Mining Approximate Circular Pattern

MS. DEVYANI SHARMA[1], PROF. DR. S. M. KAMALAPUR[2]

[1]Assistant Professor, Dept of IT, Alamuri Ratnamala Institute of Engineering and Technology, Shahapur, Thane, Maharashtra, India,

[2]Associate Professor, Dept of Computer Engineering, K.K.Wagh Institute of Engineering Education & Research, Nasik, Maharashtra, India.

Abstract: Pattern matching is the process of finding occurrences of a pattern string in a text or character string. Pattern matching is the basic requirement in many applications such as spell checking, spam filtering, geometry, network traffic data, DNA and protein sequences, etc. Circular Pattern occurs in the DNA of bacteria, viruses, archaea, etc. A linear string having length n can be considered as a circular string, in which end symbols are connected thus forming a circular string. The circular string is considered as n different linear strings, all will be considered equal. Nowadays biologists are interested in finding the approximate patterns. Approximate pattern matching is the technique of finding pattern string that match a pattern approximately rather than exactly. In practical experiments errors may occur due to natural mutations in the DNA of virus, practical limitations of the lab equipments that may introduce errors, etc. Due to these reasons we need to find the approximate circular patterns. The proposed system aims to find the presence of approximate circular pattern with k number of mismatches from the input stream. If the approximate circular pattern exists the system returns the pattern. Also the proposed system finds the presence of any pattern in the input stream.

Index Terms: Circular DNA sequences, Circular Pattern Matching, data stream processing, Pattern Recognition.

I. INTRODUCTION

Circular pattern matching (CPM) has application in many areas like astronomy, computational biology, geometry. In the previous work, Gusfield proposed a technique to find circular string, SimpLiFiCPM algorithm is proposed based on filtering technique to deal with exact circular pattern matching. For cyclic sequence pair wise and multiple circular sequence alignment are identified. But in case of DNA sequence circular pattern represents the virus. This virus sequence can be modified by adding some errors in the sequence. These errors may arise due to reasons natural mutations in the DNA or lab equipment errors. Hence to identify virus of such pattern circular pattern identification with approximation is required rather than exact matching.

The circular pattern matching problem consists in finding all occurrences of the rotations C (P) of a pattern P of length m in a text T of length n. In approximation technique C (P) is k matches of characters with given text T.

Due to error occurrence in DNA sequence exact circular pattern matching may skip some important patterns in the sequence. To allow some errors in locating occurrences of pattern approximation matching technique is required. To allow fast and efficient solution for pattern matching search space reduction is

required. Search space reduction can be achieved using some filtering technique and hence filter based circular pattern matching with approximation is the domain of work.

In the below sections we are going to discuss about related work done for the proposed research area. We refer some existing research paper for completing this task. It is given as follow:

II. RELATED WORK

An approximate circular pattern matching (ACPM) problem, appears as an interesting problem in many biological contexts have been proposed in [1] and [5]. To address the problem of ACPM a simple and fast filter-based algorithm is proposed. The proposed algorithm runs twice as compared to state of the art algorithms. ACPM is an unusual problem in biology. It mainly consists of identifying complete circumstances of the rotations of patterns in the text of n length. SimpLiFiCPM is a simple and lightweight filter-based algorithm used to solve CPM problem. It is mincing solution with quadratic complexity. After having built the set of rotations of patterns searching is performed on finite set of string using classical algorithm. In first phase, it consists of pattern preprocessing by developing suffix automaton of the string. They were presented an optimal average-case algorithm for CPM. SimpLiFiCPM is proved as, an extremely fast algorithm based on 6-filters. In the context of sequence alignment, circular strings have been studied for pair wise and multiple circular sequence alignment searching. Results of all these proposed algorithms improved in additional stage of pre-processing. Pre-processing step also speed-up time of algorithm execution. Hamming distance algorithm is presented as an efficient algorithm for finding the optimal alignment and consensus sequence of circular sequences on this distance metric. SimpLiFiCPM is also referred as, SFF algorithm.

A simple filtering approach which has extraordinary number of applications in string matching algorithms has been discussed in [3]. It is efficient to solve multiple string matching as well as several approximate matching in average optimal time. To resolve the problem of multiple string matching a n-

bitparallelism or the classic AhoCorasick automaton algorithms are used. To achieve the optimal running time on average, for short patterns they modified the well-known bit-parallel algorithm to Shift-Or.

Pattern matching problems are based on either bitparallelism or the classic AhoCorasick automaton. An idea of nave or brute force algorithm is applied while the obtained algorithm is arguably one of the simplest known characterskipping string matching algorithm. Final analysis conclude that the proposed technique can be used to generate subpatterns for the AhoCorasick multiple matching algorithms, which leads to an average-optimal algorithm without any limitation on the pattern length. Bit-parallelism and brute-force search algorithms are

used with the idea of pattern-splitting. There are different algorithms which are used for multiple string matching. Performance of all algorithms depends on choice of q where, q is character in string.

A bit-parallel algorithm for perfect circular string matching problem is described in [4]. They deal with the problem of ECSM i.e Exact Circular String Matching. They have proposed two algorithms namely, Circular Simplified Backward Nondeterministic Dawg Machine (CSBNDM) and CSBNDMq for searching a circular string on text using the bit-parallel technique. The proposed algorithm uses only the composition of bitwise-logical operations and basic arithmetic operations. ECSM has straight application in circular genomic sequence searching. CSBNDM applies two tricks, first, it employs the bit-parallel technique and still retains its simplicity, as Simplified Backward Nondeterministic Dawg Machine (SBNDM) does. Thus, many checking steps can be done at the same time by doing just a few bit-vector operations. Secondly, the bit vector rotation operation can fit the requirement for checking a circular pattern and causes almost no more overheads than SBNDM does. As a result, the worst-case time complexity of CSBNDM is O(nm2/w), and its average-time complexity is O(nlogm/w) where n is the length of the text, m is the length of the pattern and w is the number of bits in a computer word. CSBNDMq is an enhancement of CSBNDM by adding the qgram technique. Its worst-case time complexity is the same as CSBNDM; however, its average-time complexity becomes O(nlog m/m). Experimental results in this algorithm show that CSBNDM and CSBNDMq are very efficient and much faster than the theoretical time-optimal algorithm CSA for the ECSM problem on random texts and patterns. The same phenomenon can also be observed on DNA sequences.

To solve the ECSM problem, linear-time algorithms have been proposed. An approach of preprocessing on patterns, suffix automaton for PP is constructed. Based on this length of the longest factors of PP appearing at every position of T is determined. Another approach of preprocessing is by using suffix tree and then search the appearance of the circular pattern on the tree with the help of the suffix links. A novel way to solve ECSM problem is to transform the problem into the multiple string matching problem. A bit-parallel algorithm which named as, CSBNDM algorithm and it is modification of SBNDM algorithm for circular string matching. A consensus problem is to find a representative string of a given set of strings that is defined in [7]. The found string is called a consensus (string), a closest string or a center string. Finding a consensus is a fundamental problem in multiple sequence alignment.

A consensus optimizes one or more metrics such as distance sum, radius, and so on. Only sum distance and radius distance are considered in this paper. The formal definitions of the distance sum and the radius as follows. Let $S = S_1,...,S_h$ be a set of h linear (or circular) strings of equal length n and X denote an arbitrary string of length n. The Distance sum of X with respect to S denoted by ES(X) is the sum of Hamming distances from the strings in S to X, i.e.,$P_1 \leq i \leq h$ $d(X,S_i)$ where d(A,B) denotes the Hamming distance between two strings A and B. The Radius of X with respect to S denoted by RS(X) is the longest Hamming distance from the strings in S to X, i.e., max $1 \leq i \leq h$ $d(X,S_i)$. Four different types of consensus problems, CS, CR, CSR, and BSR are considered in this research work. CS is the problem of finding an optimal consensus minimizing the distance sum. CR is the problem of finding an optimal consensus minimizing the radius. CSR is the problem of s finding an optimal consensus minimizing both distance sum and radius if one exists. BSR is finding a consensus whose distance sum and radius are smaller than given thresholds. This substantial analysis solves the problem for sequential strings. The problem of CS is easy to solve. Distance sum is minimized by selecting a majority symbol in each

position of the strings in S. The problem of CR is difficult to solve. Non-trivial algorithms are proposed to find a consensus and an optimal alignment for Problems CS, CR, CSR, and BSR for circular strings.

A circular pattern matching (CPM) problem is to search all occurrences of P in T. To solve the exact CPM (ECPM) problem an algorithm with suffix links is introduced in [8]. For approximation of CPM a q-gram-based algorithm is implemented. A q-gram-based algorithm is used for bidirectional edit distance. An extended version of proposed algorithm is used to solve the all-against-all variant of the CPM problem for both exact and k-approximate matches. Main goal is to build an efficient index data structure to facilitate subsequent batch of queries as efficiently as possible. In Circular Pattern Matching Problem (CPM) two efficient data structures namely, CPI-I, CPI-II represented. CPI-I and CPI-II outputs the better time and space complexity in case of construction but suffers a little in query time. In CPI-II, to reduce the space they have used compressed suffix array [8]. The problem of pair wise and multiple cyclic sequence alignments with affine gap costs, and an extension of a recent approach is explained in [9]. It is for circular RNA folding to the computation of consensus structures. To design linear sequences of pattern become a more tedious task due to the requirements of manual corrections of both alignments and subsequent analysis. A dedicated alignment tool for (short) circular sequences, which is particularly geared towards viroids and small virus RNAs also represented. While the time complexity is higher than classical alignment algorithms, it is efficient enough in practice for use with viroid and other subviral sequences.

III. SYSTEM ARCHITECTURE

Figure 1 and Figure 2 represents proposed system architecture. In the first system the given pattern is searched in the given input stream or string. ACPS-FT algorithm is used in which six filters are used. These filters are based on simple mathematical functions. In this the window size is same as that of the pattern size. The window slides by one character to search the patterns. If the filter values of

pattern and string match that means the pattern is found. The total number of found patterns are returned at the end.

In figure 2, the existing patterns are searched in the data stream. Pattern is not given as input, but the size of pattern is given as input that means pattern of which length is to be searched is specified by the user. Also the user needs to give the time interval in seconds, so that after how much time the streaming data should be generated. ACPSFT algorithm is used to find the patterns, but in this the window does not slides by one character, instead the window slides according to the specified length of the pattern to be searched in the streaming data. In this, the patterns are stored in the database. Each of the new pattern is matched with the existing patterns in the database, if the pattern is already existing then the count of that pattern is incremented. If the pattern does not match any of the patterns in the database then it is added in the database. A log file is maintained in which filtered values and count of matched patterns is maintained.

The detailed description of each block is as follows:

Input Stream : Input stream consists of strings having characters specified by the user. Input stream is a continuous stream in which the given input pattern is to be searched. The stream is generation after a specific interval of time defined by the user.

Pattern : A pattern is a sequence of characters having length less than the input stream in which it has to be searched. It is given as input the ACPS-FT algorithm[15].

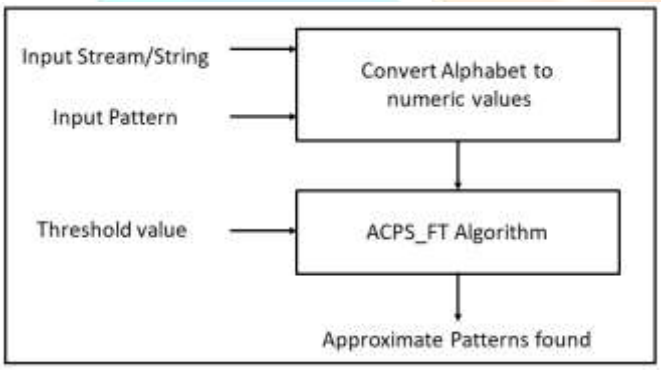

Figure 1: System architecture (Type 1)

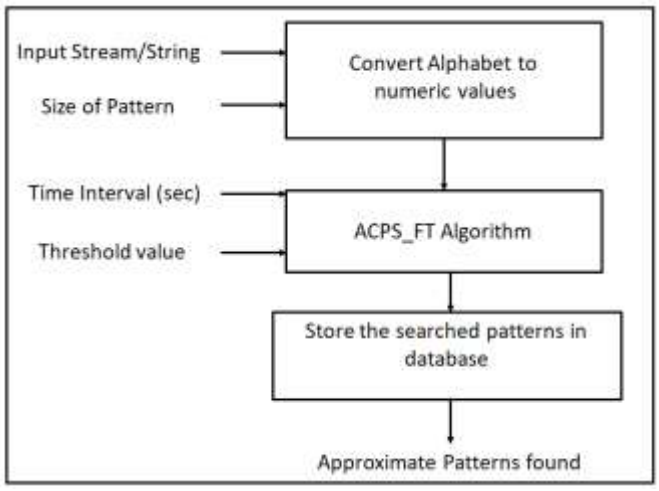

Figure 2: System architecture (Type 2)

Threshold Value : The proposed system searches the approximate circular pattern in the input stream. The threshold value is the value which decides the value of approximation of the given pattern. This value should be less than the length of the pattern. For example consider a pattern P = atcgatg, length of pattern is denoted by m = 7. So the threshold value k should be less than n. (k<m).

Convert Alphabet to Numeric Values: The characters in the input stream/string and in the given pattern are identified. The filters which are used in the ACPS-FT algorithm are based on the mathematical functions so the identified characters are assigned with a numeric value for finding the filtered values.
TimeInterval: Time interval is user defined time in seconds. This is defined for streaming data, that means the streaming data will be generated after this time interval.
ACPS-FT Algorithm[15] : The ACPS-FT is the Approximate Circular pattern Signature algorithm using the filters that are applied on the pattern and then on the input string. Initially it converts the pattern or string characters into numeric values and then filters applied on these numeric values and the end results are stored for further matching process. If the values of the pattern and the input string are matched that means the pattern is found in the input string and the matched pattern is copied in the output file. If the values are not matched then the sliding window is moved towards right by one keeping the window size same. And thus the filters are applied and the matching is done till the window reaches the end of the input string.
Store the searched Patterns in database: In streaming data existing patterns are searched. These patterns are stored in the database for further matching. If the searched pattern matches any of the pattern which is already present in the database its count will be incremented

if it is not found in the database then it will be stored in database.
Filters : The six filters used in the above algorithm are as follows.

1. Filter 1: Filter 1 is based on sum function. Value of each character is added.

2. Filter 2: Filter 2 is based on the absolute distance between the consecutive characters of the string. The absolute distance can be defined as,

3. Filter 3: Filter 3 is based on the total distance between the consecutive characters of the string.

4. Filter 4: It uses the sum() function used by filter 1 in different form. It applies sum() function on individual character.

5. Filter 5: It uses modulo() function between two progressive characters of given string.

6. Filter 6: It employs the XOR() operation. It is bitwise exclusive-OR operation between two progressive characters of P string.

Approximate Patterns Found: After the matching of whole string is done with the pattern, the found patterns are returned.

IV. RESULT

The below table describes the comparison between the overlapping

and non-overlapping technique for finding the existing pattern from the data stream. The streaming data is generated for one hour and the number of patterns found in both the cases is mentioned below. k is the approximation value for the given size of pattern.

K	Pattern Size	Overlapping	Non-overlapping
2	2000	399	3635
2	5000	2809	3329
2	10,000	1616	1877
2	20,000	1005	1205
2	50,000	305	320
5	2000	153	1599
5	5000	1094	2932
5	10,000	1099	1854

V. CONCLUSION

In this the system finds approximate circular patterns. The research work uses the ACPS-FT algorithm in which 6 simple light weight filters are defined. These filters are used for the preprocessing of the pattern and input string to find the pattern from the input stream. The system also finds the existence of any pattern in the input stream. The existing system uses the overlapping technique for finding the given pattern in the given input string. In this the patterns are not stored, the values are calculated every time the system is executed. Proposed work uses the same algorithm but instead of using overlapping technique, non overlapping technique is used to find the patterns. When a new pattern is found it is checked with the existing patterns in the database if pattern is present in the database its count is incremented and if it is not present then the pattern is added in the database.

VI. REFERENCES

[1] M. Lothaire, Applied Combinatorics on Words. New York: Cambridge Univ. Press, 2005.

[2] A.Mosig, I. L. Hofacker, P. F. Stadler, and A. Zell, "Comparative analysis of cyclic sequences: Viroids and other small circular RNAs," in Proc. German Conf. Bioinformat. (GCB), 2006, vol. P-83, Lecture Notes in Informatics, pp. 93–102.

[3] M. O. Ku¨lekci. Tara: An algorithm for fast searching of multiple patterns on text files. In Computer and information sciences, 2007. iscis 2007. 22nd international symposium on, pages 1–6, Nov. 2007.

[4] C. S. Iliopoulos and M. S. Rahman, "Indexing circular patterns," in Proc. 2nd Int. Conf. Algorithms Comput., 2008, pp. 46–57.

[5] K. Fredriksson and S. Grabowski, "Average-optimal string matching," J. Discrete Algorithms, vol. 7, no. 4, pp. 579–594, 2009.

[6] F. Fernandes, L. Pereira, and A. Freitas, "CSA: An efficient algorithm to improve circular DNA multiple alignment," BMC Bioinformat., vol. 10, pp. 1–13, 2009.

[7] T. Lee, J. C. Na, H. Park, K. Park, and J. S. Sim, "Finding optimal alignment and consensus of circular strings," in Proc. 21st Annu. Conf. Combinatorial Pattern Match., 2010, pp. 310–322.

[8] Lee, T., Na, J., Park, H., Park, K., Sim, J.: Finding optimal alignment and consensus of circular strings. In: Proceedings of the 21st Annual Conference on Combinatorial Pattern Matching, pp. 310–322 (2010)

[9] J. Lin and D. Adjeroh, "All-against-all circular pattern matching," Comput. J., vol. 55, no. 7, pp. 897–906, 2012.

[10] K.-H. Chen, G.-S. Huang, and R. C.-T. Lee, "Bit-parallel algorithms for exact circular string matching," Comput. J., vol. 57, no. 5, pp. 731–743, 2014.

[11] M. Aashikur Rahman Azim, C. S. Iliopoulos, M. Sohel Rahman, and M. Samiruzzaman, "A fast and lightweight filter-based algorithm for circular pattern matching," in Proc. ACM Conf. Bioinformat., Comput. Biol., Health, Informat., 2014.

[12] M. Aashikur Rahman Azim, C. S. Iliopoulos, M. Sohel Rahman, and M. Samiruzzaman, "A filter-based approach for approximate circular pattern matching," in Proc. Bioinformat. Res. Appl. (ISBRA 2015), Norfolk, VA, USA, Jun. 7–10, 2015, pp. 24–35.

[13] A. Marzal, S. Barrachina, "Speeding Up the Computations of the Edit Distance for Cyclic Strings.", Pattern Recognition, Int'l Conference on, Los Alamitos, CA, USA, pp. 891–894. IEEE Computer Society, Washington, DC, USA.

[14] M. Aashikur Rahman Azim, C. S. Iliopoulos, M. Sohel Rahman, and M. Samiruzzaman, "Simplificpm: A simple and lightweight filter-based algorithm for circular pattern matching," Int. J. Genomics, vol. 2015, p. 10, 2015, Art. no. 259320.

[15] Md. Aashikur Rahman Azim, Costas S. Iliopoulos, M. Sohel Rahman, and M. Samiruzzaman, "A Simple, Fast, Filter-Based Algorithm for Approximate Circular Pattern Matching," Nanbioscience., vol. 15, No. 2, 2016.

Author's Profile:

Mr. Devyani Sharma, received the Master Of Engineering degree in Computer from the K.K.Wagh Institute of Engineering, Education and Research, she received the Bachelor Of Engineering degree from NDMVP's KBTCOE, Nasik, Maharashtra. She is currently working as Assistant Professor in IT Dept with Alamuri Ratnamala Institute of Engineering and Technology, Shahapur, Thane, Maharashtra, India.

Prof. Dr. S. M. Kamalapur, currently works at the Department of Computer Engineering, K. K. Wagh Institute of Engineering, Education and Research. She does research in Artificial Intelligence and Computer Science.