Evolutionary computation approach to enhance protein multiple sequence alignments

Narayan Behera (narayan.behera@astu.edu.et)
Adama Science and Technology University
Jeevitesh M

Research Article

Keywords: Genetic algorithm, Multiple sequence alignment, Protein sequence alignment, Friedman rank test, Balibase, MCoffee, ProbCons, Ranksum

Posted Date: February 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1236304/v1

Abstract

Certain functional, structural and evolutionary relationships among the protein sequences can be inferred by protein Multiple Sequence Alignment (MSA). There are many algorithms developed over three decades but all have inherent limitations. Two important protein sequence alignment programs, ProbCons and Mcoffee stand out. Here the two best MSA algorithms are used to create more efficient computer programs through machine learning approach. The Darwinian principles of evolution are used. The evolutionary operators of a genetic algorithm such as mutation, crossover and selection are implemented to find the optimized protein sequence alignment after several iterations of the algorithm. Thus, we have developed a new MSA computational tool called Protein Alignment by Stochastic Algorithm (PASA). The efficiency of protein sequence alignments is evaluated in terms of the Total Column (TC) score. The TC score is basically the number of correctly aligned columns between the test alignments and the reference alignments divided by the total number of columns. The PASA is found to be statistically more accurate protein alignment method in comparison to other popular bioinformatics tools including ProbCons and Mcoffee.

Introduction

The Multiple Sequence Alignment (MSA) is extensively used for protein sequence analysis to find the structural, functional and evolutionary relationships among protein families. MSAs are required for building the character profiles, tracing the phylogenetic relationships, and predicting protein structures with high accuracy. But constructing a precise MSA for protein sequences is still a tough job as the required computational complexities grow exponentially with the increase of the sequence length and the number of sequences. Furthermore, it is very difficult to find an objective function to examine the alignment quality[1]. An exact solution is possible only for a small number of closely related protein sequences as discussed by Lipman et al[2]. Hence most MSA computational packages make use of heuristic progressive alignment algorithms with non- optimal solutions. Some widely used sequence alignment programs are ClustalW, Mafft, Muscle, TCoffee, M-Coffee and ProbCons. The ClustalW[3] uses a progressive alignment method to align protein sequences. In this approach, a multiple sequence alignment builds up progressively from a series of pair wise alignments by using a phylogenetic tree as the reference. The alignments of the closely related sequences are achieved initially. Then the distant ones are aligned. However, the procedure has drawbacks from the limitations of the 'early in the process' alignment mistakes. The T-Coffee approach uses a consistency-based objective function so that an information library is created by the local and the global alignments. It also uses a mixture of alignment programs and structure super-positions[4]. The most important M-Coffee program is a meta-method that takes the output of various MSA programs to merge into a single better alignment. It is an extension of the T-Coffee method, which uses a consistency approach to develop an alignment[5]. However, the Mafft program uses a Fourier Transform to determine the homologous position in which an amino acid's volume and polarity are incorporated[6]. But the Muscle method of sequence alignment is an iterative progressive alignment technique. It uses the general sum of pairs alignment score as the alignment

quality[7]. The ProbCons method is a progressive protein multiple alignment algorithm. It makes use of probabilistic consistency information to construct an alignment.[8] Advances of multiple sequence alignments were reviewed by Pei[9]. For the large-scale MSA problems, it is shown that the computation process can grow exponentially[10]. Furthermore, Paten et al.[11] have created algorithms for genomic alignments by taking into account cactus-type graphs that represent sequence alignments. An Immunological Multiple Sequence Alignment Algorithm is developed for protein multiple sequence alignment. It is tested successfully on the benchmarks of Bali Base[12]. A systematic evaluation of many popular MSA methods is performed against a bench mark data set[13]. In a recent investigation, Merge-Align method constructs consensus MSAs from multiple independent MSAs. Hence it provides an ab initio method of calculating alignment precision[14].

The Bali-base 3.0 benchmark alignment database is an assembly of 386 structural protein alignments that are verified manually[15]. This benchmark is categorized into five different groups. The first group is made of phylogenetically equidistant members of similar length, the second group has up to three orphan sequences with close relatives, the third group is made up of distantly related sequences, the fourth and the fifth groups have long terminals and internal insertions respectively. The main purpose of a benchmark is to generate a set of tests to compare the efficiencies of alternative computational tools. The best performing software package is expected to find the best alignment statistically for the uncharacterized protein sequences.

Holland (1975)[16] introduced a genetic algorithm commonly applied to optimization problems arising in science and engineering. The algorithm uses the three principles of evolution, namely, mutation, crossover and natural selection to solve an optimization problem. The genetic algorithm was successfully applied to different types of problems in biological evolution[17-20]. It has application in microarray data analysis to find the cancer genes[21]. The genetic algorithm technique is successfully implemented in different types of multiple sequence alignment problems[22-25].

Challenges for multiple sequence alignment are outlined.[26] A review provides overview of the developments of MSA and their applications in bioinformatics.[27]

Its computation is very complex. That is why several dozens of alternative methods have been developed over the past three decades. A new progressive algorithm FAMSA has been created to compute fast and align sequences efficiently for thousands of protein sequences.[28] Most popular and efficient multiple sequence alignment algorithms are recently discussed.[29]

Theory And Computational Methods

Here, we have implemented the initial alignments from the sequence alignment outputs of two MSA programs, ProbCons and MCoffees. A genetic algorithm with suitable mutation and selection operators is implemented to get a better protein sequence alignment. The genetic algorithm is a search algorithm that

imitates the processes in natural evolution as suggested by Darwin. It is usually modeled on the principles of evolution. This model employs a population of multiple sequence alignments that undergo natural selection after variation-inducing mutation processes. Finally, the population of multiple alignments evolves generation after generation till the maximum possible alignment is reached. The genetic algorithm method is explained in figure 1. It is also called evolutionary computation as it uses the fundamental evolutionary concept. The procedure is described in detail later.

In this model, a better protein multiple sequence alignment emerges after many generations of evolution. It is necessary to obtain the most optimized alignment. An examination of a multiple sequence alignment is done by using an objective function. The fitness value shows the quality of multiple sequence alignment. It also gives an insight into the implicit structural, functional and evolutionary relationships that exist among the aligned sequences. The 'sum of pairs' method finds the alignment quality. The objective is to maximize the score of alignment. The most widely used mathematical objective function is the 'sum of pairs' scores (S) for a MSA. It is defined as,

$$S = \sum_{i} \sum_{j} S(i, j) \qquad (1)$$

where i = 1, 2, .., n-1 (n is the number of sequences in the alignment), j = i +1, i+2, .., n and S(i,j) is the value obtained using structure based matrix. The alignment score of a pair of rows is the sum of the alignments of the individual pair of residues. The overall alignment score of the MSA is a sum of each pair of rows. Mutation alters one or more positions in the sequence in its initial state. Suitable mutation plays a key role in a genetic algorithm for finding the best solution. It creates diversity in the population of alignments and prevents them from stagnating at any "local optima". A mutation is implemented by inserting a gap randomly in a sequence. A gap in a sequence is generally interpreted as deletion of a character during the process of evolution. This can result in an entirely new sequence alignment. With these new sequence alignments, the evolutionary computation generally arrives at a better alignment as shown in the latter section. For each alignment in the population of alignments, gaps are inserted randomly with a fixed probability (p). It is expressed by the following formulae:

$$p = \ln(xy)/(I \times 10) \qquad (2)$$

where x is the maximum length of a sequence in the multiple sequences, y is the number of sequences and I is the number of columns with identical residues without the gaps. Thus p is fixed for a given MSA. We have analyzed a large number of alignment data by inserting gaps randomly with various probabilities to see the higher values of alignment score. As a result, we have discovered the equation 2

to serve our purpose. The gaps are inserted at the multiple sequence alignments in every iteration of the genetic algorithm.

We have used affine gap penalty for the PASA. In this scheme, two types of penalties are used for the alignment score calculation: one for gap opening and the second for gap extension. The gap opening penalty is applied only once as soon as a gap is introduced into the sequence. The gap extension penalty is calculated for each additional gap. The gap opening penalties are investigated in the range of -5 and -20 with -1 decrements in each step (such as -5, -6, -7 etc). The gap extension penalties are studied for 0, -1 and -2 corresponding to each gap opening penalty. It has been observed that a gap opening penalty of -15 and a gap extension penalty of -0.9 yield a higher alignment score. The terminal gaps are not taken into account to calculate the alignment score in this analysis. In some models, the sequence weights are incorporated in a multiple sequence alignment to correct the unequal representation. It has been observed that the inclusion of this weighting scheme gives only a small improvement in alignment accuracy (about 1%) on the Bali-base benchmark[15]. Hence PASA has not implemented any weighting scheme.

For the purpose of inserting a gap in a MSA, a random number r is generated in the range of 0 to 1. For r < p, a gap is inserted at a random position in the MSA. After insertion of a gap, the remaining sequences of that MSA are padded with gaps at the terminals. This is done to make sure that all sequences are of the same length. In a procedure involving the 'hill-climbing mutation', a new solution (sequence alignment) is obtained through a mutation from the old solution if the new solution is fitter. Otherwise, the old solution is retained. The hill climbing algorithm works as follows: For each alignment A of the population of alignments, its current fitness ƒ is calculated by using equation (1). The fitness is basically defined as the alignment score. Then A is mutated by inserting a gap randomly in one of the sequences according to the equation 2 to produce a mutant alignment M. Finally, the gaps are padded at the end of the other sequences to make all sequences with the same length. Now, if ƒ (M) is fitter than ƒ (A), then the alignment A is replaced with alignment M else the alignment A is retained.

2.1 Block shift operation

We have introduced an operator that searches a block of gaps in a MSA and shifts it to the neighboring positions. This process is illustrated in fig 2.

A block of any shape is searched randomly and then it is shifted either forward, or backward, upward or downward in the MSA. The steps are as follows: two integer random numbers r1 and r2 are generated. They refer to the sequence number in the MSA and the character position in that particular sequence respectively. The position corresponding to r2 can be a character or a gap. If it is a character then one looks forward for a gap in the same sequence. This position is taken as the gap starting position (Gs) until a residue is found. If no gap is found, then the next individual of alignment is considered. The number of residues adjacent to Gs is called Gap count (Gc). The same procedure is carried out for all other sequences in the same MSA.

From the above procedure, one can find out the block of gaps (Bg) for the forward or backward shift of the entire block of gaps. If the position corresponding to r2 is a gap, then we look backward and find out the gap starting position Gs. Similarly, we find Gs and Gc. The same procedure is carried out for all other sequences in a MSA. First, Bg is shifted towards the right by one position. Then the alignment score of the MSA is checked. If it is increased, the old MSA is replaced by the new MSA. If the alignment score is decreased, Bg is shifted again for one extra position toward the right and the new MSA score is obtained. If it is increased the old one is replaced by the new MSA. But if the alignment score has still not increased, we perform the left shift by one and two positions respectively. Then we check if the new MSA score is increased. If so, then the old MSA is replaced by the new one. If the alignment score is not increased after all the four shifts of the Bg, the old MSA is retained in the population of alignments. Then the next MSA is considered for the above procedure and so on. It must be noted here that the top and bottom block shifts are not performed because the blocks of gaps are irregular. Hence they will disrupt the entire alignment.

2.2 Block elimination operation

We have designed an operator called as the block removal as described in fig 3.

The basic rule of multiple sequence alignment is maintained. It allows the minimum number of gaps to construct a suitable multiple sequence alignment. The following steps are used. Like the above procedure, two integer random numbers r1 and r2 are generated. Using r2, a starting gap position is found and the number of gaps from that position to the very next residue of the sequence is counted. If there is no gap, then we choose the next MSA in the population. The same procedure is carried out for all other sequences. The first block of gaps corresponding to the maximum number of sequences is found. This block of gaps is eliminated from the alignment of the MSA. The same number of gaps from all other sequences is deleted. Then the alignment scores are compared before and after the block removal. If the score is increased, the new MSA is retained otherwise the old MSA is kept.

2.3 Elitism selection

In each generation of the evolutionary computation, only a fraction of the MSAs is replaced by better MSAs. Half of the high scoring alignments will survive unchanged while the other half is replaced by the alignments generated by block shifting and block removal operators. To assess the efficiency of the PASA, the protein benchmark Bali-base 3.0 is used. The program is implemented on a 3 GHz Intel Xeon Dual core processor with 8 GB RAM. Fedora core 6 is used as the operating system. The PASA program is compared with the ProbCons version 1.11, MCoffee, ClustalW[30] version 1.83, the T-Coffee version 4.96, the Mafft version 5.861 and the Muscle version 3.6. All the above programs are executed at the default modes.

In the current model, new kinds of mutation operators such as block shift and block removal operators are implemented as discussed above. The initial 100 population of MSAs from the two program outputs- ProbCons and MCoffee - are constructed with equal probability. The PASA combines and improves the alignments over successive generations. Finally the most optimized alignment is obtained at the end. It

makes use of mutation and selection operators in every generation until the required number of generations is reached. Then it finds the most optimized MSA. The difference of the best fitness for two consecutive generations is noted. If this difference is less than 1% for ten consecutive generations, the program is terminated. Otherwise it proceeds to the next generation.

The PASA method is tested on the popular benchmark Bali-base version 3. We have achieved a statistically significant enhancement of alignment over the other popular methods. The alignment quality of each bioinformatics tool is determined by measuring Quality (Q) and Total Column (TC) scores. The Q is a number of correctly aligned residue pairs between test alignment and reference alignment divided by the total number of aligned residue pairs in the reference alignment. The TC is a number of correctly aligned columns between the test alignment and the reference alignment divided by the total number of columns in the reference alignment. In general, the TC score is lower than the Q score. The scores are calculated by using software QSCORE. However, the TC score provides a more stringent measure to evaluate the efficiency of a protein sequence alignment as far as the conserved blocks of residues are concerned. Hence, we are using the TC scores to find the best alignment. Then we determine the corresponding Q for the best alignment.

In order to compare the performance of various MSA programs with the PASA method, we have conducted a non-parametric test known as the Friedman rank test. This test makes no assumption about the distribution of alignment scores across different pairs of MSA programs. Instead of using alignment score directly, the ranking of the score across the pairs of programs is used for finding the efficiency of a MSA method. Higher the alignment score of an alignment program, the better is its rank. The Ranksum is calculated as the sum of ranks for a given MSA program. The concept of null hypothesis is used to compare the efficiencies of two MSA programs in terms of the TC and the Q values. Null hypothesis assumes that a pair of programs is good in equal probability. The Ranksum is further used to calculate the P-value, which measures a probability factor for rejecting the null hypothesis. If the P-value is very small (say, ≤ 0.05), the above null hypothesis is rejected. Furthermore, the program performance is better for high values of Ranksum. If the P-value is greater than 0.05, there is no statistically significant difference between the efficiencies of two comparable MSA programs. For a set of scores (say, Q and TC), the P-values are obtained using the Friedman rank test from the statistical analysis package R[31].

Results And Discussion

The scores and the statistical significance of the alignments are summarized in tables 1 and 2. As the algorithm is stochastic, the results are dependent on the starting conditions of the program. The random numbers are called upon several times during the run of the program. The results will be different for another set of random numbers. Therefore, an average of twelve different simulations is considered to find the best results. It is observed that a large number of distinct simulations increase the computational time. But a small number of simulations decrease the chances of getting the efficient results. There is no straightforward relationship between the Q and the TC scores. A higher Q score might have been lost in

some cases while finding the best TC score. The results on Bali-base benchmark alignment database are shown in Table 1 and 2.

The PASA is found to achieve improvements of 0.7% over the MCoffee, 1.2% over the ProbCons, 14% over the ClustalW, and 9.28% over the Mafft in terms of Q scores on Bali-base 3 benchmark, as shown in Table 1. In terms of TC scores, the PASA has enhancements of 3.6% over MCoffee, 7% over ProbCons, 28% over ClustalW, 24% over Mafft, 14% over Muscle and 24% over TCoffee measured on Bali-base 3 as shown in Table 2.

Table 1: Average of Q scores on the Bali base 3 benchmark

Methods No. of alignments	R1.1 (76)	R1.2 (88)	R2 (82)	R3 (60)	R4 (49)	R5 (31)	Overall (386)	Ranksum
PASA	**59.29**	**86.72**	**85.54**	**76.88**	**70.83**	**74.9**	**75.69**	**2322**
MCoffee	58.18	86.32	85.14	76.14	70.35	74.87	75.16	2170
ProbCons	59.21	85.80	84.68	74.80	70.28	73.98	74.79	2003.5
ClustalW	46.82	79.64	79.70	65.86	61.73	63.17	66.15	732.5
Mafft	47.08	80.58	81.77	72.04	65.22	68.91	69.26	948.5
Muscle	53.23	83.31	82.99	72.16	66.80	69.61	71.35	1326
TCoffee	50.08	83.94	83.72	70.24	66.68	70.46	70.85	1305

The columns represent the average of Q score for all the alignments. The Ranksum values are found from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold. In the Bali-base software, evaluation of multiple alignment programs is divided in five hierarchical reference sets (R1 to R5) in terms of residue identities and conservation properties.

The statistically significant differences in the overall TC scores are shown in Table 2.

Table 2: Average of TC scores on the Balibase benchmark

Methods No.ofalignments	R1.1 (76)	R1.2 (88)	R2 (82)	R3 (60)	R4 (49)	R5 (31)	Overall (386)	Ranksum
PASA	**35.36**	**67.41**	**36.12**	**39.34**	**21.47**	**33.33**	**38.83**	**2312.5**
MCoffee	32.74	66.47	33.94	38.15	21.12	32.43	37.47	2060.5
ProbCons	34.02	64.79	33.50	34.53	19.64	31.17	36.27	1783
ClustalW	23.11	58.70	26.03	29.48	18.58	25.07	30.16	1030.5
Mafft	23.46	57.25	28.53	32.48	18.68	27.44	31.30	1050
Muscle	30.07	62.36	28.87	32.87	20.37	28.64	33.86	1364.5
TCoffee	25.89	62	28.23	27.88	19.02	23.83	31.14	1207

The columns represent the average of TC score for all the alignments. The Ranksum values are calculated from the Friedman test for all the alignments. The highest score in each benchmark set is highlighted in bold. In the Bali-base, the multiple alignment programs are divided in five hierarchical reference sets (R1 to R5) in terms of residue identities and conservation properties.

The Friedman rank test analysis is presented in Table 3.

Table 3: Statistical analysis on the Bali-base benchmark 3

Methods	PASA	MCoffee	ProbCons	ClustalW	Mafft	Muscle	T Coffee
PASA		+1.0 q^{-08}	+4.8q^{-12}	+<2.2q^{-12}	+<2.2q^{-12}	+< 2.2q^{-16}	+< 2.2q^{-16}
MCoffee	-< 2.2q^{-16}		+3.6q^{-06}	+ < 2.2q^{-16}	+< 2.2q^{-16}	+< 2.2q^{-16}	+< 2.2q^{-16}
ProbCons	-< 2.2q^{-16}	-1.9q^{-09}		+< 2.2q^{-16}	+< 2.2q^{-16}	+< 2.2q^{-16}	+< 2.2q^{-16}
ClustalW	-< 2.2q^{-16}	- < 2.2q^{-16}	- < 2.2q^{-16}		-1.6q^{-10}	-< 2.2q^{-16}	-< 2.2q^{-16}
Mafft	-< 2.2q^{-16}	- < 2.2q^{-16}	- < 2.2q^{-16}	(0.60)		-< 2.2q^{-16}	- 4.5q^{-16}
Muscle	-< 2.2q^{-16}	- < 2.2q^{-16}	-8.0 q^{-16}	+2.1q^{-16}	+9.0q^{-09}		(0.33)
TCoffee	-< 2.2q^{-16}	- < 2.2q^{-16}	- < 2.2q^{-16}	2.4q^{-16}	+0.3q^{-04}	(0.34)	

In the table 3, q is assigned a value of 10. Each value in the table contains the P-value assigned by the Friedman rank test. This indicates the significance of the difference of alignments between the programs. The upper triangle matrix values are derived from the Q scores on the Bali-base 3. The signs + and - represent that a program in a row performs significantly better and worse respectively than that of a program in a column. If the P-value is greater than 0.05, the difference is not significant and is shown in parentheses. For example, the PASA ranks higher than the ClustalW with a P-value of 2.2 x 10^{-16}. The lower triangle matrix values are obtained from the TC scores on the Bali-base 3.

It is also observed that PASA method is able to improve sequence alignment by 3% to 26% in terms of TC scores measured on the Bali-base benchmark 3 protein dataset in comparison to other popular alignment programs. The PASA software requires a delicate analysis of the genetic algorithm to obtain the best alignment. A number of operators, such as block insertion, block shifting, block searching in terms of the gaps and different types of crossovers (where the two MSAs are cut in a particular position and then fused together to make new MSA) have been tried. Most of those operators have improved the sum of pair scores. But in terms of Q and TC scores, they have failed. In our model, we have not any crossover operation as it is found to be disruptive for good alignment. Finding the highest alignment score of a multiple protein sequence alignment is an important open field of research. The research area is evolving rapidly.

We have used a simple concept of evolutionary optimization model. It starts the initial population of alignments as the MSA program outputs of two most efficient tools: ProbCons and MCoffee. These two

sequence alignment programs are different than the others in the sense that the average length of characters in the aligned sequences are greater than the corresponding aligned sequences found in Balibase version 3 reference alignments. As a result, the mutational procedure of gap elimination operator plays a significant role in increasing the final alignment. Consequently, we have obtained a significant enhancement of alignment in terms of Q and TC scores in comparison to the individual MSA methods. It is interesting to observe that this PASA alignment program structure is such that the program running time reduces by a factor of about 10 when the codes are written in C language instead of using Perl.

It has been reported that structural alignment programs produce outputs where 11% – 19% of the core residues are misaligned. Majority of the benchmark alignments are obtained by using the structural alignment programs. There we suspect that our PASA alignment program will provide better alignment accuracy. This type of analysis can be extended to RNA alignments although the work can be very cumbersome.

It is a well known fact in the scientific literature that although a genetic algorithm model can give better result, it takes more running time due to the stochastic nature of the algorithm. As the random numbers are called more often, the program runs slow. Here, we have tracked the program running time for the Balibase subset RV12 protein reference alignment benchmark (consisting of 88 alignments). It takes 17 minutes and 79 minutes in the cases of Probcons and MCoffee respectively while for the PASA program it is 96 minutes. The computational time can be drastically reduced when the program is allowed to run on a multi-cluster system having hundreds of nodes. The PASA tool gives statistically better alignment over the two competing programs while maintaining the same range of computational time. The traditional genetic algorithm approach towards sequence alignment like SAGA tends to build alignment from the initial random alignments of sequences. But in our approach, initial alignment solutions are near to global optimum as they are the outputs of two other important programs. Hence, it takes very less time compared to the conventional genetic algorithm programs in other models. This gives a proof that the evolutionary computation algorithm can be used as an excellent optimizer for the sequence alignment problem. Due to the stochastic nature of the computation, the genetic algorithm can converge to a local optimum. The PASA can generate more than a single suboptimal alignment depending on the number of initial population of alignments and various evolutionary operators. This feature can sometimes be necessary to select one biologically relevant MSA for further analysis.

To obtain the best sequence alignment, normally it is a tough task to choose the right MSA program over several programs available in the literature. So the PASA tool is a better alternative which combines the outputs of some of the efficient individual methods and further improves them to obtain a better protein sequence alignment. The PASA has used certain biological knowledge such as the structure based derived matrix. It has also incorporated some novel genetic algorithm operators. It creates interesting results even below the twilight zone of sequence similarity.

References

1. Thompson, J.D., et. al. Towards a reliable objective function for multiple sequence alignments. *Journal Molecular Biology*., 2001; 314,937:951

2. Lipman,D.J., et. al. A tool for multiple sequence alignment. *Proceedings of National Academy of Science*., 1989; 86, 4412:4415.

3. Thompson,J.D., et. al. CLUSTALW: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*., 1994; 22, 4673:4680.

4. Notredame,C., et. al. T-Coffee: a novel method for fast and accurate multiple sequence alignment. *Journal Molecular Biology*., 2000; 302, 205:217.

5. Wallace,I.M. et. al.. M-Coffee: combining multiple sequence alignment methods with T-Coffee. *Nucleic Acids Research*., 2006; 34, 1692:1699.

6. Katoh,K., et. al. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic Acids Research*., 2002; 30, 3059:3066.

7. Edgar,R.C. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*., 2004; 32, 1792:1797.

8. Do,C.B. et. al. ProbCons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*., 2005; 15, 330:340.

9. Pei, Jimin Multiple protein sequence alignment, *Current opinions in structural biology*, 2008; 18, 382:386.

10. Lloyd,S. and Snell,Q.O., Accelerated large-scale multiple sequence alignment. *BMC Bioinformatics*, 2011; 12, 466.

11. Paten, B. et al. Cactus: Algorithms for multiple sequence alignment. *Genome Research*, 2011; 21, 1512:1528.

12. Cutello, V. et al. Protein multiple sequence alignment by hybrid bio-inspired algorithms; *Nucleic Acids Research*., 2011

13. Thompson, J.D. et. al. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PLOS One*, 2011; 6(3), e18093.

14. Collingridge, P.W. and Kelly, Steven MergeAlign: improving multiple sequence alignment performance by dynamic reconstruction of consensus multiple sequence alignments. *BMC Bioinformatics*, 2012; 13, 117.

15. Thompson, J.D., et. al. Balibase 3.0: Latest Developments of the Multiple Sequence alignment Benchmark. *Proteins.*, 2005; 61, 127:136.

16. Holland, J.H. Adaptation in natural and artificial systems. *Univ of Michigan press, 1975; Ann Arbor, MI.*

17. Behera N & Nanjundiah V (1995) An Investigation into the role of Phenotypic Plasticity in Evolution / *Journal of Theoretical Biology Vol.172, No. 3, 225-234.*

18. Behera N & Nanjundiah V (1996) The Consequence of phenotypic plasticity in cyclically varying environments: a genetic algorithm study / *Journal of Theoretical biology*, Vol.178, No.2, 135-144

19. Behera N. &Nanjundiah V. trans-Gene Regulation in Adaptive Evolution: a Genetic Algorithm Model, *Journal of Theoretical Biology* 1997; 188,153:162.

20. Behera N. &Nanjundiah V. Phenotypic plasticity can potentiate rapid evolutionary change. *Journal of Theoretical Biology*, 2004; 226, 177:184.

21. Behera, N., Sinha, S, Gupta, R, Geoncy, A., Dimitrova, N & Mazher M (2018) Analysis of gene expression data by evolutionary clustering algorithm *IEEE Explore International conference of information technology Bhubaneswar* (DOI 10.1109/ICIT.2017.41 in 2018)

22. Zhang,C and Wong,A.K. A Genetic algorithm for multiple molecular sequence alignment. CABIOS, 1997; 13, 565:581.

23. Cai, L. et. al. Evolutionary computation techniques for multiple sequence alignment. *Proceedings of the second congress on evolutionary computation*, 2000; 2, 829-835.

24. Anbarasu, L.A., et. al. Multiple molecular sequence alignment by island parallel genetic algorithm. *Current Science* 2000; 78, 858:863

25. Notredame,C. and Higgins,D.G. SAGA: sequence alignment by genetic algorithm. *Nucleic Acids Research.*, 1996; 24, 1515:1524.

26. Kemma C and Notredame C. Upcoming challenges for multiple sequence alignment methods in high-throughput era. 2009, Bioinformatics, 25, 2455-65

27. Chatzou M et al. Multiple sequence alignment modeling: methods and applications. 2016, Briefings in Bioinformatics, 17, 1009-1023.

28. Deorowicz S et al. FAMSA: Fast and accurate multiple sequence alignment of huge protein families. 2016, Nature scientific reports, 6, article number 33964.

29. Reddy B and Fields R. Multiple sequence alignment algorithms in bioinformatics. 2021, Proceedings of SmartCom held in Las Vegas, Nevada, USA, pp 89-98

30. Thomsen, R. et. al. ClustalW alignment improver using evolutionary algorithms. *Proceedings of the fourth congress on evolutionary computation*, 2002; 1, 121:126.

31. R-package, URL: *http://www.r-project.org/*

Figures

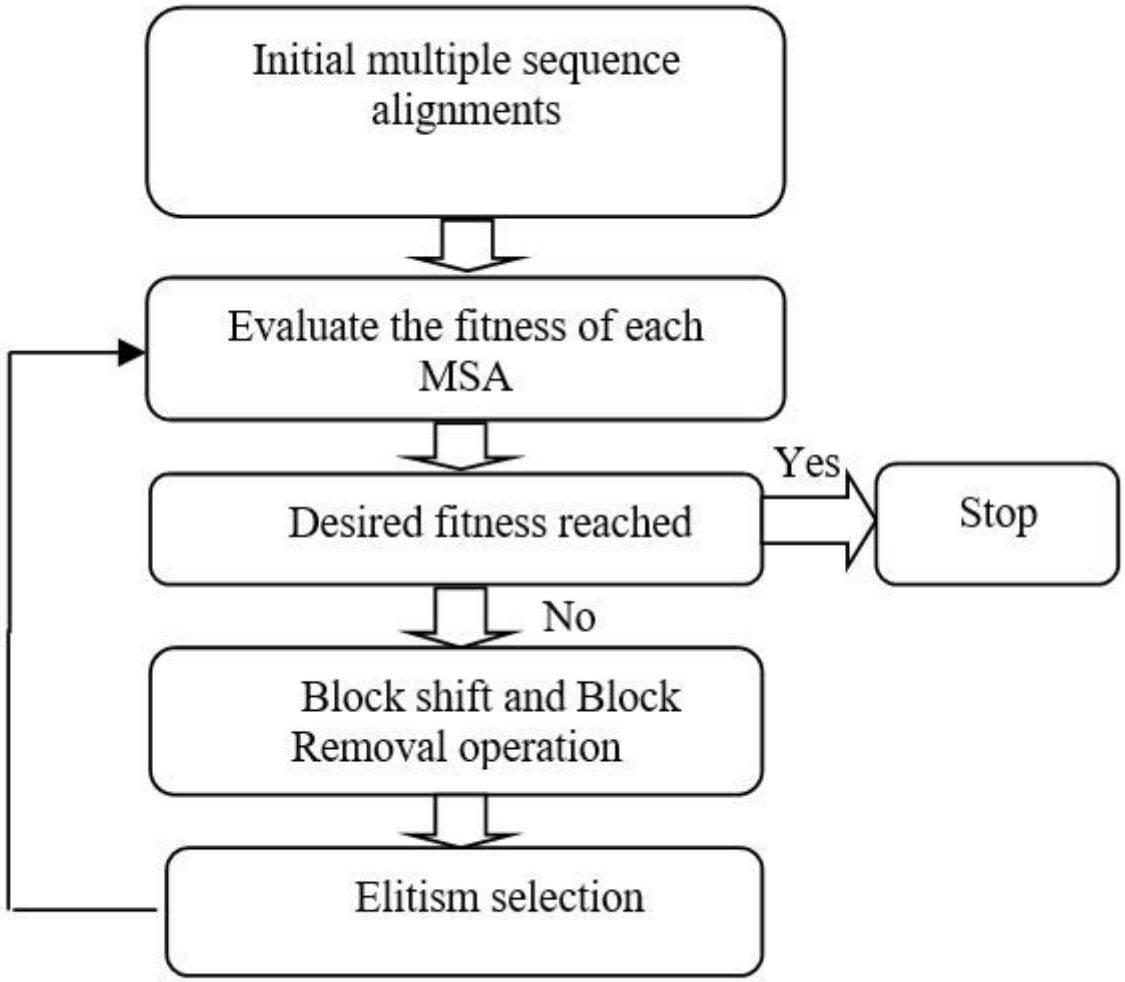

Figure 1

Flow chart of PASA

Figure 2

Block shift operator

Figure 3

Block removal operation