Informatics in Medicine Unlocked 21 (2020) 100466

Contents lists available at ScienceDirect

Informatics in Medicine Unlocked

journal homepage: http://www.elsevier.com/locate/imu

MFEA: An evolutionary approach for motif finding in DNA sequences

Faisal Bin Ashraf [a,*], Md Shafiur Raihan Shafi [b]

[a] Department of Computer Science and Engineering, Brac University, Dhaka, 1212, Bangladesh
[b] Department of Computer Science and Engineering, Southeast University, Dhaka, 1213, Bangladesh

ARTICLE INFO

Keywords:
Bioinformatics
Computational biology
DNA
Motif search
Meta-heuristic
Evolutionary algorithm

ABSTRACT

Identification of short repeating patterns in biological sequences, mostly known as motif, is important for understanding the genetic regulatory system of a living being. But weak conservation of motifs makes it an NP-hard problem and poses a challenge in computational biology. In this work, we have modeled the motif search problem from meta-heuristic perspective. We have proposed and evaluated an evolutionary approach, in which, we will search candidate motifs with a heuristic so that we can find the real motifs of the data set without exploring rigorously. Our method minimizes the trade between exploration and exploitation of the search space with a defined mutation technique using normal distribution and finds an efficient way to measure the fitness of a candidate motif to be real motif. We have used benchmark data set to evaluate the fitness of found motifs for each species, and our approach gives accurate motifs for each of them.

1. Introduction

In biology, sequence motifs are short sequence patterns, usually with fi lengths, that represent many features of DNA, RNA, and protein molecules. Sequence motifs can represent transcription factor binding sites for DNA, splice junctions for RNA, binding domains for proteins and helps to fi similarities between diff t DNA samples by assisting in Sequence Alignment [1]. Discovering sequence motifs can lead to a better understanding of transcription regulation, mRNA splicing, and the formation of protein complexes. Further-more, protein motifs can represent the active sites of enzymes or regions involved in protein structure and stability.

A DNA molecule consists of two anti-parallel chains of four types of nucleotides forming a double helix - Adenine (A), Cytosine (C), Guanine (G) and Thymine (T) where a single strand of this double helix DNA can be expressed as a string over the alphabet), = {A, C, G, T}. Proteins, which are special sequence of nucleotides that bind to DNA in a specific manner is called Transcription Factor and this sequence of nucleotides are called Binding Motifs [2]. Motif Discovery Problem arises when DNA contains instances of binding motifs and they are unknown. Example of Binding Motif:

GGCTGCA**CACGT**GTATTGC … ACGATGTCTCGC

ATCGCAT**CACGT**GACCAGT … GACATGGACGGC

TCG**CACGT**GGTGGTACAGT … AACATGACTAAA

CGTTAGGACCAT**CACGT**GA … ACAATGAGAGCG

TAGCC**CACGT**GGATCTTGT … AGAATGGCCTAT

Here, "CACGT" is repeated in all the sequences. "CACGT" is one of the motif for these given DNA sequences. DNA motif fi g problem, from optimization perspective, can be described as follows-given a set of identical length DNA sequences S = {S_1, S_2, …, S_N } of the four characters in the alphabet ε = A, C, G, T find the promising motif pattern X = X_1X_2 …, X_l of length l, X_i ∈ {A, T, C, G}.

2. Literature review

Finding motifs in DNA sequences can be discovered from diff t perception. Many principles to fi the motifs in DNA sequences are found in the literature. They are mainly classified into three categories - Simple Motif Search (SMS) [3], Edit distance based Motif Search (EMS) [4] and Planted Motif Search (PMS) [5–7]. In SMS [3], all the motifs up to a given length are found out by conditioning that, all these motifs are present in all the DNA sequences. EMS [4] fi motifs of a fi length. But, these motifs do not need to be present in all the sequences rather than at least in a certain number of sequences. On the other hand, in PMS [5–7], motifs of a fi length are found out which are present in all the given sequences. If we further explore the algorithms, we will find two types of algorithms to fi the motifs of given sequences - exact algorithms and

* Corresponding author.
E-mail address: faisal.ashraf@bracu.ac.bd (F.B. Ashraf).

https://doi.org/10.1016/j.imu.2020.100466
Received 14 May 2020; Received in revised form 20 October 2020; Accepted 20 October 2020
Available online 29 October 2020

Table 1
Representation of DNA sequence.

	Nucleotides	A	A	T	C	G	T	T	C	G	A	A	C	G
Positions		**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
	A	1	1	0	0	0	0	0	0	0	1	1	0	0
	T	0	0	1	0	0	1	1	0	0	0	0	0	0
	C	0	0	0	1	0	0	0	1	0	0	0	1	0
	G	0	0	0	0	1	0	0	0	1	0	0	0	1

Table 2
Representation of a motif.

	0	1	2	3	4	5	6
A	0.25	0	0.5	1	0.5	0	0
T	0	0.25	0	0	0.25	1	0.5
C	0.75	0	0.5	0	0	0	0
G	0.25	0.75	0	0	0.25	0	0.5

approximate algorithms. Exact algorithms aims to fi out all the possible motifs of the data set by cost of more run time whereas approximate algorithms do not find all the possible motifs rather a portion of the all the motifs in favor of less run time. In this work, we aim to fi motifs from given DNA sequences of length l proposing an approximate algorithm using heuristic information so that we always tend to get the best result without exploiting unnecessarily around the solution space, i.e, candidate motif space.

For a given natural number l, there are mainly two problems corresponding to two approaches - Consensus Approach and Positional Approach. In, consensus approach we find a string S_c of length l and a set of subsequences $M = \{M_1, M_2, \ldots, M_N\}$, in which M_i is a substring of length l in S_i, such that they minimize the objective function [8]:

$$H(S_c, M) = \sum_{i=1}^{N} d_H(S_c, M_i) \quad (1)$$

where $d_H(S_c, M_i)$ is the hamming distance between string S_c and substring M_i defined by the number of positions at which the nucleotides in two string are different. In positional approach, we find a set of substrings $M = \{M_1, M_2, \ldots, M_N\}$ and a set of starting positions $A = \{a_1, a_2, \ldots; a_N\}$, in which, each instance M_i is of length l in S_i corresponding to starting position a_i. In this approach, the objective function is information content [9]:

$$IC = \sum_{j=1}^{l} \sum_{u \in \varepsilon} Q(u, j) \log_2 \left(Q(u, j)/P_u\right) \quad (2)$$

where Q (u, j) indicates the frequency of nucleotide u in column j of the matrix M, P_u is the background frequency of u in the entire set S. In reality, the location of mi on S_i is called the binding site of DNA.

Different approaches of finding motifs in DNA sequences have been introduced in literature. They can be classified into three section with respect to the type of each search technique - Enumerative Search, Deterministic Search and Stochastic Search. Enumerative search is usually used for consensus representations. Weeder [10] is a good example of enumerative search for motif finding. Deterministic approach rely on the Expectation Maximization (EM) algorithm to optimize a motif matrix. MEME [11] and CONSENSUS [12] use EM to optimize the matrix. Stochastic Methods iteratively align a set of TFBSs and generalize a motif matrix from the set. Methods such as BioProspector [13], AlignACE [14], and MotifSampler [15] that use Gibbs sampling [16] can be further categorized as single-point searches, whereas the recently developed evolutionary algorithms (EAs) [17] implemented by the GAME [18] and GALF [19] methods can be classified as population-based searches. In this work, we will focus on stochastic search using heuristic.

One solution technique to respond to this challenge is a swarm-based approach, a natural metaphor, called Ant Colony Optimization (ACO) [20–22]. ACO is a population-based stochastic search method inspired by the foraging behavior of ant colonies. This metaheuristic has been used successfully for computing the best-known solutions for a wide range of combinatorial optimization problems.

In [23] the authors have used ACO for finding motifs. They have modeled the Motif Finding problems into the problem of finding solutions on the structural graph G = (V,E, Ω, η, T), where V is the vertex set, E is the edge set, and η and T are the set of heuristics information and pheromone trail. An acceptable solution is a path satisfying condition Ω, starting from a vertex in a subset C_0 of V, then expanded by random to the next vertex based on heuristics information and pheromone trail.

In [24] ACO and EM both have been introduce for finding the motifs more accurately. They have built the iteration by randomly extracting subsequences from the input sequences and comparing to the input sequences to find a set of potential binding sites which minimizes the sum of hamming distances. Then computing the IC value and updating pheromone levels. EM algorithm finds the maximum likelihood estimate (MLE) of the unknown motif sites conditional to the observed data by iteratively applying - Calculate the expected value of the log likelihood function and Find the positions of motif instances that can maximize the log likelihood function.

In [25] a memetic algorithm is used to find the motifs which uses a modified version of the Greedy Randomized Adaptive Research Procedure (GRASP) to build an initial population of solutions and Variable Neighborhood Search (VNS) algorithm, that is a greedy local search method that explores the solution space through systematic exchanges of increasingly distant neighborhood structures.

In this work, we adopt the concept of consensus for measuring the accuracy of the candidate motifs and proposed an evolutionary algorithm which minimizes the unnecessary exploration in the candidate motif space and leads the searching to the more accurate zone in the solution space following some mutation techniques in the sub-sequences of the DNA sequences.

3. Proposed method

We propose an evolutionary approach to find motifs from a given data set. Our proposed approach involves building an initial population of candidate motifs which are individuals of the population and then measuring the fitness of all the individuals. Following, we will regenerate the population by keeping some best parents and applying crossover and mutation among them. In every iteration, we will update the best solution and we will keep the iteration until we get to start no better solution than the best for an amount of time.

3.1. Data representation

DNA sequences of A, T, C and G can be represented in a matrix having four rows and column of sequence length. This representation will help us to compute during mutation and crossover and reduce run time while comparing the sub sequence with our candidate motifs as we will perform matrix subtraction rather than comparing strings. For example, if we have a sequence of "AATCGTTCGAACG" then its matrix representation will be similar as Table 1.

Similarly, we will represent our candidate motifs also in matrix

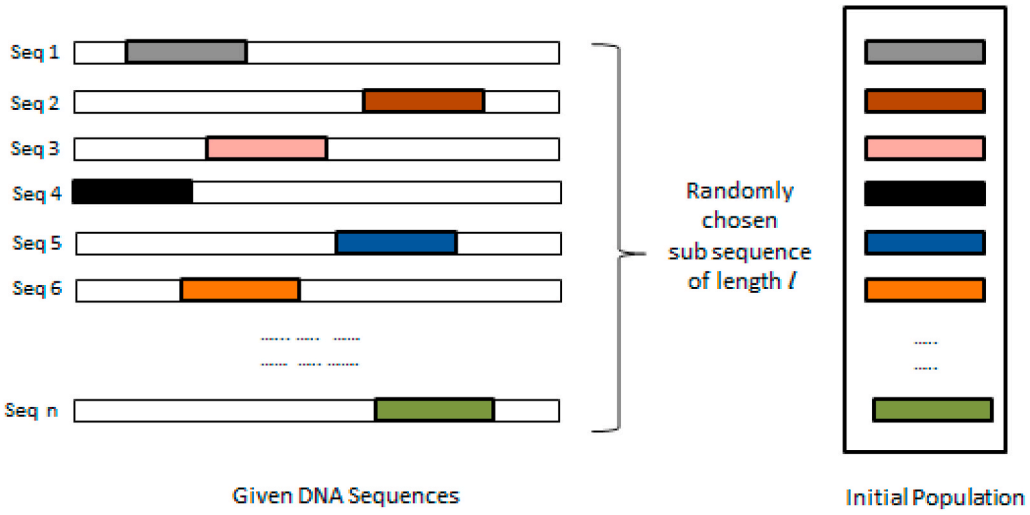


Fig. 1. Building initial population technique.

format. As we are only using 1 and 0, it will be useful for us to crossover and mutate the candidate motifs so that we can get more candidate motifs from the motif space. We will get the optimal solution of motif finding in a Motif matrix with base frequencies. For example, motif of a given data set of length 7 looks like Table 2.

3.2. Proposed algorithm

We have proposed a solution for finding DNA motifs from a given data set of a given length. We have built an evolutionary approach which finds the optimal solution of motifs of given data set. We intentionally build up initial population which leads up to the probably better solution in the solution space. Then we generate the next generation of population using the best fitted solution among the previous population and also the information about how the optimal solution look like. We give a big jump in the solution space randomly so that we can rule out the possibility of sticking to the local optima. Our proposed algorithm runs this approach several times with different initial population so that we can ensure the global optimum solution. Pseudo code of our proposed algorithm for finding motif in a data set is described below.

Algorithm for Motif Finding

$listOfBest \leftarrow \{\}$
$\mu \leftarrow$ number of parents selected
$\lambda \leftarrow$ number of children generated by parents
$l \leftarrow$ length of motif we want to find
for l times **do**
  $P \leftarrow buildInitialPopulation()$
  $Best \leftarrow \{\}$
  repeat
    for each candidate $P_i \in P$ **do**
      $AssessFitness(P_i)$
      if Best=$\{\}$ or Fitness(P_i) > Fitness(Best) **then**
        Best $\leftarrow P_i$
      end if
    end for
    $Q \leftarrow \mu$ individuals in P which fitness are greatest
    $P \leftarrow Q$
    for each candidate motif $Q_j \in Q$ **do**
      for $\frac{\lambda}{\mu}$ times **do**
        $P \leftarrow P \cup \{Mutate(Copy(Q_j))\}$
      end for
    end for
  until Best is ideal solution
  $listOfBest \leftarrow listOfBest \cup Best$
end for
$MOTIF \leftarrow Consensus(listOfBest)$
return $MOTIF$

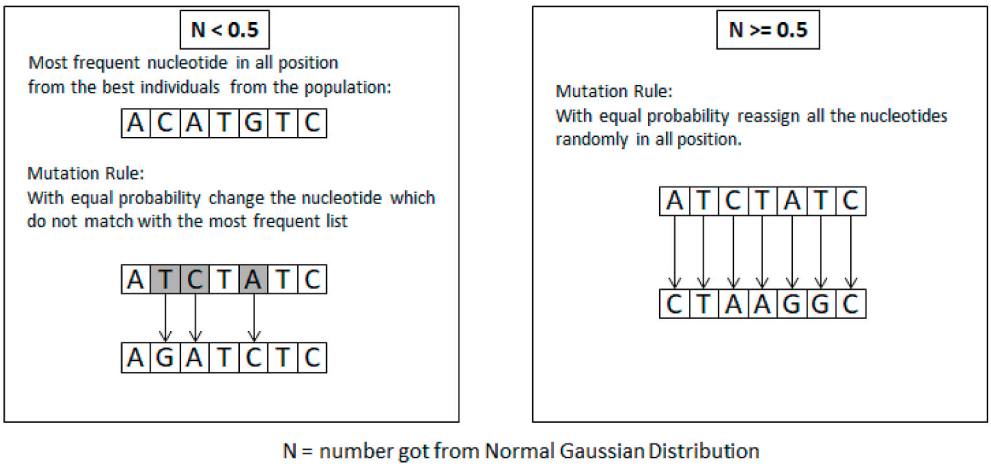


Fig. 2. Mutation technique for building next population.

Table 3
Motif matrix.

	0	1	2	3	4
A	2/4	0/4	0/4	1/4	1/4
T	0/4	1/4	4/4	1/4	1/4
C	2/4	1/4	0/4	1/4	1/4
G	0/4	2/4	0/4	1/4	1/4

We have used several methods in our algorithm. These methods are the key features to improve the performance of the solution. These methods are described in details in the following part.

3.2.1. *Building initial population*

This method will generate the initial population for solving our problem. This method will select random sub-sequence of length l from the DNA sequences and they will fill up the initial population as candidate motifs. Selecting the initial population from the data set tends to select the sub-sequences which are close to the optimal solution as they already exist in the data set. Our initial population building technique is illustrated in details in Fig. 1.

3.2.2. *Assess fitness*

In this approach, we will assess the fitness of the individuals, i.e, candidate motifs by using Equation (1). We will find the minimum distance of the candidate motif with the sequences of the data set and will get the total distance from the whole data set which will be the parameter of the fitness. The lowest value will represent the highest fitted candidate solution for the data set.

Algorithm for $AssessFitness(P_i)$

$fitness \leftarrow 0$
$l \leftarrow$ length of motif we want to find
for each sequence in data set **do**
  $tempMin \leftarrow \infty$
  for each sub-sequence of length l in the sequence **do**
    $d =$ Hamming Distance between P_i and sub-sequence
    if $d < tempMin$ **then**
      $tempMin \leftarrow d$
    end if
  end for
  $fitness \leftarrow fitness + tempMin$
end for
return $fitness$

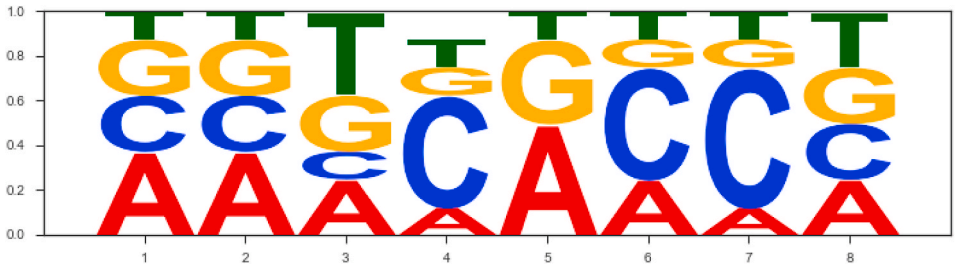


Fig. 3. Motif Logo of length 8 for data set hm01r.

Table 4
Experimental Result for different values of L.

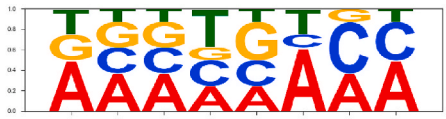
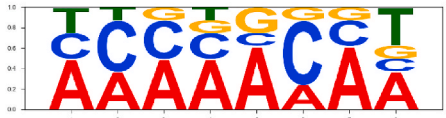
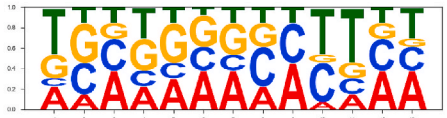
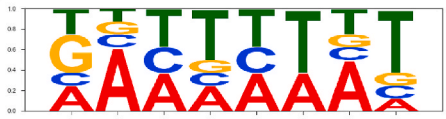
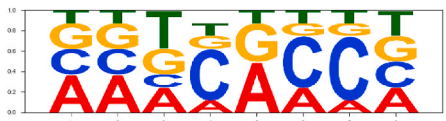
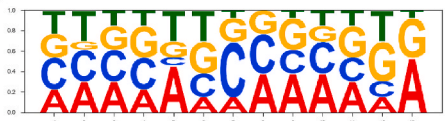
Length of Motif (L)	Accuracy (%)
8	100.0
13	100.0
18	96.91
23	98.07
28	97.42
33	91.91

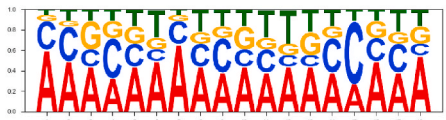
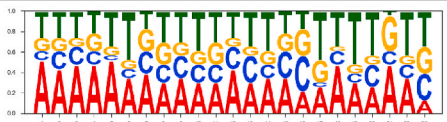
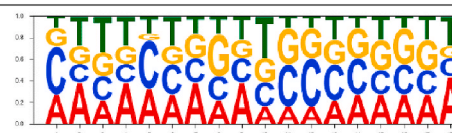
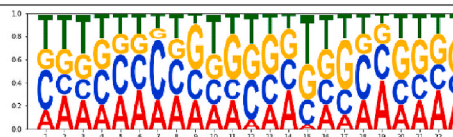
3.2.3. Mutation

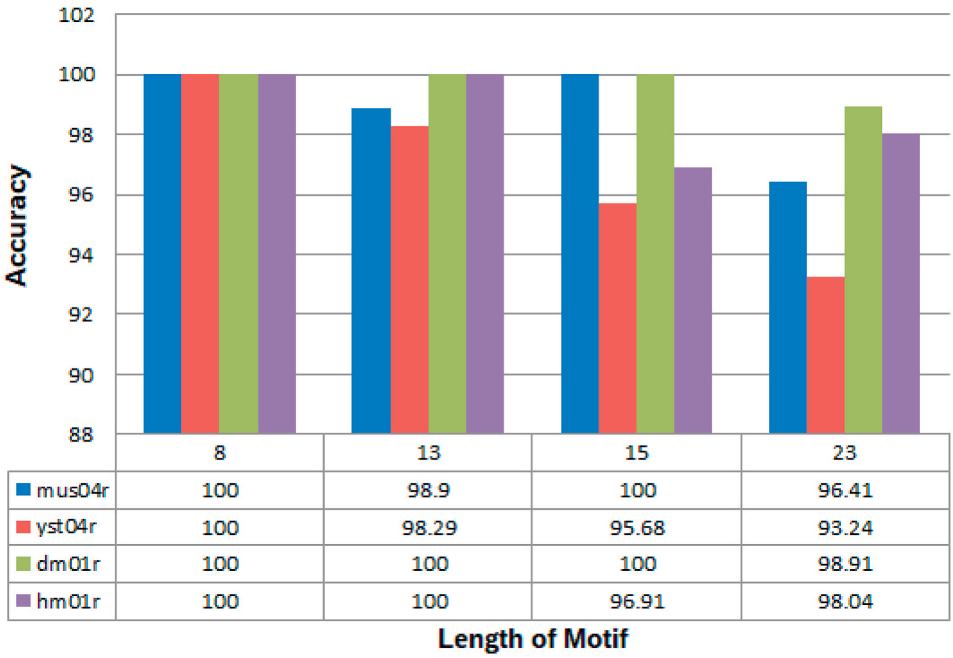
This part plays the vital role to roam around the candidate motif space and manages trade off between the exploration and exploitation of the search space. In this method, we will find the highest frequent nucleotide in every position from the best candidate motifs found in the iteration. This list gives us the best possible combination of nucleotides based on their fitness assessed so far. We maintained the trade off between exploration and exploitation by introducing Normal Gaussian Distribution. Normal distribution gives us lower numbers between 0 and 1 and rarely gives us higher numbers. We have used this technique so that we can always be in the best part of the solution space and rarely we give a big jump and go to totally different part of the solution space to find if any other best solution exists (Fig. 2).

When we stay at the current part of the solution space, we tends to move to better solution by only replacing the nucleotides which are not most frequently present with the help of a list where the highest frequent nucleotides so far at different positions are stored. And when we try to move to other part of the solution space we randomly change the nucleotides at all the positions and end up in different part of the candidate motif space.

Table 5
Motif Logo from different data set.

Data set	Length =8	Length =13
mus04r		
dm01r		
yst01r		
hm01r		

Data set	Length =18	Length =23
mus04r		
dm01r		
yst01r		
hm01r		



	8	13	15	23
mus04r	100	98.9	100	96.41
yst04r	100	98.29	95.68	93.24
dm01r	100	100	100	98.91
hm01r	100	100	96.91	98.04

Fig. 4. Accuracy of the motifs in different data set for different length.

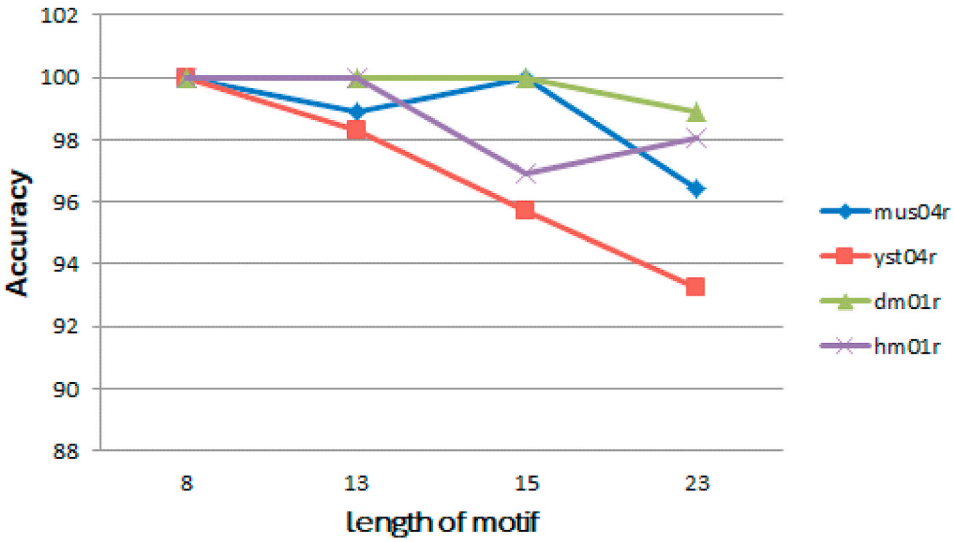


Fig. 5. Graph of accuracy vs length in different data set for different length.

3.2.4. Consensus of best individuals

This part of our proposed method calculates the frequency of each nucleotide in each position and returns a matrix containing the probability of each nucleotide in each position. For example, if the listOfBest contains the candidate solutions – {ATTCG, ACTGT, CGTAC, CGTTA} then this method will return Table 3.

4. Result

The performance of our proposed evolutionary approach for different instances is experimented using the benchmark data sets available by M. Tompa et al. [26]. This data set contains DNA sequences of fly, human, mouse and yeast. They contains three different kind of benchmark data sets - Real data sets having binding sites in their real genomic promoter sequences, Generic data sets having binding sites planted in randomly chosen genomic promoter sequences and Markov data sets having binding sites planted in sequences randomly generated according to a Markov chain of order 3. Number of DNA sequences in each of the data set varies from 5 to 35 and length of each sequence varies from 1000 to 2500.

We have experimented data sets from Ref. [26] which contains DNA sequences of different length. We ran our implemented algorithm on these data sets for finding motif of different length and calculated the accuracy which measures how much accurate our found motifs are according to the definition of motif. This accuracy is calculated by that we can confirm the validity of the found motif. For measuring accuracy, firstly, we have calculated the average number of mismatch for the l-mer containing the highest occurring nucleotide in each position from the motif matrix. We will use the formula in Equation (3) to calculates the average mismatch for any l-mer in a given data set.

$$Average\ Mismatch = \frac{1}{n}\sum_{i=1}^{n}\min\left(d_H\left(M, S_i\right)\right) \tag{3}$$

Table 6
Comparison with established methods based on Specificity of different data set.

Data set	Our Method	AlignACE	ANN-Spec	Consensus	GLAM	Improbizer	MEME	MEME3
dm05g	0.9967302	1	0.9837875	0.9771117	0.9914169	0.9893733	0.9701635	0.9591281
dm05r	0.9946866	–	–	–	–	–	–	–
dm07m	0.9946667	1	0.982	1	0.99	0.984	0.9911111	0.9773333
hm03r	0.991091	0.9875274	0.992119	1	0.9869792	0.9882812	0.9853344	0.989926
hm06g	0.9735593	1	1	1	0.9613559	0.9708475	1	1
hm07r	0.9917915	–	–	–	–	–	–	–
hm14r	0.9864442	1	0.9655892	1	0.9885297	0.9791449	0.9708029	1
hm15r	0.9959545	1	0.987737	1	0.9903919	0.9878635	0.9903919	0.9575221
hm21r	0.9918484	–	–	–	–	–	–	–
hm24m	0.9856704	0.9344933	0.9467758	1	1	0.9631525	0.9836233	0.9959058
hm25g	0.9827957	1	0.9612903	1	0.9688172	0.9526882	0.9795699	0.9516129
mus01r	0.990106	0.9448763	0.9526502	0.9597173	1	0.9809187	0.9724382	0.9540636
mus02r	0.9925867	1	0.9673814	1	0.9897354	0.9759352	0.9849453	1
mus05r	0.9728033	1	0.9717573	1	0.9801255	0.95659	0.9497908	0.9314854
mus06g	0.9832519	0.9371947	0.9253315	0.9351012	0.9832519	0.9330077	0.923238	0.9351012
mus12m	0.9822878	1	0.9498155	0.9409594	0.9800738	0.9557196	1	0.9121771
yst01g	0.9868214	0.9794999	0.988173	1	0.9858076	0.98164	1	0.9891868
yst04r	0.9918547	0.9855171	0.9809167	0.9853467	0.9831317	0.9672857	0.9868802	0.9853467
yst06g	0.9853293	1	0.9491018	1	0.9598802	0.9329341	1	1
yst07m	0.984	0.9733333	0.9486667	0.96	0.97	0.948	0.984	0.967

* - represents that the data set was not used for that specific algorithm.

Table 7
Comparison with established methods based on Specificity of different data set.

Data set	Our Method	MITRA	MotifSampler	oligodyad analysis	QuickScore	SeSiMCMC	Weeder	YMF
dm05g	0.9967302	1	0.9916894	0.9852861	0.9877384	0.976158	1	0.9846049
dm05r	0.9946866	–	–	–	–	–	–	–
dm07m	0.9946667	1	0.9864444	0.9937778	0.9764444	0.96	0.992	0.9868889
hm03r	0.991091	0.9924616	0.9908854	1	0.9909539	0.979852	1	0.9882127
hm06g	0.9735593	0.9909605	0.9862147	1	0.9715254	0.9500565	0.9837288	0.9317514
hm07r	0.9917915	–	–	–	–	–	–	–
hm14r	0.9864442	0.8800834	0.9807091	0.991658	1	0.911366	0.9843587	0.9885297
hm15r	0.9959545	0.9949431	0.9893805	1	0.9931732	0.9726928	0.9949431	0.9908976
hm21r	0.9918484	–	–	–	–	–	–	–
hm24m	0.9856704	0.988741	0.975435	1	1	0.9549642	0.9907881	0.9892528
hm25g	0.9827957	0.9569892	0.9591398	0.9645161	1	0.883871	14	0.9698925
mus01r	0.990106	0.9526502	0.9632509	1	0.9745583	0.8869258	0.9717314	0.9773852
mus02r	0.9925867	0.9887089	0.9828923	0.9944115	0.9852874	0.9671533	1	0.99099
mus05r	0.9728033	0.9769874	0.9722803	1	0.9780335	0.9241632	0.9790795	0.9048117
mus06g	0.9832519	0.9748779	0.9727844	0.9867411	0.961619	0.9064899	0.9720865	0.9832519
mus12m	0.9822878	0.9313653	0.9505535	1	1	0.896679	1	0.9830258
yst01g	0.9868214	0.9911016	1	1	0.9891868	0.9720658	0.9945934	0.9788241
yst04r	0.9918547	0.9923326	0.9844948	0.991651	0.9918214	0.9759755	0.9923326	0.9959107
yst06g	0.9853293	0.9835329	0.9988024	0.9979042	0.9784431	0.9538922	0.997006	1
yst07m	0.984	0.976	1	1	0.9776667	0.928	0.984	0.992

* - represents that the data set was not used for that specific algorithm.

where.

M = Best combination of nucleotides from motif matrix
S_i = ith sequence of the data set
n = number of sequences in the data set
d_H (M, S_i) = minimum mismatch of M comparing with all the l-mer from the sequence S_i

After calculating average mismatch for the whole data set, we calculate the accuracy of our result using Equation (4). It will give more accuracy if the average mismatch is low which means we have found better motif, which is present in all the sequences with a little mismatch.

$$Accuracy = \frac{L - X}{L} \times 100 \quad (4)$$

where.

L = Length of motif and
X = Average mismatch of best l-mer from motif matrix.

We have run our algorithm on data set of "hum01r" for finding motifs of different length. Fig. 3 shows the found motif of length 8 from the human sequences in the data set. We have experimented the data set for finding motifs of different length and the results are shown in Table 4 which represents the accuracy of found best motif following our approach. Later, we have represented the results in an Accuracy Vs Length graph.

From Table 1, we see that our approach finds motif with 100% accuracy for length of 8 and 13. We are getting 100% accuracy for shorter length because "hm01r" is a real data set of human DNA sequences and in real sequence short repeating patterns are abundant. So, our heuristic based approach finds the appropriate motif matrix for the data set. For longer length motif, our approach finds motifs having accuracy above 91%. As in the real dataset motifs of longer length is not very high and it is quite difficult to find motifs which are present in all the sequence with no mismatch. In this case our goal is to fund the repeating patterns which are repeated having the least possible number of mismatch and our method has optimized the mismatch and found the motifs having above 91% similarity.

We have applied our method to all the data set from Ref. [26] and

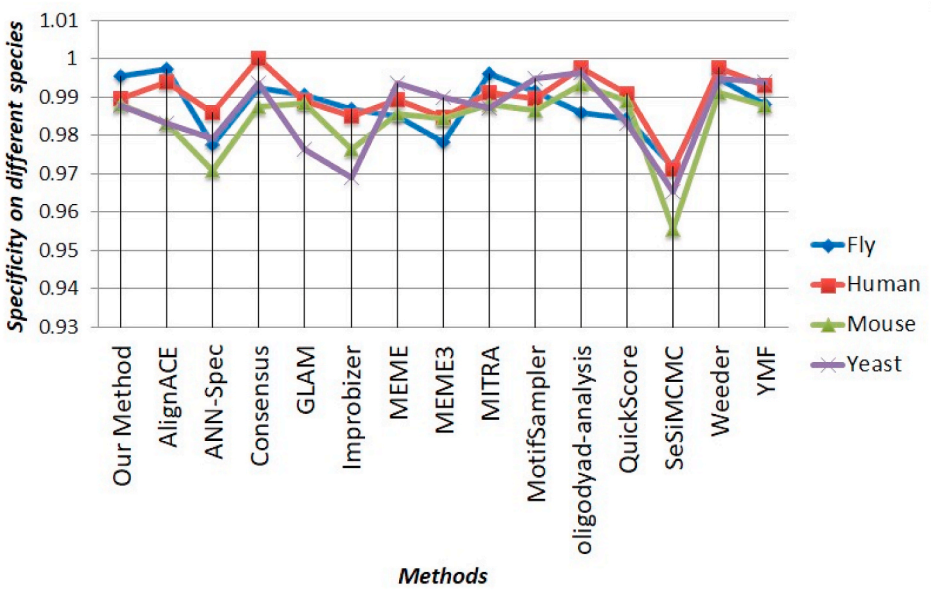


Fig. 6. Comparison with other methods for different species data.

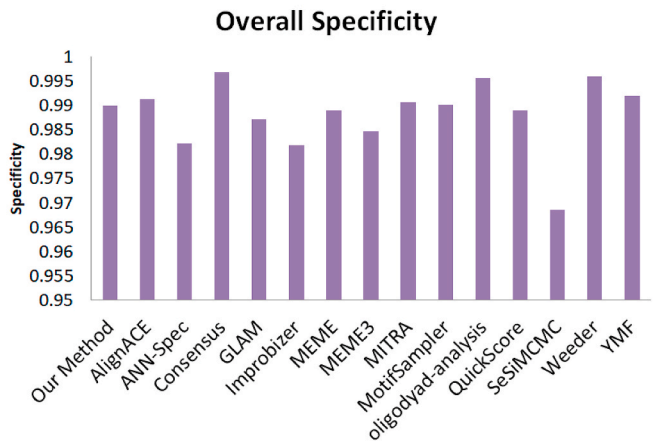


Fig. 7. Overall comparison with other methods based on Specificity.

experimented how our proposed method performs for finding motifs of different length. Found motifs of different length from different data sets are shown in the form of motif logo in Table 5.

Fig. 4 and Fig. 5 shows the accuracy for found motifs of different length in different data set. From the graph we see that for the data set "mus04r" and "yst04r", accuracy decreases as the length of motif increases. These data sets contains DNA sequences which contains 1000 base pairs. On the contrary, other data sets contain sequences of length more than 1000 and from these we get motifs having accuracy up to 100%. Whenever the length of sequence increases, the chances of finding a sub-sequence of length l which has least number of mismatch increases which is why we get more accurate motifs.

We have calculated the specificity of our found motifs for different data sets and compared with some established motif finding methods. We have compared our result with established methods available at http://bio.cs.washington.edu/assess. It contains result of AlignACE [27], ANN-spec [28], Consensus [29], GLAM [30], Improbizer [31], MEME [32], MEME3 [32], MITRA [33], MotifSampler [34], oligo/dyad-analysis [26], QuickScore [35], SeSiMCMC [36], Weeder [37] and YMF [38].

We need to define some definitions before comparing the results.

- True Positives (TP): Number of positions in both known sites as well as predicted sites.
- False Positives (FP): Number of positions in predicted sites which are not present in known sites
- True Negatives (TN): Number of positions, which are neither in known sites nor in predicted sites.
- False Negatives (FN): Number of positions that are in known sites but not in predicted sites

Specificity of the found motifs are calculated based on the following equation which gives us how accurately the algorithm performed to get the real motifs.

$$nSP = \frac{nTN}{nTN + nFP} \tag{5}$$

Tables 6 and 7 show the performance of our method with other established method in terms of specificity. Our method is an evolutionary method. These tables show that our method works better than some of the existing methods for some specific data sets like mus05r, hm25 g, hm06 g, mus01r etc. Further investigation confirms that, these data sets have less amount of number of sequences. Our method works close to other established methods for those data sets where the number of sequences is high and performs better than others in those data sets where the number of sequences is low.

Fig. 6 shows the specificity comparison of the algorithms according to different species. We see that, our algorithms outperforms all other methods for finding motifs for fly. It performs close to other established algorithms for human, mouse and yeast. And Fig. 7 shows the overall comparison with other algorithms in terms of specificity. Our proposed algorithm matches the performance of the established methods. In addition, our method can find large motifs from the data set.

We have tested our implemented method on different real data set with different values of l and d. In the real data set, motifs of larger length is rarely conserved. So, it is difficult to find motifs which is has exactly length l and highest mismatch d. Still, our method finds the optimal possible motifs of larger length and finds the accurate motifs for comparatively smaller length.

5. Conclusion

In this work, we have proposed an evolutionary approach for finding motifs in DNA sequences. We have built an initial candidate motif set

and best candidates are filtered out from these. Performing mutation in the best candidates tends to generate the optimal motif of the data set. Our approach has performed well in different data set and matches the specificity with established methods which validates the usefulness of our method. Besides, this method ensures to find motifs of very large length which is not easy to find using any other exhaustive method as it takes even days to calculate. As we are using heuristic and selecting the best candidates in every step of evolution, our method faces no difficulty in finding motif of large length.

However, our approach can find motifs with higher accuracy and specificity even for the longer length. Still there are rooms for improvement of our proposed method by introducing new technique in mutation step and modifying some of the knobs - distribution technique, number of parents and child for next step of evolution, distance measuring of DNA sequences etc. This approach can have a great impact on the biologists who need to find motifs of different length with good accuracy and specificity.

Authors' contributions

FB Ashraf.: provided the conception and design of the study, acquisition of data, analysis and interpretation of data, drafting the article, revised it critically for important intellectual content, and final approval of the version to be submitted; MSR Shafi: supplied the acquisition of data, implementing the algorithm, drafting of manuscript.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

None. No funding to declare.

References

[1] Xiao P, Cai X, Rajasekaran S. Ems3: an improved algorithm for finding edit-distance based motifs. In: 2018 IEEE 8th international conference on computational advances in bio and medical sciences (ICCABS). IEEE; 2018. 1–1.
[2] Näär AM, Boutin J-M, Lipkin SM, Victor CY, Holloway JM, Glass CK, Rosenfeld MG. The orientation and spacing of core dna-binding motifs dictate selective transcriptional responses to three nuclear receptors. Cell 1991;65(7):1267–79.
[3] Rajasekaran S, Balla S, Huang C-H, Thapar V, Gryk MR, Maciejewski MW, Schiller MR. Exact algorithms for motif search. In: APBC; 2005. p. 239–48.
[4] Pal S, Rajasekaran S. Improved algorithms for finding edit distance based motifs. In: Bioinformatics and biomedicine (BIBM), 2015 IEEE international conference on. IEEE; 2015. p. 537–42.
[5] Martinez HM. An efficient method for finding repeats in molecular sequences. Nucleic Acids Res 1983;11(13):4629–34.
[6] Nicolae M, Rajasekaran S. Efficient sequential and parallel algorithms for planted motif search. BMC Bioinf 2014;15(1):1.
[7] Price A, Ramabhadran S, Pevzner PA. Finding subtle motifs by branching from sample strings. Bioinformatics 2003;19(suppl 2):ii149–55.
[8] Jones NC, Pevzner PA, Pevzner P. An introduction to bioinformatics algorithms. MIT press; 2004.
[9] Che D, Song Y, Rasheed K. Mdga: motif discovery using a genetic algorithm. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation. ACM; 2005. p. 447–52.
[10] Pavesi G, Mauri G, Pesole G. An algorithm for finding signals of unknown length in dna sequences. Bioinformatics 2001;17(suppl 1):S207–14.
[11] T. L. Bailey, C. Elkan, et al., Fitting a mixture model by expectation maximization to discover motifs in biopolymers.
[12] Stormo GD, Hartzell GW. Identifying protein-binding sites from unaligned dna fragments. Proc Natl Acad Sci Unit States Am 1989;86(4):1183–7.
[13] Liu X, Brutlag DL, Liu JS. Bioprospector: discovering conserved dna motifs in upstream regulatory regions of co-expressed genes. In: Biocomputing 2001. World Scientific; 2000. p. 127–38.
[14] Roth FP, Hughes JD, Estep PW, Church GM. Finding dna regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantitation. Nat Biotechnol 1998;16(10):939.
[15] Thijs G, Marchal K, Lescot M, Rombauts S, De Moor B, Rouzé P, Moreau Y. A gibbs sampling method to detect over-represented motifs in the upstream regions of co-expressed genes. In: Proceedings of the fifth annual international conference on Computational biology. ACM; 2001. p. 305–12.
[16] Geman S, Geman D. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. In: Readings in computer vision. Elsevier; 1987. p. 564–84.
[17] Fogel GB, Weekes DG, Varga G, Dow ER, Harlow HB, Onyia JE, Su C. Discovery of sequence motifs related to coexpression of genes using evolutionary computation. Nucleic Acids Res 2004;32(13):3826–35.
[18] Wei Z, Jensen ST. Game: detecting cis-regulatory elements using a genetic algorithm. Bioinformatics 2006;22(13):1577–84.
[19] Chan T-M, Leung K-S, Lee K-H. Tfbs identification by position-and consensus-led genetic algo- rithm with local filtering. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM; 2007. p. 377–84.
[20] Dorigo M, Stützle T. The ant colony optimization metaheuristic: algorithms, applications, and ad- vances. Handbook of metaheuristics. Springer; 2003. p. 250–85.
[21] Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics) 1996;26(1):29–41.
[22] Stützle T, Hoos HH. Max–min ant system. Future Generat Comput Syst 2000;16(8):889–914.
[23] Huan HX, Tuyet DT, Ha DT, Hung NT. An efficient ant colony algorithm for dna motif finding. In: Knowledge and systems engineering. Springer; 2015. p. 589–601.
[24] Yang C-H, Liu Y-T, Chuang L-Y. Dna motif discovery based on ant colony optimization and expectation maximization. Proceedings of the international multi conference of engineers and computer scientists, vol. 1; 2011.
[25] Garbelini JMC, Kashiwabara AY, Sanches DS. Sequence motif finder using memetic algorithm. BMC Bioinf 2018;19(1):4.
[26] Tompa M, Li N, Bailey TL, Church GM, De Moor B, Eskin E, Favorov AV, Frith MC, Fu Y, Kent WJ, et al. Assessing computational tools for the discovery of transcription factor binding sites. Nat Biotechnol 2005;23(1):137.
[27] Ma X, Kulkarni A, Zhang Z, Xuan Z, Serfling R, Zhang MQ. A highly efficient and effective motif discovery method for chip-seq/chip-chip data using positional information. Nucleic Acids Res 2012;40(7). e50–e50.
[28] Workman CT, Stormo GD. Ann-spec: a method for discovering transcription factor binding sites with improved specificity. In: Biocomputing 2000. World Scientific; 1999. p. 467–78.
[29] Buhler J, Tompa M. Finding motifs using random projections. J Comput Biol 2002;9(2):225–42.
[30] Frith MC, Hansen U, Spouge JL, Weng Z. Finding functional sequence elements by multiple local alignment. Nucleic Acids Res 2004;32(1):189–200.
[31] Pavlidis P, Furey TS, Liberto M, Haussler D, Grundy WN. Promoter region-based classification of genes. In: Biocomputing 2001. World Scientific; 2000. p. 151–63.
[32] Bailey TL, Johnson J, Grant CE, Noble WS. The meme suite. Nucleic Acids Res 2015;43(W1):W39–49.
[33] Eskin E, Pevzner PA. Finding composite regulatory patterns in dna sequences. Bioinformatics 2002;18(suppl 1):S354–63.
[34] Thijs G, Lescot M, Marchal K, Rombauts S, De Moor B, Rouze P, Moreau Y. A higher-order background model improves the detection of promoter regulatory elements by gibbs sampling. Bioin- formatics 2001;17(12):1113–22.
[35] Régnier M. Mathematical tools for regulatory signals extraction. In: Bioinformatics of genome regu- lation and structure. Springer; 2004. p. 61–9.
[36] Favorov AV, Gelfand MS, Gerasimova AV, Ravcheev DA, Mironov AA, Makeev VJ. A gibbs sampler for identification of symmetrically structured, spaced dna motifs with improved estimation of the signal length. Bioinformatics 2005;21(10):2240–5.
[37] Pavesi G, Mereghetti P, Mauri G, Pesole G. Weeder web: discovery of transcription factor binding sites in a set of sequences from co-regulated genes. Nucleic Acids Res 2004;32(suppl 2):W199–203.
[38] Sinha S, Tompa M. Ymf: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. Nucleic Acids Res 2003;31(13):3586–8.