ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

RS-A Fast Pattern Matching Algorithm for Biological Sequences

K.K.Senapati, Sandip Mal, G.Sahoo

Abstract*— Searching a pattern in biological sequence is as good as finding a genetic code in a sequence of DNA. As every day the new gene sequences discover, it increases frequently the biological database. Till today many approaches started from Brute Force to Berry-Ravindran applied for finding pattern. Biological sequence database is too large. So an efficient pattern matching algorithm is needed to search the pattern. So many pattern matching algorithms are already developed but till today research is going on to find a new approach as to reduce the searching time. Berry –Ravindran bad character algorithm calculates bad character shift using two consecutive characters in the text immediately to the right of the window. But in our proposed algorithm, instead of two consecutive characters we introduced four consecutive character .Searching performed from the right to left of the sequence, it reduces the searching time complexity. In this technique bad character shift value calculated by RS bad character shift function. As using four variables instead of two variables, the searching time is less. The best case, worst case and average case time complexities of the new algorithm are also discussed.

***Index Terms*—Pattern Matching, RS algorithm, Biological Sequences.**

I. INTRODUCTION

Pattern matching is an efficient technology in computer knowledge. It is widely applied in text editing, literature searching, natural language identifying, biology, image manipulation, information security, etc. There are two types of pattern matching: single pattern matching and multiple patterns matching [7]. In simple single pattern matching algorithm, a pattern matching procedure is looked as key word (pattern string) searching. It divides the text with the length n into n-m+1 sub-string with the length m and detects whether each sub-string matches to pattern string with length m or not. One of the most important research area and has also been studied in the decade of computer science over the years, pattern matching has extensively been applied various computer applications, for example, in retrieval of information, information security and searching nucleotide or amino acid sequence patterns in biological sequence [3] databases. Classical single pattern matching algorithm includes KMP algorithm [2] and BM algorithm [2]. Other algorithm [2] is mostly improved on these classical algorithms In BM algorithm, pattern string moves from left to right, while character is compared from right to left. When it loses matching, the predefined excursion function adopts the max value to decide right shift value. The efficiency of an algorithm depends on two phases: the pre-processing phase and the searching phase. The characters in the pattern are preprocessed in the pre-processing phase and this information is used in the searching phase in order to reduce the character comparisons, which in turn reduces the overall execution time. The aim of the good algorithm is to minimize the work done during each attempt and to maximize the length of the shifts. In this paper, an efficient algorithm called RS is proposed. The proposed algorithm though, not linear, achieves good results. The rest of the paper is organized as follows. In section II describes the overview previous algorithm. Section III describes the proposed RS algorithm in detail. Section IV describes the working example and Section V discusses Results and Comparison previous and RS algorithm, followed by the conclusion.

II. PREVIOUS WORK

Berry-Ravindran algorithm [6] calculates the shift value by considering bad character shift for the two consecutive text characters in the text immediately to the right of the window. These shift values computed are used in searching phase to reduce the number of character comparisons, thus improves the performance of the algorithm.

In Two Sliding Window method [8], in the right of window they used an algorithm for finding pattern. Their uses two consecutive variable immediately left after the pattern. Now the work is to modify that algorithm, as to maximize the efficiency named as RS. We use four variables for calculating shift value when mismatch occur instead of two. As a consequence the function is more complex but function calculates shift value only in preprocessing phase. In the searching phase it will take fewer searches for finding the pattern in the text. The proposed algorithm has been tested with the previous algorithms for different pattern in the same text. This proposed algorithm performs better in some pattern and some it performs equal.

III. THE PROPOSED ALGORITHM

A. Pre-processing Phase:

The proposed algorithm RS four consecutive characters immediately after the pattern are considered, which scans from right to left. MBrBc [9], that scans from left to right. Initially, the indexes of the four consecutive characters in the text string from the right are (n-m- 4), (n-m-3), (n-m-2) and n-m-1) for a, b, c and d respectively in Eq. 1. The RS (a, b, c, d) of the algorithm consists in computing for each four characters a, b, c, d for all a, b, c, d,∑, the left most occurrence of abcd in the pattern.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

The RS [a, b, c, d] is defined as follows RS [a, b, c, d] =min

$$\begin{cases} m+3 & \text{if } p[m-1]=a \\ \vdots & \\ m+2 & \text{if } p[m-1]=a \end{cases} \quad (1)$$

In searching phase after each attempt, the shift of the window is calculated by using the RS[a, b, c, d] function in Eq. (1) and the maximum shift of the window is provided when these four characters abcd does not occur in the pattern. The probability of four characters present in the pattern becomes less when the alphabet size is big.

In this proposed algorithm, after each attempt the window is shifted to the left using the shift value computed for the four consecutive characters immediately left of the window. RS function calculates the shift value based on the left most occurrences of four consecutive characters, say abcd, which is immediately to the left of the window. The probability occurrence of four consecutive characters, abcd, in the pattern as compared to that of ab is less. Thus RS always provides a better shift than BrBc (Berry-Ravindran Bad character function).

It concludes from the above that the pre-processing phase helps in searching phase to improve the overall efficiency of the algorithm.

B. Searching Phase:

The procedure of the proposed algorithm is performed from right to left until a complete match or mismatch occurs. Whenever a mismatch occurs, the RS shift value is used to shift the window to the left. This procedure is repeated until the window is placed beyond (m-1) from the left of the text.

IV. WORKING EXAMPLE

In the context of the algorithm an example is elaborated the process of the RS algorithm. Let the

Pattern (P) =”GAATCAAT”, m=8
Text (T) =”CAATCTAACATCATAACCCTAAT
TGGCAGAGAGAATCAATCGAATCA, n=47

A. Preprocessing Phase:

Initially, shift=m+4=12.

The shift values are stored in two arrays next as shown in Fig. 1. To build the next array (next), consider four consecutive characters of the pattern and give it an index starting from 0. For example for the pattern structure GAATCAAT, the consecutive characters GAAT,AATC,ATCA,TCAA and CAAT are given the indexes 0,1,2,3, and 4 respectively. The shift values for the next array are calculated according to Equation 1.

Where in the pattern G A A T C A A T indexed as

G A A T -> 0
A A T C -> 1
A T C A -> 2
T C A A -> 3
C A A T -> 4

Shift value from left					
Index	0	1	2	3	4
Next					
	4	3	2	1	$\begin{cases} 1 & \text{if } p[m-2]p[m-1]=ab \\ \vdots & \\ 1 & \text{if } p[m- \end{cases}$

Fig.1. The next array

Between texts character (T) and pattern character (G).

B. Searching Phase:

The searching process for the pattern P is described as follows (Fig. 2).

First attempt: In the first attempt (Fig. 2a), aligning the sliding window with the text from the right. In this case, a mismatch occurs.

- Accordingly to RS function the shift value calculated. i.e shift1=3.
- These consecutive four characters pattern is not present in the pattern. So the process doesn’t take shift value from next array.
  Therefore the window is shifted to the left 3 steps. For the RS function gives Shift2 value is 2.
- These consecutive four characters pattern is not present in the pattern. So process doesn’t take shift value from next array. The window is shifted to the right 2 steps. Now the characters from the text at index 35, 36, 37 and 38 which are (T, C, A, A) respectively. The calculation of shift (shift1) process is as follows:
- The process find accordingly to RS function. It calculates shift1=3.
- These consecutive four characters pattern is not present in the pattern. So process doesn’t take shift value from next array. Therefore the window is shifted to the left 3 steps.

Second attempt: In the second attempt (Fig.2b), aligning the sliding window with the text from the right after shift. In this case, a mismatch occurs between text character (C) and pattern character (G), therefore process takes the four consecutive characters from the text at index 32,33, 34 and 35 which are (G,A,A and T) respectively. The calculation of shift (shift2) process is accordingly to RS.

Third attempt: In the third attempt (Fig. 2c), aligning the sliding window with the text from the right after shift. In this case, a mismatch occurs between text character (A) and pattern character (G), therefore process takes the four consecutive characters from the text at index 30, 31, 32 and 33which are (G, A, G and A) respectively. To determine the amount of shift (shift3) process follows the following steps:

- The process find accordingly to RS function. It calculates the value 2.
- These consecutive four character pattern is not present in the pattern. So we don’t take shift value from next array.

Therefore the window is shifted to the right 2 steps.

Fourth attempt: In the fourth attempt (Fig. 2d), aligning the sliding window with the text from the right after shift. In this case all the character of the pattern match with the text (index 32 to 39). So match performed and not more search required.

ISSN: 2277-3754
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 1, Issue 3, March 2012

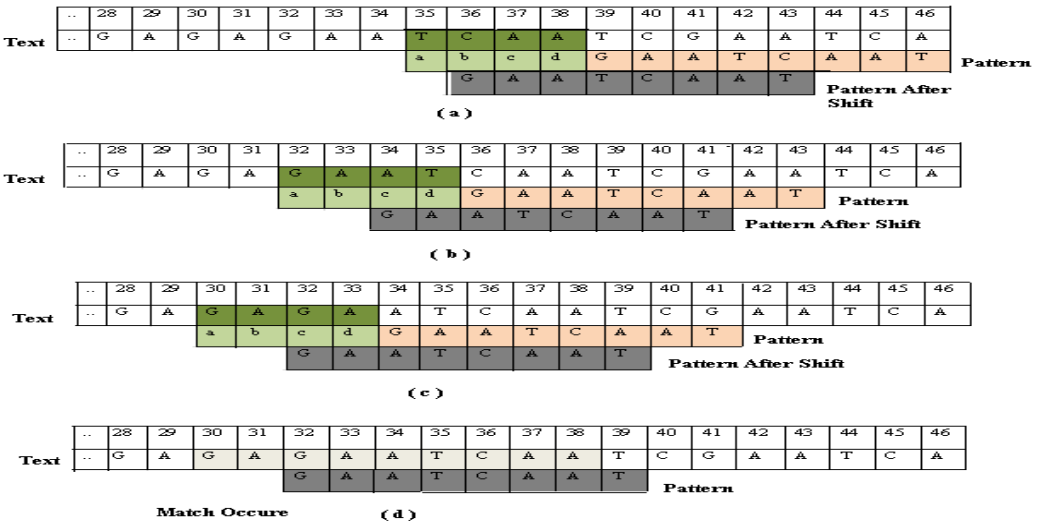

Fig.2.Working Example

V. CONCLUSION

In this paper, we presented an efficient pattern matching algorithm, called RS, which is a new dimension of pattern matching algorithm. It concludes that this algorithm works well for the large patterns with small alphabet size.(i.e. pattern "GGGTTTATGAACTTC" with alphabet size only four G,A,T,C.) This algorithm has been proposed for exact pattern matching, where in the shift value is maximized by modifying the Berry-Ravindran BrBc [6],[9] function. Therefore this algorithm can be implemented in all applications related to exact pattern matching in biological sequence databases.

REFERENCES

[1] Boyer, R.S. and Moore, J.S., "A fast string searching algorithm.Commun". ACM, 20, 762-772, 1977.

[2] Charras, C. and Lecroq, T., Handbook of Exact string matching algorithms available at the website: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.133.4896&rep=rep1&type=pd

[3] R. Thathoo, A. Virmani, S.S. Lakshmi, et al, "TVSBS: A fast exact pattern matching algorithm for biological sequences", CURRENT SCIENCE, vol. 91, no. 1, pp.47-53, 2006.

[4] Lecroq, T., 1995. Experimental results on string matching algorithms. Software practice and Experience, 25: 727-765.

[5] String searching algorithm. Commun. ACM. 20:762-772. DOI:10.1145/359842.359859.

[6] Berry, T. and Ravindran, S., A fast string matching algorithm and experimental results. In Proceedings of the Prague String logy Club Workshop '99 (Eds Holub, J.and Simanek, M.), Collaborative Report DC-99-05, Czech Technical University, Prague, Czech Republic, 2001, pp. 16–26.

[7] S Wu and U Manber, "A Fast Algorithm for Multi Pattern Searching", Technical Report TR-94-17, University of Arizona, 1994.

[8] Amjad Hudaib, Rola Al-Khalid, Dima Suleiman, Mariam Itriq and Aseel Al- Anani" A Fast Pattern Matching Algorithm with Two Sliding Windows (TSW)", Department of Computer Information Systems, University of Jordan, Amman 11942 Jordan, Journal of Computer Science4 (5): 393-401, 2008, ISSN 1549-3636

[9] K.K.Senapati, G.Sahoo, S.Sahana" An Efficient pattern matching algorithm for biological sequence". Proceedings of the International conference on Image processing, Computer Vision and Pattern Recognition (IPCV2010), VOL-II, PP-755-759, LasVegas, USA.

AUTHOR BIOGRAPHY

Kisore Kumar Senapati has obtained his M.Sc. degree in Mathematics & M.Tech in Computer science from UTKAL University, Odisha, India. He has worked for one year in an Industry at Hyderabad. He has obtained CCNA from IIIT Hyderabad and CCI (Cyber Crime Investigator) at ASCL Pune. Since 2002 he is in teaching profession. At present he is working as an Assistant Professor in CSE department in Birla Institute of Technology, Mesra, Ranchi, India. He has published and presented many papers in national and international conferences in India and abroad. He has guided many students for their Master Thesis and Project. His area of interest is algorithm design for Pattern matching, digital forensic and bioinformatics.

Sandip Mal has completed his B.Tech in Computer science and Engineering from Bankura Unnayani Institute of Engineering under West Bengal University of Technology in year 2008. At present he is a student in M.E Software Engineering in Computer Science and Engineering, Department in Birla Institute of Technology, Mesra, Ranchi, India. His research interest includes image processing and pattern recognition.

G. Sahoo received his P.G. degree from Utkal University, Orrisa in the year 1980 and Ph.D degree in the area of Computational Mathematics from Indian Institute of Technology, Kharagpur in the year 1987. He is associated with Birla Institute of Technology, Mesra, Ranchi, India since 1988. He is currently working as a professor and heading the Department of Information Technology. His research interest includes theoretical computer science, parallel and distributed computing, data security, image processing and pattern recognition.