ISSN: 0975-766X
CODEN: IJPTFI
Available Online through
www.ijptonline.com
Research Article

PROTEIN MULTIPLE SEQUENCE ALIGNMENT USING PARTITIONED OPTIMIZATION ALGORITHAMS

Manish Kumar*
CSE Department, Indian School of Mines, Dhanbad-826004, India.
Email: manishkumar@cse.ism.ac.in

Received on 05-11-2015 *Accepted on 25-11-2015*

Abstract

This article provides a general solution for the problem of Multiple Sequence Alignment of biological sequences by dividing the sequences into many sub sequences and then applying Genetic Algorithms to each sub sequences. The main objective of this research work is to maximize the similarity between the sequences by adding and shuffling gaps and then calculating the score of each column of the alignment so as to get reliable multiple sequence alignment. The experimental results show that the proposed method is able to generate better alignments for the benchmark data sets (Balibase ref. 2) than other most commonly used methods, such as the CLUSTAL X, PRRP, DIALI, RBT-GA and PILEUP8. Computational results demonstrate the superiority of the proposed approach for many sequences with different length and identity when compared with methods stated above. The new approach is more robust and obtains better mathematical and biological quality.

Keywords: Bioinformatics, Genetic Algorithms, Multiple Sequence Alignment.

I. Introduction

A Multiple Sequence Alignment (MSA) is basic task in bioinformatics. It consists of aligning several sequences in order to show the fundamental relationship and the common characteristics between a set of protein or nucleic sequences. This task is a fundamental platform for several other more complex tasks such as protein analysis, identification of functional sites in genomic sequences, structural and functional prediction of sequences and the construction of phylogenetic trees. Unfortunately, finding an accurate multiple alignments has been shown NP-hard [1]. Indeed, the MSA is an optimization problem which exhibits a great temporal and space complexity. Therefore, several methods were proposed which can be grouped in three great classes [2]. The first class includes exact methods which use a generalization of Needleman algorithm [3] in order to align all the sequences simultaneously. Although

exact methods give optimal solutions, their main shortcoming is their complexity which becomes even more critical with the increase of the number of sequences. The second class contains methods based on a progressive approach [4]. For these methods the multiple sequence alignment is built gradually according to a given order of the sequences, starting by the alignment of two sequences then it adds gradually the remaining sequences one by one to the preceding alignment. The progressive methods are simple, fast and generally give alignments of good qualities. However, their major disadvantage is the problem of the local minima and consequently they can lead to poor quality solutions. To overcome this problem, the iterative methods of the third class were showed to be promising.

The basic idea is to start by an initial alignment and iteratively refines it through a series of suitable refinements called iterations. The process is reiterated until satisfaction of some criteria. Iterative methods can be deterministic or stochastic, depending on the strategy used to improve the alignment. The first stochastic iterative algorithm proposed in the literature uses an algorithm of simulated annealing [5]. However, this algorithm is very slow and it is appropriate to be used as improver [2]. Later, several other iterative algorithms which use various strategies like Genetic Algorithms GAs [6], Tabu Search [7], were proposed. Concerning the deterministic iterative methods, they involve extracting the sequence one by one from multiple alignments and realigning them to the remaining sequences. The process is reiterated until it does not have more possible improvements. Although the iterative methods generally gives more accurate alignments than the progressive methods, their major disadvantage is their high execution time.

In order to do the fast and efficient multiple sequence alignment analysis, a lot of methods or algorithms such as dynamic programming [3], progressive [4] and iterative alignment [5] method have been developed. Very short or very similar sequences can be aligned by hand. However, most interesting problems require the alignment of lengthy, highly variable or extremely numerous sequences that cannot be aligned solely by human effort. Instead, human knowledge is applied in constructing algorithms to produce high-quality sequence alignments, and occasionally in adjusting the final results to reflect patterns that are difficult to represent algorithmically (especially in the case of nucleotide sequences).

Computational approaches to sequence alignment generally fall into two categories [8, 9, 10]: global alignments and local alignments. Calculating a global alignment is a form of global optimization that "forces" the alignment to span the entire length of all query sequences. By contrast, local alignments identify regions of similarity within long sequences that are often widely divergent overall. Local alignments are often preferable, but can be more difficult to calculate because of the additional challenge of identifying the regions of similarity. A variety of computational algorithms

have been applied to the sequence alignment problem, including slow but formally optimizing methods like dynamic programming. Although one can use the dynamic programming approach to multiple sequence alignment, it very quickly becomes impractical for more than three or four sequences as the time complexity is $O(n^N)$, where n is the length of the sequences and N is the number of sequences. Therefore, MSAs require more sophisticated methodologies than pair wise alignment.

The resolution of large systems of nonlinear equations was achieved by dividing the problem into small sub problems in such a way that each sub problem is solved by a given numerical method. The algorithm described in this paper is based on the assumption that each sub problem can be locally solved by a given sequential method. After solving each part of the problem they all are combined to get the final alignment result. This fundamental was first introduced by Talukdar in 1982[11]. The problem dealt within the work of Talukdar was a system of algebraic equations, using two or more processors, that were coordinated by a process denominated the master which uses a global memory as a mean of communication between the processors. The goal is to understand whether it is possible to have better results by dividing the problem into many sub problems or not.

II. Materials and Methods

This section detailed about the proposed approach which is based on various parameters and are described below.

Here, the given population is divided into sub population and then each subpopulation is separately evaluated using defined objective function. For instance, for a population of 100 individual's two subpopulations of 50 individuals has been produced. At each generation, all sub population will interchange information with each other regarding their best and worth individuals. Now, the best individual in one sub population will migrate to other sub population so as to replace the worth individuals.

The same process will be repeated vice –versa in order to improve the alignment quality .Here, the percentage of individuals migrated to other sub group is restricted to only 30%, as in the experimental analysis it has been observed that above 30% migration of individuals causes a low alignment quality. Later on, all the sub group populations will be added to obtain the complete alignment.

A. Representation and initial generation:

By using a non codified representation of the solutions, real multiple sequence alignments are used as data structures for each individual. This means that chromosomes are represented by arrays of characters, on which each line corresponds to a sequence in the alignment and each column represents an amino acid at a specific position. The

possible values for each component of the individual are C, S, T, P, A, G, N, D, E, Q, H, R, K, M, I, L, V, F, Y and W which are in fact the amino acids. Also, the symbol "–" is used in order to represent a gap in the sequence.

Consider k sequences to be aligned. These k sequences are generally of different lengths, say, from l_i to l_k. In the proposed approach, a candidate alignment or parent alignment in the MSA problem is represented as an array of the sequences or simply a matrix, where each sequence is encoded as an array of characters in the considered alphabet set. The maximum number of columns in the matrix is limited to W= [$\alpha \times l_{max}$], where $l_{\max} = \max\{l_1, l_2 \ldots\ldots l_k\}$ and [x] is the smallest integer greater than or equal to χ and the parameter α is a scaling factor. In this study, each matrix candidate may have different number of columns and the value α =1.2 is chosen independent for each candidate according to the probability distribution N(1.3,0.2), where N (μ,) denotes a Gaussian distribution with its mean μ and variance [12].

The population is initially randomly generated by loading each sequence to each line of the array, determining the size of the largest sequence and completing each one of the sequences with the gap sign until they reach the size of the biggest sequence plus a random number of gaps between 0 and 25% of the size of the largest loaded sequence. These gaps are randomly positioned into the sequences. After the population's initialization, the solutions are combined and mutated, producing new individuals through a defined number of generations.

B. Fitness evaluation:

To compare different alignments, a fitness function is defined based on the number of matching symbols and the number and size of gaps. In biology, this fitness function is referred to as cost function and is given biological meaning by using different weights for different types of matching symbols and assigning gap costs when gaps are used. For scoring purpose, PAM 250 Matrix has been used as a scoring matrix to calculate score between different alignments.

In the experiment the fitness is calculated as:-

Fitness =

$$\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} scoring\ matrix(\ l_i, l_j)$$

Where,

n = number of sequences,

l_i = First sequence, l_j = second sequence

C. Crossover:

In this approach, only one parent A is selected and an entirely new individual B_1 is randomly generated. The selected parent A is then crossed over with the new and randomly created individual B_1. The offspring C_1 is kept if it is better than the parent A that is measured in fitness. Otherwise, it is discarded and another entirely new individual B_2 is randomly generated and a new crossover occurs between A and B_2. The iteration goes on until the offspring C_n is better than the parent A in fitness. C_n is then kept and put in the next generation [13].

D. Mutation:

The mutation operator [14] preserves diversification in the search. The mutation operator chosen was the random mutation. This operator is applied to each offspring in the population with a predetermined probability. For a randomly chosen gene i of an individual (genel ,.., genen ,gene n+1 , ... , gene 2n), the allele gene i is replaced by a randomly chosen value from a interval]0, 1[. The probability of the mutation in this work is 0.1%. With 1000 genes positions one should expect 1000 x 0.001 = 1 genes to undergo mutation for this probability value.

III. Algorithm of the Proposed Method

Step 1: Generate random population of *n* chromosomes (suitable solutions for the problem).

Step 2: Divide the given population into a number of sub populations.

Step 3: Evaluate the fitness *f(x)* of each chromosome *x* in every sub population.

Step 4: Apply crossover and mutation within each sub populations.

Step 5: With a crossover probability (0.6%), crossover the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

Step 6: With a mutation probability (0.01%), mutate new offspring at each locus (position in chromosome).

Step 7: Go for desired number of generation.

Step 8: Stop when desired number of fitness value found or maximum number of generation reached.

Step 9: Exchange the worst individual in one population with the best ones in the other population and vice versa. Only 30% migration of best individuals is allowed.

Step 10: Add all the sub populations to obtain the complete alignment.

IV. Result

In MSA, the optimal answer is unknown and there is no concrete criterion to evaluate the quality of a given algorithm, unlike the case for Pair-wise alignment where an optimal solution can always be found. Therefore,

standard benchmarks, like BALiBASE, are provided to gauge the efficiency of MSA algorithms. With the release of version 2.0 of BAliBASE, the alignments have been manually verified and corrected by superposition of all known three-dimensional structures, using the lsqman program [15].

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 (refer table 2). The closer to the manually calculated answer, the higher would be the score. In order to evaluate the performance of the proposed approach, the experiments were carried out with different datasets (ref. 2) [16] of different lengths from the BAliBASE database.

The population size was established to 1000 individuals, it means that in every generation 1000 Childs population will be produced .The maximum number of generations was 100 for the experimental study with a crossover probability of 0.6% and mutation rate of 0.01%. The scoring matrix used for the experiment is PAM 250 for each Protein sequences.

The extensive experiments on the proposed approach have been performed with the genetic algorithm by using C programming on an Intel Core 2 Duo processor with T9400 chipset, 2.53 GHz CPU and 2 GB RAM running on the Linux platform.

In references 2, 15 test cases were considered and they were compared with methods like CLUSTAL X, PRRP, DIALI, RBT-GA and PILEUP8 . After comparison, it has been found that the proposed method successfully found good solution in 12 test cases. Only RBT-GA method for 1ajsA and 2trx dataset and PRRP for 1cpt dataset outperformed the proposed method solution.

The results obtained by using the proposed approach were quite different and interesting, covering a vast variety of situations. In summary, similar to other approaches formerly presented to solve this problem, an attempt has been made to properly align the protein sequences. Although, RBT-GA did not manage to find the identical alignments to benchmark answers for all the datasets but it was always close to the best.

The following tables explain this is in more detail. By the results demonstrated in table 2, on can conclude that the new approach presented in this paper is quite different and interesting. But like all other methods, the proposed method was unable to find the identical alignment (score equals to 1) for some of the test cases (data sets).

Table 1: Summary of the Test Results of Proposed Method.

Name of Datasets	Sequence Number	Sequence Length	With Fitness Value			Corresponding BAliscore
			Best Score	Worst Score	Avg. Score	
1ajsA	18	389	510.15	310.57	385.22	0.588
2pia	16	294	789.95	511.04	574.55	0.873
1wit	20	106	- 248.27	-422.87	-352.45	0.897
1cpt	15	434	-628.67	-985.45	-726.75	0.697
1lvl	23	473	-690.82	-977.15	-856.54	0.886
1aboA	15	80	-493.55	-812.94	-555.42	0.829
4enl	17	440	-186.54	-296.18	-284.94	0.854
1pamA	18	511	89.77	48.47	69.08	0.837
2trx	19	94	556.98	248.45	453.24	0.751
1sbp	16	262	29.64	19.85	36.14	0.859
1havA	16	242	-19.82	-57.67	-39.54	0.802
1uky	23	225	69.24	29.85	37.28	0.746
2hsdA	20	255	-623.45	-985.84	-536.22	0.754
3grs	15	237	163.88	65.52	96.45	0.886
1ubi	19	60	-78.35	-184.87	-25.56	0.833

Table 2: Experimental Results with Reference 2 Datasets of BAliBase 2.0.

Name of Dataset	CLUSTAL X	PRRP	DIALI	RBT-GA	PILEUP8	Proposed Method
1ajsA	0.324	0.227	0.000	**0.892**	0.227	0.588
2pia	0.752	0.767	0.612	0.730	0.766	**0.873**
1wit	0.557	0.76	0.724	0.825	0.476	**0.897**
1cpt	0.66	**0.821**	0.425	0.584	0.688	0.697
1lvl	0.746	0.772	0.783	0.567	0.678	**0.886**
1aboA	0.65	0.256	0.384	0.812	0.000	**0.829**
4enl	0.375	0.668	0.122	0.812	0.224	**0.854**
1pamA	0.761	0.711	0.576	0.66	0.702	**0.837**
2trx	0.87	0.87	0.734	**0.982**	0.87	0.751
1sbp	0.217	0.231	0.043	0.778	0.177	**0.859**
1havA	0.48	0.52	0.000	0.792	0.493	**0.802**
1uky	0.656	0.35	0.216	0.625	0.562	**0.746**
2hsdA	0.484	0.404	0.262	0.745	0.278	**0.754**
3grs	0.192	0.363	0.350	0.755	0.159	**0.886**
1ubi	0.482	0.056	0.000	0.795	0.000	**0.833**
Average score	0.547	0.518	0.348	0.756	0.42	**0.806**

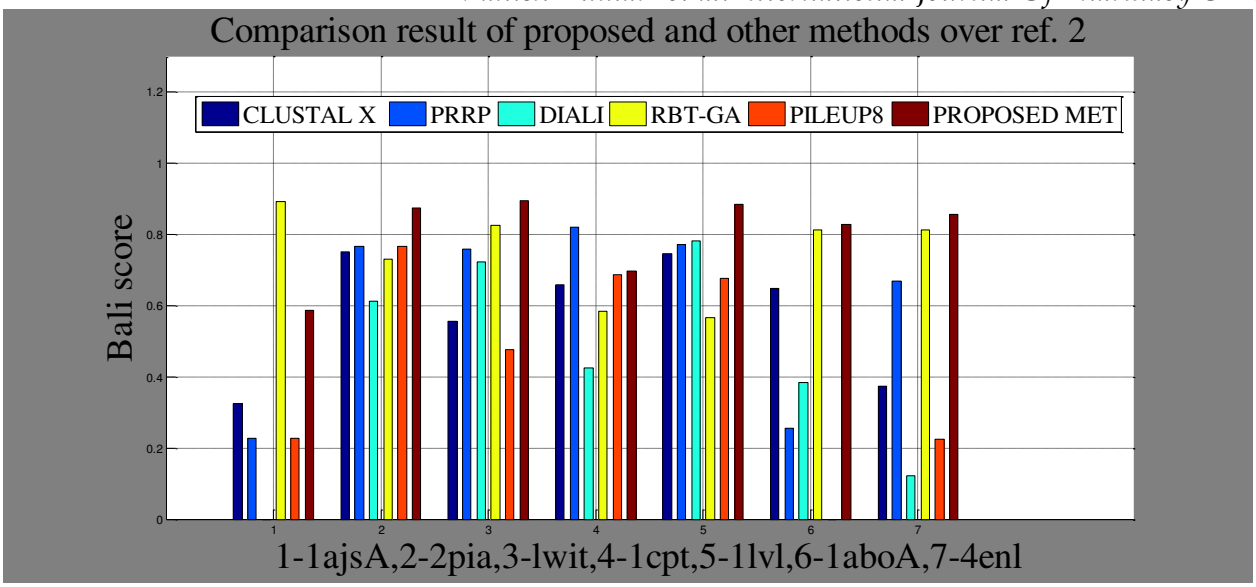

Fig.1. Bar graph comparison result of scores between proposed and other methods over ref.2.

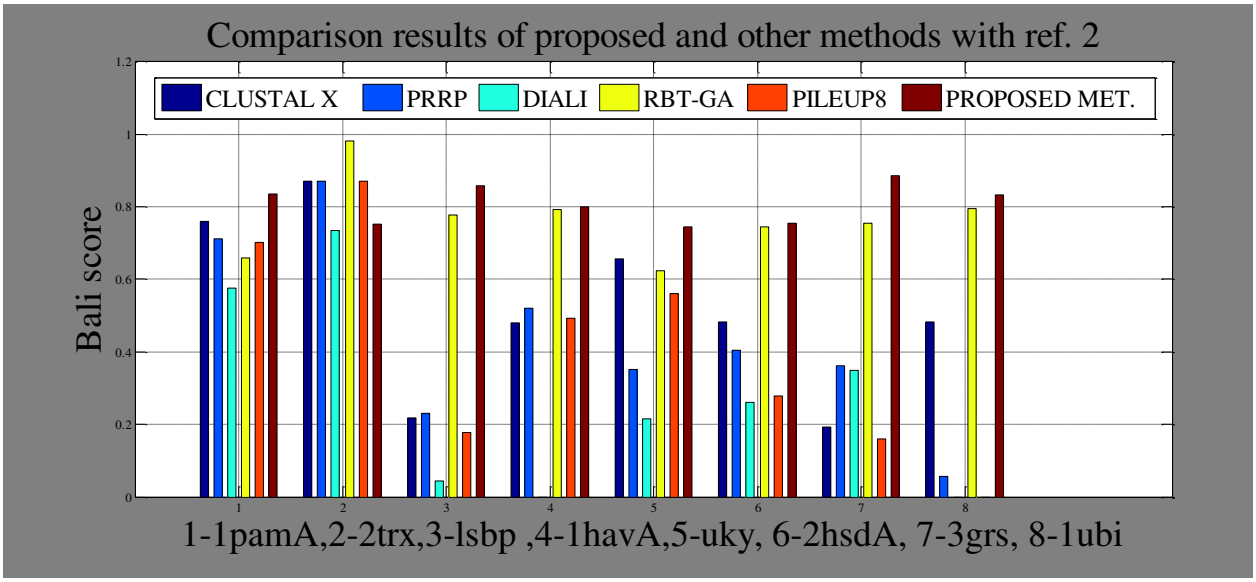

Fig.2. Bar graph comparison result of scores between proposed and other methods over ref.2.

V. Conclusion

Multiple sequence alignment (MSA) is a fundamental and challenging problem in the analysis of biological sequence. The MSA problem is hard to be solved directly, as it always results in exponential complexity with the scale of the problem. Therefore in this study, a new strategy to deal with multiple sequence alignment problems has been proposed. This strategy consists of using a combination of genetic algorithms to solve the given problem. In the proposed approach, the problem is divided into many sub problems and each sub problems are solved individually by applying genetic algorithm and then by replacing the worst individual with the best one in each sub population. In order to evaluate the efficiency and feasibility of the proposed approach, a benchmark datasets from BAliBase 2.0 is considered, because most of the methods discussed in this paper use BAliBase datasets to access the quality of the multiple sequence alignments. Compared to other methods, the proposed approach improves the mathematical and

biological quality for many sequences with different characteristics. The experimental result shows that the proposed method gives a better scope for multiple sequences alignment. During the test analysis, it was found that the solution of the proposed method was not always the best for some test cases but, it was always close to the best. The overall performance of the proposed method outperformed all of the other methods considered in this paper. It can also be concluded that the proposed method performed better than the others because of its proposed genetic operators and parameters.

References

1. L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *J. Comput. Biol.,* volume 1, pages 337-348, 1994.
2. C. Notredame. Recent progresses in MSA a survey. *Pharmaco genomic*, volume 3, pages 1–14, 2002.
3. S.B. Needlman 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*, volume 48, pages 443-453, 1970.
4. D. Feng and R. Doolittle. Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J. Mol. Evol*., volume 25, pages 351-360, 1987.
5. J. Kim , S. Pramanik and M.J. Chung. Multiple sequence alignment using simulated annealing. *Computer applications in bioscience*, volume 10, pages 419-426, 1994.
6. Pengfei Guo; Xuezhi Wang; Yingshi Han, The enhanced genetic algorithms for the optimization design, *3rd International Conference on Biomedical Engineering and Informatics,* Oct. 2010, vol.7, no., pp.2990-2994, 16-18
7. T. Riaz, Y. Wang and K.B. Li. Multiple sequence alignment using Tabu Search. *Proc. 2nd Asia-Pacific Bioinformatics Conference (APBC),* 2004, pages 223-232,
8. Wei-C C; Yu-J C; Chien-C C; Der-T L; Jan-M H. Optimizing a Map Reduce module of preprocessing high-throughput DNA sequencing data, IEEE International Conference on Big Data , 2013 , 6-9.
9. Changjin H; Tewfik, AH. Heuristic Reusable Dynamic Programming: Efficient Updates of Local Sequence Alignment, IEEE/ACM Transactions on Computational Biology and Bioinformatics , 2009, 6:4, 570-82.
10. Ankit A and Huang X., Pairwise Statistical Significance of Local Sequence Alignment Using Substitution Matrices with Sequence-Pair-Specific Distance, Proc. Int'l Conf. Information Technology, 2008, 94-99.
11. Talukdar; Theo C. G; Vibhu K. K; Decomposition for optimal power flows,1982.

12. Zne-Jung Lee, Chou-Yuan Lee Huei-Lung Yu, Kuan-Hung Liu, and Shun-Feng Su An Intelligent System for Multiple Sequences Alignment.

13. Nanuwa, S.S.; Dziurla, A.; Seker, H., Weighted amino acid composition based on amino acid indices for prediction of protein structural classes*, 9th International Conference on Information Technology and Applications in Biomedicine*, Nov. 2009,pp.1-4, 4-7.

14. 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.

15. Kleywegt GJ, Jones TA, Where freedom is given, liberties are taken, Structure, 1995, 3:535.

16. Nucleic Acids Research [http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid= 29792] visited Aug-2007. 36

Corresponding Author:

Manish Kumar*

Email: *manishkumar@cse.ism.ac.in*