Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231-6345 (Online)
An Open Access, Online International Journal Available at http://www.cibtech.org/jls.htm
2015 Vol. 5 (4) October-December, pp.19-27/Kumar
Research Article

A NEW SEARCH HEURISTIC FOR OPTIMAL ALIGNMENT OF BIOLOGICAL SEQUENCES

***Manish Kumar**
Department of CSE, Indian School of Mines, Dhanbad-826004, India
**Author for Correspondence*

ABSTRACT
Multiple sequence alignment (MSA) is an important problem in Bioinformatics since it is often used to identify evolutionary relationships and predict secondary/tertiary structure, among others. The MSA problem is hard to be solved directly, therefore, in this paper a new search heuristic (GA) is proposed for multiple sequence alignment problems of protein sequences. Different genetic operators has been proposed and with the help of these operators, an algorithm has been designed and evaluated with well know existing methods over standard BALiBASE dataset. The main objective of this study is to align multiple protein sequences of different lengths with the help of proposed fitness function and genetic operators. The results of the tests indicate that the accuracy of the provided MSA algorithm is much more than the existing alignment tools such as CLUSTAL X, SAGA, ML-PIMA, SB-PIMA, MULTALIGN, CLUSTAL W, MSA-GA w/prealign and MSA-GA .The experiments and the results presented in the paper clearly reveal the potential capability of the proposed method in optimization processing based on GA.

Keywords: *Bioinformatics; Multiple Sequence Alignment; Genetic Algorithm; Genetic Operators*

INTRODUCTION
A common and cost-effective mechanism to identify the functionalities, structures, or relationships between species is multiple-sequence alignment (MSA) (Hamidi *et al.,* 2013), in which DNA/RNA/protein sequences are arranged and aligned so that similarities between sequences are clustered together. Aligning multiple biological sequences is a key step in elucidating evolutionary relationships, annotating newly sequenced segments, and understanding the relationship between biological sequences and functions. Multiple sequence alignments have wide applicability in many areas of computational biology, including comparative genomics, functional annotation of proteins, gene finding, and modelling evolutionary processes (Auyeung and Melcher, 2005). Because of the computational difficulty of multiple sequence alignment and the availability of numerous tools, it is critical to be able to assess the reliability of multiple alignments and therefore, development of an efficient and accurate method for MSA has become an important research topic.
Sequence alignment is the arrangement of two or more sequences of "residues" that maximizes the similarities between them. If a sequence alignment occurs between two sequences, then it is called a pairwise alignment (Needleman and Wunsch, 1970; Smith and Waterman, 1981) and the main goal is to find the similar or closely related parts between two sequences. If the alignment involves more than two sequences, then it is called a multiple sequence alignment and the main goal is to find the consensus parts among the sequences. For small lengths and small numbers of sequences, it is possible to create the alignment manually. However, efficient algorithms to align such sequences are essential for alignments with more than eight sequences (Thompson *et al.,* 1994).
The MSA algorithms can be classified into three groups: dynamic programming (DP) (Zhimin and Zhong, 2013), progressive methods (Kupis and Mandziuk, 2007), and iterative methods Mohsen *et al.,* (2007). DP optimizes the sum of pairwise alignment scores and its complexity is $O(LN)$, where N is the number of sequences and L is the average length of sequences. DP is not applicable to the alignment of more than twenty sequences because the required time is intolerable. Therefore, to overcome the time and space complexity problem, heuristic methods such as progressive and iterative methods have been proposed.

Progressive is a fast and simple method. It needs N-1 steps to complete the alignment procedure, where N is the number of sequences. In each step, it adds a sequence to the alignment result of the previous step. The order in which sequences are selected to align depends on the tree which is constructed according to the pairwise distance between the sequences. This method is inaccurate and due to its greedy nature, errors occurring in the early steps propagate to the steps following. Iterative methods solve the error propagation problem to some extent. In iterative methods, an initial alignment is made and then it is iteratively improved. Iterative methods are mostly used to improve the progressive method and are applied after the progressive alignment procedure is finished. Actually, they refine the existing alignment. Some iterative methods use a genetic algorithm (Peng *et al.,* 2011) or a simulated annealing (Kirkpatrick *et al.,* 1983) optimization method.

Biological sequences databases are growing exponentially resulting in extensive demands on the implementation of new fast and efficient sequence alignment algorithms. Most of the work in the sequence alignment field has been primarily intended on providing new fast and efficient alignment methods. Smith and Waterman proposed an algorithm to find a pair of segments one from each of two long sequences such that there is no other pair of segments with greater similarity (homology) (Smith and Waterman, 1981). In this local alignment algorithm, similarity measure allowed arbitrary length deletions and insertions. A new algorithm for local alignment of DNA sequences had been proposed by Das and Dey (2004). A dynamic programming algorithm for performing a global alignment of two sequences has been proposed by Needleman and Wunsch (1970). A partitioning approach, based on ant-colony optimization algorithm has been proposed by Pan *et al.,* the approach significantly improves the solution time and quality by utilizing the locality structure of the problem (Pan *et al.,* 2006). Naznin *et al.,* (2009) designed an iterative progressive alignment method for multiple sequence alignment by using new techniques for both generating guide trees for randomly selected sequences as well as for rearranging the sequences in the guide trees. Cai *et al.,* (2000) proposed to combine existing efficient algorithms for near optimal global and local multiple sequence alignment with evolutionary computation techniques to search for better near optimal sequence alignments. Anitha and Poorna (2010) suggested an algorithm for global alignment between two DNA sequences using Boolean algebra and compare the performance of the algorithm with Needleman-Wunsch algorithm. Yue and Tang (2007) applied the divide-and-conquer strategy to align three sequences so as to reduce the memory usage from O (n3) to O (n2). They used dynamic programming so as to guarantee optimal alignment. Nasser *et al.,* (2007) provided a hybrid approach of dynamic programming and fuzzy logic to align multiple sequences progressively. They computed optimal alignment of sub-sequences based on several factors such as quality of bases, length of overlap, gap penalty. Bandyopadhyay *et al.,* proposed direct comparison methods to obtain global and local alignment between the two sequences; the method proposed an alternate scoring scheme based on fuzzy concept (Chang *et al.,* 2006). Chang *et al.,* (2006) established fuzzy PAM matrix using fuzzy logic and then estimated score for fitness function of genetic algorithm using fuzzy arithmetic (Pengfei *et al.,* 2010). Their experimental results evidenced fuzzy logic useful in dealing with the uncertainties problem, and can be applied to protein sequence alignment successfully.

The sole aim of the researchers has been to develop efficient alignment algorithms based on different and latest techniques. In this paper, a novel approach to solve MSA problem of protein sequences using genetic algorithm has been proposed. A genetic algorithm (GA) may be described as a mechanism that mimics the genetic evolution of a species (Peng *et al.,* 2011). It is a non-analytical optimization technique that can give solutions to hard optimization problems that traditional techniques fail to solve. It is based on a simulated evolution, where processes such as crossover, mutation and survival of the fittest help to "evolve" good solutions to a given problem (Bandyopadhyay *et al.,* 2005).

In this study, with the help of genetic operators a new mechanism is proposed and compared with well know existing methods in order to know the quality of the sequences aligned using standards BALiBASE datasets (Berman *et al.,* 2000). At first, an initial generation consisting of individuals (chromosomes) is initialized using random gap insertion at different locations of the sequences. Thereafter, with the help of modified genetic operators such as the crossover and mutations an effort has been made to obtain a

reliable alignment of multiple protein sequences. Using BALiBASE dataset and PAM 250 matrix (Dayhoff *et al.,* 1978), a score comparison is shown by which one can determine the quality and efficiency of the proposed method.

The rest of the paper is organised as follow. In the next section the proposed approach is discussed, followed by the section which explains the experiments performed in order to validate and observe the test results. Finally, the concluding section presents the final considerations.

MATERIALS AND METHODS

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..... 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 (Lee *et al.,* 2005).

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.

Fitness evaluation: In order to evaluate the fitness of the sequence alignment, the Sum of pair method (SPM) is used in this paper. *Sum of Pair Method (SPM)* By using SPM, the fitness of a multiple sequence alignment can be determined by using equation (1a) and (1b). In equation (1a), *S* is the cost of the multiple alignment. *L* is the length (columns) of alignment, S_l is the cost of the l^{th} column of *L* length. *N* is the number of sequences, A_i (A_j) the aligned sequence *i* (*j*) and *cost*(A_i ,A_j) is the alignment score between the two aligned sequences A_i and A_j. When $A_i \neq$ '-' and $A_j \neq$ '-' then *cost* (A_i, A_j) is determined from the PAM 250 matrix , a mutation probability matrix. The cost function includes the sum of the substitution costs of the insertion/deletions using a model with affine gap penalties as shown in (1b). Where, *G* is the gap penalty, *g* is the cost of opening a gap, *x* is the cost of extending the gap by one and *n* is the length of the gap. By this way, the fitness of a multiple sequence alignment is calculated. The complexity of this function is *O(N2L)*.

$S = \sum_{l=1}^{L} s_l$ where $s_l = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} cost(A_i A_j)$ (1a)

$G = g + nx$ (1b)

The score is calculated by scoring all the pair wise comparison between each residue in each column of an alignment and adding the scores together.

This score will act as a measure to evaluate fitness of the population at each generation. Score for each column for the given sequences is calculated as per the data available in the PAM 250 Matrix (Pervez *et al.*, 2014).

Crossover: Crossover is conducted on the strings in the mating pool. A simple crossover operation involves randomly selecting two strings, denoted as A_1 and A_2, randomly selecting a cutting position in each string, then cutting each string into two substrings (head and tail) at the cutting position and finally

creating two new strings by exchanging the heads of A_1 and A_2. The frequency of the crossover operation is controlled by a crossover probability (Gen and Cheng, 1970).

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.

Proposed Algorithm

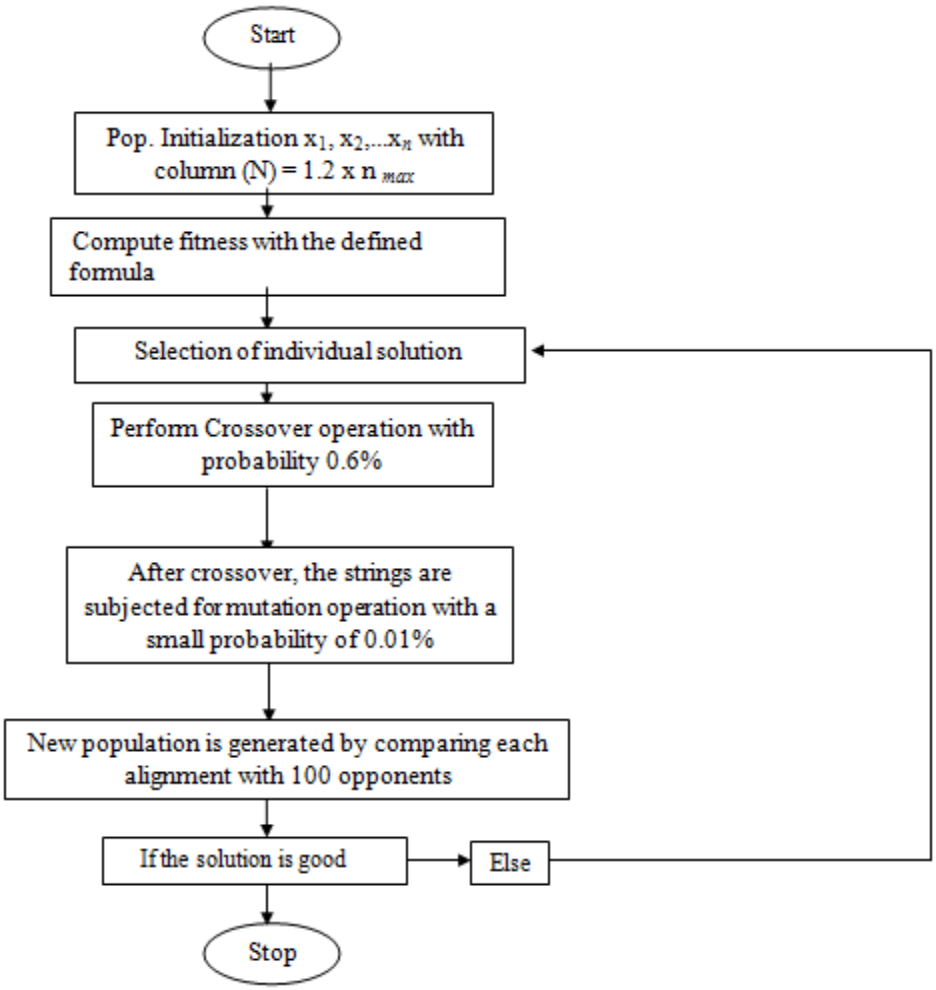

Figure 1: Proposed Approach

Mutation: After crossover, the strings are subjected to mutation. Mutation is the alteration of string elements. In a mutation process, a string is randomly selected from the mating pool and a mutation position in it is selected. The element at the mutation position may be changed or eliminated. The frequency of the mutation operation is controlled by a mutation probability, which is 0.01% for this study (Otman and Jaafar, 2012). The mutation operator 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.

New Generation: In this paper, tournament selection Hong *et al.*, (2007) is implemented for selection operator. This selection scheme is to determine which alignments in the selection pool are to become parents for the next generation in the algorithm. In the selection process, each alignment is compared with 100 opponents that are randomly selected from the selection pool. For each comparison, in which the fitness of the alignment is equal to or higher than that of the opponent, the alignment receives a win. The alignments with the highest number of wins are selected to be the parent alignments for the next generation.

Termination Condition: In this proposed work, the best solution score is recorded and the experiment is made to terminate after 100 generations. As, in the experimental analysis it has been observed that, there is a slight change in the best solution after 100 generations.

RESULTS AND DISCUSSION

For comparing the presented work to some other MSA approaches, the BAliBASE 2.0 is used. BAliBASE 2.0 is the most commonly employed protein alignment benchmark. BAliBase version 2.0 contains eight reference sets. Each reference has a variety of alignment problems. Reference 1 contains small numbers of equidistant sequences. The orphan or unrelated sequences are considered in reference 2.

The main objective of the work demonstrated in this paper, is to observe the role of different genetic operators in solving MSA problem of protein sequences. Here, crossover and mutation operator is proposed which is slightly different than the traditional genetic algorithm. These operators are exclusively used in performing the experiment. The population is initialized randomly by placing gaps (-) at required position. After that, with the help of proposed fitness function scores for every column in the sequence is calculated and then the sequences (individuals) are subjected for crossover and mutation operations with a probability of 0.6% & 0.01% respectively. After the genetic operations are performed, again the fitness scores of each sequences is calculated and based on this score the individuals of poor quality are discarded (which having the lower scores) and the individuals having the higher score goes for the next generation as parents.

With the above statement, we can conclude that the crossover and mutation operator discussed earlier in this paper are playing an important role for the alignment of protein sequences. This can be observed by the results obtained, which clearly indicates a better outcome as compared to other methods. The selection of genetic operators with their probability of been considered in the experiment is also very important. The case discussed here allows only 0.6% of crossover probability and a very small 0.01% of mutation probability. This probability factors of genetic operators has resulted an optimal result in the experimental analysis and a change in this may vary the outcome of the results, which is observed during the experimental analysis.

In the proposed algorithm, 100 independent runs have been considered for each dataset and the scores of each run are recorded in accordance with the BAliscore. BAliscore scores a solution (multiple sequence alignment) in the range of 0.0 and 1.0. If the solution is identical with the corresponding manually created reference alignment then the score is 1.0. A score of 0.0 gives the indication that nothing matching with the reference alignment. The score between 0 and 1 indicates that some part matches with the reference alignment. Here, quality of an aligned sequence is judged by the scores it obtains after successfully aligning.

In order to test the feasibility of the proposed approach, the population size was established to 1000 individuals and the maximum number of generations (iteration) was 100 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. In this study, the experiments for the proposed approach have been performed using genetic algorithm with C programming on an Intel Core 2 Duo processor having 2.53 GHz CPU with 2 GB RAM running on the Linux platform.

Performance of the Proposed Method with ref. 2: The 8 datasets of reference 2 shown in table 1 are of different lengths and sequences. In order to compare the proposed method with respect to BAliscore, the proposed approach was compared with that of CLUSTAL X, SAGA, ML-PIMA, SB-PIMA and

MULTIALIGN. From comparison it can be seen that out of 8 test cases, the proposed method has successfully overcome other methods solutions in 7 test cases and in one test case, the proposed method solution were close to the best.

Table 1: Experimental results with reference 2 datasets of BAliBase 2.0

Name of Dataset		Sequence Number	Sequence Length	CLUSTAL X	SAGA	ML-PIMA	SB-PIMA	MULTALIGN	Proposed Method
	1idy	19	60	0.515	0.548	0.000	0.000	0.401	**0.578**
	1csy	19	99	0.154	0.154	0.000	0.000	0.154	**0.257**
	1r69	20	76	0.675	0.475	0.675	0.675	0.675	**0.851**
Ref.	1tvxA	16	69	**0.552**	0.448	0.241	0.241	0.138	0.397
2	1tgxA	19	71	0.727	0.773	0.543	0.678	0.696	**0.892**
	Kinase	18	287	0.848	0.867	0.651	0.755	0.83	**0.926**
	1ped	18	388	0.834	0.835	0.647	0.651	0.741	**0.856**
	2myr	17	482	0.904	0.825	0.75	0.727	0.894	**0.927**
Average Score				0.651	0.615	0.438	0.465	0.566	**0.710**

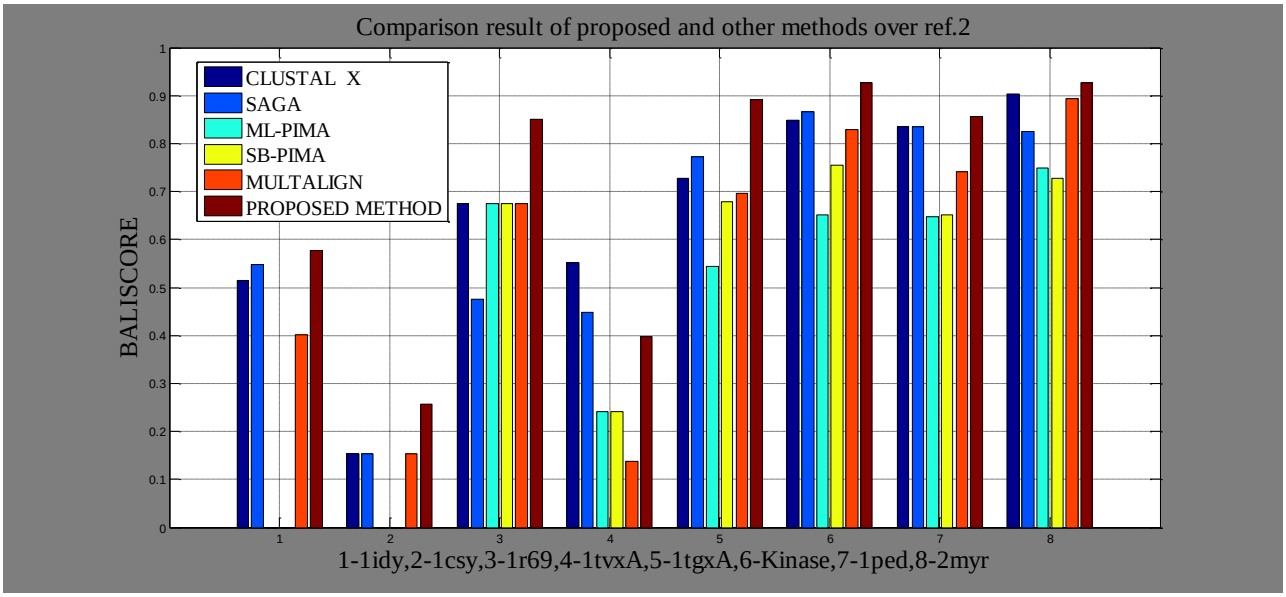

Figure 2: Bar graph comparison result of scores between proposed and other methods over ref.2

Performance of the Proposed Method with ref. 4 & 5: In this experimental study, two test cases each from ref. 4 & 5 were considered; again for all the test cases the proposed method outperformed the other methods. Only, for kinase2 dataset in ref. 4 the proposed method was not able to produce the optimal result.

Table 2: Experimental results with reference 4 & 5 datasets of BAliBase 2.0

Name of Dataset		Sequence Number	Sequence Length	CLUSTAL W	SAGA	MSA-GA w/prealign	MSA-GA	Proposed Method
Ref. 4	1dynA	6	848	0.000	0.000	0.034	0.038	**0.075**
	Kinase2	7	468	0.630	0.364	0.635	**0.71**	0.257
Ref. 5	2cba	8	328	0.628	0.767	0.621	0.422	**0.784**
	S51	15	301	0.75	0.831	0.73	0.528	**0.926**
Average Score				0.502	0.490	0.505	0.424	**0.510**

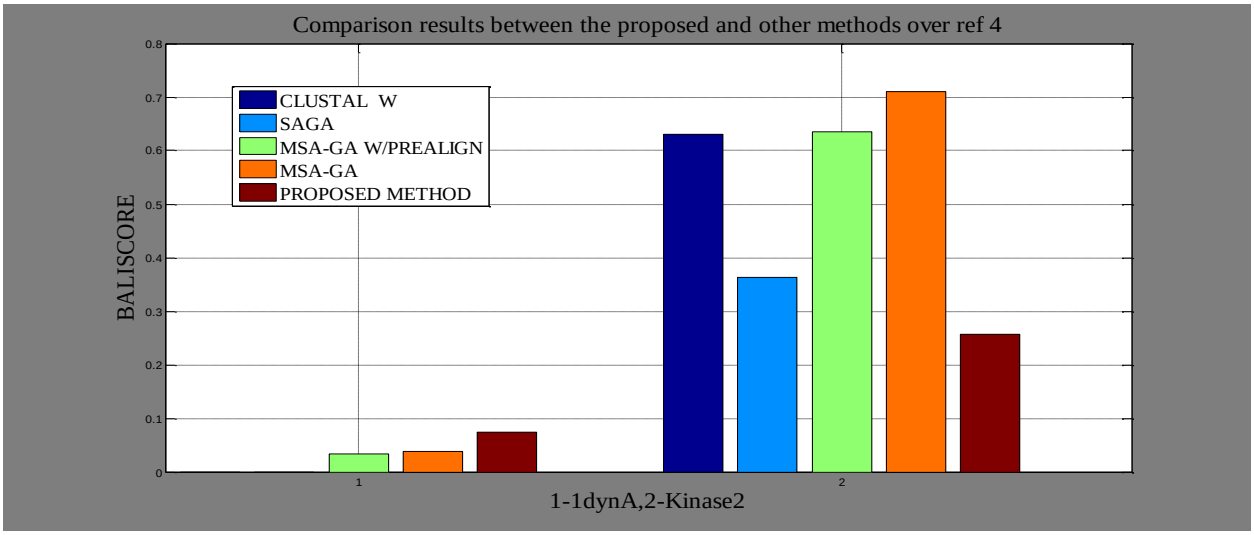

Figure 3: Bar graph comparison result of scores between proposed and other methods over ref.4

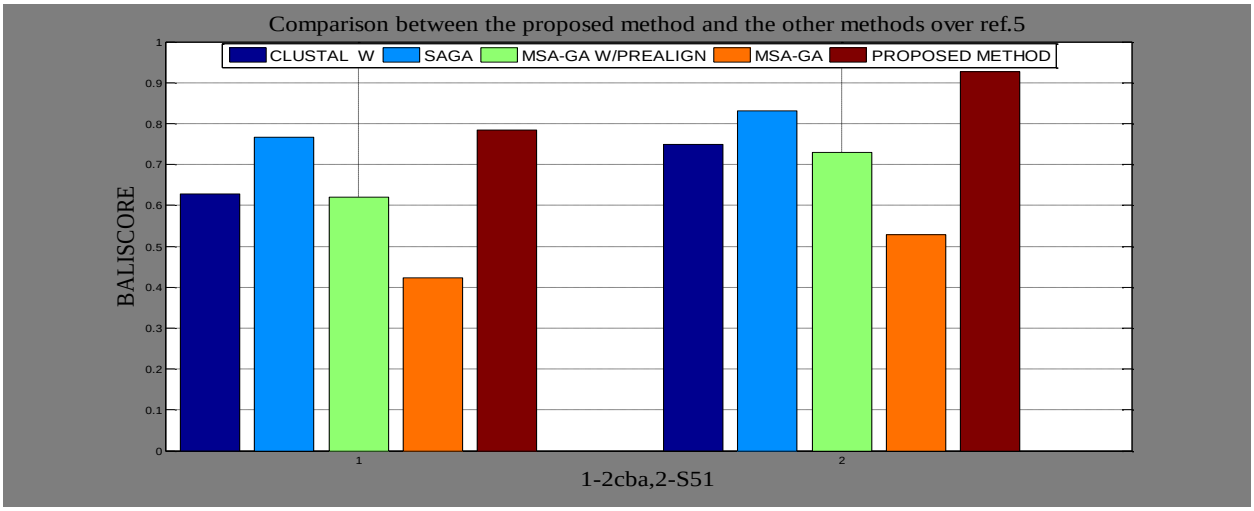

Figure 4: Bar graph comparison result of scores between proposed and other methods over ref.5

Conclusion

Multiple sequence alignment is a computationally hard optimization problem which involves the consideration of different possible alignments in order to find an optimal one, given a measure of goodness of alignments. Genetic Algorithms is an optimization technique that is effective for this type of problems. Therefore, the research reported in the paper is aimed at developing a new technique for efficient multiple sequence alignment using genetic algorithm. In this study, the effects of the genetic operators, which are used in multiple sequence alignment problem, to find the biologically meaningful alignment has been investigated. It has been observed in the experimental analysis that the genetic operators discussed in this paper plays an important role in getting the sequences successfully aligned, which is observed by the fitness score of each sequences. The alignments are performed considering the standard BALiBASE datasets and were compared with well known existing methods. Extensive experiments were done to evaluate performance and scalability of the method. Results show that the proposed method is efficient and offers a real advantage for large-scale multiple protein sequence alignment.

REFERENCES

Anitha V and Poorna B (2010). DNA Sequence Matching using Boolean Algebra. *International Conference on Advances in Computer Engineering* 212-216.

Auyeung A and Melcher U (2005). Evaluations of protein sequence alignments using structural information. *International Conference on Information Technology: Coding and Computing* **2** 748-49.

Bandyopadhyay SS, Paul SS and Konar A (2005). Improved Algorithms for DNA Sequence Alignment and Revision of Scoring Matrix. *Proceedings of International Conference on Intelligent Sensing and Information Processing* 485-490.

Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov I and Bourne P (2000). The protein data bank. *Nucleic Acids Research* **28**(1) 235–42.

Cai L, Juedes D and Liakhovitch E (2000). Evolutionary computation techniques for multiple sequence alignment. *Proceedings of the 2000 Congress on Evolutionary Computation* 829-835.

Dayhoff MO, Schwartz RM and Orcutt BC (1978). A model of evolutionary change in proteins. *Atlas Protein Sequence Structure* **5**(3) 345–351.

Farhana N, Ruhul S and Daryl E (2009). Iterative Progressive Alignment Method (IPAM) for Multiple Sequence Alignment. *Computers & Industrial Engineering* 536-541.

Feng Y and Jijun T (2007). A Divide-and-Conquer Implementation of Three Sequence Alignment and Ancestor Inference. *IEEE International Conference on Bioinformatics and Biomedicine* 143-150.

Gen M and Cheng R (1970). *Genetic Algorithms and Engineering Design* (John Wiley & Sons) New York.

Hamidi S, Naghibzadeh M and Sadri J (2013). Protein multiple sequence alignment based on secondary structure similarity. *International Conference on Advances in Computing, Communications and Informatics* 1224-1229.

Hong Y, Kwong S, Ren Q and Wang X (2007). A comprehensive comparison between real population based tournament selection and virtual population based tournament selection. *IEEE Congress on Evolutionary Computation* 445-452.

Kirkpatrick S, Gelatt JCD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220** 671–80.

Kupis P and Mandziuk J (2007). Evolutionary-Progressive Method for Multiple Sequence Alignment. *IEEE Symposium on Computational Intelligence and Bioinformatics and Computational Biology* 291-97.

Mohsen B, Balaji P, Devavrat S and Mayank S (2007). Iterative Scheduling Algorithms. *IEEE INFOCOM Proceedings.*

Muhammad Tariq P, Masroor Ellahi B, Asif N, Muhammad A, Ali Raza A, Naeem A, Tanveer H, Nasir N, Salman Q, Usman W and Muhammad S (2014). Evaluating the Accuracy and Efficiency of Multiple Sequence Alignment Methods**.** *Evolutionary Bioinformatics* 205–217.

Needleman SB and Wunsch CD (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48**(3) 443–453.

Needleman SB and Wunsch CD (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology* **48** 443-453.

Otman A and Jaafar A (2012). Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem. *International Journal of Emerging Sciences* **2**(1) 61-77.

Pan Y, Chen Y, Juan C, Wei L and Ling C (2006). Partitioned optimization algorithms for multiple sequence alignment. *Proceedings of the 20th International Conference on Advanced Information Networking and Applications* **5**.

Peng Y, Dong C and Zheng H (2011). Research on Genetic Algorithm Based on Pyramid Model. *2nd International Symposium on Intelligence Information Processing and Trusted Computing* 83-86.

Pengfei G, Xuezhi W and Yingshi H (2010). The enhanced genetic algorithms for the optimization design. *3rd International Conference on Biomedical Engineering and Informatics* **7** 2990-2994.

Pin-Teng C, Lung-Ting H, Kuo-Ping L, Chih-sheng L and Kuo-Chen H (2006). Protein Sequence Alignment Based on Fuzzy Arithmetic and Genetic Algorithm. *IEEE International Conference on Fuzzy Systems* 1362-1367.

Sara N, Gregory LV, Monica N and Alison M (2007). Multiple Sequence Alignment using Fuzzy Logic. *Proceedings of the IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology* 304-311.

Smith TF and Waterman MS (1981). Identification of common molecular subsequences. *Journal of Molecular Biology* **147**(1) 195–197.

Smith TF and Waterman MS (1981). Identification of common molecular subsequence. *Journal of Molecular Biology* **147** 195-197.

Swagatam D and Debangshu D (2004). A new algorithm for local alignment in DNA sequencing. *Proceedings of IEEE Conference on INDICON* 410-413.

Thompson JD, Higgins DG and Gibson TJ (1994). CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research* **22**(22) 4673–4680.

Zhimin ZH and Zhong CW (2013). Dynamic Programming for Protein Sequence Alignment *International Journal of BioScience and Bio Technology* **5**(2).

Zne-Jung L, Chou-Yuan L, Huei-Lung Y, Kuan-Hung L and Shun-Feng S (2005). An Intelligent System for Multiple Sequences Alignment. *International Conference on Systems, Man and Cybernetics* (IEEE) **2** 1042-1047.