International Journal of Scientific and Technical Advancements
IJSTA
ISSN: 2454-1532

# Efficiency of Using Sequence Discovery for Polymorphism in DNA Sequence

S. Kalaiselvi[1], A. Meena[2]
[1, 2]Computer Science, Dr. SNS. Rajalakshmi CAS, Coimbatore, Tamil Nadu, India-641049
Email address: [1]kulandaikalai@gmail.com,[2]mithra1710@gmail.com

***Abstract-*** DNA-deoxyribonucleic acid, sequencing provide method for efficient large-scale discovery of markers for use in Human Genes. Discovery options include large-scale gene-enriched genome sequencing and whole genome sequencing. Sequencing of the transcriptome of genotypes varying for identify genes with patterns of expression that could explain the phenotypic variation. This thesis presents a effective sequence pattern discovery to find polymorphic motifs in human DNA sequence to detect hidden genetic medical conditions using three structure LDK Algorithms, HVS and MDFS. To find out the disease of Ebola sequence pattern matching with polymorphism DNA sequence. The position scrolling matrix used for find report of test result of sequence pattern.

***Keywords***—LDK-length distance K-support; HVS-hierarchical verification; MDFS-matching depth first search.

## I. INTRODUCTION

### A. *Overview of Data Mining*

Data mining means "mining" knowledge from large data set. It is defined as "the process of discovering meaningful new inter relationship, patterns and mode by finding into large amounts of data stored in a data set".

*a)* *Sequence Mining*: Sequence Mining means finding sequential patterns among the large dataset. It finds out frequent substring as patterns from a dataset. With massive amounts of data continuously being gathered, many industries are becoming interested in digging sequential patterns from their databases.

*b)* *The goal of sequential data mining*:The approach is to discover frequently occurring patterns but not identical. The challenge in discovering such patterns is to allow for some noise in the matching process. To find such a method first is to find the definition of a pattern, and then definition of similarity between two patterns. This similarity definition of the two patterns can vary from one application to another.

*c)* *DNA sequence mining*: DNA sequence is an important mean to study the structure and function of the DNA sequence. In this thesis, based on the characteristics of the DNA sequence an algorithm will proposed which uses the maximal frequent pattern segments based on adjacent maximal frequent. DNA sequences use an alphabet {A, C, G, T} representing the four nitrogenous bases Adenine, Cytosine, Guanine and Thymine.

DNA sequencing, technique used to determine the nucleotide sequence of DNA (deoxyribonucleic acid).

### B. *Human DNA*

Every human has his / her unique genes. Genes are made up of DNA and therefore the DNA sequence of each human is unique. However the DNA sequences of all humans are 99.9% identical, which means there is only 0.1% difference. It contains genetic instructions of an organism and is mainly composed of nucleotides of four types. Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). The pattern itself may not be exactly known, because it may involve insertion, deletion, or replacement of the symbols.

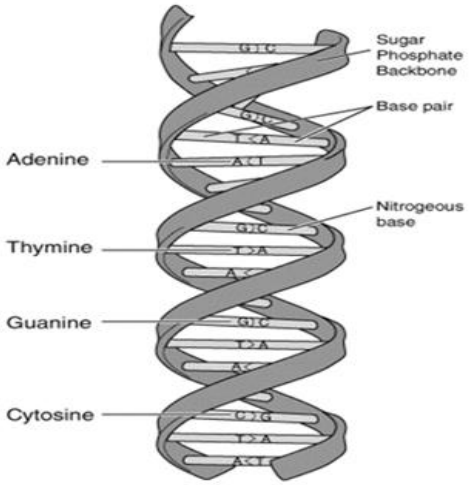


Fig. 1. Structure of the DNA.

When we know a particular sequence is the cause for a disease, the trace of the sequence in the DNA and the number of occurrences of the sequence defines the intensity of the disease. As the DNA is a large database we need an efficient algorithm to find out a particular sequence in the given DNA. We have to find the number of repetitions and the start index and end index of the sequence, which can be used for the diagnosis of the disease and also the intensity of the disease by counting the number of pattern matching strings, occurred in a gene database.

### C. *Biological Motivation for Pattern Discovery*

Nucleotide and protein sequences contain patterns or motifs that have been preserved through evolution because



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.

they are important to the structure or function of the molecule. In proteins, these conserved sequences may be involved in the binding of the protein to its substrate or to another protein, may comprise the active site of an enzyme or may determine the three dimensional structure of the protein. Discovery of motifs in protein and nucleotide sequences can lead to Determine of function and to elucidation of evolution relationship among sequences.

### D. Objective of the Research

The main aim of this system is to Discovering frequent approximate sequences selecting randomly one subsequence of length W from each input sequences. This thesis proposed a method for model (L, d, k) and tried to improve the performance of existing algorithm. Some of the existing methods Hierarchical Verification Algorithm (HVA) and Matching with Depth First Search (MDFS).The number of comparisons per character (CPC ratio) which is equal to (Number of comparisons/file size) can be used as a measurement factor. The main objective of this this work is to identify the motif sequence pattern of Ebola virus infected patients DNA Sequence to determine the immunity resistant.

### E. Scope of the Research

The Scope of the system is identify a genotyping platform and molecular marker that are amenable to high-throughput genotyping for EBOLA Virus. This approach explores alternative approaches for virus infection identification, using single nucleotide polymorphism (SNP) molecular markers and high throughput genotyping platforms which reduce the cost per data point. SNPs are unambiguous, genome-wide molecular markers that allow identification and traceability through-out the production chain.

### F. Problem Definition

Sequential pattern mining deals with data in large sequential data sets. The results of pattern mining can be used for planning and prediction. The difficulty in discovering frequent patterns is to allow for some noise in the matching process. The most important part of pattern discovery is the definition of a pattern and similarity between two patterns which may vary from one application to another. Discovering frequent patterns is computationally expensive process and counting the instances of pattern requires a large amount of processing time. These algorithms differ in their ways of traversing the item set lattice, dimension and the way in which they handle the database; i.e., how many passes they make over the entire database and how they reduce the size of the processed database in each passes.

## II. Literature Review

### A. Medical Research

Most current successful applications of data mining in Health Informatics are in the subfield of medical research. The reason is that most of the current health related data are stored in small datasets scattered through various clinics, hospitals, and research centers. However, most applications of data mining in clinical and administrative decision support systems require homogeneous and centralized data. On the other hand, data mining methods can still be successfully applied on small and scattered datasets, and help researchers extract insightful patterns, cause and effect relationships, and predictive scoring systems from currently available data.

### B. Association Rules and Decision Trees

Association rules and decision trees for disease prediction Ordonez applies different classifiers, associative classifier and decision trees, for predicting the percentage of vessel narrowing (LDA, RCA, LCX and LM) compare to a healthy artery [15]. The dataset contains 655 patient records with 25 medical attributes. Three main issues about mining associative rules in medical datasets are mentioned in this work. A significant fraction of association rules are irrelevant and most relevant rules with high quality metrics appear only at low support. On the other hand, the number of discovered rules becomes extremely large at low support. Hence, association rules are used with constraints. Each item corresponds to the presence or absence of one categorical value or one numeric interval. First constraint is that there is a limit on the maximum item-set size. Second, the items are grouped and in each association, there is at most one from each group. The third constraint is that each item can only appear in antecedent or consequent. The result from associative classifier is compared with two decision tree algorithms CN4.5 and CART. The authors demonstrate that associative rules can do better than decision trees for predicting diseased arteries.

### C. Apriori-Based Algorithms

The Apriori and Apriori All set the basis for a breed of algorithms that depend largely on the Apriori property and use the Apriori-generate join procedure to generate candidate sequences. The Apriori property states that All non-empty subsets of a frequent item set must also be frequent. It is also described as anti-monotonic, in that if a sequence cannot pass of these candidates. Frequent sequence produced from these candidates are captured, while those candidates without minimum support are removed .This procedure is repeated until all the candidates have been counted. In the first step GSP algorithm [2] finds all the length-1 candidates (using one database scan) and orders them with respect to their support ignoring ones for which support (<min_sup). Then for each level, this algorithm scans datasets to collect support count for each candidate sequence and generates candidate length (k+1) sequences from length-k frequent sequences using Apriori algorithm. This is repeated until frequent sequence or no candidate can be found.

### D. GSP

The GSP algorithm describe by Agrawal and Shrikant [1] makes multiple passes over the data. The algorithm not a main memory algorithms. If the candidates do not fit memory, the algorithm generate only the many candidates as will fit in memory and the date is scanned to count the support of these candidates. Frequent sequence produced from these candidates are captured, while those candidates without Frequent sequence produced from these candidates are captured, while



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.

those candidates without minimum support are removed. This procedure is repeated until all the candidates have been counted. In the first step GSP algorithm [2] finds all the length-1 candidates (using one database scan) and orders them with respect to their support the minimum support threshold, its entire super sequences will never pass the test.

### *E. Spade*

Besides the horizontal formatting method (GSP) [2], the sequence database can be transformed into a vertical format consisting of items' id-lists., is a list of (sequence-id, time stamp) pairs indicating the occurring time stamps of the item in that sequence. Searching in the lattice of the datasets formed by id-list intersections, the SPADE [3] (Sequential Pattern Discovery using Equivalence classes) algorithm presented by M.J.Jaki completes the mining in three passes of scanning the databases. However, additional computation time is required to transform this database of horizontal layout to vertical layout, which also requires additional storage space which is larger than that of the original sequence database.

### *F. Pattern Growth Based Algorithms*

In pattern growth algorithm partitioning of search space feature plays a major role. In each pattern growth algorithm working starts by representing the database sequence and then provides the way to partition the database search space and thus producing as less candidate patterns as possible by growing on the already mined frequent sequences, and applying the Apriori algorithm as the search space is being traversed recursively looking for frequent sequences.

(a) *Search space partitioning*: It allows the generated search space having candidate sequence to be partitioned to make the efficient use of memory. Several ways are available for the partitioning of the search space. Among which first is to partition the search space in smaller blocks in parallel. Modern techniques include projected and conditional search.

(b) *Tree projection*: In this algorithms implement a physical tree data structure representation of the search space, which searches the frequent patterns either using breadth first or depth first search and an Apriori algorithm is used for minimizing the sequence length.

(c) *Depth-first traversal*: That depth-first search of the search space makes a big difference in performance, and also helps in the early pruning of candidate sequences as well as mining of closed sequences. The fact that depth-first traversal used is that utilizes less memory, more directed search space, and thus less candidate sequence generation than breadth-first algorithms.

### *G. Application of Data Mining Techniques In CRM*

Data mining technique is used in CRM .Now a days it is one of the hot topic to research in the industry because CRM have attracted both the practitioners and academics. It aims to give a research summary on the application of data mining in the CRM domain and techniques, which are most often used. Research on the application of data mining in CRM will increase significantly in the future based on past publication rates and the increasing interest in the area.

## III. RESEARCH METHODOLOGY

### *A. DNA Input Sequences Representation*

The input of pattern discovery programs usually consists of several sequences, expected to contain the pattern. We will denote Σ the alphabet of all possible characters occurring in the sequences. Thus Σ = {A,C,G,T} for DNA sequences and Σ is a set of all 20 amino acids for protein sequences.

- *Ambiguous character* is a character corresponding to a subset of Σ. Ambiguous character then matches any character from this set. Such sets are usually denoted by a list of its members enclosed in square brackets e.g. [LF] is a set containing L and F. A-[LF]-G is a pattern in a notation used in EBOLA database. This patterns matches 3-character subsequences starting with A, ending with G and having either L or F in the middle. For nucleotide sequence there is a special letter for each set of nucleotides, where R=[AG], Y=[CT], W=[AT], S=[GC], B=[CGT], D=[AGT], H=[ACT], V=[ACG], N=[ACGT].
- *Wild-card* or don't care is a special kind of ambiguous character that matches any character from Σ. Wild-cards are denoted N in nucleotide sequences, X in protein sequences. Often they are also denoted by dot **'.'**. Sequence of one or several consecutive wild-cards is called gap and patterns allowing wild-cards are often called gapped patterns.
- *Flexible gap* is a gap of variable length. In EBOLA database it is denoted by x(i,j) where i is the lower bound on the gap length and j is an upper bound. Thus x(4,6) matches any gap with length 4, 5, or 6. They also denote a fixed gap of length i as x(i) (e.g. x(3) = ...). Finally (*) denotes gap of any length (possibly 0).
- *Following string* is an example of a EBOLA pattern containing all mentioned features: [F-x(5)-G-x(2,4)-G-*-H]. Some programs do not allow all these features, for example they do not allow flexible gaps or they allow any gaps but do not allow ambiguous characters other than a wild-card.

### *B. Discovering Frequent Approximate Sequences*

At the beginning select randomly one subsequence of length W from each input sequence. These subsequences will form our initial set of occurrences. Denote oi occurrence in sequence i.Iteration step.

Step 1: Pick randomly one sequence i.

Step 2: Compute position weight matrix based on all occurrences except oi Denote this position weight matrix P.

Step 3: Take each subsequence of sequence i of length W, and compute a score of this sub- sequence according to matrix P.

Step 4: Choose new occurrence o′i randomly among all subsequences of i of length W using probability distribution defined by the score (higher score means higher probability).

Step 5: Replace oi with o′i in the set of occurrences.



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.

Step 6: Repeat iteration steps, until some stopping condition is met.

In model (L, d, k), L is the length of the pattern, d is the maximum distance with in which two strings are considered similar and k is the minimum support or frequency required for a valid patterns. In proposed method we are considering only those strings of length L that actually occur in the dataset and compute the support for each of them by scanning the dataset. This approach will not discover patterns as the model string might not actually occur in the dataset even once.

### *C. LDK Algorithm Methodology*

We first discuss our approach for Problem of complex pattern matching where we are given a pattern P and a text T and T can contain sets of characters. The basic idea of our approach is as follows. We consider each position in T and P as a class or a set for P each position is a set of one character ( i.e. a singleton set). For each character in the alphabet, we construct a smaller similar problem, namely a restricted don't care matching problem, to be defined shortly, and solve them separately. The results are then combined to get the final result. Let us first formally define the restricted don't care matching problem.

Algorithm 1: LDK Algorithm for Complex Pattern Matching
Input:
1. Input sequence is composed of symbols from a discrete alphabet sets.
Set of items/alphabets I= {a1, a2, . . . ., am}
Input Sequential database D = { x | ∀x ϵ I}
E.g. I = {A, B, C, D}
D = ACABBDACCAB
2. Values of L, d and k.
Where
L: Length of substring
D: Distance/number of mismatches allowed
K: Minimum Support
Output:
Set of (L, d, k) model Strings.
Op = {s1, s2, . . . ., sn}
Where
Si = {si1, si2, . . .,sim} is set of instances for si.
Op = {si| d(si, sij) ≤ d, |sij|=L and m ≥ k}

Algorithm 2: Frequent Pattern Matching
Freq Pattern (m Tree, root, L, d, k)
1. process();
2. get model Tree();
3. For i=0 to i<m Tree. size()
4. compute Support();
5. End For

Algorithm 3: Suffix Tree Formation
process()
Construct count suffix tree on input dataset.

Algorithm 4: Generate Model Tree
Get model Tree()
Construct suffix tree of depth L on strings of length L.
That occurs in the count suffix tree.

Algorithm 5: Computing Support Value
Compute Support()
1. old Matches=model. parent. matches;
2. if(model.parent.id=1)
3. new matches=expand Matches()
4. else
5. for k=0 to k<old Matches. size()
6. new Matches. Add All(expand Matches())
7. End for
8. set Matches(new Matches)
9. for i=0 to model Tree. size()
10. x=model Tree. get(i)
11. if(distance()<=d)
12. model. matches. add(x)
13. model. Model Support +=x. support;
14. End for
15. End

Frequently occurring pattern of model (L, d, k) a string of length L that occurs minimum k times in the input dataset, with each occurrence being within a Hamming distance of d from the model string. Hamming distance between strings S and P .Now we are ready to formally state the algorithm in the form of Algorithm 1. The analysis of Algorithm 1 is straightforward. Step 2 and Step 3 can be done implicitly, while performing Step 4. Therefore in Step 1 we basically perform Step 4 |Σ| times requiring O(|Σ|nlogm) time. Step 6 can be done incrementally in the loop and hence the total running time would remain the same i.e. O(|Σ|n log m). It is easy to verify that, if P contains sets of characters instead of T, we just need to swap the definitions of T′ and P′ in Step 2 and Step 3 respectively. We would also like to note that, if both T and P are degenerate the algorithm wouldn't work properly. In the rest of this section this thesis practical efficiency implementation technique of our approach.

## IV. Implementation and Results

In this section we introduce two examples were usage of pattern finding tools helped researchers to make new discoveries. There are of course many such examples. The two examples chosen here illustrate, types of discoveries that can be made and how these discoveries can be verified by biological experiment.

### *A. Performance Evaluation Using EBOLA Dataset*

Proteins secreted by Mycobacterium EBOLA, the causative agent of EBOLA, are often targets of immune responses by an infected host that can lead to protection from the disease. Thus the identification and study of these secretory proteins is a first step to the design of more effective vaccines targeted against this dreadful pathogen. Proteins are directed to the outer membrane of bacterial cells by a short N-terminal sequence of amino acids called a signal peptide.



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.

The signal peptide, which remains attached to the internal face of the membrane, is cleaved to release the protein to the external environment. Conserved features of both the signal peptide and cleavage site, such as length and amino acid composition, make them amenable to identification by sequence analysis.

The recently released nucleotide sequence of the M. EBOLA genome was screened for these conserved features and a number of positive secretory proteins were detected [Gomez et al., 2000].Evidence suggests the existence of other as yet undefined mechanisms for protein secretion in M. EBOLA .Identifying a natural reservoir for Ebola virus has eluded researchers for decades. Dataset for DNA This data string is used as text data contains the information (Homo Sapiens ,DNA ID, DNA Family.

### *B. Result Analysis Comparison Character*

The below DNA sequence dataset has been taken for the testing of the proposed algorithm. The DNA biological sequence.

S∈Σ*of size n = 1024 and pattern P∈Σ*. Let S be the following DNA sequence

The DNA Data for sequence S is very large. For different patterns P's the number of occurrences and the number of comparisons is huge. To check whether the given pattern presents in the sequence or not we need an efficient algorithm with less comparison time and complexity. By the current technique different patterns are analyzed and using these results plots the graph. From the below experimental results, improvement has be seen that LDK algorithm gives good performance compared to the some of the popular methods. reduces the time complexity during worst case from O((n-m+1)m) to O(nm+1). This time complexity is hugely depends on the selected prime number, q. So selecting the right prime number gives this algorithm a satisfiable optimization in terms of worst-case time complexity.

- Different pattern sizes has been taken from the DNA sequence ranging from 1 to 20 randomly and tested by 1024 DNA characters. CPC ratio decreases and is less than 1 in case of the proposed technique where as in other cases CPC is more than 1 in some of the existing methods Hierarchical Verification Algorithm (HVA) and Matching with Depth First Search (MDFS).
- We can see the following fields for comparison of different algorithms like pattern text, number of characters in the pattern, number of occurrences of a pattern, the proposed method and the number of comparisons with comparisons per character.
- The number of comparisons per character (CPC ratio) which is equal to (Number of comparisons/file size) can be used as a measurement factor, this factor affects the complexity time, and when it is decreased the complicity also decreases. Reduction in number of comparisons.
- The ratio of comparisons per character has gradually reduced and is less than 1.
- Suitable for unlimited size of the input file.
- Once the indexes are created for input sequence we need not create them again.
- For each pattern we start our algorithm from the matching character of the pattern which decreases the unnecessary comparisons of other characters.
- It gives good performance for DNA related sequence applications.

TABLE I. Comparison of CPC between algorithms.

| Algorithm | Comparisons Per Character (CPC) |
|---|---|
| HVS | 4 |
| MDFS | 3 |
| LDK | 1 |

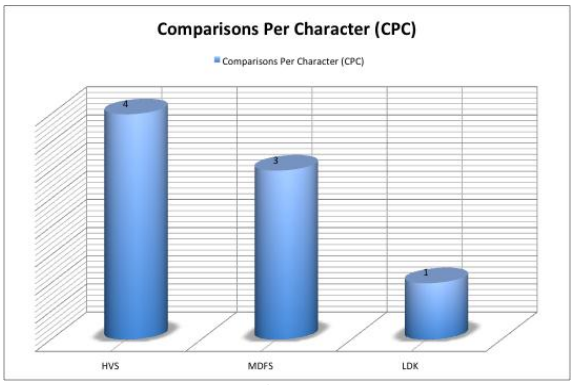


Fig. 2. Comparison of CPC between algorithms.

### *C. Result Analysis - Time Complexity*

In Best case doesn't differ much from the original algorithm, but the in average case complexity can be improved significantly. Due to imposing of constraint of matching complex patterns the HVS algorithm and MDFS algorithms takes more time than the LDK algorithm as shown on figure.

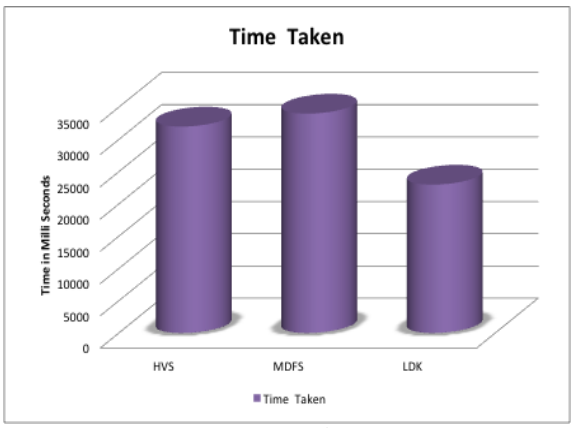


Fig. 3. Shows the results of time complexity.

## V. CONCLUSION

Pattern discovery is an important area of bioinformatics. The algorithms for pattern discovery use wide range of computer science techniques, ranging from exhaustive search,



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.

elaborate pruning techniques, efficient data structures, to machine learning methods and iterative heuristics. The tools developed by computer scientists are today commonly used in many biological laboratories. They are important to handle large scale data, for example in annotation of newly sequenced genomes, and organization of proteins into families of related sequences. They are also important in smaller scale thesis, because they can be used to detect possible sites of interest and assign putative structure or function to proteins. Thus they can be used to guide biological experiments, decreasing the time and money spent in discovering new biological knowledge about the Ebola virus effecting of the peoples.

## REFERENCES

[1] R. Agrawal and R. Srikant, "Mining sequential patterns," *In 11th International Conference on Data Engineering*, 1995.
[2] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo, "Fast discovery of association rules," 1996.
[3] R. Agrawal, T. Imielinsk, and A. Swami, "Mining association rules between sets of items in large databases," *In Proceedings of the ACMSIGMOD Conference on Management of Data*, Washington, D.C. , pp. 207-216, 1993.
[4] P. Hoffman, "DNA visual and analytic data mining," *Proceedings of the 8th IEEE Visualization '97 Conference*, 1997.
[5] P. Papapetrou, "Discovering frequent poly-regions in DNA sequences," *Sixth IEEE International Conference on Data Mining-Workshops (ICDMW'06)*, 2006.
[6] J. Hu, "Mining sequence features for DNA-binding site prediction," *IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, 2008. CIBCB '08*, pp. 219-222, 2008.
[7] K.-S. Leung, "Data mining on DNA sequences of hepatitis B virus," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 8, no. 2, 2011.
[8] B. Ding, "Efficient mining of closed repetitive gapped subsequences from a sequence database," *IEEE International Conference on Data Engineering*, pp. 1024 – 1035, 2009.
[9] S. Bai, "The maximal frequent pattern mining of DNA sequence," *IEEE International Conference on Granular Computing*, pp. 23–26, 2009.
[10] A. Floratou, "Efficient and accurate discovery of patterns in sequence datasets," *IEEE 26th International Conference on Data Engineering (ICDE)*, pp. 461–472, 2010.
[11] S. Li, "An optimized algorithm for finding approximate tandem repeats in DNA sequences," *IEEE Second International Workshop on Education Technology and Computer Science*, pp. 68-71, 2010
[12] C. C. Wang, "Predicting DNA-binding locations and orientation on proteins using knowledge-based learning of geometric properties," *IEEE International Conference on Bioinformatics and Biomedicine*, pp. 3–8, 2010.
[13] A. Floratou, "Efficient and accurate discovery of patterns in sequence data sets," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1154-1168, 2011.
[14] K. Atteson, "The exact probability of language-like patterns in biomolecular sequences," *In Proceedings of the 6th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 17–24, 1998.
[15] T. L. Bailey and C. Elkan, "Fitting a mixture model by expectation maximization to discover motifs in biopolymers," *In Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 28-36, 1994.
[16] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, 21(1/2), pp. 51–80, 1995.
[17] W. C. Barker, F. Pfeiffer, and D. G. George, "Superfamily classification in PIR-international protein sequence database," *Methods in Enzymology*, vol. 266, pp. 59-71, 1996.
[18] G. Benson, "Tandem repeats finder: a program to analyze DNA sequences," *Nucleic Acids Research*, vol. 27, issue 2, pp. 573–580, 1999.
[19] M. Blanchette, B., Schwikowski, and M. Tompa, "An exact algorithm to identify motifs in orthologous sequences from multiple species," *In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 37–45, 2000.
[20] E. B. Bauer, E. Rivals, and M. Vingron, "Computational approaches to identify leucine zippers," *Nucleic Acids Research*, vol. 26, issue 11, pp. 2740–2746, 1998.
[21] A. Brazma, I. Jonassen, I. Eidhammer, and D. Gilbert, "Approaches to the automatic discovery of patterns in biosequences," *Journal of Computational Biology*, vol. 5, issue 2, pp. 279–305, 1998.
[22] A. Califano, "SPLASH: structural pattern localization analysis by sequential histograms. Bioinformatics, vol. 16, issue 4, pp. 341–347, 2000.
[23] R. Cowan, "Expected frequencies of DNA patterns using Whittle's formula," *Journal of Applied Probability*, 28(4), pp. 886–892, 1991.
[24] E. Coward, and F. Drab- los, "Detecting periodic patterns in bio- logical sequences," *Bioinformatics*, vol. 14, issue 6, pp. 498–507, 1998.
[25] B. Dorohonceanu and C. G. N. Manning, "Accelerating protein classification using suffix trees," *In Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB)*, pp. 128–133, 2000.
[26] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison, "Biological Sequence Analysis," Cambridge University Press, Calculating 30, 1998.



S. Kalaiselvi and A. Meena, "Efficiency of using sequence discovery for polymorphism in DNA sequence," *International Journal of Scientific and Technical Advancements*, Volume 2, Issue 1, pp. 95-100, 2016.