40	CCTTAGTCTATCGAGTGCTA
50	GATCCTTAGTGCG

Table 7. Length-4 Subsequences from Spanning Tree

Index	Pattern	Position
1	ATCC	(20:4), (50:2)
2	ATCG	(30:2), (40:10)
3	AGTG	(40:14), (50:8)
4	TGCG	(10:2), (50:10)
5	TGCT	(30:7), (40:16)
6	TCCT	(20:5), (50:3)
7	TCGA	(20:1), (30:3)

Table 8. Length-4 Subsequences Alphabetical Order

Index	Pattern	Position
1	AGTG	(40:14), (50:8)
2	ATCC	(20:4), (50,2)
3	ATCG	(30:2), (40:10)
4	TCCT	(20:5), (50:3)
5	TCGA	(20:1), (30:3)
6	TGCG	(10:2), (50:10)
7	TGCT	(30:7), (40,16)

Table 9. Hash Table

Letter	Start_Index	End_Index
A	1	3
T	4	7

This method first generates the spanning tree of fixed length 4 with minimum support 2, shown in Fig.3 (for A & T only). Then length-4 pattern with pattern position table is formed by spanning tree traversal, as shown in Table 7. This pattern position table should be arranged in alphabetical order, so that binary search can be applied for joining process, which is shown in Table 8. Then a hash table is created for the letters A,T,G,C with their start_index and End_index position shown in Table 9 (for A & T only).

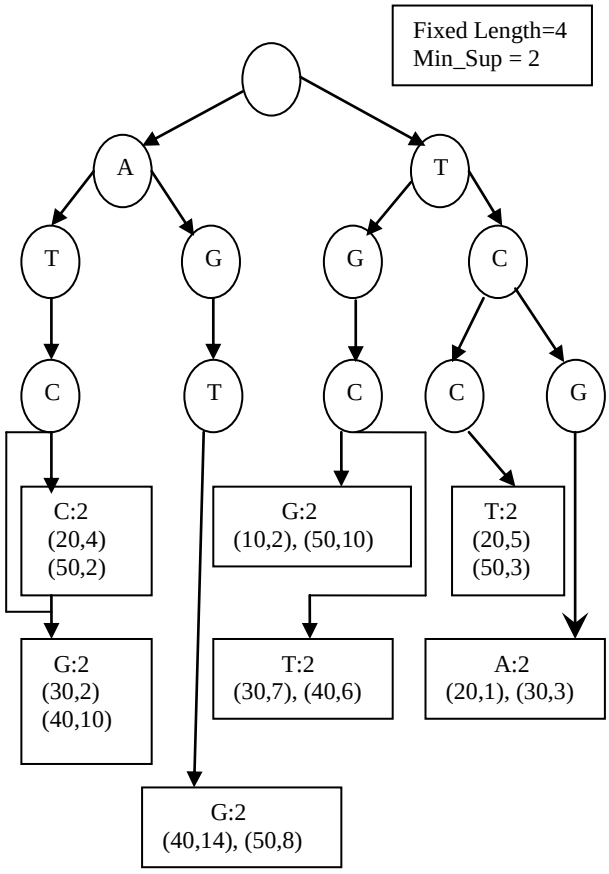


Fig 3: Spanning Tree for A and T

The joining process to find length-5 pattern is as follows: for example, the length-5 pattern for ATCC may be in one of the form like ATCCA, ATCCG, ATCCT, and ATCCC. As per hash table 9, the second letter of ATCC, which is "T" starts with 4th index and ends with 7th Index. Now by applying binary search to check ATCCT, we need to compare only index-4 & index-5. As there is match occurs at index-4, which is selected. Now compare the positions of ATCC, which is at (20, 4) with the positions of TCCT which is at (20:5) are adjacent. Hence the length-5 pattern ATCCT occurs at (20:4) is generated.

2.4.1 Advantages and disadvantages

The advantage of this method is applying binary search for pattern joining which reduces the computational time and also this method avoids the repeated scanning of the DB.

The disadvantage of this method is that the intermediate tables must be arranged in alphabetical order before applying binary search, which is a time consuming process. Also, the intermediate tables require large memory space.

2.5 Apriori Algorithm

This algorithm is widely used FCP algorithm. This algorithm forms the base for most of subsequent modern FCP algorithms. This is a level wise search algorithm based on prior knowledge of frequent sequences. In this algorithm, to explore (k+1) subset prior k subsets are used. This algorithm initially scans the database and creates 1-item frequent dataset called L_1, in which the items must satisfy the minimum support. Subsequently, 2-item dataset L_2 is created with the help of L_1, L_3 is created by L_2 and so on, i.e. L_{k+1} is created by L_k. Each L_k's require complete scan of the database.

Apriori property: "All non-empty subsets of a frequent item set must also be frequent".

The Apriori property helps to reduce the search space. By the above definition, if an item set "I" does not pass the minimum support threshold min-sup then "I" is not frequent. If an item "A" is added with "I", i.e. (I U A) is not frequent either. This property is called as "anti-monotone", means that down-ward closed i.e. if a set can't pass a test, all of its supersets will fail the same test as well.

Apriori algorithm follows a two steps process called Join & Prune.

1. Join Step

The set L_k is formed by joining L_{k-1} by itself, i.e. ($L_{k-1} \bowtie L_{k-1}$). In this step the item set must be in lexicographic order and the duplications should be avoided.

2. The Prune step

The items in L_k must satisfy the minimum support threshold which is determined by scanning of the database. To do this, the super set of L_k is formed by including all frequent "k" item sets and thus reduces the table size by comparing with minimum support threshold and by scanning the database.

Table 10. Sample Biological DB

SequenceID	Sequence
100	A C -
200	T C G
300	A T C G
400	T G

Table 11. C_1: 1-item set of DB

TID	Set of Itemsets
100	{ {A} {C} {-}}
200	{ {T} {C} {G}}
300	{{A} {T} {C} {G}}
400	{{T} {G}}

Table 12. L_1: 1-item set with min_sup:2

Itemset	Support
{A}	2
{T}	3
{C}	3
{G}	3

Table 13. $\bar{C}_2$: 2-item set – $L_1 \bowtie L_1$

Itemset
{A,T}
{A,C}
{A,G}
{T,C}
{T,G}
{C,G}

Table 14. C_2: 2-item set of the DB

TID	Set of Itemsets
100	{{A, C}, {C, -}}
200	{{ T, C}, {C, G}, {T, G}}
300	{{A,T},{A,C},{A,G},{T,C},{T,G},{C,G}}
400	{{T,G}}

Table 15. L_2: 2-item set of the DB with min_sup:2

TID	Itemset	Support
100	{A,C}	2
200	{T,G}	3
300	{T,C}	2
400	{C,G}	2

Table 16. $\bar{C}_3$: 3-item set – $L_2 \bowtie L_2$

Itemset
{T,C,G}
{A,C,G}

Table 17. C_2: 3-item set of the DB

TID	Set of Itemsets
100	{{A, C, -}}
200	{{ T, C, G}}
300	{{A,T, C},{A,,T, G},{T,C,G}}

Table 18. C_3: 3-item set of the DB with min_sup:2

Itemset	Support
{T C G}	2

Initially, in the prune step of Apriori method, scans the DB and finds 1-itemset as shown in Table 11. Subsequently, L1:1-itemset is created which satisfies min_sup:2 from 1-itemset, as shown in Table 12.

The joining step performs L1 ⋈ L1 to produce $\bar{C}_2$:2-itemset for all possible 2-items in the DB without duplication as shown in Table 13. The above steps are repeated to find the FCP as shown in Table 14 to Table 18.

2.5.1 Advantage and Disadvantage of Apriori Method

This method follows a recursive step to produce accurate results. Apriroi property reduces the size of searching space considerably by including min_sup.

This method considers all possible frequent sets which is huge and involves heavy computations. This method considers all possible combinations of item sets in each level which require huge searching space, in which some of the item sets may not be available in the database itself. This method scans the DB multiple times to form 1-itemset, 2-itemset etc. Thus the time and space complexity are high for this method.

3. CONCLUSION AND FUTURE RESEARCH DIRECTION

SP-Index method scans the DB for the base patterns existing in the DB, which are present in root directory. Then with the help SP-Index trees, remaining next level patterns are generated. Surprising contiguous pattern algorithm creates spanning tree for all base patterns existing in the DB by scanning. Then, searching space is reduced with help of new measures namely "minimum information gain threshold" and "minimum confidence threshold". Location based FCP generates pattern table with patterns and their locations for all existing patterns in the DB by scanning the DB. Then sort the pattern table by last occurring position and joins them Apriori rule. Fast Contiguous FCP using Position Information also creates a spanning tree for base patterns in the DB by scanning and reduces the search space by using hash table and reduces the search time by using binary search. Apriori algorithm repeatedly scans the DB for the base pattern and compares the base pattern with the possible pattern table to produce FCP.

Apriori, SPADE, PrefixSpan algorithms are applicable to transaction sequences in market-basket analysis. As bio-sequences are fixed-items with long sequences having gaps, these algorithms are not applicable to bio-sequences.

Table 19. Comparative analysis of existing techniques

Existing Techniques	**Uniqueness**	**Issues**
SP-Index	Quickly generates the new patterns with the help of existing base pattern and its position.	Construction of root directory for the base pattern and Creation of SP – Trees needs high time and space complexity
Surprising Contiguous Pattern Mining	New measurements namely "minimum information gain threshold" and "minimum confidence threshold" are defined for support and confidence	Spanning Tree construction and repeated spanning tree traversal increases time and space complexity

Location based FCP	An intelligent way for Sorting and joining of intermediate tables to quickly produce new patterns	Large memory space needed for Intermediate tables
Fast Contiguous FCP using Position Information	Hash tables help to reduce searching space for binary search to produce new pattern	To apply binary search, all the intermediate tables must be arranged in alphabetical order which is a time consuming process
Apriori	Produces accurate results by considering all possible frequent sets	Considers all possible frequent sets in which some of the items may not be available in the database, which increases space and time complexity

All the above algorithms either generate spanning tree for the base patterns or create tables with location information for base patterns in the DB. They optimize sorting and joining methods to reduce the search space for the patterns. As biological DB contains only a fixed number of items, (for example, DNA sequence contains only A,G,T and C), by creating sub databases for these items alone and by applying heuristic approach may improve time and space complexity.

Future research direction is to develop a new algorithm which divides the SDB into sub SDBs using hashing technique. And search the required FCP in its possible sub SDBs. This algorithm will map out hash_id of required FCP from possible 2^k subsets hash_ids. Then matched hash_ids' strings are compared with the required FCP. If matching occurs then required FCP is found otherwise there is no chance of required FCP occurring over the SDB.

The advantage of this algorithm is that there is no need to scan the entire SDB for every time to find a FCP. All possible FCPs can be found from the Hash_ids set which reduce the searching time to the linear search time.

4. REFERENCES

[1] Han J., Kamber M. 2006. Data Mining: Concepts and Techniques. Elsevier, 2nd Edition, pp 230, 2006.

[2] Agrawal R., Imielinski T and Swami A. 1993. Mining association rules between sets of items in large databases. Proceedings of the ACM SIGMOD conference on Management of Data, pp 207-216, Washington DC, May 1993.

[3] Agrawal R, Srikant R. 1994. Fast Algorithms for Mining Association Rules. Proceedings of the 20th VLDB conference, Santiago.

[4] Srikant R., Agrawal R. 1996. Mining sequential patterns: Generalizations and performance improvements. 5th International Conference on Extending Database Technology, Avignon, France.

[5] Zaki M.J. 2001. SPADE: An Efficient Algorithm for Mining Frequent Sequences. Machine Learning Journal, Special Issue on Unsupervised Learning, Vol. 42, No. ½, pp 31-60, 2001.

[6] Pei J., Han J., Asl B., Chen Q., Dayal U. and Hsu M. 2001. Prefixspan: mining sequential patterns efficiently by prefix-projected pattern growth, ICDE, 2001.

[7] Hirschberg DS. 1977. Algorithms for the longest common subsequences problem. Journal of the Association for Computing Machinery. Vol 24. No 4. October 1977. pp 664-675.

[8] Huo H, Stojkovic V. 2007. A suffix tree construction algorithm for DNA sequences. Proceeding of IEEE International conference on Bioinformatics and Bioengineering (BIBE'07), 2007, Oct 14-17, Boston, MA, pp 1178-1182.

[9] Wang K., Xu Y. and Yu J. X. 2004. Scalable Sequential Pattern Mining for Biological Sequences. CIKM'04. Proceedings of the thirteenth ACM international conference on Information and knowledge management. Pages 178-187.

[10] Yang J., Wang W., Yu P.S. and Han J. 2002. Mining long sequential patterns in a noisy environment. SIGMOD, 2002.

[11] Brazma A., Jonassen I., Eidhammer I. and Gilbert D. 1995. Approaches to the automatic discovery of patterns in biosequences. Technical report, Departmetn of Informatics, University of Bergen, Norway, 1995.

[12] Rashid Md. M., Karim Md. Rezaul, Jeong B. and Choi H. 2012. Efficient Mining of Interesting Patterns in Large Biological Sequences. Genomics & Informatics. Vol 10 (1) 44-50.

[13] Blahut R. 1987. Principles and Practice of Information Theory. Addison-Wesley Longman Publishing Co., Inc. Boston, MA, 1987.

[14] Kang T.H., Yoo J.S. and Kim H.Y. 2008. Mining frequent contiguous sequence patterns in biological sequences. Proceedings of 7th IEEE International Conference on Bioinformatics and Bioengineering (BIBE'08), Athens, Oct 8-10, 2008, pp 723-728

[15] Zerin SF, Ahmed CF, Tanbeer SK, Jeong BS. 2010. A fast indexed based contiguous sequential pattern mining technique in biological data sequences. In Proceedings of 2nd International Conference on Emerging Databases (EBD'10), Jeju

[16] Karim Md. R., Rashid Md. M., Jeong B.S. and Choi H. J. 2012. An Efficient Approach to Mining Maximal Contiguous Frequent Patterns from Large DNA Sequence Databases. Genomics & Informatics. Vol 10(1) 51-57, March 2012.

[17] Tanvee M. M., Kabeer S. J., Chowdhury T. M., Sarja A. A. and Shuvo Md. T. H. 2013. Mining Maximal Adjacent Frequent Patterns from DNA Sequences using Location Information. International Journal of Computer Applications. Vol. 76 – No. 15.

[18] Zerin S. F. and Jeong B. S. 2011. A Fast Contiguous Sequential Pattern Mining Technique in DNA Sequences Using Position Information. IETE Technical Review. Vol. 28 – Issue 6.