RBT-Km: K-Means Clustering for Multiple Sequence Alignment

Javid Taheri
School of Information Technologies, J12
The University of Sydney
Sydney, NSW 2006, Australia
j.taheri@usyd.edu.au

Albert Y. Zomaya
School of Information Technologies, J12
The University of Sydney
Sydney, NSW 2006, Australia
a.zomaya@usyd.edu.au

Abstract— **This paper presents a novel approach for solving the Multiple Sequence Alignment (MSA) problem. K-Means clustering is combined with the Rubber Band Technique (RBT) to introduce an iterative optimization algorithm, namely RBT-Km, to find the optimal alignment for a set of input protein sequences. In this technique, the MSA problem is modeled as a Rubber Band, while the solution space is modeled as plate with several poles corresponding locations in the input sequences that are most likely to be correlated and/or biologically related. K-Means clustering is then used to discriminate biologically related locations from those that may appear by chance. RBT-Km is tested with one of the well-known benchmarks in this field (BALiBASE 2.0). The results demonstrate the superiority of the proposed technique even in the case of formidable sequences.**

Keywords- Multiple Sequence Alignment, K-Means Clustering.

I. INTRODUCTION

Sequence Alignment algorithms are techniques used to find similarity among several DNA/Protein sequences. These algorithms are classified into two main categories: Pair-wise and MSA algorithms, each designed for special purposes. In Pair-wise algorithms, the main goal is to find the similar or closely related parts (motifs) of two sequences; whereas in MSA, the main goal is to find the consensus parts of more than two sequences.

Several algorithms have already been suggested to solve this problem in each of the above two categories. Among these techniques, there exist several classical methods, such as Dynamic Programming (DP), that can always find the optimal alignment for any two sequences (Pair-wise). However, these techniques cannot always be generalized to MSA cases (due to the excessive computation that is incurred after the addition of each extra sequence). Therefore, using classical methods in the MSA case is almost impossible. In fact, because it has been shown that MSA is NP-Complete [1], heuristics are mainly used to solve this problem.

Regardless of the solving technique, MSAs can be categorized into three main solution categories: exact, progressive and iterative [2]. In exact methods, which are usually the generalized methods of the Needleman and Wunsch algorithm [3], all sequences are aligned simultaneously to find the optimal answer. The main drawback of this class of algorithms is their massive computational need, usually impossible to find the answer in polynomial time. In progressive algorithms, sequences are first aligned two-by-two (using an appropriate Pair-wise algorithm) before finding the final alignment. Then, an alignment guidance tree is generated based on these Pair-wise alignment scores. Starting from the two closest ones, sequences are combined step by step to find the optimal answer. In this case, currently aligned sequences are modified to get the best fit for new combining sequences. Although this class of algorithms usually manages to find reasonable alignments (especially to generate phylogenetic trees), their main disadvantage is their sensitivity to the local minima as they easily get trapped by them. Despite progressive approaches, in the iterative methods, all sequences are aligned simultaneously. Here, using one or more heuristic algorithms, an initial answer is calculated first. Then, this initial answer is improved iteratively using intelligent routines designed for this type of MSAs. Although these algorithms are not as sensitive as progressive algorithm to the local minima, they have their own drawbacks. For example, achieving a reasonable final answer for these algorithms is greatly related to their initial answers.

Based on these methods, a number of alignment algorithms are designed to solve the MSA problem, such as MULTALIGN [4], MULTAL [5], PILEUP [6] and CLUSTALX [7], which provides a graphical interface for CLUSTALW [8]. They all use a global alignment algorithm [3] to construct an alignment of the entire length of the sequences. Their main difference is in the order they combine the input sequences. MULTAL deploys a sequential branching method to align the two closest sequences before building up the final alignment by subsequently aligning the next closest sequence to it. MULTALIGN and PILEUP construct a guide tree using UPGMA [9]. This tree is then used to align larger and larger groups of input sequences. CLUSTALX that uses the alternative Neighbor-Joining algorithm [10] to construct a guide tree has one of the most sophisticated scoring systems. It considers sequence weighting, position dependant gap penalties, and the automatic switching among scoring matrices based on the degree of similarity among the input sequences. PIMA [11] uses a local DP algorithm to align only the most conserved motifs. Two versions of this method, ML_PIMA and SB_PIMA, are different based on their order of combining input sequences, maximum linkage and sequential branching algorithms, respectively. DIALIGN [12] focuses on a local

alignment based on segment-to-segment comparison to construct the final alignment. Then, an iterative procedure is deployed to combine these segments toward generating the final alignment. PRRP [13] iteratively divides the input sequences into two groups and then subsequently realign them using a global group-to-group alignment algorithm. SAGA [14] involves evolving a population of alignments in a quasi evolutionary manner to gradually improve their fitness. MAFFT [15] identifies the homologous regions by a Fast Fourier Transform (FFT) approach. Using its simplified scoring matrix, MAFFT manages to significantly reduce the CPU time and increases the accuracy of alignments even for sequences having large insertions and extensions as well as distantly related sequences of similar length. ProbCons [16], which computes posterior-probability matrices and expected accuracies for each Pair-wise comparison, applies the probabilistic consistency transformation, and then computes an expected accuracy guide tree to progressively generate the final alignment. T-Coffee [17] pre-processes a data set of all pair-wise alignments between the input sequences to generate a guide tree for the progressive alignment. T-Coffee not only does focus on the next aligned sequences but also on the whole set of input sequences. MUSCLE [18] as one of the very fast algorithms in this field has three stages: draft progressive, improved progressive, and refinement. At the completion of each stage, a multiple alignment is available and the algorithm can be terminated. The first stage builds a progressive alignment, the second stage, which might be iterated, attempts to improve the tree and builds a new progressive alignment according to this tree, and, the third stage performs iterative refinement using a variant of tree-dependent restricted partitioning. MUMMALS [19] uses probabilistic consistency and improves its alignment quality by using Pair-wise alignment and Hidden Markov Models (HMMs). Parameters for such models have been estimated from a large library of structure-based alignments. There are also other HMMs methods that use statistical models of the primary structure consensus to align input sequences [20, 21]. HMMT [22] uses a simulated annealing method to maximize the probability that an HMM represents the sequences to be aligned. RBT-I [23], RBT-L [24], and RBT-GA [25] are approaches that have overlaps with RBT-Km; however, they have different structures.

In this paper, a novel approach, RBT-Km, is presented to solve the MSA problem. The rest of the paper is structured as follows. Section 2 provides the problem statement. Sections 3 and 4 explain main concepts of K-Means clustering and RBT algorithm, respectively. In section 5, details of how Km-Means and RBT are combined to solve the MSA problem are given. Section 6 represents the simulation results. Discussion and analysis followed by conclusion are presented in sections 7 and 8, respectively.

II. Multiple Sequence Alignment

Let $\{S_1 \quad S_2 \quad \dots \quad S_N\}$ be N sequences over the alphabet set Ψ, which contains 4 and 20 characters for DNA and Proteins sequences, respectively. Also, let $\Psi' = \Psi \cup \{-\}$ be the superset of Ψ with and extra character for a 'gap'. The MSA problem can mathematically be defined as finding $S'_1 \quad S'_2 \quad \dots \quad S'_N$ with the following properties:

1. $S'_i = S_i$ for all $i = 1,2,\cdots,N$ providing that all gaps are removed from S'_i.
2. $|S'_1| = |S'_2| = \cdots = |S'_N|$ where $|S'_i|$ denotes the length of S'_i.
3. The alignment score, $Score(S') = \sum_i \sum_{j \neq i} sim(S'_i, S'_j) - \sum_i g(S'_i)$ is maximized where $sim(S'_i, S'_j)$ denotes a quotation of similarity between S'_i and S'_j, and, $g(S'_i)$ is related to gaps of S'_i.

Based on these properties, the MSA is defined as an optimization problem. However, regarding the fact that the complexity of the stated problem is exponentially increased by adding every extra sequence to the input sequences set, finding the optimal answer is not always possible. This is why classical methods like DP and Needleman's algorithm can be applied only for small number of short sequences.

III. K-Means Clustering Technique

The K-means clustering algorithm partitions a collection of n vectors $x_j; j = 1, \dots, n$ into c groups $G_i; i = 1, \dots, c$ and finds a cluster centre in each group so that the dissimilarity cost function is minimized as follows [26, 27]:

$$J = \sum_{i=1}^{c} J_i = \sum_{i=1}^{c} \left(\sum_{k:x_k \in G_i} \|x_k - c_i\|^2 \right)$$

Where $J_i = \sum_{k:x_k \in G_i} \|x_k - c_i\|^2$ is the cost function within group i. Here, the partitioned groups are typically defined by a $c \times n$ binary membership matrix U, where the element $u_{i,j}$ is '1' if the j-th data (i.e., x_j) belongs to group i, and is '0' otherwise, i.e.,

$$u_{i,j} = \begin{cases} 1 & \|x_j - c_i\|^2 \leq \|x_j - c_k\|^2 \\ & k \neq j \\ 0 & otherwise \end{cases}$$

The membership matrix U has the following properties:

1. $\sum_{i=1}^{c} u_{i,j} = 1 \qquad \forall j = 1, \dots, n$
2. $\sum_{i=1}^{c} \sum_{j=1}^{n} u_{i,j} = n$

The group centers, c_i's, are updated after each clustering step as follows.

$$c_i = \frac{1}{|G_i|} \sum_{k:x_k \in G_i} x_k$$

where

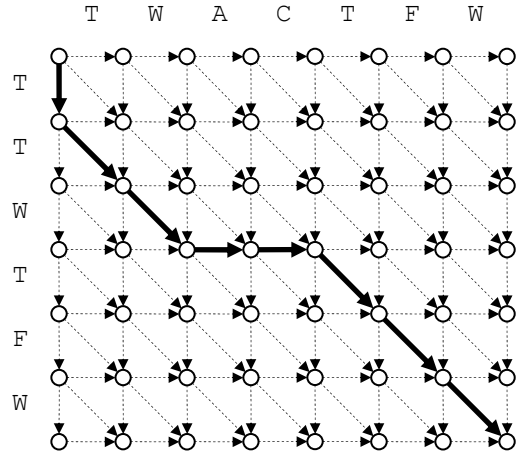
$$|G_i| = \sum_{j=1}^{n} u_{i,j}$$

The following procedure shows the overall view of this clustering technique.

Step 1: Let $X = (x_1 \quad x_2 \quad \dots \quad x_N)$ contains the input data to be clustered, where N is the number of data points.
Step 2: For all x_i's, let each x_i be a cluster centre.
Step 3: Combine heavily overlapped clusters.
Step 4: Repeat Steps 5–7 for all x_i's.
Step 5: Let m be the cluster number that x_i already belongs to.
Step 6: Find the nearest cluster centre to x_i, and let n be this cluster number.
Step 7: If $m \neq n$, remove x_i from m, add x_i to n, and update both clusters.
Step 8: Repeat Steps 4–7 until no further x_i is transferred from one cluster to another.

IV. Rubber Band Technique

The Rubber Band Technique is an iterative heuristic to solve the MSA [23-25]. In this approach, which is inspired by the general behavior of a rubber band on a plate with poles, an initial answer is generated before launching the main optimization procedure. Using several operators, this initial answer is modified iteratively to obtain better alignment scores. The following definitions are essential for the clarification of this optimization procedure.



Input Sequences
T W A C T F W
T T W T F W

Aligned Sequences
- T W A C T F W
T T W - - T F W

Figure 1. A sample Grid Answer Space with a Rubber Band

A. Grid Answer Space

The Grid Answer Space (GAS), which is the extended version of the grid table used in DP for Pair-wise alignment, is a multi-dimensional table with a sequence placed in one of its axes. The use of this table provides a unique one-to-one relationship between any possible answer of a MSA and the associated arrowed line as depicted in Fig. 1.

B. Rubber Band

Any answer for a MSA can be presented by one and only one arrowed line. This unique arrowed line for each answer is called a Rubber Band (RB).

C. Primary Pole

A Primary Pole (PP) refers to a fixed point in GAS that the RB is ought to pass by. In fact, PPs are the sections of the GAS that force the optimization procedure to align a predefined number of characters (from different sequences) with each other.

D. Secondary Pole

Secondary Poles refer to grid points in GAS that a RB passes through one-by-one to generate the final answer. In fact, PPs are usually far apart from one other, whereas secondary poles need to be adjacent. This type of poles is only used to connect PPs to each other. For brevity purposes, secondary poles are referred to as 'poles' for the rest of this paper.

E. Primary Pole Score

As described earlier, each PP points out predefined locations of input sequences that need to be aligned with each other. If these locations are augmented in a single string, the Primary Pole Score for that particulate PP is defined as the alignment score of that augmented string (with respect to the scoring matrix used for the whole alignment).

F. Sticky Poles

Sticky Poles (SPs) are imaginary poles in the system related to locations in a GAS with high Primary Pole Scores. That is, the optimization procedure can have a pole with a high Primary Pole Score if it places a PP on that special place. Therefore, each SP is located in a GAS to represent a column from the input sequences to align identical or closely related nucleotides from different input sequences with one another.

G. Alignment Score

In each MSA instant, an Alignment Score is defined to evaluate the quality of each answer. The Sum-of-Pairs Score (SPS) with Penalized Gap Opening is the criterion used in this approach. In SPS, each column in an alignment is scored by summing the scores of all pairs of characters in that column. The score of the final alignment is then summed over all column scores. In the Penalized Gap Opening scheme, there are two factors to calculate the score/cost of a gap: opening and extension. Gap opening is applied to each gap once and the gap extension corresponds to the length of each gap. The cost of a gap opening is usually considered to be 5–10 times more than that of a gap extension [23-25]. The use of two factors in calculating a gap is related to a well-known biological fact that

having few longer gaps is more plausible than having several short gaps in an alignment.

H. Rubber Band Technique Operators

The main optimization process of RBT [23, 24] consists of various operators that are launched either one or several times to iteratively improve the quality of an initially generated alignment. These operators and their order of deployment are as follows:

Step 1: Initialize Sticky Poles
Step 2: For a predefined number of Tries repeat Steps 3-7
Step 3: Move Blocks
Step 4: Move Primary Poles
Step 5: Jump Primary Poles
Step 6: Jam Primary Poles
Step 7: Align Primary Poles
Step 8: Fine Tuning

V. K-MEANS IN MULTIPLE SEQUENCE ALIGNMENT

Finding the locations, in different input sequences, that are most likely biologically related, has a direct impact on the success of the RBT algorithm [23, 24]. For example, in RBT-I [23], these locations, SPs, are estimated based on similar amino acids indexed in the input proteins. In RBT-L [24], these SPs are calculated based on the relative locations of similar amino acids in different sequences. However, after close examination of the quality of these SPs, in both versions of RBT, it had been noted that SPs that resemble motifs in an alignment are usually grouped together. Therefore, RBT-Km is designed to identify these groups and discriminate them from singular SPs that might have appeared by chance. The removal of singular SPs will help the RBT algorithm to have more of such groups in its final alignment. This should yield to finding more motifs and hopefully provide better biologically meaningful alignments. The overall procedure of RBT-Km is as follows:

TABLE 1.: BALIBASE SCORES FOR ALL ALGORITHMS

	Ref 1 (82)		Ref 2 (23)		Ref 3 (12)		Ref 4 (12)		Ref 5 (12)		Overall (141)	
Algorithm	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS	SP	CS
RBT-Km	0.915	0.871	0.954	0.794	0.924	0.830	0.944	0.884	0.987	0.979	0.927	0.861
PROBCONS(ir=100)	0.911	0.853	0.942	0.616	0.840	0.635	0.937	0.811	0.974	0.893	0.918	0.796
SPEM	0.908	0.839	0.934	0.573	0.814	0.569	0.974	0.908	0.974	0.923	0.915	0.786
PROBCONS-ext	0.900	0.825	0.942	0.591	0.843	0.611	0.938	0.810	0.981	0.922	0.912	0.776
PROBCONS	0.901	0.826	0.944	0.613	0.841	0.613	0.901	0.723	0.979	0.919	0.910	0.772
PSALIGN[PROBCONS]	0.901	0.840	0.940	0.617	0.809	0.522	0.901	0.697	0.980	0.936	0.906	0.773
PROBCONS(ir=0)	0.895	0.823	0.942	0.616	0.840	0.635	0.908	0.721	0.974	0.893	0.906	0.771
MUSCLE	0.887	0.808	0.935	0.563	0.825	0.564	0.876	0.609	0.968	0.902	0.896	0.738
PSALIGN[TCOFFEE]	0.884	0.805	0.936	0.583	0.785	0.548	0.891	0.684	0.973	0.900	0.892	0.745
MUSCLE-p	0.871	0.795	0.928	0.558	0.813	0.550	0.857	0.598	0.974	0.891	0.883	0.727
T-COFFEE	0.866	0.774	0.934	0.561	0.785	0.487	0.918	0.730	0.958	0.903	0.882	0.722
MAFFT	0.867	0.781	0.924	0.502	0.788	0.504	0.916	0.727	0.963	0.859	0.882	0.714
PRALINE	0.904	0.839	0.940	0.610	0.764	0.558	0.799	0.539	0.818	0.686	0.882	0.739
NWNSI	0.867	0.788	0.923	0.514	0.787	0.514	0.904	0.742	0.963	0.859	0.881	0.722
KALIGN	0.850	0.000	0.920	0.000	0.790	0.000	0.880	0.000	0.920	0.000	0.865	0.000
CLUSTALW	0.861	0.773	0.932	0.568	0.753	0.460	0.834	0.522	0.859	0.638	0.861	0.680
FFTNSI	0.838	0.732	0.908	0.496	0.708	0.350	0.793	0.451	0.947	0.831	0.844	0.646
PRIMEpcw,mea	0.789	0.629	0.925	0.439	0.856	0.547	0.923	0.603	0.890	0.521	0.837	0.580
DIALIGN	0.811	0.709	0.893	0.359	0.684	0.344	0.897	0.762	0.940	0.843	0.832	0.637
PRIMEpcw	0.785	0.623	0.917	0.441	0.858	0.557	0.906	0.579	0.885	0.526	0.832	0.576
Hybrid CSA	0.827	0.653	0.919	0.413	0.786	0.362	0.705	0.319	0.836	0.569	0.829	0.554
PRIMEpcw,ag	0.781	0.618	0.917	0.435	0.844	0.511	0.913	0.588	0.876	0.509	0.828	0.567
PRIMEafn,mea	0.783	0.610	0.902	0.368	0.851	0.547	0.882	0.502	0.875	0.491	0.824	0.546
PRIMEmea,mea	0.794	0.627	0.880	0.374	0.829	0.478	0.855	0.511	0.888	0.544	0.824	0.556
PRIMEafn	0.779	0.560	0.899	0.388	0.845	0.529	0.883	0.514	0.864	0.477	0.820	0.518
PRIMEafn,ag	0.775	0.598	0.898	0.381	0.823	0.493	0.869	0.447	0.859	0.476	0.814	0.530
ALIGN-M	0.766	0.000	0.884	0.000	0.684	0.000	0.911	0.000	0.917	0.000	0.803	0.000
PRRN	0.748	0.563	0.902	0.405	0.822	0.483	0.860	0.487	0.822	0.421	0.795	0.512
C-MUSCLE	0.781	0.678	0.876	0.463	0.705	0.418	0.683	0.380	0.828	0.652	0.786	0.593
C-CLUSTALW	0.774	0.667	0.858	0.446	0.651	0.347	0.630	0.338	0.760	0.516	0.764	0.563
DIALIGN-T	0.684	0.477	0.855	0.278	0.737	0.346	0.795	0.426	0.781	0.397	0.734	0.422
POA	0.666	0.451	0.857	0.265	0.733	0.343	0.805	0.412	0.754	0.323	0.722	0.397
SAGA	0.841	0.000	0.586	0.000	0.506	0.000	0.289	0.000	0.642	0.000	0.707	0.000
PILEUP8	0.832	0.000	0.429	0.000	0.323	0.000	0.710	0.000	0.639	0.000	0.696	0.000
MULTALIN	0.834	0.729	0.517	0.440	0.303	0.385	0.292	0.223	0.627	0.462	0.673	0.587
SB_PIMA	0.821	0.000	0.379	0.000	0.267	0.000	0.794	0.000	0.508	0.000	0.673	0.000
ML_PIMA	0.810	0.000	0.371	0.000	0.372	0.000	0.705	0.000	0.572	0.000	0.672	0.000

Step 1: Locate All Sticky Poles
Step 2: Use K-Means Clustering to identify large clusters.
Step 3: Delete small clusters from Sticky Poles set.
Step 4: For a predefined number of Tries repeat Steps 5–9
Step 5: Move Blocks
Step 6: Move Primary Poles
Step 7: Jump Primary Poles
Step 8: Jam Primary Poles
Step 9: Align Primary Poles
Step 10: Fine Tuning

VI. Simulation Results

In MSA, due to the fact that there is no concrete criterion to evaluate the quality of a given algorithm, standard benchmarks, such as, BALiBASE, OXBench, PREFAB and SMART, are provided to gauge the efficiency of different MSA algorithms. BALiBASE is the most popular benchmark among all of them, and therefore has been used in this work as well. BALiBASE version 3 is the last version of this benchmark. Nevertheless, because not all proposed methods have been tested to show their performances for BALiBASE version 3, BALiBASE version 2 (most commonly used benchmark) is used instead.

BAliBASE version 2.0.1 [28] was dedicated to the evaluation of multiple alignment programs and was divided into five hierarchical reference sets of: (1) equidistant sequences with various levels of conservation, (2) families aligned with a highly divergent 'orphan' sequence, (3) subgroups with less than 25% residue identity between groups, (4) sequences with N/C-terminal extensions, and (5) internal insertions. References #1, #2, #3, #4, and #5 contain 82, 23, 12, 12, and 12 benchmarks/alignments, respectively. These benchmarks/alignments have been manually verified and corrected by superposition of all known three-dimensional structures, using the lsqman program [29]. In this benchmark, an open source program is also provided to score the quality of each answer by comparing it with the one biologist found manually. The maximum score is 1.0 and is assigned to the alignments that are identical to the benchmark's answer; minimum is 0.0 and is assigned to unrelated/unrealistic answers; and, a number between 0.0 and 1.0 for the others. The closer to the manually calculated answer, the higher the score would be.

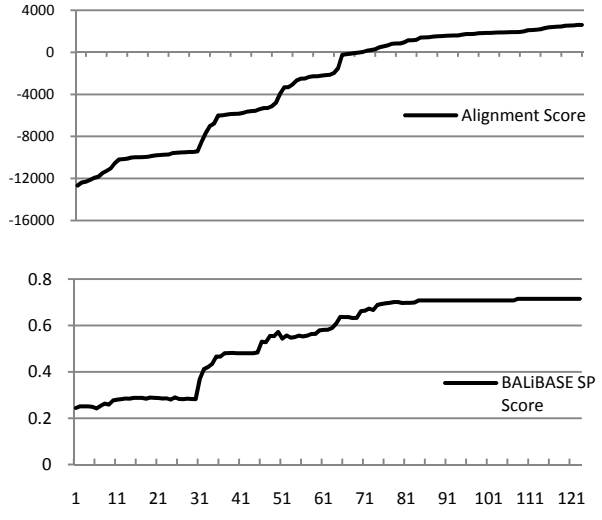


Figure 2. Alignment Score/ BALiBASE SP Score for 1wit in Reference #3

To demonstrate the performance of the approach proposed in this paper, RBT-Km is checked using all benchmarks of the BALiBASE 2.0.1. For all these benchmarks, BLOSUM35/62 scoring matrices with the gap penalty of 10 and 1 for the Gap Opening and Gap Extension is selected, respectively. Table 1 shows performance of RBT-Km compare with other approaches obtained from [18, 30-36].

VII. Discussion and Analysis

The results obtained by using RBT-Km were quite different and interesting, covering a vast variety of situations. In summary, similar to other approaches formerly presented to solve this problem, although RBT-Km did not manage to find the identical alignments to all benchmark solutions, it was generally successful. The following sections explain this in more detail.

A. Alignment Score vs. BALiBASE score

The first observation made after analyzing the solving trajectory of the benchmarks was the not very strong relationship between Alignment Score (purely depends on the Scoring matrix (BLOSUM35/62 in this case)), and the BALiBASE score (purely based on biological facts). However, they seem to be fairly related. In several cases, gaining higher Alignment Scores yields better BALiBASE scores; although, this cannot be always guaranteed. To investigate this relationship more, we executed the algorithm with different scoring matrices, gap opening and extension values. In almost all cases, the Alignment Score and BALiBASE score showed the same amount of uncorrelation. Nevertheless, it seems that in most cases, alignments with higher Alignment Scores have better BALiBASE score as well. Fig. 2 shows a sample of this uncorrelation for 1wit from Reference#3.

B. RBT-Km and BALiBASE SP score

Figs. 3-7 show the graphical representation of Table 1, sorted in descending order of BALiBASE SP score for each reference. These figures reveal that RBT-Km produces the best SP score for four References #1, #2, #3, and #5; and the second best for Reference #4. This leads to superior overall performance of RBT-Km for all benchmarks when compared with other algorithms as shown in Fig. 8.

C. RBT-Km and BALiBASE CS score

Although RBT-Km shows better performance in four out of five references for BALiBASE's SP score, the improvement for CS score is more significant. RBT-Km managed to significantly improve CS score for References #1, #2, #3, #4, and #5 for 1.8%, 17.8%, 19.5%, 7.3%, and 8.6%, respectively. We can conclude that RBT-Km's answers manifest more biological relationship among sequences as they obtained far better CS scores regardless of their similar SP scores. For example, in Reference #2 and #3, RBT-Km's SP scores are improved by only 1.2% and 8.3%, respectively; while, the CS

score is improved by almost 20% in both cases. This yields a significant 6.5% overall CS improvement for all references.

VIII. Conclusion

In this paper, a novel approach (RBT-Km) based on the combination of Rubber Band Technique and K-Means clustering is proposed to solve the Multiple Sequence Alignment problem. In this approach, K-Means clustering is used to identify locations in the input sequences where there is a higher likelihood of finding more biologically related sections (motifs). RBT is then used to find the best possible alignment. RBT-Km is tested by employing all benchmarks from BALiBASE 2.0.1. The results showed great promise of the proposed approach.

IX. Software Availability

RBT-Km is written in C++ under Microsoft Visual Net 2005 for Windows Operating System. This algorithm is a part of "ProteinAlingment" software that is developed to implement several multiple alignment algorithms. The web-based version of this software has not been implemented yet, however, researchers can have a free setup package of this software by contacting the authors.

References

[1] L. Wang and T. Jiang, "On the complexity of the multiple sequence alignment," Journal of Computational Biology, vol. 1, pp. 337 - 348, 1994.

[2] L. Abdesslem, M. Soham, and B. Mohamed, "Multiple sequence alignment by quantum genetic algorithm," in 20th International Parallel and Distributed Processing Symposium (IPDPS 2006), 2006.

[3] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," Journal of Molecular Biology, vol. 48, pp. 443 - 453, 1970.

[4] G. J. Barton and M. J. E. Sternberg, "A strategy for the rapid multiple alignment of protein sequences," Journal of Molecular Biology, vol. 198, pp. 327 - 337, 1987.

[5] W. R. A. Taylor, "Flexible method to align large numbers of biological sequences," Journal of Molecular Biology Evolution, vol. 28, pp. 161 - 169, 1988.

[6] J. Devereux, P. Haeberli, and O. Smithies, "A comprehensive set of sequence analysis programs for the VAX," Nucleic Acids Research, vol. 12, pp. 387 - 395, 1984.

[7] J. D. Thompson, T. J. Gibson, F. Plewniak, F. Jeanmougin, and D. G.

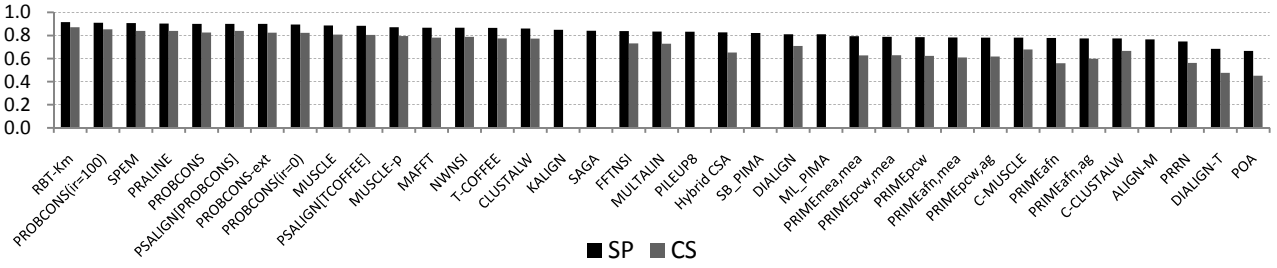


Figure 3. BALiBASE scores for Reference #1

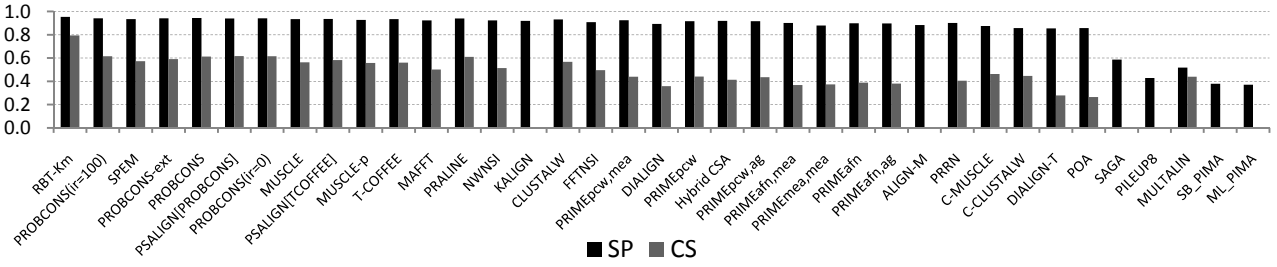


Figure 4. BALiBASE scores for Reference #2

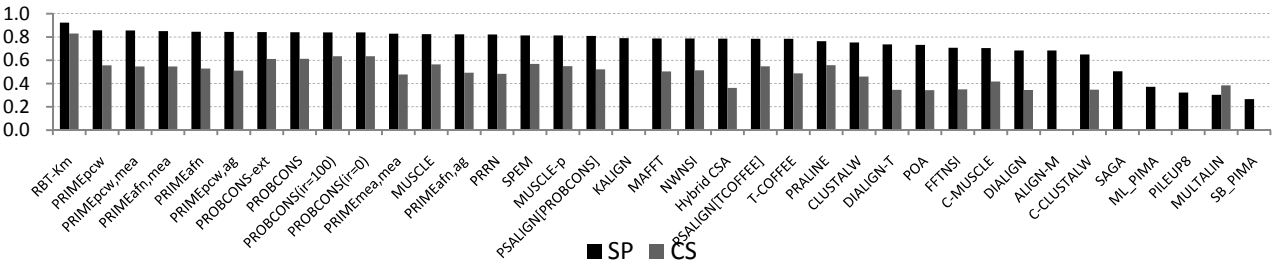


Figure 5. BALiBASE scores for Reference #3

Higgins, "The CLUSTAL_X windows interface: flexible strategies for multiple sequence alignment aided by quality analysis tools," Nucleic Acids Research, vol. 24, pp. 4876 - 4882, 1997.

[8] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "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.

[9] P. H. A. Sneath and R. R. Sokal, "Numerical Taxonomy," W.H. Freeman and Company, San Francisco, California, USA, pp. 230 - 234, 1973.

[10] Saitou and Nei, "The neighbor-joining method: a new method for reconstructing phylogenetic trees," Journal of Molecular Biology and Evolution, vol. 4, pp. 406 - 425, 1987.

[11] R. F. Smith and T. F. Smith, "Pattern-induced multi-sequence alignment (PIMA) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modelling," Protein Engineering, vol. 5, pp. 35 - 41, 1992.

[12] B. Morgenstein, A. Dress, and T. Werner, "Multiple DNA and protein sequence alignment based on segment-to-segment comparison," in Proceedings of the National Academy of Sciences, 1996, pp. 12098 - 12103.

[13] O. Gotoh, "Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments," Journal of Molecular Biology, vol. 264(4), pp. 823 - 838, 1996.

[14] C. Notredame and D. G. Higgins, "Saga - sequence alignment by genetic algorithm," Nucleic Acids Research, vol. 24 (8), pp. 1515 - 1524, 1996.

[15] K. Katoh, K. Misawa, K. Kuma, and T. Miyata, "MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform," Nucleic Acids Research, vol. 30(14), pp. 3059 - 3066, 2002.

[16] C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," Genome Research, vol. 15(2), pp. 330 - 340, 2005.

[17] C. Notredame, D. G. Higgins, and J. Heringa, "T-Coffee: A novel method for fast and accurate multiple sequence alignment," Journal of Molecular Biology, vol. 302, pp. 205 – 217, 2000.

[18] R. C. Edgar, "MUSCLE: multiple sequence alignment with high accuracy and high throughput," Nucleic Acids Research, vol. 32, pp. 1792 – 1797, 2004.

[19] 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(16), pp. 4364 - 4374, 2006.

[20] P. Baldi, Y. Chauvin, T. Hunkapiller, and M. A. McClure, "Hidden Markov models of biological primary sequence information," in Proceedings of the National Academy of Sciences, 1994, pp. 1059 - 1063.

[21] A. Krogh, I. Mian, and D. Haussler, "A hidden Markov model that finds genes in Escheria Coli DNA," Nucleic Acids Research, vol. 22, pp. 4768

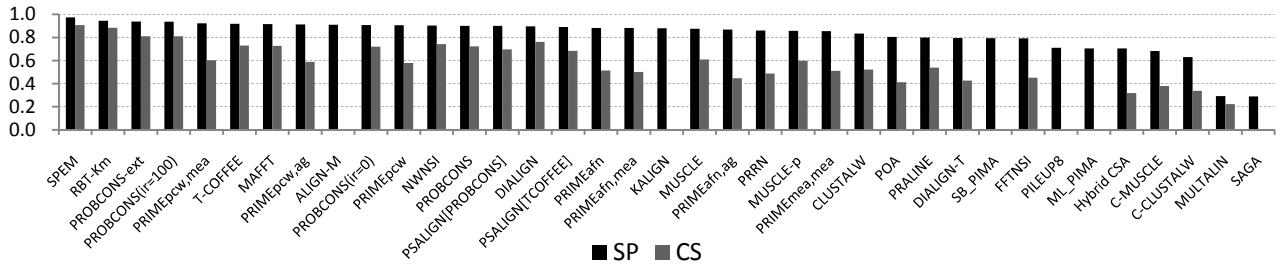


Figure 6. BALiBASE scores for Reference #4

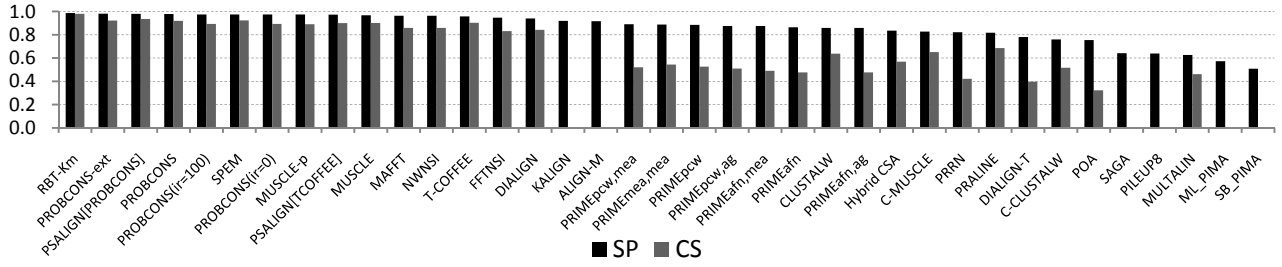


Figure 7. BALiBASE scores for Reference #5

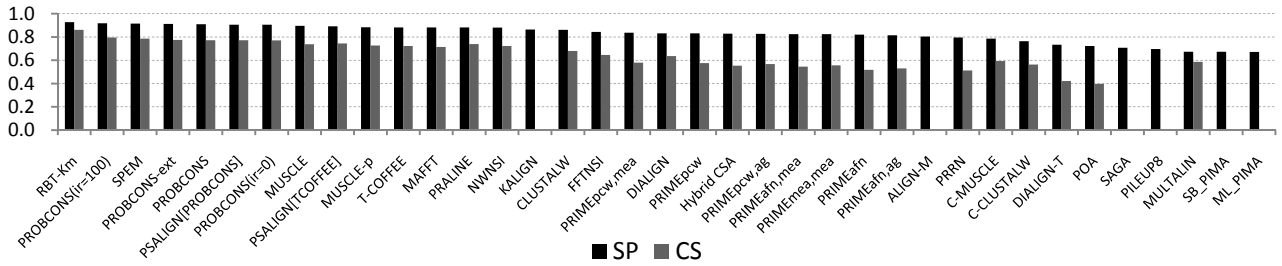


Figure 8. BALiBASE scores for all References

- 4778, 1994.

[22] S. R. Eddy, "Multiple Alignment Using Hidden Markov Models," 3rd. ISMB, pp. 114 – 120, 1995.

[23] J. Taheri and A. Y. Zomaya, "RBT-I: A Novel Approach for Solving the Multiple Sequence Alignment Problem," in The 6th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA-08), Qatar, 2008, pp. 28-36.

[24] J. Taheri, A. Y. Zomaya, and B. B. Zhou, "RBT–L: A Location Based Approach for Solving the Multiple Sequence Alignment Problem," International Journal of Bioinformatics Research and Applications (IJBRA), vol. (in press), 2009.

[25] J. Taheri and A. Y. Zomaya, "RBT-GA: A novel metaheuristic for solving the multiple sequence alignment problem," Journal of BMC Genomics, vol. (in press), 2009.

[26] J. A. Hartigan, Clustering Algorithms. New York, NY, USA: John Wiley & Sons, Inc. , 1975.

[27] T. Kanungo, D. M. Mount, N. Netanyahu, C. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithm: analysis and implementation," IEEE Transactions on Pattern Analysis and Machine Intelligence,, vol. 24, pp. 881-892, July 2002.

[28] J. D. Thompson, F. Plewniak, and O. Poch, "BAliBASE: a benchmark alignment database for the evaluation of multiple alignment programs," Bioinformatics vol. 15, 1999.

[29] G. J. Kleywegt and T. A. Jones, "Where freedom is given, liberties are taken," Structure, vol. 3, pp. 535 - 540, 1995.

[30] V. Cutello, D. Lee, G. Nicosia, M. Pavone, and I. Prizzi, "Aligning Multiple Protein Sequences by Hybrid Clonal Selection Algorithm with Insert-Remove-Gaps and BlockShuffling Operators," in Lecture Notes in Computer Science. vol. 4163 Berlin / Heidelberg: Springer, 2006 pp. 321-334.

[31] T. Lassmann and E. L. Sonnhammer, "Kalign - an accurate and fast multiple sequence alignment algorithm," BMC Bioinformatics, p. 6:298, 2005.

[32] S. Yamada, O. Gotoh, and H. Yaman, "Improvement in Speed and Accuracy of Multiple Sequence Alignment Program PRIME," IPSJ Transactions on Bioinformatics, vol. 1, pp. 2-12, 2008.

[33] P. S. Peres and E. S. d. Moura, "Application of Clustering Technique in Multiple Sequence Alignment," in Lecture Notes in Computer Science. vol. 3772 Berlin / Heidelberg: Springer, 2005, pp. 202-205.

[34] C. B. Do, M. S. P. Mahabhashyam, M. Brudno, and S. Batzoglou, "ProbCons: Probabilistic consistency-based multiple sequence alignment," Genome Research, vol. 15, pp. 330-340, 2005.

[35] S.-H. Sze, Y. Lu, and Q. Yang, "A Polynomial Time Solvable Formulation of Multiple Sequence Alignment," Journal of Computational Biology, vol. 13, 2006.

[36] M. Zhang, W. Fang, J. Zhang, and Z. Chi, "MSAID: multiple sequence alignment based on a measure of information discrepancy," Computational Biology and Chemistry, vol. 29, pp. 175–181, 2005.