2024 7th International Conference on Informatics and Computational Sciences (ICICoS) | 979-8-3503-7588-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICICoS62600.2024.10636821

DNA Pattern Matching Algorithms within *Sorghum bicolor* Genome: A Comparative Study

Dwika Lovitasari Yonia
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
6025231051@student.its.ac.id

Shintami Chusnul Hidayati
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
shintami@its.ac.id

Riyanarto Sarno
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@its.ac.id

Abdullah Faqih Septiyanto
Department of Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
7025221022@student.its.ac.id

***Abstract*—*Sorghum bicolor*, a vital cereal crop with significant roles in agriculture and biofuel production, presents a complex genome that poses substantial challenges for genetic analysis and improvement. This study provides a comprehensive comparison of DNA pattern-matching algorithms applied to the *Sorghum bicolor* genome, focusing on the accuracy, efficiency, and effectiveness of each technique. The algorithms evaluated include the simple Brute Force method, the efficient Boyer-Moore algorithm, the linear-time Knuth-Morris-Pratt (KMP) algorithm, the Enhanced First-Last Pattern Matching (EFLPM), and the Enhanced Processor-Aware Pattern Matching (EPAPM). Notably, the EFLPM and EPAPM algorithms excel at accommodating errors and mutations in DNA sequences, with EPAPM additionally leveraging parallel processing techniques to enhance performance. This comparative study highlights the crucial role of temporal complexity in selecting the most suitable DNA pattern-matching algorithm for genomic analysis.**

***Index Terms*—agricultural technology, bioinformatics, pattern matching, *Sorghum bicolor* genome**

I. Introduction

In bioinformatics, deciphering the complexities of deoxyribonucleic acid (DNA) is crucial for understanding biological phenomena at the molecular level. DNA, the foundation of life, encodes the genetic instructions vital for the development and functioning of all living organisms. These instructions are stored within the sequences of nitrogenous bases, constituting the genetic blueprint for proteins and RNA sequences [1]. The structural integrity of DNA, characterized by its iconic double helix, provides it with the resilience to withstand mechanical forces. However, it remains prone to certain deformations, such as twisting and bending [2]. The intricate sequence of nucleotide bases in DNA, consisting of adenine (A), guanine (G), cytosine (C), and thymine (T), encases extensive biophysical information. A thorough understanding of DNA's structure is critical for elucidating gene expression, DNA replication, and chromosome segregation during cell division [3].

DNA pattern matching, a challenging yet essential domain within bioinformatics, enables identifying and analyzing biological sequences, including genes or entire chromosomes. This process is crucial for locating specific sequences within databases, aiding in the search for particular genes or DNA markers [4], [5]. Moreover, DNA pattern matching is essential for discovering genetic mutations, diagnosing genetic disorders, and advancing drug development. It also plays a pivotal role in uncovering repeating patterns within DNA sequences, which are fundamental to understanding biological functions and evolutionary relationships among species [6], [7].

In computer science and qualitative analysis, pattern matching is crucial for identifying specific patterns within a dataset [8]–[10]. Applied to DNA, it aims to reveal sequences of biological significance by comparing DNA samples from various species across evolutionary distances [11], [12]. Using bioinformatics principles, pattern-matching algorithms analyze DNA sequences to detect genetic diseases, decipher genetic variations, and trace evolutionary lineages, thereby significantly advancing genetics, medicine, and evolutionary biology [13], [14].

This study seeks to deepen our understanding of DNA's structure and function within bioinformatics, focusing on the critical role of DNA pattern matching. By examining various pattern-matching algorithms, including their time complexities and comparative execution times, this research assesses their effectiveness, efficiency, and accuracy in DNA analysis, contributing to the advancement of genetic research.

Recent advancements in DNA pattern matching have led to significant improvements in algorithm efficiency and error tolerance. Notably, the Enhanced First-Last Pattern Matching (EFLPM) and the Enhanced Processor-Aware Pattern Matching (EPAPM) algorithms have demonstrated their ability to reduce the computational complexity required to process large DNA sequences effectively. These algorithms represent a significant evolution from traditional methods by integrating novel heuristics that accommodate the variability within DNA, making them particularly suitable for complex genomes such

as *Sorghum bicolor*. Exploring their performance further could provide valuable insights into their practical applications.

Specifically, this research investigates DNA pattern matching within the *Sorghum bicolor* genome, employing widely-used bioinformatics approaches and adhering to genomic analysis standards. The study is limited to DNA pattern matching and does not cover applications to phenotypic or agronomic traits. It relies on existing datasets without requiring new sequencing or experimental lab work, aiming to streamline the analysis and interpretation of results.

The rest of this paper is structured as follows: Section II reviews related work. Section III details the dataset employed in the experiments. Section IV describes the research methodology, outlining the techniques used to analyze DNA patterns within the *Sorghum bicolor* genome. Section V presents the experimental results, detailing the outcomes of the pattern-matching analysis and its implications for genetic research. Finally, Section VI concludes the paper.

II. Related Work

Pattern-matching algorithms are fundamental to bioinformatics, especially in DNA sequence analysis. This section provides an overview of foundational and contemporary algorithms that have significantly contributed to the field.

The brute-force, or "naïve" algorithm represents the most elementary approach in pattern detection, performing sequential comparisons of pattern and text characters without preprocessing. Upon a mismatch, it shifts the pattern one character right and continues until a match is found or the text concludes [15]. Despite its simplicity, this method forms the basis for understanding more complex algorithms in pattern matching.

Conversely, the Boyer-Moore algorithm is renowned for its efficiency in text search processes, conducting pattern searches by initiating comparisons from the right end of the pattern. Its innovation lies in the computation shifts following mismatches, leveraging the 'bad character' and 'good suffix' rules to reduce the number of comparisons notably. However, its performance is affected by the alphabet size and pattern length, a factor critical in DNA analysis due to the vast diversity of genetic sequences [14], [16].

The Knuth-Morris-Pratt (KMP) algorithm is frequently employed for identifying specific patterns within DNA sequences. The KMP algorithm improves efficiency by using information from previous matches to avoid redundant comparisons. It constructs a prefix table or failure function to guide the search process. It is particularly effective in analyzing lengthy DNA sequences where quick identification of genetic markers or mutations is crucial [17].

Recent innovations in pattern matching for DNA sequences include the introduction of specialized algorithms designed to optimize search efficiency. Neamatollahi *et al*. [14] introduced three algorithms: First-Last Pattern Matching (FLPM), Processor-Aware Pattern Matching (PAPM), and Least Frequency Pattern Matching (LFPM), aimed at minimizing occurrences of patterns in word processing and searches. These methods have been shown to outperform comparative algorithms in extensive DNA sequence analysis, highlighting their potential to enhance pattern-matching efficiency significantly.

Building on this foundation, recent work by Ibrahim *et al.* [5] has proposed two innovative algorithms, the Enhanced First-Last Pattern Matching (EFLPM) and Enhanced Processor-Aware Pattern Matching (EPAPM), which are advancements of the FLPM and PAPM frameworks. While FLPM employs a character-based approach similar to its predecessors, EPAPM introduces a word-processing technique that further optimizes pattern matching in complex genetic sequences. These developments highlight the ongoing evolution and significance of pattern-matching algorithms in bioinformatics, demonstrating the potential to advance genetic research through enhanced computational techniques.

III. Dataset

Sorghum bicolor, a vital cereal crop, was initially domesticated in Africa between 4,000 and 6,000 years ago. Genomic studies have unveiled evidence of selective breeding for various applications [18]. The BTx623 variety, serving as a model organism for Sorghum research, had its genome fully sequenced in 2009 [19]. This milestone represented the first genome sequencing of both a Saccharinae member and a C4 photosynthetic grass, revealing a genome of approximately 80.88 million base pairs in length. The genomic data, contributed by a global collaboration spearheaded by the University of Georgia and Rutgers [20], is hosted by the Joint Genome Institute and is accessible through the National Center for Biotechnology Information.

Data preprocessing plays a crucial role in the analysis of object patterns [21]–[23], especially in complex genomic studies. Notably, for the *Sorghum bicolor* dataset underwent BUSCO (Benchmarking Universal Single-Copy Orthologs) analysis [24]. This process evaluates genomic data quality and supports genome assembly by identifying complete, duplicated, fragmented, and missing genes. This method, which focuses on detecting single-copy orthologs—genes present precisely once in a genome—allows for assessing gene content completeness concerning a universal set of orthologs. The results from the BUSCO analysis provide a benchmark for the Sorghum dataset's quality, facilitating comparisons across datasets. The outcomes of this analysis are presented in Fig. 1. Further elaboration on the specific BUSCO analysis proce-

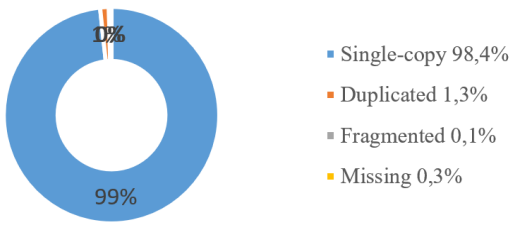

Fig. 1. The observed distribution of BUSCO genes.

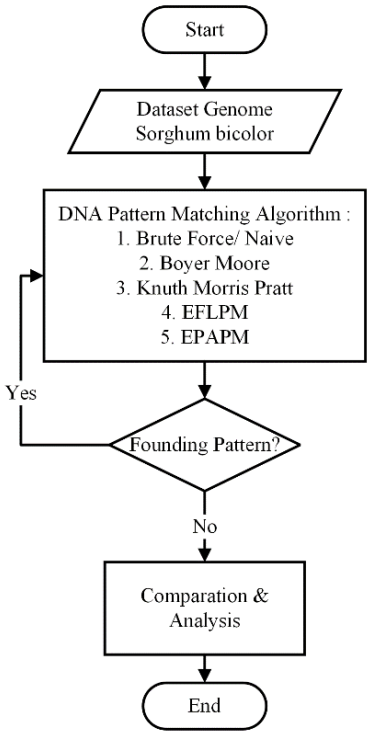

Fig. 2. Research flowchart.

dures and steps undertaken would be beneficial for thoroughly comprehending the dataset's preparation.

IV. Pattern Matching Approaches

This section outlines the DNA pattern-matching algorithms evaluated in our study, focusing on reducing time complexity. Time complexity refers to the duration an algorithm requires to complete relative to the input size [25], [26]. In bioinformatics, where DNA sequences are notably extensive, selecting algorithms capable of efficiently processing large datasets is crucial. Fig. 2 presents the research process flow diagram.

A. Brute Force Algorithm

The brute force, or "naïve", algorithm is a basic yet effective method for pattern matching, characterized by its simplicity and direct approach. It operates by sequentially matching each pattern character against the text from left to right, using two nested loops: the outer loop shifts the starting point of the pattern across the text, while the inner loop performs a character-by-character comparison at each position [27].

This straightforward method does not require a preliminary setup, making it highly accessible for solving pattern-matching problems [28]. However, its simplicity comes at the expense of efficiency, particularly with large datasets. The algorithm's time complexity increases significantly as it exhaustively checks every possible match without shortcuts to discard unlikely candidates [29], limiting its practicality for processing extensive text, such as large genomic sequences common in bioinformatics.

To provide a baseline for comparison with more advanced algorithms, the Brute Force method is employed for its simplicity, particularly valuable in the initial exploratory phases of genetic analysis [30]. It highlights scalability issues when applied to larger sequences. Parameters like matching window size and maximum iterations are adjusted to enhance efficiency and minimize computational overhead in preliminary tests, as supported by studies like Mahmud *et al*. [31], demonstrating the method's effectiveness under specific conditions.

B. Boyer Moore Algorithm

The Boyer-Moore algorithm [32] is a prominent pattern-matching strategy, particularly within bioinformatics for DNA sequence analysis. This algorithm revolutionized pattern matching by introducing an approach that analyzes the pattern and DNA sequence from right to left, contrary to the traditional left-to-right examination. When a match is found within the DNA sequence, the algorithm records the location and continues scanning the sequence iteratively, greatly improving search efficiency until no further matches are detected [33].

The essence of the Boyer-Moore algorithm lies in its utilization of two pivotal heuristics [34]: the "good suffix rule" and the "bad character rule". These heuristics enable the algorithm to expedite the search process by skillfully avoiding unnecessary comparisons. The lousy character rule facilitates skipping sections by aligning the pattern with the rightmost mismatched character in the DNA sequence, leveraging the static nature of the DNA sequence to bypass irrelevant comparisons efficiently. Conversely, the good suffix rule optimizes search efficiency by enabling pattern shifts to match a previously found suffix in the DNA sequence, effectively using known matches as anchors for future comparisons. This strategic approach reduces the number of comparisons needed. It capitalizes on the fixed and unchanging structure of DNA sequences, making the Boyer-Moore algorithm a powerful tool for DNA sequence analysis.

Further enhancing its applicability, the Boyer-Moore algorithm is renowned for its efficiency in searching large texts, making it ideal for bioinformatic applications involving extensive DNA sequences. The choice of this algorithm was based on its proven capability to reduce the number of comparisons in pattern searches, which significantly enhances performance when dealing with complex genetic data [32].

C. Knuth-Morris-Pratt (KMP) Algorithm

The Knuth-Morris-Pratt (KMP) algorithm [35] is designed for efficient pattern matching in extensive texts. Its foremost advantage lies in its ability to perform pattern matches in linear time, independent of the length of the pattern or the text. This attribute renders it particularly valuable for large-scale text-processing applications.

A key feature of the KMP algorithm is its preprocessing phase, where it constructs a partial match table—also known as the failure function or the longest prefix suffix array. This table is crucial as it records the pattern's structure, allowing the algorithm to skip unnecessary comparisons during the matching phase. Using the pattern to build this table, the KMP algorithm

ensures an efficient matching process. It significantly reduces redundant comparisons and accelerates text traversal without the need for backtracking [36]. During the matching phase, the algorithm uses the data from the partial match table to quickly move to the next point of inspection, thereby optimizing the efficiency of text evaluation.

The KMP algorithm is known for its efficient performance, conducting searches in linear time, which is essential for bioinformatics where prompt and precise pattern recognition is crucial. This algorithm employs a failure table that uses data from previous matches to reduce unnecessary comparisons, thus accelerating the detection of common genetic markers in extensive sequences. Adjustments to the length of the failure table and pattern selection criteria have significantly improved its effectiveness in decoding complex genomes. The durability and proven track record of KMP in computational biology make it well-suited to meet the demanding needs of genetic research, underscoring its adaptability and effectiveness in diverse scientific studies [17].

D. Enhanced First-Last Pattern Matching (EFLPM) Algorithm

The Enhanced First-Last Pattern Matching (EFLPM) algorithm is a significant bioinformatics innovation that optimizes pattern-matching efficiency within DNA sequences. Building on the First-Last Pattern Matching (FLPM) algorithm, EFLPM introduces a novel approach by merging the preprocessing and matching phases into a cohesive, singular process [5], [14]. This integration is designed to streamline the pattern-matching procedure, reducing the overall time complexity of analyzing various pattern sizes.

Central to EFLPM's approach is the strategic emphasis on the pattern's first and last characters. By contrasting these characters against the corresponding characters in the text, the algorithm capitalizes on their distinctiveness and indicative nature, which are crucial for effectively identifying potential matches [5]. This method leverages the unique properties of DNA sequences, where specific nucleotide patterns can profoundly impact the results of genetic analyses.

The EFLPM algorithm was chosen due to its innovative approach of integrating preprocessing and matching phases, which reduces the overall time complexity of analyzing patterns of various sizes. This algorithm's capability to focus on distinctive characters at the pattern's boundaries makes it particularly effective for genetic analyses where mutations often occur at known loci. The ability to effectively target these critical areas enhances the precision and relevance of gene research, demonstrating the algorithm's substantial value in bioinformatics [37].

E. Enhanced Processor-Aware Pattern Matching (EPAPM) Algorithm

Building on the Enhanced First-Last Pattern Matching (EFLPM) algorithm, which is designed to optimize pattern-matching efficiency by integrating preprocessing and matching phases, the Enhanced Processor-Aware Pattern Matching (EPAPM) algorithm [5] takes these capabilities further by incorporating parallel processing techniques. This advancement significantly enhances the performance by leveraging the computational power of modern processors, making it well-suited for handling large-scale genomic datasets.

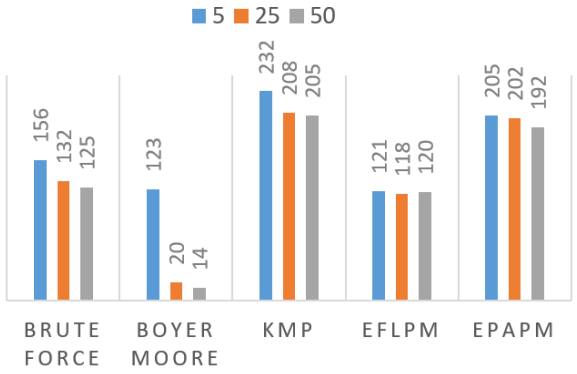

Fig. 3. Execution time comparisons (in seconds).

A fundamental aspect of EPAPM is its strategic focus on utilizing the distinct characteristics of a pattern's first and last characters, similar to EFLPM. By comparing these characters with their counterparts in the text, EPAPM efficiently identifies potential matches. This focused approach reduces the complexity of analyzing patterns of varying sizes. Combined with parallel processing, it allows for simultaneous analysis of multiple data segments, accelerating the overall pattern-matching process. The decision to adopt EPAPM was driven by its ability to effectively harness modern computing capabilities, responding to the growing demands of genetic data analysis [38].

V. Experimental Results

This study assesses the time complexity of various DNA pattern-matching approaches outlined in Section 2, including the Boyer-Moore, Brute Force, KMP, EFLPM, and EPAPM algorithms. It utilizes Google Colab, a computational platform that facilitates the execution of these algorithms.

We conducted an experimental analysis to compare the execution times of each algorithm across different DNA sequence lengths: 5, 25, and 50 nucleotides. Each algorithm's time to complete the pattern-matching task was recorded, considering short, medium, and long sequences. The execution time comparisons are presented in Fig.3, while a detail of these algorithms, emphasizing their time complexities and foundational principles, is summarized in TableI.

The Brute Force algorithm, characterized by its straightforward approach, scans for every possible match within the DNA sequence. While simple and needing a preprocessing requirement, its inefficiency with large datasets limits its practicality. However, it is helpful in situations with smaller inputs or where additional optimization techniques can narrow the search scope.

TABLE I
COMPARISON OF THE PATTERN-MATCHING APPROACHES EMPLOYED IN THE STUDY.

Algorithm	Time Complexity	Key Ideas
Brute Force [39]	$O(nm)$	Searching with all alphabets
Boyer Moore [33]	$O(n/m)$	Bad character and good suffix heuristics to determine the shift distance
Knuth-Morris-Pratt [35], [40]	$O(n+m)$	Construct an automation from the pattern
EFLPM [5]	$O(nm)$	Focuses on two characters (first and last)
EPAPM [5]	$O(n/m)$	Words with multiple characters to recognize windows

The Boyer-Moore algorithm exploits specific patterns and DNA sequence attributes to sidestep non-essential comparisons, offering a faster alternative to the Brute Force method with a time complexity of $O(n/m)$. Its efficiency shines when dealing with long patterns frequently occurring within the DNA sequence. Although pattern preprocessing is a requisite, which could be seen as a drawback for extensive patterns, this step's one-time execution offers multiple search applications, enhancing long-term efficiency.

Employing a preprocessed table to avoid unnecessary comparisons, the KMP algorithm demonstrates its efficiency with an $O(n + m)$ time complexity. This feature proves especially advantageous for handling extensive DNA sequences, underscoring its utility in large-scale text processing and bioinformatics. The EFLPM algorithm, incorporating fuzzy logic to accommodate DNA sequence errors and mutations, mirrors the Brute Force algorithm's $O(nm)$ time complexity. It distinguishes itself by focusing on comparing unique characters, potentially reducing the required number of character comparisons and thereby boosting efficiency for extensive patterns and texts, despite necessitating extra preprocessing that increases its time complexity. Lastly, the EPAPM algorithm leverages parallel computing to expedite the pattern-matching process, achieving a quicker performance than the Brute Force method with a time complexity of $O(n/m)$. Considering the intricacies of processor architecture, such as branch prediction and parallel processing, significantly enhances efficiency, offering a formidable tool for pattern-matching endeavors.

The findings particularly emphasize the necessity for ongoing innovation in DNA pattern-matching algorithms. Future enhancements could focus on integrating various analytical modalities to enhance predictive accuracy and the detection of genetic anomalies and incorporating approaches from recent research [41]–[43] could lead to significant improvements in how genetic data is interpreted, potentially revolutionizing approaches to diagnosing and understanding genetic conditions.

Moreover, there is promising potential in developing hybrid algorithms that combine the strengths of current methodologies, as evidenced by previous successes in this area [44]–[46]. Such hybrid approaches could offer more refined precision and practicality, enhancing the capabilities of genetic research tools. These innovations could lead to more robust bioinformatics tools tailored to meet the complex demands of modern genetic research and diagnostics, thereby significantly advancing the capabilities within the field of bioinformatics.

VI. CONCLUSION

This study engaged in an in-depth comparative analysis of several DNA pattern-matching algorithms, specifically Enhanced First-Last Pattern Matching (EFLPM), Enhanced Processor-Aware Pattern Matching (EPAPM), Knuth-Morris-Pratt (KMP), Boyer-Moore, and the Brute Force method, focusing on their time efficiency, effectiveness, and accuracy specifically within the *Sorghum bicolor* genome analysis.

The empirical analysis of execution times elucidates a variance in efficiency among the algorithms contingent upon the length of the DNA sequences. EFLPM proved particularly efficient with short strings, evidenced by its execution time of 121 seconds. In contrast, Boyer-Moore was proficient with long strings, demonstrating a rapid execution time of 14 seconds. EFLPM exhibited consistent performance across a range of string lengths, suggesting its resilience and adaptability for various pattern-matching scenarios. The findings accentuate the importance of selecting an appropriate algorithm based on time complexity within bioinformatics. Such a selection is pivotal, as it substantially dictates the practicality and precision of DNA sequence analyses, considerably influencing the outcomes of both research and diagnostic procedures.

Analytically, these results suggest a potential for adapting these algorithms for other complex genomes, enhancing their applicability in a wider range of bioinformatics applications. Such adaptability could be crucial for advancing genetic research and developing new therapeutic strategies. Moreover, future enhancements of these algorithms could integrate machine learning to improve precision in predicting genetic anomalies, potentially revolutionizing genetic diagnostics and treatment approaches. In summary, this study emphasizes the critical role of algorithm selection in bioinformatics and projects how these findings could influence the evolution of genetic research and therapeutic development, marking significant contributions to the field.

ACKNOWLEDGEMENT

This work was partially supported by Institut Teknologi Sepuluh Nopember, under Grant No. 1089/PKS/ITS/2024.

REFERENCES

[1] R. Termanini, "The amazing human dna system explained," pp. 17–37, 2020.
[2] M. Asmamaw and B. Zawdie, "Mechanism and applications of crispr/cas-9-mediated genome editing," *Biologics : Targets Therapy*, vol. 15, pp. 353–361, 2021.
[3] T. M. Escobar, A. Loyola, and D. Reinberg, "Parental nucleosome segregation and the inheritance of cellular identity," *Nature Reviews Genetics*, vol. 22, pp. 379 – 392, 2021.
[4] M. Najam, R. Rasool, H. Ahmad, U. Ashraf, and A. Malik, "Pattern matching for dna sequencing data using multiple bloom filters," *Biomed Res Int*, vol. 2019, p. 7074387, 2019.

[5] O. Ibrahim, B. Hamed, and T. El-Hafeez, "A new fast technique for pattern matching in biological sequences," *J Supercomput*, vol. 79, pp. 367–388, 2023.

[6] F. B. Ashraf, "Mfea: An evolutionary approach for motif finding in dna sequences," *Informatics in Medicine Unlocked*, vol. 21, 11 2020.

[7] F. Cruz, J. Gómez-Garrido, M. Gut, T. S. Alioto, J. Pons, J. Alós, and M. Barcelo-Serra, "Chromosome-level assembly and annotation of the xyrichtys novacula (linnaeus, 1758) genome," *DNA Research*, vol. 30, no. 5, 2023.

[8] Y.-H. Hsieh, S. C. Hidayati, W.-H. Cheng, M.-C. Hu, and K.-L. Hua, "Who's the best charades player? mining iconic movement of semantic concepts," in *Proc. Int. Conf. MultiMedia Modeling*, 2014, pp. 231–241.

[9] Z.-M. Liu, Y.-Y. Chen, S. Hidayati, S.-C. Chien, F.-C. Chang, and K.-L. Hua, "3d model retrieval based on deep autoencoder neural networks," in *Proc. Int. Conf. Signals and Systems*, 2017, pp. 290–296.

[10] E. Massip, S. C. Hidayati, W.-H. Cheng, and K.-L. Hua, "Exploiting category-specific information for image popularity prediction in social media," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops*, 2018, pp. 45–46.

[11] N. Sinkovics, "Pattern matching in qualitative analysis," in *The SAGE Handbook of Qualitative Business and Management Research Methods: Methods and Challenges*. SAGE Publications Ltd, May 2018, pp. 468–484.

[12] A. A. Karcioglu and H. Bulut, "The wm-q multiple exact string matching algorithm for dna sequences," *Computers in Biology and Medicine*, vol. 136, p. 104656, 2021.

[13] Y. Kim, M. Imani, N. Moshiri, and T. Rosing, "Geniehd: Efficient dna pattern matching accelerator using hyperdimensional computing," in *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2020, pp. 115–120.

[14] P. Neamatollahi, M. Hadi, and M. Naghibzadeh, "Efficient pattern matching algorithms for dna sequences," in *Proc. International Computer Conference, Computer Society of Iran*, 2020.

[15] A. V. Aho, "Algorithms for finding patterns in strings," in *Handbook of Theoretical Computer Science, Algorithms and Complexity*. Elsevier, 1990, pp. 255–300.

[16] R. S. Boyer and J. S. Moore, "A fast string searching algorithm," *Commun. ACM*, vol. 20, no. 10, p. 762–772, 1977.

[17] C. Wang, L. Gong, S. Lei *et al.*, "Genseq+: A scalable high-performance accelerator for genome sequencing," *IEEE/ACM Trans Comput Biol Bioinform*, vol. 18, no. 4, pp. 1512–1523, 2021.

[18] X. Wu, Y. Liu, H. Luo, L. Shang, C. Leng, Z. Liu, Z. Li, X. Lu, H. Cai, H. Hao, and H.-C. Jing, "Genomic footprints of sorghum domestication and breeding selection for multiple end uses," *Molecular Plant*, vol. 15, no. 3, pp. 537–551, 2022.

[19] Y. Cheng, W. Deng, Y. Lu, S. Han, Y. Lv, G. Zeng, C. Zhou, D. Zhang, and X. Shen, "Establishment of sorghum btx623 immature embryos genetic transformation and regeneration system," *Molecular Plant Breeding*, vol. 11, no. 5, 2020.

[20] Q. lin XIAO, Z. LI, Y. yun WANG, X. bin HOU, X. mei WEI, X. ZHAO, L. HUANG, Y. jun GUO, and Z. zhai LIU, "Genome-wide identification, expression and functional analysis of sugar transporters in sorghum (sorghum bicolor l.)," *Journal of Integrative Agriculture*, vol. 21, no. 10, pp. 2848–2864, 2022.

[21] M. F. Aza, N. Suciati, and S. C. Hidayati, "Performance study of facial expression recognition using convolutional neural network," in *Proc. Int. Conf. Science in Information Technology*, 2020, pp. 121–126.

[22] S. C. Hidayati, T. W. Goh, J.-S. G. Chan, C.-C. Hsu, J. See, L.-K. Wong, K.-L. Hua, Y. Tsao, and W.-H. Cheng, "Dress with style: Learning style from joint deep embedding of clothing styles and body shapes," *IEEE Transactions on Multimedia*, vol. 23, pp. 365–377, 2021.

[23] A. Widyadhana, S. C. Hidayati, D. A. Navastara, and Y. Anistyasari, "ASF-LLRDA: locality-regularized linear regression discriminant analysis with approximately symmetrical face preprocessing for face recognition," in *Asia Pacific Signal and Information Processing Association Annual Summit and Conference*, 2023, pp. 2031–2036.

[24] J. Kim and C. Kim, "A beginner's guide to assembling a draft genome and analyzing structural variants with long-read sequencing technologies," *STAR Protocols*, vol. 3, no. 3, p. 101506, 2022.

[25] T.-H. Tsai, W.-C. Jhou, W.-H. Cheng, M.-C. Hu, I.-C. Shen, T. Lim, K.-L. Hua, A. Ghoneim, M. A. Hossain, and S. C. Hidayati, "Photo sundial: Estimating the time of capture in consumer photos," *Neurocomputing*, vol. 177, pp. 529–542, 2016.

[26] G. Marzani, N. Suciati, and S. C. Hidayati, "Morphological preprocessing for low-resolution face recognition using common space," in *Proc. Int. Conf. ICT for Smart Society*, 2021, pp. 1–5.

[27] P. B. Abhale, "Handwritten english alphabet recognition," *International Journal for Research in Applied Science and Engineering Technology*, 2021.

[28] C. Barutu, N. Abdi, T. Informatika, S. Pontianak, and Kalimantan, "Brute force algorithm implementation of dictionary search," *Jurnal Info Sains : Informatika dan Sains*, 2020.

[29] Y. Riwanto, M. Nuruzzaman, S. Uyun, and B. Sugiantoro, "Data search process optimization using brute force and genetic algorithm hybrid method," *International Journal on Informatics for Development*, 2023.

[30] B. Browning, X. Tian, Y. Zhou, and S. Browning, "Fast two-stage phasing of large-scale sequence data." *American journal of human genetics*, 2021.

[31] P. Mahmud, A. Rahman, and K. H. Talukder, "An improved hashing approach for biological sequence to solve exact pattern matching problems," *Applied Computational Intelligence and Soft Computing*, 2023.

[32] L. Riza, F. D. Pratama, E. Piantari, and M. Fahsi, "Genomic repeats detection using boyer-moore algorithm on apache spark streaming," *elecommunication Computing Electronics and Contro)*, 2020.

[33] P. Pangestu and S. E. Wahyuningrum, "Word search using boyer-moore algorithm," 2021.

[34] D. Manalu, M. I. Hutapea, R. L. Raja, A. P. Silalahi, H. G. Simanullang, E. Panggabean6, A. Info, D. Robinson, and Manalu, "Designing a chatbot application for web-based english learning using boyer moore algorithm," *International Journal Of Computer Sciences and Mathematics Engineering*, 2023.

[35] D. Anggreani, D. P. I. Putri, A. N. Handayani, and H. Azis, "Knuth morris pratt algorithm in enrekang-indonesian language translator," *Proc. International Conference on Vocational Education and Training*, pp. 144–148, 2020.

[36] M. Baig, S. Sivakumar, and S. Nayak, "Optimizing performance of text searching using cpu and gpus," *Advances in intelligent systems and computing*, vol. 1119, pp. 141–150, 2020.

[37] P. Blaney, B. A. Walker, A. Mikulasova, D. Gagler, B. Ziccheddu, B. T. Diamond, Y. Wang, L. Hou, S. Liu, A. Poos, A. Cirrincione, K. Maclachlan, E. Mai, H. Goldschmidt, S. Z. Usmani, M. Raab, N. Weinhold, O. Landgren, F. E. Davies, F. Maura, and G. Morgan, "Somatic hypermutation in enhancer regions shapes non-coding myeloma genome, generating dna breaks and driving etiology through mutation and structural variation," *Blood*, 2023.

[38] K. Yelick, A. Buluç, M. Awan, A. Azad, B. Brock, R. Egan, S. Ekanayake, M. Ellis, E. Georganas, G. Guidi, S. Hofmeyr, O. Selvitopi, C. Teodoropol, and L. Oliker, "The parallelism motifs of genomic data analysis," *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, vol. 378, 2020.

[39] S. Phalke, Y. Vaidya, and S. Metkar, "Big-o time complexity analysis of algorithm," *2022 Proc. International Conference on Signal and Information Processing*, pp. 1–5, 2022.

[40] S. S.Dawood, "Empirical performance evaluation of knuth morris pratt and boyer moore string matching algorithms," vol. 23, pp. 134–143, 2020.

[41] S. C. Hidayati, Y.-L. Chen, C.-L. Yang, and K.-L. Hua, "Popularity meter: An influence- and aesthetics-aware social media popularity predictor," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 1918—1923.

[42] S. C. Hidayati and Y. Anistyasari, "Body shape calculator: Understanding the type of body shapes from anthropometric measurements," in *Proc. ACM Int. Conf. Multimedia Retrieval*, 2021, pp. 461—-465.

[43] S. C. Hidayati, M. R. Fiqih Thalib, and A. Munif, "The influence of user profile and post metadata on the popularity of image-based social media: A data perspective," in *Proc. Int. Conf. Artificial Intelligence in Information and Communication*, 2024, pp. 806–811.

[44] D. D. Thang, S. C. Hidayati, Y.-Y. Chen, W.-H. Cheng, S.-W. Sun, and K.-L. Hua, "A spatial-pyramid scene categorization algorithm based on locality-aware sparse coding," in *Proc. IEEE Int. Conf. Multimedia Big Data*, 2016, pp. 342–345.

[45] S. C. Hidayati, K.-L. Hua, Y. Tsao, H.-H. Shuai, J. Liu, and W.-H. Cheng, "Garment detectives: Discovering clothes and its genre in consumer photos," in *Proc. IEEE Conf. Multimedia Information Processing and Retrieval*, 2019, pp. 471–474.

[46] D. Aulia, R. Sarno, S. C. Hidayati, and M. Rivai, "Optimization of the electronic nose sensor array for asthma detection based on genetic algorithm," *IEEE Access*, vol. 11, pp. 74 924–74 935, 2023.