Pattern Segmented Positioning For Hasty Matching Using Dynamic Size Sliding Window

Jayanthi Manicassamy and P. Dhavachelvan

***Abstract*—- Merging computational biology with few other sciences for solving real world biological problems involves, various high throughput techniques for better decision making. Sequence analysis plays major role in identifying the type of disease suspected from the given gene expression through pattern matching. Which is also used in analyzing proteins structural differences, motif search, text mining etc… Performance plays a major role in extracting matched patterns from a given sequence or from a database independent of the size of the sequence. To extract pattern match from a large sequence from a database it takes more time in order to reduce searching time we have proposed an approach that reduces the search time with accurate retrieval and positioning of the matched pattern in the sequence. This algorithm makes use of window sliding technique in which, incorporation of segmenting the whole pattern that is to be searched is done in order to reduce the search time. Each segmentation pattern is taken for pattern match in a particular sequence starting from the first segmentation if a match found next segmentation will be checked in combination with the previous segmentation. This process takes place in multiple sequences until retrieving the exact pattern sequence or sequence related information's from the database. Matching a pattern in a sequence or searching a sequence itself, independent of the sequence size or any number sequence that exists. This algorithm is considered to be advanced compared to that of sliding window technique, used for analyzing the exact pattern from the required sequence.**

***Index Terms*— Algorithm, Bioinformatics, Databases, Pattern matching, Sequence Analysis, Text Mining.**

I. INTRODUCTION

Bioinformatics is a multi disciplinary science that uses methods and principle, from mathematics and computer science for analyzing biological experimental data's where sequence analysis plays a vital role, for various analyses like discrimination of cancer from the gene expression, mutations evolution, protein-protein interaction in cellular activities etc… In area of research pattern matching is a pivotal theme in various applications in computational biology for data analysis like feature extraction, searching, disease analysis, structural analysis etc… which also has slow impact on other areas of application development [10]. The main aspects involved in pattern matching in a sequences is discrimination of diseases evaluated from the gene expression, Sub-cellular localization [2] from experimental data through protein pattern matching etc… Researches have been carried out for finding conserved patterns for signification mutations detection and for 3-D structures of protein interaction in cellular activities. However, pattern matching is not only making sufficient influences in sequence analysis but it also has to meet its demand in many areas.

Pattern matching focuses on finding the occurrences of a particular pattern of in a text. Generally, pattern matching algorithms make use windowing technique using whose size is equal to the pattern length. The searching process starts by aligning the pattern to the left end of the text and then the corresponding characters from the pattern and the text are compared. Character comparisons continue until a whole match is found or a mismatch occurs. Some pattern matching algorithms concentrate on the pattern itself [7]. The performance of the algorithms can be enhanced when comparisons are done in a specific order [6].

In this article we have present an enhanced algorithm that reduces the time of extraction of sequence for the matched pattern based on the search made in the respected selective database in spite of taking into consideration of the pattern and the sequence. It makes use of single dynamic size window. Basically window size will be based on the segmentation taken from the pattern to be searched. Size of the window will be incremented if match found by adding the next series segment of the pattern and the process of pattern matching will be continued. To reduce the number of comparison and comparison process time primarily the pattern to be searched is segmented.

II. Pattern Segmented And Matching Using Dynamic Sliding Window

In this section, explanation of how the new algorithm works in order to reduce the time of retrieval of matches of the related pattern or searching for the existence of that particular pattern. However the size of the pattern to be searched or the size of the sequence where the pattern match has to be retrieved. This search of extracting the exact pattern match could be carried out on multiple sequences. This algorithm is considered to have an advantage over the other current algorithms of sliding window. Pattern matching and recognizing is mostly used in the area of bioinformatics to identify the disease level, for finding structural differences etc…

This segmented positioning for pattern matching using dynamic windows size consists of three major phase that are to be carried out they are Segmentation phase, Identification phase and Extraction phase. Segmentation phase involves searched pattern segmentation depending on the size of the pattern. Identification phase is used to find the partially

Jayanthi Manicassamy and P. Dhavachelvan are from Department of Computer Science, Pondicherry University, Kalapet, Pondicherry, India.

matching sequence from the database. Finally the extraction phase process involves in retrieving the exact matched sequence with positional match information's.

A. *Segmentation*

Sequence (S) or pattern is taken as input to extract the matched pattern or any information from the database. Since the main aim of this approach is to reduce the time of for pattern and give an accurate result in better time. The main reason why the search time is more is due to the large size (S_L) of the pattern is to be search. Check if the pattern to be recognized is of fixed length that of maximum of limited size (L_S).

Let's consider for example sequence S

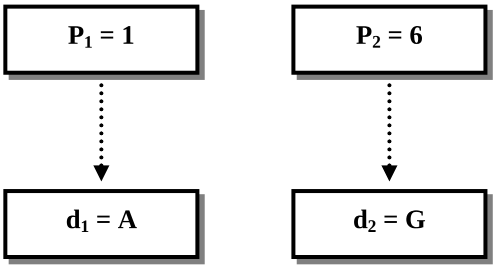
ACGTCGTAGG

The Sequence should be a combination of "A, C, G, T" since amino acid is a combination of gene expression. $S_{L\,=}$ 10, L_S = 5. The limited size length L_S depends on the users that are going to develop applications using this algorithm. The sequence and size can of any length. If the S_L very large them segment (s_i) S of L_S where s represents segment. Here i = 1, 2…n. Since S_L is 10 then segment S. Here by considering the taken example

s_1 = ACGTC

s_2 = GTACG

Store the first position of segment actual position of S in P_i of the segments and its data value in d_i where P_1 = 1 contains d_1 = "A" and P_2 = 6 contains d_2 = "G".



B. *Identification*

In this step d_i will be searched in sequence where to be checked for pattern match or in database for the sequence DS_i. Taking the first sequence into consideration first d_1 will be searched DS_i and stores its occurrence position in a_1 and in the same ways d_2 will be searched in stored in a_2 (*d_i will be searched for position and will be stored in a_i where a, is an array*) check for value (V) = comparison of two array difference. = (*positional value of the second array*) or (P_i) i.e. (*$array_1$ – $arrary_2$ = P_2*) by now d_1 is taken and made a match comparison in different sequences.

Let us consider DS_1 as

ACGTCGTACG

Which is to be checked for pattern then data1 will be searched and its positional value will be stored in a_1 (0, 7), and the taken data2 performs the same operation and store in a_2 (5, 9). v = a_1 (position i) – a_2 (position i) = P_2 store this value in SV = Position i since if the sequence doesn't match the comparison should be carried out from the next position onwards if v = P_2. Then starting search position Sp = a_i of s1 (considering the taken example Sp = a_1), Search ending position Se = a_i of s_2 (Consider the example taken Se = a_2). Since v = P_2 when the value of a_1 and a_2 (0, 5) the Sp = a_1 (Sp =0) and Se = a_2 + L_S (where Se = 5 + 5 = 10).

C. *Extraction*

In this step pattern extraction and information retrieved by considering Sp and Se where comparison should be made with S and DS_1 for pattern match by position wise, if there is complete match found that Pattern match score = 0 else 1. Pattern Match score = Σ all Matches score. Matches score of position i = 0 if match found the value of i increases based on the positional increment.

This algorithm could be followed for any length of sequence that could be searched from any database. The segmentation can be of any number depending on the size of the pattern that has to be recognized, based on which the sequence segmentation will be incremented.

III. Conclusion

As new techniques and approaches invokes day to day in computation biology for various reasons sequence analysis and text mining plays a major role. Taking consideration of time duration is important in spite of considering the requirement for development of high throughput technique for better decision making. As the sequence grows for pattern recognition the run time becomes more in searching and extracting the accurate information to avoid this we have specified an enhanced algorithm in this article.

The concept of searching the sequence by means of pattern segmentation and by using dynamic window provides a bit faster technique to find the required pattern from the database giving a preference over other techniques in the number of comparison. This algorithm could also be kept as an outline for text mining to extract or for finding the required information from the required database.

References

[1] Yong Huang, Lingdi Ping, Xuezeng Pan, Guoyong Cai, "A Fast Exact Pattern MatchingAlgorithm for Biological Sequences", IEEE digital library, 2008, pp 8-12.

[2] Charras, C. and T. Lecroq, Handbook of Exact String Matching Algorithms. First Edition. King's College London Publications. ISBN: 0954300645. 2004.

[3] David He and John Parkinson, "SubSeqer: a graph-based approach for the detection and identification of repetitive elements in low-complexity sequences", ACM Portal, Feb 2008, pp 1016–1017.

[4] Richard J. Edwards, Norman E. Davey and Denis C. Shields, “CompariMotif: quick and easy comparisons of sequence motifs“, ACM Portal, March 2008, pp 1307–1309.

[5] He, L., F. Binxing and J. Sui,. The wide window string matching algorithm. Theor. Compu.Sci., 332: 391-404. DOI: 10.1016/j.tcs.2004.12.002, 2005.

[6] Amjad Hudaib, Rola Al-Khalid, Dima Suleiman, Mariam Itriq and Aseel Al-Anani, “A Fast Pattern Matching Algorithm with Two Sliding Windows (TSW)”, Journal of Computer Science, 2008, pp 393-401.

[7] Charras, C. and T. Lecroq, “Handbook of Exact String Matching Algorithms”, First Edition. King’s College London Publications, ISBN: 0954300645, 2004.

[8] Smyth, W.F., “Computing Patterns in Strings”, First Edition. Pearson Addison Wesley. United States. ISBN: 978-0-201-39839-7, 2003.

[9] Chung-Chih Lin, Yuh-Show Tsai, Yu-Shi Lin, Tai-Yu Chiu, Chia-Cheng, Jeremy C. Simpson and Chun-Nan, “Boosting multiclass learning with repeating codes and weak detectors for protein subcellular localization”, ACM Portal, Sep 2007, pp 3374–3381.