Available online at www.qu.edu.iq/journalcm
JOURNAL OF AL-QADISIYAH FOR COMPUTER SCIENCE AND MATHEMATICS
ISSN:2521-3504(online) ISSN:2074-0204(print)

A Review of Planted Motif Problem Types

Wajih Abdul Ghani Abdul Hussain[a], Hussein Keitan Al-Khafaji[b], Thekra Abbas[c]
[a]*Mustansiriyah University, College of Science, Department of Computer Science, Baghdad, Iraq, Email: wajeeh.abdulgani@uomustansiriyah.edu.iq*
[b]*Al-Rafidain University College, Department of Computer Communication Engineering, Baghdad, Iraq, Email: hussain.ketan.elc@ruc.edu.iq*
[c]*Mustansiriyah University, College of Science, Department of Computer Science, Baghdad, Iraq, Email: thekra.abbas@uomustansiriyah.edu.iq*

ARTICLE INFO

Article history:
Received: 27/ 10 / 2024
Rrevised form: 4 /11/2024
Accepted : 13 /1/ 2025
Available online: 30 /3/2025

Keywords:

Bioinformatics, motif mining, planted motif mining, deep learning

ABSTRACT

The field of bioinformatics faces significant challenges in the extraction of meaningful knowledge from the vast and complex biological data, particularly in gene regulation and motif identification. This research delves into the intricate problem of planted motif mining (PMP), an NP-complete challenge critical for numerous applications, including disease diagnostics, forensic medicine, and environmental monitoring. Various algorithms have been developed to address this problem, ranging from deterministic polynomial-time algorithms to advanced deep learning methods. This study provides a comprehensive review of these methodologies, highlighting their efficiencies, limitations, and practical applications. The paper discusses profile-based and pattern-based algorithms, emphasizing the importance of approximation algorithms for handling extensive biological data. Additionally, the role of deep learning in motif mining is explored, showcasing the advancements brought by CNN-based, RNN-based, and hybrid models. The integration of these innovative techniques holds promise for improving the accuracy and speed of motif identification, ultimately enhancing our understanding of genetic regulation and its implications in health and disease.

MSC..

https://doi.org/10.29304/jqcsm.2025.17.11961

1. Introduction

The rapid growth of biological data has prompted collaborative efforts to understand and analyze this data to enhance daily life. Despite significant efforts, bioinformatics still requires substantial work for effective knowledge extraction. There are a lot of factors that make the challenges harder; including vast amount of data contained within a genome, lack of techniques to extract useful knowledge, and complexity of laboratory tests in biology for validating accurate information.

Genes are distinct segments on DNA chromosomes responsible for encoding proteins. Gene expression begins when a transcription factor binds to a specific region called the binding site, located before the gene. In the eukaryotic nucleus, a transcription factor protein plays an important role in regulating transcription activation. These binding sites exhibit similar patterns, referred to as motifs [1].

∗Corresponding author: Wajih Abdul Ghani Abdul Hussain

Email addresses: wajeeh.abdulgani@uomustansiriyah.edu.iq

Communicated by ‘sub etitor’

An essential objective in bioinformatics involves uncovering motifs. The motif-finding problem is considered NP-complete, leading to ongoing research to solve it using deterministic algorithms with polynomial time complexity. [2]

The benefit of motif mining is that it is useful in numerous applications, such as detecting the predisposition to disease, Diagnostics, forensic medicine, prosthesis laboratories, criminal laboratories, corpse identification, medicines manufacturing, and uncovering chemical and nuclear environmental pollution that causes genetic mutations. [3]

Gene regulation from single cell detection is very important for many cancers remedy. Detecting mutations allows for the administration of drugs tailored to the genetic makeup of a patient's tumor or cancer cells. [4]

Planted motif problem (PMP) is one of the famous disciplines in motif mining. This problem involves a group of bio sequences with similar or different lengths, beside two parameters l and d such that $0 \leq d < l < n$. the aim is specifying substring of length l with d allowed mismatch. These substrings (with length l) are named motifs.

For example, consider three sequences: CATACGT, ACAAGTC, and AATCGTG. With a motif length (l) of 3 and a maximum allowed mismatch (d) of 1, the substring CAT is identified as a motif. In the first sequence, CAT occurs at the initial position with no mismatches. In the second sequence, it appears at the second position with one mismatch, and in the third sequence, it emerges at the third position with one mismatch as well. Another perspective is that CAT qualifies as a motif because substrings CAT from the first sequence, CAA from the second sequence, and CGT from the third sequence fall within the 1-neighbourhood of CAT.

PMS algorithms fall into two main categories: **profile-based** algorithms and **pattern-based** algorithms [5]. The first category focuses on predicting where motifs will initiate within each sequence, while the second category aims to discover motifs directly. Both types of algorithms classified as either **approximate** or **exact**. Approximation algorithms, also known as heuristic algorithms, might not always provide accurate results. These algorithms use a location frequency matrix to conclude either upon meeting the specified end condition or reaching a predefined number of iterations. [6, 7].

On the other hand, exact algorithms use exhaustive enumeration search and consistently yield optimal solutions [8]. Examples of this type are Moby Dick [9] and YMF [10, 11]. Its time complexity becomes impractical due to the extensive biological data, making it suitable only for searching short sequence motifs.

In the field of PMS, approximation algorithms are generally speedup and more common than exact algorithms, although they lack a guarantee of always producing the correct motif. This review exhibits some frequently different types of planted motif problems.

2. Literature Review of Motif Identification Methods

In reference [12], the researchers introduced algorithms PMSi and PMSP, drawing inspiration from concepts employed in PMS1. These algorithms exhibit enhanced space efficiency compared to PMS1, leading to improve running times across various scenarios. Notably, PMSP surpasses both PMS1 and PMSi in challenging instances, demonstrating superior performance. Additionally, it competes favorably with existing exact algorithms while utilizing considerably less memory. This feature enables PMSP to successfully solve the previously unsolved challenging instance (17, 6).

PMS1 effectively solves instances such as (9, 2), (11, 3), and (13, 4) within a few minutes, and voting method successfully addresses these cases along with (15, 5) in 22 minutes. However, as the value of d increases, the algorithms experience higher memory requirements. Notably, PMSP demonstrates the capability to handle challenging scenarios such as (l, d) = (15, 5) and (17, 6).

PMSprune [13] adopts a similar approach to [12], with the inclusion of additional features. This approach produced the strategy of generating group of l-mers such that the difference between the l-mers in first sequence and these l-mers is equal or less than d, and then this approach checks these l-mers is motif or not. PMSprune presents an improved algorithm to generate neighbors and effective technique to prune the results.

The algorithm starts with constructing the similarities (neighborhoods) for every l-mer (x) in first sequence S1 using a depth d tree structure T(x). Each node in T(x) represents a similar (neighbor) of S1. This algorithm traverse the tree using depth first order and checks the largest and smallest distance to neighbors in all sequences. If the d H (y, S) value falls within the specified d, it outputs y; otherwise, it undergoes a pruning process, where y represents l-mers in each sequence and S represents set of t sequences. This dynamic pruning technique will be eliminate the search space.

The stemming algorithm [14] aims to reduce the complexity of computational search, independent of the alphabet size $|\sum|$. It begins by constructing a set of candidate motifs and then generates neighborhoods for these candidates, and then, the neighbors are intersected to create a group of candidate motifs that similar with each other. In the first, the algorithm identifies all necessary motifs with a few non-motifs and place them in superset C. after that, a specific strategy of pruning can be applied to extract the necessary motifs from C. Instead of scanning through all neighborhoods, this method operates within a search scope that adjusts according to the size of the candidate set C, reducing the computational search region.

The PMS4 algorithm[15] serves as an acceleration mechanism. This approach operates in two distinct steps: the initial step involves identifying a set of candidate motifs, while the subsequent phase verifies each individual candidate motif, it is a true motif or not. The fast approach employed by PMS4 follows two steps, the first step, it executes any effective algorithm for searching on a subset of k chosen sequences (where $k < t$) rather than of all t sequences, aiming to specify a C collection of motifs. Significantly, the (l, d) patterns recognized in the initial step encompass a superset of motifs found in the t sequences. The particular choice of k sequences might differ across various algorithms. In the second step, this algorithm evaluates the truth of each l-mer in set C.

PMS5 [16] efficiently expands the algorithm of PMS1 by incorporating innovative concepts along with the neighbor generation approach from PMSprune. This algorithm starts with a set S comprising t sequences, producing the collection of (l, d) motifs, denoted as Ml,d(S). For each collection of 3(A, B, C) l-mers, which represents the popular similar motifs (neighbors) such that A represents an l-mer from S1, B is an l-mer from S2i, C is an l-mer from S2i+1, where $1 \le i \le (t-1)/2$. For determining the popular similar neighbors, the algorithm utilizes the concept of tree from PMSprune and (ILP) Integer Linear Programming involving 10factors. This method includes an initial preprocessing stage that creates numerous ILP instances. These instances must be resolved and stored in memory as a lookup table, resulting in higher memory demands.

The PMS6 algorithm[17] builds upon the PMS5 algorithm, enhancing its preprocessing steps and reducing the size of the lookup table through incorporation of effective method of hashing. Unlike PMS5, this method distinguishes itself by determining motifs equivalent to l-mer (x) in sequence S1 through a two-phase process. In the initial phase, five classes were used to handle triplets of l-mers (A, B, C), denoted as C(n1,..., n5), based on computed values of n1 to n5. These triplets are assigned to their respective classes according to the calculated values. The computation of n1 to n5 for each triplet follows the methodology proposed by PMS5. In the second phase, the algorithm calculates the motifs from sequences x, S2i, and S2i+1, where i ranges from 1 to (t-1)/2, based on equivalence class values. It's worth noting that PMS6 consistently demonstrates faster performance in comparison to PMS5.

The PairMotif algorithm [18] efficiently diminishes the search space of motif by producing candidate motifs from selected pairs with relatively greater distances. This process unfolds through three iterative steps. In the initial step, it creates pairs of l-mers, where one is from the S1, and the other is from the remaining Sr where $(2 \le r \le t)$, ensuring they are positioned within a Hamming distance of 2d. In the second step, two rules of filter are applied to each selected pair of l-mers, reducing the number of l-mers to be examined in the next stage. In the final step, the method identifies popular similar neighbors for each remaining pair of l-mers, for each widely occurring neighbor y, it carries out a validation process to ascertain its status as a motif. The experimental findings from PairMotif validate its effectiveness and consistency across different lengths of sequences.

The qPMS7 algorithm [19] expands upon the concept and shows more flexible method in qpms field. This method supposes that two integers i , j ranges between (1 to t) and there are l-mers u and v such that $u \in Si$, $v \in Sj$, and $M \in \{Bd(u) \cap Bd(v)\}$ which represents qpms motif, excluding Si and Sj. Consequently, for every two sequences such (sequence (i) and sequence (j), the method will examines all potential pairs of l-mers (u, v) to distinguish all elements within the intersection of Bd(u) and Bd(v).

PMS8 [20] offers an effective solution to the planted search motifs by proffer the innovative concept of neighbor production. Additionally, it establishes efficient condition for 3 (l-mers) to share popular neighbors. The PMS8 algorithm works well in two distinct phases: the first one is sample driven, the second is the pattern driven.
During the first phase, it initiates generation of l-mers from various sequences, storing them in a two dimensional array (R) with size of $(t \times (n-l+1))$, Where the index of each row matches the sequence number of the input. Initially, the algorithm chooses an l-mer (u) from the first sequence, adds it to the stack, and eliminates all l-mers (y) from matrix that are dissimilar with y is greater than 2d.afterwards, it iteratively chooses one candidate motif (l-mer)

from each sequence,adds it to the stack (pushing), and eliminates l-mers from matrix (R) that are not part of the popular neighborhoods in the stack. This process continues until the size of stack arrives a specified value.
Upon reaching the threshold, the algorithm transitions to the pattern driven step. In this phase, it generates the common neighborhoods for the l-mers in the stack. After that, the algorithm checks every neighbor l-mer, to see if there's a (l, d) motif planted within it.

qPMS9 [21] represents highly effective parallel quorum planted motif search technique, offering a significant enhancement in the execution time compared to PMS8. The advancements of qPMS9 over PMS8 are multi-faceted. Firstly, it introduces a novel string reordering approach, effectively boosting performance by combining an enhanced method of pruning within the space of search. Additionally, qPMS9 addresses the resolution of qPMS instances that were not previously covered by PMS8. Notably, qPMS9 successfully tackles l, d as follows: (28, 12), (30, 13) by using a single-core computer within a sensible time frame.

In the reference [22], a sophisticated and highly effective randomized technique called qPMS10 for addressing the PMS problem can be introduced.
The algorithm comprises three stages. Initially, it acquires sample sequences, with minimal time consumption. Next, it applies qPMS9 to the sample, which could be parallelized given the availability of a parallel version. Finally, it verifies whether each candidate motif qualifies as a genuine motif for the original input. This algorithm deals with (30, 15) and q = 100.
We've evaluated our algorithm using two different sets of inputs. The first set follows the conventional parameters, with n = 20 and m = 600, while the second set has larger dimensions with n = 1000 and m = 600.

In their work [23], CaiyanJia, Ruqian Lu, and Lusheng Chen introduced Apriori-Motif algorithm, this algorithm works without knowing the motif length (l). This algorithm employs a specific search strategy (breadth-first search) to prune the search region, this method utilized the concept of downward closure property used in Apriori. The work used the suffix tree and index structure inthis work.
This method used a breadth-first search to explore the entire pattern space, which is indexed by the consensus tree structure. Besides that, it can extract motifs with lengths ranging from d + 1 to l, and by using the downward closure property.

In reference [24], the algorithm Ex-Motif, designed to extract all frequent structured motifs with a quorum of q. This method holds potential applications, such as the identification of single/composite regulatory binding sites included into DNA sequences.
This approach utilized an inverted index of symbol positions and it achieved the enumeration of all structured motifs through positional joins over this index. Simultaneously, variable gap constraints are taken into account during these joins, resulting in significant efficiency gains. To optimize both time and space, the algorithm retained only the start positions of each intermediate pattern during the positional join process.

The researchers in reference [25]introduced a scalable parallel system called ACME. This approach depends on a methodical strategy capable of handling sequences of gigabyte scale, marking the inaugural support for super maximal motifs. ACME is a flexible parallel system adaptable for utilization on desktops with multi-core processors or across thousands of processors in cloud environments.
ACME is capable of accommodating sequences that are three orders of magnitude lengthier, such as the entirety of the human genome in DNA format, and can manage large alphabets like the English alphabet representing Wikipedia content. It can scale up to 16,384 CPUs on a supercomputer and allows for flexible deployment in cloud environments. The novelty of ACME stems from (i) the sequence in which it navigates through the search space and (ii) the manner in which it accesses information within the suffix tree.
This system works as follows:

- A parallel method suggested that breaks down the motif extraction procedure into smaller tasks, enabling the effective utilization of thousands of processors. ACME can expand to 16,384 processors on an IBM BlueGene/P supercomputer and can complete a query in just 18 minutes, a task that would require over 10 days on a high-end multi-core machine.
- An automated tuning technique was developed that enables nearly optimal resource utilization, particularly beneficial in cloud-based environments.
- A cache-efficient method was devised for traversing the search space, reducing serial execution time by nearly tenfold.

- The experimental assessment involves large real-world datasets across various architectures, both locally and in cloud environments.

Pavesi and et al, introduced in reference [26] the Weeder algorithm, which relies on identifying count matching patterns with particular and highly pronounced mismatches. Initially, motifs are depicted using a consensus sequence. Then, by comparing the consensus with k-mers of sequences and allowing for a specific number of mutations, after that, the k-mers are evaluated for significance using specific metrics.

The FMotif algorithm [27] introduced a constructed suffix tree to identify motifs with lengthy parameters of l and din extensive sequences of DNA under the ZOMOPS constraints. This suggested tree enhances efficiency compared to the standard suffix tree by minimizing redundant scans of sequences.

Buhler and et al in reference [28] devised a random projection algorithm for a PMP that condenses each l-mer in the input data into a smaller space through hashing.
At first, a transformation occurs, moving from an l-dimensional space to a k-dimensional subset for all subsequences within the input set. This is achieved by randomly picking k positions from the l available positions. Subsequently, each l-mer is assigned to the suitable bucket via hashing through this transformation. Subsequently, buckets containing l-mers surpassing a certain threshold are deemed "qualified buckets". To ensure reliability, random hashing is repeated n times to guarantee each qualified bucket appears more than once. In the end, a profile is generated for every eligible bucket to identify the most likely l-mer in the sequence, presented as consensus sequences.

A proposed algorithm named Ref-Select [29] aims to choose reference sequences for PMP. Reference sequences are those devoid of motif instances, thus the method endeavors to select reference sequences that produce as few potential motifs as feasible. The method comprises two primary sphases. Initially, for each pair of sequences in the dataset D, The count of potential motifs produced from these pairs is determined by assessing the Hamming distance between each pair of l-mers. Subsequently, the reference set is chosen to minimize the set's count of candidate motifs.
Methods for identifying motifs fall into two primary categories: those relying on techniques of molecular biology and those employing methods of bioinformatics. Popular technique of molecular biology include Dnasefootprinting, gel mobility shift methods, SELEX, CHIP, among others [30]. While these methods can effectively reveal the structure of motifs, they often entail extended timeframes and high costs. Gene chip technology has emerged to address the need for extensive approaches, playing an important function in biology techniques. Protein immunop recipitation (ChIP) has also been employed to capture numerous fragments of DNA bound to specific TF. Moreover, sequence technologies appeared in second-generation used in tests, resulting in technologies such as ChIP-chip [31] and ChIP-seq [32].

Utilizing the algorithm of Expectation Maximization (EM) [33], MEME performs an iteration of EM on each candidate sequence once. The motif with the highest likelihood is selected for iteration until the convergence criteria are satisfied. This method is iterated until optimization is complete. However, it is important to note that such an algorithm might lead to local optima instead of reaching the globally optimal solution, especially when dealing with identification of short motif.

The work referenced in [34] introduced a genetic algorithm for motif identification that relies on method of Gibbs sampling to form the matrix of weights. Based on scoring method, a scoring function is used to modify the population during iterations. The consideration of motif count in the sequence encompasses instances with no motif or multiple motifs.

In reference [35], researchers introduced a novel algorithm for motif discovery in biological data by leveraging pushdown automata representation and employing mining tree techniques. The approach involves several key steps:

- The algorithm begins by defining a grammar that facilitates the creation of a pushdown automata notation suitable for biological data, encompassing DNA, RNA, and proteins.
- In the next phase, a set of motif sequences transfer from a biological bank undergoes transformation into a collection of pushdown automata, following the grammar established in the initial step.

- In the third phase, we generate a corresponding pushdown automaton for a given sequence, which serves as the input data for analysis. The final stage involves applying graph mining techniques, such as extracting frequent sub-trees or utilizing AprioriGraph, to derive sub-trees from the pushdown automata created in step three. Subsequently, all resultant trees are compared with a set of trees generated in step two. This comparison aims to identify and emphasize the motifs specific to the provided input data.

In reference [36], an algorithm named APMS, designed as an approximate qPMS solution, is introduced to efficiently handle extensive DNA datasets. This is achieved primarily through the acceleration search of neighboring substring and the elimination of redundant substrings. The experimental findings demonstrate that APMS excels not only in identifying implanted (l, d) motifs but also in executing processes orders of magnitude faster compared to the latest state-of-the-art qPMS algorithms.
The process and fundamental concepts of APMS can be outlined as follows:

- Identify high-frequency substrings of a specified length, k, from the dataset and store them in a set A. This step serves as preparation for generating seeds.
- Create a seed, m', using the k-mer x, which has the highest frequency in A.
- Refine the seed, m', generated by x. The refinement involves finding motif m by searching for d-neighbors of m'.
- If the refinement of m' yields a (l, d) motif, m, then filter out redundant k-mers in A using m.
- If set A is not empty, return to step b); otherwise, output the obtained (l, d) motifs in descending order based on their consensus sequence score.

In reference [37],**t**he researchers introduced a novel algorithm called MCES for discovering planted (l, d) motifs. This algorithm identifies motifs by extracting and merging emerging substrings.
Specifically, for the management of larger datasets, we have devised a MapReduce-based approach to distributedly extract emerging substrings. Results from experiments conducted on simulated data reveal that:

i) MCES efficiently and effectively identifies (l, d) motifs in input sequences ranging from thousands to millions, surpassing the speed of leading (l, d) motif discovery algorithms like F-motif and TraverStringsR;
ii) MCES demonstrates the capability to identify motifs of unknown lengths with higher accuracy compared to the rival algorithm CisFinder. Additionally, the validity of MCES is verified on real datasets.

In reference [38], the iGibbs algorithm employs a methodology where it takes a fasta or gbk DNA file as input and generates a nucleotide list to predict a starting position through random sampling. It incorporates motif length, a lower iterative value, and calculates probability and position ranking scores using the Position Weight Matrix (PWM). Implementation of the algorithm is carried out using Python, Python(x,y), and Biopython. The evaluation of iGibbs involves testing with varying motif lengths (12, 18, and 24) and different base lengths (5,000, 10,000, and 15,000) at various iteration levels. Results indicate that iGibbs exhibits superior average runtimes of 7, 10, and 23 seconds for motif lengths 12, 18, and 24, respectively, compared to 12, 32, and 60 seconds in the existing Gibbs sampling algorithm available at http://ccmbweb.ccv.brown.edu/gibbs/gibbs.html. To verify the accuracy of the motif results, the hamming distance is employed for identifying contiguous strings, and the minimum edit distance is utilized for consensus sequence comparisons.

Zainab Muhammed Jameel proposed in her thesis [39] a new technique to discover simple motifs (SM) and compound motifs (CM) hidden in DNA databases using finite states automata (FSA).

In reference [40], Hasnaa Imad Al-Shaikhli developed the Strong Motif Finder (SMF) and Quorum Strong Motif Finder (qSMF) algorithms for planted motif discovery in DNA. The main goal of her work with both algorithms is to reduce the execution time while achieving equal or higher prediction accuracy.

Huo and colleagues in [41] introduced the GARP algorithm, which enhances genetic algorithms (GA) to detect the planted motif by utilizing the random projection strategy (RPS). The rationale for employing RPS prior to GA is to locate favorable initial points that can serve as the initial population for the genetic algorithm, rather than relying on random selection.

In reference [42], Paul et al introduced method for the Planted Motif Problem (PMP), employing a population clustering technique to identify multiple and subtle motifs. Initially, the population is initialized by randomly choosing segments of the specific length to create a candidate consensus motif. After that, every sequence in dataset is checked to identify equivalent substrings, which are subsequently sorted based on the number of mismatches from the candidate motif. Following this, these substrings are evaluated depending on the scoring function. At the

end, the method computed the fitness of an individual (Cluster) to choose parents for use in Genetic Algorithms (GA).

3- Motif Mining and deep learning

In the past few years, deep learning has demonstrated significant success across diverse applications, prompting researchers to explore its application in the mining of DNA or RNA motifs. Motif mining within deep learning primarily encompasses three main frameworks: CNN-based models, RNN-based models, and hybrid CNN–RNN-based models.

DeepSEA [43] employed a convolutional neural network (CNN) with three layers, where the first one is 320 kernels, the second is 480 kernels, and third is 960 kernels. Convolutional layers (top layer) had a broader spatial range for input, while the lower-level layers of the convolutional network could capture more intricate features. To comprehensively capture information from the entire sequence data, DeepSEA introduced a specific network layer which its position on up of third convolutional layer, this called (FCN) fully connected network. This FCN layer ensured that all nodes get input from every output of preceding layer.

DeepSNR [44] utilized CNN-based deep learning approach. The convolutional segment of the DeepSNR model shared a similar structure with the DeepBind network. However, DeepSNR introduced a distinctive feature, such as the deconvolution network mirroring the convolution network, aiming to diminish the activation size and amplify activations by employing a combination of unpooling and deconvolution operations.

Dilated [45] employed the approaches of deep learning by utilizing multilayer CNN. Dilated method emphasized the acquisition of a mapping, tracing the DNA region from the sequence of nucleotide to pinpoint the location of the regulatory marker within that space. Through This approach could achieve a hierarchical depiction of the input area, surpassing the capacity of standard convolution. This enabled scaling to larger sequences both before and after the application of the convolutional operation.

DanQ [46] implemented a sequential approach consisting of a single-layer CNN followed by a bidirectional LSTM (BLSTM). The initial layer in this method and by using the convolution filteringwas designed to identify the location of motif in the sequence. In contrast to DeepSEA, the convolution step in the Danlayer to grasp the motif. Subsequent to the largest pooling layer, the model incorporated a BLSTM layer.

KEGRU [47] employed a model with incorporating k-mer and a single layer of GRU, instead of incorporating a CNN layer, KEGRU relied predominantly on the k-mer and incorporating layers to fulfill the retrieval the features of CNN present in alternative methods. This structural choice enabled KEGRU to excel in handling sequence relationships, ultimately proving effective in RNA motif mining with its adept performance.

iDeeps [48] implemented a combination of CNNs with a bidirectional long short-term memory (BLSTM) network to concurrently discern binding sequence and structure motifs from RNA sequences. The CNN module within iDeeps possessed the ability to catch the understandable motif of RNA-binding proteins (RBPs). The inclusion of a BLSTM network in the iDeeps framework not only enhanced performance in identifying binding sequences but also facilitated the straightforward capture of structural motifs.

Table (1) illustrates some types of planted motif search algorithms (rows) depending on various factors (columns) that effect on the time and memory obtained. These factors encompass searching type, quorum

or not, static or structure motif, data structure used, single core or multicore machine, maximum values of l, d, t, n and finally the time taken for these parameters.

Table(1): Some types of planted motif search algorithms

Algorithm Name	Exhaustive Search (Exactly Results)	Heuristic Search (Abbroxim-atly Results	Qurom Planted Motif	Simple Motif Type	Structured Motif Type	Data Structure or Technique used	Single core machine	Multi_core machine Or Parallel system	Maximum value for l, dand t , n	Time taken
PMSi	Yes	-	-	Yes	-	Tree search	Yes	-	(13, 4) t=20 n=600	18 min
PMSP	Yes	-	-	Yes	-	Tree search	Yes	-	(17,6) t=20 n=600	12 hour
Voting	Yes	-	-	Yes	-	Hash table	Yes	-	(15,5) t=20 n=600	22 min
PMSprune	Yes	-	-	Yes	-	Tree search	Yes	-	(19,7) t=20 n=600	10 hour
Stemming	Yes	-	-	Yes	-	Neighborhood generation	Yes	-	(15, 5) t=20 n=600	12.31 sec
PMS4	Yes	-	-	Yes	-	Tree Search + Lookup table	Yes	-	(23,9) t=20 n=600	54 hour
PMS5	Yes	-	-	Yes	-	Tree Search + Lookup table	Yes	-	(23,9) t=20 n=600	54 hour
Pair Motif	Yes	-	-	Yes	-	Branch and bound	-	Yes	(27,9) t=20 n=600	10 hour
PMS6	Yes	-	-	Yes	-	Tree Search + Lookup table+hashing	Yes	-	(23,9) t=20 n=600	19.19 hour
qPMS7	Yes	-	Yes	Yes	-	Tree Search	Yes	-	(21,7) t=q=20 n=600	11.6 hour
PMS8	Yes	-	-	Yes	-	Tree Search	Yes	-	(25,10) t=6 n=6000	15.45 hour
qPMS9	-	Yes	Yes	Yes	-	Tree Search	-	Yes (48 core)	(30, 13) t=q=20 n=600	51.02 hour
	-	Yes	Yes	Yes	-	Tree Search	Yes	-	(25,10)) t=q=20 n=600	6.3 hour
qPMS10	-	Yes	Yes	Yes	-	Tree Search	-	Yes	(23, 9) t=1000 n=600 q=100	2.49 hour
Apriori-Motif	Yes	-	-	Yes	-	Index structure + Suffix tree	Yes	-	(15,4) t=20 n=600	3.134 hour
Ex-motif	Yes	Yes	Yes	Yes	Yes	Inverted index and Hash function	Yes	-	-	-
ACME	Yes	-	-	Yes	-	Suffix tree	-	Yes (160	(15, 3)	580

								core)		sec
Weeder	Yes	-	Yes	Yes	-	Suffix tree	Yes	-	-	-
Fmotif	Yes	-	Yes	Yes	-	Suffix tree	Yes	-	(15,5) t=20 n=600	26.4 hour
Random Projection	-	Yes	-	Yes	-	Hashing	-	Yes	(19,6) t=20 n=600	0.96 sec
MEME	-	Yes	-	Yes	-	Expectation Maximization	Yes	-	t=20 n=600	7.6 sec
APMS	-	Yes	Yes	Yes	-	Acceleration of neighboring substring search	Yes	-	(21,8) t=3000 n=200 q=0.5%	28.9 sec
MCES	Yes	-	-	Yes	-	MapReduce-based approach + Tree search	-	Yes	(25, 10) t=3000 n=200	22 sec
iGibbs sampling	-	Yes	Yes	Yes	-	Gibbs sampling + Markov chain monte carlo (MCMC)	Yes	-	(24, 0) t=1 n=15000	23 sec
SMF and qSMF	-	Yes	Yes	Yes	-	Two Dimensional Array	Yes	-	(30,12) t=20 n=600	-

4- Discussion

The identification of motifs within biological sequences remains a crucial challenge in bioinformatics, with significant implications for understanding genetic regulation, disease predisposition, and personalized medicine. Throughout this review, we have examined a variety of algorithms designed to tackle the Planted Motif Problem (PMP), highlighting their respective methodologies, strengths, and limitations.

Early algorithms such as PMS1 laid the groundwork for motif discovery but faced challenges with efficiency and scalability. Subsequent advancements, including PMSP and PMSprune, introduced more space-efficient approaches and dynamic pruning techniques, significantly improving performance.

Profile-based and pattern-based algorithms represent two primary approaches to motif identification. While profile-based algorithms predict motif initiation sites, pattern-based algorithms aim to discover motifs directly. Exact algorithms, like Moby Dick and YMF, guarantee optimal solutions through exhaustive searches but are often impractical for large datasets due to their high time complexity. Approximation algorithms, though faster, may not always yield accurate results.

Recent developments have focused on reducing computational complexity and enhancing scalability. Algorithms such as PMS4, PMS5, and PMS6 introduced innovative preprocessing steps and neighborhood generation techniques, while PMS9 and qPMS10 leveraged parallel processing to handle larger datasets effectively. The introduction of ACME showcased the potential for scalable parallel systems to manage gigabyte-scale sequences, marking a significant leap forward in motif mining capabilities.

Deep learning approaches have also begun to show promise in motif discovery. Methods like DeepSEA and DeepSNR employ convolutional neural networks (CNNs) to capture intricate features within sequence data, offering new avenues for motif mining. These models can process extensive datasets with high accuracy, making them suitable for modern biological research where data volume is continually increasing.

Despite these advancements, challenges remain. The inherent complexity of biological sequences, the vast amount of data, and the need for efficient and accurate algorithms continue to drive research in this field. The integration of deep learning techniques with traditional bioinformatics methods offers a promising

direction for future studies, potentially overcoming current limitations and uncovering new biological insights.

5- Conclusion and future work

Motif identification in biological sequences is a critical aspect of bioinformatics, with applications ranging from disease diagnostics to personalized medicine. Over the years, numerous algorithms have been developed to address the Planted Motif Problem (PMP), each contributing to the incremental advancement of the field.

This review highlights the evolution of motif discovery algorithms, from early exact methods like PMS1 to advanced approximation techniques such as qPMS9 and deep learning models like DeepSEA. The continuous improvement in space efficiency, processing speed, and scalability underscores the dynamic nature of this research area.

The transition from traditional bioinformatics approaches to deep learning signifies a paradigm shift, leveraging the power of neural networks to handle the complexity and volume of modern biological data. As deep learning models continue to evolve, they are expected to play an increasingly important role in motif discovery, offering unprecedented accuracy and efficiency.

Future research should focus on integrating deep learning with existing bioinformatics methods, exploring hybrid models that combine the strengths of both approaches. Additionally, addressing the challenges of data heterogeneity, algorithm scalability, and interpretability will be crucial for further advancements.

In conclusion, while significant progress has been made, the quest for efficient and accurate motif discovery algorithms continues. The synergy between bioinformatics and deep learning holds great promise for the future, potentially leading to groundbreaking discoveries and innovations in understanding the complexities of genetic regulation and disease mechanisms.

References

[1] H. Leung and F. Chin, **"Algorithms for Challenging motif problems"**, *JBCB*, 43—58, (2005).

[2] Pankaj Agarwal, Sanjeev K. Yadav & A.P. Shukla, **"Solving Planted-Motif Problem – A New Approach"**, Department of Computer Science & Engineering, Krishna Institute of Engineering & Technology, Ghaziabad, U.P,15th International Conference on Advanced Computing and Communications, (2007).

[3] Ali Basim Yousif. **"Simple and Structured Motif Mining in DNA, RNA, and Protein Data Using Knowledge Discovery Techniques"**. Doctoral dissertation at University of Al-Mustansiriya, (2023).

[4] Atheel Sabih Shaker,Saadaldeen Rashid Ahmed, "**Information Retrieval for Cancer Cell Detection Based on Advanced Machine Learning Techniques",** Al-Mustansiriyah Journal of Science, Volume 33, Issue 3, (2022).

[5] Stormo, G, **"DNA binding sites: representation and discovery"**. Bioinformatics, 16:16{23, (2000).

[6] Du, Y.H.,Wang, Z.Z.: **"Review on computational prediction of transcription factor blinding sites"**. Life Sci. Res. 10(2), 24–31 (2006)

[7] Li, T.T., Jiang, B.,Wang, X.W.: **"Tutorial for computational analysis of transcription factor binding sites"**. ActaBiophys. Sin. 24(5), 334– 347 (2008)

[8] Brazma, A., Jonassen, I., Eidhammer, I., Gilbert, D.: **"Approaches to the automatic discovery of patterns in biosequences"**. J. Comput.Biol. 5, 279–305 (1998)

[9] Bussemaker, H.J., Li, H., Siggia, E.D.: **"Building a dictionary for genomes: identification of presumptive regulatory sites by statistical analysis"**. Proc. Natl. Acad. Sci. USA 97(18), 10096–10100 (2000).

[10] Sinha, S., Tompa, M.: **"Discovery of novel transcription factor binding sites by statistical overrepresentation"**. Nucleic Acids Res.30(24), 5549–5560 (2002)

[11] Sinha, S., Tompa, M.: **"YMF: a program for discovery of novel transcription factor binding sites by statistical over representation"**. Nucleic Acids Res. 31(13), 3586–3588 (2003)

[12] Jaime Davila, SudhaBalla, and Sanguthevar Rajasekaran,**"Space and Time Efficient Algorithms for Planted Motif Search"**,V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 822–829, 2006._c Springer-Verlag Berlin Heidelberg (2006).

[13] Sanguthevar Rajasekaran, Sudipta Pathak, "**An experimental comparison of PMSprune and other algorithms for motif search**",Int. J. Bioinformatics Research and Applications, Vol. 10, No. 6, (2014).

[14] P.P. Kuksa, and V. Pavlovic, **"Efficient Motif Finding Algorithms for Large-Alphabet Inputs"**, BMC Bioinformatics,http://www.biomedcentral.com/1471-2105/11/S8/S1, vol. 11, no. 8, article S1, (2010).

[15] S. Rajasekaran, and H. Dinh, **"A Speedup Technique for (l, d)Motif Finding Algorithms,"** BMC Research Notes, https://doi.org/10.1186/1756-0500-4-54, vol. 4, no. 54, (2011).

[16] H. Dinh, S. Rajasekaran, V.K. Kundeti, **"PMS5: an efficient exact algorithm for the (l, d)-motif finding problem",** BMC Bioinformatics, doi: 10.1186/1471-2105-12-410. vol. 12, no. 410, (2011).

[17] S. Bandyopadhyay, S. Sahni, S. Rajasekaran, **"PMS6: A faster algorithm for motif discovery"**, Proceedings of the second IEEE International Conference on Computational Advances in Bio and Medical Sciences, vol. 10, no. 4-5, pp 369-383, (2014).

[18] Q. Yu, H. Huo, Y. Zhang and H. Guo, **"Pair Motif: A New Pattern- Driven Algorithm for Planted (l, d) DNA Motif Search",** PLoS One, https://doi.org/10.1371/ journal.pone.0048442, vol. 7, no. 10:e48442, (2012).

[19] Z. Wei, and S.T. Jensen, **"GAME: detecting cis-regulatory elements using a genetic algorithm"**, Bioinformatics, vol. 22, no. 13, pp. 1577- 1584, (2006).

[20] M. Kaya, **"MOGAMOD: Multi-objective genetic algorithm for motif discovery"**, Expert Systems with Applications, vol. 36, no. 2, pp. 1039-1047, (2009).

[21] H. Huo, Z. Zhao,V. Stojkovic and L. Liu, **"Optimizing genetic algorithm for motif discovery"**. Mathematical and Computer Modeling, vol. 52, no. 11, pp. 2011-2020, (2010).

[22] Peng Xiao, Soumitra Pal, Sanguthevar Rajasekaran, **"qPMS10: A Randomized Algorithm for Efficiently Solving Quorum Planted Motif Search Problem",** IEEE International Conference on Bioinformatics and Biomedicine (BIBM), USA, (2016).

[23] CaiyanJia, Ruqian Lu, Lusheng Chen, **"A Frequent Pattern Mining Method for Finding Planted Motifs of Unknown Length in DNA Sequences"**, International Journal of Computational Intelligence Systems, Vol. 4, No. 5, 1032-1041, China, September, (2011).

[24] Yongqiang Zhang, Mohammed J. Zaki, **"exMotif: Efficient Structured Motif Extraction"**, Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180, USA.

[25] Majed Sahli, Essam Mansour, PanosKalnis, "**ACME: A scalable parallel system for extracting frequent patterns from a very long sequence ",** The VLDB Journal 23:871–893, © Springer-Verlag Berlin Heidelberg (2014).

[26] Zhang Y, Wang P, Yan M. **"An entropy-based position projection algorithm for motif discovery"**. BioMed Res Int 2016; (2016).

[27] 11. Sharov AA, Ko SH. **"Exhaustive search for over-re-presented DNA sequence motifs with CisFinder"**. DNA Res (2009);16(5):261-273.

[28] Kilpatrick AM, Ward B, Aitken S. **"Stochastic EM-based TFBS motif discovery with MITSU"**. Bioinformatics (2014); 30(12):i310–i318.

[29] Siebert M, Söding J. **"Bayesian Markov models consis-tently outperform PWMs at predicting motifs in nucleo-tide sequences"**. Nucleic Acids Res (2016); 44(13):6055-6069.

[30] Latchman, D.S.: **"Transcription Factors: A Practical Approach"**. Oxford University Press, Oxford (1993).

[31] Wu, B., et al.: "**Identify target genes involved in transcription factor GCF2 that promotes cell migration in tumor cell BEL**"-7404.Genomics Appl. Biol. 34(1), 35–40 .(2015).

[32] Haruka, O., Wataru, I.: MOCCS**: "Clarifying DNA-binding motif ambiguity using ChIP-Seq data"**. Comput. Biol. Chem. 63, 62–72. (2016).

[33] Tamura,K., Peterson, D., Peterson, N., Stecher,G., Nei, M.,Kumar, S.: **"MEGA5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods"**. Mol. Biol. Evol. 28, 2731–2739 (2011)

[34] Xiaochun Sheng, Kefeng Wang, **"Motif identification method based on Gibbs sampling and genetic algorithm"**,ClusterComput DOI 10.1007/s10586-016-0699-x,© Springer Science+Business Media New York (2016).

[35] Lounnas Bilal, Bouderah Brahim, Moussaoui Abdelouahab, **"Biological Motif Discovery Algorithm based on Mining Tree Structure "**, International Journal of Computer Applications (0975 – 8887) Volume 69– No.4, May (2013).

[36] QIANG YU AND XIAO ZHANG, **"A New Efficient Algorithm for Quorum Planted Motif Search on Large DNA Datasets"**, IEEE Access, Digital Object Identifier 10.1109/ACCESS.(2019). 2940115

[37] Qiang Yu, HongweiHuo*, Xiaoyang Chen, HaitaoGuo, Jeffrey Scott Vitter and Jun Huan, **"An Efficient Motif Finding Algorithm for Large DNA Data Sets"**, IEEE International Conference on Bioinformatics and Biomedicine (2014).

[38] Makolo AU and Lamidi UA, **"Motif Discovery in DNA Sequences Using an Improved Gibbs (i Gibbs) Sampling Algorithm"**, Journal of Computer Science & Systems Biology, Nigeria (2018).

[39] Zainab Muhammed Jameel, **"DNA Motif Mining Using Finite State Automata"**, Iraqi Commission for Computers and Informatics Institute for Postgraduate Studies, pp.4,30-31, (2017).

[40] Hasnaa Imad Al-Shaikhli,**"Approximate Algorithms for Regulatory Motif Discovery in DNA"**, Doctoral Dissertation Western Michigan University, USA, (2019).

[41] Fan Y, Wu W, Liu R, Yang W. **"An iterative algorithm for motif discovery"**. Procedia Computer Science (2013); 24:25-29.

[42] Wang X, Song T, Wang Z, Su Y, Liu X. MRPGA: **"motif detecting by modified random projection strategy and genetic algorithm".** J Computational Theoretical Nano-science (2013); 10(5):1209-1214.

[43] Zhou J, Troyanskaya OG, **"Predicting effects of noncoding variants with deep learning–based sequence model"**. Nat Methods (2015); 12:931–4.

[44] Salekin S, Zhang JM, Huang Y. **"A deep learning model for predicting transcription factor binding location at single nucleotide resolution"**. In: 2017 IEEE EMBS International Conference on Biomedical & Health Informatics. (2017), pp. 57-60.

[45] Gupta A, Rush AM. Dilated convolutions for modeling long distance genomic dependencies. arXiv:1710.01278. (2017).

[46] Quang D, XieX.DanQ: **"a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences**". Nucleic Acids Res (2016); 44:e107–7.

[47] Shen Z, Bao W, Huang D-S. **"Recurrent neural network for predicting transcription factor binding sites"**. Sci Rep (2018); 8:1–10.

[48] Pan X, Rijnbeek P, Yan J, et al. **"Prediction of RNA protein sequence and structure binding preferences using deep convolutional and recurrent neural networks"**. BMC Genomics (2018); 19:511.