



Chemical reaction optimization for RNA structure prediction

Rayhanul Kabir¹ · Rafiqul Islam¹

Published online: 30 August 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

RNA Structure Prediction (RSP) is an optimization problem, where a stable secondary structure is obtained from an RNA primary sequence. To solve the RSP problem, many exact and metaheuristic algorithms were established in recent years. We have proposed an approach based on metaheuristic algorithm named Chemical Reaction Optimization (CRO) to solve the RSP problem. CRO is a population-based metaheuristic which has been employed in different optimization problems and works better than all other related existing algorithms. We have redesigned the reaction operators of CRO algorithm and calculated the minimum free energy of the RNA structure to solve RSP problem. The operators spread out the population entirely on the solution space using both local and global searches and find the better structure, which makes the proposed algorithm more unique. We have designed a novel operator called Repair function to verify and remove the repeated stem from the solution of an RNA sequence, which makes the process more time efficient. Both the quality of solutions and execution time are considered in designing the basic operators and the repair function. Thus, the proposed methodology gives robustness, efficiency, and effectiveness in solving the problem. The results of the proposed CRO based algorithm for RSP problem are compared with genetic algorithm (RNAPredict), simulated annealing algorithm (SARNA-Predict), coincidence algorithm (COIN), two-level particle swarm optimization algorithm (TL-PSOfold) and Changing Range Bat Algorithm (CRBA) to present that, the proposed work gives better results than those. The significance testing using Kruskal-Wallis test followed by post-hoc analysis also proves that the proposed work outperforms the five related methods.

Keywords Chemical reaction optimization · RNA structure prediction · Ribonucleotide sequences · Minimum free energy

1 Introduction

RNA has a fundamental role in protein synthesis and it is also important in genetic and evolution process. RNA is a macromolecule made from the nucleotide sequence. Four types of nucleotides are possible in RNA: adenine (A), uracil (U), cytosine (C), and guanine (G). Usually, RNA is single-stranded but stable double helix structures of correspondent strands can be formed. RNA bends and folds back to form hydrogen bonds between correspondent nucleotides and builds base pairs. The canonical base pairs in RNA are the stable Watson-Crick base pairs A-U and C-G, and the less stable ‘wobble’ pair G-U [1]. In protein

synthesis, three categories of RNA are used: messenger RNA (mRNA), ribosomal RNA (rRNA), and transfer RNA (tRNA). The mRNA contains the information needed for protein synthesis, while the rRNA is a ribosome component and tRNA transfers amino acids to the ribosome as basic materials for protein synthesis [2].

RNA has three types of structure; the nucleotide sequence is the primary structure. The secondary structure is the bonded base pairs in a two-dimensional way. RNA molecules in 3D space are called tertiary structure. RNA structure prediction is a process to calculate possible legal stems and select some of them to obtain an optimal result by constructing the secondary structure of RNA. At present, determining the structure of RNA has become the target of many researchers as it is one of the main issues in inventing new drugs and finding out the genetic diseases. Determining secondary structure is the first action in predicting the 3-D structure of RNA and interpretation of the biological function of the RNA molecules. The secondary structure provides numerous information about molecule structure. X-ray crystallography and Nuclear Magnetic Resonance

✉ Rayhanul Kabir
nabid.ku.cse@gmail.com
Rafiqul Islam
dmri1978@gmail.com

¹ Computer Science, Engineering Discipline, Khulna University, Khulna 9208, Bangladesh

(NMR) spectroscopy are also used to obtain the secondary structure. These processes are difficult, slow and costly. On that account, it is needed to implement mathematical and computational methods to solve the RSP problem.

Dynamic Programming (DP) was the first approach to solving RSP problem based on the minimization of the free energy and successfully predicts the secondary structure of the small pseudoknot-free RNA molecule [3]. DP based mfold was developed by Zuker [4] for inventing the pseudoknot-free RNA secondary structure. Thermodynamic models were used to calculate the free energy of an RNA secondary structure [5].

The use of Simulated Annealing (SA) for predicting RNA secondary structures was first described by Schmitz and Steger [6]. In this method, iterative formation and separation of the single base pair through SA is used. Tsang and Grypma presented a permutation-based algorithm SARNA-Predict, for RSP problem which is based on simulated annealing [2]. They studied the synchronism behavior of the algorithm. The method requires a process of removing conflicted helix to correct the structure. This process is quite tedious especially in a sequence that has a large number of possible helices. N. McMillan investigated a solution for RSP problem using Ant Colony Optimization (ACO) [3]. Initially, all feasible stems are recognized using a brute-force algorithm for a particular RNA sequence. Then, using ACO, new stems are summed up probabilistically by an ant to build a possible secondary structure. The procedure is repeated for a number of ants and also for a particular iteration, the pheromone trails for all the structures are improved depending on the ideal ant, in the matter of minimum free energy.

Particle Swarm Optimization (PSO) is known as useful in solving many different types of optimization problems and known for being able to find out the global optimal outcomes in the solution space [7]. Several methods were introduced based on the PSO for the RSP problem. HelixPSO was introduced for finding RNA secondary structures with minimum free energy by Geis and Middendorf [8]. Another approach is set-based PSO to optimize the RNA molecule structure using an advanced thermodynamic model which was proposed by Neethling and Engelbrecht [9]. An improved PSO (IPSO) model was presented by Liu [10]. The authors designed an objective function according to the number of selected stems, the average length of selected stems and the minimum free energy. Another author, Xing, introduced PSOfold based on improved PSO. An adaptive parameter controller of PSO is applied to promote the balance between exploration and exploitation [11, 12]. In PSOfold, some crucial stems were ignored while predicting secondary structure. Genetic Algorithm (GA) based method named RNAPredict was proposed by Wiese using thermodynamic models [7]. In this method, a set of feasible

helices is produced from a given RNA sequence with helix generation algorithm [1]. A permutation of the helix numbers represents each chromosome. If two or more helices in any chromosome share some common base pairs, then the conflict helices are not added to the RNA fold. After that crossover and mutation operators are used to improve the chromosome to find a minimum free energy finally [13]. For the sequences with a large number of helices, this method is time-consuming. For global optimization, another meta-heuristic algorithm is known as the Bat algorithm. Changing Range Bat Algorithm (CRBA) was introduced for finding RNA Secondary Structure Prediction by Zihua, Li, Cao and Zhu [32]. This paper represents an updated version of Bat algorithm. Bat algorithm describes microbats echolocation behavior. To predict RNA secondary structure CRBA showed the prediction for ten shorter sequences. And the results were compared with mfold.

In this paper, we have proposed an algorithm based on the Chemical Reaction Optimization (CRO) to solve the RNA structure prediction problem. Our main target is to find out the most stable secondary structure of an RNA sequence. The chemical reaction is a process involving the rearrangement of the molecular and ionic structure of substances. There is a common nature of this universe that every molecule or ion that is not in the stable state wants to be stable by chemical reaction. The CRO algorithm follows this exact behavior. The important feature of CRO is its searching ability. It has both local and global search properties. Another feature of CRO is the high flexibility of designing reaction operators and population generation. These two features help CRO to fit for any optimization problem and finding out the global best results of the optimization problems. In recent years CRO has successfully solved many optimization problems and showed better results than other meta-heuristics approaches. To solve 0-1 knapsack problem, the CRO with Greedy strategy showed a better result than ant colony optimization, genetic algorithm and quantum-inspired evolutionary algorithm [14]. CRO was also applied to the quadratic assignment problem [15], cognitive radio spectrum allocation problem [16], and network coding optimization [17]. To solve multiple choice 0-1 knapsack problem, the artificial CRO outperforms the genetic algorithm [18].

Contribution and novelty of this work are summarized below.

1. We have redesigned four reaction operators: On-wall ineffective collision, Decomposition, Inter-molecular ineffective collision and Synthesis to find the global optimal point. These operators make the whole process able to search best structures of RNA sequences. On the other hand, the proposed method gives the most

- stable structures for both shorter and longer sequences because of these operators.
2. A novel efficient approach has been proposed here. Chemical Reaction Optimization (CRO) algorithm has been successfully applied to solve different NP-hard problems, however it was not used or proposed for solving RNA structure prediction problem.
 3. A new solution generation process has been introduced for CRO Algorithm. The process is efficient in generating valid solution for RNA structure prediction problem.
 4. Repair function is one of our other novel tasks. With the help of repair function we verify and remove the duplicate stem number(s). While a solution is generated by any of the four basic operators of CRO, if there exist duplicates of stem numbers during the construction process of secondary structure then duplicate stem numbers are taken into consideration repeatedly, consequently the process takes lot a time. So the repair function makes the CRO algorithm robust and time efficient.
 5. Our proposed work follows the hydrogen bond model (INN-HB) which is a group of thermodynamics model to estimate minimum free energy (MFE). This model is subtle and easy to implement and takes less time to obtain minimum free energy than all other models. These properties of the proposed work make it possible to give better results than other methods. The outcomes of the proposed work are compared with the previous related methods such as RNAPredict (GA) [7], the SARNA-Predict (SA) [2], COIN [13], TL-PSOfold [19] and CRBA [32] to show the performance of the proposed method.

2 Problem statement

The RSP is a problem to anticipate RNA secondary structure. Here, an RNA sequence is given to compute the correct secondary structure. The secondary structure of RNA is described by a list of base pairs formed from the primary sequence.

Let $S = s_1, s_2, \dots, s_n$ be an RNA sequence. Here S is a string of alphabet $\Sigma = \{a, u, g, c\}$. A pair (x, y) is called a base pair (complimentary) if $\{x, y\} = \{a, u\}$ or $\{x, y\} = \{g, c\}$. Pairs like $\{a, g\}$, $\{c, u\}$, $\{a, c\}$ are not treated as a base pair [20]. The most stable and common of these base pairs are $\{g, c\}$, $\{a, u\}$, and $\{g, u\}$, and their opposite, $\{c, g\}$, $\{u, a\}$, and $\{u, g\}$. When all the pairs are built, the RNA strand folds back to produce the secondary structure. Our main objective is to maximize the number of the stem to create RNA secondary structure from a given sequence and select the most stable secondary structure. The stability of a structure depends on the Gibbs free energy (ΔG). The structure with minimum

energy is accepted. (ΔG) is used to calculate the total energy of different structures (RNA) of the same sequence. We use the individual nearest-neighbor hydrogen bond model (INN-HB) [21] for calculating the free energy of a helix in the RNA secondary structure. Now we define an objective function for the RNA structure prediction.

$$R = \min\{\Delta G_i\}; \text{ where } 1 \leq i \leq n; n = \text{number of secondary structure for one sequence} \quad (1)$$

$$\Delta G_{37}^{\circ} = \Delta G_{37}^{\circ}{}_{init} + \sum [\Delta G_{37}^{\circ}{}_{NN}] + \Delta G_{37}^{\circ}{}_{AU/GUend}(\text{perAU/GUend}) + \Delta G_{37}^{\circ}{}_{sym} \quad (2)$$

This Eq. (2) is widely used to calculate the fitness of the RNA secondary structure. The values of these parameters are taken based on [22]. In Table 1, the meaning of every symbol is given.

In RNA structure pseudoknots are formed by the cross of base pairs, the cross of stems can form pseudoknotted structure. It is difficult for a prediction algorithm to compute large RNA molecules with pseudoknots. Jens and Robert proposed an algebraic dynamic programming algorithm for finding RNA pseudoknotted structure that takes a lot of time and space because of the pseudoknotted structure of RNA [20]. Another approach for RNA pseudoknotted structure of predigesting model based on minimum free energy presented by Rivas and Eddy which takes more time and space [23]. The problem for predicting RNA pseudoknotted secondary structure is an NP-complete and maximizing the number of stacking pairs makes it NP-hard. In mimic RNA structure, there exist pseudoknots. Several publications show that extending the RNA structures including arbitrary pseudoknots indicates the problem of finding the optimum structure is NP-hard [24, 25].

3 Related works

In the recent years, different approaches were proposed to predict RNA secondary structure. Some of them are briefly described here.

3.1 Dynamic programming

DP (Dynamic Programming) is a method of solving any complex problem by splitting up the problem into small problems. Then each small problem is solved and the results are stored, and later, they are calculated by the recursive algorithm. DP based algorithm, mfold was developed by Zuker [4] for inventing the pseudoknot-free RNA secondary

Table 1 Symbol table for (2)

Symbol	Meaning
$\Delta G_{37init}^{\circ}$	It is a constant and used for the entropy loss when the first base pair is formed and initiation occurs.
$\Delta G_{37AU/GUend}^{\circ}(perAU/GUend) + \Delta G_{37sym}^{\circ}$	Per each AU or GU pair at the end of a helix, it is applied once.
ΔG_{37sym}°	It is a symmetry correction. 0.43kcal/mol for self-complementary duplexes and 0 for otherwise.
$\sum [\Delta G_{37NN}^{\circ}]$	It is a sum of all base pairs that are formed by sliding window of size two through a helix from left to right. The value for each base pair depends on the nearest neighbor base pair of that base pair.

structure with minimized free energy. For the prediction of secondary structure of RNA, “mfold web server” represents a number of closely nearly connected software applications accessible on World Wide Web (WWW). The web server takes the primary RNA sequence as input and computes pseudoknot-free secondary structure with minimized free energy (MFE) and predicts some suboptimal structures. A possible secondary structure is anticipated by computing the sum of free energies from each optimal substructure, for all probable combinations of substructures and at last, the combination with minimum free energy is selected. In the mfold (multiple folds) web server, users get some options to choose a window for optimal substructure and push some selected base pairs to take up some reserve information into account. The main algorithm brings the energy dot plot matrices for the base pairs involved in the folding. Although dynamic programming algorithm produces optimal and suboptimal structures with minimum free energy, it is incapable of considering the kinetic effects concerned to easily approachable states in RNA folding.

3.2 Simulated annealing

Simulated annealing (SA) is a searching approach for finding the comparative global optimum of a given operation. The method is a metaheuristic to estimate global optimization within vast search space. The use of Simulated Annealing for predicting RNA secondary structures, using free energy minimization approach, was first proposed by Schmitz and Seger [6]. The secondary structure is predicted by iterative formation and disruption of single base pairs. Another prediction based on SA was presented by Tsang and Grypma, a permutation-based algorithm [2]. The structure of SARNA-Predict is developed by four elements. These are i) state representation, ii) perturbation/mutation function, iii) evaluation function and iv) decision mechanism. They compared the performance of the inversion mutation operator to the percentage swap mutation operator used in SARNA-Predict. Then, they evaluated the use of different annealing schedules used in tandem with the two mutation operators. For testing data,

13 different RNA sequences were used and they were taken from the comparative RNA web site [26]. The results of the experiments of the process include sensitivity, specificity, and F-measure. It also includes Known bps, Predicted bps, true positive base pairs (TP), false positive base pairs (FP), and false negative base pairs (FN) [2]. It showed good results in term of sensitivity, selectivity, and F-measure. The results suggest that the inversion mutation operator could be used in place of the swap mutation operator when using geometric scheduled simulated annealing. It is useful only for lower energy sequences. In adaptive scheduling, at a given temperature the iterations number in the search space is variable which takes a long time in searching.

3.3 Genetic algorithm

A genetic algorithm is usually used to produce ideal solutions for optimization and search problems using mutation, crossover, and selection operators. Wiese proposed a method named RNAPredict based on thermodynamic models [7]. In that paper, at first, they evaluated the performance of the stacking energy based thermodynamic models, Individual Nearest Neighbor Hydrogen Bond (INN-HB) model precisely by assessing the statistical correlation between the free energy in a structure and the number of true positive base pairs. Then they measured the accuracy of 19 structures from four RNA classes predicted by RNAPredict by comparing them to known structures and compare the prediction accuracy of RNAPredict and the mfold DPA. They demonstrated that evolutionary computation is a feasible technique for RNA secondary structure prediction. They also demonstrated that RNAPredict gives more minimum ΔG structures than mfold [7]. TP, FP, and FN have been calculated and showed the sensitivity, specificity, and F-measure to further quantify the performance of RNAPredict and mfold. Another approach was proposed by Tong and Cheung [27]. They predicted RNA secondary structures with pseudoknots using GA. They named it GAKnot and used two valid datasets for testing. TheGAKnot was shown to be the best prediction method in terms of accuracy and speed compared to several competitive prediction methods

[27]. One dataset is constructed from a subset of sequences used in HotKnots that contains 41 sequences and another dataset contains RNA secondary structures with pseudoknots. They calculated sensitivity and positive prediction value (PPV) to evaluate the attainment of the method. GA gives a population of solutions as suboptimal structures and makes it probable to search not only the MFE structures but also other structures that are closer to the natural fold. GA is also suitable for estimating certain energy parameters but it takes a long time for predicting RNA structure of sequences with a large number of helices.

3.4 Two-level particle swarm optimization algorithm

A set-based algorithm named two-level particle swarm optimization algorithm (TL-PSOfold) was applied to solve RSP problem [19]. The first objective of the method is to maximize the number of the stems for a given RNA sequence. After that, they have analyzed the minimum free energy. The procedure consists of an initialization stage, where the sets of levels one and two for each swarm are initialized, and then the procedure goes through two levels. In each level, the velocity and position of the level are updated, and the fitness value is evaluated. After that, the best and global best values are updated for the level. If the procedure meets the stopping criteria, it exports global best secondary structure. The outcomes of TL-PSOfold is compared with HelixPSO v1, HelixPSO v2, PSOfold, SetPSO, IPSO, FPSO, RNAfold, mfold and RNA-Predict, SARNA-Predict in terms of sensitivity, specificity, and F-measure. They have also shown the hypothesis test using the Kruskal-Wallis test followed by post-hoc analysis. All the results and analyses prove the higher prediction accuracy of TL-PSO than the other algorithms.

3.5 Coincidence algorithm

An evolutionary algorithm named coincidence algorithm was introduced by Srikamdee [13] to solve RSP problem. The main task of the algorithm was to obtain the feasible combination of helices from a sequence that has the minimum free energy among all combinations. At first, all the helices from a sequence is found by a helix generation algorithm. For this, the canonical base pairs are obtained at the beginning of the algorithm. After getting the population, they evaluated the population by calculating free energy by the INN-HB model. In the next step, the good solutions are selected as candidate solutions. At last the generator matrix is updated, which contains the probability of any helix that can appear together. They have taken 10 sequences for testing and the test results are compared with the two algorithms SARNA-Predict and RNA-Predict.

3.6 Changing range bat algorithm

Bat algorithm is another metaheuristic algorithm for global optimization. Zhihua, Li, Cao and Zhu introduced a new approach of Bat algorithm known as Changing Range Bat Algorithm (CRBA) [32]. This paper represents an updated version of Bat algorithm. Bat algorithm describes microbats echolocation behavior. The original bat algorithm has different variables for virtual bats: frequency, velocity, location etc. The bat algorithm adjusts its frequency first and updating its velocity and position next. By walking randomly, a new solution is picked from the current solution for the best solution. This paper improves the exploitation capability by generating each bat location using the random walk. Average loudness and rate of the pulse are used in the new algorithm, which indicates loudness decreases while the rate of pulse emission increases. The new algorithm improves on the previous the bats algorithm. Here the first bat has the best performance and the performance goes down with the list. They have taken 10 different sequences to predict secondary structure. The results of the experiments of the process include sensitivity, specificity, and F-measure. And to measure these three, this method calculated TP, FP, and FN. CRBA only took shorter sequences to compare with mfold. For some of the sequences, the results of mfold were better than CRBA.

3.7 Advances and assessment of 3D structure prediction

Miao and Eric discussed advances and assessment of 3D structure prediction [33]. This paper first described sequence (1D) to secondary (2D) structure, and then non-Watson-Crick (WC) base pair tertiary structure (3D). Then, it described previous module detection programs, namely RMDetect, JAR3D which can correctly identify standard RNA modules to a degree. RNA Puzzles contains 17 puzzles, which must be tackled by each program. It also indicated score which programmer can look at and improve upon it. They showed some results in term of INF for some sequences.

4 Chemical reaction optimization

Chemical reaction optimization (CRO) is a relatively new invented metaheuristic for optimization that is influenced by the nature of chemical reactions. In an atomic glimpse, a chemical reaction begins with some inconsistent molecules with enormous energy. The molecules collaborate with each other through some fundamental reactions and ultimately, they are altered to those with minimum energy to confirm

their presence. Such a characteristic is inflicted in CRO to solve optimization problems. When a molecule occupies excessive energy, which means it is unstable, it manipulates itself to reduce excessive energy to stabilize, and this process is called chemical reaction [28]. Chemical reaction ends with a stable product with minimum energy which is alike to a stepwise process of finding the optimal point. The energy of molecules is of two types: kinetic energy (KE) and potential energy (PE). KE is a positive number and it evaluates the tolerance of the system accepting a worse solution than the current one. PE is represented as the objective function value of the corresponding solution. The change of the molecule from its unstable to a stable state is generated by a collision and two types of collisions exist such as unimolecular and inter-molecular collisions. The first type defines the status when the molecule hits on some external elements like a wall of a container. The second type defines the collision between a molecule and other molecules.

There are four elementary reactions of CRO: on-wall ineffective collision, inter-molecular ineffective collision, decomposition, and synthesis. Whereas an on-wall or

inter-molecular ineffective collision occurs, the number of molecules earlier and then remains the same. Besides, when decomposition and synthesis occurs, the number becomes more or less [28]. For neighborhood search, the on-wall ineffective reaction is applied where the minimal change in molecular structure gives almost identical solutions [29]. Decomposition refers to the situation when a molecule hits a wall and then breaks into several molecules. Drastic change occurs in the molecular structure of new molecules in this reaction. The solution bounces to other portions of the solution space and in this way a global search is attained. Inter-molecular collision contains inter-molecular ineffective collision and synthesis. An inter-molecular ineffective collision occurs when multiple molecules clash with each other and after that bounce away. The molecularity remains unchanged before and after the process. This reaction is applied to the local search where the reactant molecules have minimal change in their molecular structure. Synthesis acts opposite of decomposition. A synthesis reaction occurs when multiple molecules hit against each other and integrate together. The elementary reaction can only occur when

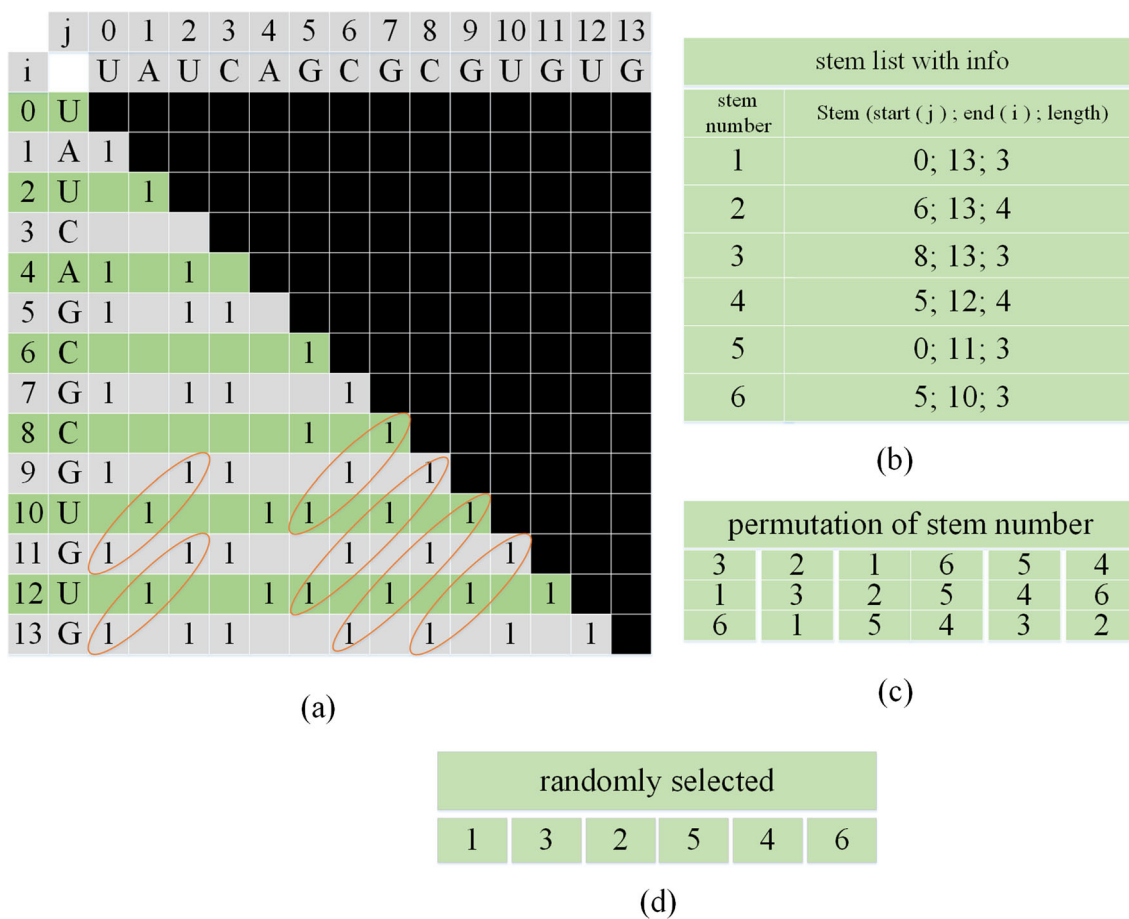


Fig. 1 (a) a sequence matrix in which the individual row and column represent the primary sequence (b) reported stem list with info of start, end, and length (c) Some sequences after permutation (d) a randomly selected sequence of stem numbers

it assures the energy conservation condition as given below:

$$\sum_{i=1}^k (PE_{\omega_i} + KE_{\omega_i}) \geq \sum_{i=0}^l PE_{\omega'_i} \quad (3)$$

Here, ω and ω' are the structures of input and output molecules, respectively used in a reaction. The number of molecules that takes part in the reaction is denoted by k .

Literally, the steps of the chemical reaction optimization are achieved by generating a population of initial molecules(solutions) randomly first, and then calculating each solution's objective function value as its PE and initialize each molecule's other attributes. Then by mean of the parameter Molecoll (1 or 0) we choose one or two molecules from the population randomly by selecting one of the four elementary reactions and generate the new molecule(s) according to the corresponding reaction layout. If there is acceptable energy for the new molecule(s) to be generated, change the original molecule(s) with the new one(s) and then renew the KE. When the stopping criterion is met, the global minimum solution is obtained. This solution is the resultant solution.

4.1 Algorithm design

In the proposed methodology, we have applied CRO to solve RSP problem and CRO is a population-based metaheuristic, it contains three stages: initialization, iterations and the final stage. In initialization, the parameters of the algorithm are assigned, the algorithm analyzes the solution space in iterations, and it terminates with the best output so far in the final stage.

4.1.1 Initialization and population generation

At first, the values are assigned to the parameters such as PopSize, KELossRate, MoleColl, buffer, InitialKE, two thresholds (α and β) and the initial population of molecules is constructed in the initialization phase. To construct the population of molecules possible candidate stems are found. The construction of candidate stems is easy: for a given RNA sequence with length n , we take a matrix. In the matrix, a distinct row and a column denote the corresponding nucleotide in the input RNA sequence.

When, an element in the matrix can build a canonical Watson-Crick base pair (AU, GC) or Wobble base pair (GU) then the element is marked as 1 otherwise, it will be 0. When the matrix is filled, we can find a list of maximal stems. Maximal stems are found by checking the 1s from bottom left to top right of the diagonals (shown in ellipses). A feasible RNA structure has a minimum stem length of three. That means, if a stem has three base pairs or more, it is taken for secondary structure. If the number of 1s in a diagonal is three or more than three, we consider

that stem for the count otherwise, the stem is neglected. When all the maximal stems are found, we build a list of stems. In the list, each element stores the stem number, the start, the end, and length of the stem. After that, by permuting the stem numbers of the stem list, we select a sequence which is said to be the randomly selected molecule (Fig. 1).

Algorithm 1 represents a generation of the population. After population generation, it is needed for initialization of the parameters of each molecule. To do this, "Molecule-CRO" class is built with some attributes and methods to initialize the parameters of the molecules. The attributes are ω , PE, KE, NumHit, MinStruct, MinPE, MinHit. There are five methods in the class, including the class constructor and the four elementary reactions. The constructor identifies the details of a molecule when it is created according to the class. The initial set of solutions randomly generates in the solution space, a random solution to ω in the constructor is assigned. Algorithm 2 represents the Molecule-CRO class. PopSize, a number of molecules are created from "Molecule-CRO" to produce the initial population of molecules. PE is measured by the (2).

Algorithm 1. population generation

Input: sequence, number of population

```

1: for  $i \leftarrow 0$  to sequence length do
2:   for  $j \leftarrow 0$  to  $i$  do
3:     if  $j < i$  then
4:       if sequence[ $i$ ] and sequence[ $j$ ] from
         basepair then
5:         zero_one[ $i, j$ ]  $\leftarrow$  1
6:       else
7:         zero_one[ $i, j$ ]  $\leftarrow$  0
8:       end if
9:     end if
10:  end for
11: end for
12: for  $i \leftarrow$  (sequence length - 1) to 0 do
13:   for  $j \leftarrow 0$  to (sequence length - 2) do
14:     if diagonal 1s found then
15:       Trace the first, last positions, and
         length, and assign them to information table
16:     end if
17:   end for
18: end for
19: for  $i \leftarrow 0$  to the number of the population do
20:   sequence number  $\leftarrow$  randomly generate
     an array where each value is between 0 to
     information table length
21:   Add sequence number in the molecule list
22: end for

```

Algorithm 2. Class: Molecule_CRO

```

1: Molecule_CRO()
2: {
3:   Generate  $\omega$  randomly in the solution space
4:    $KE \leftarrow InitialKE$ 
5:    $PE \leftarrow f(\omega)$ 
6:    $NumHit \leftarrow 0$ 
7:    $MinStruct \leftarrow \omega$ 
8:    $MinHit \leftarrow 0$ 
9:    $MinPE \leftarrow PE$ 
10: }
11: OnWallIneffectiveCollision()
12: IntermolecularIneffectiveCollision()
13: Decomposition()
14: Synthesis()
15: end class

```

4.1.2 Iteration and operator design

After the initialization, an operator is chosen to trigger. The decision depends on certain criteria. We have reconstructed the four reaction operators of CRO for the RSP problem. The four reaction operators are described below.

On-wall ineffective collision For neighborhood search, the on-wall ineffective reaction is used. Here, the new solution is near to the initial solution. In Fig. 2 m_new is the newly generated molecule from molecule m . Algorithm 3 represents the on-wall ineffective collision.

Algorithm 3. OnWallIneffectiveCollision(m)

Input: molecule $m[1,2,\dots,n]$

Output: m_new

```

1: copy the values from  $m$  to  $m\_new$ 
2: select  $i$  and  $j$  randomly from  $[1, 2, \dots, n]$ 
3: if  $(m[i] + r) \leq m.length$  then
4:    $m\_new[i] \leftarrow m[i] + r$ 
5: else
6:   if  $m[i] > r$  then
7:      $m\_new[i] \leftarrow m[i] - r$ 
8:   else
9:      $m\_new[i] \leftarrow r - m[i]$ 
10:  end if
11: end if
12: Repair( $m\_new$ )

```

A position i in the molecule m is randomly chosen and a random integer r chosen within the length of the molecule m . Next, we verify the following condition to put a value to the i^{th} position of m_new :

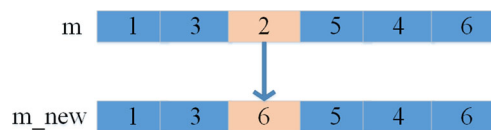


Fig. 2 On-wall Ineffective Collision

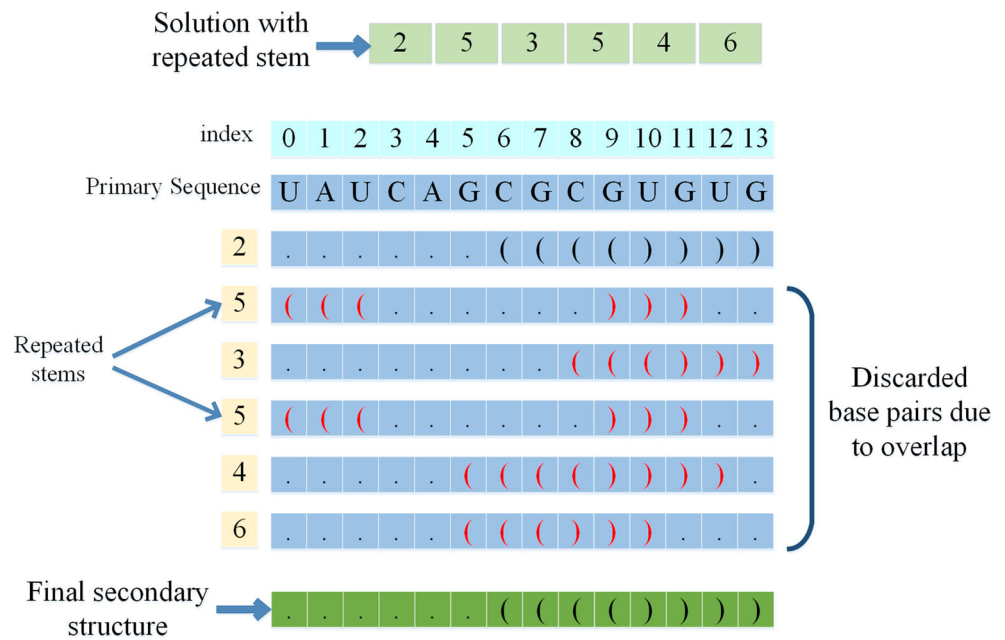
- i) If the value $m[i] + r$ is less than or equal to the length of m , $m[i] + r$ is assigned to $m_new[i]$.
- ii) If $m[i]$ is greater than r , the value $m[i] - r$ is assigned to $m_new[i]$.
- iii) Otherwise the value, $r - m[i]$ is assigned to $m_new[i]$.

The other values are copied from the respective positions (indices) of m to m_new . Suppose that $i = 3$ then $m[i] = 2$ as shown Fig. 2. We select $r = 4$ and get $m[i] + r = 6$, which is equal to the length of m . So, the value 6 is put to $m_new[i]$. After getting the new molecule m_new a Repair function (operator) is called to correct it.

Repair function Repair function is applied to the resultant (output) solution generated by any of the four basic operators. When a new solution is produced by any of the four operators, it may have duplicate repeated stem number(s). As a result, during the constructing process of the secondary structure, the duplicate stem numbers are taken into account repeatedly. This repeated consideration of duplicate stem number(s) takes a lot of execution time. In Fig. 3, an example of constructing process of a secondary structure is given. A solution with a duplicate stem is taken in this example. From the figure, we can see that, the same phenomenon is happening twice for the repeated stems. If we do not remove the duplicate stem then the same checking to be done twice. This repetition of the same task makes the procedure time-consuming. For this, it is required to remove the duplicated stem(s) to avoid excessive run time.

The main task of the repair function is to remove the duplicate stem number from the resultant solution and adjust the values in the solution by shifting the values after the duplicate stem number. So, there will be no repetition in the solution and no extra consideration is needed during the process of secondary structure. In Fig. 4, m is a solution built by any operator and m_new is the solution after applying repair function. In the solution, stem number 5 has a duplicate number. After eliminating the duplicate stem number(s) by the repair function, m_new is the updated solution and it makes the whole process time efficient. Algorithm 4 is for the repair function. Here, 0 is assigned to every index of the flag[i] at the beginning. Then within a loop, the values of flag array are checked and have been changed by putting 1s. If there is any repetition, it would be detected because it was already changed (there is 1 in that

Fig. 3 Constructing a secondary structure of a solution with repeated stems



index of the flag array). So, the respective value is removed and the length is decreased.

Algorithm 4. Repair(*m*)

Input: molecule *m*[1,2,...,n]

Output: *M*

```

1: for i ← 1 to m.length do
2:   flag[i] ← 0
3: end for
4: for j ← 1 to m.length do
5:   if flag[m[j]] = 0 then
6:     flag[m[j]] ← 1
7:   else
8:     remove value at index j from m
9:     j ← (j - 1)
10:  end if
11: end for
    
```

Decomposition The decomposition is used to analyze another area of the search space. There are enormous changes in the molecular structure of the newly generated product by decomposition. At first in this reaction, the input

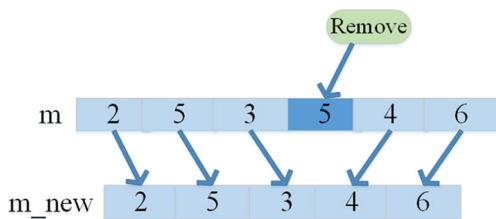


Fig. 4 Repair function

molecule is divided into two equal parts. For the output of this operator, we get two molecules *m*₁ and *m*₂. The values from the first half of *m* are copied and pasted to the respective half of *m*₁. Next, the values of the second half of *m* are copied and pasted them to the respective half of *m*₂. The other halves of *m*₁ and *m*₂ are created by randomly generated values from [0...n], where n is the length of the molecule. The scenario is depicted in Fig. 5.

Hence, the reaction looks for another region of solution space to find out the global minimum solution. The algorithm for decomposition is given in Algorithm 5.

Algorithm 5. Decomposition(*m*)

Input: molecule *m*[1,2,...,n]

Output: *m*₁ and *m*₂

```

1: mid ← m.length/2
2: Copy values from 0 to mid - 1 positions of m and put them to the respective positions of m1.
3: Copy values from mid to n - 1 positions of m and put them to the respective positions of m2.
4: Pick random values from [0...n] and put those mid to n - 1 positions of m1.
5: Pick random values from [0...n] and put those 0 to mid - 1 positions of m2.
6: Repair(m1)
7: Repair(m2)
    
```

Inter-molecular Ineffective Collision This reaction operator uses two molecules *m*₁ and *m*₂ randomly from the population and produces two new molecules *m*₁-new and *m*₂-new as shown in Fig. 6.

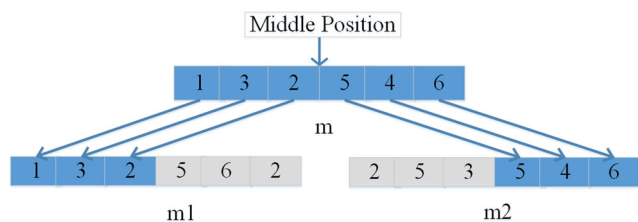


Fig. 5 Decomposition

Here two points x_1 and x_2 have been chosen randomly such that the points divide both input solutions into three parts. Then the odd parts of the m_1 and the even part of the m_2 join up to form m_1 -new and the even part of the m_1 and the odd parts of the m_2 join up to form m_2 -new. The two-third portions of the molecule remain unchanged and one-third is changed. The points x_1 and x_2 are picked randomly and with the help of these two values, the new molecule would be created. If the index i is between x_1 and x_2 , m_1 -new takes values from m_1 and m_2 -new takes values from m_2 , otherwise m_1 -new gets values from m_2 and m_2 -new gets values from m_1 . Algorithm 6 represents this reaction.

Algorithm 6. IntermolecularIneffectiveCollision(m_1, m_2)

Input: two molecules m_1, m_2

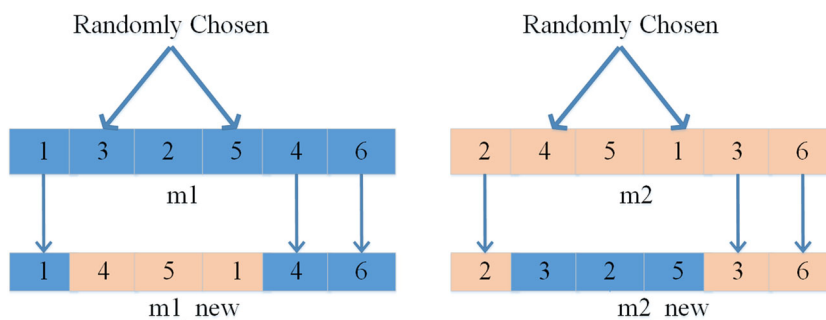
Output: m_1 -new and m_2 -new

```

1: Select twopoint  $x_1, x_2$  randomly
2: for  $i \leftarrow 1$  to  $m_1$ .length do
3:   if  $i < x_1$  or  $i > x_2$  then
4:      $m_1$ -new[ $i$ ]  $\leftarrow m_1$ [ $i$ ]
5:      $m_2$ -new[ $i$ ]  $\leftarrow m_2$ [ $i$ ]
6:   end if
7:   if  $x_1 \leq i \leq x_2$  then
8:      $m_1$ -new[ $i$ ]  $\leftarrow m_2$ [ $i$ ]
9:      $m_2$ -new[ $i$ ]  $\leftarrow m_1$ [ $i$ ]
10:  end if
11: end for
12: Repair( $m_1$ -new)
13: Repair( $m_2$ -new)

```

Fig. 6 Inter-molecular Ineffective Collision



Synthesis Two molecules m_1 and m_2 from the population are combined to form a new molecule m_{new} in the synthesis reaction. This reaction is the opposite of decomposition reaction. Two molecules m_1 and m_2 are chosen randomly from the population as shown in Fig. 7. The second step is to take a random value between 0 and 1. This random value helps to determine the molecule to choose. If the random value is less than 0.5, m_1 is chosen to select value(s) otherwise, m_2 is chosen to select the value(s) for m_{new} .

Algorithm 7. Synthesis(m_1, m_2)

Input: two molecules m_1, m_2

Output: m_{new}

```

1: Generate a random variable  $x$  between 0 to 1
2: for  $i \leftarrow 1$  to  $m_1$ .length do
3:   if  $x < 0.5$  then
4:      $m_{new}$ [ $i$ ]  $\leftarrow m_1$ [ $i$ ]
5:   else
6:      $m_{new}$ [ $i$ ]  $\leftarrow m_2$ [ $i$ ]
7:   end if
8: end for
9: Repair( $m_{new}$ )

```

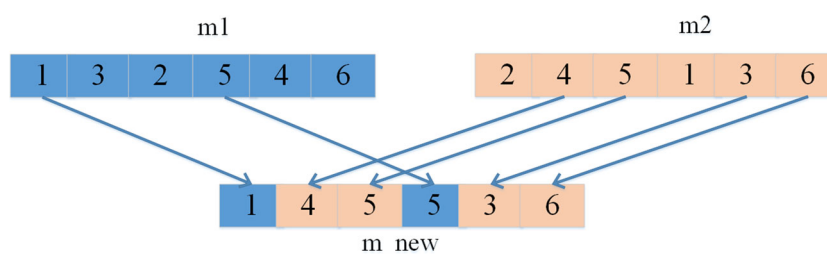
4.2 Parameter settings

The efficiency of the CRO algorithm depends on values of the parameters. The results of RNA Structure Prediction (RSP) using CRO algorithm greatly depend on the values of these parameters. The parameters with their respective values in CRO have been given in Table 2.

4.3 Operator selection

In this section, we describe the operator selection process in the CRO algorithm. A flowchart (Fig. 8) is given that depicts the stages step by step and the selection of the operator. The START and END express the initiation and the completion of CRO algorithm.

Fig. 7 Synthesis



The algorithm starts with the stage initialization then a number of iterations are executed and at the final stage, it terminates. In the initialization, values are assigned to some variables and control parameters. Based on the actions of decomposition and synthesis, the number of solution spaces can be changed as the CRO is a population-based metaheuristic algorithm.

At first, the population is created by producing the number of solutions randomly which is called POPSize. From these solutions, the feasible solution is searched out. A number of iterations are executed at the iteration stage and a collision is chosen. A decision is made whether the collision is a unimolecular or an intermolecular collision by generating a random number t which lies between 0 and 1. Unimolecular collision occurs if t is greater than MoleColl else an inter-molecular collision takes place (if there is only one molecule in the population then the unimolecular collision takes place). The flowchart of operator selection is shown in Fig. 8. After that, an applicable number of molecules from the population is selected randomly, based on the decision of collision type. Literally, the molecules occupied in collision forms on their physical location in the container. Later a decision of collision type is made by testing the criteria of decomposition and synthesis (at the left of the flowchart: on-wall ineffective collision

or decomposition, at the right of the flowchart: intermolecular ineffective collision or synthesis). The iteration stage continues until it meets the termination criteria. The termination criteria may be defined by the maximum number of iterations performed. The output solution is determined in the final stage with the lowest value found in the objective function.

5 Experimental results

For the experiment, two datasets were taken from the RNA STRAND v2.0 where in the first dataset there are 20 RNA sequences and in the second dataset there are 10 RNA sequences [30]. These sequence sets or datasets had been taken from the website (<http://www.rnasoft.ca/strand/>). The characteristic of each sequence of data set 1 is given in Table 3 and the information about each sequence of data set 2 is given in Table 4. The first column is for the sequence number which is maintained properly in other tables. The second column is for sequence name with an accession number. The third column defines the RNA class. The fourth column is for the length and the last column shows the number of base pairs in the sequence. Parameter settings for our proposed method were given in Table 2. For the comparison of the experimental results, we use the word CRO for our proposed work and other five algorithms such as RNAPredict, SARNA-Predict, COIN, TL-PSOfold and Changing Range Bat Algorithm are represented by GA, SA, COIN, TL-PSO and CRBA respectively.

The prediction accuracy is calculated by comparing the predicted structure with the known structure. The performance measures for the predicted structure are defined as follows: a total number of true positive base pairs of the prediction is TP, the total number of false positive base pairs of the prediction is FP, and the total number of false negative base pairs of the prediction is FN. On the other hand, sensitivity is the proportion of correctly predicted base pairs of the total number of base pairs found in the known structure. It is computed using (4). The specification is the proportion of all predicted true positive base pairs to all predicted base pairs. It is computed by (5). F-measure and INF are computed using

Table 2 Parameters of CRO for RSP problems with their initial values

Symbol	Value	
	With repair	Without repair
POPSize	200	200
KELossRate	0.8	0.8
MoleColl	0.3	0.3
InitialKE	2	3
A	1	1
B	8	8
Buffer	0	0
NumHit	0	0
MinHit	0	0
PE	Objective function	Objective function

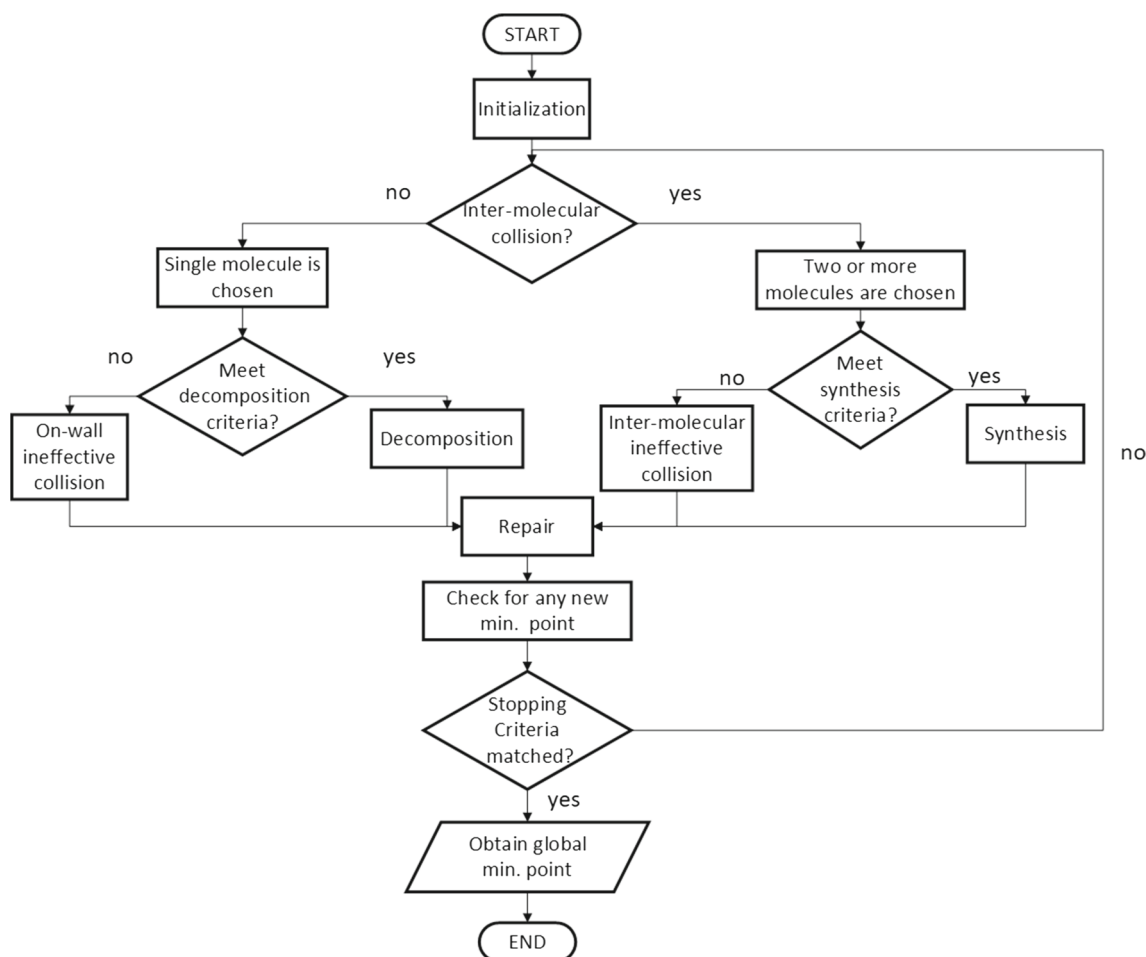


Fig. 8 Flowchart of operator selection

Table 3 Dataset 1: RNA sequence detail taken from the RNA STRAND V2.0

Seq No.	Sequence	RNA class	Accession number	Length (nt.)	#Base pair
S-1	G.stearothermophilus	5s rRNA	AJ251080	117	38
S-2	S.cerevisiae	5s rRNA	X67579	118	37
S-3	E. coli	5s rRNA	V00336	120	40
S-4	H.marismortui	5s rRNA	AF034620	122	38
S-5	T.aquaticus	5s rRNA	X01590	123	40
S-6	D.radiodurans	5s rRNA	AE002087	124	40
S-7	M.anisopliae(3)	Group I intron, 23S RNA	AF197120	394	120
S-8	C.saccharophila	Group I intron, 16S RNA	AB058310	454	126
S-9	M.anisopliae(2)	Group I intron, 23S RNA	AF197122	456	115
S-10	A.lagunensis	Group I intron, 16S RNA	U40258	468	113
S-11	H. rubra	Group I intron, 16S RNA	L19345	543	141
S-12	A. griffini	Group I intron, 16S RNA	U02540	556	131
S-13	P.leucosticta	Group I intron, 16S RNA	AF342746	605	121
S-14	C. elegans	16S RNA	X54252	697	189
S-15	D. virilis	16S RNA	X05914	784	233
S-16	A.cahirinus	16S RNA	X84387	940	260
S-17	X. laevis	16S RNA	M27605	945	251
S-18	H.sapiens	16S RNA	J01415	954	266
S-19	A.fulgens	16S RNA	Y08511	964	265
S-20	S.acidocaldarius	16S rRNA	D14876	1492	458

Table 4 Dataset 2: RNA sequence detail taken from the RNA STRAND V2.0

Seq No.	Sequence	Length (nt.)	#Base pair
S-1	Zea mays	81	22
S-2	Aspergillus fumigatus	73	20
S-3	Canis familiaris	84	20
S-4	Drosophila melanogaster(1)	73	20
S-5	Acidobacteria bacterium	89	24
S-6	Aquifex aeolicus	95	31
S-7	Drosophila melanogaster(2)	125	24
S-8	Kluyveromyces lactis	139	21
S-9	Sulfolobus shibatae	260	59
S-10	Alpha Purple Bacteria	289	71

language C# 6.0 and Microsoft Visual Studio 2013 IDE had been used.

$$Sensitivity = \frac{TP}{TP+FN} \tag{4}$$

$$Specificity = \frac{TP}{TP+FP} \tag{5}$$

$$F - measure = \frac{2*sensitivity*specificity}{sensitivity+specificity} \tag{6}$$

$$INF = \sqrt{\frac{TP}{TP+FP} * \frac{TP}{TP+FN}} \tag{7}$$

(6) and (7) respectively. We have taken the best results of ten iterations for every sequence of the proposed method. Our algorithm was implemented on Asus personal laptop. Model number: K46CA with Intel Corei3-3217u CPU @ 1.80GHz (4CPUs), 1.8GHz, 8192MB RAM and running on Windows 10 (64 bit). For the implementation, programming

The proposed CRO, GA and SA methods have been tested using 20 RNA sequences given in Table 3. The results of TP, FP, and FN found in the CRO, GA and SA are shown in Table 5. The proposed CRO gives the better TPs for all sequences than all other methods. The results of CRO for both with and without function are given in the table. The results of CRO with repair function are better than those of CRO without repair function. For the sequences of short length, the values of TP, FP, and FN are the same or close to the previous methods. The best outcomes are shown in bold text.

Table 5 Results of CRO, GA and SA in terms of TP, FP, and FN

Seq	TP				FP				FN			
	GA	SA	CRO Without repair	CRO with repair	GA	SA	CRO without repair	CRO with repair	GA	SA	CRO without repair	CRO with repair
S-1	23	22	35	35	10	20	3	2	15	16	3	3
S-2	33	33	34	34	6	4	0	0	4	4	3	3
S-3	10	20	37	38	29	26	1	1	30	20	3	2
S-4	27	27	34	37	3	3	0	0	11	11	4	1
S-5	33	26	37	38	3	23	1	1	7	14	3	2
S-6	25	18	36	36	8	21	0	2	15	22	4	4
S-7	75	67	95	96	46	52	32	32	45	53	25	24
S-8	86	64	94	97	51	126	47	42	40	62	32	29
S-9	55	48	87	92	80	90	54	47	60	67	28	23
S-10	68	67	85	91	63	64	51	39	45	46	28	22
S-11	79	74	113	112	82	88	59	62	59	64	28	29
S-12	81	79	107	107	80	100	61	61	50	52	24	24
S-13	63	45	88	91	90	203	74	70	58	76	33	30
S-14	55	43	127	129	147	160	163	156	134	146	62	60
S-15	65	55	167	170	177	191	146	144	168	178	66	63
S-16	74	111	165	166	154	207	143	145	186	149	95	94
S-17	93	103	169	169	147	133	157	156	158	148	85	85
S-18	89	111	171	174	161	146	149	147	177	155	97	94
S-19	82	92	172	176	160	162	152	149	183	173	93	89
S-20	224	219	336	345	412	254	157	144	235	249	123	114

Table 6 Results of CRO, GA, and SA in terms of sensitivity and specificity

Seq	Sensitivity				Specificity			
	GA	SA	CRO without repair	CRO with repair	GA	SA	CRO without repair	CRO with repair
S-1	60.5	57.9	92.1	92.1	69.7	52.3	92.1	94.5
S-2	89.2	89.1	91.8	91.8	84.6	89.1	100	100
S-3	25.0	50	92.5	95	25.6	43.5	97.4	97.4
S-4	71.1	71	89.5	97.3	90	90	100	100
S-5	82.5	65	92.5	95	91.7	53.0	97.3	100
S-6	62.5	45	90	92.3	75.8	46.2	100	94.7
S-7	62.5	55.8	79.2	80	62	56.3	74.8	75
S-8	68.3	50.8	74.6	76.9	62.8	33.7	66.7	69.7
S-9	47.8	41.7	75.6	80	40.7	34.7	61.7	66.1
S-10	60.2	59.2	75.2	80.5	51.9	51.1	62.5	70
S-11	57.2	53.6	80.1	79.4	49.1	65.6	65.6	64.4
S-12	61.8	60.3	81.7	81.7	50.3	44.1	63.6	63.6
S-13	52.1	37.2	72.7	75.2	41.2	18.1	54.3	56.5
S-14	29.1	22.7	67.2	68.2	27.2	21.1	43.8	45.2
S-15	27.9	23.6	71.6	72.9	26.9	22.3	52.7	53.3
S-16	28.5	42.6	63.5	63.8	32.5	34.9	53.6	53.4
S-17	37.1	41	66.5	66.5	38.8	43.6	51.8	52
S-18	33.5	41.7	63.8	64.9	35.6	43.1	53.4	54.2
S-19	30.9	34.7	64.9	66.4	33.9	36.2	53.1	54.1
S-20	48.8	46.7	73.2	75.1	35.2	46.3	68.2	70.5

Table 7 Results of CRO, GA, and SA in terms of F-measure and INF

Seq	F-measure				INF			
	GA	SA	CRO without repair	CRO with repair	GA	SA	CRO without repair	CRO with repair
S-1	64.8	55	92.1	93.3	64.94	55.03	92.1	93.29
S-2	86.8	89.1	95.7	95.7	86.87	89.1	95.81	95.81
S-3	25.3	46.5	94.9	96.2	25.3	46.64	94.92	96.19
S-4	79.4	79.38	94.4	98.6	79.99	79.94	94.59	98.64
S-5	86.8	58.4	94.9	96.2	86.98	58.69	94.87	97.47
S-6	68.5	45.6	94.7	92.3	68.83	45.6	94.87	93.49
S-7	62.2	56.05	76.9	77.4	62.25	56.05	76.97	77.46
S-8	65.4	40.5	70.4	73.2	65.49	41.38	70.54	73.21
S-9	44	37.88	67.9	72.4	44.11	38.04	68.3	72.72
S-10	55.7	54.85	68.2	74.9	55.9	55	68.56	75.07
S-11	52.8	49.28	72.2	71.1	53	59.3	72.49	71.51
S-12	55.5	50.94	71.5	71.5	55.75	51.57	72.08	72.08
S-13	46	24.4	62.2	64.5	46.33	25.95	62.83	65.18
S-14	28.1	21.87	53	54.4	28.13	21.89	54.25	55.52
S-15	27.4	22.93	61.1	62.1	27.4	22.94	61.43	62.33
S-16	30.3	38.4	58	58.1	30.43	38.56	58.34	58.37
S-17	37.9	42.26	58.2	58.3	37.94	42.28	58.69	58.8
S-18	34.5	42.39	58.2	59	34.53	42.39	58.37	59.31
S-19	32.3	35.43	58.4	59.6	32.37	35.44	58.7	59.94
S-20	40.9	46.5	70.5	72.7	41.45	46.5	70.66	72.76

Table 8 The results of CRO, COIN method in terms of TP, FP, and FN

Seq	TP			FP			FN		
	No.	COIN	CRO without repair	CRO with repair	COIN	CRO without repair	CRO with repair	COIN	CRO without repair
S-2	33	34	34	1	0	0	4	3	3
S-4	33	34	37	0	0	0	5	4	1
S-7	82	95	96	35	32	32	38	25	24
S-9	59	87	92	66	54	47	56	28	23
S-10	71	85	91	46	51	39	42	28	22
S-11	81	113	112	68	59	62	60	28	29
S-12	80	107	107	83	61	61	51	24	24
S-14	50	127	129	91	163	156	139	62	60
S-15	66	167	170	149	146	144	167	66	63
S-17	107	169	169	129	157	156	144	85	85

Table 9 The results of CRO, COIN in terms of sensitivity and specificity

Seq	Sensitivity			Specificity			
	No.	COIN	CRO without repair	CRO with repair	COIN	CRO without repair	CRO with repair
S-2		89.2	91.8	91.8	97.1	100	100
S-4		86.8	89.5	97.3	100	100	100
S-7		68.3	79.2	80	70.1	74.8	75
S-9		51.3	75.6	80	47.2	61.7	66.1
S-10		62.8	75.2	80.5	60.7	62.5	70
S-11		57.4	80.1	79.4	54.4	65.6	64.4
S-12		61.1	81.7	81.7	49.1	63.6	63.6
S-14		26.5	67.2	68.2	35.5	43.8	45.2
S-15		28.3	71.6	72.9	30.7	52.7	53.3
S-17		42.6	66.5	66.5	45.3	51.8	52

Table 10 The results of CRO, COIN in terms of F-measure and INF

Seq	F-measure			INF			
	No.	COIN	CRO without repair	CRO with repair	COIN	CRO without repair	CRO with repair
S-2		93	95.7	95.7	93.07	95.81	95.81
S-4		93	94.4	98.6	93.17	94.59	98.64
S-7		69.2	76.9	77.4	69.19	76.97	77.46
S-9		49.2	67.9	72.4	49.21	68.3	72.72
S-10		61.7	68.2	74.9	61.74	68.56	75.07
S-11		55.9	72.2	71.1	55.88	72.49	71.51
S-12		54.4	71.5	71.5	54.77	72.08	72.08
S-14		29.5	53	54.4	30.67	54.25	55.52
S-15		43.9	61.1	62.1	29.48	61.43	62.33
S-17		56.2	58.2	58.3	43.93	58.69	58.8

Table 11 Results of CRO, TL-PSO method in terms of TP, FP, and FN

Seq	TP			FP			FN		
	TL-PSO	CRO without repair	CRO with repair	TL-PSO	CRO without repair	CRO with repair	TL-PSO	CRO without repair	CRO with repair
S-1	27	35	35	7	3	2	11	3	3
S-2	33	34	34	5	0	0	4	3	3
S-4	31	34	37	5	0	0	7	4	1
S-5	36	37	38	3	1	1	4	3	2
S-14	88	127	129	102	163	156	94	62	60
S-15	104	167	170	127	146	144	123	66	63
S-17	122	169	169	123	157	156	126	85	85
S-18	132	171	174	139	149	147	128	97	94
S-19	106	172	176	127	152	149	151	93	89
S-20	276	336	345	168	157	144	182	123	114

Table 12 Results of CRO, TL-PSO method in terms of sensitivity and specificity

Seq	Sensitivity			Specificity		
	TL-PSO	CRO without repair	CRO with repair	TL-PSO	CRO without repair	CRO with repair
S-1	71.1	92.1	92.1	79.4	92.1	94.5
S-2	89.2	91.8	91.8	86.8	100	100
S-4	81.6	89.5	97.3	81.6	100	100
S-5	90.0	92.5	95	92.3	97.3	100
S-14	48.4	67.2	68.2	46.3	43.8	45.2
S-15	45.8	71.6	72.9	45.1	52.7	53.3
S-17	49.2	66.5	66.5	49.8	51.8	52
S-18	50.8	63.8	64.9	48.7	53.4	54.2
S-19	41.3	64.9	66.4	45.5	53.1	54.1
S-20	60.3	73.2	75.1	62.2	68.2	70.5

Table 13 Results of CRO, TL-PSO method in terms of F-measure and INF

Seq	F-measure			INF		
	TL-PSO	CRO without repair	CRO with repair	TL-PSO	CRO without repair	CRO with repair
S-1	75.5	92.1	93.3	75.14	92.1	93.29
S-2	88	95.7	95.7	87.99	95.81	95.81
S-4	83.8	94.4	98.6	81.6	94.59	98.64
S-5	91.1	94.9	96.2	91.14	94.87	97.47
S-14	47.3	53	54.4	47.34	54.25	55.52
S-15	45.4	61.1	62.1	45.41	61.43	62.33
S-17	49.5	58.2	58.3	49.5	58.69	58.8
S-18	49.7	58.2	59	49.74	58.37	59.31
S-19	43.3	58.4	59.6	43.35	58.7	59.94
S-20	61.2	70.5	72.7	61.24	70.66	72.76

Table 14 Results of CRO, CRBA method in terms of TP, FP, and FN

Seq	TP			FP			FN		
	CRBA	CRO without repair	CRO with repair	CRBA	CRO without repair	CRO with repair	CRBA	CRO without repair	CRO with repair
S-1	17	20	20	7	8	7	5	2	2
S-2	12	15	16	5	6	6	8	5	4
S-3	7	14	13	14	12	10	13	6	7
S-4	12	17	16	14	11	11	8	3	4
S-5	17	18	20	2	3	2	7	6	4
S-6	12	21	21	15	13	14	19	10	10
S-7	16	17	19	18	15	14	8	7	5
S-8	16	15	16	5	4	4	5	6	5
S-9	26	35	37	24	18	17	33	24	22
S-10	22	47	49	43	32	30	49	24	22

In Tables 6 and 7, the results of Sensitivity, Specificity and F-Measure, INF of CRO are compared with the Sensitivity, Specificity, and F-Measure, INF of the GA and the SA respectively. The proposed method shows relatively better results than the other two methods. The best outcomes are shown in bold text. Results of CRO with repair function are better than those of CRO without repair function.

In the papers [13], the method COIN was tested for 10 sequences. The CRO has been tested for the same 10 sequences. Table 8 shows the results of CRO and COIN in terms of TP, FP and FN. We have given the results of CRO and COIN with respect to sensitivity, specificity in Table 9 and in Table 10 the results of CRO and COIN with respect to F-measure and INF have been given. From the results, it can be proved that CRO gives better results than those of

COIN. The results of CRO with and without repair function are shown. The best results are given in the bold text.

In the papers [19], the method TL-PSO was tested for a specific 10 sequences. The CRO has been tested for the same 10 sequences. Table 11 depicts the results of CRO and COIN in terms of TP, FP and FN. We have shown the results of CRO and COIN with respect to sensitivity, specificity in Table 12. Table 13 shows the results in terms of F-measure and INF. From the results, it can be seen that CRO gives better results than those of TL-PSO. The results of CRO with and without repair function are shown. The best results are given in the bold text.

The method Changing Range Bat Algorithm (CRBA) was tested for 10 sequences in the paper [32]. This method took the dataset 2 for the experiment. Our method CRO has

Table 15 Results of CRO, CRBA method in terms of sensitivity, specificity, and F-measure

Seq	Sensitivity			Specificity			F-measure		
	CRBA	CRO without repair	CRO with repair	CRBA	CRO without repair	CRO with repair	CRBA	CRO without repair	CRO with repair
S-1	77.3	90.91	90.91	70.8	71.43	74.07	73.9	80	81.63
S-2	60	75	80	70.6	71.43	71.43	64.9	73.17	75.47
S-3	35	70	65	33.3	53.85	58.33	34.1	60.87	61.49
S-4	60	85	80	46.2	60.71	60.71	52.2	70.83	69.04
S-5	70.8	75	83.33	89.5	85.71	90	79.1	80	86.54
S-6	38.7	67.74	67.74	44.4	61.76	60	41.4	64.62	63.64
S-7	66.7	70.83	79.17	47.1	53.13	54.84	55.2	60.71	64.79
S-8	76.2	71.43	76.19	76.2	78.95	78.95	76.2	75	77.54
S-9	44.1	59.32	62.71	52	66.04	67.31	47.7	62.5	64.93
S-10	31	66.2	69.01	33.8	59.49	61.04	32.4	62.67	64.78

Table 16 Simulation results in terms of Sensitivity and Specificity of CRO with repair function

Seq	Sensitivity				Specificity			
	Best Soln.	Worst Soln.	Avg Soln.	Std. Dev.	Best Soln.	Worst Soln.	Avg. Soln.	Std. Dev.
S-1	92.1	84.1	88.1	2.2	94.5	91.4	92.5	1.8
S-2	91.8	78.3	86.4	3.8	100	96.6	98.7	1.6
S-3	95	82.5	88.2	3.5	97.4	91.6	93.9	2.8
S-4	97.3	89.4	92.1	2.1	100	100	100	0
S-5	95	87.5	91.5	2.1	100	92.3	96.1	2.4
S-6	92.3	80	84.5	4.0	94.7	91.4	93.0	1.9
S-7	80	68.3	73.5	3.7	75	65.6	70.0	3.0
S-8	76.9	67.4	72.6	3.0	69.7	59.7	64.4	3.3
S-9	80	67.8	72	4.0	66.1	56.5	60.6	2.8
S-10	80.5	66.3	73.4	4.2	70	58.5	62.3	3.1
S-11	79.4	70.9	75.9	3.5	64.4	56.0	61.4	4.5
S-12	81.7	71.7	76.1	3.0	63.6	58.0	62.1	2.4
S-13	75.2	66.1	69.8	2.2	56.5	48.7	54.1	3.9
S-14	68.2	62.9	65.5	1.5	45.2	41.0	43.0	1.1
S-15	72.9	65.2	68.7	2.8	53.3	48.4	50.3	1.9
S-16	63.8	59.2	61.9	1.7	53.4	49.5	51.5	1.3
S-17	66.5	60.6	64.0	1.8	52	47.8	50.7	1.2
S-18	64.9	58.2	62.3	2.0	54.2	48.5	52.3	1.6
S-19	66.4	61.5	63.6	1.3	54.1	50.3	52.5	1.0
S-20	75.1	69.9	72.7	1.8	70.5	66.8	68.3	1.2

Average is denoted by Avg, Standard deviation is denoted by Std Dev and Solution is denoted by Soln

Table 17 Simulation results in terms of F-measure and INF of CRO with repair function

Seq	F-measure				INF			
	Best Soln.	Worst Soln.	Avg Soln.	Std. Dev.	Best Soln.	Worst Soln.	Avg. Soln.	Std. Dev.
S-1	93.3	87.6	90.2	1.8	93.29	87.67	90.27	1.99
S-2	95.7	86.5	92.1	2.6	95.81	86.97	92.35	2.47
S-3	96.2	86.8	90.9	2.7	96.19	86.93	91.01	3.13
S-4	98.6	94.4	95.8	1.1	98.64	94.55	95.97	0
S-5	96.2	90.9	93.7	1.8	97.47	89.87	93.77	2.24
S-6	92.3	85.3	87.9	2.1	93.49	85.51	88.65	2.76
S-7	77.4	66.9	71.7	3.3	77.46	66.94	71.73	3.33
S-8	73.2	63.6	68.2	3.1	73.21	63.43	68.38	3.15
S-9	72.4	61.6	65.8	3.3	72.72	61.89	66.05	3.35
S-10	74.9	63.7	67.4	3.3	75.07	62.28	67.62	3.61
S-11	71.1	61.7	67.5	5.1	71.51	63.01	68.27	3.97
S-12	71.5	64.1	69.1	3	72.08	64.49	68.74	2.68
S-13	64.5	56.1	61.8	4.3	65.18	56.74	61.45	2.93
S-14	54.4	49.6	51.9	1.3	55.52	50.78	53.07	1.28
S-15	62.1	55.5	58.4	2.2	62.33	56.18	58.78	2.31
S-16	58.1	53.9	56.2	1.5	58.37	54.13	56.46	1.49
S-17	58.3	53.4	56.6	1.4	58.8	53.82	56.96	1.47
S-18	59	52.9	56.8	1.7	59.31	53.13	57.08	1.79
S-19	59.6	55.3	57.5	1.1	59.94	55.62	57.78	1.14
S-20	72.7	68.3	70.4	1.4	72.76	68.33	70.47	1.47

Average is denoted by Avg, Standard deviation is denoted by Std Dev and Solution is denoted by Soln

Table 18 Simulation results in terms of Sensitivity and Specificity of CRO without repair function

Seq	Sensitivity				Specificity			
	Best Soln.	Worst Soln.	Avg Soln.	Std. Dev.	Best Soln.	Worst Soln.	Avg. Soln.	Std. Dev.
S-1	92.1	78.9	85.9	4.2	92.1	90.9	93.3	2.3
S-2	91.8	81.1	85.9	4.8	100	96.7	98.2	1.6
S-3	92.5	85	89	3.3	97.4	91.8	95.6	2.4
S-4	89.4	84.21	87.4	2.8	100	100	100	0
S-5	92.5	82.5	88.5	3.7	97.3	92.1	93.3	2.2
S-6	90	85	86.5	2.2	100	91.9	94.5	3.3
S-7	79.2	67.5	72.8	3.4	74.8	66.9	69.9	1.9
S-8	74.6	71.4	73.4	1.4	66.7	62.4	63.8	1.5
S-9	75.6	66	72.5	3.9	61.7	57.1	61.4	2.4
S-10	75.2	69	72.2	2	62.5	58.2	61.9	2.2
S-11	80.1	68.7	75	6.1	65.6	56	61.4	4.5
S-12	81.7	71.7	78	3.9	63.6	58	62.1	2.4
S-13	72.7	66.1	72.2	4.8	54.3	48.7	54.1	3.9
S-14	67.2	60.3	64	2.9	43.8	40.2	42.2	1.5
S-15	71.6	62.6	66	2.9	52.7	47.5	49.8	1.9
S-16	63.5	60	62.8	2.7	53.6	50	52.6	1.9
S-17	66.5	62.2	63.7	1.5	51.8	49.5	51	1.9
S-18	63.8	59.7	62.6	2.3	53.4	50.3	52.3	1.8
S-19	64.9	61.5	65.1	2.1	53.1	51	53.3	1.4
S-20	73.2	69	70.9	1.9	68.2	66.5	67.1	0.7

Average is denoted by Avg, Standard deviation is denoted by Std Dev and Solution is denoted by Soln

Table 19 Simulation results in terms of F-measure and INF of CRO without repair function

Seq	F-measure				INF			
	Best Soln.	Worst Soln.	Avg Soln.	Std. Dev.	Best Soln.	Worst Soln.	Avg. Soln.	Std. Dev.
S-1	92.1	84.5	89.5	2.8	92.1	84.69	89.52	3.11
S-2	95.7	88.2	91.6	2.9	95.81	88.56	91.84	2.77
S-3	94.9	88.1	92.2	2.8	94.92	88.33	92.24	2.81
S-4	94.4	91.4	93.2	1.6	94.55	91.77	93.49	0
S-5	94.9	88	90.9	2.5	94.87	87.17	90.87	2.85
S-6	94.7	88.3	90.3	2.6	94.87	88.38	90.41	2.69
S-7	76.9	67.8	71.4	2.1	76.97	67.2	71.34	2.54
S-8	70.4	67.1	68.3	1.4	70.54	66.75	68.43	1.45
S-9	67.9	61.2	66.5	3	68.3	61.39	66.72	3.06
S-10	68.2	63.1	66.6	2	68.56	63.37	66.85	2.1
S-11	72.2	61.7	67.5	5.1	72.49	62.03	67.86	5.24
S-12	71.5	64.1	69.1	3	72.08	64.49	69.6	3.06
S-13	62.2	56.1	61.8	4.3	62.83	56.74	62.5	4.33
S-14	53	48.3	50.8	1.9	54.25	49.23	51.97	2.09
S-15	61.1	54	57	2.3	61.43	54.53	57.33	2.35
S-16	58	54.5	57.3	2.2	58.34	54.77	57.47	2.26
S-17	58.2	55.2	56.7	1.7	58.69	55.49	57	1.69
S-18	58.2	54.6	57	2	58.37	54.8	57.22	2.03
S-19	58.4	55.8	58.6	1.6	58.7	56	58.91	1.71
S-20	70.5	67.9	69	1.2	70.66	67.74	68.97	1.15

Average is denoted by Avg, Standard deviation is denoted by Std Dev and Solution is denoted by Soln

Fig. 9 The graph for the comparison of the CRO without repair function, CRO with repair function, GA, SA, COIN and TL-PSO in terms of (a) sensitivity, (b) specificity and (c) F-measure

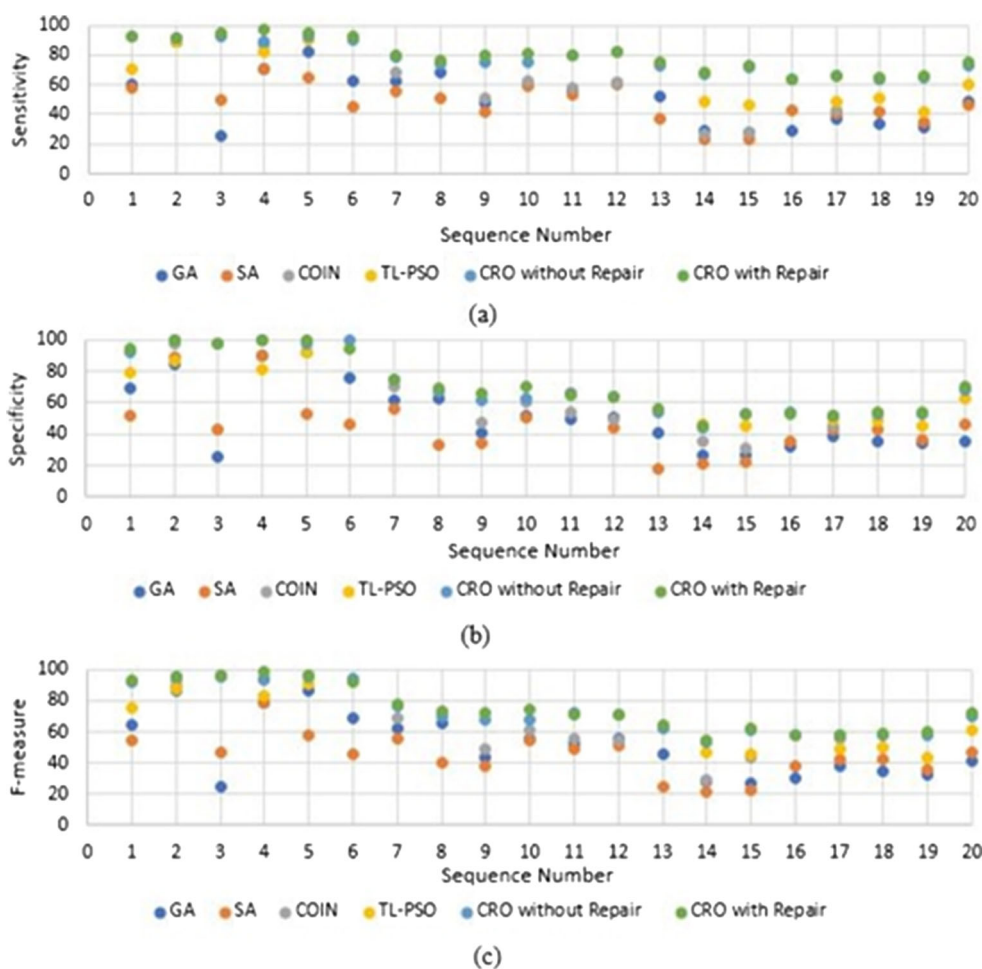


Table 20 Execution time of CRO, GA [7] and SA [2] (in seconds)

Seq No.	Length	Execution time			
		GA	SA	CRO without repair	CRO with repair
S-1	117	98.07	398.23	59.37	51.52
S-2	118	101.89	401.34	86.74	81.81
S-3	120	99.09	397.56	73.21	69.05
S-4	122	102.66	479.08	68.6	62.29
S-5	123	120.45	481.98	111.17	114
S-6	124	123.55	490	92.00	74.75
S-7	394	604.67	502.78	121.34	107.21
S-8	454	689.78	509.33	112.33	89.56
S-9	456	670.72	511.48	145.32	98.45
S-10	468	703.45	508.22	642.74	490.49
S-11	543	750.67	630.07	603.3	385.44
S-12	556	748.89	527.93	445.45	390.45
S-13	605	790.45	550.3	479.34	269.1
S-14	697	905.7	578.25	523.76	211.41
S-15	784	922.67	705.98	673.34	523.59
S-16	940	1200.78	708.4	678.5	497.73
S-17	945	1178.66	700.71	690	458.89
S-18	954	1223.35	711.2	659.4	5 499.08
S-19	964	1267.56	809.33	734.45	567.98
S-20	1492	3598.33	3002.48	2519.45	1166.24

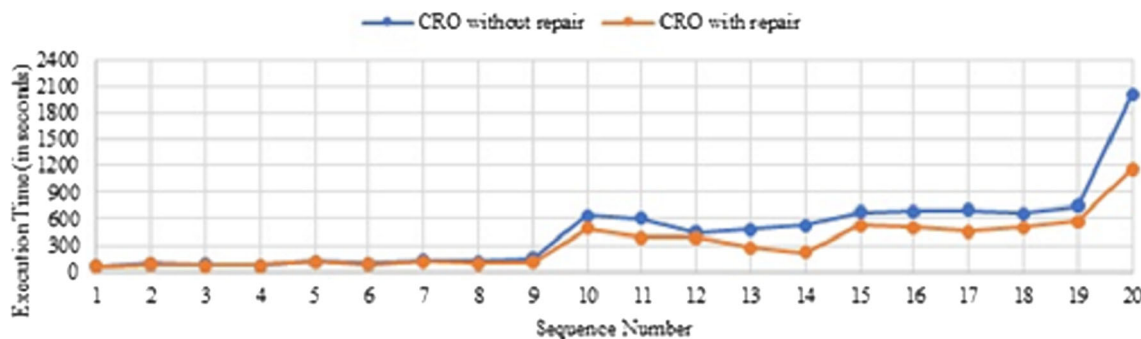


Fig. 10 Line graph for the comparison of execution time of the CRO without repair function and with the repair function

been tested for the same dataset. Table 14 depicts the results of CRBA and CRO in terms of TP, FP and FN. We have also shown the results of sensitivity, specificity, and F-measure in the Table 15. For all ten sequences, the outcomes of CRO are more appreciable than CRBA.

The simulation of the best, the worst, the average solutions, and the standard deviation of the sensitivity, specificity, and F-measure of each sequence by the proposed method are given. In Tables 16 and 17 it shows the same outcome in terms of F-measure and INF. Tables 16 and 17 show the results of CRO with repair function and Tables 18

Table 21 CRO is compared with the SA [2], TL-PSO [19], RNAfold [31] in terms of Minimum Free Energy (in KCAL/MOL)

Seq No.	Minimum free energy			
	SA	TL PSO	RNAfold	CRO
S-1	-47.04	-47.2	-48.1	-57.85
S-2	-57.52	-47.4	-48.3	-48.89
S-3	-60.54	-	-	-64.14
S-4	-54.94	-53.4	-53.4	-68.8
S-5	-58.32	-55.8	-57.1	-69.87
S-6	-66.6	-	-	-70.56
S-7	-158.54	-	-	-173.24
S-8	-178.39	-	-	-211.4
S-9	-185.74	-	-	-189.68
S-10	-182.06	-	-	-183.15
S-11	-223.45	-	-	-246.84
S-12	-208.45	-	-	-207.32
S-13	-197.23	-	-	-217.05
S-14	-175.82	-119.2	-126.8	-255.31
S-15	-201.11	-132.0	-136.4	-253.59
S-16	-228.3	-	-	-344.21
S-17	-305.83	-221.4	-233.1	-380.63
S-18	-299.66	-207.4	-222.8	-383.63
S-19	-284.02	-176.8	-196	-353.97
S-20	-709.08	-702.8	-779.6	-784.91

Table 22 Secondary structure comparison of Saccharomyces cerevisiae

Base-pair form by CRO		Benchmark base-pair form	
i	j	i	j
1	117	1	117
2	116	2	116
3	115	3	115
4	114	4	114
5	113	5	113
		6	112
7	111	7	111
8	110	8	110
9	109	9	109
14	64	14	64
15	63	15	63
16	62	16	62
17	61	17	61
18	60	18	60
19	59	19	59
20	58	20	58
21	57	21	57
26	53	26	53
27	52	27	52
28	51	28	51
29	48	29	48
30	47	30	47
31	46	31	46
32	45	32	45
		66	107
67	106	67	106
68	105	68	105
69	104	69	104
70	103	70	103
78	97	78	97
79	96	79	96
80	95	80	95
81	94	81	94
		82	93
83	92	83	92
84	91	84	91
85	90	85	90

Table 23 Sensitivity comparison by Kruskal-Wallis test followed by post-hoc analysis

	GA	SA	COIN	TL-PSO	CRO
GA	—	⊗	⊗	⊗	★
SA		—	⊗	★	★
COIN			—	⊗	★
TL-PSO				—	★
CRO					—

The sign ★ represents significant and ⊗ represents not significant

and 19 show the results of CRO without repair function. From these tables, it can be seen that there are enough changes in the results while using the repair function. Repair function improves the results of the sensitivity, specificity, F-measure, and INF.

Figure 9 represents the graphs of the comparison of the proposed methods of GA and SA, COIN and TL-PSO in terms of sensitivity, specificity, and F-measure. From the graphs, it is proved that CRO outperforms the mentioned four methods. We have also implemented two algorithms of GA and SA to prove the time efficiency of our proposed work. After implementing two mentioned algorithms, it was concluded that the CRO takes less time in predicting RNA structure than GA and SA. The comparisons of execution time between the proposed method and the methods of GA as well as SA are demonstrated in Table 20. The execution time is shown in seconds in the table. The execution time of CRO with and without repair function CRO is given in Table 20 to show that the proposed method with repair function is faster than without repair function. Figure 10 represents the line graph which shows the comparison of the execution time of the proposed method (CRO) with and without the repair function.

In Table 21 the proposed method (CRO) is compared with the methods of SA and RNAfold [31] and TL-PSO method in terms of Minimum Free Energy (in KCAL/MOL). The values of minimum free energy in method RNAfold were given in the paper of TL-PSO [19], where RNAfold gives the best results than TL-PSO. So, we

Table 24 Specificity comparison by Kruskal-Wallis test followed by post-hoc analysis

	GA	SA	COIN	TL-PSO	CRO
GA	—	⊗	⊗	⊗	★
SA		—	★	⊗	★
COIN			—	⊗	★
TL-PSO				—	★
CRO					—

The sign ★ represents significant and ⊗ represents not significant

Table 25 F-Measure comparison by Kruskal-Wallis test followed by post-hoc analysis

	GA	SA	COIN	TL-PSO	CRO
GA	—	⊗	⊗	⊗	★
SA		—	⊗	⊗	★
COIN			—	⊗	★
TL-PSO				—	★
CRO					—

The sign ★ represents significant and ⊗ represents not significant

have compared the results of CRO and those of RNAfold in terms of minimum free energy to verify that our proposed method also outperforms the RNAfold in minimum free energy. As in the other two methods RNAPredict (GA) and COIN there were no minimum free energies of the tested sequences, so we could not provide their results. Any algorithm which has not tested for a particular sequence is denoted by — in the table. From the results, it is proved that CRO gives the lowest free energy as well as most stable structures for nineteen sequences than any other methods. For only one sequence the method of SA gives better result than CRO. However, the results of SA and CRO are very close.

Here, in Table 22 for the sequence, *Saccharomyces cerevisiae*, the known base pairs and the predicted base pairs of the proposed method is shown. The *Saccharomyces cerevisiae* has 38 base pairs. The primary sequence of *Saccharomyces cerevisiae* is given below. Both the known secondary structure of *Saccharomyces cerevisiae* and the predicted structure are given in dot parenthesis form in Table 17. In the table, *i* denotes the index of the opening parenthesis and *j* denotes the index of the closing parenthesis index. The pairs (6-112), (66-107) and (82-93) are not predicted by the proposed method. That is why the base pairs are called false negative (FN) base pairs. They are shown in bold text. Tables 23, 24, 25, and 26 represent the results of significance testing using Kruskal-Wallis test followed by post-hoc analysis to confirm that, the proposed method significantly outperforms the algorithms of GA, SA,

Table 26 INF comparison by Kruskal-Wallis test followed by post-hoc analysis

	GA	SA	COIN	TL-PSO	CRO
GA	—	⊗	⊗	⊗	★
SA		—	⊗	⊗	★
COIN			—	⊗	★
TL-PSO				—	★
CRO					—

The sign ★ represents significant and ⊗ represents not significant

COIN and TL-PSO. The results of every algorithm for all sequences are not available. For example, *T. aquaticus* has been used in TL-PSO and CRO (the proposed method) but it was not used in SA.

So, when the significance was tested, the groups were formed by removing the unavailable sequences. The significance has been tested for the results of sensitivity, specificity, F-measure, INF. The results of sensitivity, specificity, F-measure and INF are compared in Tables 23–26 where the lower portion has been removed to avoid repetition. From the comparison, it is verified that the proposed method outperforms all other algorithms.

Sequence: GGUUGCGGCCAUUACUACCAGAAAG-CACCGUUUCCCG
UCCGAUCAACUGUGUUAAGCUGGUAGAGCCUGA
CCGAGUAGUGUAUGGGUGACCAUACGCGAAACU
CAGGUGCU GCAAUCU

Secondary structure by CRO:

Dot-parenthesis form: [(((((((.....(((((((.....(((((((.....
.....)))))).....)))))).....)))))).....)]

Benchmark secondary structure:

Dot-parenthesis form: [(((((((.....(((((((.....(((((((.....
.....)))))).....)))))).....)))))).....)]

6 Conclusion

Over the past decade, prediction of RNA structure achieved vital attention to researchers, as it has a crucial role in molecular biology. This work presents the popular optimization problem of RNA secondary structure prediction which is implemented using population-based metaheuristic algorithm called Chemical Reaction Optimization (CRO). The operators of CRO have been redesigned to solve the RSP problem using CRO algorithm. The hard tasks during the implementation of the problem were calculating the free energy of the structure and maintaining the accuracy of the predicted structure which depends on the value of TP (True base pairs), FP (False positive base pairs) and FN (False negative base pairs). The results of the RNA structure prediction problem using CRO algorithm are compared with the related metaheuristic algorithms such as RNAPredict, the SARNA-Predict, COIN, TL-PSOfold and CRBA and the results show that the proposed RNA structure prediction problem using CRO algorithm performs very well and gives more stable structures. The additional repair function helps to predict RNA structure in less time by eliminating the duplicate stem from the resultant solution.

We have not done any experiment with the pseudoknotted RNA sequences because of its complication in estimating

the structures. In future work, the experiment of the RNA with pseudoknots could be considered.

Compliance with Ethical Standards

Conflict of interests The authors have no conflict of interest.

References

1. Ray SS, Pal SK (2013) RNA Secondary structure prediction using soft computing. *IEEE/ACM Trans Comput Biol Bioinform* 10:2–17
2. Grypma P, Tsang HH (2014) SARNA-Predict: using adaptive annealing schedule and inversion mutation operator for RNA secondary structure prediction. In: 2014 IEEE symposium on computational intelligence in multi-criteria decision-making (MCDM). IEEE, pp 150–156
3. McMellan N (2006) RNA secondary structure prediction using ant colony optimisation. Master Thesis, School of Informatics, University of Edinburgh
4. Zuker M (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res* 31:3406–3415
5. Eddy SR (2004) How do RNA folding algorithms work? *Nat Biotechnol* 22:1457
6. Schmitz M, Steger G (1996) Description of RNA folding by simulated annealing. *J Mol Biol* 255:254–266
7. Wiese KC, Deschenes AA, Hendriks AG (2008) Rnapredict—an evolutionary algorithm for RNA secondary structure prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)* 5:25–41
8. Geis M, Middendorf M (2007) A particle swarm optimizer for finding minimum free energy RNA secondary structures. In: *Swarm intelligence symposium, 2007. SIS 2007. IEEE*. IEEE, pp 1–8
9. Neethling M, Engelbrecht AP (2006) Determining RNA secondary structure using set-based particle swarm optimization. In: *IEEE congress on evolutionary computation, 2006. CEC 2006. IEEE*, pp 1670–1677
10. Liu Y, Dong H, Zhang H et al (2011) Prediction of RNA secondary structure based on particle swarm optimization. *Chem Res Chin Univ* 27:108–112
11. Xing C, Wang G, Wang Y et al (2011) A novel method for RNA secondary structure prediction. In: 2011 7th international conference on natural computation (ICNC). IEEE, pp 1136–1140
12. Xing C, Wang G, Wang Y et al (2012) PSOFold: a metaheuristic for RNA folding. *J Comput Inf Syst* 8:915–923
13. Srikamdee S, Wattanapornprom W, Chongstitvatana P (2016) RNA secondary structure prediction with coincidence algorithm. In: 16th international symposium on communications and information technologies (ISCIT), 2016. IEEE, pp 686–690
14. Truong TK, Li K, Xu Y (2013) Chemical reaction optimization with greedy strategy for the 0–1 knapsack problem. *Appl Soft Comput* 13:1774–1780
15. Lam AY, Li VO (2010) Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans Evol Comput* 14:381–399
16. Lam AY, Li VO (2010) Chemical reaction optimization for cognitive radio spectrum allocation. In: *Global telecommunications conference (GLOBECOM 2010), 2010 IEEE*. IEEE, pp 1–5
17. Pan B, Lam AY, Li VO (2011) Network coding optimization based on chemical reaction optimization. In: *Global telecommunications conference (GLOBECOM 2011), 2011 IEEE*. IEEE, pp 1–5
18. Truong TK, Li K, Xu Y et al (2013) An artificial chemical reaction optimization algorithm for multiple-choice knapsack

problem. In: Proceedings on the international conference on artificial intelligence (ICAI). The Steering Committee of the World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp), p 1

19. Lalwani S, Kumar R, Gupta N (2016) An efficient two-level swarm intelligence approach for RNA secondary structure prediction with bi-objective minimum free energy scores. *Swarm Evol Comput* 27:68–79
20. Mizuno H, Sundaralingam M (1978) Stacking of Crick Wobble pair and Watson-Crick pair: stability rules of GU pairs at ends of helical stems in tRNAs and the relation to codon-anticodon Wobble interaction. *Nucleic Acids Res* 5:4451–4462
21. Xia T, SantaLucia J Jr, Burkard ME et al (1998) Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* 37:14719–14735
22. Turner DH, Mathews DH (2009) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res* 38:D280–D282
23. Rivas E, Eddy SR (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots1. *J Mol Biol* 285:2053–2068
24. Liu Z, Zhao S, Ye H et al (2016) The PTAS of Prediction for RNA Pseudoknotted Structure. In: 12th international conference on computational intelligence and security (CIS), 2016. IEEE, pp 1–4
25. Akutsu T (2000) Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discret Appl Math* 104:45–62
26. Cannone JJ, Subramanian S, Schnare MN et al (2002) The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics* 3:2
27. Tong K-K, Cheung K-Y, Lee K-H, Leung K-S (2013) GAKnot: RNA secondary structures prediction with pseudoknots using Genetic Algorithm. In: 2013 IEEE symposium on computational intelligence in bioinformatics and computational biology (CIBCB). IEEE, pp 136–142
28. Lam AY, Li VO (2012) Chemical reaction optimization: a tutorial. *Memetic Computing* 4:3–17
29. Islam MR, Saifullah CK, Asha ZT, Ahamed R (2018) Chemical reaction optimization for solving longest common subsequence problem for multiple string. *Soft Comput*, pp 1–25
30. Andronescu M, Bereg V, Hoos HH, Condon A (2008) RNA STRAND: the RNA secondary structure and statistical analysis database. *BMC Bioinformatics* 9:340
31. Lorenz R, Bernhart SH, Zu Siederdisen CH et al (2011) ViennaRNA package 2.0. *Algorithms Mol Biol* 6:26
32. Cui Z, Cao Y, Li F, Zhu Z (2015) Changing range bat algorithm for RNA secondary structure prediction. *J Comput Theor Nanosci* 12:1968–1971
33. Miao Z, Westhof E (2017) RNA Structure: advances and assessment of 3D structure prediction. *Annu Rev Biophys* 46:483–503



Rayhanul Kabir currently working as Software Engineer in Data Path Ltd in Bangladesh. He has completed his B.Sc. in Computer Science and Engineering from Khulna University, Bangladesh. He has several publications on different international conference and journal. His research interest includes Bio-informatics, Information Security, Data mining etc.



Rafiqul Islam obtained Ph.D. in Computer Science from Universiti Teknologi Malaysia (UTM) in 1999 and a combined Master (MS) and Bachelor Degree in Engineering (Computers) from Azerbaijan Polytechnic Institute (Azerbaijan Technical University at present) in 1987. He was a visiting fellow (a post doctoral researcher) in Japan Advance Institute of Science and Technology (JAIST) in 2001. He worked as head of the Discipline of

Computer Science and Engineering of Khulna University and as the Dean of the School of Science, Engineering and Technology of Khulna University. He worked as a Professor in the Department of Computer Science of American International University-Bangladesh (AIUB) in Dhaka about five years. Currently he is working as a Professor of Computer Science and Engineering Discipline of Khulna University. He has 24 years of teaching and research experiences. He has published about 90 papers, which have been published in international and national journals as well as in referred international conference proceedings published by IEEE, Springer and others. He has reviewed papers submitted to International journals of IEEE, Wiley, Springer; papers submitted to journals of different universities of Bangladesh and papers submitted to several International Conferences. His research areas include design and analysis of algorithms in the area of cloud computing, external sorting, Information security, data compression, bioinformatics, information retrieval, grid computing and data mining etc.