Evolutionary Intelligence (2021) 14:1707–1726
https://doi.org/10.1007/s12065-020-00444-2

RESEARCH PAPER

DNA motif discovery using chemical reaction optimization

Sumit Kumar Saha[1] · **Md. Rafiqul Islam**[1] · **Mredul Hasan**[1]

Received: 19 July 2019 / Revised: 13 May 2020 / Accepted: 24 June 2020 / Published online: 11 July 2020

Abstract
DNA motif discovery means to find short similar sequence elements within a set of nucleotide sequences. It has become a compulsory need in bioinformatics for its useful applications such as compression, summarization, and clustering algorithms. Motif discovery is an NP-hard problem and exact algorithms cannot solve it in polynomial time. Many optimization algorithms were proposed to solve this problem. However, none of them can show its supremacy by overcoming all the obstacles. Chemical Reaction Optimization (CRO) is a population based metaheuristic algorithm that can easily fit for the optimization problem. Here, we have proposed an algorithm based on Chemical Reaction Optimization technique to solve the DNA motif discovery problem. The four basic operators of CRO have been redesigned for this problem to search the solution space locally as well as globally. Two additional operators (repair functions) have been proposed to improve the quality of the solutions. They have been applied to the final solution after the iteration stage of CRO to get a better one. Using the flexible mechanism of elementary operators of CRO along with the additional operators (repair functions), it is possible to determine motif more precisely. Our proposed method is compared with other traditional algorithms such as Gibbs sampler, AlignACE (Aligns Nucleic Acid Conserved Elements), MEME (Multiple Expectation Maximization for Motif Elicitation), and ACRI (Ant-Colony-Regulatory-Identification) by testing real-world datasets. The experimental results show that the proposed algorithm can give better results than other traditional algorithms in quality and in less running time. Besides, statistical tests have been performed to show the superiority of the proposed algorithm over other state-of-the-arts in this area.

Keywords Motif discovery · Binding sites · Information content · Chemical reaction optimization · Meta-heuristics

1 Introduction

Generally, a motif is an idea, subject, theme, pattern etc. which repeats itself and has some significance, especially in a musical work, literary, artistic or set of sequences[1]. In bioinformatics, motif discovery means the process of determining motifs within a set of DNA, RNA or protein sequences where motif means a widespread amino-acid sequence or nucleotide pattern that captures a biological significance. Motifs are generally short sequence patterns of a fixed length that express important functional or structural features in protein sequences and nucleic acids such as active sites, transcription binding sites, interaction interfaces or splice junctions[2]. They can appear in an approximate or exact form within a family or subfamily of sequences. In other words, a pattern common to a set of DNA, RNA or protein sequences that shares same biological property, such as functioning as binding sites for a particular protein is called motif. So we can say that the problem of identifying short similar sequence elements shared by a set of protein or nucleotide sequences with a general biological function is known as motif discovery[3].

Figure 1 shows an example of motif discovery. Position weight matrix (PWM) have been used to express motifs. PWM is the representation of the occurrences of nucleotide at each position of a motif. Let the number of DNA sequences, $n = 6$ having a length $L = 30$ for each sequence. Here we have to discover a motif of width $W = 8$ using PWM, f where PWM is usually used to express motifs[4]. Now from every sequence, we get a motif instance of length 8. These six motif instances have used in PWM. Now from

✉ Sumit Kumar Saha
sumit.ku.cse@gmail.com

Md. Rafiqul Islam
dmri1978@gmail.com

Mredul Hasan
mredul38@cseku.ac.bd

[1] Computer Science and Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

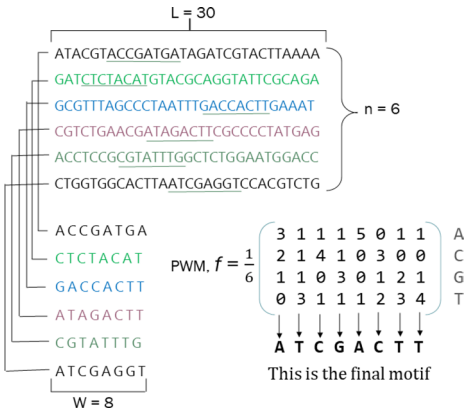


Fig. 1 Motif discovery of DNA sequences

the PWM, we select nucleotide with the highest occurrences in every position and get a motif of length 8. Thus the final motif from these instances has been discovered.

In the era of bioinformatics revolution, the volume of biological sequences is increasing in public databases. That is why motif discovery gradually becomes a fundamental problem in molecular biology and computer science[5]. The capability to predict the function, structure, or behavior of biological entities or motifs such as proteins and genes, additionally cooperation among them, play a major role in the analysis of information to describe biological mechanisms. Motif discovery is, therefore, an important field of bioinformatics. There are two main ways to discover a motif using biological experiments and computing approaches, i.e; bioinformatics. But biological experiments are very costly and time-consuming processes. So computing approaches are extensively used to discover motifs[6]. But exact motif discovery is a tough problem. Because the lengths of motifs are generally very short such as up to 30 nucleotides, although the regulatory regions accommodate motifs are very long such as a range from several hundred to several thousand nucleotides. Again the mutations of the actual instances of motifs are added as a burden[5].

Many algorithms were proposed to predict motifs, such as Gibbs sampler[7], MEME (Multiple Expectation Maximization for Motif Elicitation)[8], GA (Genetic Algorithm) [9–12], GARPS (Genetic Algorithm with Random Projection Strategy)[13], ACO (Ant Colony Optimization), ACO-Motif (An Efficient Ant Colony Algorithm for DNA Motif Finding)[6], EMACO (Ant Colony Optimization (ACO) and Expectation Maximization (EM))[14], MFACO (Motif Finding using Ant Colony Optimization)[15], ACRI (Ant-Colony-Regulatory-Identification)[16], MotifSuite[17], MotifSampler, Bioprospector (BioProspector is an algorithm which is used to discover sequence motifs from a set of DNA sequences)[18], an iterative algorithm (based on GA with addition operator) for motif discovery[5] etc. Gibbs sampler and MEME have drawbacks of dropping into local optimum easily. The consuming time of Gibbs sampler is lower but less prediction accuracy, and MEME is superior to the other methods by its prediction accuracy but time-consuming[5]. Again there are some heuristic methods for predicting motifs, such as particle swarm optimization, Tabu search algorithm, and Simulated Annealing[19]. Some of the basic limitations of these algorithms are low prediction accuracy of binding sites and nucleotide levels. Another limitation of transaction factors is the pattern model to detect the regularity among the binding sites[20]. To enable the biologist in determining functional motifs from statistical artifacts, many algorithms do not produce good motif statistics. For this reason, valid motifs can be rejected, or time may be wasted by searching random motifs. The main drawback of genetic algorithm based on statistical significance is the lack of a mechanism to identify false positives[9]. Though modified genetic algorithm[12] gets better results than Gibbs, MEME, Consensus and genetic algorithm[10]. But it has a low contextual connection with the motifs which are introduced by TRANSFAC (A Database on Transcription Factors and Their DNA Binding Sites)[21]. Though all algorithms have some limitations, they produce better results in some restricted inputs criteria.

Motif discovery has several important application areas. It is widely used in locating regulatory sites and drug target identification[5]. Mainly, it is used to analyze the information for describing biological mechanisms. Besides, motif discovery has become the main part of several higher-level algorithms handle with time series specially rule-discovery, compression, summarization, and clustering algorithms.

In this paper, we have proposed a nature inspired meta-heuristic approach called Chemical Reaction Optimization (CRO) that mimics the interaction behaviour of molecules participated in a chemical reaction. CRO showed promising results in the case of various optimization problems. We choose CRO algorithm to solve the motif discovery problem because this algorithm searches the solution space globally as well as locally. Thus it gives the benefits of both GA and SA[22–24]. It has the flexibility to adapt with different optimization problems according to the requirements by redefining its four operators as well as by using additional operators as needed. This algorithm always tries to find out a stable solution as like chemical reaction in the real world. CRO facilitates to avail variable size population, which permits the system to adapt automatically the problem being solved. When diversification is required, decomposition operator is triggered to produce more molecules in order to explore the solution space for finding out the optimal solution. On the other hand, the algorithm triggers synthesis for merging molecules when intensification is required. As a result, the probability of resultant molecules to be selected for manipulation is increased. It also follows the law of conservation

of energy where energy can be transformed from one form to different entities forms. The total amount of energy held by the molecules and buffer remains constant that makes the algorithm unique than other existing meta-heuristic algorithms. Besides, one can construct a molecule (solution) for different attributes that suit the problem to be solved. This advantage provides the flexibility to design and manage different operators as needed[22–24].

For properties and efficiency of CRO we refer two papers[23, 25]. Besides CRO is an algorithm which was used to solve many optimization problems effectively such as channel assignment problem in wireless mesh networks[22], shortest common supersequence[26], longest common subsequence[27], RNA structure prediction[28], transportation scheduling optimization by a collaborative strategy in supply chain management with TPL[29], quadratic assignment problem, resource-constrained project scheduling problem[30], RNA secondary structure prediction with pseudoknots[31], optimization of protein folding in HP cubic lattice model[32] etc. So we have designed the CRO algorithm by redesigning four basic operators and designing two additional operators to solve the motif discovery problem for finding better results than the existing algorithms.

Contribution and novelty of the proposed work are given below.

- A new population generation process has been introduced to make our proposed approach more efficient.
- The basic four operators of CRO have been redesigned to make suitable with DNA motif discovery problem. These operators help our proposed approach to search the solution space locally as well as globally.
- The values of the parameters are defined very carefully to find the global optimal solution efficiently.
- We have introduced an additional operator called repair function to improve the quality of the solution by searching all the neighboring solutions of the existing best solution. The local optimization technique is used as a second repair operator, which improves the quality of the binding sites.
- The results of the proposed work have been compared with several state-of-the-arts and statistical tests have been performed to show the efficiency of the proposed method compared with the other methods.

1.1 Basic concepts

Motif discovery problem can be stated as follows. Let a set of N DNA sequences is represented as $S = \{S_1, S_2, \ldots, S_N\}$, where $S_i = s_1, s_2, \ldots, s_{l_i}$ and l_i is the length of the sequence i. We have to find out the possible accurate motif pattern $X = x_1, x_2, \ldots, x_l$ of length l where x_i, and $s_i \epsilon \{A, C, G, T\}$.

Motif discovery is based on a defined score function that calculates the similarity of the motif pattern with its occurrences. There are two approaches for the given motif length l[10]. These are as follows:

1. **Consensus approach** Find a motif S_c of length l and a set of motif instances $M = \{m_1, m_2, \ldots, m_N\}$, where m_i is the motif instance of S_i so that S_c minimizes the total hamming distances given in Eq. 1.

$$T_{HD}(S_c) = \sum_{i=1}^{N} H_D(S_c, S_i) \tag{1}$$

where

$$H_D(S_c, S_i) = min\{H(S_c, m) : m \, is \, a \, subsequence \, of \, S_c \, of \, length \, l\} \tag{2}$$

and $H(S_c, m)$ is the hamming distance between the motif S_c and motif instances m. Here hamming distance means the number of positions in S_c and m where the nucleotides are not same. If we assume M as a consensus matrix where the ith row is the motif instance m_i and $c(k, j)$ is denoted as the number of nucleotides $k \epsilon \{A, C, G, T\}$ in column j. Then the CSC_M (consensus score) is defined as:

$$CSC_M = \sum_{j=1}^{l} \left(\max_{k \in \{A,C,G,T\}} \left(c(k,j) \right) \right) \tag{3}$$

2. **Positional approach** Find a set of motif instances $M = \{m_1, m_2, \ldots, m_N\}$ with a length l of each motif and a set of positions, $P = \{p_1, p_2, \ldots, p_N\}$ where p_i is the starting position of motif instance m_i in sequence S_i, then the objective function information content, IC_M of this approach is defined as:

$$IC_M = \sum_{j=1}^{l} \sum_{k \in \{A,C,G,T\}} F(k,j). \log_2 \frac{F(k,j)}{F_k} \tag{4}$$

where $F(k, j)$ represents the frequency of nucleotide k to be in position j of the matrix M and F_k indicates its background frequency in the entire set S.

In bioinformatics, motif discovery is an NP-hard problem[6]. If we use real biological DNA sequences where the length of the nucleotides (or amino-acid) are very large, then it is not possible to find the exact motif in polynomial time. So here we solve this problem using chemical reaction optimization (a metaheuristics method) and our target is to find optimal or near-optimal solutions. For finding a motif, we use the consensus approach, and to find their positions within the input DNA set the positional approach is used.

2 Related works

Different approaches were proposed for optimization of the Motif Discovery problem. Each algorithm has some drawbacks as well as some efficiencies. Some of the approaches are described below.

2.1 Greedy mixture learning for multiple motif discovery in biological sequences

Greedy mixture learning for multiple motif discovery in biological sequences (Greedy EM) was proposed by Blekas et al.[33]. This algorithm uses incremental methods for Gaussian mixture learning for finding significant motifs within a set of DNA sequences. A mixture of motif model with a greedy fashion is learned by adding motifs incrementally to the mixture as far as some stopping criteria are met. This method starts with one motif that models the background. Then a new candidate motif is added at every step. By local search using partial EM (Expectation Maximization) steps and global search for tuning the parameters, this algorithm finds a great initialized value for the parameters of the new candidate motif. This method uses original kd-trees to reduce the running time for querying the nearest neighbor. Greedy EM uses real datasets from the PRINTS database[34] and the PROSITE database of protein families[35] to compare their results with the MEME[8] algorithm. Greedy EM finds a great initialized value for the parameters of the new motif but it fails to find multiple motifs with variable length. The time complexity of the initialization procedure is reduced by the kd-tree technique[33].

2.2 Motif discovery using a genetic algorithm

In 2005 Che et al. proposed motif discovery using a genetic algorithm (MDGA)[10]. MDGA uses a generic framework of the genetic algorithm to explore all possible search spaces of the starting position of the motifs within different target sequences. In this algorithm, the number of initial population is selected randomly and kept fixed during evolution. A new individual is generated from two parents using crossover and mutation in every iteration. Thus the number of new individuals reduces to the half of the existing population. Then the new individuals are merged with the existing population and worst one-third individuals are eliminated from the total individuals. MDGA uses CRP dataset (contains 18 sequences with length 105 nucleotides)[36], YDR02c dataset (consists of 15 target genes of transcription factor YDR02c)[37] and AZFI dataset (consists of 24 sequences in which each sequence has variable lengths, ranging from 175 to 1228)[37] to compare its efficiency with the other algorithms such as Gibbs sampler[7], Bioprospector[18] and AlignACE[38] etc. It gives higher prediction accuracy than Gibbs sampler[7] and Bioprospector[18] with the CRP dataset[36]. From YDR20c sequence dataset[37], MDGA gets truer motif pattern from a statistical point of view and in AZFI sequence dataset[37], it consumes less time than AlignACE[38].

2.3 Motif discovery using evolutionary algorithms

In 2009 Shao et al. proposed motif discovery using evolutionary algorithms[39]. This algorithm integrates bacterial foraging optimization algorithm and Tabu Search (TS), it is also known as a TS-BFO algorithm. In this method, one candidate motif is referred to as one bacteria to undergo the evolution. There are four steps in a bacteria's foraging action: chemotaxis, swarming, reproduction, and elimination and dispersal. TS-BFO uses SCPD datasets[40] and TRANSFAC datasets[21] to compare its efficiency with the efficiency of other approaches such as DE/EDA [DE/EDA algorithm combines global information extracted by estimation of distribution algorithm (EDA) with differential information obtained by Differential evolution (DE)][41], MotifSampler, MEME (Multiple EM for Motif Elicitation)[8] etc. TS-BFO algorithm uses self-control multi-length chemotactic step approach to extend the search space, remove local extremum, and speed up the constringency. It cannot generate the similar individuals in each step, guides the search orientation, and discovers the global solution[39].

2.4 Motif finding using ant colony optimization

Bouamama et al.[15] proposed motif discovery using ant colony optimization (MFACO) algorithm in 2010. As a local heuristic optimization search step, this algorithm integrates a modified Gibbs sampling method. MFACO builds a weighted directed graph $G(V, E)$ with V is the set of nodes and E is the set of edges. This graph contains $4l$ nodes organized in a grid of four rows and l columns where l is the motif's length. Every ant builds a solution incrementally by traversing the graph to complete a tour. MFACO searches both in the space of motif patterns and starting position. So it has better chances to detect potential motif. Three datasets used in FMGA (finding motifs by genetic algorithm)[11] and *E. coli* CRP binding sites[36] are used to test the performance of MFACO. The three datasets consist of 6, 9, 18 sequences respectively where each sequence has an equal length of 3001 nucleotides. For this datasets, MFACO can acquire better performance in terms of motif accuracy than MEME (Multiple EM for Motif Elicitation)[8], Motif Sampler, BioProspector (BioProspector algorithm is used to discover sequence motifs from a set of DNA sequences)[18], and FMGA (Finding Motifs by Genetic Algorithm)[11] within a reasonable computational time. *E. coli* dataset contains 18 sequences with length 105 nucleotides. For this datasets, MFACO is able to find the exact starting positions of the motifs identified by Footprinting while the

other approaches such as BioProspector[18], MDGA (Motif Discovery using A Genetic Algorithm)[10] failed.

2.5 Optimizing genetic algorithm for motif discovery

Hongwei et al.[13] proposed a new algorithm in 2010 named GARPS that optimizes Genetic Algorithm (GA) via Random Projection Strategy (RPS) to identify (l, d)-motifs. Though the initial population used in this algorithm is generated from RPS that makes it capable of fast convergent to the best solution, the overall structure of GARPS is derived from the simple genetic algorithm. In the creation of every new generation, a simple mutation operator named the one-point crossover and keeping the best mechanism are used. Generation after generation, these steps are repeated iteratively in a while loop. During these iterations, new individuals appear because crossover and mutation operators are performed on the population. The best individuals survive using the best-keeping mechanism is guaranteed by the selection operator. As GARPS progresses, the average fitness of the population is increased and it stops when no more improvement can be made. The GARPS algorithm was compared with the Projection Algorithm and showed the better results. They used several data including eighteen sequences of identified binding sites of cAMP receptor protein (CRP)[36], seven sequences of identified binding sites of PDR3[42]. This algorithm cannot find the extremely weak planted motif unless the algorithm reports a sufficient number of patterns[13].

2.6 DNA motif discovery based on ant colony optimization and expectation maximization

Yang et al.[14] proposed a framework in 2011 with the combined ability of the Ant Colony Optimization (ACO) and Expectation Maximization (EM) known as EMACO. ACO is effective in global search and EM is efficient to maximize the likelihood of parameter estimation that makes these two algorithms adequately complementary. Initially, some potential binding sites are randomly extracted from the given sequences. Next, ACO applies iteratively over all these solutions to construction and updates pheromone in search of good motifs. To maximize the likelihood of parameter, the EM algorithm uses predictions found from ACO. Expectation step of EM, calculate the expected value of the log-likelihood function given the observed data under the current estimation of the missing motif sites. The maximization step finds the positions of motif instances. After applying this two algorithm, the post-processing procedures applied to refine the predicted results. Finally, those predicted binding sites are given as Motif predictions output. EMACO algorithm was compared with GAME (Genetic Algorithm for Motif Elicitation)[43] and GALF (Genetic Algorithm with Local Filtering)[39] and predicts better motifs under most circumstances. EMACO conducted experiments on eight real datasets named CREB, CRP, E2F, ERE, MEF2, MyoD, SRF, TBP which were previously constructed by the authors of GAME[39]. It has low standard deviations for prediction which indicates its stable performance[14].

2.7 An iterative algorithm for motif discovery

In 2013 iterative algorithm for motif discovery was proposed by Fan et al.[5]. This method uses the common GA framework and finds the motifs with three operations in GA and a new Addition operation proposed in this algorithm. This method contains three operators such as mutation, addition, and deletion. This method starts with short motifs whose length is three. So there are total 64 initial individuals because each site is chosen from A, C, G, T. Now the length of each individual adds one each epoch by three operators until the length of the optimal motif reaches to the standard length. Throughout the method, the population number of individuals is kept 64. The iterative algorithm is a parallel random search which is helpful to implement parallel computing to increase the computational efficiency of the method. This method also can avoid dropping into the local optimum. This algorithm uses both simulated and biological data to test the effectiveness of this algorithm. The biological data set used in this method is download from the SCPD database[40]. The iterative algorithm achieves a higher score than Gibbs Sampler, GA, and GARPS in terms of the data CRP[36].

2.8 An Ant Colony Optimization based algorithm for identifying gene regulatory elements

Liu et al.[16] proposed an Ant Colony Optimization based algorithm for identifying gene regulatory elements (ACRI) in 2013. This paper focused on specific type of motif such as de-novo motif. De-novo motif is a type of motif in which the length of the motif is predefined. This algorithm detects all possible binding sites of a transcription factor from the upstream of co-expressed genes. It takes a set of sequences and a length of the motif as input. A special digraph is created where each node except the last one represents a sequence from the set of input sequences, the last node indicates the termination point, and each edge between two nodes represents a possible starting position of a binding site in the corresponding sequence. Each ant builds a solution by traversing each node once and picking one edge between two nodes. Then the best solution is searched by various optimization. ACRI used five transcriptional factors of *Saccharomyces cerevisiae* from the uniform database SCPD[40], five transcriptional factors of *Homo Sapiens* from the uniform database JASPAR[44] and 18 gene sequences contain *E. coli* transcription factor binding sites[36] to compare the results with the

algorithms, Gibbs sampler[7], AlignACE[38], MEME[8] etc. ACRI gets a higher quality of solutions at a very high speed compared with other existing related algorithms.

2.9 An efficient ant colony algorithm for DNA motif finding

In 2015 Huan et al. proposed an efficient ant colony algorithm for DNA motif finding (ACOMotif)[6]. ACOMotif uses a simple memetic scheme and applies ACO with reinforcement search technique. It uses the same structural graph $G(V, E)$ of MFACO[15] but the heuristic information, pheromone update rule and local search technique are quite different. $G(V, E)$ has $4l$ vertices organized in four rows and l columns where l is the length of the motif. The path through starting vertex to the last vertex that is made by each ant defines the acceptable solution for the motif. Then ACOMotif applies local search for the potential motif. This method uses the hill-climbing technique for local search. Additionally, it applies relax method to find the binding site of every motif. ACOMotif used *H.sapiens* dataset[44], *E. coli* dataset[36], SCPD dataset[40], ERE and E2F to compare its efficiency with the efficiencies of MFACO (motif discovery using ant colony optimization)[15], ACRI (Ant-Colony-Regulatory-Identification)[16], EMACO (DNA Motif Discovery based on Ant Colony Optimization and Expectation Maximization)[14] and MotifSuite[17]. Where *H.sapiens* dataset contains 6, 9, and 12 sequences respectively, and each sequence contains 3001 nucleotides, *E. coli* dataset holds 18 sequences and each sequence has 105 nucleotides, both ERE and E2F have 25 sequences and each contains 200 nucleotides. The experimental results show that ACOMotif/R-ACOMotif is superior in comparison with the other algorithms.

2.10 A genetic algorithm for motif finding based on statistical significance

In 2015 a genetic algorithm for motif finding based on statistical significance was proposed by Gutierrez et al.[9]. This approach proposes a new computational technique with a genetic algorithm that uses several statistical coefficients. It represents the candidate motifs using a position in which instances is situated. The only restriction is that they are over-represented in at least a few sequences. So before starting the method, all input sequences are merged in a single supersequence. Then the supersequence is divided into subsequences of a random length disregarding the length of every sequence to generate more diverse solutions faster. Finally, the solutions are filtered and clustered to generate final solutions after applying the method for each given motif width. This method was tested with the assessment provided by the study performed by Tompa et al.[45]. This assessment contains 52 datasets of four different organisms (human, yeast, fly, and mouse) and four negative controls. This algorithm successfully predicts many of the sites with the high number of true positives both in site level and nucleotide level. The main disadvantage of this approach is the lack of a system to detect false positive. It generally detects a known motif, but with more instances than it really has[9].

3 Chemical reaction optimization

A nature-inspired metaheuristic algorithm for optimization named Chemical Reaction Optimization (CRO) was proposed by Lam and Li[30]. CRO has been successfully applied to solve many NP-hard problems and obtained better performance compared to other metaheuristic algorithms. CRO loosely couples chemical reaction with optimization that obeys two laws of thermodynamics. The first law commonly known as energy conservation rule states that total energy of a system remains constant. So, according to the first law of thermodynamics we can write,

$$\sum_{i=1}^{popSize(t)} (PE_i(t) + KE_i(t)) + Buffer(t) = C \tag{5}$$

where $PE_i(t)$ and $KE_i(t)$ denote the potential and kinetic energy of the molecule i at time t respectively, $Buffer(t)$ is the energy of the surrounding as well as the energy of the central buffer at time t, and C is a constant.

CRO is a multi-agent algorithm where the molecule is a manipulated agent having some essential attributes such as the molecular structure (z), the potential energy (PE), the kinetic energy (KE), the number of hits ($NumHit$) and other parameters. The excessive energy of a molecule means instability. An unstable molecule always tries to be stable with low energy. This phenomenon is similar to searching for the optimal point of the optimization problem. To obtain stability, molecules undergo four basic reactions named onwall ineffective collision, decomposition, inter-molecular ineffective collision, and synthesis. Here ineffective collisions mean a small change in the molecular structure that refers to local search while decomposition and synthesis mean a massive change in the molecular structure that refers to global search. As CRO follows the energy conservation rule so, any of the reactions will only take place when the following equation is satisfied:

$$\sum_{i=1}^{t}(PE_{zi} + KE_{zi}) \geq \sum_{i=1}^{s}(PE_{z'i}) \tag{6}$$

where t is the number of reactants, s is the number of products, Z and Z' are the structures of the molecule before and after the reaction.

Table 1 Various parameters of CRO and their algorithmic definitions

Symbol	Algorithmic definition
PopSize	The number of solutions (solution space)
KELossRate	The loss rate of kinetic energy in an elementary reaction
MoleColl	Makes a decision whether a uni-molecular or an inter-molecular collision will occur
Buffer	The initial energy of the system
α and β	Make a decision whether an effective or an ineffective collision will occur
NumHit	The total number of hits a molecule has experienced
Minstruct	The structure of a solution having minimum potential energy
MinPE	The potential energy of a solution having minimum structure
MinHit	The number of hits when a molecule experiences a minimum structure

3.1 Parameters of CRO

In CRO, molecules are the manipulated agents having some attributes. Table 1 lists the attributes and their algorithmic definitions.

3.2 Operator selection

This section describes the basic scheme of CRO and operator selection. Figure 2 shows a flowchart of CRO to depict the whole process. The process starts with the initialization stage. In this stage, the number of populations and the other parameters are initialized. Then the iteration stage starts and a number of iterations are performed. In each iteration, one of the elementary reaction happens and the required number of molecules are selected from the population randomly. At first of each iteration, a random number v between 0 and 1 is generated to take the decision, if uni-molecular or inter-molecular collision will occur. If $v > MoleColl$ or only one molecule remains, then the uni-molecular collision occurs else the inter-molecular collision takes place. Then for this collision, a definite number of molecules are selected from the population randomly. Now for uni-molecular collision (left side of the flowchart), a condition is checked with a parameter α if the onwall ineffective or decomposition reaction will occur. Similarly, for inter-molecular collision (right side of the flowchart), a condition is checked with a parameter β if the inter-molecular ineffective or decomposition reaction will occur. The value of the parameters $MoleColl, \alpha, \beta$ are assigned at the initialization state. After each elementary reaction, if any best solution is found, it is saved. The iteration stage continues until any stopping criterion is met. In the final stage, a global best solution is found. The operator repair1 is applied to the final solution to search for the better solution. Then the binding sites of the better solution are located. At last, operator repair2 is applied to improve the quality of the binding sites.

4 CRO for DNA motif discovery problem

In this paper, we solve the DNA Motif Discovery problem using a well-known population-based metaheuristic algorithm, Chemical Reaction Optimization (CRO). CRO is an algorithmic framework that can solve optimization problems efficiently. It is a variable population based algorithm that means there are different numbers of molecules in different iterations. Here we have proposed an algorithm to find the DNA motif using four basic operators of CRO. The operators are redesigned and an additional operator (repair operator) is designed to find out the best solutions. Another repair function is used to find the better binding sites that give a better result. The proposed algorithm is named here as DMD_CRO (DNA Motif Discovery using CRO). For

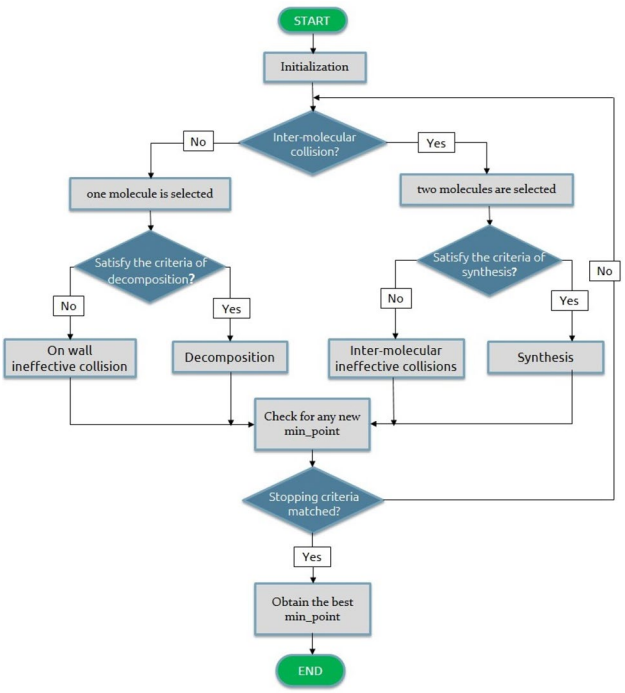


Fig. 2 A flowchart of CRO

implementation, the code of the proposed DMD_CRO can be found here[1].

4.1 Basic structure of DMD_CRO

Our proposed DMD_CRO algorithm has a difference from the basic CRO algorithm. The difference is after the iteration stage when we find the final solution, two additional repair operators are applied to this final solution to get better potential motif and binding sites. Algorithm 1 shows the pseudo code of DMD_CRO.

Algorithm 1 DMD_CRO

Set the parameters $popSize$, $KELossRate$, $MoleColl$, $Buffer$, $InitialKE$, $iteration$, $NumHit$, $MinHit$. α, and β;
Initialize $popSize$ number of solutions randomly as initial population;
for ($int\ i \leftarrow 1\ to\ iteration$) **do**
  Randomly generate k between 0 and 1;
  // condition check for uni-molecular reaction
  if $k > Molecoll$ **then**
    Randomly choose one solution S_m;
    // condition check whether decomposition occurs or not
    if $NumHit_{S_m} - MinHit_{S_m} > \alpha$ **then**
      Call Decomposition();
    else
      Call On-wall ineffective collision();
      $NumHit_{S_m} \leftarrow NumHit_{S_m} + 1$; // increase the value of $NumHit_{S_m}$ by one
      $InitialKE_{S_m} \leftarrow InitialKE_{S_m} - (InitialKE_{S_m} * KELossRate)$; // update initial kinetic energy
    end if
  else
    Randomly choose two solutions S_{m1} and S_{m2};
    // condition check whether synthesis occurs or not
    if $(InitialKE_{S_{m1}} + InitialKE_{S_{m2}}) \leq \alpha$ **then**
      Call Synthesis();
    else
      Call Inter-molecular ineffective collision();
      $NumHit_{S_{m1}} \leftarrow NumHit_{S_{m1}} + 1$; // increase the value of $NumHit_{S_m}$ by one
      $NumHit_{S_{m2}} \leftarrow NumHit_{S_{m2}} + 1$; // increase the value of $NumHit_{S_m}$ by one
      $InitialKE_{S_{m1}} \leftarrow InitialKE_{S_{m1}} - (InitialKE_{S_{m1}} * KELossRate)$; // update initial kinetic energy of $InitialKE_{S_{m1}}$
      $InitialKE_{S_{m2}} \leftarrow InitialKE_{S_{m2}} - (InitialKE_{S_{m2}} * KELossRate)$; // update initial kinetic energy of $InitialKE_{S_{m2}}$
    end if
  end if
  Searching for any new $best\ solution\ S_b$ using the $objective\ function$;
end for
Apply $Repair1\ operator$ on S_b to get the better solution;
Locate binding sites for the better solution;
Apply $Repair2\ operator$ on binding sites to get better binding sites;
Output: $Overall\ best\ solution$, $binding\ sites\ of\ best\ solution$, $and\ the\ value\ of\ the\ objective\ function$.

4.2 Solution representation and population generation

Let, for a given set of N DNA sequences $S = \{S_1, S_2, \ldots, S_N\}$, we have to find a motif where the length of the motif, $l = 7$. So there are $4^7 = 16384$ possible patterns as each site is selected from $\sigma = \{A, C, G, T\}$. We generate patterns randomly from all possible patterns and calculate their information contents according to Eq. 4. Then from the 100 patterns, 20 patterns with higher information contents are taken to use for exploring the solution space. When population generation is completed, each symbol $\{A, C, G, T\}$ of each possible pattern is encoded by a unique numerical value. We have used 0, 1, 2, 3 for the symbols A, C, T, G respectively. Figure 3 shows an example of solution representation.

4.3 Reaction operators

For the DMD_CRO algorithm, we have represented four reaction operators to find out the solutions and introduced two additional operators to get better results. The following subsections describe the operators used in the algorithm.

4.3.1 On-wall ineffective collision

This molecular reaction is used to search for the neighborhood solution (local search). We use the one-difference operator as shown in Fig. 4 for this elementary reaction. A position is chosen randomly in the molecule to change the value of this position. Let S_m is a molecule to which the on-wall ineffective collision is applied. The values of the solution S_m are copied to a solution S'_m. A position i of the molecule S_m is randomly selected where $1 \leq i \leq l$ (length of the motif). Next, the value of ith position of S'_m has to be changed. For this, we generate a value $r \epsilon \{0, 1, 2, 3\}$ such that $r \neq S_m[i]$ and put the value of r in the ith position of S'_m. Thus a new solution S'_m is created. In Fig. 4, $i = 3$ and the value $S_m[3] = 2$. Now we randomly generate a value r between 0 and 3 such that $r \neq 2$, so $r = 3$ is selected and put it in $S'_m[i]$. Algorithm 2 shows the pseudo code of On-wall ineffective collision.

Algorithm 2 On-wall ineffective collision

Input: Solution $S_m[1, 2, ..., l]$. // l is the length of the motif
Copy solution S_m to form S'_m; // $S_m \leftarrow S'_m$
Randomly generate an integer i between 1 and l; // Choose a position
Randomly generate another integer r between 0 and 3 such that $r \neq S_m[i]$; // generates another nucleotide other than the present nucleotide
$S'_m[i] \leftarrow r$; // generate new solution
Output: Solution S'_m.

[1] https://drive.google.com/open?id=1cEFVklnTfc5QZMxtSPhLSPFN25nJwm6K

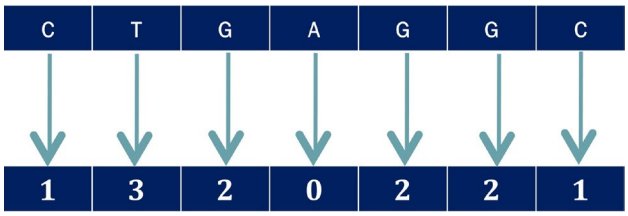


Fig. 3 Solution representation

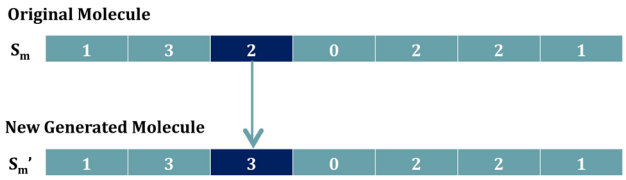


Fig. 4 On-wall ineffective

4.3.2 Decomposition

This reaction is implemented to enable the algorithm for exploring the other region of solution space (global search). Here we have used a popular half-total exchange operator as decomposition shown in Fig. 5. In decomposition, two new molecules are generated from an original molecule. Let S_d is an original molecule to which we apply this reaction. At first, the molecule S_d is divided into two parts. Then we copy values of the first part of S_d to a new molecule S_{d1} and randomly generate values of the remaining part of S_{d1}. Similarly, the values of the last part of S_d are being copied to the respective part of another new molecule S_{d2} and randomly generate values of the remaining part of S_{d2}. Algorithm 3 depicts the pseudo code of Decomposition.

Algorithm 3 Decomposition

1: **Input:** Solution $S_d[1, 2, ..., l]$. // l is the length of the motif
2: Set $b \leftarrow \frac{l}{2}$; // we mean integer division
3: Copy $[1, 2, .., b]$ values from S_d to form S_{d1};
4: Copy $[(b + 1), .., l]$ values from S_d to form S_{d2};
5: Randomly generate $[(b + 1), .., l]$ values between 0 and 3 and put them to S_{d1};
6: Randomly generate $[1, 2, .., b]$ values between 0 and 3 and put them to S_{d2};
7: **Output:** Solutions S_{d1} and S_{d2}.

4.3.3 Inter-molecular Ineffective Collision

In this elementary reaction, a well-known two-point crossover operator is used as shown in Fig. 6. Two molecules S_{c1} and S_{c2} are randomly selected from the solution space. Then

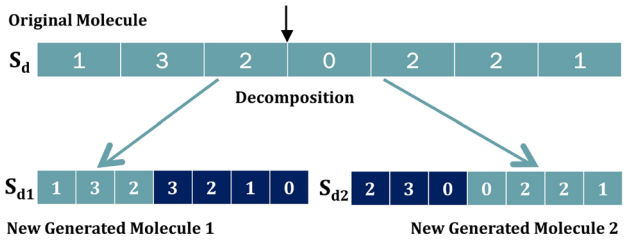


Fig. 5 Decomposition reaction

two points p_1 and p_1 from the molecule are randomly chosen where $p_1 < p_2$. Now we divide both molecules S_{c1} and S_{c2} into three parts with these two points. Then the values from the first and third parts of S_{c1} are copied to the respective positions of a new molecule S_{n1}. The values of the second part of S_{c2} are being copied to the respective positions of the new molecule S_{n1}. Similarly, another new solution S_{n2} is created from the first and third parts of S_{c2} along with the second part of S_{c1}. Algorithm 4 gives the pseudo code of inter-molecular ineffective collision.

Algorithm 4 Inter-molecular ineffective collision

1: **Input:** Two solutions $S_{c1}[1, 2, ..., l]$ and $S_{c2}[1, 2, ..., l]$. // l is the length of the motif
2: Randomly choose two positions p_1 and p_2 between 1 and l such that $p_1 < p_2$;
3: Copy solution S_{c1} to form S_{n1};
4: Copy solution S_{c2} to form S_{n2};
// generate two new solutions by exchanging their positional values
5: **for** $(int\ i \leftarrow p_1\ to\ p_2)$ **do**
6: $S_{n1}[i] \leftarrow S_{c2}[i]$;
7: $S_{n2}[i] \leftarrow S_{c1}[i]$;
8: **end for**
9: **Output:** Solutions S_{n1} and S_{n2}.

4.3.4 Synthesis

The probabilistic select operator depicted in Fig. 7 is used for this elementary reaction[26]. Synthesis takes two molecules S_{m1} and S_{m2} randomly from the solution space and produces a new molecule S'_m. This reaction is the opposite of the decomposition operator. At first, the frequency of each symbol $\{A, C, G, T\}$ for both S_{m1} and S_{m2} are calculated and the values of the frequencies are put in two different arrays. Then to find a proper symbol for the ith position of S'_m, we compare the frequency of the ith symbol of S_{m1} with the frequency of the ith symbol of S_{m2} and take the symbol with the highest frequency as the value of the ith position of S'_m. Now the frequency of the selected symbol for the molecule is decreased by one from the solution array. This procedure

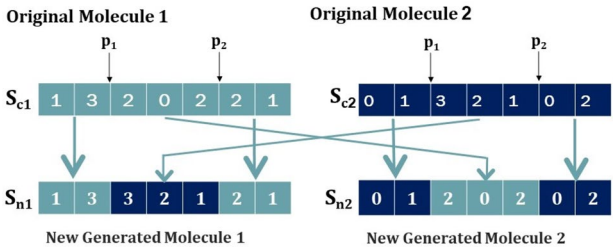


Fig. 6 Inter-molecular ineffective reaction

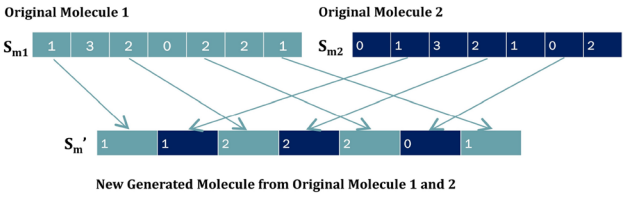


Fig. 7 Synthesis reaction

repeats for selecting every symbol. Algorithm 5 shows the pseudo code of synthesis.

Algorithm 5 Synthesis

Input: Two solutions $S_{m1}[1, 2, ..., l]$ and $S_{m2}[1, 2, ..., l]$. // l is the length of the motif
Calculate the frequencies of $0, 1, 2, 3$ used in $S_{m1}$and put in $array1[0, 1, 2, 3]$;
Calculate the frequencies of $0, 1, 2, 3$ used in $S_{m2}$and put in $array2[0, 1, 2, 3]$; //
// take the symbol with the highest frequency as the value of the i^{th} position of $S_m^{'}$
for $(int\ i \leftarrow p_1\ to\ p_2)$ **do**
  if $array1[S_{m1}[i]] > array2[S_{m2}[i]]$ **then**
    $S_m^{'}[i] \leftarrow S_{m1}[i]$;
    $array1[S_{m1}[i]] \leftarrow array1[S_{m1}[i]] - 1$; // decreased the frequency of selected symbol
  else
    $S_m^{'}[i] \leftarrow S_{m2}[i]$;
    $array2[S_{m2}[i]] \leftarrow array1[S_{m2}[i]] - 1$; // decreased the frequency of selected symbol
  end if
end for
Output: Solution $S_m^{'}$.

4.3.5 Operator Repair1

The operator repair1 is applied to the final solution S_m to improve the result by the local search to get potential motif. At first, we copy the values of S_m to form a new solution S'_m. Now the value of the first position of S'_m has been changed by one of $\{0, 1, 2, 3\}$ such that $S_m[0] \neq S'_m[0]$. Compute the information content $T = IC(S_m)$ and $T' = IC(S'_m)$ using Eq. 4. If T' is not greater than T, then we change the value of the second position of S'_m and do the same again. But if T' is greater than T, then S_m is updated by S'_m and again the technique is applied to the updated solution S_m. The operator repair1 is stopped when we do not get a better result by checking all the positions of S_m and output the updated best solution S_m. Algorithm 6 gives the pseudo code of the process.

Algorithm 6 Repair1

Input: The final solution $S_m[1, 2, ..., l]$. // l is the length of the motif
for $(int\ i \leftarrow 1\ to\ l)$ **do**
  Copy the values of S_m to the new solution $S_m^{'}$; // $S_m^{'} \leftarrow S_m$
  for $(int\ v \leftarrow 1\ to\ 3$ such that $v \neq S_m[i])$ **do**
    Copy v to $S_m^{'}[i]$; // $S_m^{'}[i] \leftarrow v$
    Compute information content $T = IC(S_m)$ and $T^{'} = IC(S_m^{'})$ using equation 4;
    if $T < T^{'}$ **then**
      $S_m \leftarrow S_m^{'}$; // update the solution
      go to step: 2;
    end if
  end for
end for
Output: The new best solution S_m.

Figure 8 shows an example of the operator repair1. Here an initial solution S_m is taken with information content 11.78. Now we have changed the value of the 1^{st} position of S_m using $\{0, 1, 2, 3\}$ to get three new solutions S_{m11}, S_{m12} and S_{m13} such that $S_{m11} \neq S_{m12} \neq S_{m13}$. Next, the information contents of S_{m11}, S_{m12} and S_{m13} are computed. But a larger information content than the initial solution has not been found. Now the value of the 2^{nd} position of S_m is changed to get S_{m21}, S_{m22} and S_{m23} similarly. But still, a larger information content has not been obtained. Next, the value of the 3^{rd} position has been changed and got a solution S_{m32} with larger information content 12.17. So the solution S_m is updated by S_{m32}. At this moment, we have to reapply the repair1 operator to this updated solution. The repair1 operator searches all the neighboring solutions of the existing solution to get a better one. If any better solution is found then the existing solution is replaced by this better solution and we repeat the process. This process continues until all the neighboring solutions are worse than the existing solution. But the CRO operators search one or two local or global solution(s) of the existing solution(s). Since the searching space by repair operator is very large compared to the traditional CRO operators. So this additional operator helps the proposed DMD_CRO algorithm to search the solution space efficiently in finding better solutions. That is why the possibility to find better solutions by the CRO with the operator repair1 is more than the CRO without this operator.

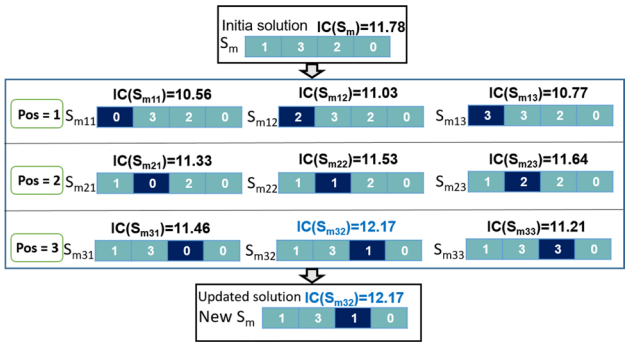


Fig. 8 Repair1 operator

4.3.6 Locate binding sites

To get the binding sites for a solution S_m, we have to find a position p_i for each input sequence S_i which can minimize the hamming distance in Eq. 2. Thus a set of positions $P = \{p_1, p_2, \ldots, p_N\}$ is found which is known as the binding sites for the solution S_m.

4.3.7 Operator repair2

We have used operator repair2 (repair function), which is a modified version of a local optimization technique for subsequence tuple in ACRI[16]. This operator is applied to the binding sites $P = \{p_1, p_2, \ldots, p_N\}$ to find better binding sites. The value of information contents for existing binding sites P is calculated using Eq. 4. At first, each position in P is changed to get six new binding sites $P_k = \{p_1 + k, p_2 + k, \ldots, p_N + k\}$ where $-3 \leq k \leq 3$ and $k \neq 0$. Again the value of information content of P_k for each value of k is calculated using Eq. 4. Thus we get new six information contents for six values of k. At last, we have to find out the binding sites having the highest information content among the six information contents and old ones. Algorithm 7 gives the pseudo code of the process.

Algorithm 7 Repair2

1: **Input:** The binding sites $P = \{p_1, p_{12}, ..., p_N\}$. // N is the number of sequences
2: **for** $(int\ k \leftarrow -3\ to\ 3)$ such that $k \neq 0$ **do**
3: Add k with each position of P to get new binding sites $P^{'}$;
4: Compute the value of information content $IC^{'}$ for $P^{'}$ using equation 4;
5: **if** $IC^{'} > IC_{best}$ **then**
6: Copy the positions of $P^{'}$ to P_{best}; // $P_{best} \leftarrow P^{'}$;
7: $IC_{best} \leftarrow IC^{'}$; // update the information content value
8: **end if**
9: **end for**
10: **Output:** The best binding sites P_{best} and the value of information content IC_{best}.

Figure 9 shows an example of the process of the repair2 operator. Here $P = \{59, 53, \ldots 76\}$ are initial binding sites having information content, $IC = 11.923$. Next, we get $P = \{56, 50, \ldots 73\}$ by adding $k = 3$ in each position of P and compute the information content value, $IC = 10.568$ for $P^{'}$. Similarly, the binding sites and information content for each value of k are computed. From Fig. 9, the highest information content value $IC = 13.091$ is found with binding sites $P = \{61, 55, \ldots 78\}$ for $k = 2$. So $P = \{61, 55, \ldots 78\}$ and $IC = 13.091$ are the final outputs of the operator repair2.

5 Experimental results and analysis

The proposed DMD_CRO algorithm was tested with several datasets given in ACRI[16] for evaluation purpose. We implemented our algorithm in C# programming language using Microsoft Visual C# 2013 and executed using an Intel Core i5 computer with 2.50 GHz CPU and 4 GB RAM under Windows 10 operating system (64 bit). For an effective test, we compared the results of DMD_CRO with Gibbs sampler[7], AlignACE[38], MEME[8] and ACRI[16]. The datasets used in the experiments contain five transcriptional factors of *Homo sapiens*, 18 gene sequences contain *E. coli* transcription factor binding sites[36] and RAP1 of *Saccharomyces cerevisiae* from SCPD[40]. The ACRI algorithm solved the de-novo motif discovery problem. We have also designed DMD_CRO to solve the same type of problem. De-novo motif is a type of motif in which the length of the motif is predefined.

5.1 Experimental setup

In the proposed DMD_CRO algorithm, there are some key parameters. We investigated for the best value by testing over the 18 gene sequences of *E. coli* transcription factor binding sites dataset for these key parameters. The tunning process was demonstrated in Fig. 10 for α, β, *iteration*, and *KELossRate*. In the first row and first column of Fig 10, a line graph has been drawn to show the effect of the value of α over the value of information content (used in Eq. 4). Here α has been plotted in the x-axis and information content (IC) has been plotted in the y-axis. From the graph, it can be seen that the highest value of *IC* is obtained for $\alpha = 1$. Similarly, we get highest values of *IC* for $\beta = 350$, *iteration* $= 2000$, and *KELossRate* $= 0.2$ respectively.

Besides these parameters, several parameters named *popSize*, *MoleColl*, *InitialKE* were used in the experiment. Table 2 shows all parameters and their respective values. The termination condition of the proposed algorithm was set upon the value of these parameters.

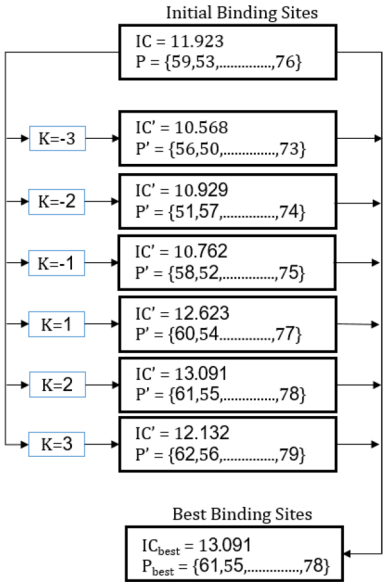


Fig. 9 Repair2 operator

5.2 Analysis for transcription factor binding sites of *Homo sapiens*

The experiments of the proposed DMD_CRO algorithm were performed using *Homo sapiens* for transcription factor binding sites from the uniform database JASPAR. We selected five transcriptional factor binding sites as tested data. Table 3 shows the dataset (also used in ACRI).

The dataset was tested using our proposed algorithm and ACRI and created weblogo using the http://weblogo.berkeley.edu/logo.cgi website. Table 4 shows the results. The second and third columns show generated weblogo about its corresponding sequence using DMD_CRO and ACRI respectively. The weblogos of DMD_CRO and ACRI are similar to the real weblogos. These point out the effectiveness of our proposed DMD_CRO algorithm that means DMD_CRO algorithm is correct. We did this experiment to prove the effectiveness and correctness of our algorithm. In Tables 3 and 4, TF means sequence name.

5.3 Analysis of CRP binding sites of *E. coli*

Another benchmark dataset for identifying the regulatory elements is the CRP binding sites of *E. coli*. In this dataset, there are 18 sequences having a length of 105 for each sequence. Table 5 shows the 18 sequences of the CRP binding sites for *E. coli*.

To find the motif starting positions from these sequences, we used Information Content as objective function stated in Eq. 4. Like most of the popular computing methods, we set the *motiflength* = 22. We have executed DMD_CRO algorithm five times using the same parameter settings as shown in Table 2. In Table 6, the worst and best-found motif starting position for each sequence of five consecutive runs for both without and with repair operator is shown.

Now, Table 7 shows the experimental results of our proposed DMD_CRO algorithm in comparison with MEME[8], ACRI[16], Gibbs sampler[7], and AlignACE[38] using the best found motif starting positions of the CRP binding sites of *E. coli*. The found binding sites are acceptable if the difference between the actual position and detecting position is 10[16]. In Table 7 binding sites column denotes the actual position of the motif in the sequence. The found binding sites of MEME, ACRI, Gibbs sampler, and AlignACE were taken from ACRI[16] paper. MEME, ACRI, Gibbs sampler,

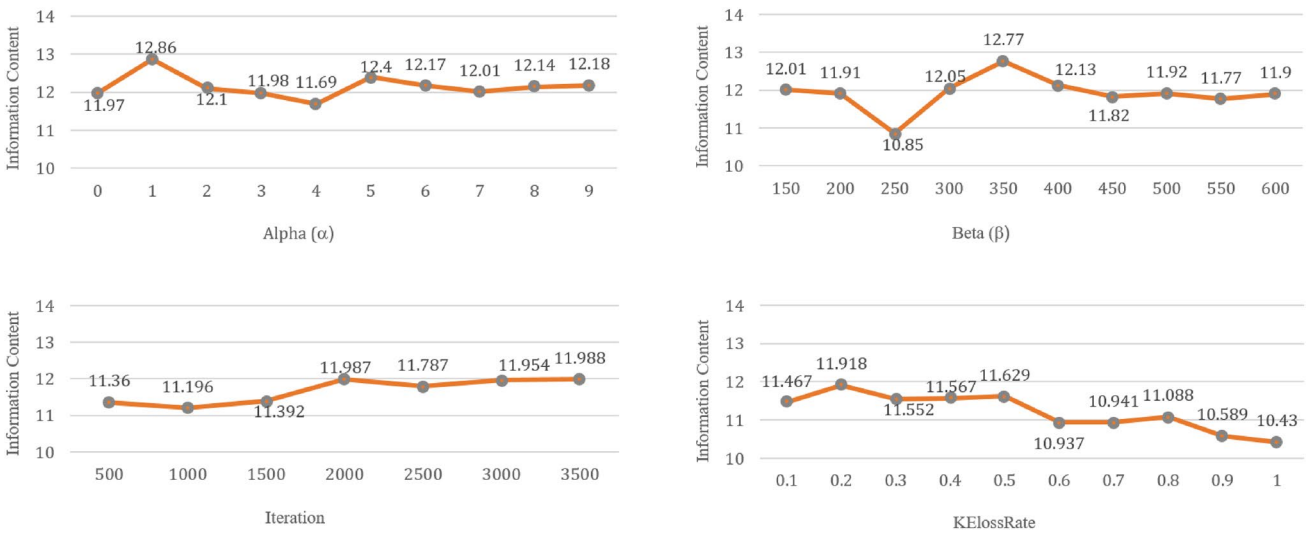


Fig. 10 Parameters tuning of CRO algorithm

Table 2 Parameters of CRO algorithm for finding motif

Symbol	Value
popSize	200
KELossRate	0.2
MoleColl	0.3
iteration	2000
InitialKE	500
α	1
β	350

Table 3 The five transcriptional factors of Homo sapiens

TF	Length	Consensus sequence
ELK4	9	ACCGGAAGT
E2F1	8	TTTGGCGC
FOXD1	8	GTAAACAT
USF1	7	CACGTGG
RELA	10	GGGAATTTCC

and AlignACE columns show motif positions found using them. In DMD_CRO column, we show the best motif starting position for each sequence from Table 6. The positional difference between actual position and position found using respective algorithms have been shown in error columns. Here DMD_CRO (without repair) gives all binding sites successfully but some of the results have to be improved. The DMD_CRO algorithm with repair shows all the binding sites successfully and the results are better than those of MEME and ACRI algorithms. So the better results than the other related algorithms were obtained by the proposed algorithm with repair operators.

In Tables 8 and 9, we compared the information content values of DMD_CRO with other four algorithms: Gibbs sampler, MEME, AlignACE, and ACRI. Values of information content (IC) were calculated using Eq. 4. We executed DMD_CRO algorithm 18 times to get the information content distributions by DMD_CRO. The information content distributions of other algorithms were taken directly from ACRI[16]. Here Table 8 depicts the information content distributions and Table 9 represents the worst, average, and best information content values of the respective algorithms. The higher information content value denotes a better solution. From Tables 8 and 9, it is clear that the quality of the solutions found by DMD_CRO (with and without repair operator) is higher than the other algorithms.

5.4 Statistical significance test

The previous subsections express that the performance of the DMD_CRO algorithm is better than the other traditional algorithms in terms of the quality of the results. In this subsection, we examine whether there is statistical significance between DMD_CRO and other traditional algorithms. The Student's t-test and the Mann-Whitney U test were used for this purpose.

5.4.1 Comparison using student's t-test

The information content values of Table 8 were used to calculate the t-values using Eq. 6.

$$t\text{-}value = \frac{|\overline{V_1} - \overline{V_2}|}{\sqrt{\frac{{\sigma_1}^2}{n_1} + \frac{{\sigma_2}^2}{n_2}}} \tag{7}$$

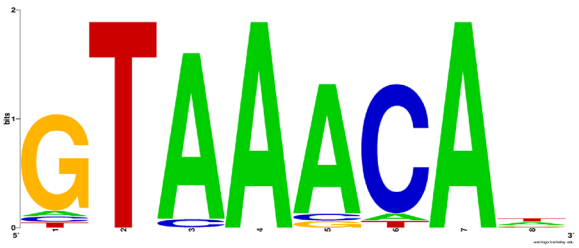
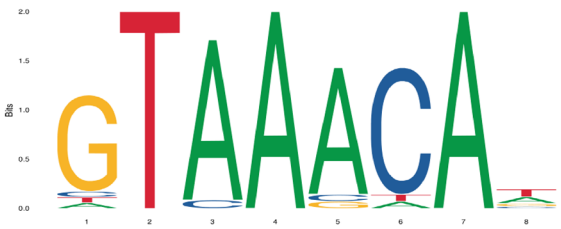
Where $\overline{V_1}$, $\overline{V_2}$ are the average information content values, σ_1, σ_2 are the standard deviations, and n_1, n_2 are the numbers of samples for group 1 and group 2 respectively. Each group has 18 samples, so the degree of freedom is $(18 + 18 - 2) = 34$. The significance level $\alpha = 0.05$ was chosen to get the critical value $t_{crit.} = 2.032$ at 34 degrees of freedom from the t-distribution table. We set the null hypothesis that there is no statistically significant difference between DMD_CRO with other algorithms. If $t\text{-}value > t_{crit.}$ or $t\text{-}value < -t_{crit.}$, then the null hypothesis can be rejected and decided that there is a statistically significant difference between DMD_CRO with other algorithms. Table 10 shows the *t-values* of different algorithms compared with DMD_CRO. Using the data of Table 8, for DMD_CRO (without repair), the average information content value, $\overline{V_1} = 11.24$, standard deviation, $\sigma_1 = 0.985$, and the number of samples, $n_1 = 18$ and for Gibbs sampler, $\overline{V_2} = 9.229$, $\sigma_2 = 0.124$, and $n_2 = 18$. Now, using Eq. 6 $t\text{-}value = 8.58$ for Gibbs sampler compared with DMD_CRO (without repair), which is shown in the first row and first column in Table 10. Similarly, the other values were calculated. In Table 10, all the *t-values* are greater than the $t_{crit.} = 2.032$. So we can reject the null hypothesis and conclude that DMD_CRO has statistical significant difference compared with the other related algorithms.

5.4.2 Comparison using Mann-Whitney U Test

The two-tailed Mann–Whitney *U* test was used to compare DMD_CRO with other algorithms. We have considered the significance level $\alpha = 0.05$ to get the critical value $Z_{crit.} = 1.96$. Then $Z_{stat.}$ was calculated from the information content values of Table 8 using Eq. 7.

$$Z_{stat.} = \frac{U - \frac{n_1 n_2}{2}}{\sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}} \tag{8}$$

Table 4 The experimental results for five transcriptional factors of *Homo sapiens* of DMD_CRO and ACRI

TF	DMD_CRO	ACRI [16]
ELK4		
E2F1		
FOXD1		
USF1		
RELA		

where U denotes the lowest sum between the positive and negative ranks of the information contents of DMD_CRO and any other algorithm, and n_1, n_2 are the numbers of samples for these two algorithms. The null hypothesis states that there is no statistical significance between DMD_CRO with other algorithms. The alternative hypothesis defines that there is statistically significant. If $Z_{stat.} > Z_{crit.}$ or $Z_{stat.} < -Z_{crit.}$, then we can reject the null hypothesis and accept the alternative hypothesis. Table 11 shows the calculated values of U and $Z_{stat.}$ of different algorithms compared

Table 5 The 18 sequences of the CRP binding sites for *E. coli*

Sequence no.	Sequence
1	TAATGTTTGTGCTGGTTTTTGTGGCATCGGGCGAGAATAGCGCGTGGTGTGAA
	AGACTGTTTTTTTGATCGTTTTCACAAAAATGGAAGTCCACAGTCTTGACAG
2	GACAAAAACGCGTAACAAAAGTGTCTATAATCACGGCAGAAAAGTCCACATTG
	ATTATTTGCACGGCGTCACACTTTGCTATGCCATAGCATTTTTATCCATAAG
3	ACAAATCCCAATAACTTAATTATTGGGATTTGTTATATATAACTTTATAAATT
	CCTAAAATTACACAAAGTTAATAACTGTGAGCATGGTCATATTTTTATCAAT
4	CACAAAGCGAAAGCTATGCTAAAACAGTCAGGATGCTACAGTAATACATTGAT
	GTACTGCATGTATGCAAAGGACGTCACATTACCGTGCAGTACAGTTGATAGC
5	ACGGTGCTACACTTGTATGTAGCGCATCTTTCTTTACGGTCAATCAGCAAGGT
	GTTAAATTGATCACGTTTTAGACCATTTTTTCGTCGTGAAACTAAAAAAACC
6	AGTGAATTATTTGAACCAGATCGCATTACAGTGATGCAAACTTGTAAGTAGAT
	TTCCTTAATTGTGATGTGTATCGAAGTGTGTTGCGGAGTAGATGTTAGAATA
7	GCGCATAAAAAACGGCTAAATTCTTGTGTAAACGATTCCACTAATTTATTCCA
	TGTCACACTTTTCGCATCTTTGTTATGCTATGGTTATTTCATACCATAAGCC
8	GCTCCGGCGGGGTTTTTTGTTATCTGCAATTCAGTACAAAACGTGATCAACCC
	CTCAATTTTCCCTTTGCTGAAAAATTTTCCATTGTCTCCCCTGTAAAGCTGT
9	AACGCAATTAATGTGAGTTAGCTCACTCATTAGGCACCCCAGGCTTTACACTT
	TATGCTTCCGGCTCGTATGTTGTGTGGAATTGTGAGCGGATAACAATTTCAC
10	ACATTACCGCCAATTCTGTAACAGAGATCACACAAAGCGACGGTGGGGCGTA
	GGGGCAAGGAGGATGGAAAGAGGTTGCCGTATAAAGAAACTAGAGTCCGTTTA
11	GGAGGAGGCGGGAGGATGAGAACACGGCTTCTGTGAACTAAACCGAGGTCAT
	GTAAGGAATTTCGTGATGTTGCTTGCAAAAATCGTGGCGATTTTATGTGCGCA
12	GATCAGCGTCGTTTTAGGTGAGTTGTTAATAAAGATTTGGAATTGTGACACA
	GTGCAAATTCAGACACATAAAAAAACGTCATCGCTTGCATTAGAAAGGTTTCT
13	GCTGACAAAAAAGATTAAACATACCTTATACAAGACTTTTTTTTCATATGCC
	TGACGGAGTTCACACTTGTAAGTTTTCAACTACGTTGTAGACTTTACATCGCC
14	TTTTTTAAACATTAAAATTCTTACGTAATTTATAATCTTTAAAAAAAGCATT
	TAATATTGCTCCCCGAACGATTGTGATTCGATTCACATTTAAACAATTTCAGA
15	CCCATGAGAGTGAAATTGTTGTGATGTGGTTAACCCAATTAGAATTCGGGAT
	TGACATGTCTTACCAAAAGGTAGAACTTATACGCCATCTCATCCGATGCAAGC
16	CTGGCTTAACTATGCGGCATCAGAGCAGATTGTACTGAGAGTGCACCATATG
	CGGTGTGAAATACCGCACAGATGCGTAAGGAGAAAATACCGCATCAGGCGCTC
17	CTGTGACGGAAGATCACTTCGCAGAATAAATAAATCCTGGTGTCCCTGTTGA
	TACCGGGAAGCCCTGGGCCAACTTTTGGCGAAAATGAGACGTTGATCGGCACG
18	GATTTTTATACTTTAACTTGTTGATATTTAAAGGTATTTAATTGTAATAACG
	ATACTCTGGAAAGTATTGAAAGTTAATTTGTGAGTGGTCGCACATATCCTGTT

with DMD_CRO. Using the data of Table 8, the number of samples, $n_1 = 18$ for DMD_CRO (without repair) and $n_2 = 18$ for Gibbs sampler. The lowest sum, $U = 2$ for Gibbs sampler. Now using Eq. 7, we get $Z_{stat.} = -5.046$ for Gibbs sampler compared with DMD_CRO (without repair), which is shown in the first row and first column in Table 11. Similarly, the other values were calculated. In Table 11, all the $Z_{stat.}$ values are lower than $Z_{stat.} = -5.046$. So the null hypothesis can be rejected and concluded that DMD_CRO has statistical significant difference compared with the other algorithms.

These two significance tests prove the superiority of DMD_CRO algorithm over other state-of-the-arts in this area.

5.5 Running time analysis

For the analysis of running time, we implemented the ACRI algorithm in our experimental platform. For this testing purpose, five ants were used for ACRI and initial population size was fixed to five for DMD_CRO. We set a value to the *iteration* parameter and executed each algorithm five times. So five running times were found for each algorithm. Now

Table 6 The worst and best motif starting positions of the CRP binding sites of *E. coli* for DMD_CRO

No.	Binding sites	DMD_CRO (without repair)		DMD_CRO (With repair)	
		Worst position	Best position	Worst position	Best position
1	17,61	36	61	36	61
2	17,55	31	55	58	55
3	76	19	76	19	76
4	63	21	61	43	61
5	50	24	50	83	50
6	7,60	63	4	40	7,60
7	42	60	38	24	44
8	39	74	35	74	41
9	9,80	71	9	12	80
10	14	70	14	17	14
11	61	34	65	29	61
12	41	32	41	32	41
13	48	64	48	64	48
14	71	74	68	74	71
15	17	75	17	29	17
16	53	84	53	44	53
17	1,84	41	80	34	83
18	78	7	78	74	78

Table 7 Comparison of the results of DMD_CRO with MEME, and ACRI for the 18 sequences of the CRP binding sites for *E. coli*

No.	Binding sites	Gibbs sampler[7]	Error	AlignACE[38]	Error	MEME[8]	Error	ACRI[16]	Error	DMD_CRO (without repair)	Error	DMD_CRO (with repair)	Error
1	17,61	59	− 2	63	2	61	0	63	2	61	0	61	0
2	17,55	53	− 2	57	2	55	0	57	2	55	0	55	0
3	76	74	− 2	78	2	76	0	78	2	76	0	76	0
4	63	59	− 4	65	2	63	0	65	2	61	2	61	2
5	50	11	− 39	52	2	13	− 37	52	2	50	0	50	0
6	7,60	5	− 2	9	2	7	0	9	2	4	3	7,60	0
7	42	40	− 2	26	− 16	42	0	44	2	38	4	44	2
8	39	37	− 2	41	2	39	0	41	2	35	4	41	2
9	9,80	7	− 2	11	2	9	0	11	2	9	0	80	0
10	14	12	− 2	16	2	14	0	16	2	14	0	14	0
11	61	59	− 2	63	2	35	− 16	63	2	65	4	61	0
12	41	47	6	43	2	34	− 7	43	2	41	0	41	0
13	48	46	− 2	50	2	48	0	50	2	48	0	48	0
14	71	69	− 2	73	2	71	0	73	2	68	3	71	0
15	17	15	− 2	19	2	75	58	19	2	17	0	17	0
16	53	43	− 4	55	2	6	− 47	55	2	53	0	53	0
17	1,84	25	24	68	− 16	27	26	95	4	80	4	83	1
18	78	74	− 4	80	2	16	− 2	78	0	78	0	78	0

Table 8 Comparison of the information content distribution of the results by different algorithms

No.	Gibbs sampler[7]	MEME[8]	AlignACE[38]	ACRI[16]	DMD_CRO (without repair)	DMD_CRO (with repair)
1	9.412	10.032	9.651	10.01	10.833	**13.743**
2	9.175	9.075	9.887	10.28	10.995	**11.377**
3	9.324	10.02	9.576	9.987	12.38	**13.582**
4	9.123	10.05	9.624	10.403	**11.777**	**11.777**
5	9.006	9.117	10.235	10.457	11.67	**12.672**
6	9.4	9.892	9.71	10.184	10.855	**12.735**
7	9.207	9.554	9.010	9.895	10.818	**14.420**
8	9.312	10.124	9.934	10.258	12.442	**13.363**
9	9.246	9.646	9.807	10.354	11.364	**14.551**
10	9.203	9.439	9.853	10.421	11.135	**12.541**
11	9.311	9.121	10.12	10.53	12.184	**12.808**
12	9.029	9.16	9.399	10.415	10.876	**13.266**
13	9.345	9.684	9.976	10.38	11.028	**13.414**
14	9.280	9.773	9.825	10.286	10.959	**12.655**
15	9.319	9.024	9.769	10.179	10.705	**13.289**
16	9.217	9.008	10.314	10.30	10.541	**13.042**
17	9.012	9.105	9.011	10.14	11.043	**13.705**
18	9.201	9.32	9.835	10.431	11.195	**11.977**

Bold values indicate the best results

Table 9 Comparison of the computation information content with different algorithms

Algorithm	Information content (worst)	Information content (average)	Information content (best)
ACRI[16]	9.895	10.273	10.530
MEME[8]	9.008	9.508	10.124
AlignACE[38]	9.010	9.752	10.314
Gibbs sampler[7]	9.006	9.229	9.412
DMD_CRO (without repair)	10.541	11.267	12.442
DMD_CRO (with repair)	**11.377**	**13.051**	**14.551**

Bold values indicate the best results

Table 10 *t-value* of information content between DMD_CRO with other algorithms

Algorithm	Gibbs sampler [7]	MEME [8]	AlignACE [38]	ACRI [16]
t-value compared with DMD_CRO (without repair)	8.590	6.900	6.0354	4.099
t-value compared with DMD_CRO (with repair)	16.978	13.795	12.946	11.0532

we made the average of these five running times for each algorithm to get the final running time. Then the value of *iteration* was changed to find the running times for different values of *iteration*. Thus the running times were calculated for all values of *iteration*.

Table 12 depicts the running time comparison between ACRI and DMD_CRO using the CRP binding sites of *E. coli* dataset. A line graph for this comparison has been depicted to better visualization as shown in Fig. 11. The running times of these two algorithms have been plotted under various iterations. From Table 12, it can be observed that when

Table 11 U and $Z_{stat.}$ of information content between DMD_CRO with other algorithms

Algorithm	Gibbs Sampler[7]		MEME[8]		AlignACE[38]		ACRI[16]	
	U	$Z_{stat.}$	U	$Z_{stat.}$	U	$Z_{stat.}$	U	$Z_{stat.}$
Compared with DMD_CRO (without repair)	2	− 5.046	29	− 4.35	2	− 3.907	2	
Compared with DMD_CRO (with repair)	0	− 5.11	0	− 5.11	0	− 5.11	0	− 5.11

the number of iterations is 30 the running time of DMD_CRO (without repair) is less than that of ACRI. On the other hand, when the number of iterations is 45 DMD_CRO (with repair) takes less time than ACRI.

Similarly, Table 13 and Fig. 12 give the results and graphs of the running time comparison between ACRI and DMD_CRO using RAP1 of *Saccharomyces cerevisiae*. From Table 13, it can be noticed that when the number of iterations is 15 both DMD_CRO (without repair) and DMD_CRO (with repair) take less time than ACRI.

From Figs. 11 and 12, it can be observed that when the number of iterations increases, then the running time of ACRI also increases rapidly but in the case of DMD_CRO the running time increases very slowly. It proves that DMD_CRO takes less running time than the ACRI when the number of iterations increases.

6 Conclusions

This paper is concerned with a renowned NP-hard combinatorial problem called motif discovery from biological sequences. Nowadays, as the demand for analyzing important biological sequences is rapidly growing with the time, so researchers have focused on solving this problem. It is very useful and has great applications in the field of bioinformatics. Several algorithms were proposed with good results but there still need more precise identification of motif in a shorter period of time. Here a population-based metaheuristic algorithm Chemical Reaction Optimization (CRO) is selected to solve the motif discovery problem. Four basic operators of CRO have been redesigned to find the solutions. Besides, one additional repair operator has been designed to find better potential motif and another one is used to search for better binding sites. We compared the results of the proposed DMD_CRO algorithm with Ant Colony Optimization (ACO) based algorithm ACRI, Gibbs sampler, MEME, which are the state-of-the-arts. From the results, it can be concluded that in the case of five transcriptional factors of Homo sapiens dataset the found sequence logos are identical to the sequence logos by DNA footprinting method. In the case of the 18 sequences of the CRP binding sites for Escherichia coli dataset, DMD_CRO with repair operator gets better results than the other algorithms. The repair operators help our proposed DMD_CRO algorithms to get better results efficiently and effectively. Besides, the statistical tests demonstrate the superiority of DMD_CRO algorithm over other algorithms, which are state-of-the-arts.

Table 12 Running time comparison for the 18 sequences of the CRP binding sites for *E. coli*

No.	Iteration number	ACRI[16] (ms)	DMD_CRO (without repair) (ms)	DMD_CRO (with repair) (ms)
1	5	**37.6**	99.8	167.6
2	10	**58.2**	111.6	164.6
3	15	**74.2**	112	186.2
4	20	**98**	110.4	171.8
5	25	**111.6**	113	161.8
6	30	118.2	**115**	185.6
7	35	161.8	**114**	185.6
8	40	176.8	**117.4**	191.4
9	45	189	**145.6**	178.4
10	50	213.8	**132.6**	189.8
11	55	237	**132.8**	182.8
12	60	238.6	**133.4**	219.4
13	65	288.2	**135.6**	154.8
14	70	293	**138.6**	180.4
15	75	312.8	**143.4**	197.6
16	80	344.4	**139.4**	205.6
17	85	346.4	**117.4**	141.4
18	90	371.2	**145.8**	209
19	95	383.6	**151.2**	214.6
20	100	388.4	**153.6**	233.4

Bold values indicate the less running time

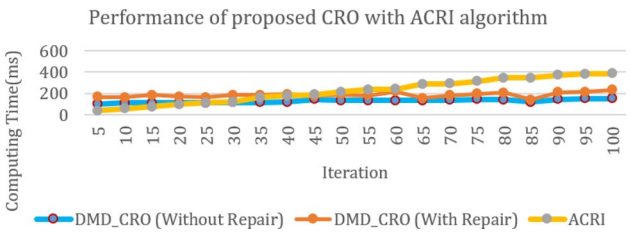


Fig. 11 Running time comparison for the 18 sequences of the CRP binding sites for *E. coli*

Table 13 Running time comparison for the RAP1 of *Saccharomyces cerevisiae*

No.	Iteration number	ACRI [16] (ms)	DMD_CRO (without repair) (ms)	DMD_CRO (with repair) (ms)
1	5	**87.6**	167.8	201.4
2	10	**166.8**	187.6	210.6
3	15	234.8	**196.4**	222.2
4	20	303.8	**192**	223.4
5	25	384.2	**207.4**	236.4
6	30	450.6	**209.8**	241.2
7	35	506	**210.6**	244
8	40	588.2	**220.4**	260.4
9	45	657.6	**230.8**	259.6
10	50	734	**222.2**	258.2
11	55	801.8	**235.8**	271
12	60	866.4	**237.8**	269.6
13	65	948.8	**243.8**	265.4
14	70	1018.4	**247.4**	271.6
15	75	1099.6	**251.4**	278
16	80	1150.8	**256.4**	277
17	85	1227.2	**256**	283
18	90	1301.2	**268.4**	291.6
19	95	1388.8	**293.2**	326.2
20	100	1424	**269.8**	294.6

Bold values indicate the less running time

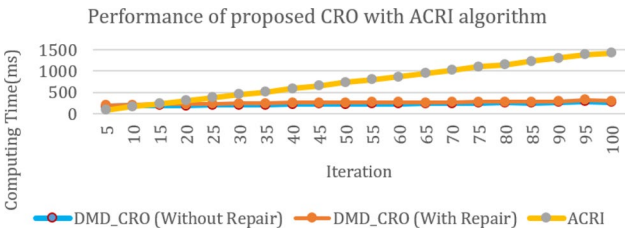


Fig. 12 Running time comparison for the RAP1 of Saccharomyces cerevisiae

To define the right values for the CRO parameters is a very difficult task. More statistical tests for proper parameters setting can be done to improve the results. The four operators of CRO can be modified to best suit for this problem. Better population initialization also can be beneficial.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Douglas Harper. motif. (1848, n.d.) Dictionary.com Unabridged. In https://www.dictionary.com/browse/motif
2. El Haj Mohamed AS, Elloumi M, Thompson JD (2016) Motif discovery in protein sequences, pattern recognition—analysis and applications, S. Ramakrishnan, IntechOpen, 14th Dec 2016, https://doi.org/10.5772/65441. https://www.intechopen.com/books/pattern-recognition-analysis-and-applications/motif-discovery-in-protein-sequences
3. Zambelli F, Pesole G, Pavesi G (2012) Motif discovery and transcription factor binding sites before and after the next-generation sequencing era. Brief Bioinform 14(2):225–237
4. Wikipedia contributors. Position. Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 1 Jan. 2019. Web. 13 May. 2019
5. Fan Y, Wu W, Liu R, Yang W (2013) An iterative algorithm for motif discovery. Procedia Comput Sci 24:25–29. ISSN 1877-0509. https://doi.org/10.1016/j.procs.2013.10.024. (http://www.sciencedirect.com/science/article/pii/S1877050913011666)
6. Huan HX et al (2015) An efficient ant colony algorithm for DNA motif finding. In: Knowledge and systems engineering. Springer, Cham, pp 589–601
7. Neuwald AF, Liu JS, Lawrence CE (1995) Gibbs motif sampling: detection of bacterial outer membrane protein repeats. Protein Sci 4(8):1618–1632
8. Bailey TL et al (2006) MEME: discovering and analyzing DNA and protein sequence motifs. Nucleic Acids Res 34(suppl2):W369–W373
9. Gutierrez JB, Frith M, Nakai K (2015) A genetic algorithm for motif finding based on statistical significance. In: International conference on bioinformatics and biomedical engineering. Springer, Cham
10. Che D, Song Y, Rasheed K (2005) MDGA: motif discovery using a genetic algorithm. In: Proceedings of the 7th annual conference on Genetic and evolutionary computation. ACM
11. Liu FFM et al (2004) FMGA: finding motifs by genetic algorithm. In: Proceedings. Fourth IEEE symposium on bioinformatics and bioengineering. IEEE
12. Al Daoud E (2013) Efficient DNA motif discovery using modified genetic algorithm. Int J Comput Intell Appl 12(03):1350017
13. Huo H, Zhao Z, Stojkovic V, Liu L (2010) Optimizing genetic algorithm for motif discovery. Math Comput Model 52(11–12): 2011–2020. ISSN 0895-7177 https://doi.org/10.1016/j.mcm.2010.06.003. (http://www.sciencedirect.com/science/article/pii/S0895717710002748)
14. Yang C-H, Liu Y-T, Chuang L-Y (2011) DNA motif discovery based on ant colony optimization and expectation maximization. In: Proceedings of the International multi conference of engineers and computer scientists. vol 1
15. Bouamama S, Boukerram A, Al-Badarneh AF (2010) Motif finding using ant colony optimization. In: International conference on swarm intelligence. Springer, Berlin
16. Liu W, Chen H, Chen L (2013) An ant colony optimization based algorithm for identifying gene regulatory elements. Comput Biol Med 43(7): 922–932. ISSN 0010-4825. https://doi.org/10.1016/j.compbiomed.2013.04.008. (http://www.sciencedirect.com/science/article/pii/S0010482513000978)
17. Claeys M et al (2012) MotifSuite: workflow for probabilistic motif detection and assessment. Bioinformatics 28(14):1931–932
18. Liu X, Brutlag DL, Liu JS (2000) BioProspector: discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes. Biocomputing 2001:127–138
19. Kirkpatrick S Jr, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. Science 220(4598):671–680

20. Hu J, Li B, Kihara D (2005) Limitations and potentials of current motif discovery algorithms. Nucleic Acids Res 33(15):4899–4913
21. Wingender E et al (1996) TRANSFAC: a database on transcription factors and their DNA binding sites. Nucleic Acids Res 24(1):238–241
22. Lam AYS, Li VOK (2012) Chemical reaction optimization: a tutorial. Memet Comput 4(1):3–17
23. Islam MR, Khaled Saifullah CM (2019) Mahmud MR (2019) Chemical reaction optimization: survey on variants. Evolut Intell 12(3):395–420
24. Lam AYS, Li VOK, Xu J (2012) On the convergence of chemical reaction optimization for combinatorial optimization. IEEE Trans Evolut Comput 17(5):605–620
25. Chaabani A, Bechikh S, Said LB (2018) A new co-evolutionary decomposition-based algorithm for bi-level combinatorial optimization. Appl Intell 48(9):2847–2872
26. Khaled Saifullah CM, Md Rafiqul I (2016) Chemical reaction optimization for solving shortest common supersequence problem. Comput Biol Chem 64:82–93
27. Islam MR et al (2018) Chemical reaction optimization for solving longest common subsequence problem for multiple string. Soft Comput. https://doi.org/10.1007/s00500-018-3200-3
28. Rayhanul K, Rafiqul I (2019) Chemical reaction optimization for RNA structure prediction. Appl Intell 49(2):352–375
29. Rafiqul Islam M, Mahmud R, Pritom RM (2019) Transportation scheduling optimization by a ollaborative strategy in supply chain management with TPL using chemical reaction. Neural Comput Appl. https://doi.org/10.1007/s00521-019-04218-5
30. Lam AYS, Li VOK (2009) Chemical-reaction-inspired metaheuristic for optimization. IEEE Trans Evolut Comput 14(3):381–399
31. Islam MR, Islam MS, Sakeef N (2019) RNA Secondary Structure Prediction with Pseudoknots using chemical reaction optimization algorithm. IEEE/ACM Trans Comput Biol Bioinform. https://doi.org/10.1109/TCBB.2019.2936570
32. Islam MR et al (2019) Optimization of protein folding using chemical reaction optimization in HP cubic lattice model. Neural Comput Appl 32:3117–3134
33. Blekas K, Fotiadis DI, Likas A (2003) Greedy mixture learning for multiple motif discovery in biological sequences. Bioinformatics 19(5):607–617
34. Attwood TK et al (2000) PRINTS-S: the database formerly known as PRINTS. Nucleic Acids Res 28(1):225–227
35. Hofmann K, Bucher P, Falquet L, Bairoch A (1999) The PROSITE database, its status in 1999. Nucleic Acids Res 27(1):215–219. https://doi.org/10.1093/nar/27.1.215
36. Stormo GD, Hartzell GW (1989) Identifying protein-binding sites from unaligned DNA fragments. Proc Natl Acad Sci 86(4):1183–1187
37. Harbison CT et al (2004) Transcriptional regulatory code of a eukaryotic genome. Nature 431(7004):99
38. Roth FP et al (1998) Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole-genome mRNA quantitation. Nat Biotechnol 16(10):939
39. Shao L, Chen Y, Abraham A (2009) Motif discovery using evolutionary algorithms. In: 2009 international conference of soft computing and pattern recognition. IEEE 2009
40. Zhu J, Zhang MQ (1999) SCPD: a promoter database of the yeast *Saccharomyces cerevisiae*. Bioinformatics (Oxford, England) 15(7):607–611
41. Sun J, Zhang Q, Tsang EPK (2005) DE/EDA: a new evolutionary algorithm for global optimization. Inf Sci 169(3–4):249–262
42. Wolfger H et al (1997) The yeast ATP binding cassette (ABC) protein genes PDR10 and PDR15 are novel targets for the Pdr1 and Pdr3 transcriptional regulators. FEBS Lett 418(3):269–274
43. Chan T-M, Leung K-S, Lee K-H (2007) TFBS identification by position-and consensus-led genetic algorithm with local filtering. In: Proceedings of the 9th annual conference on Genetic and evolutionary computation. ACM
44. Bryne JC et al (2007) JASPAR, the open access database of transcription factor-binding profiles: new content and tools in the 2008 update. Nucleic Acids Res 36(suppl1):D102–D106
45. Tompa M et al (2005) (2005) Assessing computational tools for the discovery of transcription factor binding sites. Nat Biotechnol 23(1):137