ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 9, July 2015*

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

An Elitist Model for Obtaining Alignment of Multiple Sequences using Genetic Algorithm

Shouvik Chakraborty [1], Arindrajit Seal [2], Mousomi Roy [3]

P.G. Student, Department of Computer Science & Engineering, University of Kalyani, West Bengal, India[1,2,3]

ABSTRACT: Multiple Sequence Alignment is a well known and computationally one of the most challenging problem in the field of molecular biology. It takes a major role in the domain of molecular sequence analysis. An alignment is basically the arrangement of two (pair-wise alignments) or more (multiple alignments) sequences of 'residues' (i.e. amino acids or nucleotides) that maximizes the similarities among them. In algorithm, the problem basically consists of opening and extending gaps in the multiple sequences. The goal is to maximize an objective function (measurement of similarity). In the recent past, Genetic Algorithm (GA) based solutions have emerged as useful tool for solving MSA problem. Genetic algorithms - a well known category of evolutionary algorithms are well suited for various problems of this nature because gaps and residues are discrete units. This paper proposes an elitist solution using genetic algorithm for determining alignment of multiple sequences. Different genetic operators like crossover rate, mutation rate etc. can be specified by the user. Results have been recorded depending upon observations & experiments w.r.t variable parameters like variable number of generations vs. fixed population size & vice versa, variable mutation & crossover rates. The datasets have been chosen from the BAliBASE standard benchmark alignment suite for experimental work & analysis.

KEYWORDS:Multiple Sequence Alignments, Genetic Algorithm, Elitism, Molecular Biology

I. INTRODUCTION

Multiple Sequence Alignment (MSA) is one of the most active and difficult ongoing research problems. Multiple sequence alignment of DNA, RNA, or amino acids is important for biologists to study similarities in sequences which often lead to similarities in function and provide priceless evolutionary information. The evolutionary history of the sequences is concluded by the alignment. It is very difficult to align three or more sequences of biologically relevant length. Efficient computational algorithms may be used to produce better results. The basic problem of MSA algorithms is their exponential time complexity with the considerable large input data set. In fact computationally MSA is considered to be a NP Complete problem [2] and therefore most of the multiple sequence alignment programs use various heuristic methods [1]. Genetic algorithm is very effective method for solving the MSA problem. It searches through the solution space effectively and generates good alignment results. The main advantage of genetic algorithms over any other optimization methods is that you need not provide a particular algorithm to solve a given problem. Genetic algorithm needs only a fitness function to evaluate different solution and applicable for parallel computers. Proposed algorithm always ensures that the $(i+1)^{th}$ generation of populations will have better fitness value as compared to their predecessors and it can be expected that the model provides at least near to optimal alignments after some N number of generations. The overall purpose is to improve the alignment with each generation. There are some proposed iterative methods to improve the problem of MSA. For example, evolutionary approach such as SAGA [3] based on genetic algorithm have been applied efficiently to solve the MSA problem. Two different objective functions have been used for searching large search space very efficiently but in increases time complexity. Another algorithm in which multiple cut points have been selected [4] based on genetic algorithm and divide and conquer technique. According to the author cutting the sequences decreases complexity and shows some significant results. But the drawback is this method was not useful for protein sequences. The working principle of the model is to encode the number of gaps into a chromosome and with the introduction of gaps at different positions. This paper also explores the possibility of applying GA based solution for MSA problem. One such solution is proposed and presented in this paper. The paper is organized as follows: Section II describes the Multiple Sequence Alignment problem with Genetic Algorithm, Section III describes our proposed elitist model for Multiple Sequence Alignment problem, and Section IV shows Experimental Results with graphical representations, Section V is for the Conclusion and then the References section.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 9, July 2015*

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

II. MULTIPLE SEQUENCE ALIGNMENT USING GENETIC ALGORITHM

Multiple molecular sequence alignment has the characteristic of having a very high computational complexity; the stochastic optimization method is used in GA. The GA encodes the solution as chromosomes and fitness is evaluated by using the suitable and efficient fitness function. Large, complex or poorly understood search space can be efficiently searched using it.

The genetic algorithm starts with the set of solutions consisting of chromosomes (strings of DNA or Amino Acids) called population. Genetic Algorithm is one of the well known search methods to find the suitable solution from the search space. Some problems in which the global optimum cannot be determined, the solution based on Genetic Algorithm is very effective [5]. Selection operator is used to select the chromosomes from one generation to another depending upon the fitness value. Two chromosomes are combined to produce the new offspring with a particular rate of crossover. This rate indicates the percentage chromosome alteration. The basic idea behind the crossover operator is providing the best chromosome when the characters of the parents are swapped or interchanged. Genes are altered within the chromosome by the mutation operator with the mutation rate. The resultant chromosomes are measured using the fitness function. The main objective of mutation in GA is to introduce genetic diversity. It is evident from the literature, the crossover rate should be kept high, about 75%-95% (Some studis and results show 60% is best choice for some problem) and the mutation rate should be kept very low, about 0.5%-1% [6].
Outline of Genetic Algorithm:

1. Produce an initial population of individuals (Chromosomes)
2. Evaluate the fitness of all chromosomes using a fitness function
3. While the termination condition not met do
3.1 select fitter individuals for reproduction
3.2 Recombine (crossover) between chromosomes
3.3 Mutate chromosomes
3.4 Evaluate the fitness value of the modified individuals
3.5 Generate the new population

Sometimes we can lose the best solution due to crossover or mutation. The process may rediscover the best solution in some sub sequent generations but there is no surety of it. Elitism can be used to solve this problem. Elitism carries a small subset of fittest solution to the next generations. This method can have some huge impact because we ensure that we never lost the best solution ever found. Preserved solutions can be used in the generation of parents when breeding the rest of the next generation.

III. PROPOSED APPROACH

In this section we present our algorithm for solving the MSA problem. We propose Genetic Algorithm Based Elitist Model to solve the MSA as shown in Figure 3.

A. **Initialization of Population**

We have generated P number of random chromosomes to initialize the population. The value of P can be given by the user. The chromosome representation is described below.

B. **Chromosome Representation**

Chromosome can be represented in various ways. The main goal of Multiple Sequence Alignment is to insert gaps in proper places so that the maximum fitness can be achieved. Suppose we have n number of sequences with variable lengths. To generate a chromosome, at first we have calculated the length of the maximum sequence. Then we multiply a constant between 1 and 2 with it to fix the length of each chromosome. After multiplication, the length is increased by some percentage (for example, if we multiply with a constant 1.2, then the length will be increased by 20%) which is the fixed maximum length (l_{fm}) of every sequence. The lengths of the different sequences are different. Therefore number of gaps for each sequence may be different. Now, if the l_{fm} is L. So, we have L places. Let us assume that the original length of n sequences as follows: l1, l2, l3,..... ln. So, the total number of gaps (G) in will be

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 9, July 2015

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

$$\sum_{i=1}^{n}(l_{fm} - l_i)$$

The length of each chromosome will be G where (l_{fm}-l_i) (for i = 1,2,.....,n) number of places are reserved for i^{th} sequence, which represents the number of gaps allowed in each sequence. Now in every sequence gaps can be inserted in different positions, and with different possible combinations. For i^{th} sequence, we have generated (l_{fm}-li) number of random numbers between 1 to l_{fm} (because possible positions of gaps can only be in 1 to l_{fm}). By repeating this process for each sequence we can get one chromosome. In this way we can generate first P number of chromosomes. Consider two sequences as follows:

ACDE
GAHFIC

Here maximum length is 6. So if we multiply 1.2 with 6 we will get 8 (by considering the ceil value). So our l_{fm} is 8. For the first sequence number of allowed gaps is 8-4 = 4 and for the second sequence 8-6=2. The length of our chromosome will be (4+2) = 6. Now for the first sequence, we generate 4 random integers between 1 and 8 (inclusive) and for the second sequence 2 random integers between 1 and 8 (inclusive). No repetition is allowed in a particular sequence. When repeated values exist in a single chromosome then they must belongs to different sequences. The chromosome can be expressed as shown in Figure 1.

5	2	7	1	3	7

Figure 1: Chromosome representation

So, we will get the alignment for the above chromosome as shown in Figure 2.

-	-	A	C	-	D	-	E
G	A	-	H	F	I	-	C

Figure 2: Alignment corresponding to the chromosome representation

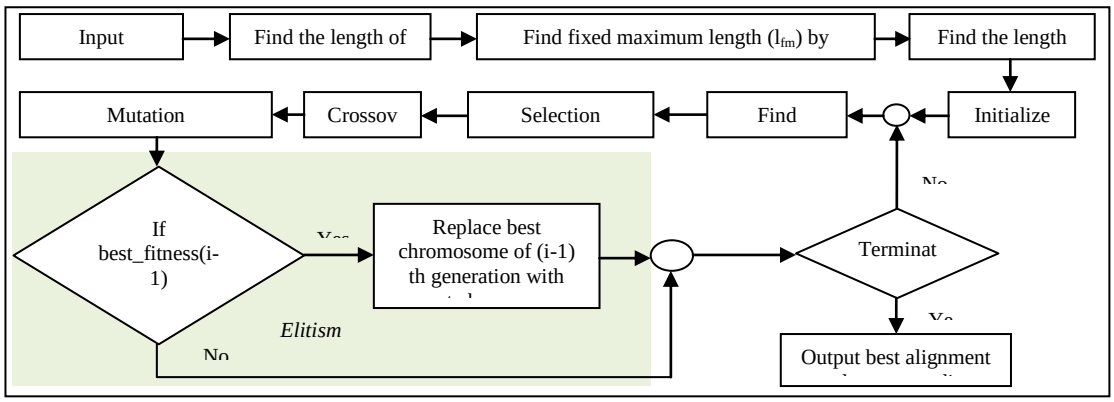


Figure 3: Genetic Algorithm Based Elitist Model

C. **Selection/Reproduction**

The first basic operator applied on population is reproduction. Chromosomes are selected from the population for crossover operation. According to Darwin's Theory of survival of the fittest, the fittest solutions should survive and create new offspring [7]. That is why reproduction operator is sometimes known as the selection operator.

In our model, we have used tournament selection (binary) method. In this method two random chromosomes are taken and after comparing their fitness values, better one is selected.

D. **Crossover**

Crossover is a process of taking more than one parent chromosomes and producing offspring from them. In our model we have taken two random chromosomes and then performed the uniform crossover on them. But the main difficulty of the

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization Volume 4, Special Issue 9, July 2015

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

crossover operation is that the duplicate data can appear for a particular sequence. To overcome this problem we have performed the sequence wise uniform crossover. To perform the sequence wise crossover a binary mask (whose length is equal to the number of sequences) is generated depending upon the crossover probability. Information is exchanged of those sequences where the bit is high. In this way we can get child chromosomes in our population.

E. **Mutation**

Mutation is a genetic operator that can be used to maintain genetic diversity from one generation of a population to the next. Mutation can alters one or more than one genes in the chromosomes depending upon a probability. Mutation is applied on the fittest set of the population after crossover phase and its size is equal to P. Mutation probability must be kept low enough because mutation is a very small change in the chromosome. But altering a particular value is little difficult because we should take care about the fact that there must not any duplicate value for a particular sequence. The conventional mutation operator used in TSP [8] cannot be applied here because in this problem we must shuffle the position of the gaps. In our model, mutation has been performed by changing each value depending upon the mutation probability but, changing a value may generate some chromosomes which are not allowed. So the mutation algorithm given below:

1. For a particular value in a chromosome first check that mutation probability is acceptable or not. If yes then mutate the value using the following steps.
2. First get the sequence number to which that value belongs.
3. Change the value with a unique value (i.e. a value which is not present in chromosome corresponding to that sequence number)
4. Repeat the above steps for all values in a chromosome.

F. Fitness Function

The fitness function determines how "good" an alignment is. Fitness functions plays an important role in determining the performance of evolutionary algorithms. The most common strategy that is used, albeit with significant variations, is called the "Sum-Of-Pair" [9] Objective Function. In this method, for each location on the aligned sequences, one of three situations will occur: match, mismatch or a gap (aligned with a character or gap). The fitness or scoring function of each individual is calculated by the formula:-

$$\sum_{i=1}^{n} Fitness\ score\ for\ each\ pair\ of\ sequences$$

$$\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \boldsymbol{ScoringMatrix}(i,j)$$

We have used PAM 250 scoring matrix.

A gap penalty value must be settled for determining the cost of aligning an amino acid with a gap. We have used linear gap penalty value to penalize a gap. Linear gap penalty can be expressed as

$$\gamma(g) = -gd$$

$\gamma(g)$= gap penalty score of a gap of length g, d = gap opening penalty, g = gap length

We have used gap opening penalty value 1. When a gap is aligned with another gap then we have added 0 to our fitness value.

G. Elitism

Elitism means the best string seen up to the current generation which is preserved in a location either inside or outside the population. Elitism can be introduced in various ways. For example, we can preserve the best solution of initial population in a separate location and then compare it with the best result of the new generation and replace it if it is better. Some of the top fitted chromosomes are transferred directly into the next generations or the best of the i^{th} generation can be compared with the worst of the $(i+1)^{th}$ generation and replace it if it is better. Elitism always ensures that the best solution found up to a particular generation will not be lost during the successive generations.

Elitism can rapidly increase the performance of GA by not losing the best-found solutions. In our approach, the best solution of the i^{th} generation is compared with the worst solution of $(i+1)^{th}$ generation and if it is worse then, it gets replaced.

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 9, July 2015*

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

IV. EXPERIMENTAL RESULTS

The algorithm is implemented using MATLAB R2014a. We have tested our algorithm with some datasets from the Reference alignments of BAliBASE [10] as the benchmark for the experiment. BAliBASE is a well known standardized set of benchmark reference for multiple sequence alignments. These datasets are used as input to our multiple sequence alignment programs using GA. The datasets were chosen to cover a range of different sequence lengths. The details of the datasets used in this paper are given in Table 1.

Table 1: The Summary of the BAliBASE reference test cases used in this paper

Sequence Id	Reference	Property	No. of Sequences	Sequence Names
RV11_BB11001	Reference 1	Very divergent conservation (<20% identity)	4	>1aab_, >1j46_A, >1k99_A, >2lef_A
RV12_BB12003	Reference 1	medium to divergent sequences (20-40% identity)	8	>1aho_, >1bmr_, >1i6f_A, >AEP_MESMA, >SCAT_MESMA, >SCX6_ANDAU, >SCX6_CENLL, >BIRT_PARTR
RV11_BB11013	Reference 1	Very divergent conservation (<20% identity)	5	>1idy_, >1hst_A, >1tc3_C, >1aoy_>1jhg_A
RV30_BB30027	Reference 3	Equidistant divergent families or subgroups with < 25% residue identity between groups	20	>1dvh_, >C553_DESVM, >C553_DESDN, >CYSD_CHLLT, >1b7v_A, >1cno_A, >1h1o_A, >CYSD_CHLTE, >C554_PARSP, >1fj0_A, >155c_, >C550_PARPN, >1akk_, >CYC_ASPNG, >CYC_DEBHA, >CYC_SCHPO, >CY2_RHOGL, >CY2_RHOVA, >CY22_RHOFU, >CY21_RHOCE
RV11_BB11025	Reference 1	Very divergent conservation (<20% identity)	4	>1tvx_A, >1prt_F, >1sap_, >1lt5_D
RV11_BB11029	Reference 1	Very divergent conservation (<20% identity)	4	>1ycc_, >1cno_A, >1a56_, >1h31_B

In order to examine the validity of our algorithm, we test number of series with amino acid sequence. We have calculated the fitness score of each one of these datasets. The experimental results are shown in Table 2. Here cp represents crossover probability and mp represents mutation probability, P is the population size and G is the number of generations. We have used PAM 250 matrix for compute these fitness values.

Table 2: Fitness scores of different sequences

Sequence Id	Without Elitism					Elitist model				
	cp=0.5, mp=0.03	cp=0.65, mp=0.06	cp=0.85, mp=0.1	P	G	cp=0.5, mp=0.03	cp=0.65, mp=0.06	cp=0.85, mp=0.1	P	G
RV11_BB11001	-312	-242	-268	25	250	-114	-58	-53	25	250
RV12_BB12003	-1883	-2009	-2085	20	250	-1282	-1382	-1751	20	250
RV11_BB11013	-738	-727	-746	25	250	-612	-656	-680	25	250
RV30_BB30027	-24758	-25211	-25189	15	250	-23937	-24255	-24309	15	250
RV11_BB11025	-466	-460	-543	30	250	-352	-403	-398	30	250
RV11_BB11029	-657	-660	-642	20	250	-500	-493	-562	20	250

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 9, July 2015*

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India

We have tested our approach with different test cases and different crossover and mutation probability. We have presented a generation wise analysis of test cases with crossover probability 0.65 and mutation probability 0.06 in Table 3 and Table 4. In Table3 the data values shows the result without elitism. We can observe that data values for any case is fluctuating and sometime the best solution found so far is lost. But the data values from Table 4 (shows the result of the elitist model) clearly shows that we will not lose the best value found in a generation.

Table 3: Generation wise analysis (Without Elitism)

No. of Generations	RV11_BB11001	RV12_BB12003	RV11_BB11013	RV30_BB30027	RV11_BB11025	RV11_BB11029
1	-345	-2140	-846	-25360	-490	-598
5	-360	-2265	-789	-25081	-474	-654
10	-336	-2115	-743	-24680	-496	-592
25	-266	-1979	-787	-25195	-490	-663
50	-316	-2058	-817	-25242	-476	-623
100	-290	-2095	-788	-24938	-500	-623
200	-303	-1963	-762	-25156	-494	-603
250	-242	-2009	-727	-25211	-460	-660

Table 4: Generation wise analysis (Elitist model)

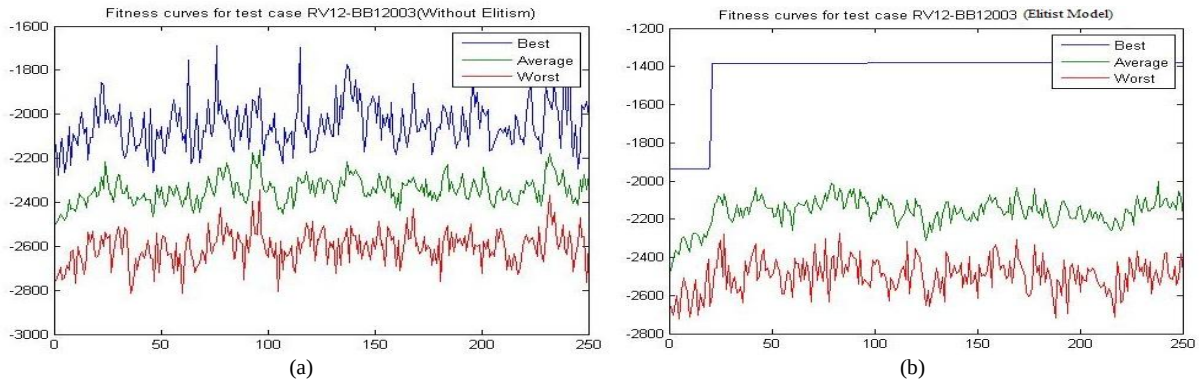
No. of Generations	RV11_BB11001	RV12_BB12003	RV11_BB11013	RV30_BB30027	RV11_BB11025	RV11_BB11029
1	-318	-1938	-797	-25115	-529	-629
5	-250	-1938	-769	-25115	-479	-583
10	-231	-1938	-748	-24862	-479	-583
25	-212	-1384	-710	-24809	-437	-552
50	-177	-1384	-710	-24603	-403	-552
100	-160	-1382	-689	-24255	-403	-546
200	-129	-1382	-677	-24255	-403	-493
250	-58	-1382	-656	-24255	-403	-493

From uneven ups and downs of the best fitness curve in Figure 4(a) and 5(a) it is clear that, the best value found in a particular generation may change in the successive generations but Figure 4(b) and 5(b) shows that the curve for the best fitness achieved in each generation is monotonically increasing i.e. we do not lose the best solution of each generation. The sequences of the test case RV11_BB11001 and the corresponding alignment by inserting gaps is given below

GKGDPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKGKFEDMAKADKARYEREMKTYIPPKGE
MQDRVKRPMNAFIVWSRDQRRKMALENPRMRNSEISKQLGYQWKMLTEAEKWPFFQEAQKLQAMHREKYPNYKYRPRRKAKMLPK
MKKLKKHPDFPKKPLTPYFRFFMEKRAKYAKLHPEMSNLDLTKILSKKYKELPEKKKMKYIQDFQREKQEFERNLARFREDHPDLIQNAKK
MHIKKPLNAFMLYMKEMRANVVAESTLKESAAINQILGRRWHALSREEQAKYYELARKERQLHMQLYPGWSARDNYGKKKKRKREK

GKGDPKKPR-G-K-MSSY-A-FF-VQTSR---EE-HKK-KHPDA-S---VNF--SEFSK--KCSERW-KT-MS-AKEKGK-FEDMAK-AD-KARYERE--MKTYIPPKGE
MQDRVKRPMN--AFI-V-W-S-RDQRR-KMALE-NP-RMRNSEISKQLGYQW-KM-L-TEAEK-WPFFQ-EA--QKL--Q-AMHREKY-PNY-KYRPRRKAKM-LP---K
MKKLKKHPDFPKKPLTPY-FRFFMEKRAK-YAKLHPEMSN-LDLT-KI-L-SKKYKEL-PEKKKM---KYIQDFQREKQEF--ERNL-AR--F-RED-HP-DLIQNA-KK
MHIK-K-P-LN-AFM--LYM-K-EMRANV-VAE--S-TLKESAAINQ-ILGRRW--HA-LSREE-Q-AKYYE-LARKER-QL-HMQLY-PGWSARDNYGKKKKRK-REK-

The alignment has been obtained at crossover probability 0.65 and mutation probability 0.06 with 30 number of population. The fitness value of the above aligned sequence is -55.



(a) (b)

Figure 4: Fitness curves for test case RV12_BB12003 (a) without elitism (b) elitist model

ISSN (Online) : 2319 - 8753
ISSN (Print) : 2347 - 6710

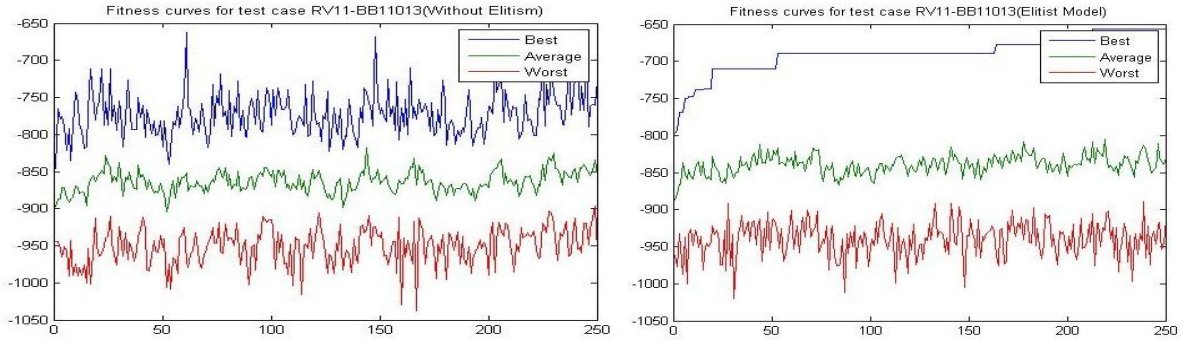
International Journal of Innovative Research in Science, Engineering and Technology

An ISO 3297: 2007 Certified Organization *Volume 4, Special Issue 9, July 2015*

National Conference on Emerging Technology and Applied Sciences-2015 (NCETAS 2015)

On 21st & 22nd February, Organized by

Modern Institute of Engineering and Technology, Bandel, Hooghly-712123, West Bengal, India



(a) (b)

Figure 5: Fitness curves for test case RV11_BB11013 (a) without elitism (b) elitist model

V. CONCLUSION

The current experiments show that GA itself is sufficient to solve the problem. Our algorithm uses modified GA with elitism trace sequence method for gap insertion with local search optimization for solving multiple sequence alignment problem.Some of the existing tools uses it for optimizing the functioning algorithm. Moreover it also helps to find the optimal pattern and fitness scores which may play a significant role in deducing evolutionary relationship among different organisms. Our algorithm is still fairly slow for very large test cases (for example >50 or so sequences). In future, it is desirableto design an optimization process with that of progressive approach for combining the speed and accuracy.

REFERENCES

[1] Lecture notes on 'Sequence Alignment' by Kun-Mao Chao, ofComputer Science and Information Engineering NationalTaiwanUniversity (2005).
[2] Wang L and Jiang T. On the complexity of multiple sequence alignment. IEEE Comput Biol, 4:337-48, 1994.
[3] C. Notredame and D.G. Higgins. "SAGA: sequence alignment by genetic algorithm", Nucleic Acids Research, volume 24(8): 1515–1524, 1996.
[4] Shyi-Ming Chen, Chung-Hui Lin, and Shi-Jay Chen, "Multiple DNA Sequence Alignment Based on Genetic Algorithms and Divide-and-Conquer Techniques", International Journal of Applied Science and Engineering (2005). 3, 2: 89-100
[5] http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html
[6] Jordi Bosch Pagans, Fernanda Strozzi, Jose-Manuel Zaldivar Comenges, "Beer Game Order Policy Optimization using Genetic algorithms", Liuc Papers n. 179, Serie Metodi quantitativi 17, suppl. a ottobre 2005
[7] Horng, J.T., Wu, L.C., Lin CM, and Yang, B.H. (2005). A genetic algorithm for multiple sequence alignment. Soft Computing, 9, 407-420.
[8] Otman Abdoul et. al., "Analyzing the Performance of Mutation Operators to Solve the Travelling Salesman Problem".
[9] C. Notredame, "Recent progresses in MSA a survey", pharmacogenomic, volume 3, pages 1–14, **2002**.
[10] Hyrum D.Carroll et.al **(2007):** "DNA reference alignment benchmarks based on tertiary structure of encoded proteins", Bioinformatics, 23(19) 2648– 2649. http://dna.cs.byu.edu/mdsas/download.shtml.'