International Journal of
Scientific Engineering and Technology Research

www.ijsetr.com

ISSN 2319-8885
Vol.03,Issue.35
November-2014,
Pages:6916-6920

Comparison of Three Pattern Matching Algorithms using DNA Sequences

NYO ME TUN[1], THIN MYA MYA SWE[2]

[1]Dept of Research and Development, Computer University, Mandalay, Myanmar, E-mail: nyomizin@gmail.com.
[2]Dept of Research and Development, Computer University, Mandalay, Myanmar.

Abstract: Pattern matching is commonly used in computer science and information processing such as text editor in computing, database queries, bioinformatics, language syntax checker, music content retrieval, DNA sequences matching, search engines and many more applications. There are many forms of pattern matching. Among them, the most well-known form is bioinformatics application, DNA sequences analysis of various diseases which are stored in database for retrieval and comparison. To get high quality results in a short time is to use pattern matching algorithms because of DNA database is very complex and huge and not to retrieve easily. This paper presents three pattern matching algorithms, Knuth-Morris-Pratt (KMP), Brute-Force and Boyer-Moore. The main process of this system is to find the matched DNA sequence in DNA database that find the matched DNA sequence in DNA database using these pattern matching algorithms. In this system, at least two DNA sequences are required to analyze. This system compares similarity values with threshold value and stores particular result which is diseased or not. Finally, this system can identify optimal result according to each result using voting. This system is implemented with Java programming language and the comparison results are demonstrated with bar graph with respect to similarity and processing time of each algorithm.

Keywords: Pattern Matching, DNA Sequences Analysis, Medical Diagnosis.

I. INTRODUCTION

Exact sequence matching is a vital component of bioinformatics which is the application of computer technology to management and analysis of biological data. Computer technologies are applied to gather, store, analyze and merge biological data. Therefore, bioinformatics is an interface between biological and computational sciences [2]. DNA Pattern matching, the problem of finding subsequences within a long DNA sequence has many applications in computational biology. As the sequences can be long, matching can be an expensive operation, especially as approximate matching is allowed [6]. There are many biological sequence data in different databases. The biological and bibliographical sequence data are growing in these databases at an exponential rate. The computational demands are to find matched sequence and to analyze these sequence data contained in these databases. Exponential growth of gene databases enforces the need for efficient information extraction methods and algorithms for DNA sequencing exploiting existing large amount of DNA information available in DNA database. Manually it is very tedious and time consuming [1].

Deoxyribonucleic acid (DNA) is an acronym for the molecule deoxyribonucleic acid and it is also called a double helix because it is a double-stranded molecule. It is mainly composed of nucleotides of four types. The four bases in DNA are Adenine (A), Cytosine (C), Guanine (G), and Thymine (T). A DNA sequence is a representation of a string of nucleotides contained in a strand of DNA. For example: ATTCGTAACTAGTAAGTTA. The DNA sequencing techniques have allowed the vast amount of data to be analyzed in a short span of time. So, pattern matching techniques plays a vital role in computational biology for data analysis related to biological data such as DNA sequences[1][2][6]. This paper presents and analyzes three pattern matching algorithms to find the matched DNA sequence in DNA database. Pattern-matching algorithm matches the pattern exactly or approximately within the text. The organization of this paper is as follows. SectionII presents the related work of pattern matching process. In sectionIII, theoretical background for DNA sequence matching is given. In sectionIV, proposed system is demonstrated. In sectionV, system implementation is described. In sectionVI, the experimental results are shown. Finally, in sectionVII, the conclusions and references are given.

II. RELATED WORK

The pattern matching algorithms can be applied for detecting the unusual patterns present in the gene database. It can show how the disease can be transformed from parents to their children and can identify the presence of the disease on hereditary basis and its impact. Moreover, KMP algorithm known as DNA sequencing algorithm is used to detect unusual patterns in DNA sequences and to analyze these sequences in the gene database [1][2]. Approximate String Matching Algorithm, fuzzy string searching technique is used to find strings that match a pattern approximately. It is applied in finding approximate substring matches inside a given string and finding dictionary strings that match the pattern approximately. Another application of approximate string matching can be effectively used to retrieve fast video as it uses the content based video retrieval in contrast with the traditional video retrieval which was slow and time consuming. String based video retrieval method first converts the unstructured video into a curve and marks the feature string of it. Approximate string matching is then used to retrieve video quickly [3][4].

III. THEORETICAL BACKGROUND

A. Pattern Matching

DNA pattern matching is a fundamental and upcoming area in computational molecular biology. Pattern matching is an important task of the pattern discovery process in today's world for finding the structural and functional behavior in proteins and genes. Although pattern matching is commonly used in computer science and information processing, it can be found in everyday tasks. The string matching can be described as: given a specific strings P generally called pattern searching in a large sequence/text T to locate P in T. if P is in T, the matching is found and indicates the position of P in T, else pattern does not occurs in the given text. As the size of the data grows it becomes more difficult for users to retrieve necessary information from the sequences. Hence more efficient and robust methods are needed for fast pattern matching techniques. It is one of the most important areas which have been studied in bioinformatics. Pattern matching techniques has two categories and is generally divides into:

- Single pattern matching
- Multiple pattern matching techniques.

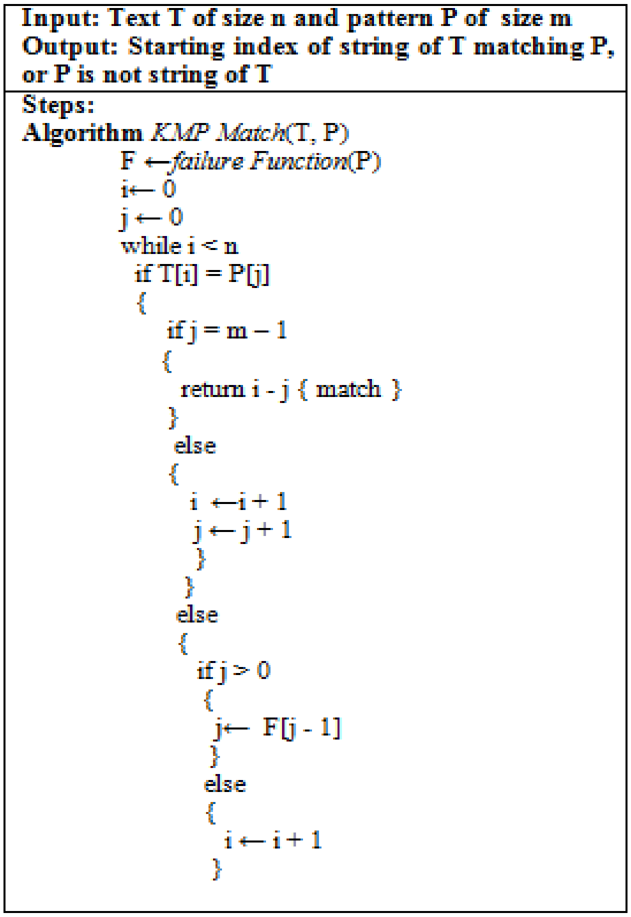
Single pattern matching is to find all occurrences of the pattern in the given input text. Suppose, if more than one pattern are matched against the given input text simultaneously, then it is known as, multiple pattern matching. Multiple pattern matching can search multiple patterns in a text at the same time. It has a high performance and good practicability, and is more useful than the single pattern matching algorithms [5][6][7]. Let P = {p_1, p_2, p_3,...,p_m} be a set of patterns of m characters and T={t_1,t_2,t_3…,t_n} in a text of n character which are strings of nucleotide sequence characters from a fixed alphabet set called Σ={A C, G, T}[7]. There are various pattern matching algorithms. These efficient algorithms are used to the sequence of DNA in the DNA database. The present day pattern matching algorithms match the pattern exactly or approximately within the text [1][2][7]. Among them, this paper applies three pattern matching algorithms. They are:

- Knuth-Morris-Pratt
- Brute-Force and
- Boyer-Moore

B. Knuth-Morris-Pratt (KMP) Algorithm

The KMP algorithm is linear sequence matching algorithm and plays an important role in bioinformatics, DNA sequence matching, disease detection etc. It scans the sequence from left to right. This algorithm preprocess the pattern string P using failure function. There is a match, increase the current indices. If not, consult the failure function the new index in P here needs to continue checking P against T. The algorithm repeat this process until find a match of P (pattern) in T(Text) or index for T reach n, the length of T. The main part of the KMP algorithm is the while-loop which performs comparison between a character in T and a character in P for each iteration. This algorithm doesn’t need backtracking and shift more than one position. Therefore, KMP is called as an intelligent search algorithm [1] [2].

TABLE I: Knuth-Morris-Pratt (KMP) Pattern Matching Algorithm



Input: Text T of size n and pattern P of size m
Output: Starting index of string of T matching P, or P is not string of T

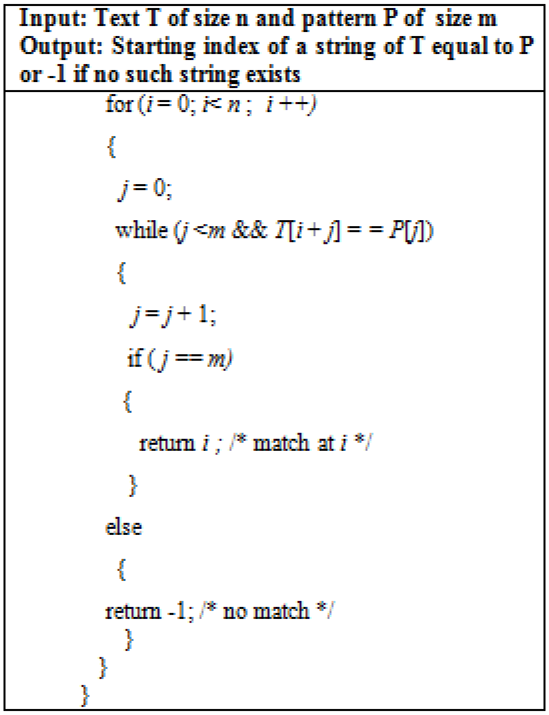
Steps:

```
Algorithm KMP Match(T, P)
        F ←failure Function(P)
        i← 0
        j ← 0
        while i < n
         if T[i] = P[j]
         {
           if j = m – 1
           {
             return i - j { match }
           }
            else
           {
             i ←i + 1
             j ←j + 1
             }
           }
          else
          {
            if j > 0
            {
              j← F[j - 1]
            }
            else
            {
              i ← i + 1
            }
```

C. Brute-Force Algorithm

The brute force algorithm is a powerful technique that is used to search the pattern. This algorithm is simplest string method which is used to match each character of pattern to the corresponding character in text until all characters are found to match successful search; or mismatch is detected. This algorithm scans the sequence from left to right direction. This algorithm doesn’t need preprocessing phase. It consists of two nested loops, with the outer loop indexing through all possible starting indices of the pattern in the text, and the inner loop indexing through each character of the pattern, comparing it to its potentially corresponding character in the text [2] [6].

TABLE II: Brute-Force Pattern Matching Algorithm



Input: Text T of size n and pattern P of size m
Output: Starting index of a string of T equal to P or -1 if no such string exists

```
        for (i = 0; i< n ;  i ++)
        {
         j = 0;
         while (j <m && T[i + j] = = P[j])
         {
           j = j + 1;
           if ( j == m)
           {
             return i ; /* match at i */
           }
        else
         {
        return -1; /* no match */
          }
        }
      }
```

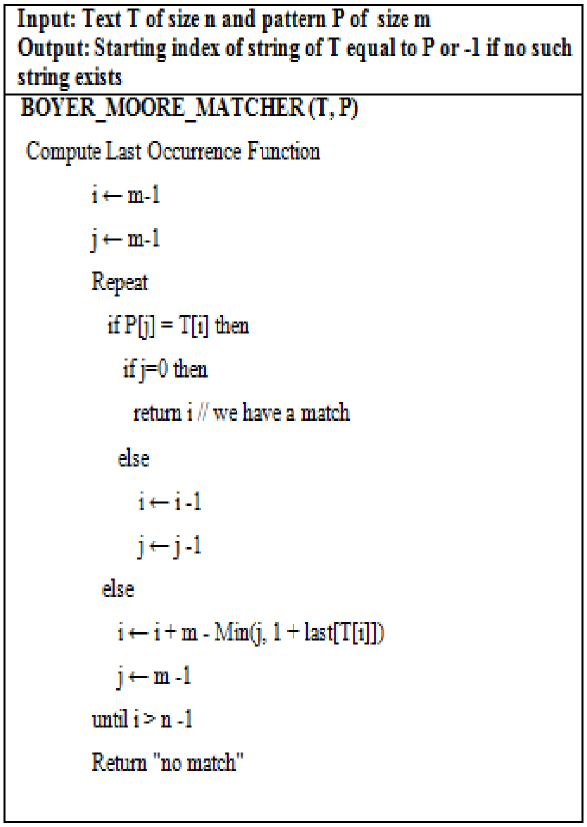
D. Boyer-Moore Algorithm

Boyer-Moore algorithm is an efficient string searching algorithm that can determine whether or not a match of a particular string exists within another string. So, this algorithm is used in bioinformatics mostly, for disease detection. This algorithm scans the matching of the two sequences from right to left direction. Therefore, this algorithm takes backward approach. If no mismatch occurs, then the pattern has been found. Otherwise, the algorithm computes a shift; that is, an amount by which the pattern is moved to the right before a new matching attempt is undertaken. In case of a mismatch, the knowledge gained from the last occurrence function helps in calculating the new index to start the next search. This algorithm preprocesses the pattern P using the last-occurrence function L mapping ∑, where L(c) is defined as: the largest index i such that P[i] = c or -1 if no such index exists [1] [2].

E. DNA (Deoxyribonuclei Acid)

Deoxyribonucleic Acid (DNA) is a nucleic acid that contains genetic instructions. DNA is a double helix structure. It contains three components: a five-carbon sugar (Deoxyribose), a series of phosphate groups, and four nitrogenous bases. The four bases in DNA are Adenine (A), Thymine (T), Guanine (G), and Cytosine (C). Thymine and adenine always come in pairs. Likewise, guanine and cytosine bases come together too. Every human has his/her unique genes. Genes are made up of DNA; therefore the DNA sequence of each human is unique. The sequence of DNA constitutes the heritable genetic information in nuclei, plasmids, mitochondria, and chloroplasts that forms the basis for the developmental programs of all living organisms [1][2][6][7].

TABLE III: Boyer-Moore Pattern Matching Algorithm



Input: Text T of size n and pattern P of size m
Output: Starting index of string of T equal to P or -1 if no such string exists

```
BOYER_MOORE_MATCHER (T, P)
 Compute Last Occurrence Function
        i ← m-1
        j ← m-1
        Repeat
          if P[j] = T[i] then
            if j=0 then
              return i // we have a match
            else
              i ← i -1
              j ← j -1
          else
            i ← i + m - Min(j, 1 + last[T[i]])
            j ← m -1
        until i > n -1
        Return "no match"
```

The process of determining the DNA sequence is used to map out the sequence of the four nucleotides that comprise a strand of DNA. Studying the DNA sequence is necessary in basic research studying fundamental biological processes, as well as in applied fields such as diagnostic and forensic research. The four nucleotides (A, T, G, C) are grouped to form words and these words make sentences which are called genes. The occurrence of unwanted or mutated words causes the disease. Every disease will have its own words or sequences on the occurrence of the mutated sequence in the DNA [1][2][6]. This paper matches the DNA sequences of three diseases called HIV, Lung cancer and Leukaemenia.

IV. PROPOSED SYSTEM

This paper presents three pattern matching algorithms for DNA sequence matching**.** The main process of this paper is finding the matched DNA sequence in DNA database using pattern matching algorithms. When entering human DNA sequence (suspected DNA sequence) and the type of disease to be checked as input, the matched DNA sequence are searched using three pattern matching algorithms: Knuth-Morris-Pratt (KMP), Brute-Force and Boyer-Moore. The total matched indices are got and similarity values with respect to entire sequence are computed. Then, the similarity values are compared with threshold value (user defined value) and stores the particular results which is diseased or not. Finally, the optimal result is sent back according to each stored result using voting method. Fig.1 is the proposed system architecture for the DNA sequence matching.

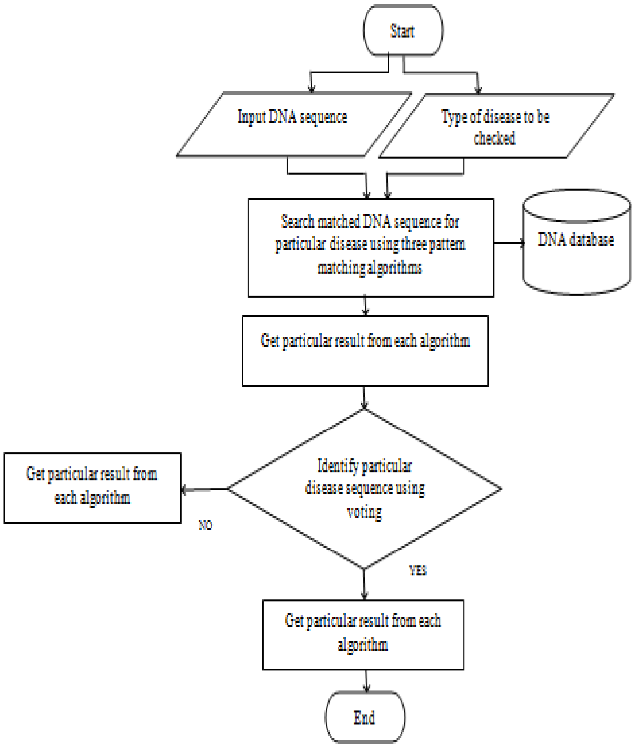

Start
Input DNA sequence
Type of disease to be checked
Search matched DNA sequence for particular disease using three pattern matching algorithms
DNA database
Get particular result from each algorithm
Identify particular disease sequence using voting
Get particular result from each algorithm
NO
YES
Get particular result from each algorithm
End

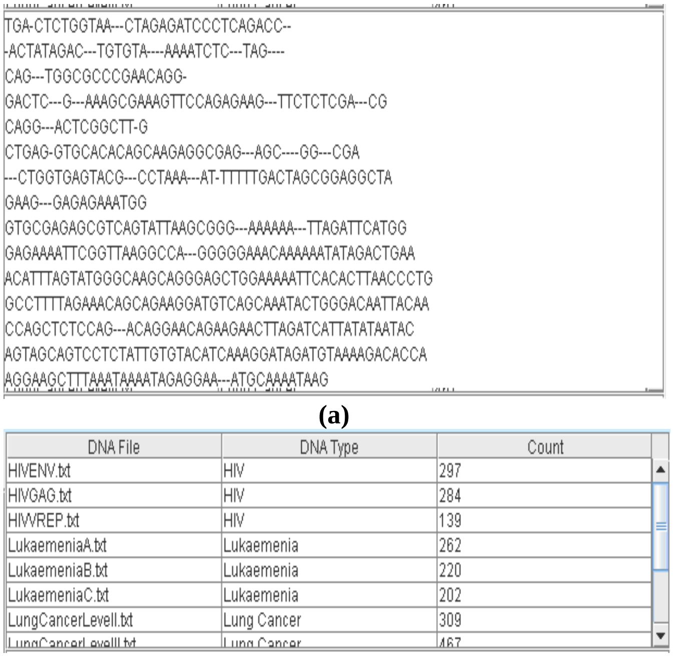
Fig.1. System Architecture Design.

V. SYSTEM IMPLEMENTATION

This system is implemented using Java programming language and tested using different DNA sequence with different file size. Pattern matching techniques are used to search the matched DNA sequences in mostly bioinformatics. The main process of the system is finding the matched DNA sequence in a set of DNA database. The pattern matching algorithms are used to find the matched sequence and the total matched indices are used to compute similarity value with respect to entire sequence. Since DNA sequences are very large and complex, it is impossible to analyze the vast amount of data in a short of span. The pattern matching algorithms are efficient to trace the sequence of DNA in the DNA database. These efficient pattern matching techniques can give optimal result for particular diseased DNA sequence. All the three algorithms require at least two DNA sequences. One of these sequences is generally a suspected DNA sequence and other is a diseased sequence. Moreover, pattern matching techniques is used to optimize the time and to analyze the vast amount of data in a short span of time.

VI. EXPERIMENTAL RESULTS

The results of the proposed system are described in this section. Fig.3 describes the particular result of checked disease for particular algorithm. From this Fig.3, we can decide the optimal result for particular disease. Fig.2 (a) is the input DNA, Figure 2 (b) presents the types of DNA available in this system. Fig.3 describes the result of the system. Finally, the similarity and processing time are shown as Fig.4 which is two comparison results of three pattern matching algorithms. Threshold value used in this system is 0.75 (user defined value).



```
TGA-CTCTGGTAA---CTAGAGATCCCTCAGACC--
-ACTATAGAC---TGTGTA----AAAATCTC---TAG----
CAG---TGGCGCCCGAACAGG-
GACTC---G---AAAGCGAAAGTTCCAGAGAAG---TTCTCTCGA---CG
CAGG---ACTCGGCTT-G
CTGAG-GTGCACACAGCAAGAGGCGAG---AGC----GG---CGA
---CTGGTGAGTACG---CCTAAA---AT-TTTTTGACTAGCGGAGGCTA
GAAG---GAGAGAAATGG
GTGCGAGAGCGTCAGTATTAAGCGGG---AAAAAA---TTAGATTCATGG
GAGAAAATTCGGTTAAGGCCA---GGGGGAAACAAAAAATATAGACTGAA
ACATTTAGTATGGGCAAGCAGGGAGCTGGAAAAATTCACACTTAACCCTG
GCCTTTTAGAAACAGCAGAAGGATGTCAGCAAATACTGGGACAATTACAA
CCAGCTCTCCAG---ACAGGAACAGAAGAACTTAGATCATTATATAATAC
AGTAGCAGTCCTCTATTGTGTACATCAAAGGATAGATGTAAAAGACACCA
AGGAAGCTTTAAATAAAATAGAGGAA---ATGCAAAATAAG
```

(a)

DNA File	DNA Type	Count
HIVENV.txt	HIV	297
HIVGAG.txt	HIV	284
HIVVREP.txt	HIV	139
LukaemeniaA.txt	Lukaemenia	262
LukaemeniaB.txt	Lukaemenia	220
LukaemeniaC.txt	Lukaemenia	202
LungCancerLevell.txt	Lung Cancer	309
LungCancerLevelII.txt	Lung Cancer	467

(b)

Fig.2. (a) Input DNA, (b). DNA Types available in this system.

Algorithm	Disease Type	Max-Match	Total Length	Similarity value	Disease	Result using voting
Knuth-Morris-Pratt	HIV	37	44	0.84090	YES	YES
Brute-Force	HIV	37	58	0.63793	NO	
Boyer-Moore	HIV	37	44	0.84090	YES	

Fig.3. Result of the system.

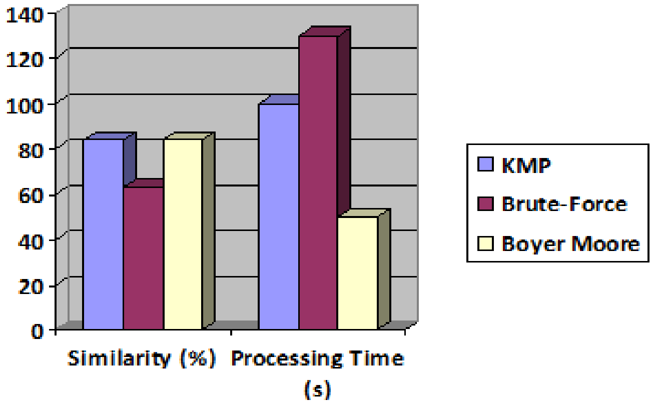


Fig.4. Two Comparison Results of Three Pattern Matching Algorithms.

VII. CONCLUSION

This paper provides a path in diagnosing the disease by the identification of presence of diseased DNA sequence in DNA database. The pattern matching algorithms are effectively used in matching DNA sequences because of DNA database is very complex and huge and not to retrieve easily. The pattern matching process is needed to keep pace with ever growing demands in sequence comparison. The required sequences are suspected sequence and diseased sequence. These algorithms in this system are easy to implement and effective in multiple detections of DNA sequences. This paper identifies the particular disease on DNA sequence using three pattern matching algorithms. Finally, this paper gives optimal result according to each result using voting.

VIII. ACKNOWLEDGEMENTS

My Sincere thanks to my supervisor Dr.Thin Mya Mya Swe, Lecture, Research and Development Department, Computer University (Mandalay), Myanmar for providing me an opportunity to do my research work. I express my thanks to my Institution namely Computer University (Mandalay) for providing me with a good environment and facilities like internet books, computers and all that as my source to complete this research. My heart-felt thanks to my family, friends and colleagues who have helped me for the completion of this work.

IX. REFERENCES

[1] Kuhu Shukla, Samarjeet Borah, Sunil Pathak, “An analysis of influential DNA sequencing Algorithms”.

[2] S.Rajesh, S. Prathima, Dr. L.S.S.reddy, “Unusual pattern detection in DNA database using KMP algorithm”.

[3] Nimisha Scingla, Deepark Garg, “String Matching algorithms and their Applicability in various Applications”.

[4] Vidya Saikrishna Prof. Akhtar Rasool and Dr. Nilay, Khare Department, “String Matching and its Application in Diversified Fields”.

[5] “String and pattern matching Algorithms”.

[6] Rajib Paul, Pooja Roy, “Improved Pattern Matching To Find DNA Patterns”.

[7] Raju Bhukya, DVLN Somayajulu “Exact Multiple Pattern Matching Algorithm using DNA Sequence and Pattern Pair”.

[8]http://www.broadinstitute.org/cgibin/cancer/datasets.cgi.

[9]http://www.pombase.org/downloads/dna datasets.