An Efficient Cooperative Method to Solve Multiple Sequence Alignment Problem

Lamiche Chaabane(✉)

Computer Science Department, Mohamed Boudiaf University, M'sila, Algeria
Lamiche07@gmail.com

Abstract. In this research work, we propose a cooperative approach called simulated particle swarm optimization (SPSO) which is based on metaheuristics to find an approximate solution for the multiple sequence alignment (MSA) problem. The developed approach uses the particle swarm optimization (PSO) algorithm to discover the search space globally and the simulated annealing (SA) technique to improve the population leader «gbest» quality in order to overcome local optimum problem. Simulation results on BaliBASE benchmarks have shown the potent of the proposed method to produce good quality alignments comparing to those given by other existing methods.

Keywords: Cooperative approach · Multiple sequence alignment
SPSO · PSO · SA · BaliBASE benchmarks

1 Introduction

Multiple sequence alignment (MSA) is a crucial tool in molecular biology and genome analysis. It has been considered as one of the important tasks in bioinformatics [1]. It helps to construct a phylogenetic tree of related DNA sequences, to predict the function and structure of unknown protein sequences, by aligning with other sequences whose function and structure is already known, and to allow comparison of the structural relationships between sequences by simultaneously aligning multiple sequences and constructing connections between the elements in different sequences [2].

Discovering optimal alignment in multiple biological sequence data is known as a NP-complete problem [3]. It has been identified as a combinatorial optimization problem [4], which is solved by using an exact or approximate algorithms. These algorithms lead to exploit various genetic information to determine evolutionary relationships among living beings [3].

Recently, the trend has shifted to the use of iterative algorithms in order to tackle the MSA problem. These approaches are based on the improvement of the given initial alignment through a series of some iterations until a stopping criterion is reached. They include genetic algorithm (GA) [5], simulated annealing algorithm (SA) [6], particle swarm optimization (PSO) [7], GA-ACO algorithm [8], Ant Colony Algorithm [9] and so on. Generally, these metaheuristics are able to find nearly optimal solutions for large instances in a reasonable processing time.

In this study, we propose a hybrid approach called SPSO algorithm to solve the MSA problem. The developed model is the cooperation between both PSO algorithm

and simulated annealing technique. The remainder of the paper is organized as follows: Sect. 2 presents a brief review of the researches related to the proposed framework. In Sect. 3, both PSO and SA concepts are described. In Sect. 4, our proposed SPSO algorithm is explained in detail. In Sect. 5, the simulation results are provided. Finally, the study is concluded in Sect. 6.

2   Background

A brief review of some related works in the multiple sequence alignment field using iterative methods is presented in this section. Riaz et al. [10] presented a tabu search algorithm to align multiple sequences. The framework of his work consists to implement the adaptive memory features typical of tabu searches in order to obtain multiple sequences alignment where the quality of an alignment is measured by the COFFEE objective function. In [11], the authors proposed a novel approach to multiple sequence alignment based on Particle Swarm Optimization (PSO) to improve a sequence alignment previously obtained using Clustal X.

In Ref. [12], the authors presented an approach to the MSA problem by applying genetic algorithm with a reserve selection mechanism to avoid premature convergence in GA. A better results are obtained compared with those produced by classical GA. The authors in [13] proposed an algorithm based on binary PSO algorithm to address the multiple sequence alignment problem. Simulation results using SP score measure and nine BaliBASE tests case showed that the proposed BPSO algorithm has superior performance when compared to ClustalW and SAGA algorithms.

An artificial bee colony algorithm for solving MSA problem is introduced in [14]. In Ref. [15], Cutello et al. presented an immune inspired algorithm (IMSA) to tackle the multiple sequence alignment problem using ad-hoc mutation operators. Experimental results on BALIBASE v.1.0 show that IMSA is superior to PRRP, CLUSTALX, SAGA, DIALIGN, PIMA, MULTIALIGN and PILEUP8. In [16], simulated annealing technique was applied to solve MSA problem using a set of DNA benchmarks of HIV virus genes of human and simian.

In Ref. [17], the authors proposed a hybrid algorithm using a GA and cuckoo search algorithm to improve multiple sequence alignment. The obtained results are compared with ClustalW by using five different datasets. Recently, an efficient method by using multi-objective genetic algorithm (MSAGMOGA) to discover optimal alignments is proposed in [18]. Experiments on the BAliBASE 2.0 database confirmed that MSAGMOGA obtained better results than MUSCLE, SAGA and MSA-GA methods.

3   Preliminaries

3.1   Outline of the Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an adapted algorithm developed the first time by Kennedy and Eberhart [19], inspired by bird flocking and fish schooling. PSO used a

population of individuals called particles and two primary operators: velocity update and position update. During each generation, each particle moves toward the particles according to its best position and the global best position. In addition, a new velocity value for each particle is calculated based on its current velocity, the distance from its previous best position and the distance from the global best position. The evolution of the swarm is governed by the following equations:

$$V^{(k+1)} = w.V^{(k)} + c_1.rand_1.\left(pbest^{(k)} - X^{(k)}\right) + c_2.rand_2.\left(gbest^{(k)} - X^{(k)}\right). \quad (1)$$

$$X^{(k+1)} = X^{(k)} + V^{(k+1)}. \quad (2)$$

where:
X is the position of the particle,
V is the velocity of the particle,
w is the inertia weight,
$pbest$ is the best position of the particle,
$gbest$ is the global best position of the swarm,
$rand1$, $rand2$ are random values between 0 and 1,
$c1$, $c2$ are positive constants which determine the impact of the personal best solution and the global best solution on the search process, respectively, k is the iteration number.

Concerning the stopping condition, generally PSO algorithm terminates when a set number of times or until a minimum error is achieved. All parameters of PSO algorithm are fixed experimentally in order to have a good compromise between the convergence time of the algorithm and the final solution quality.

3.2 Outline of the Simulated Annealing (SA)

Simulated annealing (SA) is a general probabilistic local search algorithm proposed by Kirkpatrick et al. [20] to solve difficult optimization problems. It is inspired by the annealing of solids in physics. SA models the slow cooling process of solids to achieve the minimum energy as an analogy, reaching the minimum function value. As a result, it attains an optimal/near-optimal solution by implementing an iterative cooling process from a high temperature, at which solid particles are in the liquid phase. Simulated annealing utilizes a control parameter, temperature T, for the cooling process. The solid is allowed to attain the thermal equilibrium for every T degree that has its energy E probabilistically distributed, as given in Eq. (3), where k_b is the Boltzmann constant.

$$P(E) = e^{\left(\frac{-E}{k_b T}\right)}. \quad (3)$$

In the combinatorial optimization context, if we aim to find a good solution then we move from a solution to one of its neighbors in the search space according to a probabilistic criterion. If the cost decreases then the solution is retained and the move is accepted. Otherwise, the move is accepted only with a probability depending on the cost increase and the temperature parameter T [20].

4   Proposed Method

PSO performs excellently in the case of global search but it is not efficient in local search. It suffers from its weak local search ability and the local minima limit. On the other hand, SA is good in local search while less good in global search. However, it takes advantage of the acceptance of candidate solutions by the use of metropolis criteria and must escape from local optimum to search in other solution space.

In order to construct an intelligent algorithm which can be effectively avoid weaknesses and fully use the advantages of both PSO and SA algorithms, we propose a hybrid approach which combines the PSO with simulated annealing, so the new hybrid algorithm called Simulated Particle Swarm optimization (SPSO) conducts both global search and local search in every iteration. According to this hybridization manner, the probability to obtain better solutions significantly increases. At each iteration, the proposed hybrid SPSO algorithm consists of applying PSO algorithm in order to guide global search, and use SA to improve the *gbest* which helps PSO to escape from local optimum and increase the convergence speed of SPSO algorithm. The flowchart of the proposed SPSO is presented in Fig. 1.

4.1   PSO Components of MSA Problem

Particle Representation. Each particle represents a potential solution to the MSA problem, effectively it corresponds to a sequence alignment. A particle is then represented as a set of vectors, where each vector specifies the positions of the gaps in each one of the sequences to be aligned [21].

Swarm Initialization. The size of the whole swarm is determined by the user. The initial set of particles is generated by adding gaps into each sequence at random position, thus, all the sequences have the same length L in which its value is 1.2 times of the longest sequences [21].

Fitness Evaluation. A parameter to determine which alignment will survive in the next generation is its fitness value. A formal definition of the sum-of-pairs (SPS) of multiple sequence alignment is introduced which is used as a tool to compute fitness. The score assigned to each alignment is the sum of the scores (SP) of the alignment of each pair of sequences. The score of each pair of sequences is the sum of the score assigned to the match of each pair of symbols, which is given by the substitution matrix. The score of a multiple alignment is given as follows:

$$Score(A) = \sum_{i=1}^{k-1} \sum_{j=i+1}^{k} S(A_i, A_j). \tag{4}$$

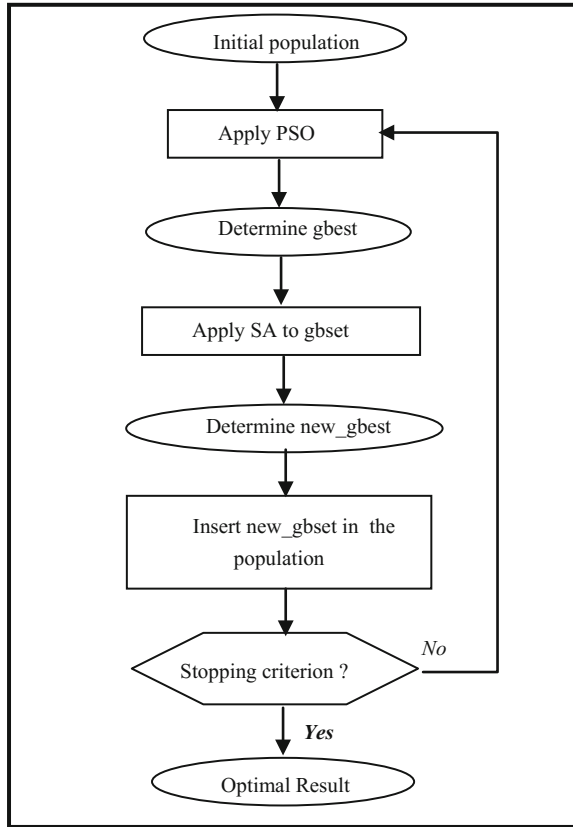where the $S(A_i, A_j)$ is the alignment score between two given sequences A_i and A_j.

Fig. 1. SPSO algorithm flowchart.

Particle Move. In the PSO algorithm, each particle moves towards the leader at a speed proportional to the distance between the particle and the leader. In this paper, this distance will be measured as the proportion of gaps that do not match in the sequences, according to the formula:

$$Distance = \frac{no\,matching\,gaps}{total\,gaps}. \tag{5}$$

To move particles towards the leader, an operator similar to the crossover operator from genetic algorithms is used [21]. It consists to select a crossover point which divides the alignment into two segments, and then a segment of the particle is replaced with a segment from the leader. This replacement is achieved removing from the particle the gaps that are in the segment, and then adding the gaps from the leader's segment.

4.2   SA Components of MSA Problem

According to the basic elements of simulated annealing cited above, components of our proposed algorithm for MSA problem are as follows:

Cost Function. The cost function is used to evaluate the quality of each alignment in the swarm. In order to retain the same paradigm to measure the quality of the solution, we decide in this study to use the sum-of-pairs (SPS) as a cost function which is the same one of that used in the PSO algorithm.

Initial Solution. The generation of an initial solution is an important step towards getting a final improved alignment. In our developed method, the initial solution is constructed by insertion of gaps randomly in each sequence of the alignment.

Generation of Neighbors. In the multiple sequence alignment context, neighbors of a current solution are obtained by employing some perturbations to the gaps positions into the different sequences. In this work, we choose to apply a simple efficient strategy to change positions of these gaps, this mechanism is called LocalShuffle operator, its main idea is as follows: firstly, picks a random amino acid from a randomly chosen sequence in the alignment and checks whether one of its neighbors is a gap. If this is the case, the algorithm swaps the selected amino acid with a gap neighbor. If both neighbors are gaps then one of them is picked randomly [22].

Choice of Cooling Schedule. An effective cooling schedule is essential to reducing the amount of time required by the algorithm to find an optimal solution. Several temperature decreasing schemes have been proposed in the literature, they include static schedules and adaptive schedules [16]. Here, we propose to use the most common cooling function which is defined by $T_{k+1} = \alpha.T_k$. This function decreases the temperature value by a α factor, where $\alpha \in [0.70, 1.0[$. The pseudo-code of our SA is given below.

Pseudo-code of SA algorithm

```
Fix Tinitial, Tfinal, α
Set T ←Tinitial, Scurrent ←Sinitial, score(Scurrent) ← score(Sinitial)
While (T > Tfinal) do
    Snew ←LocalShuffle(Scurrent)
    Compute the alculate score of Snew
   d:=score(Snew)- score(Scurrent)
   If (d >= 0) then
        Scurrent ← Snew
        If score (Snew) > score(Sbetter) then
            Sbetter ← Snew
       end if
   else
   If exp(d/T ) > random(0,1) then
        Scurrent ← Snew
    end if
   end if
    T= α.T
end while
```

After the description of the key components of both PSO and SA algorithms, the pseudo-code of our SPSO procedure is summarized as follows:

SPSO Pseudo-code

---

1. Generate randomly a set of particles

2. Fix ITMAX value
3. Set it ← 1
 4. While (it ≤ ITMAX) do
    4.1. Determine the leader particle *gbest*
    4.2. For each particle in population do
        a) Measure the distance between *gbest* and the particle
        b) Select randomly a crossover point
        c) Apply particle move mechanism
   End for
    // Calculate the new leader particle gbest
   4.3 Update *gbest*
   4.4 *New_gbest* ← SA(*gbest*)   *// Apply SA to improve gbest*
   4.5 Insert *New_gbest* into the current population
   4.6. it ← it + 1
5. End while
// Output the result.
6. Return *gbset* value and its correspondent particle.

---

5 Simulation and Results

The proposed approach is implemented in Java language. All tests have been fulfilled on a PC with an 2.66 GHz Intel Pentium IV processor and 4 GB RAM. We conducted some experiments in order to demonstrate the effectiveness of our SPSO algorithm. For that, a set of benchmark sequences with different identities and lengths are chosen from the BAliBase 1.0 library [23]. Characteristics of the used data and all parameters setting of the algorithm SPSO are summarized in Tables 1 and 2 respectively.

Table 1. Characteristics of benchmark sequences.

Types	Name	N	(Min, Max)	Identity
Short	SH3	5	(80, 49)	<25%
	Cytochrome c	5	(70, 87)	20%–40%
	Serine protease	5	(66, 82)	>35%
Medium	Protein kinase	5	(263, 276)	<25%
	Anthranilate isomerase	4	(247, 259)	20%–40%
	Serine protease	5	(222, 245)	>35%
	Aminotransferase	4	(358, 387)	<25%
Long	Glutamyl-trna synthetase	5	(438, 486)	20%–40%
	Taq DNA polymerase	5	(806, 928)	>35%

Table 2. Parameter settings for experiments.

Parameter	Value
Number of iterations (*ITMAX*)	1000
Swarm size	50
Initial temperature ($T_{initial}$), final temperature (T_{final})	100, 0.0001
Score matrix, gap open	PAM250, 10
c_1, c_2 constants	1.49618

All obtained results by GA, PSO, ABC and our SPSO algorithm using the best, the worst and the average SPS values of 10 experiments are summarized in Tables 3, 4 and 5 respectively.

The results plotted in Tables 3, 4 and 5 show clearly the considerable improvement of scores by the proposed SPSO approach. Indeed, it produces alignments of much better quality than that of the other cited methods in both specified categories.

Table 3. Comparison results for short sequences.

Sequence name	Algorithm	Average	Best	Worst
SH3	GA [14]	0.60.90	0.6829	0.5463
	PSO [14]	0.6444	0.6543	0.6296
	ABC [14]	0.6895	0.7654	0.6543
	Our SPSO	**0.7698**	**0.8712**	**0.8197**
Cytochrome c	GA [14]	0.5141	0.5398	0.4891
	PSO [14]	0.6252	0.6407	0.6116
	ABC [14]	0.6451	0.6854	0.5922
	Our SPSO	**0.7206**	**0.7417**	**0.6806**
Serine protease	GA [14]	0.7455	0.8065	0.6921
	PSO [14]	0.7120	0.7331	0.7008
	ABC [14]	0.7956	0.8168	0.7732
	Our SPSO	**0.8878**	**0.8904**	**0.8505**

Table 4. Comparison results for medium sequences.

Sequence name	Algorithm	Average	Best	Worst
Protein kinase	GA [14]	0.3508	0.3620	0.3453
	PSO [14]	0.4848	0.5065	0.4765
	ABC [14]	0.4951	0.5272	0.4765
	Our SPSO	**0.5277**	**0.5801**	**0.5312**
Anthranilate isomerase	GA [14]	0.3317	0.3440	0.3244
	PSO [14]	0.3813	0.4117	0.3529
	ABC [14]	0.3825	0.4292	0.3682
	Our SPSO	**0.4144**	**0.4549**	**0.4218**

(*continued*)

Table 4. (*continued*)

Sequence name	Algorithm	Average	Best	Worst
Serine protease	GA [14]	0.3792	0.4334	0.3579
	PSO [14]	0.2626	0.2725	0.2507
	ABC [14]	0.4178	0.4480	0.4036
	Our SPSO	**0.4646**	**0.4703**	**0.4209**

Table 5. Comparison results for long sequences

Sequence name	Algorithm	Average	Best	Worst
Aminotransferase	GA [14]	0.3646	0.3719	0.3548
	PSO [14]	0.5819	0.5903	0.5645
	ABC [14]	0.5819	0.5936	0.5741
	Our SPSO	**0.6602**	**0.6378**	**0.6612**
Glutamyl-trna synthetase	GA [14]	0.3738	0.3824	0.3648
	PSO [14]	0.3987	0.4154	0.3828
	ABC [14]	0.4040	0.4254	0.3857
	Our SPSO	**0.4316**	**0.4449**	**0.4207**
Taq DNA polymerase	GA [14]	0.2500	0.2644	0.2441
	PSO [14]	0.2533	0.2588	0.2484
	ABC [14]	0.2569	0.2615	0.2520
	Our SPSO	**0.2603**	**0.2882**	**0.2794**

In order to further assess the potent of our SPSO algorithm, a second experiment using another datasets is performed, where the goal is to compare our SPSO approach with TLPSO-MSA [24] technique. In this experiment, three different sets of protein coming from the BaliBASE 3.0 database [25] are selected. The average SP and TC scores [26] are portrayed in Table 6.

Table 6. Comparison results on the selected test cases.

Sequence set	N	TLPSO-MSA [24]		Our SPSO	
		SP	TC	SP	TC
RV11	38	0.80	0.71	**0.84**	**0.74**
RV12	44	**0.92**	0.80	0.86	**0.81**
RV20	41	0.91	0.64	**0.92**	**0.68**
Overall average		0.87	0.72	**0.87**	**0.74**

From Table 6, it can be seen that our SPSO outperforms clearly TLPSO-MSA method in terms of TC score for all cases. In terms of SP score, it finds results better than TSPSO-MSA for RV11 and RV20 protein families and produces a competitive results on RV12 dataset.

6   Conclusion

In this work, we contributed to the ongoing research by proposing a hybrid model for finding optimized alignments to the MSA problem. The developed approach uses the characteristics of the random search, global convergence of the PSO and the potent of the simulated annealing to update the global optimal solution. The performance of the proposed SPSO algorithm is judged on a set of BaliBase benchmark problems and is favorably compared with other algorithms in the literature. The results demonstrate that the proposed approach is overall more effective to find better alignments in a reasonable processing time.

In the future, we will revise our score function to make this score more realistic, also, we can use another intelligent heuristic to generate the initial whole swarm in order to increase the convergence speed of the algorithm. In addition, we can incorporate other efficient mechanisms for the neighborhood structure to improve the global solution quality. A comparison of the proposed method with some other aligners such as Clustal W, SAGA or MULTALIGN is possible to verify its effectiveness.

References

1. Thompson, J.D., Thierry, J.C., Poch, O.: RASCAL: rapid scanning and correction of multiple sequence alignments. Bioinformatics **19**(9), 1155–1161 (2003)
2. Jiang, T., Wang, L.: On the complexity of multiple sequence alignment. J. Comput. Biol. **1**, 337–378 (1994)
3. Bonizzoni, P., Della Vedova, G.: The complexity of multiple sequence alignment with SP-score that is a metric. Theor. Comput. Sci. **259**, 63–79 (2001)
4. Papadimitriou, C.H., Steiglitz, K.: Combinatorial Optimization: Algorithms and Complexity. Dover Publications, New York (1998)
5. Horng, J.T., Wu, L.C., Lin, C.M., Yang, B.H.: A Genetic algorithm for multiple sequence alignment. Soft Comput. **9**, 407–420 (2005)
6. Hernández-Guía, M., Mulet, R., Rodríguez-Pérez, S.: A new simulated annealing algorithm for the multiple sequence alignment problem. The approach of polymers in a random media. Phys. Rev. E **72**, 1–7 (2005)
7. Lei, C.W., Ruan, J.H.: A particle swarm optimization algorithm for finding DNA sequence motifs. In: Proceedings IEEE, pp. 166–173 (2008)
8. Lee, Z.J., Su, S.F., Chuang, C.C., Liu, K.H.: Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. Appl. Soft Comput. **8**, 55–78 (2008)
9. Chen, L., Zou, L., Chen, J.: An efficient ant colony algorithm for multiple sequences alignment. In: Proceedings of the 3rd International Conference on Natural Computation (ICNC 2007), pp. 208–212 (2007)
10. Riaz, T., Wang, Y., Li, K.B.: Multiple sequence alignment using tabu search. In: Proceedings of 2nd Asia-Pacific Bioinformatics Conference (APBC), Dunedin, New Zealand, pp. 223–232 (2004)
11. Xu, F., Chen, Y.: A method for multiple sequence alignment based on particle swarm optimization. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. LNCS (LNAI), vol. 5755, pp. 965–973. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-04020-7_104

12. Chen, Y., Hu, J., Hirasawa, K., Yu, S.: Multiple sequence alignment based on genetic algorithm with reserve selection. In: Proceedings of International Conference on Networking, Sensing and Control (ICNSC), pp. 1511–1516 (2008)
13. Long, H.X., Xu, W.B., Sun, J., Ji, W.J.: Multiple sequence alignment based on a binary particle swarm optimization algorithm. In: Proceedings of Fifth International Conference on Natural Computation, pp. 265–269 (2009)
14. Lei, X., Sun, J., Xu, X., Guo, L.: Artificial bee colony algorithm for solving multiple sequence alignment. In: Proceedings of 2010 IEEE Fifth International Conference on BIC-TA, pp. 337–342 (2010)
15. Cutello, V., Nicosia, G., Pavone, M., Prizzi, I.: Protein multiple sequence alignment by hybrid bio-inspired algorithms. Nucleic Acids Res. **39**(6), 1980–1992 (2011)
16. Liñán-García, E., Gallegos-Araiza, L.M.: Simulated annealing with previous solutions applied to DNA sequence alignment. ISRN Artif. Intell. **2012**, 1–6 (2012)
17. Abu-Srhan, A., Al Daoud, E.: A hybrid algorithm using a genetic algorithm and cuckoo search algorithm to solve the traveling salesman problem and its application to multiple sequence alignment. Int. J. Adv. Sci. Technol. **61**, 29–38 (2013)
18. Kayaa, M., Sarhanb, A., Alhajjb, R.: Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. Comput. Meth. Programs Biomed. **114**, 38–49 (2014)
19. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proceedings of IEEE International Conference on Neural Networks, Perth, vol. 4, pp. 1942–1948, (1995)
20. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
21. Rodriguez, P.F., Nino, L.F., Alonso, O.M.: Multiple sequence alignment using swarm intelligence. Int. J. Comput. Intell. Res. **3**(2), 123–130 (2007)
22. Lamiche, C.: An effective alignment technique to solve MSA problem. Asian J. Math. Comput. **17**(2), 2395–4213 (2017)
23. Thompson, J.D., Plewniak, F., Poch, O.: BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. Bioinformatics **15**, 87–88 (1999)
24. Lalwani, S., Kumar, R., Gupta, N.: A novel two-level particle swarm optimization approach for efficient multiple sequence alignment. Memet. Comput. **7**(2), 119–133 (2015)
25. Thompson, J.D., Koehl, P., Ripp, R., Poch, O.: BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins **61**(1), 127–136 (2005)
26. Thompson, J.D., Plewniak, F., Poch, O.: A comprehensive comparison of multiple sequence alignment programs. Nucleic Acids Res. **27**(13), 2682–2690 (1999)