Multiple Guide Trees in a Tabu Search Algorithm for the Multiple Sequence Alignment Problem

Tahar Mehenni(✉)

Computer Science Department,
University Mohamed Boudiaf of M'sila, 28000 M'sila, Algeria
tmehenni@univ-msila.dz

Abstract. Nowadays, Multiple Sequence Alignment (MSA) approaches do not always provide consistent solutions. In fact, alignments become increasingly difficult when treating low similarity sequences. Tabu Search is a very useful meta-heuristic approach in solving optimization problems. For the alignment of multiple sequences, which is a NP-hard problem, we apply a tabu search algorithm improved by several neighborhood generation techniques using guide trees. The algorithm is tested with the BAliBASE benchmarking database, and experiments showed encouraging results compared to the algorithms studied in this paper.

Keywords: Multiple sequence alignment · Tabu search · Neighborhood · Guide tree

1 Introduction

Multiple sequence alignment (MSA) is a very interesting problem in molecular biology and bioinformatics. Although the most important regions of DNA are usually conserved to ensure survival, slight changes or mutations (indels) do occur as sequences evolve. Methods such as sequence alignment are used to detect and quantify similarities between different DNA and protein sequences that may have evolved from a common ancestor.

Sequence alignment is the way of inserting dashes into sequences in order to minimize (or maximize) a specified scoring function [1,26]. There are two classes of sequencing; pairwise sequence alignment (PwSA) and multiple sequence alignment (MSA). The latter is simply an extension of pairwise alignments that align 3 or more sequences. Both MSA and PwSA can further be categorized as global or local methods. As global methods attempt to align entire sequences, local methods only align certain regions of similarity.

The majority of multiple sequence alignment heuristics is now handled using progressive approach [13]. Progressive also known as hierarchical or tree methods, generate a multiple sequence alignment by first aligning the most similar sequences and then adding successively less related sequences or groups to the alignment until the entire query set has been incorporated into the solution. Sequence relatedness is described by the initial tree that is based on Pair

A. Amine et al. (Eds.): CIIA 2015, IFIP AICT 456, pp. 141–152, 2015.
DOI: 10.1007/978-3-319-19578-0_12

wise alignments which may include heuristic Pair wise alignment methods. Some well-known programs using progressive strategies are ClustalW [28], Muscle [6], MULTAL [12] and T-COFFEE [20]. This approach has the advantages of speed and simplicity. However, its main disadvantage is the local minimum problem, which comes from the greedy nature of the approach.

Another approach is to prune the search space of the Dynamic Programming (DP) algorithm for simultaneously aligning multiple sequences, e.g., MSA [11, 18], OMA [23] etc. Algorithms of this approach often find better quality solutions than those of the progressive approach. However, they have the drawbacks of complexity, running time and memory requirement, so they can only be applied to problems with a limited number of sequences (about 10).

The iteration-based approach is also applied to the multiple sequence alignment. Iterative alignment methods produce alignment and refine it through a series of cycles (iterations) until no further improvements can be made. It is deterministic or stochastic depending on the strategy used to improve the alignment. This approach includes iterative refinement algorithms, e.g,. PRRP [10], simulated annealing [14], genetic algorithms (SAGA [19], MAGA [29]), Ant Colony [3] and Swarm Intelligence [15]. Therefore, they can evade being trapped in local minima.

In this paper, we present an iteration-based approach using tabu search features to find the global alignment of multiple sequences, where the neighbors are generated using a set of operations on the guide tree of the initial solution.

The remaining of the paper is organized as follows. In section 2, we present the related work in MSA using tabu search. Section 3, describes our algorithm. Experimental results are presented in section 4 and the study is concluded in section 5.

2 Related Work

Tabu Search (TS) [8, 9] was developed by Fred Glover in 1988. It was initiated as an alternative local search algorithm addressing combinatorial optimization problems in many fields like scheduling, computer channel balancing, cluster analysis, space planning etc. Tabu search is an iterative heuristic approach that uses adaptive memory features to align multiple sequences. The adaptive memory feature, a tabu list, helps the search process to avoid local optimal solutions and explores the solution space in an efficient manner.

In [24], authors propose a tabu search algorithm for multiple sequence alignment. The algorithm implements the adaptive memory features typical of tabu searches to align multiple sequences. Both aligned and unaligned initial solutions are used as starting points for this algorithm. Aligned initial solutions are generated using Feng and Doolittles progressive alignment algorithm [7]. Unaligned initial solutions are formed by inserting a fixed number of gaps into sequences at regular intervals. The quality of an alignment is measured by the COFFEE objective function [21]. In order to move from one solution to another, the algorithm moves gaps around within a single sequence and performs block moves.

This tabu search uses a recency-based memory structure. Thus, after gaps are moved, the tabu list is updated to avoid cycling and getting trapped in a local solution.

[17] develops in his thesis several tabu searches that progressively align sequences. He begins by a simple tabu, called Tabu A, using Dynamic Programming (DP). Then, he proposes other modified versions of tabu search, using at each time a new feature for the previous algorithm, like subgroups alignment, intensification and diversification.

In this paper, we develop a novel tabu search algorithm, by adapting similar procedures of Tabu search developed by [17], and adding a new and efficient technique for generating neighbors using guide trees.

3 Algorithm Overview

We first give a general description of the tabu search components of our method (initial solution, neighborhood generation and intensification method), and then provide a summarizing pseudo-code description of the main algorithm.

Tabu search works by starting from an initial solution, and iteratively explores the neighborhood of current solution by generating the moves called neighbors. In each iteration, the neighbors are evaluated through the alignment score and the best neighbor, provided it is not in the tabu list, is selected and applied to the current solution. This produces a new current solution for the next iteration. The applied neighbor is added to the tabu list and it is not allowed for a specified number of iteration called tabu tenure.

3.1 Initial Solution

The generation of an initial solution is an important step towards getting a final improved alignment. A good initial solution can effectively converge faster and hence cut the computational cost. The initial solution of the tabu search is represented by a tree that is generated using the neighbor-joining guide tree (NJ) [25], which fixes the order of the partial alignments in the progressive alignment.

The NJ method constructs guide trees by clustering the nearby sequences in a stepwise manner. In each step of the sequence clustering, it minimizes the sum of branch lengths, selecting the two nearest sequences/nodes and joining them. Next, the distance between the new node and the remaining ones is recalculated. This process is repeated until all sequences are joined to the root of the guide tree. Figure 1 gives an example of a guide tree produced by 5 sequences.

The MSA is obtained from the tree as follows: the pair of sequences on the lowest level are aligned first. Then, the entire branch containing these two sequences is aligned starting from the lowest level and progressing upward to sequences on higher levels. After the MSA is determined, the alignment is scored.

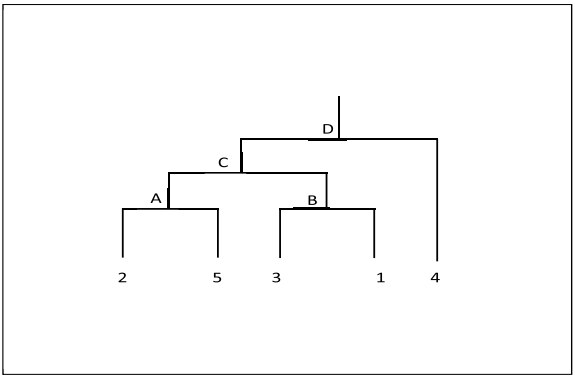

Fig. 1. An example of a guide tree generated by NJ Clustering Algorithm as Initial solution for the Tabu Search

The most popular scoring scheme is the sum of all pairwise alignments score: Sum-of-Pairs Score (SP).

$$SP = \sum_{i=1}^{n-1} \sum_{j=i}^{n} Score(S_i, S_j) \tag{1}$$

where

$$Score(S_i, S_j) = \max \begin{cases} (S_{i-1}, S_{j-1}) + s(x_i, y_j) \\ (S_{i-1}, S_j) - d \\ (S_{i-1}, S_j) + d \end{cases}$$

where $s(x_i, y_j)$ is the score for matching symbols x_i and y_j and d is the penalty for introducing a gap.

3.2 Neighborhood Generation

The neighborhood of the current solution may be generated by one of the four ways: swapping, node insertion, branch insertion or distance variation.

Generation by Swapping. The simplest way of generating a neighborhood is swapping the order of the sequences (i.e. leaves) while maintaining the same guide tree topology. the number of guide trees generated by swapping is $n(n-1)/2$, where n is the number of sequences to be aligned. Figure 2 shows two guide trees (b and c) generated from the initial guide tree a by swapping the order of the sequences.

Generation by Node Insertion. Neighbors can be generated from the current solution (i.e. the current guide tree) by performing certain insertions of nodes. The node insertion makes it possible to move a sequence node to another location

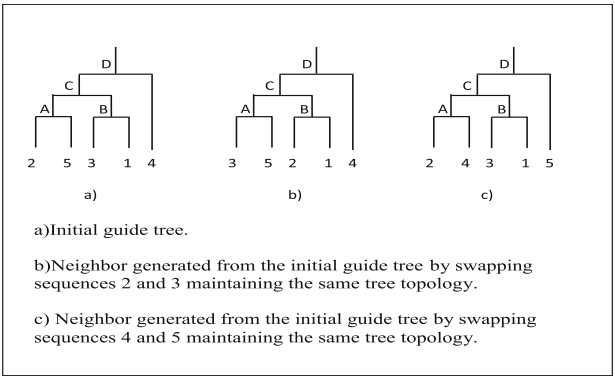

Fig. 2. Two examples of neighbors generated by swapping technique from the initial solution

of the guide tree. This will change the topology of the initial guide tree, and the new guide tree can be considered as a neighbor of the original one.

The neighborhood can be generated randomly by this technique, since the topology of the initial guide tree is not predetermined. However, we can make only n node insertions to obtain exactly n neighbors, by selecting randomly a node to share one of the sequences (leaves) of the guide tree. More precisely, for each sequence, we choose randomly a node and move it to share this sequence, and so on. Figure 3 shows two guide trees (b and c) obtained by inserting nodes to share predetermined sequences of the initial guide tree a.

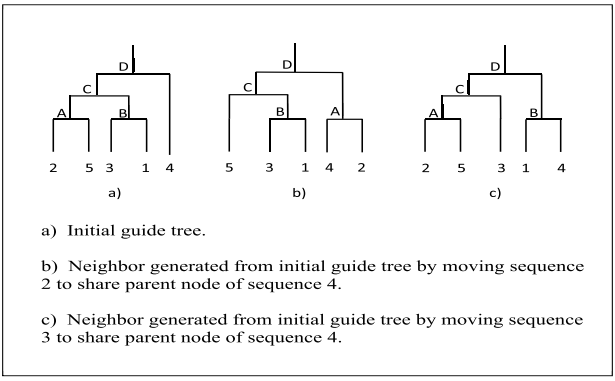

Fig. 3. Two examples of neighbors generated by node insertion technique from the initial solution

Generation by Branch Insertion. Another way to generate neighbors from the current guide tree is the branch insertion, which is moving a branch of the guide tree (or a sub-tree) to another location. The new guide tree resulting of

this move is considered as a neighbor of the current guide tree. This will change the topology of the initial guide tree.

Neighbors are generated randomly by branch insertion move. However, we can make only n branch insertions to generate exactly n neighbors for the current guide tree. For each sequence, we choose randomly a branch (or sub-tree) and move it to share this sequence, and so on. Figure 4 shows two guide trees (b and c) obtained by inserting branches to share predetermined sequences of the initial guide tree a.

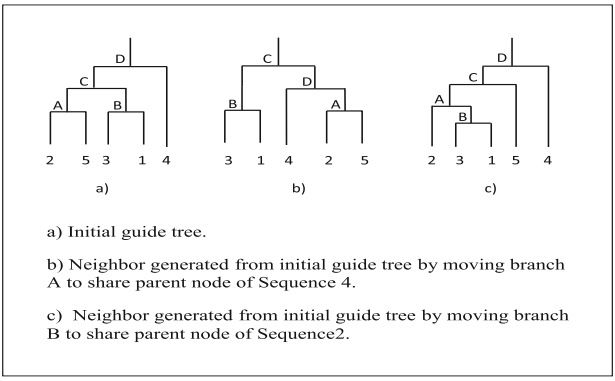

Fig. 4. Two examples of neighbors generated by branch insertion technique from the initial solution

Generation by Distance Variation. The last technique used to generate a neighborhood is the distance variation. Since the initial guide tree is obtained using NJ clustering algorithm, we can produce N different guide trees based on the NJ clustering algorithm, N being defined by the user. Each tree corresponds to a variation of the original obtained by NJ but adding some random noise into the distances in order to introduce some variability. The variation introduced in the guide tree is low enough to keep the distance criteria but significant enough to provide the necessary flexibility to generate multiple alternative trees [22]. Figure 5 shows two guide trees (b and c) produced by adding variation to distances in the NJ clustering algorithm used to obtain the initial guide tree a.

3.3 Intensification Method

Generally, an intensification procedure revisits and examines good solutions. It maintains the good portions of this solution and searches to find a better neighboring solution.

When a single MSA continues to have the highest score for many iterations, the intensification phase aims to escape the local minima by taking out a solution from the tabu list and restart another search process.

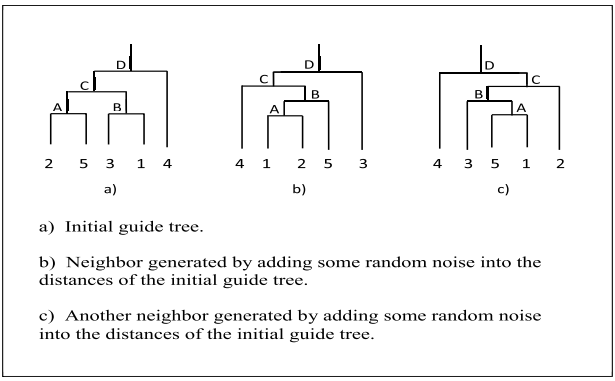

Fig. 5. Examples of neighbors generated by distance variation technique from the initial solution

3.4 Tabu Search Algorithm

Our Tabu Search algorithm consists of generating a neighborhood of a multiple sequence σ using the techniques cited above, i.e. Swapping (SWP), Node insertion (NI), Branch insertion (BI) and Distance variation (DV). The best MSA σ' having the higher score S_{max} is selected for the next iteration and put in the tabu list $TabuList$. This process is iterated until a T_{max} global running time is met . The pseudo-code of our tabu search algorithm is given in Algorithm 1. The details of this algorithm are explained below.

Algorithm 1. Tabu Search Algorithm for MSA

```
1: procedure GTreeTabu
2:     Generate σ an initial MSA using NJ algorithm;
3:     S_max:=Score(σ); σ_max := σ; TabuList:=[];
4:     while not T_max do
5:         Generate a neighborhood N(σ) using: SWP, NI, BI or DV.
6:         set σ' such that
7:         S_σ':=max_{η∈N(σ)}Score(η) and σ' ∉ TabuList
8:         if S_σ' > S_max then
9:             S_max := S_σ'; σ_max := σ'
10:            Insert σ' in TabuList
11:        end if
12:        set σ:=σ'
13:    end while
14: end procedure
```

After generating an initial solution using NJ clustering algorithm, its score is computed. While a time execution T_{max} is not reached, the tabu search is iteratively executed. Each iteration begins by generating the neighborhood of

the current solution by one of the techniques among: Swapping, Node insertion, Branch insertion, Distance variation. For each neighbor, we compute its score in order to set the best neighbor having the highest score as the new current solution. This new solution is inserted in the tabu list which has a variable length depending on the number of iterations with or without improvement. If there is improvement in a certain number of continuously iterations, the length is increased in order to insert other possible solutions. The length of tabu list is decreased if within many iterations there is no improvement. In this case, a solution will be get out from the tabu list in order to restart another search process in the intensification mode.

4 Experimental Results and Discussion

The proposed approach is implemented in MATLAB and tested on Intel Core i3-380M Laptop with 2 GB. To demonstrate the effectiveness of our approach, we have evaluated it on BAliBASE 2 benchmark base [2]. BAliBASE is a database of manually refined multiple sequence alignments. It can be viewed at http://www-igbmc.u-strasbg.fr/BioInfo/BAliBASE2/index.html or can be downloaded from ftp://ftp-igbmc.u-strasbg.fr/pub/BAliBASE2/.

BAliBASE database is divided into five reference sets. Reference 1 contains alignments of equidistant sequences of similar length, with no large insertions or extensions. Reference 2 aligns up to three "orphan" sequences (less than 25% identical) from reference 1 with a family of at least 15 closely related sequences. Reference 3 consists of up to 4 sub-groups, with less than 25% residue identity between sequences from different groups. The alignments are constructed by adding homologous family members to the more distantly related sequences in reference 1. Reference 4 contains alignments of up to 20 sequences including N/C-terminal extensions (up to 400 residues), and Reference 5 consists of alignments including internal insertions (up to 100 residues) [2].

We analyzed the tabu search results from two aspects. The very first set of tests was aimed at to verify the efficiency of our techniques of generating the neighborhood. The techniques are: Swapping (SWP), Node Insertion (NI), Branch Insertion (BI) and Distance Variation (DV). For each neighborhood technique, we ran an extensive set of tests on all the datasets provided by BAliBASE, and computed the scores. The scores using tabu search with each neighborhood generation technique are shown in Table 1. The Number of Test Cases in Reference 1, Reference 2, Reference 3, Reference 4 and Reference 5 are respectively 82, 23, 12, 12 and 12.

One can see in Table 1 that all the neighborhood generation techniques perform well in average for all the reference sets. However, it seems that Branch Insertion and Distance Variation give the best results for all the sequences of Reference 2, Reference 3, Reference 4 and Reference 5. Node insertion gives best results for sequences of Reference 1. We can see that, for all the datasets provided by BAliBASE, Swapping is not the adequate neighborhood technique. This can be explained by the nature of the neighbors generated by a certain technique.

Table 1. Results given by tabu search using four neighborhood generation techniques on the BAliBASE benchmark database

Neighbor-hood	Reference 1	Reference 2	Reference 3	Reference 4	Reference 5	Average
SWP	90.0	93.0	76.3	87.4	85.1	86.36
NI	**90.1**	90.0	78.5	85.6	93.3	87.50
BI	90.05	**93.8**	80.7	**93.7**	**97.9**	**91.23**
DV	90.0	93.5	**82.0**	91.8	95.1	90.48

For the Swapping technique, the neighbors have the same topology, so they are not very different and this will not give more amelioration of the alignment score. For the rest of techniques, the neighbors have not the same topology, but Branch insertion and Distance variation techniques seem to generate more complex guide trees, and this will give more chances to explore different solution spaces and thus, ameliorate the alignment score.

In order to verify the efficiency of our algorithm, we performed another set of tests where the results of our tabu search algorithm using a certain neighborhood technique is compared to other MSA tools. For each references set, we use the adequate neighborhood generation technique which gives the best results, and compare it to the most competitive MSA tools in the literature, such as CLUSTALW 1.83 [28], SAGA [19], MUSCLE [6], ProbCons [5], T-Coffee [20], SPEM [30], PRALINE [27], IMSA ([4] and Tabu Search developed by [24] (called in this paper TS-Riaz) . Except for SAGA and TS-Riaz, which are taken from [24], the results of the other programs are taken from the work of Layeb et al. [16].

The results of our method illustrate clearly the effectiveness of using Tabu Search to perform the multiple sequence alignment. As it can be seen in Table 2, our algorithm performs well in all the references sets. Our method gives good results compared to the other MSA tools. In fact, it gives the second best score for the sequences set Reference 4, the third best score for Reference 3 and Reference 5, and it is in the fourth place for the remaining sets, i.e. Reference 1 and Reference 2. We can see in Table 2 that our Tabu search using Branch Insertion neighborhood technique has a good place for three sequences sets over five, i.e. Reference 2, Reference 4 and Reference 5. Using the Distance Variation neighborhood technique gives the third best score for Reference 3 set, and Node Insertion gives the fourth best score for Reference 1. It can be seen overall, that our tabu search method using Branch Insertion neighborhood technique gives in average the second best score compared to the other algorithms studied in the paper.

Table 2. Results given by Tabu Search using neighborhood techniques compared with other methods on the BAliBASE benchmark database.

Method	Reference 1	Reference 2	Reference 3	Reference 4	Reference 5	Average
CLUSTALW	85.8	93.3	72.3	83.4	85.8	84.12
SAGA	82.5	**95.4**	77.7	78.0	86.8	84.08
MUSCLE	**90.3**	64.4	**82.2**	91.8	**98.1**	85.36
ProbCons	90.0	**94.0**	**82.3**	90.9	**98.1**	91.06
T-Coffee	86.8	**93.9**	76.7	92.1	94.6	88.82
SPEM	**90.8**	93.4	81.4	**97.4**	97.4	**92.08**
PRALINE	**90.4**	94.0	76.4	79.9	81.8	84.5
IMSA	83.4	92.1	78.6	73.0	83.6	82.14
TS-Riaz	76.0	88.9	71.5	77.3	90.5	80.84
TS-SWP	90.0	93.0	76.3	87.4	85.1	86.36
TS-NI	**90.1**	90.0	78.5	85.6	93.3	87.50
TS-BI	90.05	**93.8**	80.7	**93.7**	**97.9**	**91.23**
TS-DV	90.0	93.5	**82.0**	91.8	95.1	90.48

5 Conclusion

In this paper we have demonstrated the efficiency of using tabu search to align multiple sequences. Our algorithm uses several neighborhood generation techniques. To evaluate our approach, we have used BAliBASE benchmark. Firstly, we studied different techniques to produce the neighborhood, then we compared our algorithm to the most recent and competitive MSA tools. We have observed through experiments on BAliBASE that for Reference 1 and Reference 2, the alignments generated by our method are encouraged. For the remaining references, tabu search performs better than most of the other methods studied in this paper.

There are several issues for future work. First, tabu search comes with a number of parameters that can be experimented with to observe the respective effect on the search process. The parameters like tabu list size, tabu tenure, termination criteria, and neighborhood size can have a direct influence on the quality of the final alignment. Further studies are needed to test different scoring schemes and tabu search features.

References

1. Abbas, A., Holmes, S.: Bioinformatics and management science: some common tools and techniques. Operations Research 52(2), 165–190 (2004)
2. Bahr, A., Thompson, J.D., Thierry, J.C., Poch, O.: BAliBASE (benchmark alignment database): enhancements for repeats, transmembrane sequences and circular permutations. Nucleic Acids Res. 29(1), 323–326 (2001)
3. Blum, C., Valles, M.Y., Blesa, M.J.: An ant colony optimization algorithm for DNA sequencing by hybridization. Computers and Operations Research 38, 3620–3635 (2008)
4. Cutello, V., Nicosia, G., Pavone, M., Prizzi, I.: Protein multiple sequence alignment by hybrid bio-inspired algorithms. Nucleic Acids Research 39(6), 1980–1990 (2010)
5. Do, C., Mahabhashyam, M., Brudno, M., Batzoglou, S.: ProbCons: Probabilistic consistency-based multiple sequence alignment. Genome Res. 15(2), 330–340 (2005)
6. Edgar, R.: MUSCLE: Multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 32, 1792–1797 (2004)
7. Feng, D., Doolittle, R.: Progressive sequence alignment as a prerequisite to correct phylogenetic trees. Journal of Molecular Evolution 24(4), 351–360 (1987)
8. Glover, F., Laguna, M.: Tabu Search. Kluwer Academic Publishers, Boston (1997)
9. Glover, F., Taillard, E., de Werra, D.: A user's guide to tabu search. Ann. Oper. Res. 41, 3–28 (1993)
10. Gotoh, O.: Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. J. Mol. Biol. 264, 823–838 (1996)
11. Gupta, S.K., Kececioglu, J.D., Schaffer, A.A.: Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. J. Comp. Biol. 2(3), 459–472 (1995)
12. Higgins, D.G., Taylor, W.R.: Multiple sequence alignment, Protein Structure Prediction -Methods and Protocols. Humana Press (2000)
13. Kemena, C., Notredame, C.: Upcoming challenges for multiple sequence alignment methods in the high-throughput era. Bioinformatics 25, 2455–2465 (2009)
14. Kim, J., Pramanik, S., Chung, M.J.: Multiple sequence alignment using simulated annealing. Comp. Applic. Biosci. 10(4), 419–472 (1994)
15. Lalwani, S., Kumar, R., Gupta, N.: A review on particle swarm optimization variants and their applications to multiple sequence alignments. Journal of Applied Mathematics and Bioinformatics 3(2), 87–124 (2013)
16. Layeb, A., Selmane, M., Bencheikh ELhoucine, M.: A new greedy randomized adaptive search procedure for multiple sequence alignment. International Journal of Bioinformatics Research and Applications (2011)
17. Lightner, C.: A Tabu Search Approach to Multiple Sequence Alignment. Ph.D. thesis, North Carolina State University, Raleigh, North Carolina (2008)

18. Lipman, D., Altschul, S., Kececioglu, J.: A tool for multiple sequence alignment. Proc. Natl. Acad. Sci. 86, 4412–4415 (1989)
19. Notredame, C., Higgins, D.G.: SAGA: Sequence alignment by genetic algorithm. Nucl. Acids Res. 24, 1515–1524 (1996)
20. Notredame, C., Higgins, D., Heringa, J.: T-Coffee: a novel method for fast and accurate multiple sequence alignment. J. Mol. Biol. 302, 205–217 (2000)
21. Notredame, C., Holmes, L., Higgins, D.: COFFEE: an objective function for multiple sequence alignments. Bioinformatics 14(5), 407–422 (1998)
22. Orobitg, M., Guitaro, F., Cores, F., Llados, J., Notredame, C.: High performance computing improvements on bioinformatics consistency-based multiple sequence alignment tools (2014), http://dx.doi.org/10.1016/j.parco.2014.09.010
23. Reinert, K., Stoye, J., Will, T.: An iterative method for faster sum-of-pairs multiple sequence alignment. Bioinformatics 16, 808–814 (2000)
24. Riaz, T., Wang, Y., Li, K.: Multiple sequence alignment using tabu search. In: Proceeding of Asia-Pacific Bioinformatics Conference (APBC 2004), pp. 1–10 (2004)
25. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. 4(4), 406–425 (1987)
26. Shyu, C., Sheneman, L., Foster, J.: Multiple sequence alignment with evolutionary computation. Genetic Programming and Evolvable Machines 5, 121–144 (2004)
27. Simossis, V., Heringa, J.: PRALINE: a multiple sequence alignment toolbox that integrates homology-extended and secondary structure information. Nucleic Acids Res. 33, 289–294 (2005)
28. Thompson, J., Higgins, D., Gibson, T.: ClustalW: improving the sensitivity of progressive multiple sequence weighting, position-specific gap penalties and weight matrix choice. Nucleic Acids Res. 22, 4673–4680 (1994)
29. Yokoyama, T., Watanabe, T., Taneda, A., Shimizu, T.: A web server for multiple sequence alignment using genetic algorithm. Genome Informatics, 12, 382–383 (2001)
30. Zhou, H., Zhou, Y.: SPEM: improving multiple sequence alignment with sequence profiles and predicted secondary structures. Bioinformatics 21, 3615–3621 (2005)