Computer Methods and Programs in Biomedicine 208 (2021) 106237

Contents lists available at ScienceDirect

Computer Methods and Programs in Biomedicine

journal homepage: www.elsevier.com/locate/cmpb

Accurate consistency-based MSA reducing the memory footprint

Jordi Lladós*, Fernando Cores, Fernando Guirado, Josep L. Lérida

INSPIRES Research Center, Universitat de Lleida. Jaume II. 69, 25001 Lleida, Spain

ARTICLE INFO

Article history:
Received 13 December 2020
Accepted 8 June 2021

Keywords:
Multiple sequence alignment
Consistency
T-coffee
Dynamic programming

ABSTRACT

Background and Objective: The emergence of Next-Generation sequencing has created a push for faster and more accurate multiple sequence alignment tools. The growing number of sequences and their longer sizes, which require the use of increased system resources and produce less accurate results, are heavily challenging to these applications. Consistency-based methods have the most intensive CPU and memory usage requirements. We hypothesize that library reductions can enhance the scalability and performance of consistency-based multiple sequence alignment tools; however, we have previously shown a noticeable impact on the accuracy when extreme reductions were performed. **Methods:** In this study, we propose Matrix-Based T-Coffee, a consistency-based method that uses library reductions in conjunction with a complementary objective function. The proposed method, implemented in T-Coffee, can mitigate the accuracy loss caused by low memory resources. **Results:** The use of a complementary objective function with a library reduction of ≥30% improved the accuracy of T-Coffee. Interestingly, ≥50% library reduction achieved lower execution times and better overall scalability. **Conclusions:** Matrix-Based T-Coffee benefits from accurate alignments while achieving better scalability. This leads to a reduction in memory footprint and execution time. In addition, these enhancements could be applied to other aligners based on consistency.

1. Introduction

Multiple sequence alignment (MSA) is important in several research domains in molecular biology and bioinformatics, such as epidemiology, phylogenetic tree reconstruction, 3D structure prediction, and hidden Markov modelling (HMM). These fields use MSA to infer residue-level homology, structural, or functional identity [1,2].

The optimal alignment of two sequences can be performed using the Needleman-Wunsch (NW) algorithm and dynamic programming [3,4]. All the best possible alignments are found by completing the Dynamic Programming (DP) matrix according to three score parameters: match, mismatch, and gap scores, which are obtained using substitution matrices such as PAM and BLOSUM. Next, the algorithm generates the optimal path using a traceback technique. The time complexity for this algorithm grows exponentially with the number of sequences and their lengths; therefore, heuristic algorithms are required. In this case, the goal of MSA tools is to seek an alignment that maximizes its accuracy, as approximated by the sum of the similarities for all pairs of sequences (SP score) or the Total Column (TC) score.

Progressive alignment is a widely used heuristic. This process builds a final MSA by combining the pairwise alignments. It starts by using the most similar pair of sequences and ends with the more distantly related, following the order of a guide tree. The most popular progressive alignment implementation is the Clustal family; ClustalW [5] and ClustalΩ [6] are the most representative approaches. The biggest drawback of this method is that if an error is made in the initial steps of the alignment, this is propagated until it reaches the root of the tree. Therefore, other approximations based on the progressive method have appeared, such as iterative algorithms or consistency-based methods. In iterative algorithms, such as MUSCLE [7], MAFFT [8], and ProbCons [9], the greediness of the progressive alignment method is overcome through a process of alignment refinement that optimizes the obtained result. Evolutionary and genetic algorithms [10–12] are an enhancement of the iterative algorithms that use a stochastic process to improve the final solution. Alternatively, consistency-based methods use consistency information about different pairwise alignments to improve the result. T-Coffee (TC) [13] is the most representative method in this category. It combines the COFFEE consistency-based scoring function [14] with the progressive alignment algorithm.

* Corresponding author.
E-mail addresses: jordi.llados@udl.cat (J. Lladós), fernando.cores@udl.cat (F. Cores), fernando.guirado@udl.cat (F. Guirado), josepllluis.lerida@udl.cat (J.L. Lérida).

https://doi.org/10.1016/j.cmpb.2021.106237

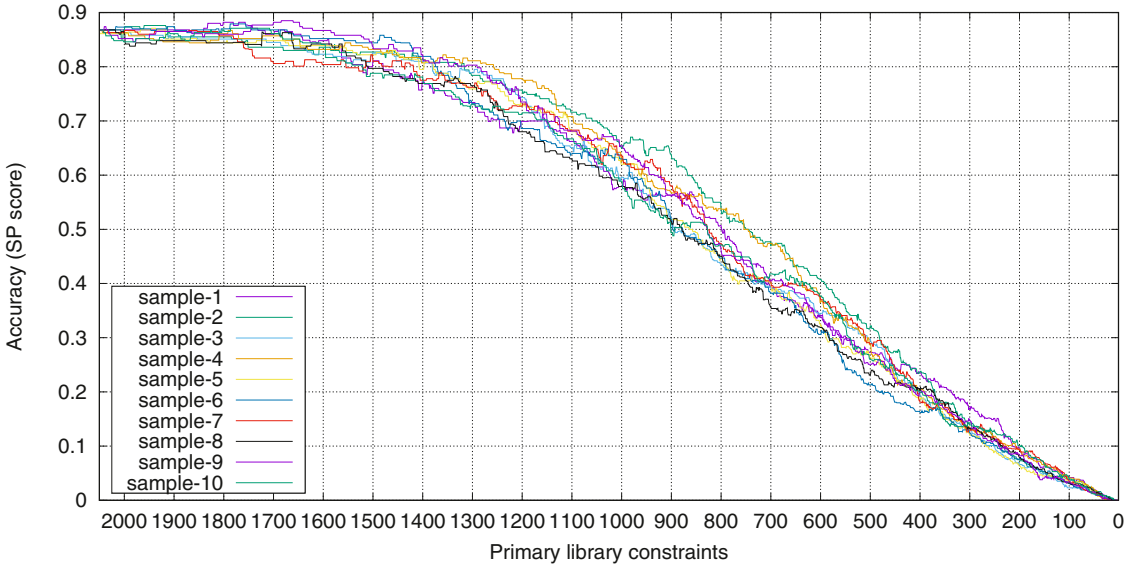

Fig. 1. Random discarding policy when solving the BB12020 dataset.

Other common MSA programs that use consistency are Probcons [15], Probalign [16], and the L-INS-i variant of MAFFT [8].

Recent developments in genome sequencing have increased the necessity for reliable and fast MSA methods that take into account both the number of the sequences to be aligned and the size (i.e., the number of residues) of each alignment. Consistency-based methods are more stable from the point of view of accuracy when there is an increase in the number of sequences. These methods have the ability to consider pairwise information from all the sequences during each step of the progressive alignment. This information allows the alignment of two residues that would be separated without having a global view of how they are aligned within the rest of the sequences. Nonetheless, these methods are severely limited by their memory requirements due to the consistency library size. For example, in T-Coffee, the consistency library size is $O(N^2L^2)$, where N is the number of sequences and L the length of the sequence. These requirements considerably limit performance and scalability.

There are two main approaches to improving the scalability of consistency-based aligners. The first approach increase the volume of computational resources (memory, disk) available to store and process consistency information. This can be accomplished by distributing and processing the consistency information in parallel using multiple computers via distributed paradigms and frameworks, such as MPI and Hadoop-Spark. The main drawback of this alternative is that the quadratic growth of the consistency requirements leads to significant costs. Furthermore, its applicability is limited to a few thousand sequences.

The second approach is to reduce the size of the consistency library by discarding the least significant information from an alignment point of view. However, it is very difficult to know which consistency data is significant when assessing with biological datasets. Therefore, reducing the size of the consistency library can improve the scalability and performance of the consistency-based methods but this reduction can negatively affect the alignment accuracy. This impact is negligible for small and medium datasets; however, the required volume of memory is extremely large when aligning large datasets. This leads to high library reduction to fit the computer memory constraints. In such cases, the loss of accuracy is more noticeable; therefore, consistency MSA tools with memory reduction are not viable.

This effect is demonstrated in Fig. 1, where the alignment quality for the BB12020 BAliBASE dataset is plotted when the number of constraints in the library is diminished by randomly discarding them one by one. This specific dataset was shown due the small number of constraints, and multiple samples were generated to ensure that the behaviour when discarding constraints was representative. We found that more constraints being discarded led to a loss of accuracy. Furthermore, these data highlight the importance of a good constraint discarding policy.

To reduce the memory requirements of the consistency-based MSA tools, a previous study [17] presented a library-generation optimization method that was capable of reducing the amount of consistency data stored. In this library-generation optimization method, the user defines the maximum library size that should be maintained and the method selects the constraints to be stored. The Memory-Efficient Library (MEL) constraint selection policy takes advantage of the constraint selection rules derived from the analysis of the consistency that appear in the optimal alignments. MEL is based on prioritizing higher weighted constraints, discarding residues evenly across all sequences, and balancing the consistency among the alignment columns.

The proposed MEL library reduction method was implemented in T-Coffee, T-Coffee-MEL[1], and its impact is shown in Fig. 2, where the behaviour of the random discarding policy is also plotted for comparison. These data show that accuracy is better maintained using the MEL when compared with the random policy, with the latter showing a linear fall in accuracy. However, when an extreme reduction is applied, the accuracy remains highly compromised.

Further analysis showed that this effect is produced by the dynamic programming stage, which incorporates alignment generation. In consistency-based methods, the DP matrix is populated using an objective function that represents the similarities between the residues of the sequences that will be aligned, based on consistency information. This produces the final alignment using a backtrack technique. When the consistency library is smaller because some constraints have been discarded, the objective function may not return any score; therefore, the DP matrix is populated by empty cells. Under these circumstances, the backtrack step cannot determine the correctness of the residue alignment or whether a

[1] The T-Coffee-MEL source and its installation instructions can be found at: github.com/jllados/TCoffee-MEL.

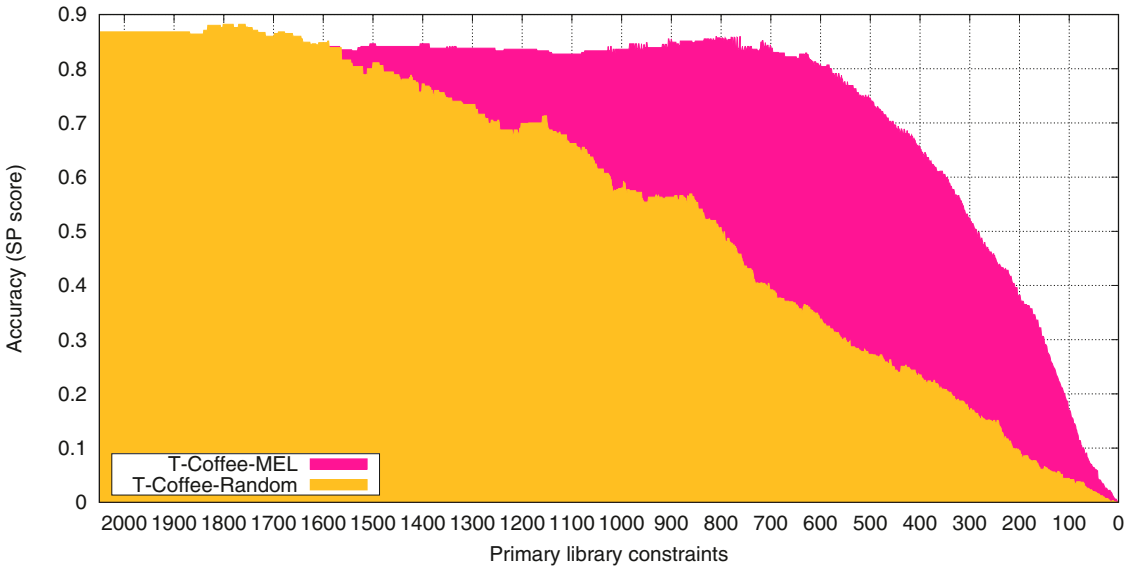

Fig. 2. Comparison between the MEL and the random discarding policy for the BB12020 dataset.

gap is required between them. This latter option greatly increases the length of the final alignment and negatively affects the accuracy.

In this paper, we propose an innovative method that complements the COFFEE objective function with substitution matrices to populate the DP Matrix gaps generated by library reduction. This new approach has low-memory requirements; therefore, it can scale up with increases in the number of sequences. The combination of the two methods, called the Matrix-Based T-Coffee (MBT-Coffee[2]) produce more accurate alignments with lower execution times.

2. Methods

The MBT method is based on the T-Coffee consistency method, where the consistency library has been reduced by applying the MEL optimization method [17], and completed in the progressive alignment stage using substitution matrices.

Library generation proceeds in two phases: (1) the consistency library and (2) extended library construction. In T-Coffee, the consistency library is generated prior to the alignment stage, from computing all-against-all possible pairwise alignments of the sequences. This is stored as a list of residue matches between those sequences, which can be represented by an $S_{1..N}xS_{1..N}$ matrix (Fig. 3), where each cell, $(S_i - S_j)$ when $i \neq j$, contains a list of residue matches between those sequences. Each residue match is represented by a constraint, $\{x, y, W_{(x,y)}\}$, where x is a residue of sequence i matched with y, a residue of sequence j, and a weight $W_{(x,y)}$, representing its correctness. Each constraint is used in the progressive alignment stage to fill the dynamic programming matrix.

The consistency library provides information about the consistency of each sequence pair. The extended library, which is created on-the-fly during the progressive alignment stage, is a reweighting process that uses the transitivity property to include indirect information about constraints. For example, in an MSA containing three sequences, x, y, and z, if position x_m aligns with position z_k, and position z_k aligns with y_n in the projected x-z and z-y alignments, then the x_m must align with y_n in the projected x-y alignment for consistency.

An example of how the extension improves the final alignment is presented in Fig. 4. Briefly, a pairwise alignment between S1 and S2 can produce an initial incorrect MSA because the global information for alignment of all the sequences is not available. The library extension enables the use of the transitivity property, which takes into account the effect of residue alignment on other sequences, which in turn, affects the initial alignment of S1 and S2.

Thus, the MEL policies are applied during the primary library building to reduce the amount of consistency data stored. MEL dictates whether a constraint should be maintained in the consistency library, taking into account the threshold of maximum memory allowed by the user.

The implementation of MEL is based on a temporary queue structure in which all the library constraints are stored and sorted by their weight. Next, each constraint is evaluated to determine its place in the library. The evaluation function checks the memory available, whether the evaluated pairwise alignment is a leaf node of the progressive alignment guide tree, and the number of constraints already assigned to this part of the domain. As result, the final library has a subset of constraints that are prioritized according to the following criteria: (1) have a high weight; (2) pertain to closely related sequences, the closer to the guide-tree leaves the better; and (3) cover all the domains of the alignment, were each column of the sequence is represented.

Due the use of consistency library reductions the COFFEE scores retrieved by the dynamic programming matrix may be incomplete or empty. This significantly reduces the memory consumption; however, it has a negative impact on the accuracy. The solution proposed in the present paper is able to mitigate the accuracy decrease via complement of the objective function. It complements the original COFFEE score by aggregation, therefore a combination of two scores is always used. This implies that if a cell is empty in the dynamic matrix, all the weight of evaluating it falls to the complementary score. The complementary proposal has to accomplish two essential requirements: (1) the new score must be calculated without the need for large computing resources (memory and CPU) so that an average workstation is not compromised and (2) it must provide high-quality information capable of improving the resulting alignment.

The most commonly used objective functions in the MSA field are based on a substitution matrix; however, their low accuracy

[2] The MBT-Coffee source and its installation instructions can be found at: github.com/jllados/MBT-Coffee.

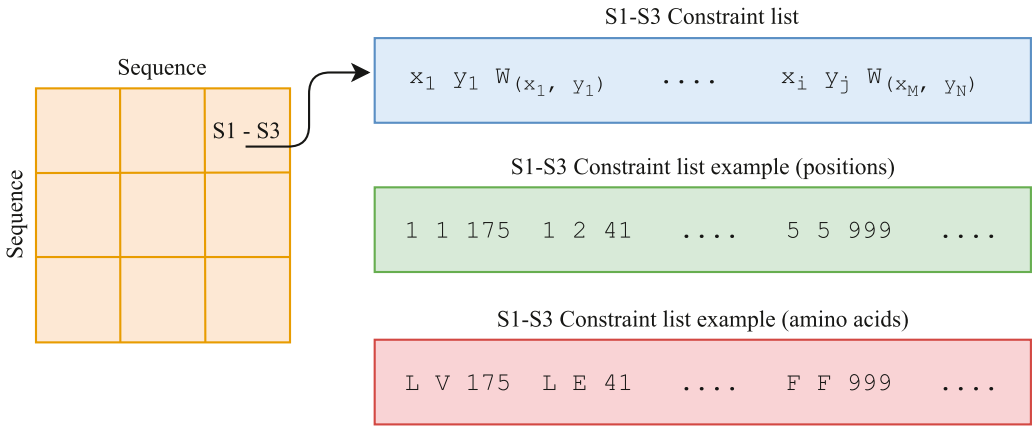

Fig. 3. T-Coffee library structure example.

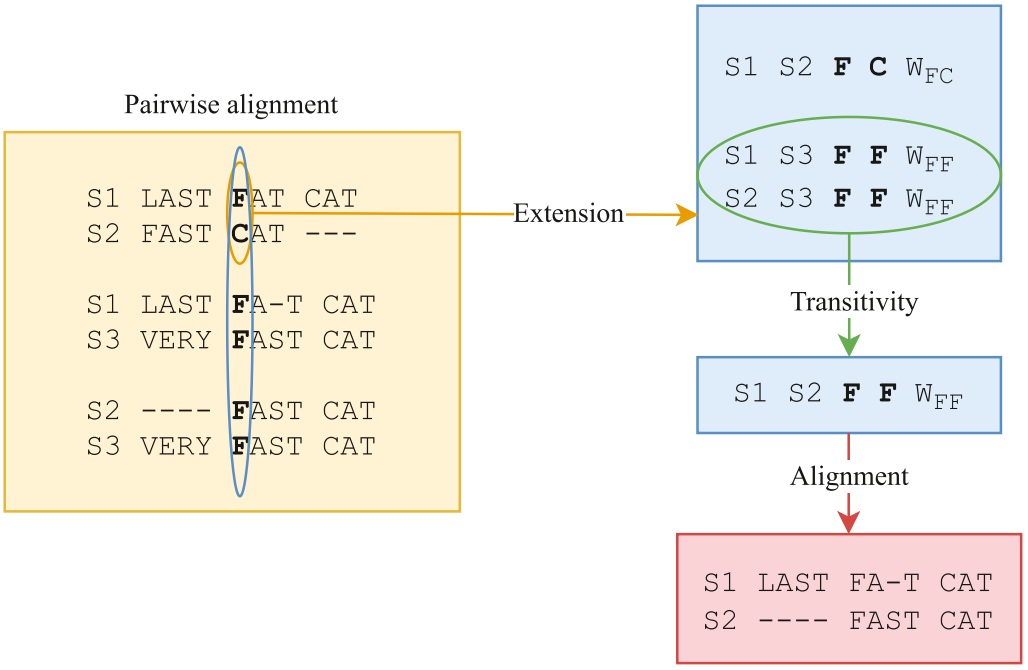

Fig. 4. T-Coffee extended library example.

results have led to the development of refining techniques. Our method is based on the same approach, when an empty cell from the consistency library is accessed, it provides a score generated through the substitution matrix.

A crucial aspect of our proposal is to determine the substitution matrix and the gap penalties to be used. There is controversy over the optimal substitution matrix for any sequence alignment. Blosum, PAM, and GONNET were the most widely used to find evolutionarily divergent protein sequences; however, recent papers have shown the importance of using an adaptive matrix solution, in which different similarity scoring matrices are applied depending on the similarity of the sequences being aligned.

Edgar [18] proved that the use of low-identity matrices does not produce more accurate alignments. Furthermore, they demonstrated an improvement using the VTML200 matrix and empirically tested the best combination of gap penalties that work with the VTML200 matrix: 2 and 0.1 for the open and extended penalty, respectively. To decide the substitution matrix, the most used ones in the literature were empirically tested, obtaining similar average accuracy according to different sequence variability. Given our empiric test and Edgar results [18], we chose to apply the VTML200 substitution matrix in our proposal.

Multiple guide tree alternatives were also tested. This is important as the sequences are progressively aligned according to the topology of the guide tree. It is well known that a guide tree can produce better alignments according to the percentage of similarity of the sequences. In addition, it can also influence the insertion of gaps [19]. T-Coffee defaults to NJ trees, however we obtained a better compromise using UPGMA trees (boostrap=1000). Both methods are commonly used in MSA, although they do not scale well when aligning thousands of sequences.

The supplementary material shows the choices made in more detail.

2.1. Matrix-based T-Coffee algorithm

To demonstrate that our proposal improved the objective function and correctly replaced the missing information in the reduced consistency library, we used T-Coffee as the evaluation aligner. This was because we were familiar with its protocol and code. However,

our proposal can be implemented in any MSA tool based on a consistency library.

This implementation, MBT-Coffee, maintains the main structure of T-Coffee and starts building the consistency library while also applying our MEL library reduction (Algorithm 1, line 1). This leads

Algorithm 1 MBT-Coffee.

```
1: Calculate the library with MEL
2: for each node ∈ GUIDE TREE do
3:     S_i= Left child Node S_j= Right child Node
4:     Calculate the profile matrices for S_i and S_j
5:     Calculate the gap coefficients
6:     DYNAMIC_PROGRAMMING(S_i[1..M] and S_j[1...N])
7: end for

8: procedure DYNAMIC_PROGRAMMING(S_i, S_j, M, N)
9:     Calculate the minimum conversion cost for S_i and S_j
10:    Divide: Find optimum midpoint (mid_i, mid_j)
11:    Conquer: recursively around midpoint
12:    DYNAMIC_PROGRAMMING(S_i, S_j, mid_i, mid_j)
13:    DYNAMIC_PROGRAMMING(S_i + mid_i, S_j + mid_j, M − mid_i, N − mid_j)
14: end procedure
```

to an optimized library in the memory space.

Once the reduced library has been built, the progressive alignment phase begins following the order provided by the guide tree (Algorithm 1, line 2). For each iteration of the alignment, two sequences or two aligned groups of sequences, represented by the the left child node and right child node, are aligned generating a new one. For each child node a scoring matrix is built. These matrices represents each position of a residue in the sequence or the profile (an aligned group of sequences) with possible amino acid combinations, which are scored by the VTML200 matrix (Algorithm 1, lines 4-5). These are used in the dynamic programming phase to determine the complementary score of each residue match by averaging the pairwise combinations (*residue_position* – *amino_acid*) of $(S_i - S_j)$.

The next step initiates the recursive dynamic programming phase. In the present study, we used Myers and Miller [3] dynamic programming. This is based on the divide-and-conquer optimization method to save memory space for the storage of the backtrack matrix. The algorithm optimizes the alignment of two sequences by reducing the number of cells that are evaluated in the dynamic programming matrix. First, it divides the length of S_i in half (midpoint). Next, the best residue, S_j, is found to align to the midpoint. All pairs of residues are calculated recursively on left-upper and right-lower sides of the midpoint until aligned. The Myers and Miller algorithm performs better than the T-Coffee default method aligner, which has shown accuracy losses when the length of the sequences increases [20].

Thus, when aligning the sequences, the dynamic programming (Algorithm 1, lines 8-14) generates the corresponding calls to the modified functions to populate the scores in the dynamic programming matrix. To perform this, we modified two essential functions. First, the objective function, which initially evaluated the COFFEE score for $(S_i - S_j,\ x_m - y_n)$ as the sum of the weight of the given constraint plus the constraints generated via the transitivity property. These were divided by the sum of the weights related with $(S_i - x_m)$ and $(S_j - y_n)$.

We assessed the COFFEE score against the score obtained by the average pairwise matrix scores for the same combination of sequences/residues. Multiple combinations of both scores were tested against different library sizes. Overall, a more heavily weighted COFFEE score obtained better accuracy results. Therefore, the factor from the complementary score was fixed to 25%.

Second, we altered the affine gap cost. The original version of the algorithm uses the following formula to evaluate the cost of a residue:

$$f(k) = \begin{cases} g + h * (k), & \text{if } k > 0. \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Where k is a residue position, g is the penalty for opening a new gap and h is the penalty for extending an existing gap by one residue. We adapted the dynamic programming method to use specific gap values for each combination of (*residue_position* – *amino_acid*). This values are generated by literature proven methods. Such methods increase or decrease the gap penalty when a criterion is met. These criteria are evaluated sequentially, so when one is met, the evaluation stops. 1) The penalties are lowered when a gap is found, otherwise they are increased, 2) if a hydrophilic amino acid sequence is identified, the penalty is lowered, and 3) the penalty is modified based on the probability that the residue precedes a gap in known protein structures. Overall, a position specific gap penalty system provided a better alignment when compared with fixed gap penalties.

3. Results

This study was conducted with three main goals: (1) to validate the improved accuracy when applying the MBT-Coffee; (2) to confirm that accuracy is maintained when the number of sequences increases; and (3) to evaluate the computational execution time of the new algorithm. The execution environment was one computing node with 2x Intel(R) Xeon(R) CPU E5-2609 v4 1.70GH and 64GB DDR3 RAM.

Accuracy testing was performed using the BAliBASE 4.0 dataset [21]. This database includes high-quality documented and manually refined reference alignments based on 3D structural superpositions. The scalability evaluation, which requires thousands of sequences, was performed with the HomFam [22] benchmarking suite, which provides large datasets using Pfam families. To validate the results of aligning a Pfam family, the Homstrad site includes some reference alignments for the corresponding Pfam family that were previously de-aligned and shuffled into the dataset. After the alignment process, the reference sequences were extracted and compared with the originals in Homstrad using qscore [23]. This tool determines the accuracy of the alignment by the fraction of identically, and thus correctly, aligned amino acid pairs between a test and reference MSA.

HomFam contains almost one hundred sets. We randomly selected five families to evaluate our proposal (GEL, PDZ, rrm, rvp, and sdr). The results for each experiment correspond to five executions with the aim of verifying the robustness of MBT-Coffee.

3.1. BAliBASE evaluation: library reduction and alignment accuracy

First, we evaluated the accuracy for the whole BAliBASE benchmark; the initial consistency library size was reduced until it was completely empty. For each reduction step, we executed our MBT-Coffee proposal and compared the results with T-Coffee-MEL.

Figure 5 a shows the size of the library compared with the percentage of the memory size cleared by the reduction. These results show that when the library size decreases, all methods tend to produce lower-accuracy alignments. While the accuracy of the MEL approach was compromised when the library contains a very low number of constraints, MTB-Coffee was able to maintain a good result.

Figure 5 b shows the execution time required to align the whole BAliBASE. Having fewer entries in the library had repercussions on

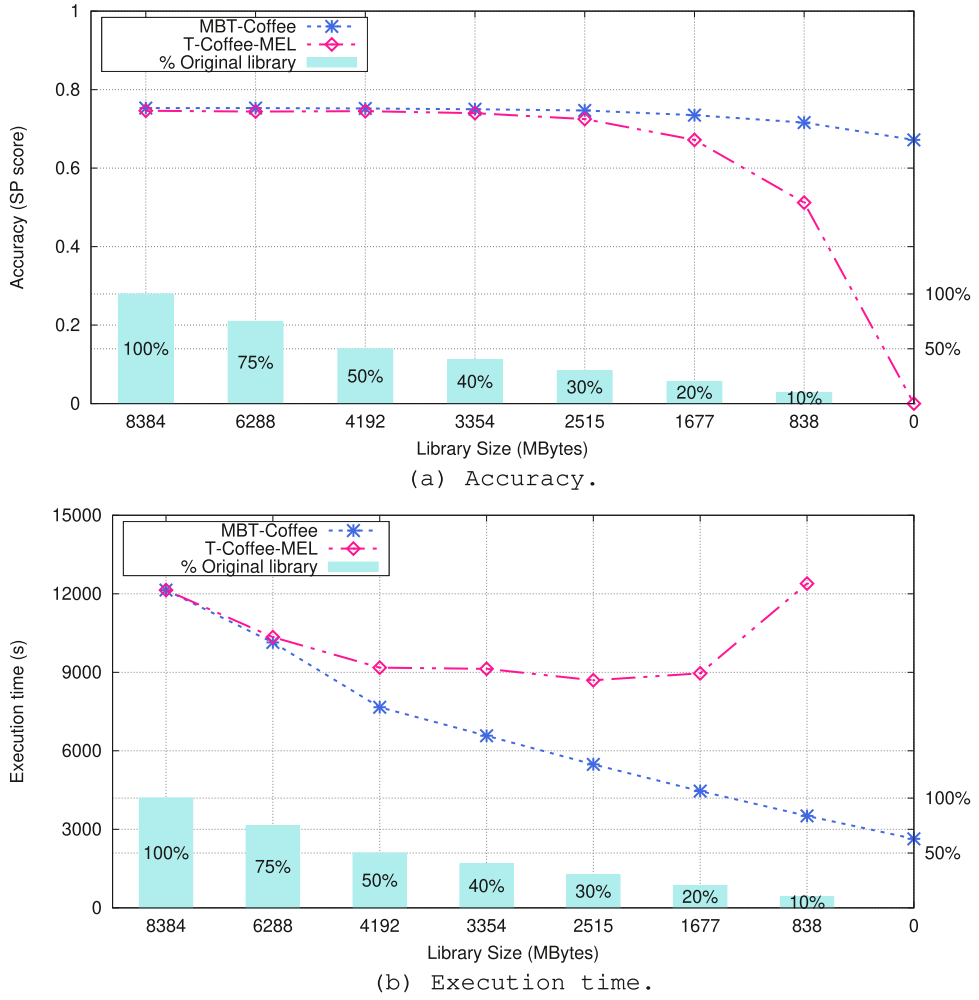

Fig. 5. Comparison of MBT-Coffee and T-Coffee-MEL with multiple reductions using BAliBASE.

the computation of the COFFEE function; it led to more empty cells in the T-Coffee-MEL dynamic programming matrix, which produced longer alignments because more gaps are included. More computation is required in the dynamic programming for longer partial alignments as it depends on the length of the sequences. This is observed by the longer execution times. MBT-Coffee was not affected by this effect because the substitution matrix led to fewer gaps. Thus, it obtained a nearly linear reduction in execution time when the library is reduced.

Figure 6 shows the concrete results for the BB12020 dataset. There was a significant improvement in accuracy when the MBT-Coffee protocol was applied. In the worst case, with an empty library and no consistency values, MBT-Coffee showed an alignment accuracy of 0.78, losing only 9% when compared with using a whole library.

Table 1 shows the differences in accuracy between MBT-Coffee, MEL, and the original T-Coffee. The latter does not have any type of library reduction implemented; therefore, it uses the whole consistency library and maintains the same value for all the experimental cases. These data show that a 2515 MB (30% of the library) reduction maintains a better accuracy than the original T-Coffee with the minimum library size. This represents a large memory reduction and lowers the execution time by 37% and 55% for MEL and T-Coffee, respectively.

In addition, we compared the accuracy of the MBT-Coffee with a 50% reduction to those obtained with other MSA tools in the literature (Table 2). The first column indicates the aligner used. The results for the BAliBASE subgrouping are in columns 2-7, and the last column refers to the average score over all families. These results show that MTB-Coffee is the third best aligner, surpassing the average accuracy of T-Coffee (0.752 vs. 0.743) and ClustalΩ (0.752 vs. 0.748).

Table 1
Accuracy comparison of MBT-Coffee, T-Coffee-MEL and T-Coffee with multiple reductions using the SP score from BAliBASE.

Library MB (%)	MSA tool		
	MBT-Coffee	T-Coffee-MEL	T-Coffee
8384 (100%)	0.753	0.746	0.743
6288 (75%)	0.753	0.744	–
4192 (50%)	0.752	0.745	–
3354 (40%)	0.750	0.740	–
2515 (30%)	0.747	0.725	–
1677 (20%)	0.735	0.672	–
838 (10%)	0.716	0.512	–
0 (0%)	0.672	0	–

3.2. Sequence scalability evaluation

In the experimental study, we chose the 50% memory reduction scenario as it provides a good compromise between accuracy and memory usage. We decided not to test MEL separately because it is incorporated into the MBT-Coffee protocol. The datasets evaluated were extracted from the HomFam benchmark, adding from 100 to

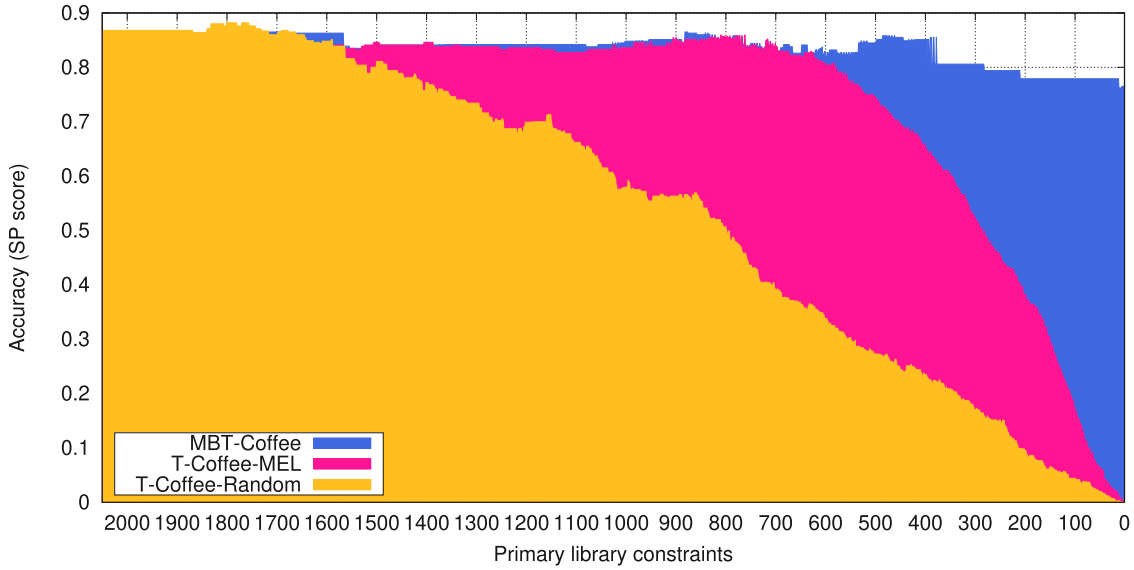

Fig. 6. Comparison of MBT-Coffee, T-Coffee-MEL and T-Coffee with a random discard policy for the BB12020 dataset.

Table 2
Accuracy comparison of MSA tools using the SP score from BAliBASE.

MSA tool	BAliBASE Reference						Average
	RV11	RV12	RV20	RV30	RV40	RV50	
MSAProbs	0.562	0.885	0.852	0.765	0.827	0.782	0.785
MAFFT L-INS-i	0.521	0.874	0.845	0.762	0.829	0.776	0.771
MBT-Coffee 50%	0.526	0.875	0.834	0.737	0.779	0.759	0.752
ClustalΩ	0.481	0.847	0.823	0.760	0.799	0.736	0.748
T-Coffee	0.502	0.845	0.820	0.730	0.800	0.733	0.743
Muscle	0.465	0.846	0.809	0.713	0.760	0.706	0.724
Dialign-tx	0.423	0.814	0.789	0.648	0.710	0.662	0.682
ClustalW	0.415	0.798	0.773	0.636	0.696	0.649	0.669

1,000 sequences, so that they can be aligned with all the methods used in the comparison.

Figure 7 a shows the accuracy compared between MBT-Coffee and T-Coffee. There was an improvement in the accuracy when more sequences are added, which was unexpected. In a previous study [24], we have reported that some constraints in the library may add noise to the alignment instead of useful information, which is much more noticeable in larger datasets. However, those constraints tend to be minimized when the library is reduced. The use of these approaches, new scoring function, and library reduction complement each other, which results in incremental changes in average accuracy. Fig. 7 b shows that the time required to execute the progressive alignment decreases by half when compared with the original T-Coffee.

Table 3 compares the results obtained with those from other MSA tools. The first column indicates the aligner used. The accuracy results for HomFam when adding sequences from 100 to 1,000 are in columns 2-5, and the last column refers to the average score among all the sequence sizes. These data show that MBT-Coffee with a 50% library had a superior performance when compared with all other aligners.

To further evaluate the improvement of the MBT-Coffee, a second analysis is performed on the alignments obtained in Table 3. This analysis focuses on demonstrating that the alignments have been improved based on a parsimony score. The maximum parsimony [25] is an optimization method which objective is to identify the most parsimonious tree. To this end, it determines the minimum number of changes required in a given phylogeny when a cost is associated to transitions between character states. Using the parsimony score, we can evaluate a precomputed tree to assess how parsimonious it is. Table 4 shows the results. The parsimony score is calculated by generating a phylogenetic tree from the final alignment and based on this, the parsimony is calculated using the optimally criteria from Sankoff algorithm [26]. The data shows that MBT-Coffee 50% achieves better Sankoff parsimony than T-Coffee. Although in this case, the MAFFT family of aligners performs better, while the Clustal family worsens.

Table 3
Accuracy comparison of MSA tools using the TC score from HomFam sets.

MSA tool	N° of sequences				Average
	100	200	500	1000	
MBT-Coffee 50%	0.492	0.472	0.499	0.463	0.481
MAFFT L-INS-i	0.429	0.469	0.495	0.486	0.470
MSAProbs	0.477	0.474	0.453	0.452	0.464
ClustalΩ	0.453	0.435	0.451	0.406	0.436
Muscle	0.427	0.401	0.415	0.447	0.422
T-Coffee	0.445	0.401	0.382	0.358	0.397
MAFFT	0.407	0.314	0.349	0.349	0.355
ClustalW	0.401	0.371	0.336	0.310	0.355

4. Discussion

In this paper, we proposed an innovative way to maintain accuracy while reducing the size of the consistency library used in MSA tool protocols. This proposal was implemented and evaluated

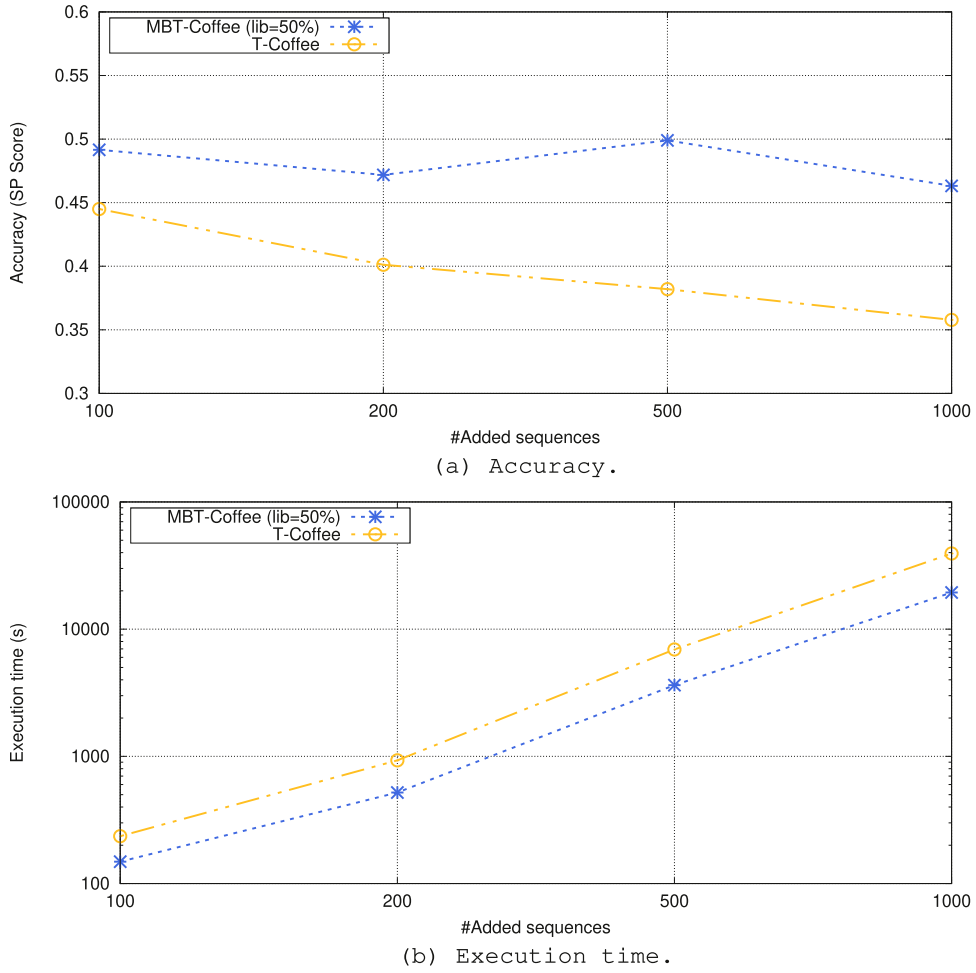

Fig. 7. Comparison of MBT-Coffee and T-Coffee upon adding sequences from HomFam sets.

Table 4
Comparison of the MSA tools using the parsimony score with the Sankoff algorithm along with HomFam sets (Lower is better).

MSA tool	N° of sequences				Average
	100	200	500	1000	
MAFFT L-INS-i	4174	7269	15179	26146	13192
MSAProbs	4185	7320	15302	26422	13307
MAFFT	4161	7310	15473	26896	13460
MBT-Coffee 50%	4224	7473	15690	27336	13681
T-Coffee	4179	7337	15712	28190	13855
ClustalΩ	4298	7509	16049	27966	13956
Muscle	4288	7494	15952	28927	14165
ClustalW	4410	7809	16917	30068	14801

on the T-Coffee MSA tool, a well-known consistency-based aligner. The proposed method, named MBT-Coffee, was based on the MEL library reduction and incorporates a new objective function based on the use of both the consistency library and a substitution matrix to improve the final alignment accuracy. Furthermore, these changes lead to a significant reduction in execution time.

When evaluating BAliBASE, our results demonstrated the effectiveness of the tool by achieving higher accuracy and better execution times when the library reduction is set to $\geq$30%. We hypothesise that a 50% reduction is a good compromise between accuracy and execution time; however, this can be further reduced when required. This improvement was demonstrated on the HomFam benchmark evaluation, where the datasets were larger.

In addition, HomFam was used to determine the scalability of MBT-Coffee when the number of sequences grows. Our results demonstrated significant improvements in accuracy and execution time when compared with other MSA tools from the literature.

This study was restricted to small datasets due to the computational requirements of the aligners that were analysed. Therefore, the HomFam datasets and number of total sequences used were limited in the scalability analysis so that they can be aligned with all the methods used in the comparison. For similar reasons, traditional benchmarks, such as BAliBASE and HomFam were selected to evaluate the accuracy of the aligners. In future work, we want to expand the comparison using other benchmarks, such as QuanTest [27] and ContTest [28], which are more focused on scalability.

We found a reduction of the execution time for MBT-Coffee when compared with the original T-Coffee method. Both methods follow exponential growth trends; however, MBT-Coffee uses a smaller consistency library. This leads to a reduction in the final alignment computational time. This adds flexibility to the application requirements and allows for adjustments in the running time and alignment accuracy.

Finally, it should be noted that the techniques presented in this paper can be adapted for the T-Coffee aligner and other MSA methods based on consistency, such as MAFFT L-INS-i, Probcons, and Probalign, with minor adjustments. However, it is important to consider the impact on the quality of the alignment when reduc-

ing the consistency elements present in the library. Furthermore, the recovery when applying substitution matrices can vary from one method to another.

5. Conclusion

In summary, we present a novel approach to reducing computational time without compromising accuracy in consistency library-based MSA tools, the MBT-Coffee.

In the future, we aim to evaluate the use and impact of other substitution matrices. In addition, we will assess the possibility of creating a new matrix based on the discarded consistency information.

Ethical approval

This work did not need any ethical approval.

Funding

This work was supported by the MINECO-Spain under contracts TIN2017-84553-C2-2-R and PID2020-113614RB-C22.

Data availability

The data underlying the results presented in the study are available from http://www.lbgi.fr/balibase/BalibaseDownload/BAliBASE_R1-5.tar.gz & http://www.clustal.org/omega/homfam-20110613-25.tar.gz.

Declaration of Competing Interest

None of the authors have any conflicts of interest to declare.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.cmpb.2021.106237.

References

[1] A.A. Schäffer, E.L. Hatcher, L. Yankie, L. Shonkwiler, J.R. Brister, I. Karsch-Mizrachi, E.P. Nawrocki, VADR: validation and annotation of virus sequence submissions to GenBank, BMC Bioinf. 21 (211) (2020), doi:10.1186/s12859-020-3537-3.
[2] M. Chatzou, E.W. Floden, P. Di Tommaso, O. Gascuel, C. Notredame, Generalized bootstrap supports for phylogenetic analyses of protein sequences incorporating alignment uncertainty, Syst. Biol. 67 (6) (2018) 997–1009, doi:10.1093/sysbio/syx096.
[3] E.W. Myers, W. Miller, Optimal alignments in linear space, Bioinformatics 4 (1) (1988) 11–17, doi:10.1093/bioinformatics/4.1.11.
[4] S.B. Needleman, C.D. Wunsch, A general method applicable to the search for similarities in the amino acid sequence of two proteins, J. Mol. Biol. 48 (3) (1970) 443–453.
[5] J.D. Thompson, D.G. Higgins, T.J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice, Nucl. Acids Res. 22 (22) (1994) 4673–4680.
[6] F. Sievers, D.G. Higgins, Clustal omega for making accurate alignments of many protein sequences, Protein Sci. 27 (1) (2018) 135–145, doi:10.1002/pro.3290.
[7] R.C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic Acids Res. 32 (5) (2004) 1792–1797.
[8] K. Katoh, D.M. Standley, MAFFT multiple sequence alignment software version 7: improvements in performance and usability, Mol. Biol. Evol. 30 (4) (2013) 772–780.
[9] C.B. Do, M.S. Mahabhashyam, M. Brudno, S. Batzoglou, ProbCons: probabilistic consistency-based multiple sequence alignment, Genome Res. 15 (2) (2005) 330–340.
[10] B. Chowdhury, G. Garai, A bi-objective function optimization approach for multiple sequence alignment using genetic algorithm, Soft Comput. 24 (20) (2020), doi:10.1007/s00500-020-04917-5.
[11] M. Kaya, A. Sarhan, R. Alhajj, Multiple sequence alignment with affine gap by using multi-objective genetic algorithm, Comput. Methods Programs Biomed. 114 (1) (2014) 38–49.
[12] Á. Rubio-Largo, M.A. Vega-Rodríguez, D.L. González-Álvarez, A hybrid multi-objective memetic metaheuristic for multiple sequence alignment, IEEE Trans. Evol. Comput. 20 (4) (2016) 499–514.
[13] J.F. Taly, C. Magis, G. Bussotti, J.M. Chang, P. Di Tommaso, I. Erb, J. Espinosa-Carrasco, C. Kemena, C. Notredame, Using the T-coffee package to build multiple sequence alignments of protein, RNA, DNA sequences and 3D structures, Nat. Protocols 6 (11) (2011) 1669–1682.
[14] C. Notredame, L. Holm, D.G. Higgins, COFFEE: an objective function for multiple sequence alignments., Bioinformatics (Oxford, England) 14 (5) (1998) 407–422.
[15] C.B. Do, M. Brudno, S. Batzoglou, ProbCons: probabilistic consistency-based multiple alignment of amino acid sequences, in: AAAI, 2004, pp. 703–708.
[16] U. Roshan, D.R. Livesay, Probalign: multiple sequence alignment using partition function posterior probabilities, Bioinformatics 22 (22) (2006) 2715–2721.
[17] J. Lladós, F. Cores, F. Guirado, Optimization of consistency-based multiple sequence alignment using big data technologies, J. Supercomput. 75 (3) (2019) 1310–1322.
[18] R.C. Edgar, Optimizing substitution matrix choice and gap parameters for sequence alignment, Bmc Bioinf. 10 (1) (2009) 396.
[19] S. Capella-Gutiérrez, T. Gabaldón, Measuring guide-tree dependency of inferred gaps in progressive aligners, Bioinformatics 29 (8) (2013) 1011–1017.
[20] J. Lladós, F. Guirado, F. Cores, J.L. Lérida, C. Notredame, Recovering accuracy methods for scalable consistency library, J. Supercomput. 71 (5) (2015) 1833–1845, doi:10.1007/s11227-014-1362-z.
[21] J.D. Thompson, P. Koehl, R. Ripp, O. Poch, BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark, Proteins 61 (1) (2005) 127–136.
[22] F. Sievers, D. Dineen, A. Wilm, D.G. Higgins, Making automated multiple alignments of very large numbers of protein sequences, Bioinformatics 29 (8) (2013) 989–995.
[23] R.C. Edgar, qscore, URL http://drive5.com/qscore/qscore_src.tar.gz.
[24] J. Lladós, F. Guirado, F. Cores, Scalable Consistency for large-scale multiple sequence alignments., in: Proceedings of the 14th International Conference on Computational and Mathematical Methods in Science and Engineering, vol. 3, 2014, pp. 840–851.
[25] A. Goëffon, J.-M. Richer, J.-K. Hao, Progressive tree neighborhood applied to the maximum parsimony problem, IEEE/ACM Trans. Comput. Biol.Bioinf. 5 (1) (2008) 136–145.
[26] D. Sankoff, Minimal mutation trees of sequences, SIAM J. Appl. Math. 28 (1) (1975) 35–42.
[27] Q. Le, F. Sievers, D.G. Higgins, Protein multiple sequence alignment benchmarking through secondary structure prediction, Bioinformatics 33 (9) (2017) 1331–1337.
[28] G. Fox, F. Sievers, D.G. Higgins, Using de novo protein structure predictions to measure the quality of very large multiple sequence alignments, Bioinformatics 32 (6) (2015) 814–820.