IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-7, July 2018 E-ISSN: 2347-2693

# A Novel Approaches to Study Different DNA Pattern Matching Algorithms over Two Compression Techniques

**G.Dutta[1*], A. Mukherjee[2]**

[1] Computer Science & Engineering, Brainware Group of Institutions, MAKAUT, Kolkata, India
[2] Computer Science & Engineering, Brainware Group of Institutions, MAKAUT, Kolkata, India

*Corresponding Author: gourabdutta15@gmail.com, Tel.: 9903432438

**Available online at: www.ijcseonline.org**

Accepted: 19/Jul/2018, Published: 31/Jul/2018

***Abstract*—** Pattern matching is a technique to find the given pattern over the text within a database. Different types of algorithms are used to find the desired pattern over a text. For easy retrieval of DNA sequences of various diseases which are stored in large databases and comparison happens in sequence analysis. By using DNA pattern matching if it is found that a particular sequence occurs again and again and by counting the no of occurrence to find the existence and intensity of a disease. Compression is a technique for reducing the quantity of data used to represent any content without excessively reducing the quality of the data(ie. image, video etc.). Data compression is the process of encoding information using fewer bits than an uncoded representation is also making a use of specific encoding schemes. This procedure also reduces the number of bits required to store over the disk. For large amount of data, compression is a technique that makes storing easier. Different techniques are used for data compression. In this paper to find out a particular pattern in the given compressed DNA sequence using Brute-force, Boyer-Moor and KMP string matching algorithm and also measure performance of those algorithms more efficiently. Those algorithms are executed over the compressed DNA sequence to measure their performances and to avoid wasteful comparison. Two different techniques, ¼ th compression and Huffman compression is used for compressing the DNA sequences and compared which one is better among these two different techniques for pattern matching.

***Keywords*—**Bioinformatics, DNA Pattern Matching, Brute-force, Boyer-Moor, KMP, ¼ th compression, Huffman coding.

## I. Introduction

DNA is the basic blueprint of life and it can be viewed as a long sequence over the four alphabets A, C, G and T. DNA contains genetic instructions of an organism. It is mainly composed of nucleotides of four types Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). DNA sequence file size is very large. The amount of DNA extracted from the organism is increasing exponentially. In computational biology pattern matching techniques plays a vital role in various applications for data analysis related to protein and gene. 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. For DNA pattern matching it is very difficult to find the pattern from the large Text because the text size is large so the time and space complexity is very high for DNA pattern matching technique. For this reason we have first compressed the DNA sequence file by using some traditional pattern matching technique. DNA compression refers to reducing the number of bit used to represent a DNA sequence without excessively reducing the meaning of the original data. It also reduces the number of bits required to store the DNA sequence. Compressing something means decreasing its size. Then the space complexity automatically reduces. After compression the pattern matching technique run over this compressed file.

In this research a survey is made on the performance of three pattern matching algorithm namely Brute-force, Boyer-Moor and KMP, executed over compressed DNA sequences. ¼th compression and Huffman compression is

used for compressing the DNA sequences to avoid unnecessary comparisons in the pattern and comparerdwhich one is better among these two different compression techniques for pattern matching.
In Section II, related work is discussed. Section III is about motivation. DNA pattern matching algorithms (ie. *Brute-Force Algorithm*, *Boyer-Moore Algorithm*, *Kunth-Morris-Pratt's (KMP) Algorithm)* are discussed in Section IV. In Section V DNA file compression techniques are explained. Section VI is all about implementation.

## II. Related Work

Different authors have worked on DNA pattern matching problem and file compression technique. Authors in paper [1] demonstrated an approach for looking for a pattern in a file that is stored in its compressed form. They first decompressed the file and then applied known pattern matching algorithms in the decoded file. In many cases, however, in particular on the Internet, files are stored in their original form, for if they were compressed, the host computer would have to provide memory space for each user in order to store the decoded file. This requirement is not reasonable, as many users can simultaneously quest the same information reservoir which will demand an astronomical quantity of free memory. On the other hand if an user transferred the compressed files to the personal computer then file has to be decoded.

In paper [2],de Moura, et al. proposed a compression scheme that uses a Huffman coding on words.

In paper[3],Raju Bhukya, DVLN Somayajulu proposed different types of pattern matching techniques over DNA sequences. Different researchers all over the world compared different types of string matching algorithms on text and DNA sequences to reduce space complexity and time complexity by first compressing the DNA sequence and then executing different types of string matching algorithm over this compressed file.

In the paper [4] authors showed that approximate string matching on compressed text is slower than uncompressing the text and proposed an approach to reduce the problem to multi pattern searching of pattern pieces plus direct verification and local decompression of candidate text areas. They also proved experimentally that this solution is 10 to 30 times faster than previous work.

In paper [5] Mamta Sharma used Huffman Coding to compress the JPEG image file and proved that the Huffman Coding is lossless compression over JPEG file. Different researchers all over the world compare different types of string matching algorithm on text and DNA sequences.

In paper [6] authors compare the performance of Brute-force, Boyer-Moore, Finite automata algorithm and KMP algorithm on text in terms of complexity.

Authors in paper [7] uses KMP algorithm to search the unusual pattern in DNA database.

In Paper [8] it was shown that Parallel KMP algorithm based on MPI provides higher efficiency than the traditional pattern matching algorithm. Different authors have worked on DNA pattern matching problem. In paper [9], authors compressed the DNA file using 1/4th Compression technique and then run derivative Boyer-Moore (d-BM) over this DNA file.

## III. MOTIVATION

All of the previous works [1,2,3,4,5,6,7,8,9] mainly focus on either the huffman compression techniques over image file or ¼ th compression technique over DNA file or string pattern matching techniques to find pattern from samples or first apply ¼ th compression technique over DNA file after that pattern matching technique apply over this compress DNA file. This paper compares the performance of different string matching algorithms over compressed DNA sequences. ¼th compression and Huffman compression is used for compressing the DNA sequences and sequence for searching of a desired DNA pattern in large DNA file and compared which one is better among these two different compression techniques for pattern matching.

## IV. DNA PATTERN MATCHING ALGORITHM

This paper surveys the performance of three DNA pattern matching technique namely Brute-force, Boyer-Moor and KMP over different compression technique.

### *A. Brute-Force Algorithm*

This algorithm 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. The worst case complexity of this algorithm is O(nm) where n is the number of characters in text and m is the number of characters in pattern.

### *B. Boyer-Moore Algorithm*

The algorithm of Boyer-Moore compares the pattern with the text from right to left. If at all the text symbol which is compared with the rightmost pattern symbol does not occur in the pattern, then the pattern can be shifted by m positions behind this text symbol. The Boyer-Moore algorithm works the fastest when the alphabet is of moderately sized and the pattern is relatively long. The algorithm scans the characters of the pattern from right side to left side beginning with the rightmost character. A possible placement of pattern P during the testing of text T, a mismatch of text character T[i] = c with the corresponding pattern character P[j] is handled as follows: If c is not present anywhere in P, then shift the pattern P completely past T[i]. Otherwise, shift P until an occurrence of character c in P gets aligned with T[i]. This technique avoids lots of needless comparisons by significantly shifting pattern relative to text. The complexity of this algorithm is also O (nm). It is faster than the brute-force algorithm. This algorithm works in worst case for images and DNA sequences.

### *C. Kunth-Morris-Pratt's(Kmp) Algorithm*

The Knuth-Morris-Pratt (or "KMP") algorithm avoids unnecessary comparison and runs in order of (n + m) time in worst case. The main features of this algorithm are as follows

Knuth-Morris-Pratt algorithm compares the pattern over the text from left-to-right, but the shifts of the pattern more hyper intelligent than the brute-force algorithm.

When a mismatch occurs, first algorithm finds the largest prefix of the pattern which is also a suffix in the pattern and then that number of bits is shifted to avoid redundant comparisons.

The Knuth-Morris-Pratt algorithm is faster than the brute-force algorithm.

## V. DNA FILE COMPRESSION TECHNIQUE

The maximum DNA sequences reported today is three billion. By taking these statements as benchmark, the number of memory space requirement will increase. The format of the standard DNA data is written in ASCII code format with eight bit data width as shown in Table I, in which each characters is been represented with its specific ASCII code. This initial problem contributed for large memory space requirement.

**TABLE I. ASCII CODE DNA CHARACTER DATA REPRESENTATION**

| Name | Characters | ASCII | Data Bit |
|---|---|---|---|
| Adenine | A | 65 | 01100101 |
| Cytosine | C | 67 | 01100111 |
| Guanine | G | 71 | 01110001 |
| Thymine | T | 84 | 10000100 |

This paper uses ¼ th DNA compression technique to compress the data to get rid of huge memory space requirement and unnecessary comparison problem.

### *A. ¼ th Compression Technique*

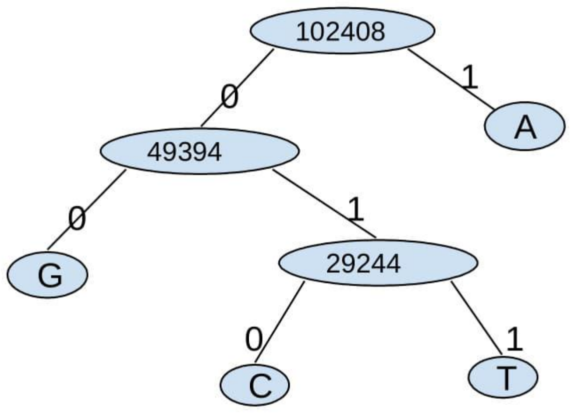
In this technique the DNA characters is assigned with two bit data representation as shown in Table II, where A is represented by “00”, C with “01”, G with “10” and lastly T with “11”.

**TABLE II. DNA SEQUENCE CHARACTERS REDUCTION DATA ASSIGNMENT USING ¼ th COMPRESSION TECHNIQUE.**

| Name | Characters | Reduction Data |
|---|---|---|
| Adenine | A | 00 |
| Cytosine | C | 01 |
| Guanine | G | 10 |
| Thymine | T | 11 |

### *B. Huffman Compression Technique*

In computer science and information theory Huffman coding is an entropy encoding algorithm used for lossless data compression.The term refers to the use of a variable-length code table for encoding a source symbol (such as a character in a file) where the variable-length code table has been derived in a particular way based on the estimated probability of occurrence for each possible value of the source symbol. Huffman coding uses a specific method for choosing the representation for each symbol, resulting in a prefix-free code (that is, the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol) that expresses the most common characters using shorter strings of bits than are used for less common source symbols. Huffman was able to design the most efficient compression method of this type. No other mapping of individual source symbols to unique strings of bits will produce a smaller average output size when the actual symbol frequencies agree with those used to create the code. A method was later found to do this in linear time if input probabilities (also known as weights) are sorted. For a set of symbols with a uniform probability distribution and a number of members which is a power of two, Huffman coding is equivalent to simple binary block encoding,e.g., ASCII coding.
If the number (frequency) of the character ‘A’ is 53014, number (frequency) of the character ‘T’ is 18804, number (frequency) of the character ‘G’ is 20150 and number (frequency) of the character ‘C’ is 10440 in a DNA text file then the tree will be as follows:



In huffman compression technique DNA characters is assigned with different number of bit data representation as shown in Table III for sample DNA file size of 1 lakh, the Huffman code will be ‘1’ for ‘A’ , ‘00’ for ‘G’, ‘010’ for ‘C’ and ‘011’ for ‘T’.

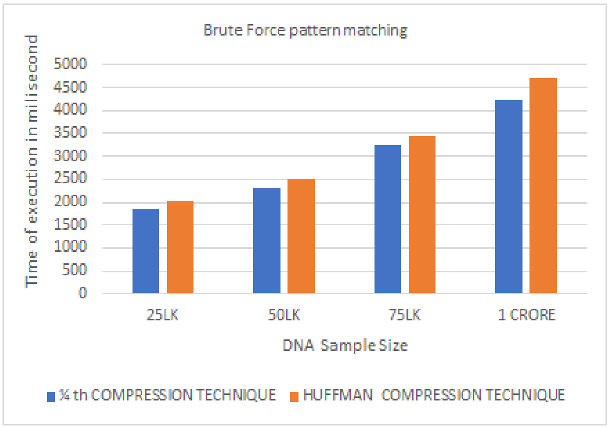
**TABLE III. PROPOSED DNA SEQUENCE CHARACTERS REDUCTION DATA ASSIGNMENT USING HUFFMAN COMPRESSION TECHNIQUE.**

| Name | Character | Frequency(No of character) | Data Bit |
|---|---|---|---|
| Adenine | A | 53014 | 1 |
| Cytosine | C | 10440 | 010 |
| Guanine | G | 20150 | 00 |
| Thymine | T | 18804 | 011 |

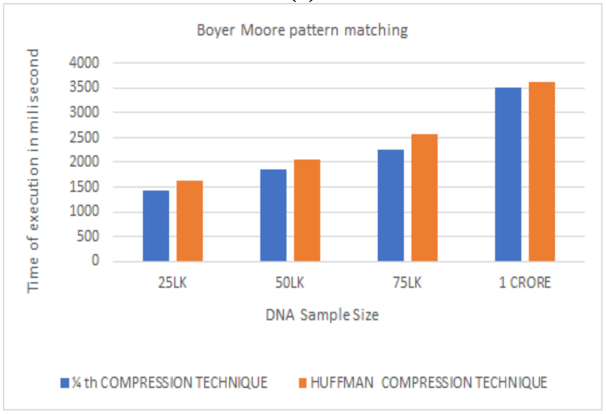
## VI. IMPLEMENTATION

In this paper we have first compressed the given pattern and text using 1/4th compression technique and then run Brute-force, Boyer-Moor and KMP algorithm on the compressed sequences. Again compressed the same uncompress pattern and text using huffman compression technique and then again run Brute-force, Boyer-Moor and KMP algorithm on this compressed sequences. C programming language is used to implement the survey work. The comparisons have been done by taking the sample size of 1 lakh, 5 lakh, 25 lakh, 50 lakh, 75 lakh and 1 crore and pattern is ATGC. The results are shown in

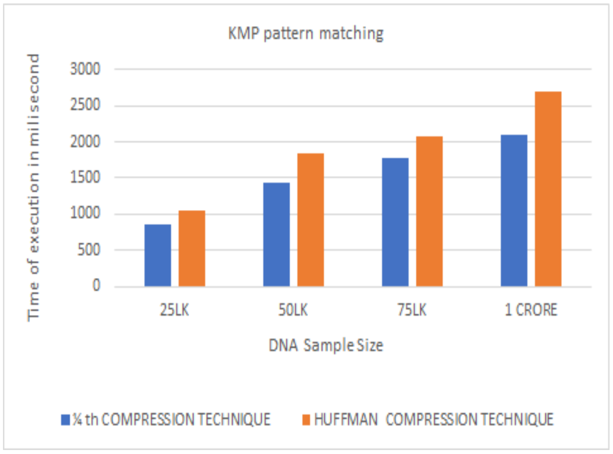
**[Figure-1] and [Figure-2]** .



(a)

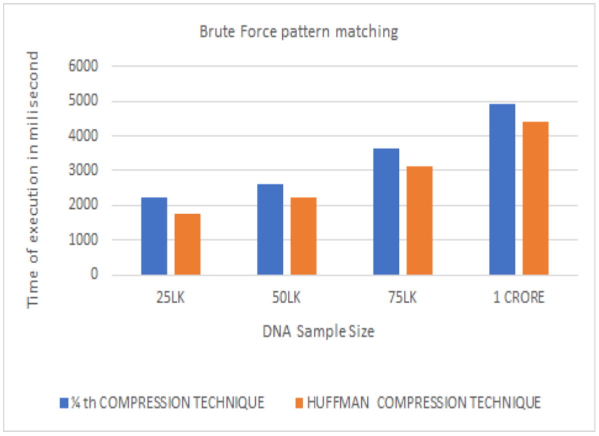


(b)

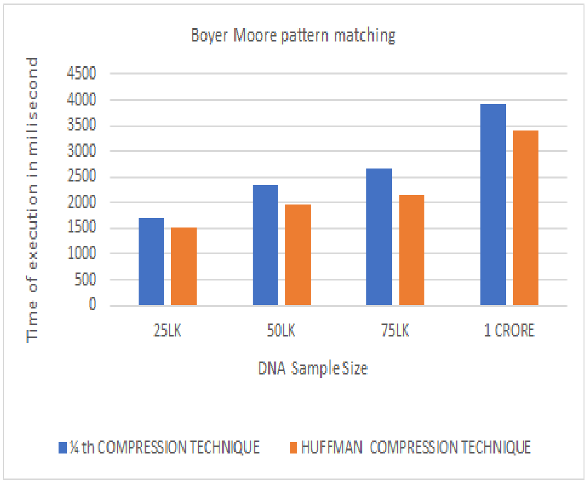


(c)

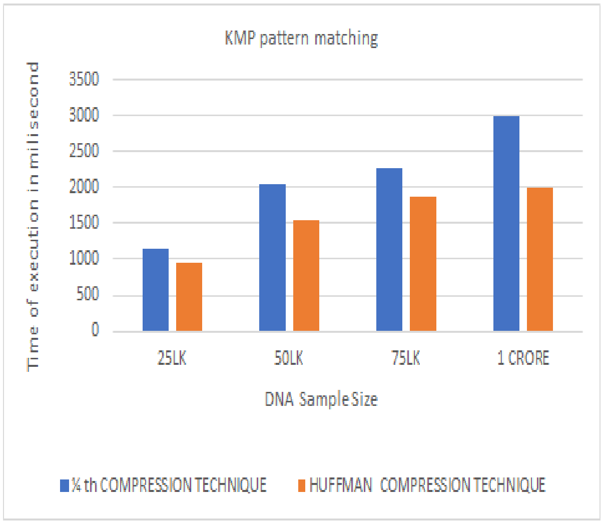
Figure-1.If four character of DNA sequence more or less equally distributed.



(a)



(b)



(C)

Figure-2.If any one or two character present more

## VI. CONCLUSION

In this paper,the various kind of string matching algorithms were studied with compressed biological sequences such as DNA. From the studying, it is analyzed that Brute-force, Boyer-Moor and KMP string matching algorithm to find out a particular pattern in the given DNA sequence more efficiently over huffman compression technique if out of four character(A,T,G,C) of DNA sequence any one or two character present more into the DNA file and in this case ¼ th compression technique is not more efficient for find out a particular pattern. And also analyzed that Brute-force, Boyer-Moor and KMP string matching algorithm to find out a particular pattern in the given DNA sequence is more efficiently over ¼ th compression technique if four character of DNA sequence more or less equally distributed over the DNA sequences text file in this case huffman compression technique is not more efficient.

## VII. REFERENCES

[1] S.T. Klein, D. Shapira, "*A New Compression Method for Compressed Matching*",IEEE conference on Data Compression (DCC), August, 2002.

[2] E.S.de Moura,G.Navarro,N.Ziviani and R.Baeza-Yates,"*Direct pattern matching on compressed text*", In Proc. 5th International Symp. on String Processing and Information Retrieval, IEEE Computer Society, pp. 90-95,1998.

[3] Raju Bhukya, DVLN Somayajulu, "*Exact Multiple Pattern Matching Algorithm using DNA Sequence and Pattern Pair*",International Journal of Computer Applications (IJCA),Vol. 17 No.8,pp: 32-38, March 2011.

[4] G.Navarro,T.Kida,M.Takeda,A.Shinohara,S.Arikawa, "*Faster Approximate String Matching over Compressed Text*",IEEE conference on Data Compression (DCC),August, 2002.

[5] Mamta Sharma,"*Compression Using Huffman Coding*",International Journal of Computer Science and Network Security(IJCSNS), VOL.10 No.5, May 2010.

[6] Priya jain,Shikha Pandey, "*Comparative Study on Text Pattern Matching for Heterogeneous System*", International Journal of Computer Science & Engineering Technology (IJCSET), Vol. 3 No. 11 , pp: 537-543, Nov 2012.

[7] S.RAJESH,S.PRATHIMA,Dr.L.S.S.REDDY,"*Unusual Pattern Detection in DNA Database Using KMP Algorithm*", International Journal of Computer Applications, Vo;.1, No. 22, pp: 1-5, 2010.

[8] Panwei Cao, Suping Wu, "*Parallel Research on KMP Algorithm*", IEEE International Conference on Consumer Electronics, Communications and Networks (CECNet), May, 2011.

[9] Lei Chen, Shiyong Lu, Jeffrey Ram, "*Compressed Pattern Matching in DNA Sequences*", Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB 2004).

## Authors Profile

*Mr. G Dutta* pursed Bachelor of Technology in Computer Science & Engineering from Narula Institute of Technology, Kolkata in 2011 and Master of Technology in Computer Science & Engineering from Institute of Engineering and Management, Kolkata in 2013. He is Currently working as an Assistant Professor in the department of Computer Science & Engineering at Brainware Group of Institutions, Barasat Kolkata. His main research work focuses on Bioinformatics,Cloud Computing. He has 5 years of teaching experience.

*Ms. A Mukherjee* pursed Bachelor of Technology in Computer Science & Engineering from B.P.Poddar Institute of Management & Technology, Kolkata in 2011 and Master of Technology in Computer Science & Engineering from Institute of Engineering and Management, Kolkata in 2013. She is Currently working as an Assistant Professor in the department of Computer Science & Engineering at Brainware Group of Institutions, Barasat Kolkata. Her main research work focuses on Bioinformatics, Image Processing. She has 5 years of teaching experience.