ISSN:0975-9646

Sanil Shanker KP / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (4) , 2014, 5426-5428

Locating Sequence Blocks using Logical Match

Sanil Shanker KP

Dept. of Computer Science ,
Farook College, Kozhikode,India.

Abstract- In 2011, Sanil et al put forward the concept of Logical Matching Strategy. This technique helps to locate repeating blocks of sequential pattern of finite length. The paper explains the experimental verification of the logical match by using the data of various trinucleotide repeat diseases.

Keywords- Logical Matching Strategy, Sequence Pattern, Trinucleotide Repeat, Tandem Repeat.

I. INTRODUCTION

Pattern matching in DNA sequence consisting of four characters A, C, G and T is one of the prominent applications of string matching [1]. A very important problem in computational biology is to locate the repeating blocks of sequences[2,3]. A tandem repeat in a genomic sequence is a string of nucleotides which is characterized by a certain motif (sequence pattern) followed by at least two copies of the motif[1,2]. Molecular biological investigations into trinucleotide repeats have revealed pathogenesis of various disease models in humans[5,6]. These diseases, including Friedrich's ataxia, Huntington's disease, Fragile X mental retardation and Kennedy's disease are the result of a dramatic increase in the number of copies of a tri nucleotide pattern. This paper explains the way to locate the exact positions of the repeating sequence blocks in the text sequence of the trinucleotide repeat disease using the concept of logical match[4].

II. LOGICAL MATCHING STRATEGY

The Logical Matching Strategy is based on the concept of string matching. In this method, the characters in the sequence pattern are pre-processed to generate the indices. The information from pre- processing phase is used to match the indices of pattern with those of the text. The technique is explained in the following example with random data.

Example

In the simulation, the method is demonstrated with random data, where the text is known data, and the pattern is the data to be used as the search query.

Text=> CGTACCTCGAATCGA
Pattern => TCGAA
n = 15, m = 5

Pre-processing Phase:

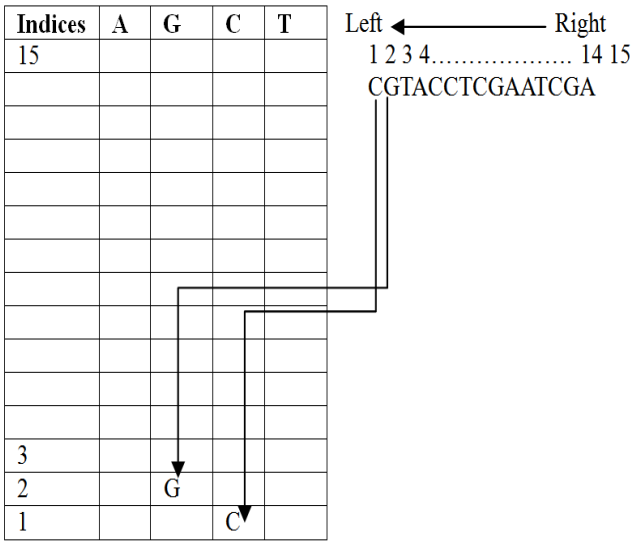
To generate indices of text, construct a table in which each column is used for storing different alphabets in the text. Shift the Text from right to left so that each alphabet coincides with its corresponding index in its respective column (Fig.1).

From the Fig. 2, the indices of the Text can be arranged as,

Indices of the Text => < A (4, 10, 11, 15);
G (2, 9, 14);
C (1, 5, 6, 8, 13);
T (3, 7, 12) >

Similarly the indices of the Pattern can be generated by shifting the alphabet from right to left and can be stored in the respective columns in the table(Fig. 3).

Fig.1



This way the indices corresponding to each alphabet in the mentioned texts are given in Fig. 2

Fig. 2: Input Text

Indices of the input text	**A**	**G**	**C**	**T**
15	A			
14		G		
13			C	
12				T
11	A			
10	A			
9		G		
8			C	
7				T
6			C	
5			C	
4	A			
3				T
2		G		
1			C	

The indices of the Pattern obtained can be written as

Indices of the Pattern=> < A (4, 5);
G (3);
C (2);
T (1) >

Fig. 3: Input Pattern

Indices of the Pattern	**A**	**G**	**C**	**T**
5	A			
4	A			
3		G		
2			C	
1				T

Match the indices of Pattern with the indices of Text:
Select the lowest index value alphabet in the Pattern and then the indices corresponding to the same alphabet in the Text have to arrange in a row. That is, the lowest index value alphabet in the Pattern is T and the index value is 1. The indices of T in the Text obtained from Fig. 2 are 3,7,12. It can be written as,

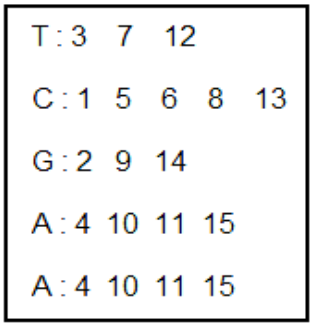
T: 3 7 12
In the same way the next higher index value for the alphabet in the Pattern is 2 and the respective alphabet is C. The indices of C in the Text are,
C: 1 5 6 8 13

The next higher index value in the Pattern is 3 and the alphabet is G, the corresponding indices of the alphabet G in the Text are given by
G: 2 9 14

A is the next higher index value alphabet in the Pattern and the indices of A obtained from the Text are
A: 4 10 11 15

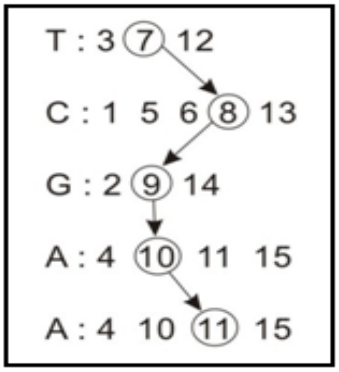
The highest index value 5 in the Pattern also corresponds to the alphabet A and the respective indices in the Text are
A: 4 10 11 15

Hence the matching indices of all the alphabets in the Pattern are given by,

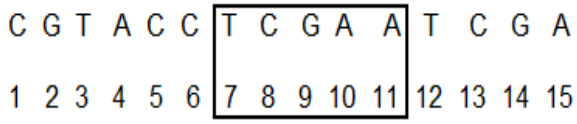


To locate the Pattern in the Text, consider the lowest index value of the alphabet T, that is 3, then check whether the next higher value of 3, that is 4, exists there among the indices of C or not. If 4 is there in the indices of C, then check whether the next higher value 5 is there in the indices of G or not. If 4 is not an index value of C, then take the next higher index value of T, 7 and check whether the next higher value 8 exists in C or not. If exists then search for the next value 9 in the indices of G. If 9 exists among the indices of G, then proceed to the next alphabet A to locate the next higher index value 10. If 10 is one of the indices for A then find whether 11 is included in the indices of the last alphabet A. In the same way, take the highest indices value of T, 12 and search for the next higher index value in C and so on.
The location of the Pattern in the Text is the one in which all the alphabets satisfy the consecutive index values, that is, the location of the Pattern TCGAA in the Text CGTACCTCGAATCGA is given in Fig. 4,

Fig. 4: Location of the Pattern in the Text



The Pattern TCGAA is located in the Text CGTACCTCGAATCGA in the locations 7 8 9 10 11. That is the Pattern occurs in the Text location 7.



III. EXPERIMENTAL RESULT

The experimental verification of the method, logical match was done by using the data from NCBI databank for the trinucleotide repeat diseases. For simulating the method, the program has been written in C language under Linux platform. The output of the simulation results for Friedreich's ataxia, Huntington's disease, Fragile XA syndrome and Kennedy's disease are given in Table- 1. The locations of the pattern in the text for each disease are conveyed through the Figures 5- 8.

IV SUMMARY

This method helps to find the exact locations of motif in trinucleotide repeat diseases. The experimental verification of the logical match was done by using the data of various trinucleotide repeat diseases such as Friedreich's ataxia, Huntington's disease, Fragile XA syndrome and Kennedy's disease. This method finds application in sequence analysis in locating biologically meaningful segments.

Table 1: Locations of tandem repeat for various.

Diseases	***Tandem repeat locations***
Friedreich's ataxia	1976-1979, 2184-2208
Huntington's disease	196-256, 968-971, 1126-1132, 1291-1294
Fragile XA syndrome	13762-13765, 13786-13792, 13832-13853
Kennedy's disease	1286-1349, 1370-1385, 1553-1556

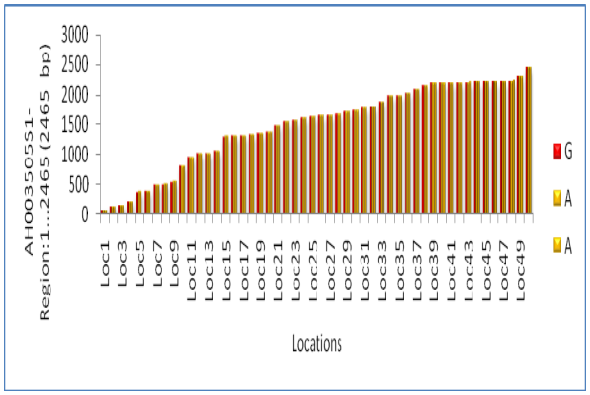


Fig. 5: Friedreich's ataxia (Locus: AH003505S1, Region: 1...2465 (2465 bp))

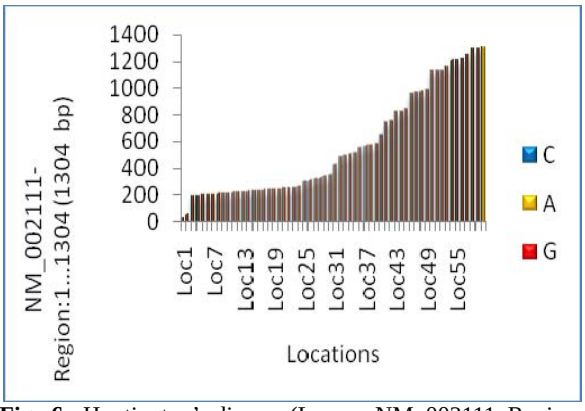


Fig. 6: Huntington's disease (Locus: NM_002111, Region: 1...1304 (1304 bp))

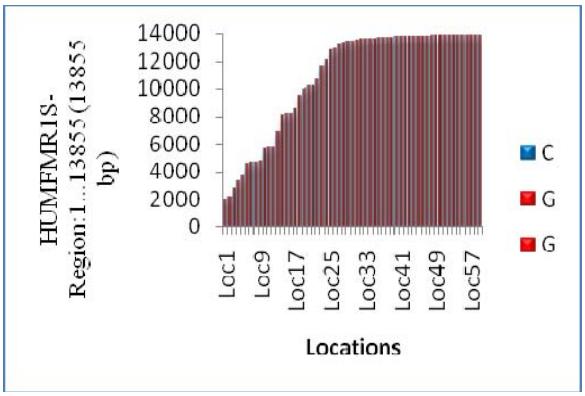


Fig. 7: Fragile XA syndrome (Locus: HUMFMR1S, Region: 1...13855(13855 bp))

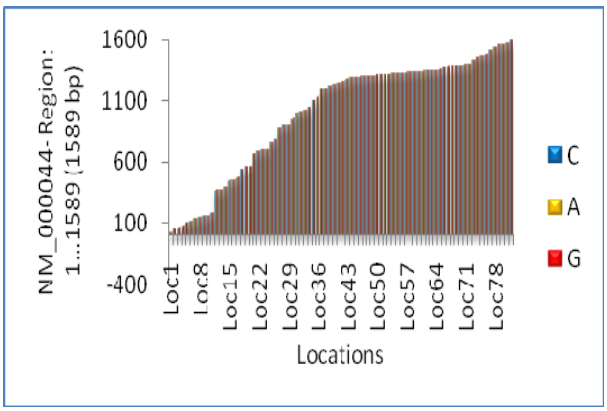


Fig. 8: Kennedy's disease (Locus: NM_000044, Region:1…1589 (1589 bp))

REFERENCES

[1] Gusfield D, Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press 1997.

[2] Pevzner P A, Computational Molecular Biology: An Algorithmic Approach, MIT Press , Cambridge, MA, 2000

[3] Pevzner P A and Waterman M S, Open Combinatorial problems in computational molecular biology, Proc. Third Israel Symp. Theo. Comp. Syst. IEEE Computer Society Press, (1995) 158 – 173.

[4] Sanil S K P, Elizabeth S and Austin J, A note on two applications of Logical Matching Strategy, Applied Artificial Intelligence, 25 (2011) 708–720

[5] Epplen J T and Gencik M, Trinucleotide Repeat Expansions: Mechanisms and Disease Associations, Nat. Ency. Hum. Gen., (2003) 634- 638.

[6] Everett C M and Wood. N W, Trinucleotide repeats and neurodegenerative disease, Brain, 127 (2004) 2385- 2405.