An Efficient Metaheuristic Approach for Finding Motifs from DNA Sequences

Syed Md. Shamsul Alam*, Ibna Kowser*, Md. Al-Junaed Islam*,
Shurid Shahriar Zaman*, Tahseen Tayeb Kabir* and Faisal Bin Ashraf*
* Department of Computer Science and Engineering,
Brac University, Dhaka, Bangladesh

Abstract—**Finding patterns of the short sequences in DNA, RNA protein sequence has immense biological significance. The characterization and recognition of motifs is therefore an important method for a more in-depth understanding of genes or proteins in their structure, function and relations of evolution. This is one of the classical problems in the field of computational biology and which is an NP Hard problem. In this paper, we have proposed an evolutionary approach to get the motifs from DNA sequence by searching candidate motifs using heuristic way from the data. We have included various mutation techniques in an evolutionary approach and found an efficient way to calculate the fitness of our candidate motifs. We have evaluated the fitness of found motifs from our approach with benchmark data sets. Our method performs better results in terms of accuracy and specificity.**

I. Introduction

Deoxyribonucleic acid, or DNA, is a kind of molecule found inside the nucleus of a cell. In the human body, DNA serves as the basic biological function. The major role is to store and code the genetic information of a body. DNA is a polymer made up of nucleotides. Phosphate, sugar, and nitrogenous base are three crucial elements in the nucleotide's fundamental structure. Adenine, Guanine, Cytosine, and Thymine are the four kinds of nucleotides found in DNA. Pyrimidines and purines are the two types of nucleotides that make up DNA. Purines include Adenine and Guanine, whereas pyrimidines include Thymine and Cytosine. [1].

DNA motif is a sequence pattern of nucleic acid sequences that include regulatory proteins of DNA bonding sites, also known as Transcription Factor (TF). Where proteins may be present, DNA motifs combine with structural motifs. Though motifs are found on double-stranded DNA, TF also binds to double-stranded DNA. Motif discovery is defined as the finding of motifs without prior knowledge of the patterns' appearance. A single strand of double helix DNA may be represented as a string across the letters F = A, T, G, C. The emergence of motif discovery occurs while DNA includes binding motifs with unknown patterns. Let us see a short example on finding motif from a DNA sequence.

AGGTACA**CTCA**TGATGCACCTGTA
CTTGATTCACATGA**CTCA**TGACAT
CCGTAACTGCTTGCA**CTCA**AACAT
TGTTAGGA**CTCA**TCACACGACAAT
GAGT**CTCA**CTGATCTGAGTCAGAA

There are five distinct DNA sequences in the example above. We acquire a common string that is repeated in every sequence by iterating them, which is CTCA. As a result, CTCA is one of the motifs in the DNA sequences above.

II. Related Works

The discovery of these factors is a crucial task for molecular biology. The motif finder, which includes the class MotifFinder and the findMotif method, as well as four separate motif finding algorithms, two heuristic PROJECTION or ePattern-Branching algorithms, and two similar Algorithms PMS1 and PMSP, provides in-depth knowledge of the structure, role, and developmental connections of genes or proteins. SMS, PMS, and EMS are the three primary paradigms for identifying tiny, functional patterns of peptides, transcriptional regulatory components, composite regulatory patterns, DNA diagrams, and distinctions across protein families, among other things. Most previous literatures however, classified algorithms of motif searching in two key groups according to the combinatorial methodology employed in their design:

1. Word-based methods (string-oriented) mostly focused on full description, i.e. oligonucleotide frequency count and contrast, and
2. Probabilistic models with estimates of the function parameters by maximum probability.

One of the motif-finding algorithms is Oligo-Analysis, which is based on van Helden's word-based approaches [2]. Despite being theoretically simple, their algorithms have been successful in deleting motifs from the majority of yeast regulatory families (Saccharomyces cerevisiae). Sinha and Tompa [3] developed a YMF algorithm based on the same methodology. The notion was derived from an examination of the transcription factor's yeast binding locations. The concept was extracted from an analysis of the identified yeast binding sites for the transcription factor. These were based on word-based method. An approximate algorithm has been proposed with bucketing technique by projecting at random positions in another work [4]. A threshold value was used to select the buckets of motifs. Hertz's [5] first attempts to find a matrix describing transcription factor, Lawrence and Reilly implemented the EM [6] for motif searching and Gibbs sampling approach [7] were probabilistic based method.

Nature-inspired algorithms, such as ANN, FS, and SI [8] have been utilized as model models for a variety of real-world

978-1-6654-0906-3/21/$31.00 ©2021 IEEE

issues. The heuristic approach [9] is a method of prioritizing the pathways from an algorithm's beginning state to its goal state or end state above alternative paths in that process. This is used to discover a solution to a problem, and the solution is computed in the last state, also known as the target state. A heuristic approach is a method for discovering a problem's solution. It's a faster approach to getting decent enough results. Knowledge is a secondary consideration in this strategy. This method is a logical system that does not have to be exact or ideal, and it is a flexible technique for making rapid judgments, especially when dealing with complex data. The heuristic technique gives a rapid, simple, and easy-to-implement answer. Because the Heuristic method is practical, it may be used to provide rapid and practical short-term solutions to scheduling and planning problems. It can lead to deeper analysis of possible concerns through ease of use testing. This method may be used in conjunction with other user-friendly testing philosophies. Assigning the appropriate Heuristic method can assist in recommending the most effective remedial procedures [10].

Gonzalez et al proposed MOABC/DE [11], which is meant to adapt the Artificial Bee Colony [12] algorithm to a multi-objective context. The MACS [13] algorithm improves the fundamental Cuckoo Search [14] algorithm by combining parallel, incentive, information and adaptive strategies. FMGA [15], [16] algorithm is used for location motive position which was based on Genetic Algorithm [17]. Reddy et al developed PMbPSO [18] based on PSO [19] to find motifs.

III. Proposed Method

A. Data Representation

In our proposed method, we are trying to calculate the positions of each nucleotide in a sequence of data. We can easily do it by building a matrix. In a matrix, there are 4 rows which are four nucleotides and the columns are the representation of the given sequence. We are using binary formula and if the index of row and column matches then that [row, columns]'s value will be 1 otherwise 0. In this way, we get the position of each nucleotide and this will help us to do our future work like mutation and crossover. Also, we are trying to represent our candidate motif in the same way. We are doing it to get more motifs easily after doing the crossver and mutation. As an example, we take 4 subsequences of candidate motif which length is 6 – ATCGGA, TGCTAT, AGTTAG, and CTGCTG. Table I represents the matrix form of subsequences of candidate motif.

TABLE I
Candidate Motif Representation

	0	1	**2**	**3**	**4**	**5**
A	0.5	0	0	0	0.5	0.25
C	0.25	0	0.5	0.25	0	0
T	0.25	0.5	0.25	0.5	0.25	0.25
G	0	0.5	0.25	0.25	0.25	0.5

B. Metaheuristc Method

We are using heuristic approaches, especially genetic algorithms for finding motifs. For finding the best motif we take initial population from the dataset, find the fitness score of that population, access population for finding the best population and after doing crossover and mutation we get the final output. We will update the fitness score after getting the better fitness value. We set a limit to run the iteration to get the best motif in the shortest possible time. Fig 1 shows the flowchart of our proposed method.

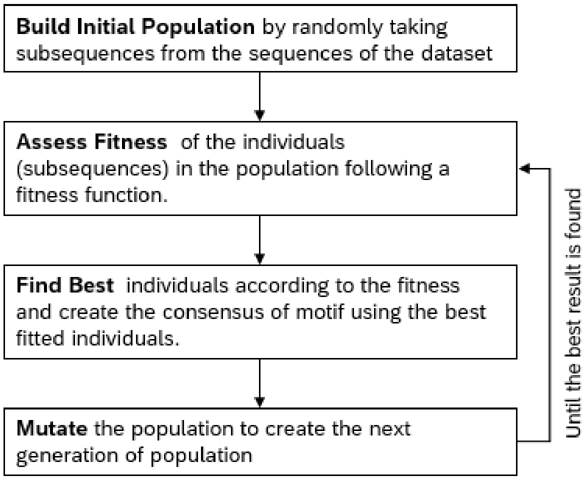

Build Initial Population by randomly taking subsequences from the sequences of the dataset
Assess Fitness of the individuals (subsequences) in the population following a fitness function.
Find Best individuals according to the fitness and create the consensus of motif using the best fitted individuals.
Mutate the population to create the next generation of population
Until the best result is found

Fig. 1. Pipeline of Proposed Method

1) Build Initial Population: The working principle of this step is to generate the initial population from a given sequence. In datasets there will be n number of DNA sequences and this method will find the l length of random sub-sequences. These sub- sequences will fill up the initial population. As the subsequences of the initial population are already in datasets, they can be candidate for optimal solution.

2) Fitness Function: In this step, our main target is to get less number of mismatches. To do this, we have to run two fitness steps. One step is running to get the fitness in sequence and another one is running to get fitness in the whole dataset. Firstly, to get fitness in a sequence we will use Equation 1.

$$D(x, S) = Min(\sum_{S_i \in S} d(x, S_i)) \tag{1}$$

Here, S_i is the i^{th} subsequence of equal length of x in sequence $S.d(x, S_i)$ calculates number of mismatch as distanc between the two subsequenes. Then, we get the fitness of this candidate motif x, with respect to the whole dataset using Equation 2.

$$Fitness(x) = Avg(D(x, S)), \forall_S \in Dataset \tag{2}$$

This fitness function will represent how much portion of the candidate motif is available in the dataset.

3) Assess Population: In this step, our main target is to get the best population from the selected candidate motifs (subsequences). Firstly, we take some subsequences from the population. Then, we calculate the fitness of these subsequences by using the Fitness formula. Now we have fitness values of these subsequences and we store these values. Then, we sort these values in a descending order. Now, we know which subsequence has the better fitness value. To illustrate, we take top 10 subsequences which have the better fitness values and counting them as our best population. We will do this step for several times to get the best population repeatedly. It will help us to get the best candidate motif and help in the mutation step.

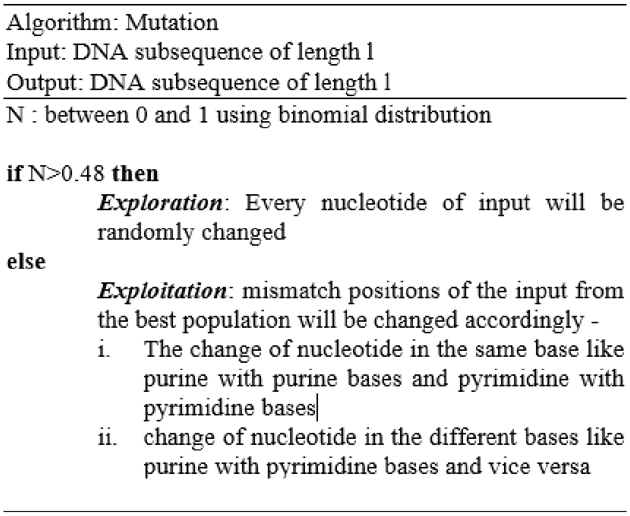
4) Consensus Sequence: In this step, we basically count the number of selective nucleotides in a fixed position. We take the best population as inputs which we find from the assess population and then get numerical numbers as outputs. We have to do the whole step for all four nucleotides. Then we divide the output with the number of subsequences which we take as inputs. In the output the number of columns has to be equal to the length of subsequences and the summation of each column has to be one.

As example, let our length of motif is 10 and we have 4 subsequences which are “CATGAGCTAC”, “ACACGTCGAT”, “TGCACAGATG”, “GTCGTTGACA” Then the output will look like table 7.

TABLE II
CONSENSUS SEQUENCE

A	.25	.25	.25	.25	.25	.25	0	0.5	0.5	.25
C	.25	.25	0.5	.25	.25	0	0.5	0	.25	.25
T	.25	.25	0	0.5	.25	.25	0.5	.25	0	.25
G	.25	.25	.25	0	.25	0.5	0	.25	.25	.25

5) Mutation: Mutation plays the crucial part to meander around the candidate motif space. In this step, we will retrieve the best nucleotide based on frequency at each position in the best population. After comparing the best nucleotide with the best population, we get the mismatch position and then we keep up the exchange off in two ways; exploration and exploitation. We have used binomial distribution for maintaining the exchange off between exploration and exploitation. The binomial distribution often produces lower values between 0 and 1 and seldom produces higher numbers. We’ve utilized this strategy to ensure that we’re always in the best portion of the solution space, and that we only sometimes take a large leap and move to a completely new region of the solution space to see if there’s another best answer. We tend to progress to a better solution when we stay in the current section of the solution space by simply replacing the nucleotides that are not most commonly present with the help of a list that stores the most frequently occurring nucleotides so far at different places. When we try to shift to a new section of the solution space, we end up changing nucleotides at all places at random, and we end up in a different portion of the candidate motif space. Fig 2 represents our mutation algorithm.



```
Algorithm: Mutation
Input: DNA subsequence of length l
Output: DNA subsequence of length l
N : between 0 and 1 using binomial distribution

if N>0.48 then
        Exploration: Every nucleotide of input will be
        randomly changed
else
        Exploitation: mismatch positions of the input from
        the best population will be changed accordingly -
        i.   The change of nucleotide in the same base like
             purine with purine bases and pyrimidine with
             pyrimidine bases
        ii.  change of nucleotide in the different bases like
             purine with pyrimidine bases and vice versa
```

Fig. 2. Mutation Algorithm

IV. Result Analysis

We have run our algorithm on the ‘hm05r’ dataset which contains 3 sequences with 3000 nucleotides to fi the motifs of diff t lengths. Fig 3 shows the found motif of length 10 from the human sequences in the data set. Table 10 contains the result of accuracy of motif from diff t length of ’hm05r’ dataset.

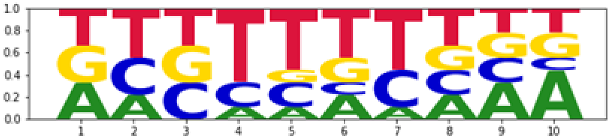


Fig. 3. Motif Logo of Length 10 for Dataset ”hm05r”

From table 8, we can see that for 8 length we get more than 90% accuracy. However, as the length gets longer, the accuracy of motif becomes lesser. Because, it is difficult to find motif of longer length which have zero mismatch with all the sequence of a dataset. After experimentation, all the found motif are available in our github repository (click here). In addition, we have shown the result in Fig 4.

TABLE III
MOTIF’S ACCURACY FROM ’HM05R’ DATASET

Length	8	13	15	23
Accuracy	95.83	82.05	80.00	73.91

From Fig 4, it is clear that we get higher accuracy for short sequences of motifs. On the other hand, for 13 and 15 lengths of motifs we get a very close range of accuracy. For 13 lengths of motifs, we get the best accuracy from the dm05r dataset. For 15 length of motifs, we also get the best accuracy from the dm05r dataset. For long lengths of motifs we get lower accuracy compared with the short length of motifs. We

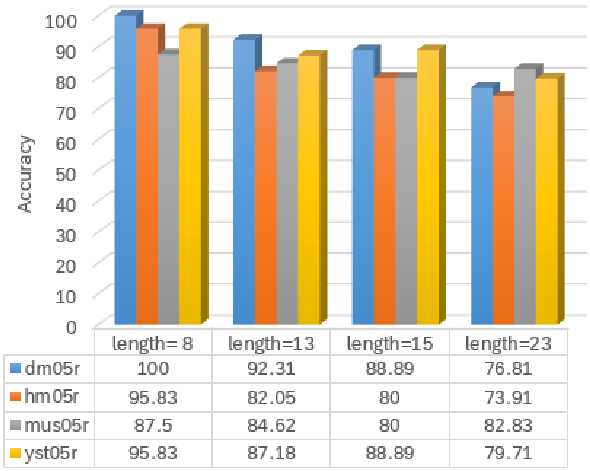


Fig. 4. Accuracy of the motifs with different lengths in different dataset

understand from our work that whenever the length of motif increases, the chance of finding subsequence of larger length which has the least number of mismatches decreases and for this reason, we get lower accuracy for the long length of motifs.

A. Comparison with existing methods

We have compared our proposed algorithm in various data set with other established motif finding algorithms by calculating specificity. Some established methods are available at ‘Assessment of Computational Motif Discovery Tools’ and we have compared our method with it. The result of AlignACE [20], ANN-spec [21], Consensus [22], GLAM [23], Improbizer [24], MEME [25], MEME3 [25], MFEA [26], MITRA [27], MotifSampler [28], oligo/dyad-analysis [29], QuickScore [30], SeSiMCMC [31], Weeder [32] and YMF [3] are contained by it.

Before comparing the results we have to define some definitions.

- True Positives (TP): Number of positions in familiar sites and anticipated sites.
- True Negatives (TN): Number of positions that are neither in familiar sites nor in anticipated sites.
- False Positives (FP): Number of positions in anticipated sites that are not present in familiar sites.
- False Negatives (FN): Number of positions in familiar sites that are not present in familiar sites.

It denotes how accurately the algorithm performed to get the actual motifs. We will compare performance of the algorithms based on Equation (3).

$$nSP = \frac{nTN}{(nTN + nFP)} \tag{3}$$

We have proposed an evolutionary method. The table shows that our pro- posed method performs better than most of the existing methods. To illustrate, the datasets ‘mus01r’, ‘hm02r’, ‘mus05r’, ‘yst04r’, ‘hm03r’, ‘mus02r’ etc. gives a better result than other methods. Besides this, our proposed model also works better for a longer number of sequences as well as a short number of sequences. Furthermore, our method also performs better for longer sequences and for short sequences.

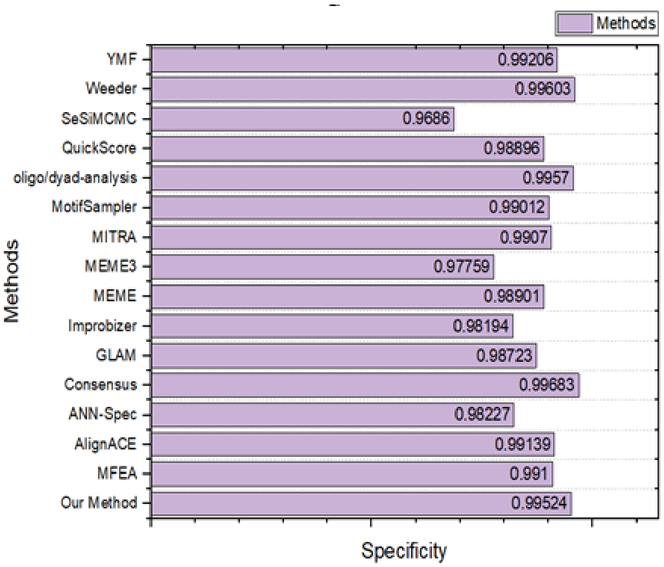


Fig. 5. Specificity of different algorithm

Figure 5 shows the specificity comparison between our proposed method with other existence methods on the basis of species like fly , mouse, human and yeast. In this graph, our algorithm gives best results on fly , human and yeast than other existing algorithms and for the yeast our method performs very close to the established algorithms. Fig 6 shows the comparison of overall specificity of our proposed method with other existing methods. Our algorithm can find longer motifs from the dataset.

V. Conclusion

To conclude, we have suggested an evolutionary process to find motifs in DNA sequence. We have generated a set of initial candidate motifs and sorted out the best candidates from all of these. Doing mutation in the best candidates appears to produce the data set’s optimum candidate motif. In numerous data sets, our method has performed well and fits the precision with accepted procedures that confirm our method’s usefulness. In addition, this approach means that a very large size of motifs can be detected, which is not easy to locate using any other exhaustive process since it requires even days to measure. As in every step of evolution, we are using heuristic and picking the best candidates, our approach faces no difficulty in discovering large length motifs. Nevertheless, there are some places where modification can be possible in our proposed algorithm. As the accuracy percentage decreases for higher length motifs, we can introduce some new steps to get better accuracy for higher length motifs. We can improve our mutation method to get motifs that are more accurate. Furthermore, we can work on time duration. We can reduce memory consumption. If we can do this modification then

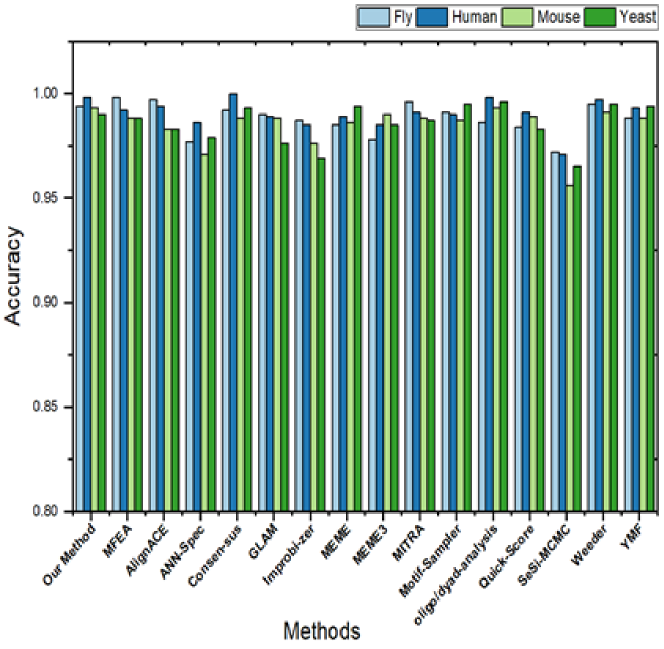


Fig. 6. Comparison with other algorithm

this approach can be efficient for finding best-fitted motifs of different length with higher accuracy in the near future.

References

[1] R. Rettner, "Dna: Definition, structure discovery," Live Science, 12 2017. [Online]. Available: https://www.livescience.com/37247-dna.html

[2] J. van Helden, B. André, and J. Collado-Vides, "A web site for the computational analysis of yeast regulatory sequences," *Yeast*, vol. 16, pp. 177–187, 01 2000.

[3] S. Sinha and M. Tompa, "Ymf: a program for discovery of novel transcription factor binding sites by statistical overrepresentation," *Nucleic Acids Research*, vol. 31, pp. 3586–3588, 07 2003.

[4] F. B. Ashraf, A. I. Abir, M. S. Salekin, and M. A. Mottalib, "Rppmd (randomly projected possible motif discovery): An efficient bucketing method for finding dna planted motif," in *2017 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 2017, pp. 509–513.

[5] G. Z. Hertz, G. W. Hartzell, and G. D. Stormo, "Identification of consensus patterns in unaligned dna sequences known to be functionally related," *Bioinformatics*, vol. 6, pp. 81–92, 1990.

[6] C. E. Lawrence and A. A. Reilly, "An expectation maximization (em) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins: Structure, Function, and Genetics*, vol. 7, pp. 41–51, 1990.

[7] M. Nicolae and S. Rajasekaran, "qpms9: An efficient algorithm for quorum planted motif search," *Scientific Reports*, vol. 5, p. 7813, 01 2015. [Online]. Available: https://www.nature.com/articles/srep07813

[8] W. Fang, X. Li, M. Zhang, and M. Hu, "Nature-inspired algorithms for real-world optimization problems," *Journal of Applied Mathematics*, vol. 2015, pp. 1–2, 2015.

[9] P. Paruchuri, J. P. Pearce, M. Tambe, F. Ordonez, and S. Kraus, "An efficient heuristic approach for security against multiple adversaries," *Proceedings of the 6th international joint conference on Autonomous agents and multiagent systems - AAMAS '07*, 2007.

[10] B. Frankovič and I. Budinská, "Advantages and disadvantages of heuristic and multi agents approaches to the solution of scheduling problem," *IFAC Proceedings Volumes*, vol. 33, pp. 367–372, 06 2000.

[11] A. Rubio-Largo, D. L. Gonzalez-Alvarez, M. A. Vega-Rodriguez, J. A. Gomez-Pulido, and J. M. Sanchez-Perez, "Mo-abc/de - multiobjective artificial bee colony with differential evolution for unconstrained multiobjective optimization," *2012 IEEE 13th International Symposium on Computational Intelligence and Informatics (CINTI)*, 11 2012.

[12] *AN IDEA BASED ON HONEY BEE SWARM FOR NUMERICAL OPTIMIZATION*, 2005.

[13] Y. Zhang, L. Wang, and Q. Wu, "Modified adaptive cuckoo search (macs) algorithm and formal description for global optimisation," *International Journal of Computer Applications in Technology*, vol. 44, p. 73, 2012.

[14] M. Kaya, "Mogamod: Multi-objective genetic algorithm for motif discovery," *Expert Systems with Applications*, vol. 36, pp. 1039–1047, 03 2009.

[15] F. A.Hashim, M. Mabrouk, and W. Al-Atabany, "Review of different sequence motif finding algorithms," ResearchGate, 04 2019. [Online]. Available: https://www.researchgate.net/publication/332890910_Review_of_Different_Sequence_Motif_Finding_Algorithms

[16] F. Liu, J. Tsai, R. Chen, S. Chen, and S. Shih, "Fmga: finding motifs by genetic algorithm," *Proceedings. Fourth IEEE Symposium on Bioinformatics and Bioengineering*.

[17] A. Makolo, "A comparative analysis of motif discovery algorithms," *Computational Biology and Bioinformatics*, vol. 4, p. 1, 2016.

[18] U. Reddy, M. Arock, and A. Reddy, "Planted (l, d) - motif finding using particle swarm optimization," *International Journal of Computer Applications*, vol. ecot, pp. 51–56, 12 2010.

[19] *Human promoter prediction based on sorted consensus sequence patterns by genetic algorithms*. Proceedings of the International Congress on Biological and Medical Engineering, 2002.

[20] X. Ma, A. Kulkarni, Z. Zhang, Z. Xuan, R. Serfling, and M. Q. Zhang, "A highly efficient and effective motif discovery method for chip-seq/chip-chip data using positional information," *Nucleic Acids Research*, vol. 40, pp. e50–e50, 01 2011.

[21] C. T. WORKMAN and G. D. STORMO, "Ann-spec: A method for discovering transcription factor binding sites with improved specificity," *Biocomputing 2000*, 12 1999.

[22] J. Buhler and M. Tompa, "Finding motifs using random projections," *Journal of Computational Biology*, vol. 9, pp. 225–242, 04 2002.

[23] M. C. Frith, "Finding functional sequence elements by multiple local alignment," *Nucleic Acids Research*, vol. 32, pp. 189–200, 01 2004.

[24] P. PAVLIDIS, T. S. FUREY, M. LIBERTO, D. HAUSSLER, and W. N. GRUNDY, "Promoter region-based classification of genes," *Biocomputing 2001*, 12 2000.

[25] T. L. Bailey, J. Johnson, C. E. Grant, and W. S. Noble, "The meme suite," *Nucleic Acids Research*, vol. 43, pp. W39–W49, 05 2015.

[26] F. B. Ashraf and M. S. R. Shafi, "Mfea: An evolutionary approach for motif finding in dna sequences," *Informatics in Medicine Unlocked*, vol. 21, p. 100466, 2020.

[27] E. Eskin and P. A. Pevzner, "Finding composite regulatory patterns in dna sequences," *Bioinformatics*, vol. 18, pp. S354–S363, 07 2002.

[28] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouze, and Y. Moreau, "A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling," *Bioinformatics*, vol. 17, pp. 1113–1122, 12 2001.

[29] M. Tompa, N. Li, T. Bailey, G. Church, B. De Moor, E. Eskin, A. Favorov, M. Frith, Y. Fu, W. Kent, V. Makeev, A. Mironov, W. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, and Z. Zhu, "Assessing computational tools for the discovery of transcription factor binding sites," *Nature biotechnology*, vol. 23, pp. 137–44, 02 2005.

[30] M. Régnier, "Mathematical tools for regulatory signals extraction," *Bioinformatics of Genome Regulation and Structure*, pp. 61–69, 2004.

[31] A. V. Favorov, M. S. Gelfand, A. V. Gerasimova, D. A. Ravcheev, A. A. Mironov, and V. J. Makeev, "A gibbs sampler for identification of symmetrically structured, spaced dna motifs with improved estimation of the signal length," *Bioinformatics*, vol. 21, pp. 2240–2245, 02 2005.

[32] G. Pavesi, P. Mereghetti, G. Mauri, and G. Pesole, "Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes," *Nucleic Acids Research*, vol. 32, pp. W199–W203, 07 2004.