International Academy of Engineering and Medical Research, 2017
Volume-2, ISSUE-4
Published Online April 2017 in IAEMR (http://www.iaemr.com)

A Review on Mining Approximate Circular Pattern

Ms.Devyani D. Sharma
Department of Computer Engineering
K.K.Wagh Institute of Engineering, Education and Research, Nashik, Maharashtra, India.
Email-devyanisharma09@yahoo.in

Prof. Dr. S. M. Kamalapur
Department of Computer Engineering
K.K.Wagh Institute of Engineering, Education and Research, Nashik, Maharashtra, India.
Email-smkamalapur@kkwagh.edu.in

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 work 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 will return the pattern. Also the system finds the presence of any pattern in the input stream.**

Keywords—Circular DNA sequences, Circular Pattern Matching, Pattern Recognition.

I. INTRODUCTION

Circular pattern matching (CPM) has application in many areas like astronomy, computational biology, geometry[1]. 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[2]. 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 [14]. 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[13] 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[5] 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. 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[3] 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 thanthetheoreticaltime-optimalalgorithmCSAfortheECSM 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[5]. 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[6] 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 = S1,...,Shbe 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.,P1≤i≤h d(X,Si) 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., max1≤i≤hd(X,Si). Four different types of consensus problems[8], 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 [9]. 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 [10]. The problem of pair wise and multiple cyclic sequence alignments with affine gap costs, and an extension of a recent approach is explained in [11]. 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. The detailed description of each block is as follow:

Input Stream : Input stream consists of number strings having characters A, C, G and T which are used for the representation of DNA sequences of different organisms. Input stream is a continuous stream in which the given input pattern is to be searched.

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.

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).

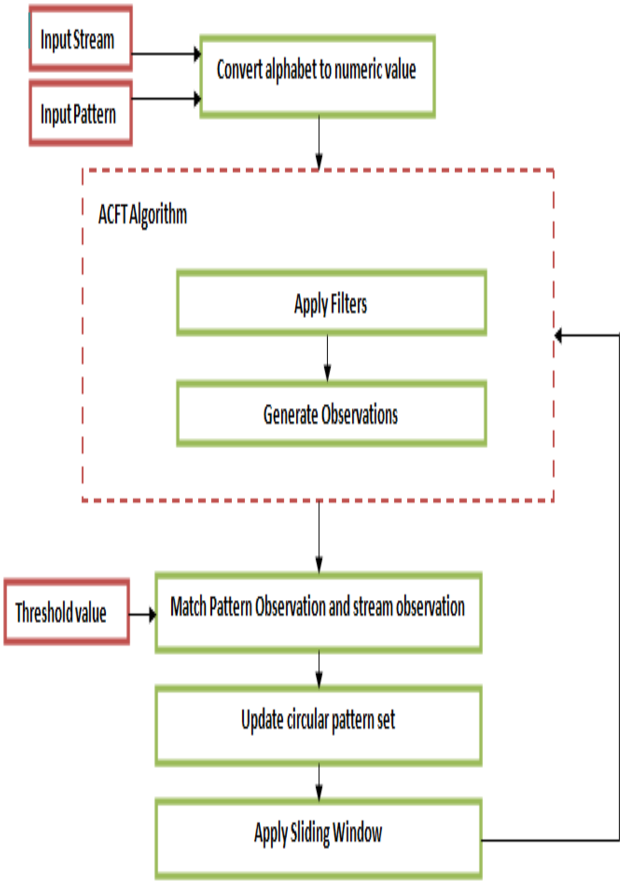
Input Stream
Input Pattern
Convert alphabet to numeric value
ACFT Algorithm
Apply Filters
Generate Observations
Threshold value
Match Pattern Observation and stream observation
Update circular pattern set
Apply Sliding Window

Fig. 1. System Architecture (Type 1)

ACPS-FT Algorithm[12][13] : 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.

Filters : The six filters used in the above algorithm are as follows.[14]

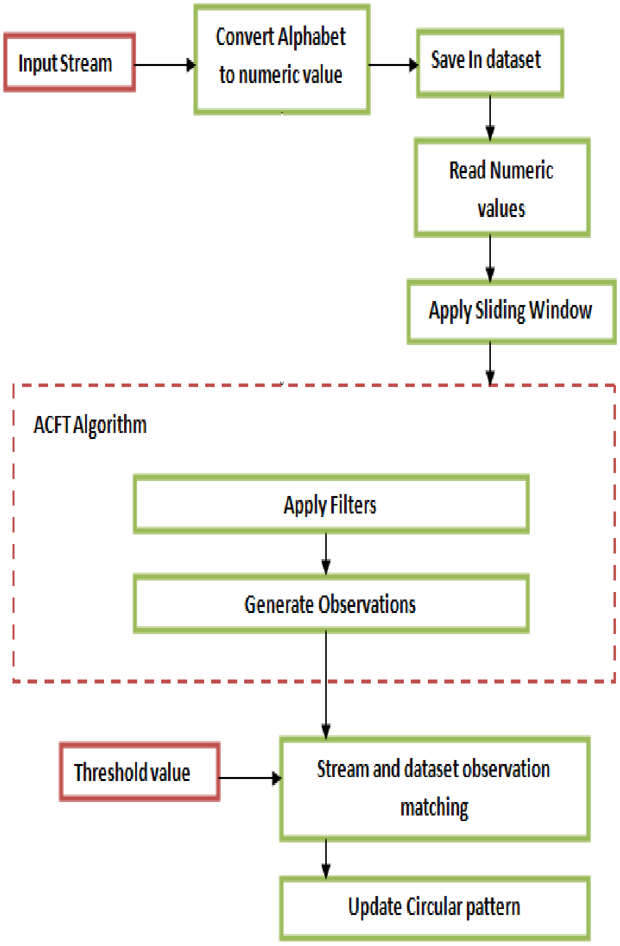

Fig. 2. System Architecture (Type 2)

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. The total distance can defined by following formula:

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.

Pattern Observation : In pattern observation phase the input stream is observed if a particular pattern exist in the stream.

Sliding window : In Sliding window a window of size equal to the size of the pattern is select for the input stream and the filters are applied on this selected stream. If the values of the pattern and the window stream are matched then the stream in the window is copied to the output file. If the values do not match then the window is moved one position to the right keeping the window size same and filters are applied on this window stream.

Check Observation : In this phase the filter values of pattern and the stream are checked if they have a matching value or not. If the values is matched then the occurred pattern is written in the output file. If the values do not match then the window is moved.

Generate Circular Pattern Set : In this the output of the system is stored. It consists of the matched patterns found in the input stream.

IV . CONCLUSION

Thus in this paper we have seen the different algorithms uptil now which are used to search the circular patterns from the data. These algorithms are used to find the variants of circular patterns that are exact circular pattern and approximate circular pattern. Also a new system is proposed in which the approximate patterns are searched from the input data stream.

V. REFERENCE

[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] 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.

[13] 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.

[14] 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.

ACKNOWLEDGMENT

The authors would like to thank Prof. Dr. S. S. Sane, Head of the Computer Department and Prof. Dr. K. N. Nandurkar, Principal of K.K.W.I.E.E.R.,Nashik. for their kindsupport and suggestions. We would also like to thank all the staff members of Computer Engineering Department and our collegues who knowingly or unknowingly helped us to complete this work.

Authors' Profiles

Ms. Devyani D. Sharma received the B. E. degree in Informaion Technology Engineering from Savitribai Phule Pune University, Maharashtra, India. Currently pursuing M. E. degree in Computer Engineering from Savitribai Phule Pune University. Her current research interest includes pattern matching, data mining and information retrieval.

Prof. Dr. S. M. Kamalapur Associate Professor in Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik.She has completed her Ph.D and M.E from Savitribai Phule Pune University, Maharashtra, India. Her research interest includes image processing, pattern mining, graph analysis.