An improved chemical reaction-based approach for multiple sequence alignment

Rohit Kumar Yadav* and Haider Banka

Department of Computer Science and Engineering, Indian Institute of Technology (Indian School of Mines), Dhanbad 826 004, India

In bioinformatics, multiple sequence alignment (MSA) is an NP-hard problem. Nature-inspired approaches can provide an approximate solution compared to conventional approaches. In this article, the MSA problem is dealt with using chemical reaction optimization (CRO). The limitations of CRO are slow convergence and low population diversity. Therefore, the initialization process is improved by pairwise alignment technique which maintains diversity. In the performance analysis, we have taken benchmark datasets from Bali base version 2.0. The Bali score of the proposed approach is compared with those of the existing approaches such as SB-PIMA, SAGA, RBT-GA and GAPAM, HMMT. Simulation results confirm the superiority of the proposed approach over others.

Keywords: Bioinformatics, chemical reaction optimization, multiple sequence alignment, population diversity.

MULTIPLE sequence alignment (MSA) is an alignment problem where more than three amino acids or protein sequence participate in the alignment. We can solve many biological problems with the help of MSA. It is useful to suggest primary and secondary structures of RNAs and proteins[1,2]. MSA can reconstruct phylogenetic tree. We can also predict the function of unknown amino acid sequences using the phylogenetic tree. MSA can find similarity between sequences, which is helpful to know the function and structure of similar protein or amino acid sequences[3,4]. However, MSA maximizes the matching component as well as minimizes the mismatching component, which is problematic. There are many problems in bioinformatics such as finding ancestral and hereditary relationships, which can be solved by MSA. Hence, the MSA problem must be dealt with in the efficient manner.

In 1970, Needleman and Wunsch[5] proposed an algorithm using dynamic programming (DP) to solve the MSA problem. This algorithm is useful to solve pairwise sequence alignment problem. DP can also solve the MSA problem and give an optimal solution. However, the process is time-consuming, especially when the number and length of sequences are large. Hence the MSA problem becomes NP-hard and it is not feasible to use DP to solve the MSA problem. We need to develop new algorithms for the same.

*For correspondence. (e-mail: rohit.ism.123@gmail.com)

The MSA problem can be solved in a systematic way by progressive alignment. This approach is less complex in terms of time and space for solving the MSA problem[6,7]. This method has used repeatedly Needleman and Wunsch algorithm to find guide tree between MSA. The progressive alignment method initially aligns more similar sequences, after which it incrementally aligns more dissimilar sequences or groups of sequences in the initial alignment. CLUSTAL W is the standard representation of the progressive method[8]. In the first step, we assign weights to every pair of sequences in a partial alignment. We assign small weights for the most similar sequences and big weights for most divergent sequences. Next, we take a substitution matrix which defines the score between two residues of a protein sequence based on similarity. Two types of gaps are introduced in the third step. The first is residue-specific gap and the second is locally residue gap penalties. These steps are integrated into CLUSTAL W, which is freely available. Progressive alignment method performs better for MSA package in terms of accuracy and time. However, this method has some limitations – dependency on initial alignment and choice of scoring scheme. In other words, we need to align more similar sequences in the initial stage. If not, the solution may be trapped in local optima.

An iterative method initially starts with a random solution and improves the solution in an iterative manner until no more upgradation is possible. In this case, the result does not depend on the initial population. The main aim of this method is to find the global optimum. In order to solve the MSA problem, the objective of the iterative method is to find an alignment which is the globally optimal alignment. Simulated annealing is an example of the iterative process. Hidden Markov Model Training (HMMT) method is based on simulated annealing process[9]. The drawback of the iterative method is that the solution may be trapped in local optima. So researchers have switched to another method for solving the MSA problem, which is evolutionary or nature-inspired method.

The evolutionary method starts with a random initial population. In the second step, we calculate fitness value of each solution using an objective function. In the third step, we modify the initial solution using some operators and continuously use this operator until we reach the global optimum. In this method, the initial solution is

originated in a random way, after which we apply evolutionary operators to enhance the similarity of MSA. Some algorithms available are based on evolutionary computations for MSA[10–14], while others are based on genetic algorithms for MSA such as SAGA[14], MSA-GA[15], RBT-GA[16] and genetic algorithm with progressive alignment method (GAPAM)[17]. In the case of GAPAM, Naznin *et al.*[17] have taken 56 different types of datasets from reference sets 1–5. The limitation of these evolutionary-based algorithms is that the result may be trapped in local optima.

An improved chemical reaction optimization (ICROMSA) has been proposed to solve the MSA problem. We have developed a technique to generate initial molecular structure which is helpful to converge to global optimal alignment. We have also compared the classical chemical reaction optimization (CRO) and ICROMSA with respect to some Bali base datasets and found that the latter can perform better in most cases.

Basics of CRO

Most of the swarm intelligence techniques developed earlier are based on constant population, but CRO is based on variable length population[18]. In CRO, a solution is represented by the structure of a molecule. A molecule has two types of energy – potential energy (PE) and kinetic energy (KE). The structure of a molecule is represented by its PE. The motion of a molecule is defined by its KE. PE function is considered as a quality of the molecule, i.e. fitness function. PE of a molecule can be expressed as an objective function as follows

$$PE_z = f(x). \quad (1)$$

where PE_z means potential energy of molecular structure z and $f(x)$ is a fitness function.

The potential energy of a molecular structure is the fitness value of a molecule. Objective function is defined by f and molecular structure is represented by z. For example, suppose a molecule changes its structure from z to z', this is only possible if $PE_z \geq PE_{z'}$ or $PE_z + KE_z \geq PE_{z'}$. Kinetic energy (KE) of a molecule defines the degree of local optimum. There are mainly two types of collision among molecules – uni-molecular collision and inter-molecular collision. Uni-molecular reaction can be divided into two types – on-wall ineffective collision and decomposition. Inter-molecular collision can also be categorized into two types – inter-molecular ineffective collision and synthesis.

On-wall inadequate reaction

Here, molecules hit the wall and return but there is a change in some of the molecular properties. Figure 1 graphically explains this collision.

Suppose the present molecular structure is z and the modified molecular structure is z'. Then this change is only possible if

$$PE_z + KE_z \geq PE_{z'} \quad (2)$$

since we know that this collision is not much more vigorous. Hence the difference between actual molecule and resultant molecule is small. We get $KE_z = (PE_z + KE_z - PE_z) \times p$, where p lies between [KELossRate, 1], the range of KELossRate is 0 to 1 and $(PE_z + KE_z - PE_{z'}) \times (1 - p)$ is the amount of energy lost when the molecule hits the wall. We keep this energy in buffer, which can be used for decomposition reaction.

Algorithm 1. On-wall-ineffective (K, buffer)
Input: Molecule K and buffer
1. Find β_1 = Neighbour (β)
2. Calculate $PE_{\beta 1}$
3. ***If*** $(PE_\beta + KE_\beta \geq PE_{\beta 1})$
4. Find r a random number between [KE LOSS Rate, 1]
5. $KE_{\beta 1} = (PE_\beta + KE_\beta - PE_{\beta 1}) \times r$
6. Upgrade buffer = buffer + $(PE_\beta + KE_\beta - PE_{\beta 1}) \times (1 - r)$
7. upgrade the molecular structure of K by $\beta = \beta 1$
8. ***end if***
9. ***output*** K and buffer

Decomposition

Here, the molecules hit the wall and convert into two or more components. This collision is robust and the structure of resultant molecules is different from the actual molecule. Figure 2 provides a graphical representation of this collision.

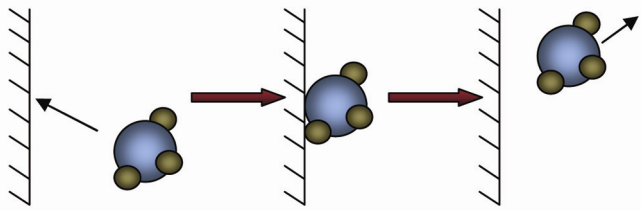

Figure 1. On-wall ineffective collision.

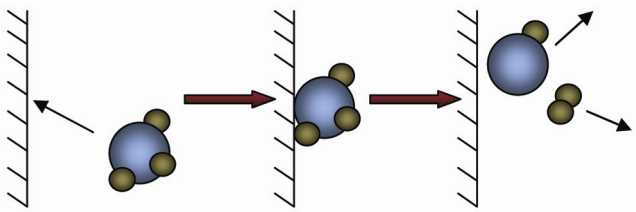

Figure 2. Decomposition.

The condition for decomposition is (NHits[z] – MHits[z]) > α. In this collision, the shape of the original molecule is z while those of the resultant molecules are z_1' and z_2'. Suppose the original molecule has more energy (PE + KE) to capitalize the resulting molecules of PE of then the change is allowed as follows

$$\mathrm{PE}_z + \mathrm{KE}_z \geq \mathrm{PE}'_{z_1} + \mathrm{PE}'_{z_2}, \tag{3}$$

Let $\mathrm{temp}_1 = \mathrm{PE}_z + \mathrm{KE}_z - \mathrm{PE}'_{z_1} - \mathrm{PE}'_{z_2}$.

Therefore, $\mathrm{KE}'_{z_1} = \mathrm{temp}_1 \times q$ and $\mathrm{KE}'_{z_2} = \mathrm{temp}_1 \times (1 - q)$, where q is randomly generated from the interval [0, 1]. Since potential energy of z, z_1 and z_2 is approximately the same. Hence, when kinetic energy of molecule z is very large, then only eq. (3) satisfied. But according to properties of on-wall ineffective reaction, kinetic energy of a molecule always decreases. Hence we have drawn some energy from buffer to favour for satisfying the criteria of decomposition collision. Due to this reason, some energy is drawn from buffer for satisfying the criteria of decomposition.

$$\mathrm{PE}_z + \mathrm{KE}_z + \mathrm{buffer} \geq \mathrm{PE}'_{z_1} + \mathrm{PE}'_{z_2}. \tag{4}$$

When eq. (4) holds then we can get

$$\mathrm{KE}'_{z_1} = (\mathrm{temp}_1 + \mathrm{buffer}) \times q_1 \times q_2, \tag{5}$$

$$\mathrm{KE}'_{z_2} = (\mathrm{temp}_1 + \mathrm{buffer} - \mathrm{KE}'_{z_1}) \times q_3 \times q_4, \tag{6}$$

where q_1, q_2, q_3 and q_4 are randomly generated from the interval [0, 1]. Since the buffer already stores a large amount of energy, we multiply two random numbers in both eqs (5) and (6) to ensure that KE'_{z_1} and KE'_{z_2} are not too large.

Also, buffer = buffer + $\mathrm{temp}_1 - \mathrm{KE}'_{z_1} - \mathrm{KE}'_{z_2}$. If eqs (3) and (4) are not satisfied, the decomposition reaction does not hold and the molecule has its original structure z.

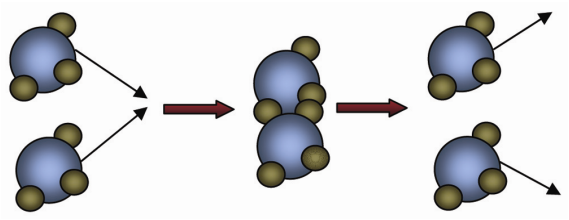

Figure 3. Intermolecular ineffective collision.

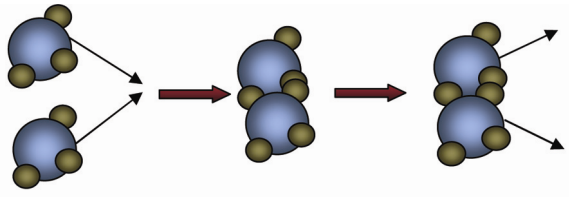

Figure 4. Synthesis collision.

Algorithm 2. Decompose (K, buffer)

Input: Molecule K and central energy

1. Obtain $\beta 1$ and $\beta 2$ from β
2. Determine $\mathrm{PE}_{\beta 1}$ and $\mathrm{PE}_{\beta 1}$
3. Let temp = $\mathrm{PE}_\beta + \mathrm{KE}_\beta - \mathrm{PE}_{\beta 1} - \mathrm{PE}_{\beta 1}$
4. Generate a Boolean variable
5. ***If*** (temp ≥ 0)
6. Boolean = TRUE
7. Find a random number p between [0, 1]
8. $\mathrm{KE}_{\beta 1} = \mathrm{temp} \times p$
9. $\mathrm{KE}_{\beta 2} = \mathrm{temp} \times (1 - p)$
10. Determine new molecule K_1 and K_2
11. Assign $\beta 1$, $\mathrm{PE}_{\beta 1}$, and $\mathrm{KE}_{\beta 1}$ to the molecular structure of K_1 and $\beta 2$, $\mathrm{PE}_{\beta 2}$ and $\mathrm{KE}_{\beta 2}$ to the molecular structure of K_2
12. ***else if*** (temp + buffer ≥ 0)
13. Boolean = TRUE
14. Get k_1, k_2, k_3 and k_4 randomly in meanwhile [0, 1]
15. $\mathrm{KE}_{\beta 1} = (\mathrm{temp} + \mathrm{buffer}) \times k_1 \times k_2$
16. $\mathrm{KE}_{\beta 2} = (\mathrm{temp} + \mathrm{buffer} - \mathrm{KE}_{\beta 1}) \times k_3 \times k_4$
17. Upgrade buffer = temp + buffer – $\mathrm{KE}_{\beta 1} - \mathrm{KE}_{\beta 2}$
18. Assign $\beta 1$, $\mathrm{PE}_{\beta 1}$ and $\mathrm{KE}_{\beta 1}$ to the molecular structure of K_1 and $\beta 2$, $\mathrm{PE}_{\beta 2}$ and $\mathrm{KE}_{\beta 2}$ to the molecular structure of K_2
19. ***else***
20. Boolean = FALSE
21. ***end if***
22. ***output*** K_1, K_2, Boolean and buffer.

Inter-molecular ineffective reaction

Here two or more molecules collide with each other and then return. There is little change in the result. This is similar to the on-wall inadequate reaction. The difference between the two reactions is only the number of molecules. In the first case only one molecule participates in the collision, whereas in the second case two or more molecules participate. In this collision KE is not drawn from the central buffer; there is only interchange among the molecules. Figure 3 is an example of inter-molecular ineffective collision.

Suppose z_1 and z_2 are two original molecules and after the inter-molecular ineffective collision we get two different molecular structures z_1' and z_2'. Since this collision is not vigorous, the molecular structures of z_1' and z_2' are not much distinct from the original molecules z_1 and z_2. The change is only possible if the following condition is satisfied.

$$\mathrm{PE}_{z_1} + \mathrm{KE}_{z_1} + \mathrm{PE}_{z_2} + \mathrm{KE}_{z_2} \geq \mathrm{PE}'_{z_1} + \mathrm{PE}'_{z_2}. \tag{7}$$

Let $\mathrm{temp}_2 = (\mathrm{PE}_{z_1} + \mathrm{KE}_{z_1} + \mathrm{PE}_{z_2} + \mathrm{KE}_{z_2} - \mathrm{PE}'_{z_1} - \mathrm{PE}'_{z_2})$, Then, $\mathrm{KE}'_{2_1} = \mathrm{temp}_2 \times r$ and $K'_{z_2} = \mathrm{temp}_2 \times (1 - r)$, where r is a random number between [0, 1].

Algorithm 3. Intermolecular ineffective (K_1, K_2)
Input: molecules K_1, K_2 with their profile
1. Find $\beta 1^1$ = Neighbour ($\beta 1$) and $\beta 2^1$ = Neighbour ($\beta 2$)
2. Determine $PE_{\beta 1}{}^1$ and $PE_{\beta 2}{}^1$
3. Let temp = ($PE_{\beta 1}$ + $PE_{\beta 2}$ + $KE_{\beta 1}$ + $KE_{\beta 2}$) – ($PE_{\beta 1}{}^1$ + $PE_{\beta 2}{}^1$)
4. ***If*** (temp ≥ 0)
5. Find a random number q between [0, 1]
6. $KE_{\beta 1}{}^1$ = temp × q
7. $KE_{\beta 2}{}^1$ = temp × (1 – q)
8. update the profile of K_1 by $\beta 1 = \beta 1^1$, $PE_{\beta 1} = PE_{\beta 1}{}^1$ and $KE_{\beta 1} = KE_{\beta 1}{}^1$ and the profile of K_2 by $\beta 2 = \beta 2^1$, $PE_{\beta 2} = PE_{\beta 2}{}^1$ and $KE_{\beta 2} = KE_{\beta 2}{}^1$
9. ***end if***
10. ***output*** K_1 and K_2.

Synthesis

In this reaction, two or more components collide with each other to form new components. In this collision change is much more effective. Hence the difference between original and resultant molecule is more. Figure 4 is an example of the synthesis reaction. If $KE_{z1} \le \lambda$ and $KE_{z2} \le \lambda$, this is favourable case for synthesis. In this reaction the original molecules are z_1 and z_2, and the resultant molecule is z'. This change is applicable if the following condition is satisfied

$$PE_{z1} + KE_{z1} + PE_{z2} + KE_{z2} \ge PE'_z, \quad (8)$$

Then we get

$$KE'_z = PE_{z1} + KE_{z1} + PE_{z2} + KE_{z1} - PE_z'. \quad (9)$$

If eq. (8) is not satisfied the molecules return to their original structures. In this case, PE does not change, but KE of the resultant molecule is larger than the original molecules. In this reaction, secure molecule has greater ability to escape from local optimum. Any one of the above reactions can hold in each iteration.

Algorithm 4. Synthesis (K_1, K_2)
Input: Molecules K_1 and K_2 with their profile.
1. Calculate β from $\beta 1$ and $\beta 2$
2. Determine PE_β
3. Generate a Boolean variable
4. Generate a new molecule K
5. ***If*** ($PE_{\beta 1}$ + $PE_{\beta 2}$ + $KE_{\beta 1}$ + $KE_{\beta 2}$ ≥ PE_β)
6. Boolean = TRUE
7. $KE\beta = PE_{\beta 1} + PE_{\beta 2} + KE_{\beta 1} + KE_{\beta 2} - PE_\beta$
8. Assign β, PE_β and KE_β to the profile K
9. ***Else***
10. Boolean = FALSE
11. ***end if***
9. ***output*** K and Boolean

There are mainly three stages in CRO – initialization, iteration, and final stage. Figure 5 shows a flow chart of the CRO algorithm.

Proposed method

In the proposed method, we have used a novel CRO (ICROMSA) to find an approximate solution to the MSA problem. Here the initialization process is improved compared to the basic CRO. ICROMSA generates the new solution using elementary reactions such as on-wall ineffective, synthesis, inter-molecular ineffective and decomposition.

Initialization scheme

Here, we have considered the Needleman and Wunsch[5] algorithm to generate the initial population. The process of initialization is shown in Figure 6 and described as follows: (1) Let the problem be defined as given in Figure 7 *a*. The first pair is selected as given in Figure 7 *b*. Next, the alignment is computed by considering pairwise alignment approach, as shown in Figure 7 *c*. In this way, alignment of each pair is estimated. (2) In this step, a random permutation between 1 and N is generated. Suppose $N = 4$, then the random permutation is generated between 1 and 4. If the obtained random permutation is (1, 2, 3, 4), then the complete alignment is generated, as shown in Figure 7 *d*. In the generalized way, k number of solutions are generated using k random permutations between 1 and N.

Molecular representation

In the MSA problem, dimension of a molecule is equal to the number of profiles (n). Let $X_i = (X_{i1}, \ldots, X_{id}, \ldots, X_{in})$ be the ith molecule. In profile representation, all the protein elements in MSA are replaced by 0 and the gap is filled by 1. Then the binary sequence in the column is converted into the decimal value which represents the $X_{i,d}$ for all $1 \le d \le n$.

Figures 8 and 9 show the molecular representation. Initially, the entire protein element in Figure 8 is replaced by 1 and gap is filled by 0 as shown in Figure 9. Then the complete binary sequence in each column is converted into the decimal value which represents the profile as shown in the last row of Figure 9; the resultant decimal sequence of all the profiles is (5, 0, 0, 0, and 10).

Fitness function

The fitness value of each molecule is determined using sum of pair approach. First, the sum of pair's symbol is calculated for each column. Then, the sum of all the

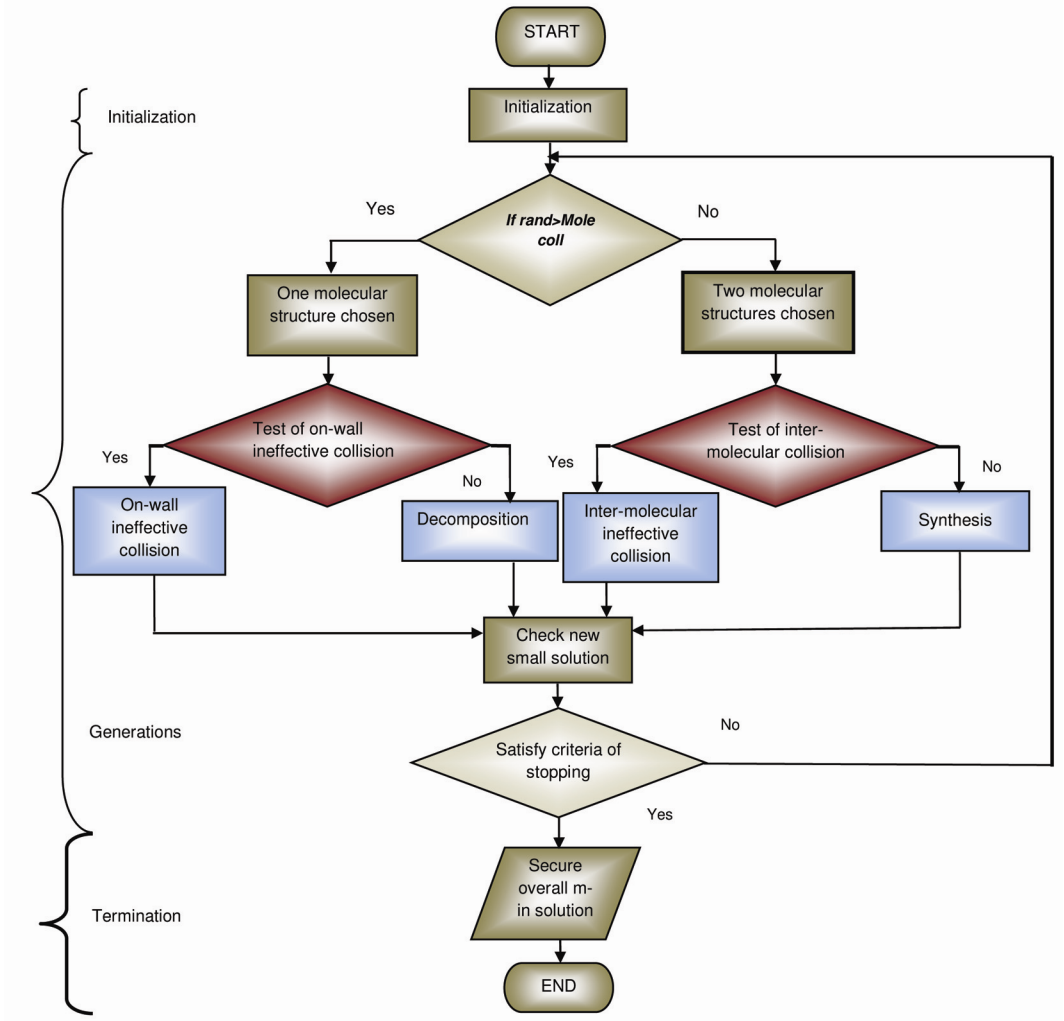

Figure 5. Flow chart of chemical reaction optimization.

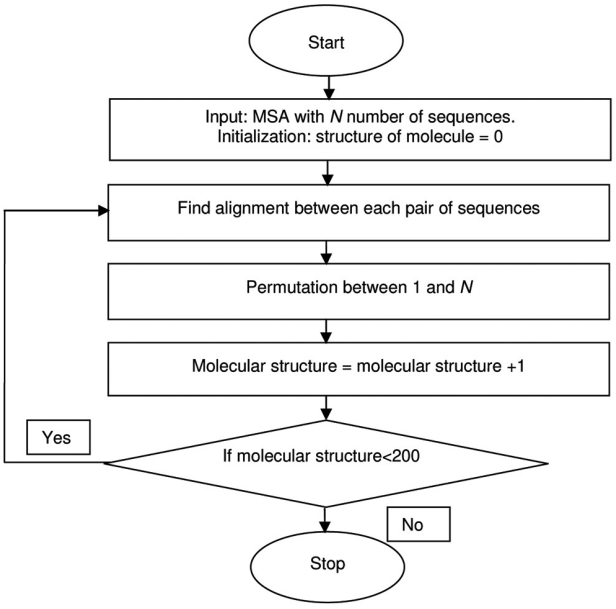

Figure 6. Flow chart of an initial generation.

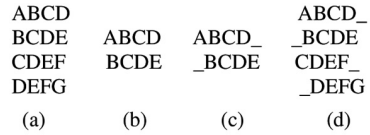

Figure 7. Complete process of an initial generation.

A	B	C	D	_
_	B	C	D	E
C	D	E	F	_
_	D	E	F	G

Figure 8. Initial solution.

	0	0	0	0	1
	1	0	0	0	0
	0	0	0	0	1
	1	0	0	0	0
Molecule →	5	0	0	0	10

Figure 9. Encoding scheme.

columns scores is considered to determine fitness function of the corresponding molecule as

$$T = \sum_{l=1}^{L} T_l, \text{ where } T_l = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{Cost}(A_i, A_j), \quad (10)$$

where T is the total cost of MSA and L is the number of columns in MSA. T_l is defined in terms of cost of the lth column and the number of sequences in complete alignment is N. Cost(A_i, A_j) is equal to the alignment score of two sequences A_i and A_j. Cost(A_i, A_j) is defined by PAM matrix. When $A_i \neq$ '_' and $A_j \neq$ '_'. Also when $A_i =$ '_' and $A_j =$ '_' then Cost(A_i, A_j) = 0. Finally, Cost(A_i, A_j) = 1, when $A_i =$ '_' and $A_j \neq$ '_' or $A_i \neq$ '_' and $A_j \neq$ '_' then cost(A_i, A_j) = 1.

Solution generation

The solution generation process is based on four types of elementary reactions such as on-wall ineffective, synthesis, inter-molecular ineffective and decomposition.

On-wall ineffective: Since this reaction is not vigorous, the resultant molecule is similar to the actual molecule. In this reaction, a random position is chosen within a molecule. Thereafter, a random number generated between 0 and 2^N replaces the corresponding position. For example, a random position is selected, say third. Next, a random number is generated, say 6. Finally, the generated number is replaced with the corresponding position, i.e. 3 replace with 6. Figure 10 shows a graphical representation of on-wall ineffective reaction.

Decomposition: This reaction is robust and the difference between resultant and actual is much more. In this reaction, two random positions are selected within the molecule. Thereafter, two right circular shift operations are performed. The first operation is performed using the first randomly selected position and the other by considering the second random position (Figure 11).

Inter-molecular ineffective: This reaction is not much effective and the difference between the resultant and actual molecule is less. In this reaction, two random molecules are considered and one random position is selected in each molecule. Then, the values are exchanged between the selected random positions as shown in Figure 12.

Synthesis: In this process, two random molecules are considered and one random position is selected within these molecules (say fourth). In the next step, the values are interchanged from the fourth position to the last position between the selected molecules. In the final step, a random molecule is considered for the next generation from the newly generated molecules. Figure 13 shows a graphical representation of the synthesis reaction, while Figure 14 shows a flow chart of ICROMSA. Table 1 lists the implementation parameters of the proposed CRO.

Dataset

For comparison, we have taken a dataset from BAliBASE version 2.0. BAliBASE 1.0 (ref. 19) contains 142 reference alignments which consist of more than 1000

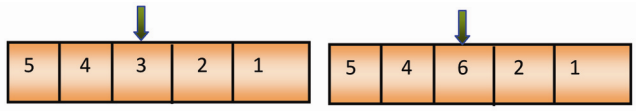

Figure 10. Graphical representation of on-wall ineffective reaction.

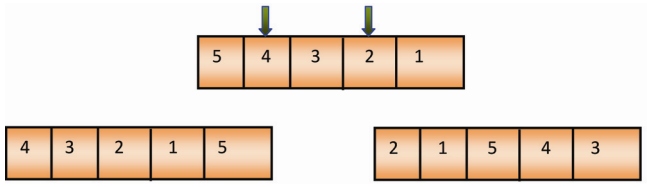

Figure 11. Graphical representation of decomposition reaction.

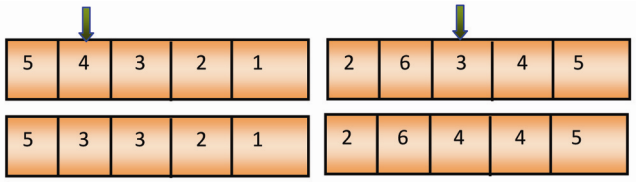

Figure 12. Graphical representation inter-molecular ineffective reaction.

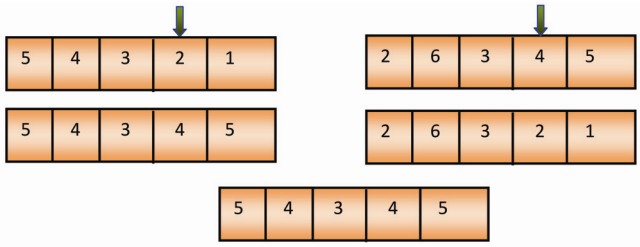

Figure 13. Graphical representation synthesis reaction.

Table 1. Implementation parameters of ICROMSA

Pop size	200
Nvars	Number of columns in the sequences
Initial kinetic energy	40
Mole coll	0.6
KE loss rate	0.8
Buffer	0
α	40
β	100

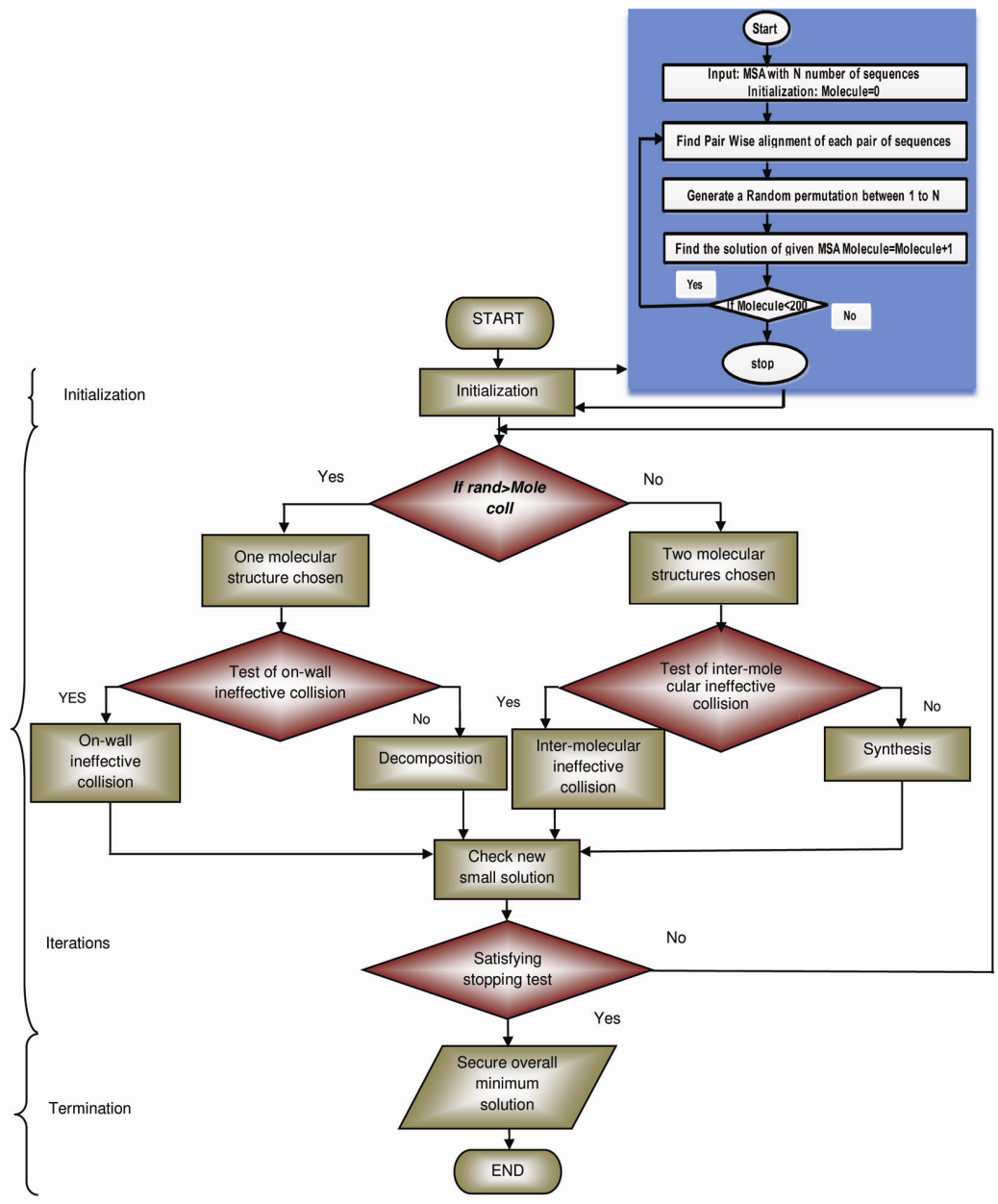

Figure 14. Flow chart of an improved chemical reaction optimization.

sequences. BAliBASE 2.0 (ref. 20) contains 167 reference alignments which consist of 2100 sequences and eight reference sets, which can be described as follows: (i) Reference set 1 contains a small number of intermediate sequences. (ii) Reference set 2 contains totally different or distinct sequences. (iii) Reference set 3 contains a set of divergent sub-families. (iv) Reference set 4 contains extended terminal extension sequences. (v) Reference set 5 contains large interior insertions or deletions. (vi) Reference sets 6–8, contain datasets in which the sequences are repeated. Bali score is used to determine the quality of algorithm. This score defines the level of similarity between manual alignment and resultant alignment. Bali score lies between 0 and 1. If the manual alignment and resultant alignment are the same then the value of Bali score is 1. If both the files are completely dissimilar, then the result is 0. A value between 0 and 1 shows the percentage of similarity match between the manual alignment file and output file obtained from the proposed or existing approach.

Experimental study

In this study, simulation is performed using C programming (Linux platform) and graphs are plotted using MATLAB (version 2013). In the performance analysis,

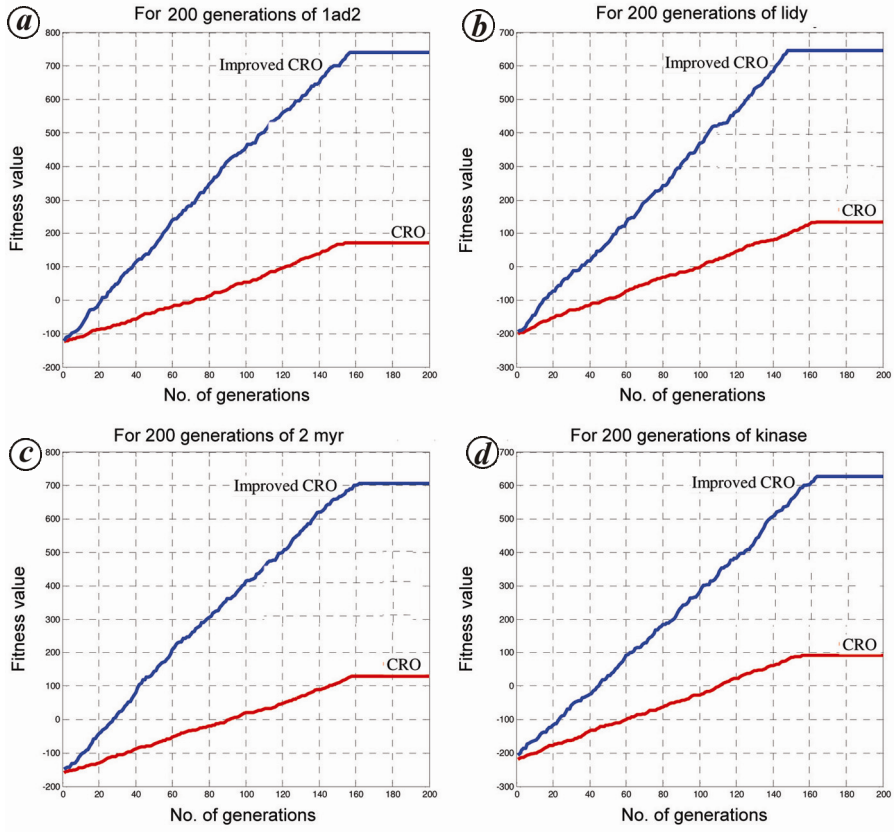

Figure 15 ***a–d.*** Performance of CRO and improved CRO per generation w.r.t. reference set 1.

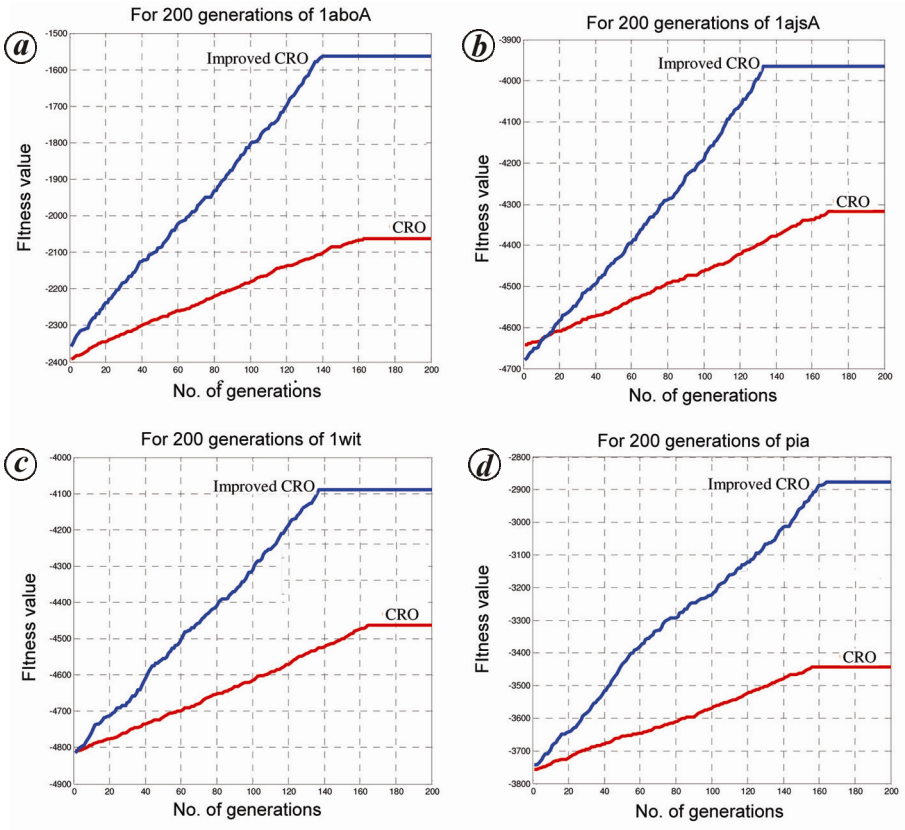

Figure 16 ***a–d.*** Performance of CRO and improved CRO per generation w.r.t. reference set 2.

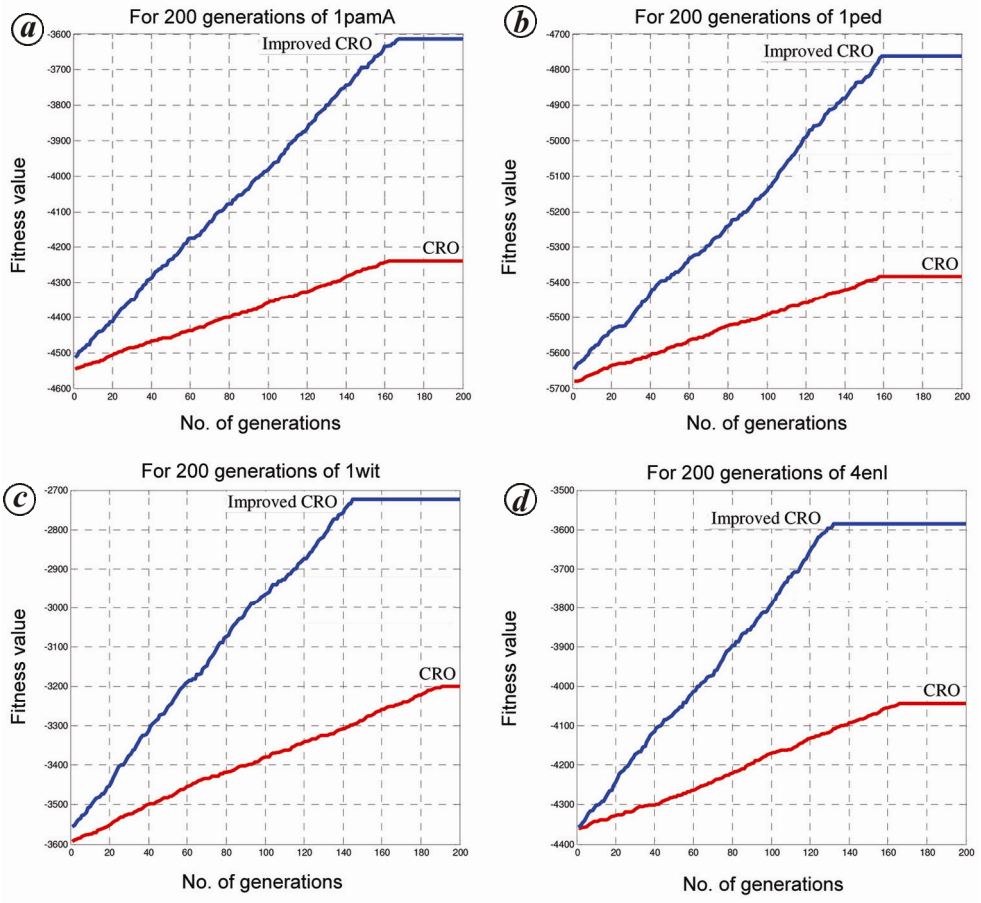

Figure 17 ***a–d.*** Performance of CRO and improved CRO per generation w.r.t. reference set 3.

first, the obtained fitness score between the improved CRO and classical CRO is compared. Then, Bali score of improved CRO algorithm is compared with the well-known existing algorithms.

Effect of improved operator in CRO

The CRO algorithm is used in optimization problems for interaction of molecules in a chemical reaction to reach a low-energy stable state. In traditional CRO, initial population is generated randomly, while the improved CRO works with an improved initial population. We have used pairwise alignment algorithm for finding the initial population. To analyse the effect of this proposed initial operator on the algorithms, two types of experiments have been conducted. Both CRO and improved CRO were run. We measured the fitness of each molecule according to fitness function. Figures 15–17 show experimental results with respect to reference sets 1–3.

Comparing the proposed method with GAPAM

To verify the efficiency of the proposed algorithm, we have taken all dataset of GAPAM[17]. In GAPAM, the authors keep best Bali score after 20 independent runs. But in this study we have taken average of 10 independent Bali score. We have taken a total of 52 test cases – 18 from reference set 1, 23 from reference set 2, and 11 from reference set 3. These are all taken from Bali base version 2.0. We have taken the approximate results of other methods reported in GAPAM[17]. Tables 2–4 provide a summary of the experimental results with respect to reference sets 1–3.

Performance of improved CRO w.r.t. reference set 1: There are several types of datasets in reference set 1, which differ in length and number of sequences. To show the superiority of the proposed algorithm in terms of Bali score, we have compared it with GAPAM, SB-PIMA, PRRP, HMMT and other well-known techniques. From Table 2, we can see that the proposed algorithm performs better in 13 out of 18 test cases, and GAPAM only five test cases. We take average solution of all methods w.r.t. all datasets of reference set 1. The average solution of the proposed method is better than some of the existing methods.

Performance of improved CRO w.r.t. reference set 2: There are various types of datasets in Bali base reference

Table 2. Experiments on reference 1 datasets of Bali base 2.0

Dataset	SAGA	SB-PIMA	DIALI	RBT-GA	CLUSTAL X	PRRP	HMMT	GAPAM	ICROMSA
2trx	0.87	0.85	0.734	0.982	0.87	0.87	0.739	**0.986**	0.921
1idy	0.548	0.000	0.000	0.997	0.515	0.37	0.353	**0.989**	0.912
1havA	0.448	0.259	0.000	0.792	0.48	0.52	0.194	0.879	**0.897**
1r69	0.475	0.675	0.675	0.9	0.675	0.675	0.000	**0.965**	0.905
1tvxA	0.448	0.241	0.000	0.891	0.552	0.207	0.276	**0.92**	0.856
1tgxA	0.773	0.678	0.63	0.835	0.727	0.695	0.622	0.878	**0.902**
1ubi	0.492	0.129	0.000	0.795	0.482	0.056	0.053	0.767	**0.839**
2myr	0.825	0.727	0.84	0.675	0.904	0.582	0.443	0.822	**0.867**
1csy	0.154	0.000	0.000	0.735	0.154	0.35	0.000	0.764	**0.813**
1aboA	0.489	0.391	0.384	0.812	0.65	0.256	0.724	0.796	**0.823**
3grs	0.282	0.183	0.350	0.755	0.192	0.363	0.141	0.746	**0.793**
1uky	0.476	0.256	0.216	0.625	0.656	0.351	0.395	0.808	**0.861**
2hsdA	0.498	0.39	0.262	0.745	0.484	0.404	0.423	0.796	**0.837**
1pamA	0.623	0.393	0.576	0.66	0.761	0.711	0.53	0.86	**0.898**
1lvl	0.726	0.62	0.783	0.567	0.746	0.772	0.539	0.781	**0.802**
Kinase	0.867	0.755	0.692	0.712	0.848	0.896	0.749	0.799	**0.888**
4enl	0.739	0.096	0.122	0.812	0.375	0.668	0.213	0.896	**0.911**
1cpt	0.776	0.184	0.425	0.584	0.66	0.821	0.388	0.875	**0.902**
1sbp	0.374	0.043	0.043	0.778	0.217	0.231	0.214	0.765	**0.787**
1ajsA	0.311	0.000	0.000	0.892	0.324	0.227	0.242	0.899	**0.944**
1ped	0.835	0.651	0.773	0.78	0.834	0.881	0.696	0.912	**0.923**
2pia	0.763	0.73	0.612	0.730	0.752	0.767	0.647	0.826	**0.859**
1wit	0.694	0.469	0.724	0.825	0.557	0.76	0.641	0.851	**0.869**
Average score	0.586	0.379	0.384	0.777	0.583	0.541	0.401	0.851	**0.869**

Table 3. Experiments on reference 2 datasets of Bali base 2.0

Dataset	MSA-GA w/prealign	CLUSTAL W	MSA-GA	SAGA	GAPAM	ICROMSA
1gpb	0.948	0.947	0.868	0.982	**0.983**	0.891
1tvxA	0.209	0.042	0.295	0.278	0.316	**0.392**
1krn	0.895	0.895	0.908	0.993	0.960	**0.966**
1taq	0.826	0.826	0.525	0.931	0.945	**0.956**
1ad2	0.845	0.773	0.821	0.917	0.956	**0.974**
2myr	0.302	0.296	0.212	0.285	0.317	**0.529**
1ycc	0.653	0.643	0.650	0.837	0.845	**0.923**
1fieA	0.942	0.932	0.843	0.947	**0.963**	0.859
1uky	0.405	0.392	0.443	0.672	0.402	**0.592**
1ldg	0.922	0.880	0.895	0.989	0.963	**0.965**
1idy	0.438	0.500	0.427	0.342	0.565	**0.683**
1sesA	0.913	0.913	0.620	0.954	**0.982**	0.943
1ar5A	0.946	0.946	0.812	0.971	**0.974**	0.933
2fxb	0.985	0.985	0.941	0.951	0.970	**0.984**
1amk	0.959	0.945	0.965	0.997	**0.998**	0.949
Kinase	0.488	0.479	0.295	0.862	0.487	**0.522**
1ped	0.687	0.592	0.501	0.746	0.498	**0.693**
3cyr	0.789	0.767	0.772	0.908	0.911	**0.929**
Average score	0.730	0.708	0.651	0.809	0.726	**0.815**

set 2. To show the superiority of proposed algorithm in terms of Bali score, we have compared it with some well-known technique. As seen in Table 3, the proposed algorithm performs better in 19 out of 23 test cases, and GAPAM in only four cases. After experimental analysis, we have seen that the proposed method does not produce best solution in all cases, but in some cases it is close to the best solution. Average score of the proposed method with respect to all test cases is also better than other existing method.

Performance of improved CRO w.r.t. reference set 3: There are many datasets in reference set 3, which is more divergent. Hence residue identities of these datasets are low. Here, we have considered 11 datasets. Table 4 shows that the proposed method is better than other

Table 4. Experiments on reference 3 datasets of Bali base 2.0

Dataset	CLUSTAL X	DIALI	HMMT	RBT-GA	PRRP	SAGA	SB-PIMA	GAPAM	ICROMSA
1uky	0.130	0.139	0.037	0.35	0.139	0.269	0.083	0.468	**0.533**
2myr	0.538	0.272	0.101	0.33	0.646	0.494	0.278	**0.813**	0.747
1ubi	0.146	0.000	0.366	0.31	0.415	**0.585**	0.000	0.386	0.527
1pamA	0.678	0.683	0.169	0.525	0.683	0.579	0.546	**0.835**	0.821
1ped	0.627	0.641	0.172	0.425	0.679	0.646	0.450	0.775	**0.839**
1ajsA	0.163	0.000	0.006	0.18	0.128	0.186	0.000	0.311	**0.417**
4enl	0.547	0.050	0.050	0.68	0.736	0.672	0.393	**0.8**	0.763
kinase	0.720	0.650	0.478	0.697	0.783	0.758	0.541	**0.828**	0.819
1idy	0.273	0.000	0.227	0.546	0.000	0.364	0.000	0.601	**0.623**
1wit	0.565	0.500	0.323	**0.78**	0.742	0.484	0.645	0.758	0.741
1r69	0.524	0.524	0.000	0.374	0.905	0.524	0.000	0.709	**0.817**
Average score	0.446	0.314	0.175	0.472	0.532	0.506	0.267	0.662	**0.695**

Table 5. Statistical analysis for the proposed method and other existing methods

Method	W^+	W^-	P	Proposed method is notable (if $P < 0.025$)
GAPAM (in reference set 1)	13	5	0.03502	No
CLUSTAL-W	15	3	0.002943	Yes
SAGA (in reference set 1)	13	5	0.0000523	Yes
MSA-GA W/prealign	14	4	0.000283	Yes
MSA-GA	17	1	0.000329	Yes
GAPAM (in reference sets 2 and 3)	25	9	$1.02e^{-1}$	Yes
RBT-GA	30	4	$4.1\ e^{-9}$	Yes
SB-PIMA	34	0	$3.5\ e^{-2}$	Yes
HMMT	34	0	$2.1\ e^{-3}$	Yes
DIALI	34	0	$7.39\ e^{-8}$	Yes
SAGA(in reference sets 2 and 3)	33	1	$4.35\ e^{-6}$	Yes
CLUSTAL(X)	34	0	$1.21\ e^{-3}$	Yes
PRRP	33	1	0.00002745	Yes

W^+, Differences above zero means positive rank; W^-, Differences below zero means negative rank; P, Probability.

methods in five test cases: SAGA in one case, RBT-GA in one case, and GAPAM in four test cases. We have calculated average score with respect to all datasets of reference set 3. The average score of the proposed method is better than some of the existing methods taken from GAPAM[17].

Statistical performance of the proposed method: We can judge the performance between two different techniques using statistical method. For a comparison between two methods, we have taken Wilcox signed-rank test[21]. Table 5 shows statistical results between the proposed method and other methods, where $W = (W^+$ or $W^-)$ is the sum of the ranks which is based on the difference between two test variables. We have considered a null hypothesis. Due to property of null hypothesis, when hypothesis is rejected then there is a significant difference between two samples. We have also considered 2.5% level of significance. If the value of P is less than 0.025, then the null hypothesis is rejected. It means that we can measure the difference between the performances of the executed algorithms, otherwise the difference is not measurable. We have computed Bali score of the proposed method and compared it with existing methods. We find that the proposed method preformed better than MSA-GA, MSA-GA w/prealign and CLUSTAL W for reference set 1. There is also a significant difference for the reference sets 2 and 3. We have found that in a single case there is no notable difference with GAPAM in reference set 1. From this observation, we can conclude that the proposed method is statistically better than other existing algorithms.

Conclusion

In the present study, we have proposed an improved CRO algorithm for the MSA problem. The initialization process of CRO has been improved for maintaining the diversity of the solution. In the experimental analysis, benchmark datasets were considered from Bali base 2.0, and the corresponding Bali score was taken which represents the performance of the proposed approach. For the sake of comparison, the proposed approach is compared with several existing approaches such as PRRP, CLUSTAL X, DIALIGN, HMMT, SB-PIMA, SAGA,

RBT-GA and GAPAM. Simulation results confirm the superiority of the proposed work over others. This implies that the proposed approach can solve the MSA problem in an effective manner.

1. Gusfield, D., *Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology*, 1997.
2. Feng, D. Johnson, M. and Doolittle, R., Aligning amino acid sequences: comparison of commonly used methods. *J. Mol. Evol.*, 1985, **21**, 112–125.
3. Bonizzoni, P. and Della Vedova, G., The complexity of multiple sequence alignment with sp-score that is a metric. *Theor. Comput. Sci.*, 2001, **259**, 63–79.
4. Carrillo, H. and Lipman, D., The multiple sequence alignment problem in biology. *SIAM J. Appl. Math.*, 1988, **48**, 1073–1082.
5. Needleman, S. B. and Wunsch, C. D., A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 1970, **48**, 443–453.
6. Taylor, W. R., A flexible method to align large numbers of biological sequences. *J. Mol. Evol.*, 1988, **28**, 161–169.
7. Feng, D. F. and Doolittle, R. F., Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol.*, 1987, **25**, 351–360.
8. Thompson, J. D. Higgins, D. G. and Gibson, T. J., Clustal W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.*, 1994, **22**, 4673–4680.
9. Eddy, S. R. Mitchison, G. and Durbin, R., Multiple alignment using hidden Markov models. *Ismb*., 1995, **3**, 114–120.
10. Cai, L. Juedes, D. and Liakhovitch, E., Evolutionary computation techniques for multiple sequence alignment. In Proceedings of the Congress on Evolutionary Computation, 2000, vol. 2, pp. 829–835.
11. Chellapilla, K. and Fogel, G. B., Multiple sequence alignment using evolutionary programming. In CEC 99. Proceedings of the Congress on Evolutionary Computation, vol. 1, 1999.
12. Horng, T. T., Lin, C. M. Liu, B. J. and Kao, C. Y., Using genetic algorithms to solve multiple sequence alignments. In Proc. GECCO, 2000, pp. 883–890.
13. Ishikawa, M., Toya, T., Totoki, Y. and Konagaya, A., Parallel iterative aligner with genetic algorithm. *Genome Inform.*, 1993, **4**, 84–93.
14. Notredame, C. and Higgins, D. G., SAGA: sequence alignment by genetic algorithm. *Nucl. Acids Res.*, 1996, **24**, 1515–1524.
15. Gondro, C. and Kinghorn, B., A simple genetic algorithm for multiple sequence alignment. *Genet. Mol. Res*., 2007, **6**, 964–982.
16. Taheri, J. and Zomaya, A. Y., RBT-GA: a novel metaheuristic for solving the multiple sequence alignment problem. *BMC Genomics*., 2009, **10**, S10.
17. Naznin, F. Sarker, R. and Essam, D., Progressive alignment method using genetic algorithm for multiple sequence alignment. *IEEE Trans. Evol. Comput.*, 2012, **16**, 615–631.
18. Lam, A. Y. S. and Li, V. O. K., Chemical-reaction-inspired metaheuristic for optimization. *IEEE Trans. Evol. Comput*., 2010, **14**, 381–399.
19. Thompson, J. D., Plewniak, F. and Poch, O., Bali base: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 1999, **15**, 87–88.
20. Bahr, A., Thompson, J. D. Thierry, J. C. and Poch, O., Bali base (benchmark alignment database): enhancements for repeats, transmembrane sequences and circular permutations. *Nucleic Acids Res*., 2001, **29**, 323–326.
21. Corder, G. W. and Foreman, D. I., *Nonparametric Statistics for Non-Statisticians: A Step-by-Step Approach*, New York, Wiley, 2009.

Received 4 September 2015; revised accepted 17 August 2016

doi: 10.18520/cs/v112/i03/527-538