Int J Pharm Bio Sci 2015 April; 6(2): (B) 386 - 398

**Research Article** **Molecular biology**

**International Journal of Pharma and Bio Sciences** **ISSN 0975-6299**

# THE IMPACT OF GENETIC OPERATORS IN SOLVING MULTIPLE PROTEIN SEQUENCE ALIGNMENT

**MANISH KUMAR***

*Department of Computer Science and Engineering,*
*Indian School of Mines, Dhanbad, Jharkhand, INDIA.*

## ABSTRACT

Protein multiple sequence alignment is a fundamental task in bioinformatics. Its importance is due to its application in the estimation of the phylogeny tree, prediction of the secondary and tertiary protein structure. Obtaining the best solution in multiple alignment (MSA) is an NP-hard problem. This research work presents a novel and efficient approach for MSA of protein sequences by improving the genetic operators of Genetic Algorithm. We asses our algorithm on different protein benchmarks, e.g., BALIBASE, Swiss Prot and we have compared the obtained results to those obtained with other alignment algorithms, e.g., CLUSTALW, using the Column Score (CS) and Fitness Score. Experimental results show that the proposed one is indeed a better algorithm that can increase solution quality, reduce running time and can increase numbers of match column.

**KEYWORDS*:* Multiple Sequence Alignment, Genetic Algorithms, Computational Biology, Protein Sequences.**

**MANISH KUMAR**
**Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, Jharkhand, INDIA.**

***Corresponding author**

# INTRODUCTION

Multiple sequence alignment (MSA), the simultaneous alignment among three or more nucleotide or amino acid sequences, is one of the most essential tools in molecular biology. Sequence alignments are used to help in finding homology between new and existing sequences, to suggest primers for polymerase chain reaction, and to predict the secondary or tertiary structure of RNA and Proteins. Therefore, the development of efficient and accurate automatic methods for multiple sequence alignments is a very important research topic. Multiple sequence alignment is a procedure in which all sequences are made equal in length by inserting gaps among them, with similar residues aligned in the same columns. Two important properties of an MSA method are its accuracy and speed. The MSA algorithms can be classified into three groups: dynamic programming (DP) [1], progressive methods [2], and iterative methods [3]. 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.Multiple sequence alignment has many applications. With respect to DNA sequences it is used to find the evolutionary relationship among the sequences that is, for phylogenetic tree construction or to establish homology between sequences. As a result homologous genes can be studied and new ortholog or paralog genes can be discovered. And in the case of protein sequences where it finds wide usage, it is used to find the structural and functional similarity among the proteins being aligned.Thus, the 3D structure of proteins can be predicted using MSA[4]. Apart from the above major applications it is also used in other important areas such as biological modeling, profiling, PCR (Polymerase Chain Reaction) primer design and data validation [5]. The motive behind MSA is to find motifs that occur in many sequences and can explain the functional similarity or mutations that cause a change in the function. Multiple sequence alignment like pairwise alignment results in many possible alignments. The optimal alignment or the accurate alignment depends on the application. Structural similarity exists if alignment of two residues in protein sequences, indicates a similar function in their 3D structure. The alignment of similar residues across multiple sequences may indicate a common ancestor if phylogeny is the criterion. In other words, after the MSA, residues that appear in the same column, which are usually called equivalent residues may be homologous or play a similar structural or functional role. The accuracy of multiple sequence alignment still remains an open problem. To align multiple sequences within limited resources, a number of methods have been developed. Many of them use iterative stochastic approaches especially genetic algorithms (GAs). They have been successfully applied to find good alignments for DNA, RNA and Protein sequences, for detail see [6,7,8,9,10,11]. In general, heuristics offer more practical solutions but usually produce quasi-optimal alignment. The progressive approach is the most commonly used heuristic method by gradually aligning the closest pair to build MSA[12,13,14,15]. Its main disadvantage is the alignment solution may be trapped in local optima, which stems from the greedy nature of this algorithm. This means that if mistakes are made in intermediate alignments, they cannot be corrected later when more sequences are added into the alignment process. Another approach is to use an extension of dynamic programming for simultaneously aligning multiple sequences, such as in Carrillo-Lipman algorithm [16], MSA [17], DGA [18], etc. In general, these algorithms often have higher quality solutions than the progressive approach. However, the drawbacks of these algorithms are the complexity in running time and memory requirements. Literature studies [19,20,21,22] says that there are still a number of challenges in aligning protein sequences. First, the locally conserved regions, that reflect functional specificities or that modulate a protein's function in a given cellular context , are less well aligned. Second, motifs in natively

disordered regions are often misaligned. Third, the badly predicted or fragmentary protein sequences, which make up a large proportion of today's databases, lead to a significant number of alignment errors [23].Although DP is the first recognized approach in optimally solving both longest common subsequence (LCS) [24] and MSA [25] problems, progressive is the most widely used approach in protein multiple sequence alignment. One of the most common tools in this field is ClustalW [26]. It represents a complicated scoring method in which the gap opening and extension parameters are adjusted due to sequence length, hydrophilic regions and so on. Most of the progressive tools use the Sum-of-Pairs (SP) scoring method such as Mafft [27]. T-Coffee [28] employs a consistency-based scoring function and considers both local and global alignment information. It provides significant improvement in accuracy. Muscle [29] uses a logexpectation scoring method. ProbCons [30] utilizes a probabilistic consistency-based scoring method based on the pair hidden Markov model (pair-HMM). Probalign [31] employs the partition function to calculate posterior probability. MSAProbs [32] calculates posterior probabilities with both the pair-HMM and partition function. Recent researches tend to use sequence structure information to improve the alignment result. PROMALS3D [33] and 3DCoffee [34] use tertiary structure information which is not known for all existing sequences.It has been stated in the literature[35,36,37,38,39,40] that none of the existing algorithms were capable of providing accurate alignments for all the test cases. As a consequence, iterative algorithms were developed to construct more reliable multiple alignments, using for example iterative refinement strategies [41], Hidden Markov Models [42], Genetic Algorithms [43] and the clustal W [44]. These methods were shown to be more successful at aligning the most conserved regions for a wide variety of test cases, although some accuracy was lost for distantly related sequences, in the 'twilight zone' of evolutionary relatedness [45].A genetic algorithm (GA) may be described as a mechanism that mimics the genetic evolution of a species [46]. 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 . The main advantage of using genetic algorithms for MSA problem is that there is no need to provide a particular algorithm to solve a given problem. It only needs a fitness function to evaluate the quality of different solutions. Also since it is an implicitly parallel technique, it can be implemented very effectively on powerful parallel computers to solve exceptionally demanding large-scale problems [47].Therefore, in this paper a genetic algorithm based approach has been used to solve the MSA problem of protein sequences. The crossover and mutation operator has been modified (refer proposed approach section V,VI) based on certain parameters to produce new generation. A new methodology for the coming generations has been proposed so as to produce the best results for all the coming generations. Later, we have defined a scoring function to calculate the fitness function and match column by inserting GAP between the sequences and calculation the total result by which we can evaluate our proposed method. All the evaluation were made on the standard BAliBASE [48]and SwissProt database [49]and compared the proposed approach with well know Clustal W method.To obtain the best possible alignment, gaps are introduced in the sequence and a scheme for penalizing these gaps must be adopted. Gaps are inserted between the residues so that identical or similar characters are aligned in successive columns.The values of gap penalties depend on the choice of matrix and must balance their values . A high gap penalty in relation to the values in the matrix will impede the appearance of gaps. On the other extreme, a too low gap penalty will cause gaps to appear everywhere in the alignmentThere are various other methods presented in the past for MSA problem such as the Clustal W[50] . It is one of the most popular progressive alignment system. Since progressive alignment is a heuristic algorithm, Clustal W is not guaranteed to find optimal alignments. Clustal W exploits the fact that homologous sequences are evolutionarily related. It builds up multiple alignments progressively with a series of

pairwise alignments, moving from the leaves upward in a guide tree that estimates the phylogeny of the sequences . Although Clustal W doesn't always find optimal alignments, in most cases those alignments give a good starting point for further automatic or manual refinement . This type of alignment is generally useful for the study of identifying regions that are highly conserved. The alignment can be further improved through sequence weighting, position-specific gap penalties and choice of weight matrix[51]. The rest of the paper is organised as follow. In the next Section we introduce concepts underlying our research work , followed by the section which explains the experiments performed in order to validate and observe our test results . Finally, the concluding Section presents final considerations.

## *PROPOSED APPROACH*

This section provides a brief idea about the representation of the sequences, problem description ,the scoring function and the role of crossover and mutation operator in the study.

### *(i) Datasets*

Two different types of datasets were used for the experimental study. The first dataset of five protein families with approximately the same length and with various numbers of sequences within each family was selected from the sequence database SwissProt [49]. The second dataset of five references was obtained from the alignment database BAliBASE [48] . Using these two datasets, results from our refined method and ClustalW were compared.

### *(ii) Representation*

In this sub section the representation of the initial generation has been presented.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 $\chi$ and the parameter $\alpha$ is a scaling factor. In this study, each matrix candidate may have different number of columns and the value $\alpha = 1.2$ 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 [52].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.

### *(iii) Scoring Function*

In this sub section, a formal definition of the sum-of-pairs of multiple sequence alignment is introduced and calculated for the experiment, which is used as a tool to calculate fitness.Sum of- pairs of multiple sequence alignment is defined as the problem of finding the alignment that has the maximum sum-of-pairs cost. Consider a family $S = (S_l, , S_k)$ of k sequences with various lengths. Each sequence element represents a character from a given alphabet set A. For DNA (RNA) sequences, the alphabet set A consists of 4 characters of nucleotides, A = {A, T (U), C, G}. For protein sequences, the alphabet set A consists of 20 characters of amino acids, A = {C, T, S, P, A, G, N, D, E, Q, H, R, K, M, I, L,

V, F, Y, W}. 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 250matrix(http://prowl.rockefeller.edu/aainfo/pam250.html.), 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).*

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

$$G = g + nx \quad \text{.....(1b)}$$

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.

***(iv)Fitness using scoring value***

For simplicity, the fitness function is chosen the same as the objective function defined in sub section

***(v) Crossover operation***

In our experiment, for the crossover operation[53] the system randomly generates a real value between zero and one and then compares it with a predefined crossover rate *CR*, where *CR*ϵ [0,1]. If the generated random number is smaller than or equal to the value of CR, then the crossover operation is performed, the genes before a randomly generated crossover point of the two chromosomes will be exchanged, and then the offspring are put into a mating pool. Otherwise, the parents of chromosomes will be put into the mating pool, where the size of the mating pool is the same as the population size. The crossover operation will be performed repeatedly until the mating pool is full. An example of the crossover operation is shown in Figure 1.

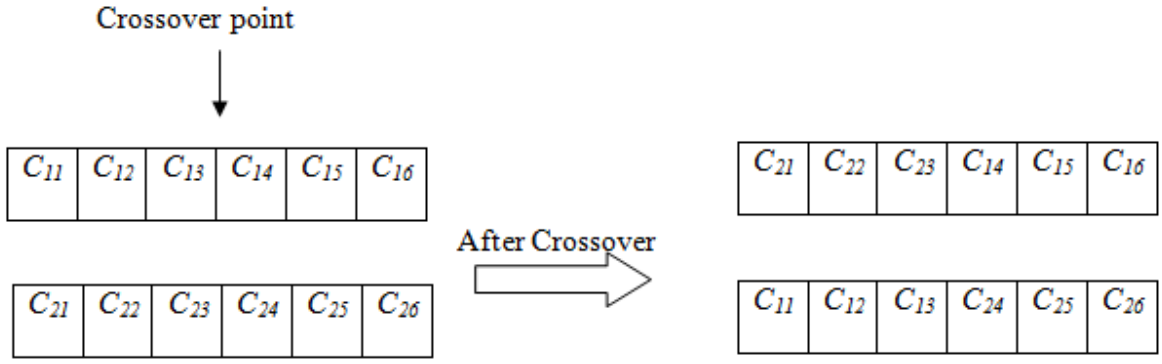


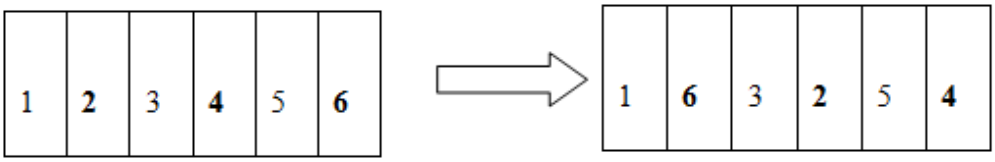

**Figure 1**
***The crossover operation.***

***(vi) Mutation Operation***

Here, an explanation of the mutation operation performed in the experiment is detailed.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[. We assume that the probability of the mutation in this work is 0.01%. With 1000 genes positions we should expect 1000 x 0.001 = 1 genes to undergo mutation for this probability value.Three genes are chosen randomly which shall take the different positions not necessarily successive i < j < l. The gene of the position i becomes in the position j and the one who was at this position will take the position l and the gene that has held this position takes the position i.

**Table 1**

***Example of Position Mutation operator***




*Parent* *child*

***(vii) New Generation***

To form the new generation, a selection strategy has been used where multiple sequence alignment from the parent and the child generations compete based on their objective fitness scores. Here, the best 50% of the parent population and the best 50% of the children population are merged together while ensuring that there is no duplication of individuals. Other approaches such as 70-30 % and 30-70% or the 60-40% and 40-60 % parent- child population has also been tested but, these approaches has not shown any effective improvement in quality of solution and hence not been considered. This new generation (50% of parent and 50% of child ) is now been considered as a parent population to continue the evolution process of the proposed approach.

***(viii) Termination Condition***

The termination conditions used for the experiment are as follows: (1) The number of generations exceeds the maximum number of generations ($G_{max}$) permitted, which is 200 for the experiment. (2) If the best fitness score does not improve over a given number of generations.When the termination conditions are satisfied, the resultant alignment matrix is obtained. The computational complexity of the proposed method was ($N^3$+$L^2$), where *L* is the length of the sequence and *N* is the number of sequences.

### *(ix) Flowchart of the Proposed Approach*

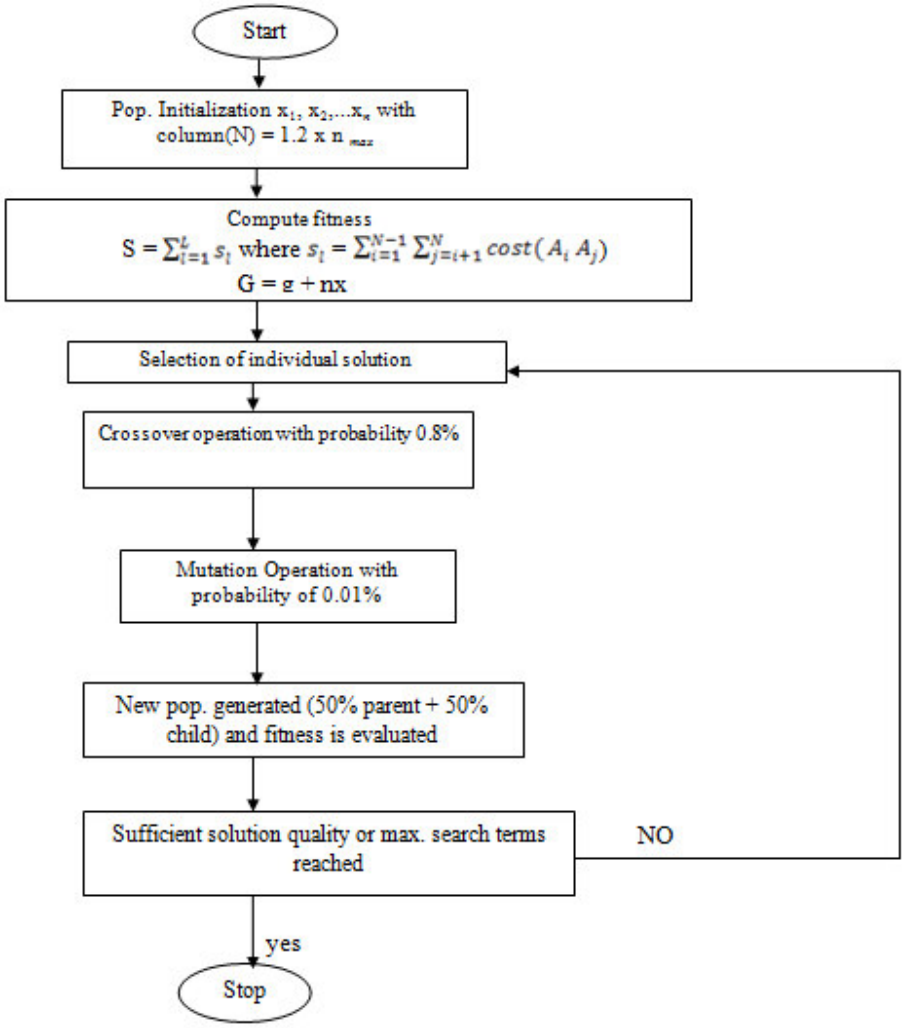



**Figure 2**
***Proposed approach***

## RESULTS AND DISCUSSION

In this section, the experimental methodology followed in this work is detailed. Moreover, results obtained with the proposed method are presented and discussed.The main objective of the research work is to observe the role of proposed genetic algorithm in solving MSA problem of protein sequences in terms of quality and scores of the sequence aligned. We performed extensive experiments on the proposed approach with genetic algorithm . The experiments were performed using C programming on an Intel Core 2 Duo processor with T9400 chipset, 2.53 GHz CPU and 2 GB RAM running on the platform, Microsoft Windows7.To perform a fair comparison with the Clustal W method, the proposed approach uses the same stop condition and population size as configured by Clustal W. The population size was established to 1000 individuals, and the maximum number of generations was 200 with a crossover probability of 0.8%, mutation rate of 0.01% and the tournament size is 2 for the experiment. The scoring matrix and the space score used for the experiment are PAM 250 and -10, respectively, for each Protein sequences. In order to evaluate our proposed

approach, we carried out the experiments for Ten datasets with different lengths from the BAliBASE database and SwissProt database . Availability of literature about performance of other related algorithms on these data sets prompted us to select them for our study. For each of the experiment, alignments were performed with the proposed approach and were compared with Clustal W . Performance, in terms of both efficiency and apparent alignment quality, are summarized for several of our experimental runs. Table 2 and 3 shows the comparison results. It is shown that the proposed algorithm has the better results for most test cases as compared to Clustal W. Bold- faced text indicates the best score among the algorithms.

**Table 2**
***Comparison of the alignment results between proposed method and Clustal W with BAliBASE datasets.***

| Data Set | | Match Column | | Fitness Score | |
|---|---|---|---|---|---|
| Category | Name | Proposed Method | Clustal W | Proposed Method | Clustal W |
| Reference 1 | 1ad2 | **18** | 16 | **2145** | 1288 |
| Reference 4 | 1ycc | **8** | 4 | **2958** | 2390 |
| | Kinase1 | **4** | 1 | **3214** | 3052 |
| Reference 5 | 1thm1 | **8** | 4 | **987** | 806 |
| | S52 | **21** | 19 | **2396** | 2262 |

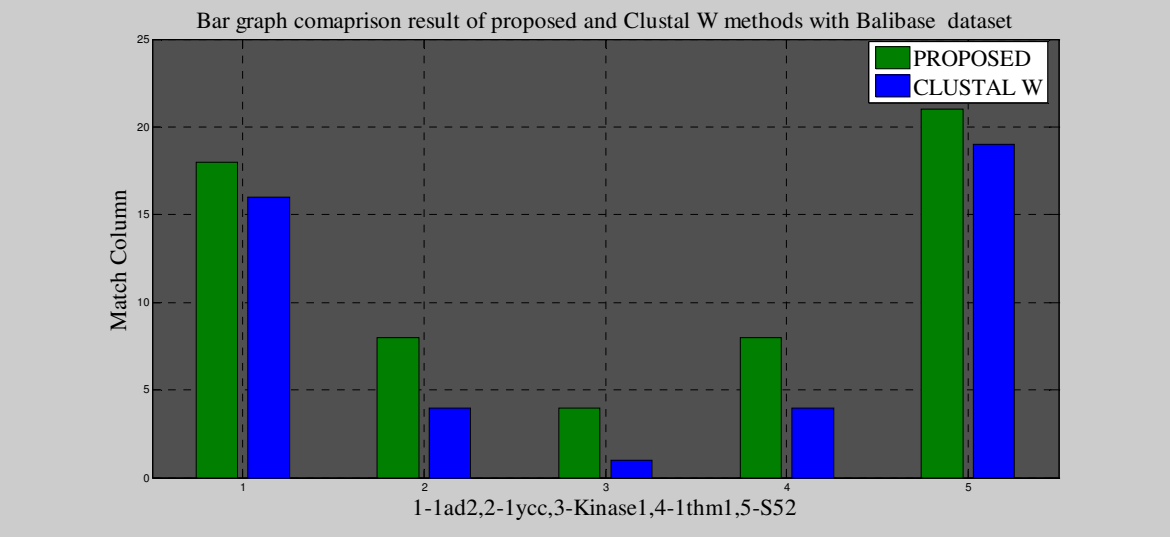



**Figure 3**
***Bar graph comparison of Match column between the proposed and CLUSTAL W method for BALiBASE dataset.***

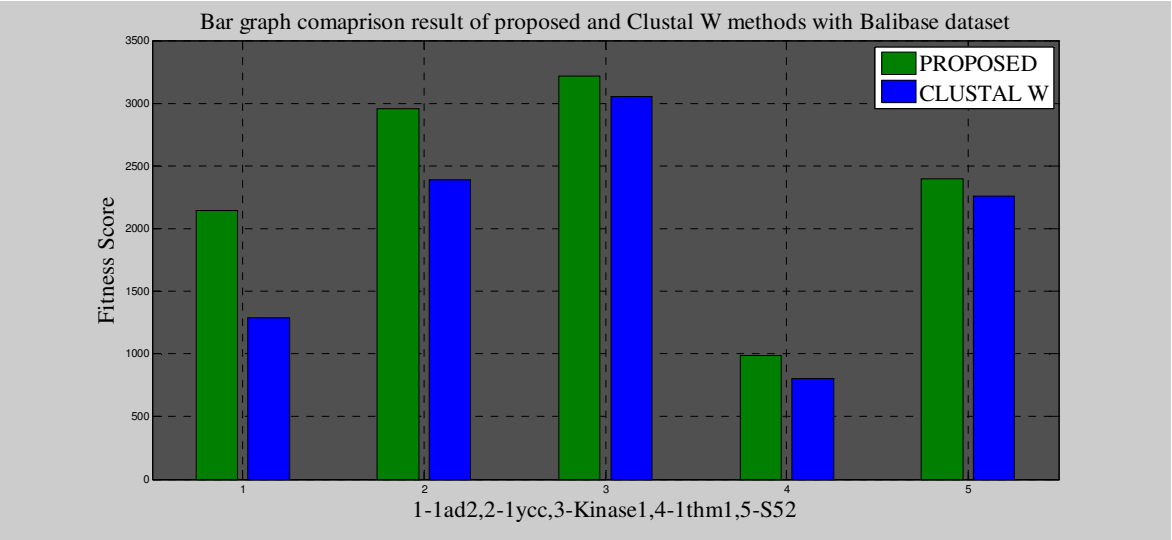



**Figure 4**

***Bar graph comparison of Fitness Score between the proposed and CLUSTAL W method for BALiBASE dataset.***

**Table 3**

***Comparison of the alignment results between the proposed method and Clustal W with SwissProt datasets.***

| Data Set | Family | Match Column | | Fitness Score | |
|---|---|---|---|---|---|
| | | Proposed Method | Clustal W | Proposed Method | Clustal W |
| Tox10 | Toxin | 4 | **7** | 4589 | **4898** |
| flavo4_1 | Flavodoxin | **55** | 50 | **3145** | 2725 |
| tim5 | TIM | **51** | 48 | **4196** | 3967 |
| tryp4 | Trypsin | **57** | 55 | **3195** | 2968 |
| apolipo4 | Apolipo | **45** | 37 | **3665** | 3386 |

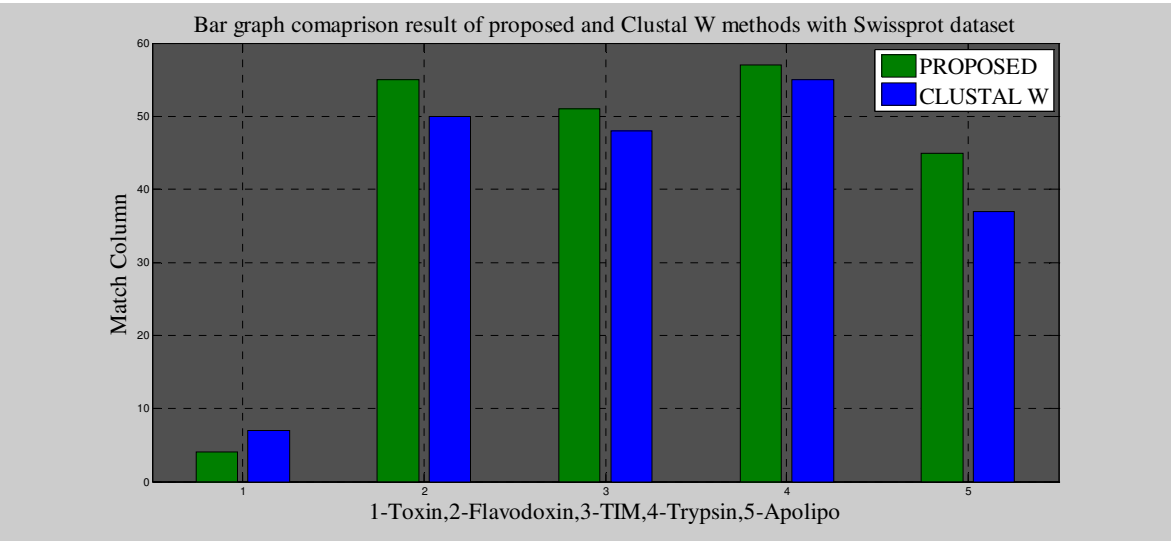



**Figure 5**

***Bar graph comparison of Match Column between the proposed and CLUSTAL W method for SwissProt dataset.***

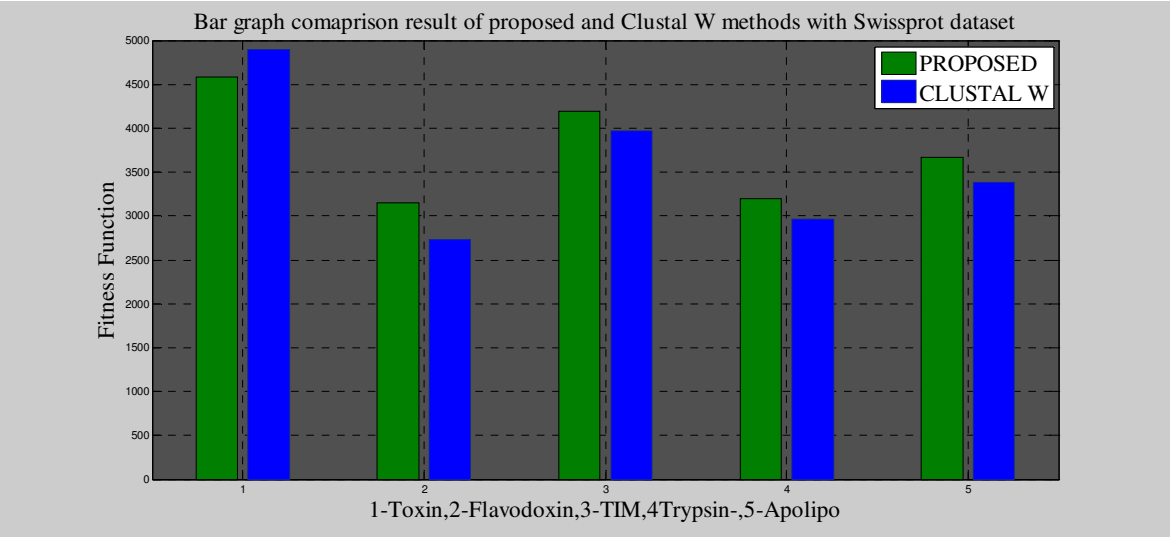



**Figure 6**
***Bar graph comparison of Fitness Score between the proposed and CLUSTAL W method for SwissProt dataset.***

## CONCLUSION

Multiple sequence alignment is very useful in many scientific fields, including biology. However, it belongs to the combinatorial optimization problems with exponential time complexity.Genetic algorithm is a optimization technique that is effective for this type of problems. In this study, we have described the genetic algorithms methodology and we demonstrated how it can be implemented to produce optimal or near-optimal solutions to the MSA problem. In this paper, an attempt has been made to show the influence of genetic operators in construction of an efficient MSA for protein sequences. The objective of this study is to validate the efficacy of the proposed approach and assess it by comparing with other commonly used algorithm for MSA over different datasets. In this work, modification in the genetic operators has been implemented so as to get optimal MSA. We have calculated the match column score and the fitness score with the proposed approach and compared it with the Clustal W method over different datasets. The experimental result shows that the proposed method gives a better scope for multiple sequences alignment, as the results of each alignment tends to improve, which is observed by the increase in fitness score. During the test analysis, it was found that the solution of the proposed method was not always the best for some test case but , it was always close to the best. Finally, other relevant works published in the literature[35, 36,37,38,39,40] were also studied to validate our results. To this respect the conclusion that can be drawn is that the novel approach proposed in this paper obtains very promising protein sequences that significantly surpass previously published Clustal W results in most of the cases.

## REFERENCES

1. 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).
2. D.-F. Feng and R. F. Doolittle, Progressive sequence alignment as a prerequisitetto correct phylogenetic trees, Journal of molecular evolution, vol. 25, pp. 351-360, (1987).

3. G. J. Barton and M. Sternberg, A strategy for the rapid multiple alignment of protein sequences. Confidence levels from tertiary structure comparisons,Journal of molecular biology, vol. 198, p. 327,( 1987).
4. Monica Soni and Arvind Pareek, Homology based 3d- structure modeling of aquaporin-2 protein int j pharm bio sci 5(4): (b) 855 – 862 , (2014).
5. Sun, L.-P.; Wang, S.; Zhang, Z.-W.; Ma, Y.-Y.; Lai, Y.-Q.; Weng, J.; Zhang, Q.-Q., Interaction of gold nano particles with Pfu DNA polymerase and effect on polymerase chain reaction, Nano biotechnology, IET , vol.5, no.1, pp.20-24, March (2011).
6. 1. Randy L. Haupt, Sue Ellen Haupt, Practical Genetic Algorithm 2nd Edn, A John Wiley & Sons, Inc., Publication (2004).
7. O. Abdoun O. and J. Abouchabaka, A Comparative Study of Adaptive Crossover Operators for Genetic Algorithms to Resolve the Traveling Salesman Problem. IJCA, Vol. 31, No. 11, (2011).
8. H. G. Cobb and J. J. Grefenstette. Genetic algorithms for tracking changing environments. In S. Forrest, editor, Proceedings of the Fifth International Conference on Genetic Algorithms pages 523–530, San Mateo, CA, (1993).
9. da Silva, F.J.M.; Sanchez Perez, J.M.; Pulido, J.A.G.; Rodriguez, M.A.V., Optimizing Multiple Sequence Alignment by Improving Mutation Operators of a Genetic Algorithm, Ninth International Conference on Intelligent Systems Design and Applications,, pp.1257-1262, (2009).
10. Ching Zhang and Andrew K. C. Wong, Toward Efficient Multiple Molecular Sequence Alignment: A System of Genetic Algorithm and Dynamic Programming IEEE Transactions on Systems, Man, and Cybernetics—Part b: Cybernetics, vol. 27, no. 6, (1997).
11. Layeb, A.; Meshoul, S.; Batouche, M., Quantum Genetic Algorithm for Multiple RNA Structural Alignment, Second Asia International Conference on Modeling & Simulation , pp.873-878, 13-15( 2008).
12. D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning Reading, MA: Addison-Wesley Publishing Company, (1989).
13. K. Chellapilla and G. B. Fogel, Multiple sequence alignment using evolutionary programming, Proceedings of the 1999 Congress on Evolutionary Computation, Washington DC, USA, pp. 445-452 (1999).
14. Huan Yu; Minghua Deng, ClustalY: speed up the guide tree building for ClustalW, 2005. Proceedings. Eighth International Conference on High-Performance Computing in Asia-Pacific Region, pp.,610, 1-1 (2005).
15. K. Katoh, K. Kuma, H. Toh, and T. Miyata, MAFFT version 5: Improvement in accuracy of multiple sequence alignment, Nucleic Acids Res., vol. 33, pp. 511-518,(2005).
16. Masuno, S, Maruyama, T,Yamaguchi, Y, Konagaya, A, An FPGA Implementation of Multiple Sequence Alignment Based on Carrillo-Lipman Method,International Conference on Field Programmable Logic and Applications, pp.489-492, 27-29 (2007).
17. Church, Philip C, Goscinski, Andrzej, Holt, Kathryn, Inouye, Michael, Ghoting, Amol, Makarychev, K. Reumann, Matthias, Design of multiple sequence alignment algorithms on parallel, distributed memory supercomputers,EMBC, Annual International Conference of the IEEE on Engineering in Medicine and Biology Society, pp.924-927, (2011).
18. Naznin, F, Sarker, R., Essam, D, DGA: Decomposition with genetic algorithm for multiple sequence alignment, IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology, pp.1-8, 2-5 (2010).
19. Chakraborty, G.; Chakraborty, Basabi, Rank and proximity based crossover (RPC) to improve convergence in genetic search, IEEE Congress on Evolutionary

Computation, vol.2, pp.1311-1316 Vol. 2, 2-5 2005.
20. Taylor WR, Protein structure comparison using SAP. Methods Mol Biol 143: 19–32, (2000).
21. A Razmara, J, Deris, S.B,Parvizpour, S., Text-Based Protein Structure Modeling for Structure Comparison, International Conference of Soft Computing and Pattern Recognition , pp.490-496,( 2009).
22. R. Mott, Alignment: Statistical Significance, Encyclopedia of Life Science, John Wiley & Sons, Ltd. (2005)
23. Lo¨ytynoja A, Goldman N Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science 320: 1632–1635, (2008).
24. Rubi, R.D, Arockiam, L, Positional_LCS: A position based algorithm to find Longest Common Subsequence (LCS) in Sequence Database (SDB), IEEE International Conference on Computational Intelligence & Computing Research pp.1-4, 18-20 (2012).
25. Taheri, J Zomaya, A.Y., RBT-Km: K-Means clustering for Multiple Sequence Alignment, IEEE/ACS International Conference on Computer Systems and Applications, pp.1-8, 16-19 (2010).
26. 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).
27. K. Katoh, K. Misawa, K. i. Kuma, and T. Miyata, MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform, Nucleic acids research, vol. 30, pp. 3059-3066, (2002).
28. C. Notredame, D. G. Higgins, and J. Heringa, T-Coffee: A novel method for fast & accurate multiple sequence alignment, Journal of molecular biology, vol. 302, pp. 205-218, (2000).
29. R. C. Edgar, MUSCLE: multiple sequence alignment with high accuracy and high throughput, Nucleic acids research, vol. 32, pp. 1792-1797, (2004).
30. C. B. Do, M. S. Mahabhashyam, M. Brudno, and S. Batzoglou, ProbCons: Probabilistic consistency-based multiple sequence alignment, Genome Research, vol. 15, pp. 330-340, (2005).
31. U. Roshan and D. R. Livesay, Probalign: multiple sequence alignment using partition function posterior probabilities, Bioinformatics, vol. 22, pp. 2715-2721, (2006).
32. Y. Liu, B. Schmidt, and D. L. Maskell, MSAProbs: multiple sequence alignment based on pair hidden Markov models and partition function posterior probabilities, Bioinformatics, vol. 26, pp. 1958-1964, (2010).
33. J. Pei, B.-H. Kim, and N. V. Grishin, PROMALS3D: a tool for multiple protein sequence and structure alignments, Nucleic acids research, vol. 36, pp. 2295-2300, (2008).
34. O. O'Sullivan, K. Suhre, C. Abergel, D. G. Higgins, and C. Notredame, 3DCoffee: combining protein sequences and structures within multiple sequence alignments, Journal of molecular biology, vol. 340, pp. 385-396, (2004).
35. S.F. Altschul, T.L. Madden, A.A. Scha¨ffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman, Gapped BLAST PSI-BLAST: A New Generation of Protein Database Search Programs, Nucleic Acids Research, vol. 25, no. 17, pp. 3389-3402, (1997).
36. J. Devereux, P. Haeberli, and O. Smithies, A comprehensive set of sequence analysis programs for the VAX, Nucleic Acids Res., vol. 12, pp. 387–395, (1984).
37. C. Notredame and D. G. Higgins, SAGA: Sequence alignment by genetic algorithm, Nucleic Acids Res., vol. 24, no. 8, pp. 1515–1524, (1996).
38. Nguyen, K.D.; Yi Pan, An Improved Scoring Method for Protein Residue Conservation and Multiple Sequence Alignment, IEEE Transactions on NanoBioscience, , vol.10, no.4, pp.275-285,(2011).

39. W.R. Pearson, Flexible Sequence Similarity Searching with the FASTA3 Program Package, Methods in Molecular Biology, vol. 132, pp. 185-219, (2000).
40. Liu Juan; Cai Zixing; Liu Jianqin, Premature convergence in genetic algorithm: analysis and prevention based on chaos operator, Proceedings of the 3rd World Congress on Intelligent Control and Automation , vol.1 pp.495-499 ,(2000).
41. Gotoh O Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. J Mol Biol 264: 823–838, (1996).
42. Kuo-ching Liang; Xiaodong Wang; Anastassiou, D., Bayesian Basecalling for DNA Sequence Analysis using Hidden Markov Models, 40th Annual Conference on Information Sciences and Systems , pp.1599-1604, (2006)
43. Peng Yong; Dong Chongjie; Zheng Huijun, Research on Genetic Algorithm Based on Pyramid Model, 2nd International Symposium on Intelligence Information Processing and Trusted Computing pp.83-86, 2-23 Oct. (2011).
44. Singh, B.R.; Al-Khedhairy, A.A.; Alarifi, S.A.; Musarrat, J., Computational prediction of small non-coding RNA within distal 3'region of 16SrRNA gene of Bacillus sp. strain SJ-101, International Conference on Bioinformatics and Biomedical Technology, pp.257-261, 16-18 (2010).
45. Blackshields G, Wallace IM, Larkin M, Higgins DG Analysis and comparison of benchmarks for multiple sequence alignment. In Silico Biol 6: 321–339, (2006).
46. Pengfei Guo; Xuezhi Wang; Yingshi Han, The enhanced genetic algorithms for the optimization design, 3rd International Conference on Biomedical Engineering and Informatics, vol.7, pp.2990-2994, 16-18 ( 2010).
47. N. L. J. Ulder, E. H. L. Aarts, H. J. Bandelt, P. J. M. Van Laarhoven, and E. Pesch, Genetic local search algorithms for the traveling salesman problem, in Proc. 1st Workshop PPSN, vol. 496. pp. 109–116 (1991).
48. Hamidi, S.; Naghibzadeh, M.; Sadri, J., Protein multiple sequence alignment based on secondary structure similarity, International Conference on Advances in Computing, Communications and Informatics, pp.1224-1229, 22-25 (2013).
49. Silveira, S.A.; Rodrigues, A.O.; de Melo-Minardi, R.C.; da Silveira, C.H.; Meira, W., ADVISe: Visualizing the dynamics of enzyme annotations in UniProt/Swiss-Prot, IEEE Symposium on Biological Data Visualization, pp.49-56, 14-15 (2012).
50. Mahram, A.; Herbordt, M.C., FMSA: FPGA-Accelerated ClustalW-Based Multiple Sequence Alignment through Pipelined Prefiltering, IEEE 20th Annual International Symposium on Field-Programmable Custom Computing Machines, pp.177-183, (2012).
51. Cai, L.; Juedes, D.; Liakhovitch, E., Evolutionary computation techniques for multiple sequence alignment, .Proceedings of the 2000 Congress on Evolutionary Computation, vol.2, pp.829-835 (2000).
52. Zne-Jung Lee, Chou-Yuan Lee Huei-Lung Yu, Kuan-Hung Liu, and Shun-Feng Su An Intelligent System for Multiple Sequences Alignment IEEE International Conference on Systems, Man and Cybernetics, vol.2 ,pp 1042-1047 (2005).
53. Jiang Feng-Guo, The hybrid genetic algorithm based on the niche's technology, 29th Chinese Control Conference , pages.5276-5279, (2010).