https://doi.org/10.31449/inf.v43i4.2621

Refin-Align: New Refinement Algorithm for Multiple Sequence Alignment

Ahmed Mokaddem, Amine Bel Hadj and Mourad Elloumi
Laboratory of Technologies of Information and Communication, and Electrical Engineering, University of Tunis, Tunisia
E-mail:moka.ahmed@yahoo.fr, amine_bel_hadj_90@yahoo.com, Mourad.Elloumi@gmail.com

Keywords: multiple sequence alignment, algorithms, refinement, block

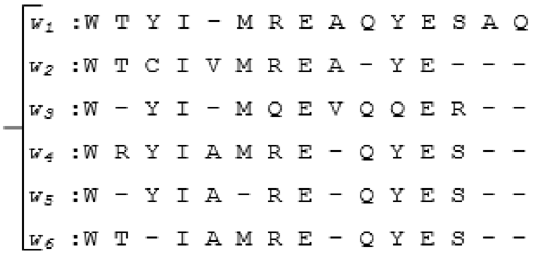
Received: December 17, 2018

In this paper, we present Refin-Align a new refinement algorithm for a multiple sequence alignment. Refining alignment consists on constructing a new more accurate multiple sequence alignment from an initial one by applying some modifications. Our refinement algorithm Refin-Align uses a new definition of block and also our multiple sequence alignment algorithm Pro-malign. We assess our algorithm Refin-Align on multiple sequence alignment constructed by different algorithms using different benchmarks of protein sequences. In the most cases treated, our algorithm improves the scores of the multiple sequence alignment.

Povzetek: Različni znani algoritmi napovedujejo zaporedje beljakovin, novo razviti algoritem pa določa najboljšo skupno vrednost na osnovi napovedi posameznih algoritmov.

1 Introduction

Multiple sequence alignment is an important task in bioinformatics. Aligning a set of sequences consists in optimizing the number of matches between the characters occurring in the same order in each sequence (figure1).



```
w1 :W T Y I - M R E A Q Y E S A Q
w2 :W T C I V M R E A - Y E - - -
w3 :W - Y I - M Q E V Q Q E R - -
w4 :W R Y I A M R E - Q Y E S - -
w5 :W - Y I A - R E - Q Y E S - -
w6 :W T - I A M R E - Q Y E S - -
```

Figure 1: Multiple sequence alignment.

Multiple sequence alignment can help biologist to predict structure and function information for a set of sequences. Indeed, we can reveal information about biological functions common to biological macromolecules from several different organisms by identifying similar regions, these regions are often an important structural or functional roles. Multiple sequence alignment can also help in the classification of macromolecules into different families according to similar sub-strings detected. In addition, multiple sequence alignment can help to construct a phylogenetic tree and analyse relationships between species in order to establish a common biological ancestor.

Although pairwise sequence alignment for two sequences can be constructed with optimal solution using the dynamic programming algorithm [1], multiple sequence alignment for more than two sequences is a NP-complete problem [2]. There are two main approaches to resolve this problem:

1. Progressive approach: it consist to align sequences gradually. Indeed, we start by aligning the most similar two sequences. Then, we align the sequences to other sequences aligned, according to a defined order. Finally, we obtain the multiple sequence alignment. All progressive multiple sequence alignment algorithms adopt the same process. The most used progressive multiple sequence algorithms are ClustalW [3], T-COFFEE [4], MUSCLE [5], MAFFT [6], GLProbs [7] and Clustal Omega [8].

   Progressive approach operates in three steps:

   (a) In the first step, we compute distances between all pairs of sequences of the set and we store these distances in a matrix called *distance matrix*. This step aims to estimate the similarity between pairs of sequences in order to distinguish the two sequences that are the first to be aligned. Many distances are used [9]. Among these distances we mention:

      - *k-mer distances* used by the algorithm MUSCLE and MAFFT,
      - *Percent of similarity* used by the algorithm ClustalW,
      - *Kimura distance* [10] used by the algorithm Clustal Omega,
      - Distance defined by the GLProbs algorithm.

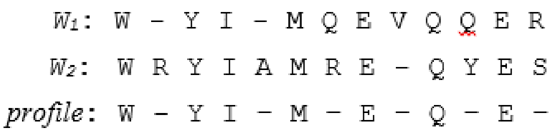
   (b) In the second step, we construct a guide tree using the distance Matrix. This step aims to define the order of aligning sequences. Two main algorithms are used to construct a guide tree:

      - UPGMA [11] used by MUSCLE, MAFFT and GLProbs

- Neighbor-joining [12] used by ClustalW and T-COFFEE

(c) In the last step, we follow the branching order of the guide tree, constructed in the previous step, to construct the multiple sequence alignment by aligning pair of sequences using the dynamic programming algorithm[1] or by *a profile-profile*[3] alignment.

A *profile* is constructed by selecting for each column of the sequence alignment the character that have the maximum occurrences in that column (Figure 2).



```
W1:      W - Y I - M Q E V Q Q E R
W2:      W R Y I A M R E - Q Y E S
profile: W - Y I - M - E - Q - E -
```

Figure 2: Profile construction.

2. Iterative approach: it consists to construct an initial multiple sequence alignment. Then, we apply a number of iterations, during each iteration we perform a set of modifications to the current alignment in order to ameliorate his score. Among this modifications, we can insert or delete of one or more gaps '-' in one or more position in the multiple of sequence alignment. The main multiple sequence alignment algorithms adopting iterative approach are genetic algorithm such as GAPAM [13] and PASA [14].

Each algorithm adopting progressive approach or iterative approach produces mistakes in multiple sequence alignment, thus, we used refinement algorithms in order to correct bad aligned residues, that can ameliorate the quality of the multiple alignment by ameliorate his scores. The process of all refinement algorithms consists to apply a set of modifications to an initial multiple sequence alignment in order to construct a new one having better scores than the previous alignment. These modifications are repeated until *convergence* (i.e. no improvement can be made on the current alignment). There are different algorithms for refinement of multiple sequence alignments:

1. RASCAL [15]: Rascal operates as follows: First, we analyse the initial multiple sequence alignment and detect the well-aligned regions by applying the Mean Distance (MD). Then, we detect the badly aligned regions. Finally, we realign the badly aligned regions.

2. REFINER [16]: when applying REFINER algorithm on a multiple sequence alignment, we realign each sequence with the profile of the multiple sequence alignment of the remaining sequences. Convergence is obtained when all the iterations is realised and each sequence is realigned.

3. RF [17]: is similar to the REFINER algorithm but the convergence is obtained when the number of iterations is equal to 2N2 where N is the number of sequences.

4. REFORMALIGN [18]: Using REFORMALIGN, we construct the final alignment indirectly. First, we start by constructing a profile to the initial multiple sequence alignment. Then, we align each sequence to the profile constructed in the first step. Finally, we merge all the sequences alignment in order to obtain the final alignment.

Thus, Refinement algorithms are used in order to enhance a multiple sequence alignment (MSA). Indeed, we start by an initial multiple sequence alignment by using one multiple sequence alignment algorithm. Then, we apply the refinement algorithm to the initial multiple sequence alignment in order to construct a new more accurate multiple sequence alignment having higher score.

2 Block definition

We propose a new algorithm called *Refin-Align* for refining multiple sequence alignment. *Refin-Align* uses a new definition of block. Indeed, a block is defined as a multiple alignment of substrings extracted from a multiple sequence alignment. A block is formed of at least two adjacent columns separated from the initial alignment on both sides by a column formed only of identical characters. Our new definition of blocks is different from the standard definition of blocks, which presents the blocks as substrings delimited by columns containing at least one gap. The blocks are extracted from the initial multiple alignment and then they will be realigned to improve their scores. A block is defined as follow:

- A set of aligned substrings
- having the same size in each sequence
- A block must contain at least two columns
- No substrings formed the block must be formed entirely of gap
- A block must not contains a column having exactly the same character.

3 Refin-Align: New refinement algorithm

The principle of our algorithm is to extract a misaligned blocks from the sequences that distort the multiple alignment and realign them. The *Refin-Align* algorithm allows improving the quality of an initial multiple alignment by iteratively realigning the blocks of the initial multiple alignment. The advantage of our new block definition is to allow

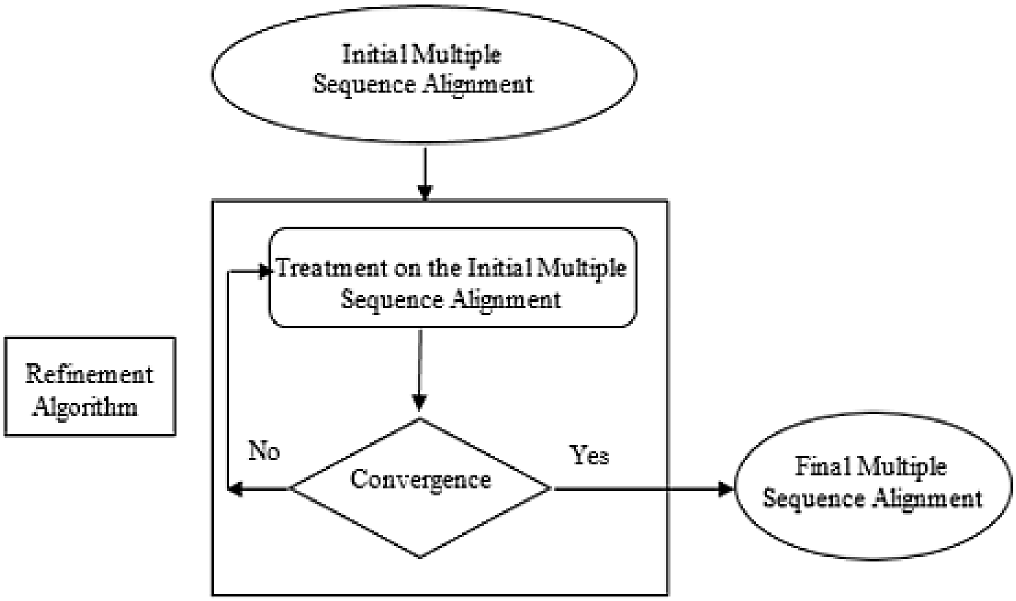
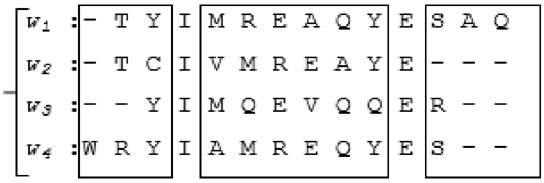

Initial Multiple Sequence Alignment
Treatment on the Initial Multiple Sequence Alignment
Refinement Algorithm
No
Convergence
Yes
Final Multiple Sequence Alignment

Figure 3: Refinement process.



```
w1 :- T Y I M R E A Q Y E S A Q
w2 :- T C I V M R E A Y E - - -
w3 :- - Y I M Q E V Q Q E R - -
w4 :W R Y I A M R E Q Y E S - -
```

Figure 4: Block extraction.

more possibility for the characters of the initial alignment to be realigned. Our *Refin-Align* algorithm operates as follows:

1. First, we extract the blocks from the initial multiple sequence alignment.

2. Then, we compute the scores of each block. We use the sum of pairs score SP[19]. The SP score correspond to the sum of the scores for all pairs of aligned characters. SP score is computed using this formula:

$$SP(A) = \sum_{i=1}^{L} \sum_{1<k<j<l} s(w_k[i], w_j[i]) \qquad (1)$$

   Where $w_k[i]$ and $w_j[i]$ are the characters in the sequences *k* and *j* that are in the ith column of the alignment *A*,*L* is the length of the alignment *A* and *s* is the score of aligning a pair of characters.

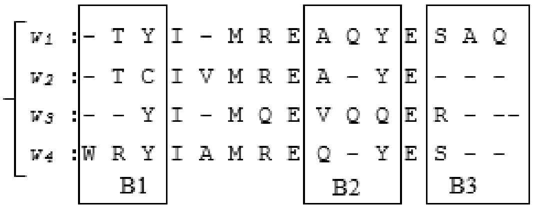
3. Then, we delete gap character from each block and we apply a multiple sequence alignment algorithm Promalign[20] to align these set of new sequences.

4. Finally, we compute the new SP scores. In the case where the scores of the new multiple alignment of blocks are higher than the previous scores, the initial alignment is replaced by the new alignment of blocks obtained.

We repeat this same process, by identifying the new blocks, until we can no longer improve the SP score of each block. The same process is applied for all the blocks of the multiple alignment.

4 Illustrative example

Let be A a multiple sequence alignment of a set of 4 sequences. From this alignment, we extract the blocks B1, B2, B3.

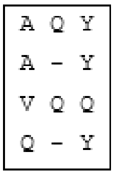


```
w1 :- T Y I - M R E A Q Y E S A Q
w2 :- T C I V M R E A - Y E - - -
w3 :- - Y I - M Q E V Q Q E R - --
w4 :W R Y I A M R E Q - Y E S - -
        B1               B2      B3
```

Figure 5: Alignment A contains three blocks B1, B2, B3.

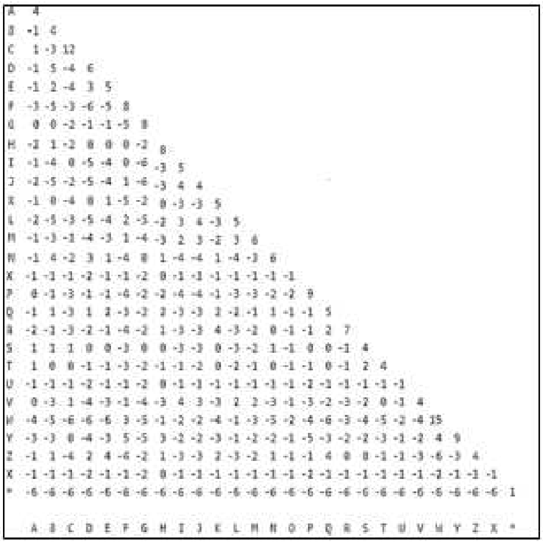
Alignment A contains three blocks B1, B2, B3, we will present the treatment of the second block B2. The same process is repeated for all the blocks.

We compute the SPb, i.e., the SP score of the block B2 before alignment, using the VTML200 Matrix [21]. (* in



```
A Q Y
A - Y
V Q Q
Q - Y
```

Figure 6: Block B2.



A	4																										
B	-1	4																									
C	1	-3	12																								
D	-1	5	-4	6																							
E	-1	2	-4	3	5																						
F	-3	-5	-3	-6	-5	8																					
G	0	0	-2	-1	-1	-5	8																				
H	-2	1	-2	0	0	0	-2	8																			
I	-1	-4	0	-5	-4	0	-6	-3	5																		
J	-2	-5	-2	-5	-4	1	-6	-3	4	4																	
K	-1	0	-4	0	1	-5	-2	0	-3	-3	5																
L	-2	-5	-3	-5	-4	2	-5	-2	3	4	-3	5															
M	-1	-3	-1	-4	-3	1	-4	-3	2	3	-2	3	6														
N	-1	4	-2	3	1	-4	0	1	-4	-4	1	-4	-3	6													
O	-1	-1	-1	-2	-1	-1	-2	0	-1	-1	-1	-1	-1	-1	-1												
P	0	-1	-3	-1	-1	-4	-2	-2	-4	-4	-1	-3	-3	-2	-2	9											
Q	-1	1	-3	1	2	-3	-2	2	-3	-3	2	-2	-1	1	-1	-1	5										
R	-2	-1	-3	-2	-1	-4	-2	1	-3	-3	4	-3	-2	0	-1	-1	2	7									
S	1	1	1	0	0	-3	0	0	-3	-3	0	-3	-2	1	-1	0	0	-1	4								
T	1	0	0	-1	-1	-3	-2	-1	-1	-2	0	-2	-1	0	-1	-1	0	-1	2	4							
U	-1	-1	-1	-2	-1	-1	-2	0	-1	-1	-1	-1	-1	-1	-1	-2	-1	-1	-1	-1	-1						
V	0	-3	1	-4	-3	-1	-4	-3	4	3	-3	2	2	-3	-1	-3	-2	-3	-2	0	-1	4					
W	-4	-5	-6	-6	-6	3	-5	-1	-2	-2	-4	-1	-3	-5	-2	-4	-6	-3	-4	-5	-2	-4	15				
Y	-3	-3	0	-4	-3	5	-5	3	-2	-2	-3	-1	-2	-2	-1	-5	-3	-2	-2	-3	-1	-2	4	9			
Z	-1	1	-4	2	4	-6	-2	1	-3	-3	2	-3	-2	1	-1	-1	4	0	0	-1	-1	-3	-6	-3	4		
X	-1	-1	-1	-2	-1	-1	-2	0	-1	-1	-1	-1	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-2	-1	-1	-1	
*	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	-6	1
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	Y	Z	X	*

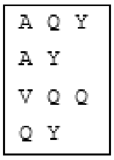
Figure 7: Matrix VTML200.

the Matrix represent the gap '-' character)

SPb:= s(A,A)+2*s(A, V) + 2*s(A, Q) + s(V, Q)+ 4* s(Q, -) + s(Q, Q) + s(-, -) + 3*s(Y,Y)+ 3* s(Y,Q).

SPb:= 0.

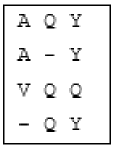
Then, we delete gap '-' characters.



```
A Q Y
A Y
V Q Q
Q Y
```

Figure 8: Block B2 without gap.

Then, we realign the block B2, we obtain the following new block.



```
A Q Y
A - Y
V Q Q
- Q Y
```

Figure 9: Realign the block B2.

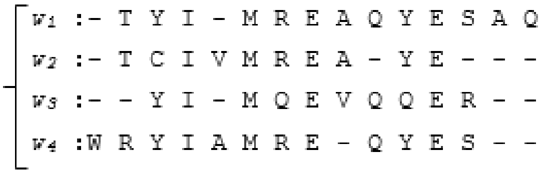
We compute the SPa score. The SP score of the block B2 after alignment.

SPa:= s(A,A)+2*s(A, V) + 2*s(A, -) + s(V, -) + 3*s(Q, -) +3*s(Q, Q) + 3*s(Y,Y) + 3* s(Y,Q)

SPa:= 1.

The score SPa after alignment is higher than the score SPb before alignment. Thus, we replace the old block B2 by the new block B2 in the initial multiple sequence alignment.

We obtain the following new multiple sequence alignment.



```
W1 :- T Y I - M R E A Q Y E S A Q
W2 :- T C I V M R E A - Y E - - -
W3 :- - Y I - M Q E V Q Q E R - -
W4 :W R Y I A M R E - Q Y E S - -
```

Figure 10: Multiple sequence alignment after refinement.

5 Experimental study

In this section, we present the experimental study realized in order to evaluate the performances of our algorithm. In this experimental study, we use the datasets extracted from several benchmarks. These benchmarks maintain reference multiple sequence alignments constructed in manually or automatically. Moreover these benchmarks continent the scores that allow to compare between the reference multiple sequence alignment in the benchmarks and the test multiple sequence alignment. We used the following scores to compare between the refined multiple sequence alignment obtained using our *Refin-Align* algorithm and the reference multiple sequence alignment in the benchmark.

- (Column Score (CS) [22] is the ratio between the number of correctly aligned columns and the number of all columns whose alignments are known.

  $$CS = 1/L * \sum_{i=1}^{L} C_i \qquad (2)$$

  $C_i = 1$ if all the character of the ith column of the test alignment well aligned in the reference alignment in the benchmark else $C_i = 0$. L the number of column where their alignment are known.

- (Sum of Pairs Score (SPS) [22] is the ratio between the number of correctly aligned pairs of character and the total number of all pairs of character whose alignments are known.

$$SPS = \frac{\sum_{i=1}^{c_t} P_i}{\sum_{r=1}^{c_r} P_r} \quad (3)$$

P_i is the number of pairs of character well aligned in the column i, C_t is the number of column in the alignment test, P_r the total number of all pairs of character whose alignments are known. C_r the number of column in the reference alignment.

We used the Qscore program [5] to compute different scores of the different multiple sequences alignments. Each benchmark uses one notation of the same scores for example BALIBASE uses SPS and CS scores however PREFAB uses respectively Q and TC scores. In our experimental study we used Q and TC scores notations. The datasets used in our experimental study are extracted from the following benchmarks for protein sequences:

1. BALIBASE [23]: This benchmark is the first benchmark dedicated to protein multiple alignment algorithms and contains a number of accurate reference alignments grouped in different references according to the nature of the set of the sequences used. The alignments are constructed based on the superposition of proteins tertiary structures and manual improvement of the results. BALIBASE in the first version contain 5 references, in the last version BALIBASE other references are included. The references are:

   - Reference 1 contains short sequences with different sizes,
   - Reference 2 is composed of sequence families aligned with one, two or three orphan sequences,
   - Reference 3 is composed by groups of sequences having 25% of identity by groups,
   - Reference 4 and 5 are composed by extensions and insertions in the sequences,
   - Reference 6, 7 and 8 are composed by repeat and circular permutation in the sequences.
   - Reference 9 contains motifs in all the sequences.

   BALIBASE uses the CS and SPS.

2. PREFAB [5]: This benchmark is made up of 1932 multiple alignments constructed automatically in the following way: The tertiary structures of two sequences are aligned by using two different superposition methods. A set of 50 homologous sequences is then extracted from databases and a multiple alignment is constructed for the whole set of sequences. PREFAB uses only the Q score that is similar to the SPS score of BALIBASE because the comparison is realized between two aligned sequences, extracted from the reference multiple sequence alignment, and the pairwise alignment of the same sequences extracted from the test multiple sequence alignment

3. OXBENCH [24]: This benchmark is constructed in an automatic way, by aligning known tertiary structures extracted from the Protein Data Bank (PDB) using the AMPS method [25]. OXBENCH uses the Q score and the TC score.

4. HOMSTRAD [26]: It contains 1032 multiple sequences alignments of protein sequences representing different structures and grouped in homologous families.

We assess our program *Refin-Align* using the following methods:

- First, we construct for every dataset an initial set of multiple sequences alignments using the following programs: Clustal Omega, MUSCLE, and MAFFT.
- Then, we compute the column score (CS) [22] and the sum of pairs scores (SPS) [22] before refinement for every multiple sequence alignment in the set.
- Then, we apply our algorithm *Refin-Align* to each multiple sequence alignment in order to obtain the refinement multiple sequence alignment. After that, we compute the column scores (CS) and the sum of pairs scores (SPS) for each multiple sequences alignment after refinement.
- Finally, we compare between the scores obtained before applying our refinement algorithm and those obtained after applying our refinement algorithm.

Scores	Q-scores		T-scores	
	Before	After	Before	After
MAFFT	73,30	**73,42**	52,48	**52,85**
MUSCLE	73,04	**73,82**	54,01	**54,63**
Clustal Omega	70,06	**71,36**	46,70	**47,14**

Table 1: Scores obtained using HOMSTRAD Benchmark

The results of the Program MUSCLE, MAFFT and Clustal Omega are respectively obtained using the program MUSCLE, the online web server of MAFFT and the online web server of Clustal Omega.

These tables below represent the SPS and CS scores obtained.

Table 1 represents the Q-scores and the TC scores obtained before refinement and the scores after refinement on a set of multiple alignment sequence extracted from HOMSTRAD Benchmark.

We benchmarked also our program *Refin-Align* on a set of datasets extracted from OXBENCH Benchmark. Table 2 shows the average of the TC scores and the Q-scores obtained.

We benchmarked also our program *Refin-Align* on several datasets extracted from PREFAB Benchmark.

Scores	Q-scores		T-scores	
	Before	After	Before	After
MAFFT	79,63	**81,63**	70,20	**71,60**
MUSCLE	80,45	**81,74**	70,21	**70,89**
Clustal Omega	84,37	**84,42**	**67,25**	67,04

Table 2: Scores obtained using OXBENCH Benchmark

The comparison of alignments for the PREFAB and the scores computing is different from other benchmarks; this is due to the method of creating this benchmark. Indeed, the reference alignments is a set of pairwise alignment extracted from the multiple sequence alignment. Thus, the Q-scores are computed between the reference pairwise alignment and the test pairwise alignment of the same sequences. In this case, the Q and TC scores are identical.

Table 3 shows the average of Q-scores obtained by applying our refinement *Refin-Align* algorithm on a set of multiple sequence alignment of datasets extracted from PREFAB.

Q-scores	Before	After
MAFFT	**62,07**	62,07
MUSCLE	65,06	**66,04**
Clustal Omega	**64,60**	64,19

Table 3: Scores obtained using PREFAB Benchmark

We also benchmarked our algorithm *Refin-Align* on all the 44 datasets of RV12 reference of BALIBASE and we compute the Q-scores and the TC scores. RV12 represents the reference 1 of the BALIBASE benchmark that contain sequences having between 20% and 40% of identity. Table 4 represents the average scores obtained.

Scores	Q-scores		T-scores	
	Before	After	Before	After
MAFFT	93,71	**93,72**	84,38	**84,40**
MUSCLE	91,55	**91,64**	80,90	**81,06**
Clustal Omega	**90,60**	90,59	79,37	**79,38**

Table 4: Scores obtained using RV12 BALIBASE Benchmark

We note that for several datasets, our refinement algorithm *Refin-Align* can ameliorate the scores of many different multiple sequences alignments obtained by different multiple sequences alignment algorithms. In fact, the refinement multiple sequence alignment after refinement have the best SPS and CS scores for the most datasets used.

6 Conclusion and perspectives

In this paper, we presented a new refinement algorithm for multiple sequence alignment called *Refin-Align*. Our algorithm adopts a new definition of block and use theses blocks to construct a new multiple sequence alignment by realigning these blocks.

We assess our algorithm using different datasets extracted from different benchmarks and using the more efficient multiple sequence alignment algorithms MUSCLE, MAFFT and Clustal Omega. For several datasets, our algorithm can ameliorate the SPS and CS scores for the initial multiple alignment.

In future work, we would like also to compare the results obtained by our program to other refinement programs. We would like also to asses our algorithm on DNA and RNA datasets. We can also improve the scores by using different alignment algorithms to align the blocks in order to obtain the more accurate multiple sequence alignment.

References

[1] Needleman B.S., Wunsch, D. C., (1970) A general method applicable to the search for similarities in the amino-acid sequence of two proteins, *Journal of Molecular Biology*, 48, pp. 443–453. `https://doi.org/10.1016/0022-2836(70)90057-4.`

[2] Wang, L., Jiang, T., (1994). On the complexity of multiple sequence alignment, *J. Comput. Biol.* 1(4), pp. 337-348. `https://doi.org/10.1089/cmb.1994.1.337.`

[3] Thompson, J. D., Higgins, D. G., Gibson, T. J., (1994). CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleid Acids Research*, 22(22), pp. 4673-4680. `https://doi.org/10.1093/nar/22.22.4673.`

[4] Notredame, C., Heringa, J., Higgins, D., (2000). T-COFFEE: A novel method for fast and accurate multiple sequence alignments. *J. Molecular Biology*, 302(1), pp. 205-217. `https://doi.org/10.1006/jmbi.2000.4042.`

[5] Edgar, R.C. (2004). MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32(5), pp. 1792-1797. `https://doi.org/10.1093/nar/gkh340.`

[6] Katoh, K., Kuma, K., Toh, H., Miyata, T., (2013). MAFFT version 7: Improvement in accuracy of

multiple sequence alignment. *Molecular Biology and Evolution*, 30(4), pp. 772-780. `https://doi.org/10.1093/molbev/mst010`.

[7] Yongtao, Y., Cheung, David, W., Wang, Yadong W., Yin, S. M., Zhang, Q., Lam, T. W., Ting, H. F.,.(2013). GLProbs: Aligning Multiple Sequences Adaptively. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 12(1), pp. 67-78. `https://doi.org/10.1109/TCBB.2014.2316820`.

[8] Sievers, Fabian, Higgins, Desmond, G. (2014). Clustal Omega, accurate alignment of very large numbers of sequences, *Multiple Sequence Alignment Methods*, (1079), pp, 105-116. `https://doi.org/10.1007/978-1-62703-646-7_6`.

[9] Mokaddem A. Elloumi, M. (2013). New distances for improving progressive alignment algorithm. *Advances in Computing and Information Technology*, (177), pp. 243-251. `https://doi.org/10.1007/978-3-642-31552-7_26`.

[10] Kimura, M. (1983). The Neutral Theory of Molecular Evolution. *Cambridge University Press*, Cambridge. `https://doi.org/10.1017/CBO9780511623486`.

[11] Sneath, P., Sokal, R., (1973). Numerical Taxonomy. *San Francisco Freeman*, pp. 230-234. `https://doi.org/10.2307/2529664`.

[12] Saitou, N., Nei, M..(1987). The Neighbor-Joining Method: A New Method for Reconstructing Phylogenetic Trees. *Mol. Biol. Evol*, 4(4), pp. 406-425. `https://doi.org/10.1093/oxfordjournals.molbev.a040454`.

[13] Naznin, F, Sarker. R, Essam D, (2012). Progressive Alignment Method Using Genetic Algorithm for Multiple Sequence Alignment. *IEEE Transactions on Evolutionary Computation*, 16(5), pp. 615-631. `https://doi.org/10.1109/TEVC.2011.2162849`.

[14] Behera, N., Jeevitesh, M. S., Josea, J., Kant, K., Dey, A., Mazher, J., (2017). Higher accuracy protein multiple sequence alignments by genetic algorithm. *Procedia Computer Science*, 108, pp. 1135-1144. `https://doi.org/10.1016/j.procs.2017.05.100`.

[15] J. D. Thompson, J. C. Thierry, O. Poch (2003) RASCAL: rapid scanning and correction of multiple sequence alignments. *Bioinformatics*, 19(9), pp 1155-1161. `https://doi.org/10.1093/bioinformatics/btg133`.

[16] Chakrabarti, S., Lanczycki, C. J., Panchenko, A. R., Przytycka, T. M., Thiessen, P. A., Bryant, S. H., (2006). Refining multiple sequence alignments with conserved core regions, *Nucleic Acids Res*,34(9), pp. 2598-2606. `https://doi.org/10.1093/nar/gkl274`.

[17] Wallace, I.M., Blackshields, G., Higgins, D.G., (2005). Multiple sequence alignments. *Current Opinion in Structural Biology*, 15, pp. 261-266. `https://doi.org/10.1016/j.sbi.2005.04.002`.

[18] Lyras, Dimitrios P., Metzler, Dirk. (2014) ReformAlign: improved multiple sequence alignments using a profile-based meta-alignment approach. *BMC bioinformatics*, 15(1), pp. 265. `https://doi.org/10.1186/1471-2105-15-265`.

[19] Altschul, S.F.(1989). Gap costs for multiple sequence alignment. *J Theor Biol.*, 138(3), pp. 297–309. `https://doi.org/10.1016/S0022-5193(89)80196-1`.

[20] Mokaddem, A.,Hadj, A. B., Elloumi, M., (2018). Pro-malign: Multiple Sequence Alignment Algorithm using Approached Profile. *Journal of software*, 13 (1), pp. 57-65. `https://doi.org/10.17706/jsw.13.1.57-65`.

[21] Muller, T., Spang. R, Vingron. M, (2002). Estimating amino acid substitution models: a comparison of dayhoff's estimator, the resolvent approach and a maximum likelihood method. *Mol Biol Evol*, 19(1), pp. 8-13. `https://doi.org/10.1093/oxfordjournals.molbev.a003985`.

[22] Thompson, J.D., Plewniak, F., Poch, O. (1999). BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1): pp. 87-88. `https://doi.org/10.1093/bioinformatics/15.1.87`.

[23] Thompson, J.D.,, Koehl P., Ripp R., Poch O.(2005) BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. *PROTEINS: Structure, Function, and Bioinformatics*, 61(1), pp. 127-136. `https://doi.org/10.1002/prot.20527`.

[24] Raghava, G. P., Searle, S. M., Audley, P. C., Barbe,r J. D., Barton, G. J., (2003). OXBENCH: a benchmark for evaluation of protein multiple sequence alignment

accuracy. *BMC Bioinformatics*, 4(1), pp. 47. https://doi.org/10.1186/1471-2105-4-47.

[25] Barton, G.J., Sternberg, M.J.: (1987). A strategy for the rapid multiple alignment of protein sequences, confidence levels from tertiary structure comparisons, *J Mol Biol*, 198(2), pp. 327-337. https://doi.org/10.1016/0022-2836(87)90316-0.

[26] Stebbings, L. A. and Mizuguchi, K., (2004). HOMSTRAD: recent developments of the Homologous Protein Structure Alignment Database. *Nucleic Acids Research*, 32, pp. 203–207. https://doi.org/10.1093/nar/gkh027.