Educational Administration: Theory and Practice
2024, 30(4), 741-748
ISSN: 2148-2403
https://kuey.net/

Research Article

Pattern Matching Of DNA Sequence Using FRCT Algorithm

Prashanth M.C[1*] and Prabhakar C J[2]

[1*],2 Department of P.G. Studies and Research in Computer Science
Kuvempu University, Shangaraghatta-577451 Karnataka, India

Citation: Prashanth M.C et.al (2024). Pattern Matching Of DNA Sequence Using FRCT Algorithm ...*Educational Administration: Theory and Practice*, *30*(4), 741-748,
Doi:10.53555/kuey.v30i4.1551

ARTICLE INFO

ABSTRACT

DNA sequences are differing with respect to their size in billions of nucleotides range.Pattern matching is significant for information processing in computer field in identifying the functional and structural behaviour of genes. In this proposed study, the pattern matching of sequences have been performed for the large set of DNA sequences. The pattern matching is effectively performed by using the Fast Reliable Cartesian Tree Algorithm (FRCT), where patterns are encoded and shifting patterns are enhanced which reduces the computational timeand assures high reliability. The proposed algorithm shows better pattern matching in terms ofexecution time compared with various existing strategies performed on suitable large DNA sequences dataset.

Keywords: DNA sequence, Pattern Matching, Cartesian Tree

I. Introduction

The field of bioinformatics has undergone significant advancements in recent years, particularly in the area of DNA sequence pattern matching. This innovative approach allows researchers to analyse vast amounts of genetic data to identify patterns and relationships between genes. The ability to accurately match DNA sequences has important implications forunderstanding genetic diseases, evolutionary relationships, and even personalized medicine. Despite the progress made in this field, challenges remain in developing robust algorithms thatcan handle the complexities of DNA sequences. DNA sequence patterns matching one or more sequences based on the provided search pattern and exhibits the positions and number of matching patterns. Pattern matching is essential for information processing in computer field in identifying the functional and structural behaviour of genes. The microbiologists frequentlysearched the significant information in databases. To retrieve the pattern, DNA database is highly complex and huge and it considered as difficult process. In similar with the large DNAsequences the DNA sequence size is grown which is critical in recognize the genomes biological features. Pattern matching process is the essential one and the precision values are based on the repository, search query and integrated system [1].The pattern matching is appropriate in fields such as phylogenetic and evolutionary biology. In these applications, specific DNA subsequence's are extracted from the genomic data of organisms' different species for the purposes of understanding their relatedness, descent, and origin. Therefore, in this context, a pattern matching algorithm must be able to search in databases spanning gigabytes to terabytes or more and in whole genomes with 3 billion base pairs [2]. On the otherhand, the DNA sequences are very long. Therefore, the time consumed for matching with the pattern is considered as the most critical metric.

DNA sequence pattern matching finds applications in various fields such as bioinformatics, genomics, medicine, forensic science, and evolutionary biology. In bioinformatics, it is used for gene prediction, genome assembly, and sequence alignment. Genomics relies on DNA sequence pattern matching to identify genetic variations and understand the genetic basis of diseases. In medicine, this technique aids in personalized medicine by analyzing individual DNA sequences for tailored treatments. Forensic scientists utilize DNA sequence pattern matching for identifying individuals or tracing ancestry. Evolutionary biologists use this method to study the evolutionary relationships between different species. These applications demonstrate the versatility and importance of DNA sequence pattern matching across scientific disciplines, highlighting its crucial role in advancing research and understanding genetic information in diverse contexts.

Current challenges in DNA sequence pattern matching research include the ever-increasing volume of data generated by advanced sequencing technologies, which poses computational and storage challenges for existing algorithms and databases. Furthermore, the complexity of genetic variations, such as single nucleotide polymorphisms (SNPs) and structural variations, necessitates the development of more sensitive and accurate pattern matching algorithms to account for these variations. Future directions in this field could involve the

integration of machine learning techniques, such as deep learning, to improve the accuracy and efficiency of pattern matching algorithms. Additionally, there is a growing need for the development of standardized benchmark datasets and evaluation metrics to facilitate the comparison of different algorithms and ensure reproducibility in research findings. These challenges and future directions are crucial for advancing DNA sequence pattern matching research and its applications in genomics and personalized medicine [3].

The historical overview of DNA sequence pattern matching reveals a significant evolution in the field of bioinformatics. Initially, the simplest form of sequence comparison involved manual alignment, which was time-consuming and error-prone. The introduction of computeralgorithms revolutionized this process, with early methods such as Smith-Waterman and Needleman-Wunsch paving the way for modern sequence similarity algorithms [4]. These algorithms not only improved the speed and accuracy of sequence alignment but also enabledthe detection of subtle sequence patterns that were previously inaccessible. As technology advanced, heuristic algorithms like BLAST and FASTA were developed to handle the increasing size of biological databases efficiently. The development of more sophisticated tools, such as hidden Markov models and machine learning approaches, further enhanced the capabilities of DNA sequence pattern matching, making it a cornerstone in genomics and molecular biology research.

It is highly difficult for the users to extract the required information from DNA sequences ifthe data size is more. Robust and effective techniques are required for the fast and reliable pattern matching techniques. Pattern matching is of two kinds based on the scenario for whichit is applied: exact pattern matching (EPM) and approximate pattern matching (APM). An EPMis highly needed for the scenario in which the accuracy expected is 100%. For instance, whenthere is a search of a record in a database using a key value, exact matching is mandatory. Equally, APM finds its application in the fields like Bioinformatics, web search engines, text mining, intrusion detection system [5], and spam filtering. The suitable pattern matching algorithm identifies whether the probability is resulted to effective or ineffective search. The problem is mentioned as, from the given pattern p with length m, text/string as T, with length n as $m \leq n$. All the occurrences of p is identified in T. It is expected to be suitable or exact matching is required. The major contribution of the study is to perform the reliable patternmatching of suitable DNA sequences using Fast Reliable Cartesian Tree- FRCT algorithm. The section II describes the literature review of the pattern matching algorithms for DNA sequences. Further the proposed FRCT algorithm is illustrated in the section III. The dataset description, results and discussion are depicted in the section IV. Finally, the proposed study is concluded in section V.

II. Related works

Methods and algorithms used in DNA sequence pattern matching play a crucial role in bioinformatics research. Various techniques have been developed to effectively identifysimilarities or differences in DNA sequences, allowing researchers to unravel genetic information and understand evolutionary relationships. The Smith-Waterman algorithm [6], which scores matches, mismatches, and gaps to determine the best local alignment between two sequences, is one often used technique. The BLAST method, which quickly scans huge sequence databases for comparable regions based on statistical significance, is another extensively used strategy. These techniques have greatly expanded genomic investigations andprovide flexible tools for sequence analysis, along with others such as the Needleman-Wunschalgorithm and Hidden Markov Models. Knowing the benefits and drawbacks of each approachis crucial to choose the best one for a given collection of datasets and research objectives.

For the purpose of predicting SF (Sequence Function), DNA patterns give SDA (Sequence Data Analysis) a useful context. It also looks at how different DNA sequences have evolved over time. Consequently, K-mer frequency was employed in [7] to categorise DNA sequences.Because, it can transform VLS (variable length sequence) into fixed length numerical feature vectors. Six datasets were used to assess the performance of the proposed solution. As a result,when compared to alternative methodologies, the analytical results confirmed the usefulness of the proposed methodology. A sequence-based computational model (SCM) for forecasting the synthesis of DNA G4 (G-quadruplex) was examined in this article [8]. The K-Mer functionfor the Human Genome (HG) via G4-seq technology is the basis for matching patterns of DNAsequences. To further enhance the prediction performance, pseudo-K-tuple GC configuration (PsekGCC) was introduced in this work [9].

Conversely, the study [10] proposed the Efficient Pattern Matching Algorithm (EPMA) as apattern matching method for DNA sequences. They evaluated the proposed method with actualcomparison data. It was found that matching DNA sequence patterns was a good use for the proposed approach. Berlin K et al. have proposed the basic mutation pattern of DNA sequencesto be matched using a distance-based method [11]. In order to do this, specialists adjusted the nucleotide pattern arrangement in the DNA hamming process to determine the basic HepatitisC Virus (HCV) pattern. One hundred sample data points were identified by experimentation that was positively impacted across age ranges. After taking into account the normalisation value, the experiment yields a matching score with a hamming value. Using an index-based shifting strategy that includes a pre-processing and searching phase, Tania Islam et al. proposedstring matching for DNA sequences [12]. Since the initial character of the pattern and the substring are the same, the pattern's second character matches the first substring's second character from left to right. This character comparison is done throughout both the pre-processing and searching phases. As the number of repeats increases, the algorithms for the protein sequence

operate more quickly than the English text. The Hash function, which searches faster on any length of codons and finds the gene sequence even when increasing thelength of its string, was proposed by Paramita Basak Upama et al. to detect the codon from large RNA sequences met and stop techniques [13].

Aashikur Rahman Azim et al. proposed approximate circular pattern matching is to locate allthe occurrences of rotations in a pattern of length in the length test using a simple, fast, filter- based algorithm [14]. Due to the significant reduction in search space caused by filtering, the proposed method operates more quickly. The proposed approach performs twice as quickly as the state of the art when compared to the simple ACSSMF technique. A Nettree technique is proposed by Youxi Wu et al. [15] for approximating the pattern maximal matching with limitations one-off and gaps. This is done using an offline occurrence algorithm in conjunction with Nettree and a heuristic search. Real-world biological data is used for experimentation, andthe proposed method's performance is compared to that of the SAIL algorithm. A method of summing and normalising scores is proposed in [16] to assess the related TF and TFBS patternsof DNA and create an efficient pipeline linking the related unified score patterns. The approachto identify the rules and binding cores with the best correlation sum scores is provided by the accurate rankings. The matching ratio for protein data bank structures verifications acts as thehighest corresponding percentage.

There are two single pattern matching algorithms (referred to as ILDM1 and ILDM2) proposed [17], each of which is made up of forward finite automation and smallest suffix automation. Chauhan Liu et al. proposed single pattern matching methods typically use a window with a size of m to scan the text. When the window is moved from the previous windowto the current window in the LDM method, valuable information generated by the forward automation is destroyed. To fully or conditionally utilise the valuable data generated by two distinct pattern matching algorithms. It is evident from the experiments that the average time complexities of the two algorithms are lower for short patterns than RF and LDM and for longpatterns than BM. An accurate approach for selecting probes from big genomes was proposedby Wing-Kin Sung et al. [18]. The homogeneity and specification criteria are used to choose high-quality probes from a set of experimental data derived from a small number of genomes that have been extensively tested by previous probe design techniques. Large genomes can have optimal short (20 bases) or long (50–70 bases) probes generated using the proposed algorithm.

In order to detect important matches of position weight matrices in linear time an online approach of this kind is proposed [19]. The proposed algorithm is based on super alphabet techniques and traditional multi-pattern matching filtrations created for precise and approximative key word matching. To conduct the experiments, six proposed algorithms— ACE, LF, ACLF, NS, MLF, and MALF—as well as other well-known base time algorithms—PLS and Naïve—are built. The ACE method, which is competitive for short matrices and whose search speed is independent of matrix length, is potentially the best of all the aforementioned techniques. For biological sequences, a Fast-Searching algorithm is proposedby Simone Faro et al. [20] that make use of several hash functions. This technique enhances the speed of existing string-matching algorithms for searching DNA sequences. The search phase of the algorithm is based on a standard sliding window mechanism, where it reads the rightmost substrings of length q of the current window of the text to calculate the optimal shiftadvancement. The pre-processing phase of the algorithm consists in computing different shift values for all possible strings of length. A solid foundation for enormous multiple lengthy pattern search is provided by the suggested algorithm.The composite BM (BoyerMoore) algorithm was proposed by Zhengda Xiong et al. which matches strings efficiently while accelerating pattern speed, was proposed for string matching [21]. It makes advantage of past matching data. Binary matching is a test conducted to assess the performance of the Boyer Moore (BM) and Composite Boyer Moore(CBM) algorithms. The proposed CBM algorithm outperforms the BM algorithm by 84%.

A comparison algorithm that generates the number of matches and mismatches using fuzzymembership values is based on the logical match technique proposed by Sanil Shanker KP et al. [22]. The CPP language is used to perform a logical match for DNA sequences on the Linuxplatform. With an O(m+n) time complexity, the proposed approach creates membership values to identify sequence similarities. The distinct approach based on both artificial and real datum(bits of information), was examined for DNA sequences of the NCBI databank. The computingtime is dependent on the length of the sequences. The Yusei Tsuboi et al. proposed pattern matching approach is used by DNA computing to address engineering challenges [23]. The parallel operation approach is designed to identify DNA molecules. Using the categorised features of the molecule, the model locates a molecule as the only existing search picture. PCRand Gel electrophoresis are then used to apply the features to construct a network. Because sorting and calculation are done simultaneously in every test, the image or pattern search timeis persistent. Peyman Neamatollahi et al. presented three pattern matching algorithms such as FLPM, PAPM, and LFPM that are specially formulated to speed up searches on large DNA sequences [24]. LFPM is a character-based pattern matching algorithm, while PAPM and LFPM perform based on the word processing approach. The proposed algorithms raiseperformance by utilizing word processing and also by searching the least frequent word of thepattern in the sequence.

III. Proposed methodology

In this proposed study, the pattern matching of sequences is performed for the large set of DNAsequences. The DNA sequences is an input sequences, in which the patterns have to be checkedwhether it is matched or not based on the given query data. Before matching those patterns initially, DNA sequences have been pre-processed in which the letters are changed to capital letters and numeral values are removed. Further, from the pattern, the index position is considered; the number of occurrences of the patterns is identified. If the pattern is not found in the input DNA sequences, it will show as the pattern does not find. The pattern matching is effectively performed by using the Fast Reliable Cartesian Tree Algorithm (FRCT)[25].Cartesian tree matching is identifying all substrings from the provided DNA sequences with similar Cartesian trees as that of provided pattern. Cartesian tree matching is similar to order preserving matching among the multiple generalized matchings, which can deal with relative orders. Compared with order preserving matching, Cartesian tree matching is more suitable.

3.1 Fast Reliable Cartesian Tree Algorithm (FRCT)

The DNA sequences pattern matching is effectively performed by using the proposed Fast Reliable Cartesian Tree Algorithm (FRCT)[25]. The Cartesian tree is a tree data structure thatrepresents an array, only focusing on the results of comparisons between numeric values in thearray. Cartesian tree matching means that two strings match if they have the same Cartesian trees. Cartesian tree matching is the problem to find all the matches in the text which have thesame Cartesian tree as a given pattern. Here, the patterns are encoded and shifting patterns areenhanced using the offset value and it further reduces the computational time and assures high reliability. The offset value is initialized for the checking of the position in performing the pattern matching; the shift value is updated with the offset value. In order to solve Cartesian tree matching without building every possible Cartesian tree, an efficient representation to storethe information about Cartesian trees, called as distance-relation representation. For the reliablein sequence, the distance relation is computed which is presented in below algorithm. The distance-relation representation has a one-to-one mapping to the Cartesian tree, so it can substitute the Cartesian tree without any loss of information.

Algorithm 1 Computing Distance Relation for Reliable of a Sequence

1: $procedure\ DISTANCERELATION(R[1..n])$
2: $NS \leftarrow nullset$
3: $for\ i \leftarrow$ **1 *to n do***
4: ***while*** $NS\ is\ not\ empty$ ***do***
5: $(value, index) \leftarrow NS.top$
6: $if\ value \leq R[i]$ ***then***
7: ***break***
8: $NS.pop$
9: ***if*** $NS\ is\ empty$ ***then***
10: $DR(S)[i] \leftarrow 0$
11: ***else***
12: $DR(R)[i] \leftarrow i - index$
13: $NS.push((R[i], i))$
14: ***return*** $DR(S)$

Based on the given string R[1. . n], the distance relation is computed for sequence reliability in linear time. For i < j, R[i] and R[j] satisfy the R[i]> R[j] which cannot be the distance relationto R[k] for k > j. While scanning from left to right, R[j] is not presented so, R[i] is stored. Non-decreasing subsequence is formed if R[i] is only stored. While considering the new value,if the values are larger than the values are popped. Distance relation- DR(S) is computed finallyfor any R[1. . n].

The failure function of string P is considered as an integer string shown in Eq. (1),

$$\pi[q]_1 = \begin{cases} max\{k: 1CT(P[1..k]) = CT(P[q-k+1..q]) for 1 \leq k < q\} & if q > 1 \\ 0 & if q = 1 \end{cases} \quad (1)$$

(1)

The largest k is the $[q]$ in which $P[1..q]$ suffix and prefix with k length showing similar Cartesian trees. For obtaining linear time text search, the failure function is utilized. For findingthe mismatch among the text and

pattern, the failure function is utilized every time while scanning the text from left to right. For fast reliable Cartesian tree, this kind of text search ideais applied shown in algorithm below.

Algorithm 2: Text search of Cartesian tree matching

1: **Procedure** $CartesianTreeMatch(T[1..n], P[1..m])$
2: $DR(P) \leftarrow DISTANCE\_RELATION(P)$
3: $\pi \leftarrow FAILURE\_FUNC(P)$
4: $len \leftarrow 0$
5: $VQ \leftarrow vacant\ deque$ # EncodethePattern
6: **for** $h \leftarrow 1$ to n **do**
7: $Pop\ Components(esteem, list) from\ back\ of\ VQ\ to\ such\ that\ esteem > T[i]$
8: *whilele n not equal to 0 do*
9: **if** $DR(T[i - len \ldots i])[len + 1] = DR(P)[len + 1]$, **then**
10: **break**
11: **else**
12: $len \leftarrow \pi[len]$
13: $Delete\ components\ (esteem, list) from\ front\ of\ VQ\ such\ that\ list < i - len$
14: $len \leftarrow len + 1$
15: $VQ.push\_back((T[i], i))$ #Update the Offset
16: **if** $len = m$, **then**
17: print "Match happened at $i - z + 1$"
18: $len \leftarrow \pi[len]$
19: $Delete\ Components(esteem, list) from\ front\ of\ VQ\ such\ that\ list \leq i - len$

Using the $O(m)$ space, text search is performed; distance relation representation of the given DNA sequences is computed online while the text is read. Certainly, the distance relation representation is not stored which costs $O(n)$ space. Moreover, for delete elements in front, non-decreasing subsequence is formed and it is stored among the text pattern matching of textcharacters. Based on this, the DR is always equal or smaller to m. The text search is performedusing $O(m)$. The computation time is constant.

IV. Experimental Results

In this section, the experimental results of the proposed FRCT algorithm for DNA sequence pattern matching are presented. The evaluation of the proposed matching algorithm is performed with respect to computational time based on pattern matching. We use four real DNA sequences in experiments obtained from NCBI database ranging from 585.2 kb to 222.6Mb in size with different base pairs (bp) in length, as summarized in Table 1. The datasets usedare mentioned in Table 1 (Complete summarization of benchmark data that is downloaded froma site named NCBI site [26]). All datasets are in FASTA format, each sequence file contains information of genome nature i.e. accession number, reference number to NCBI database and name of species in a single line starts with the '>' symbol. For each dataset, we ignore that lineand the patterns were randomly extracted from the source data. The results of pattern matchingof the proposed algorithm are presented in Table 2. For number of occurrences and comparisons of various patterns P's, the proposed algorithm results are shown in Table 2. Forthe evaluation results the patterns are randomly selected from DNA sequence data. This algorithm shows minimal complexity and less computation time. The Table 2 shows five different fields such as pattern P's, number of occurrences, number of characters, and comparison time.

Table1. Summary of Benchmark DNA sequences datasets

Datasets	Length of thesequence	Description
HUMHPRTB	56737	Human hypoxanthine phosphoribosyltransferase gene
VACCG	194711	Vaccinia virus complete genome
HUMGHCSA	66495	GH-2 and GH-1: Human growth hormones and CS-5, CS-2 and CS-1: chorionic somatomammotropin genes
MPOMTCG	186609	Marchantiapolymorpha mitochondrion complete genome
HUMDYSTROP	38770	Homo sapiens dystrophin gene intron44
HUMHBB	73308	Human beta globin area on chromosome-11.

Table 2: The results of pattern matching of the proposed algorithm for variable pattern length

Pattern (P's)	No of characters	No. of occurrence	Time Taken(ms)
AG	2	53	0.005
CAT	3	11	0.009
AACG	4	5	0.016
AAGAA	5	2	0.032
AAAAAA	6	3	0.056
AGAACGC	7	2	0.052
AAAAAAGG	8	1	0.045
GCTCATTAG	9	1	0.063
CCTTTTCCGG	10	1	0.067
TTTTGCCGTGT	11	1	0.069
TTCTTAATAAAA	12	1	0.073
GGGACCAAAAAAT	13	1	0.08
TTTTGCCGTGTTGA	14	1	0.085
CCTCCAAAAAAGGCT	15	1	0.089
GGCTGTTCAACGCTCC	16	1	0.088
TTTTCGATTGCTCATTA	17	1	0.092
GGGATTTGGCTATACTCC	18	1	0.093
GGCCTTGTCTAAAGGTATG	19	1	0.096
CCTGAGCGCGTCCTCCGTAC	20	1	0.103

4.2 Comparative analysis

The proposed method is compared with the existing methods such as Least Frequency Pattern Matching (LFPM)[24], Processor-Aware Pattern Matching(PAPM)[24], First-Last Pattern Matching(FLPM)[24], Boyer Moore (BM)[21], Brute Force(BF), Divide and Conquer Pattern Matching(DCPM)[27] using the datasets mentioned in Table 1. From Fig.1 and Table 3, it is observed that comparison with the existing methods, the proposed method shows reduced execution time in pattern matching for the selected DNA sequences and thus effective againstvarious existing strategies. The proposed FRCT algorithm is thus highly effective in pattern matching and shows high reliability and reduced computational time.

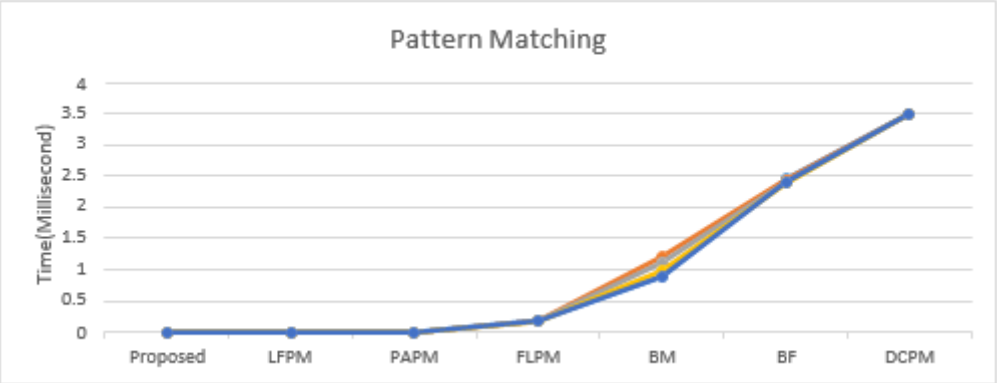

Fig.1. The pattern matching time comparison with the existing techniques

Table 3: The comparative study of the proposed method with existing methods based onmatching time

	Comparison of proposed and existing methods based on Pattern matching Time (ms)						
Pattern Length	**Proposed**	**LFPM**	**PAPM**	**FLPM**	**BM**	**BF**	**DCPM**
5	0.0036	0.0046	0.0054	0.1915	0.8913	2.4503	3.5
10	0.0016	0.0025	0.0018	0.1851	1.21216	2.4444	3.5
15	0.0018	0.0025	0.0022	0.1800	1.1167	2.4044	3.5
20	0.0022	0.0026	0.0025	0.1727	0.9852	2.3856	3.5
25	0.0016	0.0018	0.0025	0.1852	0.8966	2.4046	3.5

FRCT is an effective pattern matching algorithms that reduce time when compared to other techniques. This demonstrates that the pattern matching time optimization strategy we utilisedwas successful. FRCT is the most effective and efficient method in terms of minimising the time required for various pattern sizes, according to the table, and it outperforms other algorithms. Furthermore, FRCT yielded better outcomes than PAPM, which includes a pre- processing and matching phase. This shows that they are both superior in all pattern scaling matches. This is especially advantageous for the expansion of biological data, which isconstantly increasing in volume. The results of the experiments show that the FRCT surpass the other algorithms in terms of time cost.

V. Conclusion

In this paper, we proposed DNA sequence pattern matching method using FRCA algorithm. An effective methodology is introduced in this study for reliable DNA Sequence pattern matching from the suitable dataset using the Fast reliable Cartesian tree- FRCT algorithm. Furthermore, FRCT searches for the lowest frequency word of the pattern in order to minimizethe algorithm run time. The experimental results reveal that the proposed algorithm surpasses the other existing algorithms in terms of time cost. The results of the proposed study show thatit assures high reliability and minimal execution time compared with the existing approaches.

References

1. Cherry, K. M., & Qian, L. (2018). Scaling up molecular pattern recognition with DNA-basedwinner-take-all neural networks. Nature, 559(7714), 370-376.
2. Varsani, A., &Krupovic, M. (2017). Sequence-based taxonomic framework for the classification of uncultured single-stranded DNA viruses of the family Genomoviridae. Virusevolution, 3(1), vew037.
3. Pellegrini, M., Magi, A., & Iliopoulos, C. S. (2016). repetitive Structures in Biological Sequences: algorithms and applications. Frontiers in Bioengineering and Biotechnology, 4, 66.
4. Dawson, V. L., Dawson, T. M., & Kang, S. U. (2023). DNA Methylation signature of aging: Potential impact on the pathogenesis of parkinson's disease. Journal of Parkinson's Disease, 13(2), 145-164.
5. Busia, A., Dahl, G. E., Fannjiang, C., Alexander, D. H., Dorfman, E., Poplin, R., ... &DePristo,
M. (2018). A deep learning approach to pattern recognition for short DNA sequences. BioRxiv, 353474.
6. Mahmud, P., Rahman, A., & Hasan Talukder, K. (2023). An improved hashing approach for biological sequence to solve exact pattern matching problems. Applied Computational Intelligence and Soft Computing, 2023.
7. Phan, D., Ngoc, G. N., Lumbanraja, F. R., Faisal, M. R., Abipihi, B., Purnama, B., ... & Satou,
K. (2017). Combined use of k-mer numerical features and position-specific categoricalfeatures in fixed-length DNA sequence classification. Journal of Biomedical Science and Engineering, 10(8), 390-401.
8. Barros De Paula, R. (2021). Unveiling Global Roles Of G-Quadruplexes And G4-22 In HumanGenetics.
9. Liu, B., Chen, S., Yan, K., & Weng, F. (2019). iRO-PsekGCC: identify DNA replication origins based on pseudo k-tuple GC composition. Frontiers in Genetics, 10, 476969.
10. Tahir, M., Sardaraz, M., & Ikram, A. A. (2017). EPMA: efficient pattern matching algorithmfor DNA sequences. Expert Systems with Applications, 80, 162-170.
11. Al Kindhi, B., Hendrawan, M. A., Purwitasari, D., Sardjono, T. A., & Purnomo, M. H. (2017,November). Distance-based pattern matching of DNA sequences for evaluating primary mutation. In 2017 2nd International Conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE) (pp. 310-314). IEEE.
12. Islam, T., & Talukder, K. H. (2017, December). An improved algorithm for string matching using index based shifting approach. In 2017 20th International Conference of Computer andInformation Technology (ICCIT) (pp. 1-4). IEEE.
13. Upama, P. B., Khan, J. T., Zemim, F., Yasmin, Z., & Sakib, N. (2015, December). A new approach in pattern matching: codon detection in DNA and RNA using hash function(CDDRHF). In 2015 18th International Conference On Computer and Information Technology (ICCIT) (pp. 172-177). IEEE.
14. Azim, M. A. R., Iliopoulos, C. S., Rahman, M. S., &Samiruzzaman, M. (2016). A simple, fast,filter-based algorithm for approximate circular pattern matching. IEEE Transactions on NanoBioscience, 15(2), 93-100.
15. Wu, Y., Wu, X., Jiang, H., & Min, F. (2010, October). A Nettree for approximate maximal pattern matching with gaps and one-off constraint. In 2010 22nd IEEE International Conference on Tools with Artificial Intelligence (Vol. 2, pp. 38-41). IEEE.
16. Chan, T. M., Lo, L. Y., Sze-To, H. Y., Leung, K. S., Xiao, X., & Wong, M. H. (2013). Modeling associated protein-DNA pattern discovery with unified scores. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 10(3), 696-707.
17. Liu, C., Wang, Y., Liu, D., & Li, D. (2006, November). Two improved single pattern matchingalgorithms. In

16th International Conference on Artificial Reality and Telexistence-- Workshops (ICAT'06) (pp. 419-422). IEEE.

18. Sung, W. K., & Lee, W. H. (2003, August). Fast and accurate probe selection algorithm for large genomes. In Computational Systems Bioinformatics. CSB2003. Proceedings of the 2003 IEEE Bioinformatics Conference. CSB2003 (pp. 65-74). IEEE.
19. Pizzi, C., Rastas, P., & Ukkonen, E. (2009). Finding significant matches of position weight matrices in linear time. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 8(1), 69-79.
20. Faro, S., &Lecroq, T. (2012, November). Fast searching in biological sequences using multiple hash functions. In 2012 IEEE 12th International Conference on Bioinformatics & Bioengineering (BIBE) (pp. 175-180). IEEE.
21. Xiong, Z. (2010, December). A composite boyer-moore algorithm for the string-matching problem. In 2010 International Conference on Parallel and Distributed Computing, Applications and Technologies (pp. 492-496). IEEE.
22. Shanker, K. S., Austin, J., & Sherly, E. (2010, February). An algorithm for alignment-free sequence comparison using logical match. In 2010 The 2nd international conference on computer and automation engineering (ICCAE) (Vol. 3, pp. 536-538). IEEE.
23. Tsuboi, Y., & Ono, O. (2003, July). Pattern matching algorithm for engineering problems by using DNA computing. In Proceedings 2003 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2003) (Vol. 2, pp. 1005-1008). IEEE.
24. Neamatollahi, P., Hadi, M., & Naghibzadeh, M. (2020). Simple and efficient pattern matchingalgorithms for biological sequences. IEEE Access, 8, 23838-23846.https://www.ncbi.nlm.nih.gov/nuccore/AL158070.11.
25. Park, S. G., Amir, A., Landau, G. M., & Park, K. (2019). Cartesian tree matching andindexing. arXiv preprint arXiv:1905.08974.https://arxiv.org/abs/1905.08974
26. Wu, Y., Zhu, C., Li, Y., Guo, L., & Wu, X. (2020). NetNCSP: Nonoverlapping closedsequential pattern mining. Knowledge-based systems, 196, 105812.
27. Raju, S. V., Reddy, K. K. V. V. S., & Rao, C. S. (2018). Parallel string matching with lineararray, butterfly and divide and conquer models. Annals of Data Science, 5, 181-207.