Journal of Theoretical and Applied Information Technology
10th May 2015. Vol.75. No.1

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

INVESTIGATION STUDY: AN INTENSIVE ANALYSIS FOR MSA LEADING METHODS

MUHANNAD A. ABU-HASHEM, NUR'AINI ABDUL RASHID, ROSNI ABDULLAH, AWSAN A. HASAN AND ATHEER A. ABDULRAZZAQ

School of Computer Sciences, Universiti Sains Malaysia USM
Pulau Pinang, 11800, Malaysia

Email: abu_hashem4@yahoo.com; nuraini@cs.usm.my; rosni@cs.usm.my; athproof@yahoo.com ; aaha10_com070@student.usm.my

ABSTRACT

Multiple sequence alignment (MSA) is a cornerstone process in computational biology and bioinformatics and it is necessary for sequence analysis. Through decades of extensive research to solve the MSA problem, numerous methods have been proposed and developed. Dynamic programming-based methods suffer from the high computational cost in building MSA. Therefore researchers tend to solve MSA problem using heuristic methods as they attempt to reach optimal results in a reasonable time. Progressive alignment method is the most used method for constructing MSA. In this paper we present an extensive analysis for the leading methods and top performing methods in MSA. The leading methods of MSA are discussed individually to highlight the differences in methods and techniques they use beside the progressive alignment. Also, we present a critical analysis for MSA leading methods based on experimental results where BaliBASE database is used as a benchmark. Results show that MSAProbs algorithm has the superior accuracy performance over all leading methods while MUSCLE and Clustal W are the fastest. Although numerous algorithms have been proposed for MSA, producing an efficient MSA with high accuracy remains a huge challenge.

Keywords: *Multiple sequence alignment, MSA leading methods, Progressive alignment, Sequence alignment, Guide tree, and Computational biology.*

1. INTRODUCTION

MSA plays an essential role in identifying sequences and collecting information about them [1]. It is used for several purposes, with varying degrees of importance and motives. Representing and identifying sequence families is the most significant role of MSA. Indirectly, MSA helps in predict the structure and function of sequences by relating them to their closest similar families. It also helps build the phylogenetic tree to represent the evolutionary history of species and study the evolution of molecules [2].

Optimal MSA is considered as an NP-hard problem because the size of the problem increases radically when the number and length of sequences increase [3-9]. On the other hand, to reach optimal results using dynamic programming (DP) is an NP-complete problem [10]. The length and number of sequences are important factors to consider in MSA methods.

Dynamic programming method considers impractical for MSA, as the number of sequences in the alignment file increase, because of the high computational cost. On the other hand, many attempts is done to solve the problem using heuristic method, such as progressive alignment [11, 12], probabilistic and statistics-based alignment [13-19], iterative alignment [20, 21], and alignment based on population-based metaheuristic methods [22-32]. Progressive alignment method is widely used method for building MSA. Recent MSA methods are tend to combine progressive alignment with other heuristic-based methods to achieve optimal results. Also, it is used with many approaches beside the heuristic approach to find the optimal alignment in a reasonable time.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Benchmark databases have encouraging effects toward developing MSA methods. Many benchmark databases are built to facilitate improving the performance of MSA methods by measuring the alignment quality such as BaliBASE [33-36], OXBench [37], IRMBase [38], and SABmark [39].

In this paper we provide an extensive analysis for the top performance and leading methods of MSA. Next we give a definition for MSA Problem, guide tree, and progressive alignment method. After that we provide an extensive review for MSA leading methods.

2. PROBLEM DEFINITION

MSA is a process of aligning more than two DNA, RNA, or protein sequences. It analyzes the relationship between these sequences to determine their shared lineage or common ancestors. The results from MSA can provide information on homologous sequences and can be used to conduct phylogenetic analysis. The process of manually aligning multiple sequences is complex and time consuming. These challenges encourage researchers to contribute by constructing MSA computationally in order to help simplify and speed up the process of constructing MSA.

Say we have 'n' number of sequences S_1, S_2, ..., and S_n. The MSA of S_1, S_2, ..., and S_n is a matrix $M(i,j)$, where i is the number of sequences and j is the number of residues in each sequence.

3. GUIDE TREE

Guide tree is a binary tree that represents the relations among a group of sequences based on distance matrix scores. It is a basic process in progressive alignment methods to guide the alignment process because the sequences are organized in the tree according to their resemblance score. Sequences are branched in the tree according to the amount of similarity they share, that is, the most similar are the closer to each other. Each leaf in the tree represents a different sequence, while nodes show how far these sequences are from each other. Building a guide tree requires (1) a pre-computed distance matrix, which can be built by applying all-to-all pair-wise alignment, and (2) a clustering method to arrange the sequences in a tree according to the distances in the distance matrix.

Guide trees offer a pre-computed map that can be used as a compass to direct the progressive alignment. Progressive alignment starts constructing the alignment by aligning two sequences, and then continues aligning sequences to the previous aligned sequences. This process makes guide trees a great support to enhance the alignment score and reduce the running time.

4. PROGRESSIVE ALIGNMENT

Progressive alignment is a widely used method for building MSA. It manages the alignment by passing the sequences through two main phases: (1) tree construction phase, where the sequences are arranged according to the amount of similarity they share in a tree-like shape called guide tree, and (2) MSA construction phase. The second phase is carried out by aligning the most similar pair of sequences as a core for the alignment, and then adding sequences gradually to the previously aligned sequences depending on the similarity they share using the guide tree as a leading compass.

The alignment is built progressively starting with the closest pair of sequences moving towards aligning the next similar sequences to those already aligned. Progressive alignment handles the alignment by passing the sequences through various steps. The first step is pair-wise alignment, which requires an (n-1) × n/2 pair-wise alignment to align n sequences. Then, the pair-wise alignments are scored to identify the highly matched sequences. After the closest pair of sequences are identified, their gaps get filled using neutral elements. Next, the second closest sequence or group of sequences gets added to the original pair, and then a new alignment is established for them. Finally, after scoring the last alignment, the tree plot is constructed [12].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. MSA LEADING METHODS

MSA leading methods are the top performing algorithms in the MSA research field, as well as the most commonly used methods for comparing and measuring the performance of new algorithms in the field. All MSA leading methods are based on progressive alignment. Progressive alignment method mainly constructs the MSA in three steps: (1) constructing the distance matrix, (2) building the guide tree, and (3) building the MSA progressively.

Although all of the leading methods use progressive alignment to build MSA, they vary by the other techniques they use. To construct the distance matrix, some methods use pair-wise alignment while others use approximate method. Using pair-wise alignment to construct the distance matrix improves the accuracy of the alignment but compromises the execution time. On the other hand, using approximate distance method to construct the distance matrix speeds up the process of building the alignment but compromises the of accuracy. Furthermore, two different clustering methods are used by the leading methods to build the guide tree out of the distance matrix: UPGMA and NJ. UPGMA method is used for its simplicity and speed in constructing the tree, while NJ method is used for its robustness because it does not assume that all species have the same mutation rate. The last step in progressive alignment method is progressively constructing MSA out of the guide tree. Some leading methods applied a refinement process to the final step of progressive alignment in order to improve the alignment score, while others do not in order to avoid its overhead on execution time.

In this section, we describe the leading methods of protein MSA, which include Clustal-W, T-Coffee, MAFFT, MUSCLE, ProbCons, Probalign, DIAlign-TX, and MSAProbs.

5.1 Clustal-W

Clustal-W [40] is a progressive alignment algorithm proposed in 1994 to improve the alignment of divergent protein sequences. Clustal-W is one of the many extensions to the Clustal algorithm [41], along with Clustal V [42, 43] and Clustal X [44]. Alignment improvement using Clustal-W algorithm includes (1) assigning weights to the sequences regarding the shared similarity to mark the sequences that share high similarity from other sequences, (2) using different substitution matrices according to the shared similarity of the target sequences to refine the alignment and maximize the alignment score, (3) adding potential loop region gaps raised by the gap penalties of residue-specific and locally reduced gaps. Clustal-W algorithm is among the fastest algorithms when compared with the leading methods of MSA, but it sacrifices accuracy because it is among the lowest performing algorithm compared with other MSA leading methods.

5.2 T-Coffee

T-Coffee (Tree-based Consistency Objective Function for alignment Evaluation) algorithm [45], is an extension to the Coffee algorithm [46] proposed in 2000 to build MSA. T-Coffee algorithm is proposed to overcome the accuracy weaknesses of the original Coffee algorithm. It improves the accuracy of the Coffee algorithm by pre-processing the datasets and building a library of alignment information. On the other hand, T-Coffee algorithm sacrifices the run time compared with the leading methods.

Two main features of T-Coffee algorithm have contributed the most toward improving the accuracy of the original algorithm (Coffee). The first feature is the simplicity and flexibility of multiple alignment construction. T-Coffee algorithm constructs a library of pair-wise alignments that are generated by multiple systems and are applied to heterogeneous sources. The second feature is the use of optimization method to detect the finest pair-wise alignment from the pre-computed library.

The pre-computed library is built in several steps: (1) generating a primary library of pair-wise alignments by using two different pair-wise alignments [Clustal-W and Lalign[47]] applied to the same pair of sequences, (2) deriving of the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

primary library weights where each aligned pair of residues is assigned a weight, (3) combining the libraries by combining the alignments that are gathered using the two pair-wise methods, and (4) extending the library by assigning weight constraints to the alignment. Afterward, the algorithm builds the guide tree using NJ method out of the distance matrix. Finally, progressive alignment method is used to build the MSA.

5.3 MAFFT

MAFFT algorithm is a progressive method for building MSA proposed by Kazutaka Katoh in 2002 [48]. Katoh then has proposed two versions of the MAFFT algorithm [49] which is an improvement to the accuracy of the original algorithm and [50] uses the MAFFT algorithm to build multiple alignments of DNA sequences. Basically, MAFFT algorithm applies two techniques to build the alignment: (1) fast Fourier transformation (FFT) to identify homologous regions, (2) simplified scoring system to reduce the run time and, at the same time, improve the accuracy. MAFFT algorithm outperforms Clustal-W and MUSCLE algorithms in terms of accuracy, but still has low accuracy compared with the leading method of MSA. Furthermore, it suffers from high execution time compared with the leading methods, especially Clustal-W and MUSCLE algorithms.

Before the refinement process, the MAFFT algorithm generally includes two main stages. The first stage is building the MSA1, which includes all-to-all pair-wise alignment to build the distance matrix1, UPGMA clustering method to build the guide tree1, and progressive alignment to build MSA1 out of guide tree1. The second stage is to improve MSA1 by taking it as an input. This stage builds MSA2 by building the distance matrix2 out of MSA1 depending on the 6-tuples that the target pair of sequences share, building the guide tree2 using UPGMA method, and using progressive alignment method to build MSA2.

5.4 MUSCLE

MUSCLE (MUltiple Sequence Comparison by Log-Expectation) algorithm is a progressive MSA method proposed by Robert C. Edgar in 2004 [51]. Another version of MUSCLE algorithm is proposed in [52] , in which the focus is on improving the run time of the original one. Generally, MUSCLE algorithm consists of three stages: draft progressive to build the MSA, improved progressive to enhance the first MSA draft, and refinement for additional accuracy improvement. Although MUSCLE algorithm is among the fastest methods compared with the leading methods of MSA, many leading methods still outperform it in terms of accuracy.

As a MAFFT algorithm, MUSCLE algorithm builds the MSA twice but has a different refinement technique. MUSCLE algorithm starts with a draft progressive alignment (MSA1), which it then uses MSA1 as an input to build an improved MSA2. Then it starts a refinement stage to enhance the accuracy. Refinement stage begins with splitting tree2 (the one produced in MSA2) into two trees by choosing an edge that starts decreasing from the root. Next, the two sub-trees are aligned using profile alignment and then scored using the SP score function. If the new alignment is better than the previous one, then the new alignment replaces the previous one; otherwise, it is rejected.

5.5 ProbCons

ProbCons (Probabilistic consistency) is a progressive probabilistic method used to build MSA for protein sequences [53]. It proposes an objective function based on probabilistic consistency, which applies to the progressive method to build the alignment. To build the alignment using ProbCons algorithm, the sequences should pass through several processes, including calculating posterior probability matrices, computing the expected accuracies of pair-wise alignments, transforming the probabilistic consistency, calculating the guide tree, achieving progressive alignment, and finally, obtaining iterative refinement. ProbCons algorithm has a competitive accuracy

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

performance, but it still suffers from high computational cost especially with long sequences.

To compute posterior probability matrices, ProbCons algorithm uses HMM model to build the alignment, computes pair-wise alignments, and then maximizes the expected accuracy alignment. While most alignment methods attempt to reach optimal pair-wise alignment by aiming for the best probability alignment using Viterbi algorithm [54], ProbCons attempts to find and maximize the alignment a with high accuracy by following Equation (1):

$$E_{a^*}(accuracy(a,a^*)|x,y) = \frac{1}{min\{|x|,|y|\}} \sum_{x_i \sim y_j \epsilon\, a} P(x_i \sim y_j \epsilon\, a^*|x,y) \qquad 1$$

where, a is the alignment, a* is the targeted alignment, and x, y are the target pair of sequences.

5.6 Probalign

Probalign algorithm [55] is a progressive probabilistic MSA method that combines pair-wise posterior probabilities technique with partition function technique. It is a modified version of the Probcons method. To build the alignment using Probalign, sequences should pass through several steps, including computing posterior probability, combine posterior probabilities with partition function, and maximal expected accuracy alignment. Probalign algorithm has a competitive accuracy performance compared with Probcons algorithm and is faster than Probcons, but still slower than Clustal-W and MUSCLE.

To compute the posterior probability, Probalign algorithm follows the equation described in [53]. Then, the algorithm computes the maximal expected accuracy alignment out of the given posterior probability matrix by using the equation described in [56]. Probalign algorithm next uses the partition function to estimate the posterior probabilities. Afterward it uses the pre-computed posterior probability matrix to build the alignment.

5.7 DIalign-TX

DIalign-TX [57], an extension to the DIalign and DIalign-T algorithms, is an MSA method proposed in 2008. DIalign-TX is a progressive segment-based method that uses greedy approach along with the progressive alignment to build the alignment. The accuracy is improved compared to the original algorithm DIalign without increasing the time and space requirement. Furthermore, DIalign-TX inherits the direct greedy approach sensitivity to spurious pair-wise similarities, which bring the alignment score down while maintaining the average time performance.

DIalign-TX algorithm builds the MSA by passing the sequences through several processes, including pair-wise alignment, guide tree construction, and progressive alignment. To compute the distance matrix, two rounds of pair-wise alignment are applied. Unlike most pair-wise methods, DIalign-TX algorithm uses three inputs to build the alignment: the two sequences to be aligned and a consistent fragment set. To enhance the pair-wise alignment, a second pair-wise alignment round is applied using greedy alignment. The guide tree is built using UPGMA clustering method. Finally, progressive alignment method is used to build the MSA out of the guide tree.

5.8 MSAProbs

MSAProbs is a progressive probabilistic MSA method proposed in 2010 by Yongchao Liu et al [1]. It uses a combination of pair-HMM and partition function to compute the posterior probabilities. To improve the accuracy of the alignment, MSAProbs algorithm uses two techniques: (1) weighted probabilistic consistency transformation and (2) weighted profile–profile alignment. MSAProbs algorithm shows an improved accuracy performance compared with to the leading methods of MSA, but it is outperformed by many methods such as Clustal-W and MUSCLE in terms of time performance.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Building MSA using MSAProbs algorithm can be concluded in six steps: posterior probability calculation, distance matrix building, guide tree construction, weighted probabilistic consistency transformation, progressive alignment, and iterative refinement. The probability matrix is built by combining two probability matrices as a root mean square of the two matrices. The two probability matrices are built based on pair-HMM and partition function.

A comprehensive analysis is shown in Table 1. It provides a summary description of MSA leading methods.

Table 1: Summary of MSA Leading Methods

	Distance matrix method	Pair-wise alignment		Global alignment	Local alignment	Guide Tree construction method		Use progressive	Approach	Refinement
		With	Without			UPGMA	NJ			
Clustal-W	Dynamic programming method, approximate distance method	✓	✓	✓	-	-	✓	✓	Progressive, dynamic programming	-
MAFFT	Needleman-Wunsch algorithm, approximate distance method	✓	✓	✓	-	✓	-	✓	Progressive, Iterative	✓
MUSCLE	Kmer, Kimura distances	-	✓	-	-	✓		✓	Progressive, Iterative	✓
T-Coffee	Clustal-W and Lalign	✓		✓	✓		✓	✓	Progressive, Greedy	-
ProbCons	HMM	-	✓	-	-	Modified UPGMA	-	✓	Progressive, Iterative, probabilistic	✓
Probalign	probabilistic consistency	-	✓	-	-	✓	-	✓	probabilistic,	✓
MSAProbs	pair-HMM and partition function	✓	-	✓	-	✓	-	✓	Progressive, probabilistic, Iterative	✓
DIAlign-TX	DIalign	✓	-	-	✓	✓	-	✓	Progressive, Segment-based, graph theory Greedy, Iterative.	✓

6. RESULTS AND MEASUREMENTS

6.1 Performance Measurements

To measure and compare the results of MSA leading methods, two performance metrics were involved: execution time and accuracy. Score function tools are used as accuracy measurements to calculate the quality of an existing alignment. Two commonly used scoring methods are applied to assess the accuracy of the alignments: sum-of-pairs score (SPS) and column score (CS). SPS can be defined as the total number of aligned pairs of residues in the test alignment (the alignment we want to score) that matches the correspondence pair of residues in the reference alignment divided by the total number of aligned pairs in the reference alignment. To compute the score for an alignment using SPS, two input alignments should be provided: the target alignment (test alignment) and the reference alignment provided by the benchmark. CS is used to evaluate the quality of an alignment by counting the number of correctly aligned columns. It is defined as the total number of aligned columns in the test alignment (the alignment we want to score) that matches the correspondence columns in the reference alignment divided by the total number of aligned columns in the reference alignment. CS method uses the same inputs as SPS method.

To assess MSA leading methods we use Benchmark Alignment dataBASE (BaliBASE) benchmark [33-36]. It is a benchmark specifically intended for MSAs with big scale. It is designed to point out all the potential problems faced in MSA in order to give a better evaluation for a set of aligned sequences. BaliBASE is a manually refined benchmark with huge diversity reference sets of MSAs. These reference sets characterize many challenges in MSAs, such as small number of sequences involved in the alignment, large N/C-terminal extension sequences, and sequences with unequal evolutionary rate. Basically, BaliBase 3.0 dataset consists of 386 alignment groups distributed into

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

six reference sets, namely, RV11 (76 alignment files), RV12 (88 alignment files), RV2 (82 alignment files), RV3 (60 alignment files), RV4 (49 alignment files), and RV5 (31 alignment files).

6.2 Results

Tables 2-5 show the average SPS, CS, and execution time results for MSA leading methods on BaliBASE 3.0 benchmark. Table 2 shows the execution time of MSA leading methods on BaliBASE 3.0 benchmark reference datasets. Table 3 and Table 4 show the SPS and CS scores of MSA leading methods on BaliBASE 3.0 benchmark reference datasets. Finally, Table 5 shows the overall average values of SPS, CS, and execution time on BaliBASE benchmark.

Table 2: Execution Time Results on Balibase [the bold values represent the shortest execution time (in seconds)]

	RV11	RV12	RV2	RV3	RV4	RV5
Clustal-W	**19**	**45**	455	576	**202**	**112**
MAFFT	580	920	6924	6714	5043	3430
MUSCLE	31	46	**262**	**299**	352	117
TCoffee	1755	2491	29221	7423	1230	697
ProbCons	82	269	4623	7482	2749	1368
Probalign	47	200	3355	5101	1968	917
MSAProbs	82	286	5285	8182	2087	1382
DIAlign-TX	50	100	1461	2039	581	307

Table 3: SPS Score On Balibase (The Bold Values Represent The Highest Score)

	RV11	RV12	RV2	RV3	RV4	RV5
Clustal-W	0.5822	0.8840	0.8879	0.7714	0.7894	0.7691
MAFFT	0.6841	0.9356	0.9357	0.8708	0.9119	0.8977
MUSCLE	0.6575	0.9232	0.9151	0.8424	0.8648	0.8529
T-Coffee	0.7297	0.9436	0.9343	0.8711	0.8919	0.9017
ProbCons	0.7400	0.9459	0.9370	0.8754	0.9003	0.9015
Probalign	0.7127	0.9465	0.9354	0.8645	0.9221	0.8912
MSAProbs	**0.7459**	**0.9487**	**0.9436**	**0.8820**	**0.9254**	**0.9090**
DIAlign-TX	0.5401	0.8830	0.8901	0.7684	0.8340	0.8218

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4: CS Score on BaliBASE (the bold values represent the highest score)

	RV11	RV12	RV2	RV3	RV4	RV5
Clustal-W	0.3253	0.7559	0.3386	0.3817	0.3982	0.3650
MAFFT	0.4754	0.8426	0.4891	0.6007	0.5861	0.5904
MUSCLE	0.4332	0.8200	0.4222	0.4768	0.4532	0.4750
T-Coffee	0.5143	0.8585	0.4874	0.5637	0.5424	0.6016
ProbCons	0.5276	0.8682	0.5080	0.6005	0.5361	0.5952
Probalign	0.4857	0.8677	0.4669	0.5972	0.6123	0.5436
MSAProbs	**0.5365**	**0.8746**	**0.5408**	**0.6342**	**0.6251**	**0.6143**
DIAlign-TX	0.3172	0.7600	0.3469	0.3992	0.4517	0.4567

Table 5: Overall Average SPS and CS Scores and Overall Run Time (in seconds) on BaliBASE 3.0

	SPS	CS	Time
Clustal-W	0.7807	0.4274	1409
MAFFT	0.8700	0.6048	23611
MUSCLE	0.8426	0.5134	**1107**
T-Coffee	0.8787	0.5947	42817
ProbCons	0.8833	0.6059	16573
Probalign	0.8787	0.5956	11588
MSAProbs	**0.8924**	**0.6376**	17304
DIAlign-TX	0.7896	0.4553	4538

Table 2 shows that Clustal-W and MUSCLE have the shortest execution time among all MSA leading methods. Although Clustal-W and MUSCLE algorithms are superior to all leading methods in terms of time performance, their accuracy results are not competitive compared to the rest of MSA leading methods. On the other hand, T-Coffee and MAFFT algorithms have the longest execution time compared to MSA leading methods.

Even though MSAProbs algorithm does not compete with the other MSA leading methods in terms of time performance as it is among the longest execution time algorithms, but as shown in Tables 3-5 it has the highest accuracy among all MSA leading methods. The overall results shown in Table 5 show the superiority of MSAProbs algorithm over all MSA methods in terms of accuracy while MUSCLE algorithm is the fastest among all methods. Table 6 below presents a critical analysis of MSA leading methods.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 6: Description of MSA Leading Methods

Method	Description
Clustal-W, MUSCLE	Two of the fastest algorithms on most datasets among the leading methods in terms of execution time, but they sacrifice their accuracies, which are among the lowest performing algorithms.
T-Coffee	Improves the accuracy of Coffee algorithm but sacrifices the run time.
MAFFT	More accurate than the Clustal-W and MUSCLE algorithms but still has low accuracy compared with the leading method, and its execution time has a high computational.
ProbCons	Has competitive accuracy performance but still has high computational cost.
Probalign	An extension to the ProbCons algorithm, it has competitive accuracy performance compared with ProbCons algorithm. It is faster than ProbCons but still slower than Clustal-W and MUSCLE.
DIAlign-TX	Inherits the direct greedy approach sensitivity of spurious pair-wise similarities, which bring the alignment score down while maintaining the average time performance.
MSAProbs	Improved accuracy performance over the leading methods since it outperforms them on most of datasets, but is slower than Clustal-W and MUSCLE in terms of time performance.

Despite the variety of proposed methods and the huge amount of algorithms that have been proposed to solve the MSA problem, building an efficient and accurate MSA Remains a challenge. Furthermore, building MSA using progressive alignment is time consuming, especially when it is applied on large datasets where it can last for hours. Faster algorithms are needed because sequence databases are growing very fast.

7. CONCLUSION

In this paper we present an extensive review for MSA leading methods. Even though all leading methods applied progressive alignment, they use different methods to build the guide tree. Also they use different techniques to perform the refinement process. The results on BaliBASE benchmark show that the fastest algorithms of MSA leading methods are among the lowest accuracy performance, and the highest accuracy algorithms are among the lowest time performance. Clustal-W and MUSCLE take the lead in time performance while MSAProbs gains the highest accuracy among all leading methods. Although MSA leading methods show improved accuracy and a tremendous development in building MSA, an optimal MSA remains a big challenge. Furthermore, maintaining high accuracy without sacrificing time performance is in need.

ACKNOWLEDGMENTS

This research is supported by the UNIVERSITI SAINS MALAYSIA and has been funded by the Research University grant titled by "A GPU based high throughput multiple sequence alignment algorithm for protein data" [1001/PKOMP/817065].

REFERENCES

[1] Y. Liu, et al., "MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities," Bioinformatics, vol. 26, pp. 1958-1964, 2010.
[2] W. Pirovano and J. Heringa, "Multiple Sequence Alignment," in Bioinformatics.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

vol. 452, J. Keith, Ed., ed: Humana Press, 2008, pp. 143-161.
[3] J.-M. Richer, et al., "A new dynamic programming algorithm for multiple sequence alignment," in Combinatorial Optimization and Applications, ed: Springer, 2007, pp. 52-61.
[4] L. Wang and T. Jiang, "On the complexity of multiple sequence alignment," Journal of computational biology, vol. 1, pp. 337-348, 1994.
[5] H. T. Rausch, "Dissecting Multiple Sequence Alignment Methods," 2010.
[6] J. Ebedes and A. Datta, "Multiple sequence alignment in parallel on a workstation cluster," Bioinformatics, vol. 20, pp. 1193-1195, 2004.
[7] R. Thomsen, et al., "A clustal alignment improver using evolutionary algorithms," presented at the Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on, 2002.
[8] E. Corel, et al., "A min-cut algorithm for the consistency problem in multiple sequence alignment," Bioinformatics, vol. 26, pp. 1015-1021, 2010.
[9] I. Wohlers, et al., "Algorithm engineering for optimal alignment of protein structure distance matrices," Optimization Letters, vol. 5, pp. 421-433, 2011.
[10] K. R. Sharma, Bioinformatics: Sequence alignment and markov models: McGraw-Hill, 2009.
[11] P. Hogeweg and B. Hesper, "The alignment of sets of sequences and the construction of phyletic trees: an integrated method," Journal of molecular evolution, vol. 20, pp. 175-186, 1984.
[12] D. F. Feng and R. F. Doolittle, "Progressive sequence alignment as a prerequisitetto correct phylogenetic trees," Journal of molecular evolution, vol. 25, pp. 351-360, 1987.
[13] J. S. Liu, et al., "Bayesian models for multiple local sequence alignment and Gibbs sampling strategies," Journal of the American Statistical Association, vol. 90, pp. 1156-1170, 1995.
[14] T. K. Rasmussen and T. Krink, "Improved Hidden Markov Model training for multiple sequence alignment by a particle swarm optimization—evolutionary algorithm hybrid," Biosystems, vol. 72, pp. 5-17, 2003.
[15] J. Pei and N. V. Grishin, "MUMMALS: multiple sequence alignment improved by using hidden Markov models with local structural information," Nucleic Acids Research, vol. 34, pp. 4364-4374, 2006.
[16] R. K. Bradley, et al., "Fast statistical alignment," PLoS computational biology, vol. 5, p. e1000392, 2009.
[17] X. Deng and J. Cheng, "MSACompro: protein multiple sequence alignment using predicted secondary structure, solvent accessibility, and residue-residue contacts," BMC bioinformatics, vol. 12, p. 472, 2011.
[18] K. S. M. T. Hossain, et al., "Improved Multiple Sequence Alignments Using Coupled Pattern Mining," Computational Biology and Bioinformatics, IEEE/ACM Transactions on, vol. 10, pp. 1098-1112, 2013.
[19] J. Sun, et al., "Multiple Sequence Alignment with Hidden Markov Models Learned by Random Drift Particle Swarm Optimization," Computational Biology and Bioinformatics, IEEE/ACM Transactions on, vol. PP, pp. 1-1, 2013.
[20] M. Berger and P. J. Munson, "A novel randomized iterative strategy for aligning multiple protein sequences," Computer applications in the biosciences: CABIOS, vol. 7, pp. 479-484, 1991.
[21] G. J. Barton and M. J. Sternberg, "A strategy for the rapid multiple alignment of protein sequences: Confidence levels from tertiary structure comparisons," Journal of molecular biology, vol. 198, pp. 327-337, 1987.
[22] J. Kim, et al., "Multiple Sequence Alignment Using Simulated Annealing," Computer Applications in the Biosciences, vol. 10, pp. 419-426, Jul 1994.
[23] C. Notredame and D. G. Higgins, "SAGA: Sequence alignment by genetic algorithm," Nucleic acids research, vol. 24, pp. 1515-1524, Apr 15 1996.
[24] J. Moss and C. G. Johnson, "An ant colony algorithm for multiple sequence alignment in bioinformatics," Artificial Neural Nets and Genetic Algorithms, Proceedings, pp. 182-186, 2003.
[25] T. Riaz, et al., "A TABU SEARCH ALGORITHM FOR POST-PROCESSING MULTIPLE SEQUENCE ALIGNMENT," Journal of Bioinformatics & Computational Biology, vol. 3, pp. 145-156, 2005.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[26] C. A. Lightner, "A Tabu Search Approach to Multiple Sequence Alignment," Ph.D, 2008.
[27] L. Hai-Xia, et al., "Multiple Sequence Alignment Based on a Binary Particle Swarm Optimization Algorithm," presented at the Proceedings of the 2009 Fifth International Conference on Natural Computation - Volume 03, 2009.
[28] X. J. Lei, et al., "Multiple Sequence Alignment Based on Chaotic PSO," Computational Intelligence and Intelligent Systems, vol. 51, pp. 351-360, 2009.
[29] F. J. M. da Silva, et al., "AlineaGA—a genetic algorithm with local search optimization for multiple sequence alignment," Applied Intelligence, vol. 32, pp. 164-172, 2010.
[30] Q. Bo and W. Zhaozhi, "An efficient way of multiple sequence alignment," presented at the Software Engineering and Service Science (ICSESS), 2011 IEEE 2nd International Conference on, 2011.
[31] F. Naznin, et al., "Progressive Alignment Method Using Genetic Algorithm for Multiple Sequence Alignment," Evolutionary Computation, IEEE Transactions on, vol. 16, pp. 615-631, 2012.
[32] F. Ortuno, et al., "Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II," presented at the Evolutionary Computation (CEC), 2012 IEEE Congress on, 2012.
[33] J. D. Thompson, et al., "BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark," Proteins: Structure, Function, and Bioinformatics, vol. 61, pp. 127-136, 2005.
[34] A. Bahr, et al., "BAliBASE (Benchmark Alignment dataBASE): enhancements for repeats, transmembrane sequences and circular permutations," Nucleic Acids Research, vol. 29, pp. 323-326, 2001.
[35] J. D. Thompson, et al., "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs," Bioinformatics, vol. 15, pp. 87-88, 1999.
[36] E. Perrodou, et al., "A new protein linear motif benchmark for multiple sequence alignment software," BMC bioinformatics, vol. 9, p. 213, 2008.
[37] G. Raghava, et al., "OXBench: a benchmark for evaluation of protein multiple sequence alignment accuracy," BMC bioinformatics, vol. 4, p. 47, 2003.
[38] A. Subramanian, et al., "DIALIGN-T: an improved algorithm for segment-based multiple sequence alignment," BMC Bioinformatics, vol. 6, p. 66, 2005.
[39] I. Van Walle, et al., "SABmark—a benchmark for sequence alignment that covers the entire known fold space," Bioinformatics, vol. 21, pp. 1267-1268, 2005.
[40] J. Thompson, et al., "CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," Nucleic Acids Research, vol. 22, pp. 4673-4680, 1994.
[41] D. G. Higgins and P. M. Sharp, "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer," Gene, vol. 73, pp. 237-244, 1988.
[42] D. G. Higgins, et al., "CLUSTAL V: improved software for multiple sequence alignment," Computer applications in the biosciences: CABIOS, vol. 8, pp. 189-191, 1992.
[43] D. G. Higgins, "CLUSTAL V: multiple alignment of DNA and protein sequences," in Computer analysis of sequence data, ed: Springer, 1994, pp. 307-318.
[44] J. D. Thompson, et al., "The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools," Nucleic Acids Research, vol. 25, pp. 4876-4882, 1997.
[45] C. Notredame, "T-coffee: a novel method for fast and accurate multiple sequence alignment," Journal of Molecular Biology, vol. 302, pp. 205-217, 2000.
[46] C. Notredame, et al., "COFFEE: an objective function for multiple sequence alignments," Bioinformatics, vol. 14, pp. 407-422, 1998.
[47] X. Huang and W. Miller, "A time-efficient, linear-space local similarity algorithm," Advances in Applied Mathematics, vol. 12, pp. 337-357, 1991.
[48] K. Katoh, et al., "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," Nucleic

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Acids Research, vol. 30, pp. 3059-3066, 2002.
[49] K. Katoh, et al., "Improvement in the accuracy of multiple sequence alignment program MAFFT," GENOME INFORMATICS SERIES, vol. 16, p. 22, 2005.
[50] K. Katoh, et al., "Multiple alignment of DNA sequences with MAFFT," Bioinformatics for DNA Sequence Analysis, vol. 537, pp. 39-64, 2009.
[51] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," Nucleic Acids Research, vol. 32, pp. 1792-1797, 2004.
[52] R. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," BMC Bioinformatics, vol. 5, 2004.
[53] C. Do, et al., "ProbCons: Probabilistic consistency-based multiple sequence alignment," Genome Research, vol. 15, pp. 330-340, 2005.
[54] A. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," Information Theory, IEEE Transactions on, vol. 13, pp. 260-269, 1967.
[55] U. Roshan and D. R. Livesay, "Probalign: multiple sequence alignment using partition function posterior probabilities," Bioinformatics, vol. 22, pp. 2715-2721, November 15, 2006 2006.
[56] R. Durbin, Biological sequence analysis: probabilistic models of proteins and nucleic acids: Cambridge university press, 1998.
[57] A. R. Subramanian, et al., "DIALIGN-TX: greedy and progressive approaches for segment-based multiple sequence alignment," Algorithms for molecular biology : AMB, vol. 3, p. 6, 2008.