International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013

DOI: **10.7321/jscse.v3.n8.1**

MSA-GA: Multiple Sequence Alignment Tool Based On Genetic Approach

Ruchi Gupta[1], Dr. Pankaj Agarwal[2], Dr. A. K. Soni[3]

[1,] *Sharda University, Grater Noida, India*
[2] *Gautam Buddha Technical University, Lucknow, India*
[3]*Sharda University, Grater Noida, India*
Email: [1]*80ruchi@gmail.com*, [2]pankaj_agwl@rediffmail.com, [3]ak.soni@sharda.ac.in

Abstract. **Genetic algorithms are relatively new optimization technique that can be applied to various problems, including those that are NP-hard. We have proposed a new web based tool MSA-GA (Multiple Sequence Alignment tool based on Genetic Approach) is developed for aligning the DNA sequences in order to find out the alignment of sequence. With this tool we have developed two different Genetic crossover and mutation operators for obtaining the better score. To validate the alignment, the fitness score function is compared with other standard tool. BaliBASE dataset have been considered for experimental work & analysis. The experimental results show that the proposed method tool is better than the existing method for dealing with multiple DNA sequence alignment. We conclude that the one such effort has been to produce by MSA-GA software, with its focus on facilitating the exploration and analysis of the DNA sequence variation from an evolutionary perspective.**

Keywords: *DNA Sequences, alignment, Genetic Algorithm, Crossover, Mutation, Multiple Sequence Alignment, Evolutionary algorithm etc.*

* Corresponding Author:
Ruchi Gupta,
Ph.D Research Scholar
H.no-132 Vivekkanand Nagar Ghaziabad UP, India
Email: 80ruchi@gmail.com Tel:+009212728196

1. Introduction

Multiple sequence alignment is an optimization problem that appears in many and diverse scientific fields. During the last decade, lots of increasing interest in the field of computational biology for methods that can efficiently solve this problem for sequences such as biological macromolecules, DNA and proteins. It was one of the most important and challenging tasks in computational biology because the time complexity for solving MSA grows exponentially with the size of the considered problem [1-2].To date, most of these methods follows either the dynamic programming approach or a tree-based approach. Multiple sequence alignment (MSA) uses optimally aligning of three or more sequences of symbols with or without inserting gaps. The objective is to maximize the number of matching symbols between the sequences and also use only minimum gap insertion, if gaps are permitted. All MSA algorithms also require an objective function to determine the relative quality of each possible multiple sequence alignment.
The fitness score objective function scores a MSA using the sum of the induced pair-wise alignments.e.g, considering the following simple MSA:

A: ACT-GGT

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

B: -CTGGGT
C: CCTGG-T

This alignment induces the following pair-wise alignments:

A: ACT-GGT
B: -CTGGGT

A: ACT-GGT

C: CCTGG-T

B: -CTGGGT
C: CCTGG-T

The fitness score objective function scores the MSA i.e. N as S (N) = $a<b$ s (a, b) where s (a, b) is the pair-wise score of sequences a and b from MSA N. All the multiple sequence alignment algorithms that are practical for realistic data sets and reflect real-world models of evolution are heuristic [3], and as such they do not guarantee an optimal alignment.

Multiple sequence alignment (MSA) algorithms can be classified into three classes: iterative, progressive and exact. Genetic algorithm is one of the useful tools determining alignment of multiple sequences. Iterative methods may be implemented through evolutionary approach that use computational heuristics based on natural biological phenomena such as selection, crossover and mutation to evolve a population of candidate solutions based on an objective function because they work similarly to progressive methods but repeatedly realign the initial sequences as well as adding new sequences to the growing MSA [4]. The first evolutionary algorithm using an adequate set of problem specific crossover and mutation operators was *SAGA* [5]. *SAGA* is a straightforward implementation of a genetic algorithm, which tries to optimize a weighted sum-of-pairs function with natural or quasi-natural affine gap penalties. It is used to optimize two different objective functions and shows that they can search large solution space efficiently. But due to repeated use of fitness function it may increase its time complexity.

One more evolutionary algorithm using the same chromosome representation as *SAGA* is *EP* [6]. This EA tries to optimise the following fitness function:

Fitness = SymbolScore-GapScore;

with *SymbolScore* being the overall score for the number of matched symbols of all columns and with *GapScore* being the number of all gaps of all columns. Additionally, it provides the possibility to use a heuristic for initialization by precomputing all pairwise alignments and merging them in a sequential manner (following a random permutation of all sequences).The genetic algorithm of Cai et al. [7] uses one crossover operator (One Point Crossover) and a heuristic called Local Gap 0-1 Alignment to optimize a sum-of-pairs fitness function. Additionally gaps are inserted at random positions for mutation. Another EA is *MSA EA* [8], which uses *CLUSTAL W* to create one chromosome of the initial population. The other chromosomes are randomly initialized. Altogether one crossover (RecombineMatchedColumns) and four mutation operators (Local Shuffle, Block Shuffle, GrowMatchedColumns, and CleanUp-GapColumns) are used for optimization. One of the most appropriate GA approaches to solve the MSA problem was presented by Nguyen et. al [9], however there are still some limitations w.r.t scoring scheme.

One more useful algorithm for multiple DNA sequence alignment using genetic algorithms and divide-and-conquer techniques [10] was proposed in which optimal cut points of multiple DNA sequences were selected. According to the author experimental results showed quite significant results. Approach involves cutting of the sequences for decreasing the space complexity for sequence alignment. However alignment was possible only for multiple deoxyribonucleic acid sequences, not for protein and other nucleic acid sequences.

Further new genetic algorithms [11] were used for solving the MSA in which various dataset were tested and the experimental results were compared with other methods. But after comparison it was observed that this approach could obtain good performance in the data sets with high similarity and long sequences.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

After that effective GARS approach [12] based on Genetic Algorithm with Reverse Selection was proposed. But it suffers from premature convergence in which solution reaches locally at an optimal stage. Furthermore a new approach AlineaGA [13] was proposed which used a genetic Algorithm with local search optimization embedded on its mutation operators for performing multiple sequence alignment. But its mutation probability leads to better solutions in fewer generations and that the mutation operators had a dramatic effect in this particular domain. Recently one more web based tool iMAGA [14] is developed for aligning the intron sequences in order to find the pattern. It has two modules (i) iExtractor/ iClassifier which extracts and classifies introns (ii) iAligner/ Pattern Finder which aligns the intron sequences and finds the Patterns.
Recently a new Cyclic Genetic Approach (CGA) [15] developed with the complete knowledge of the problem and its parameters. In CGA, the values of various parameters are decided based on the problem and fitness value obtained in each generation. But the column score value varies for each execution may not give relatively better alignment.
In this paper, we proposed a web based tools for obtaining alignments of multiple sequences using evolutionary approach. Experimental results show that the proposed solution does offer better fitness accuracy rates w.r.t some existing tools.

2. Proposed MSA-GA Tool

Firstly we propose a GA-based model to solve the MSA as shown in Fig. 1. In order to reach quality solutions, we developed MSA-GA tool which contain two methods (i) single method with random number generation (ii) trace sequence algorithm for gap generation, selection of sequence on the bases of weighted factor, window frame crossover and space combined mutation operator.
The purpose of second mechanism is used to refine the alignment into one that is better than previous result while preserving its local schema.

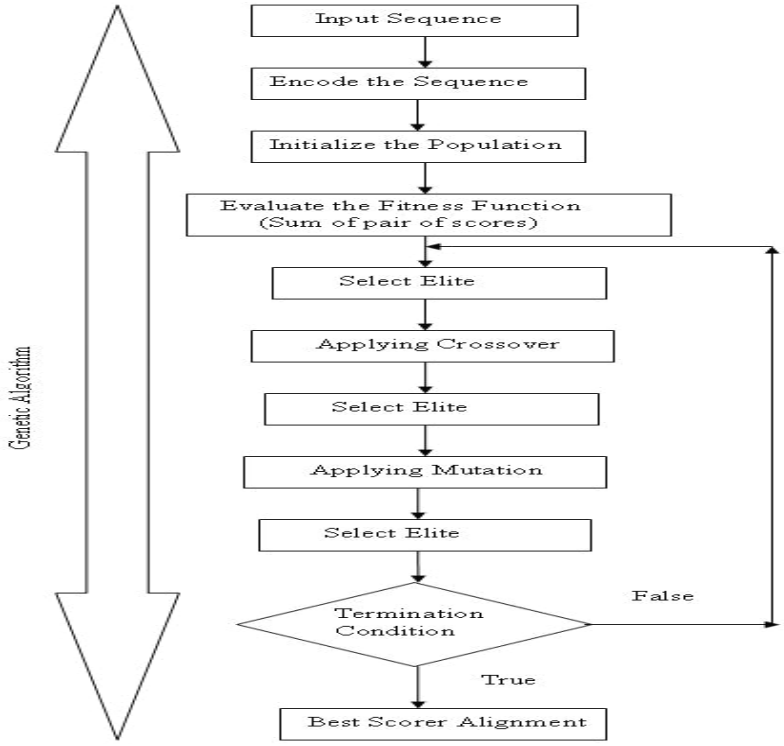

Figure 1: Evolutionary Genetic Model

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

2.1 MSA-GA: Multiple Sequence Alignment tool Based on Genetic Approach

Various MSA tools are available to for aligning multiple sequence of DNA, RNA and proteins. The proposed tool is developed specially for aligning the DNA. MSA-GA tool is divided in to two main modules. The first module applying random gap generation process, evaluation of fitness function, single crossover and adjusting spaces within some specific region is triggered when the constraint is met. The another second module used trace sequence algorithm for gap generation, selection of sequence on the bases of weighted factor, window frame crossover and space combined mutation operator .The purpose of second mechanism is used to refine the alignment into one that is better than previous result while preserving its local schema .The core module of a tool is properly aligned and compare the fitness value of aligned sequence using evolutionary genetic approach. Figure 2 shows the flow of input param of various standard and DNA sequence that can be taken as input in to the software.

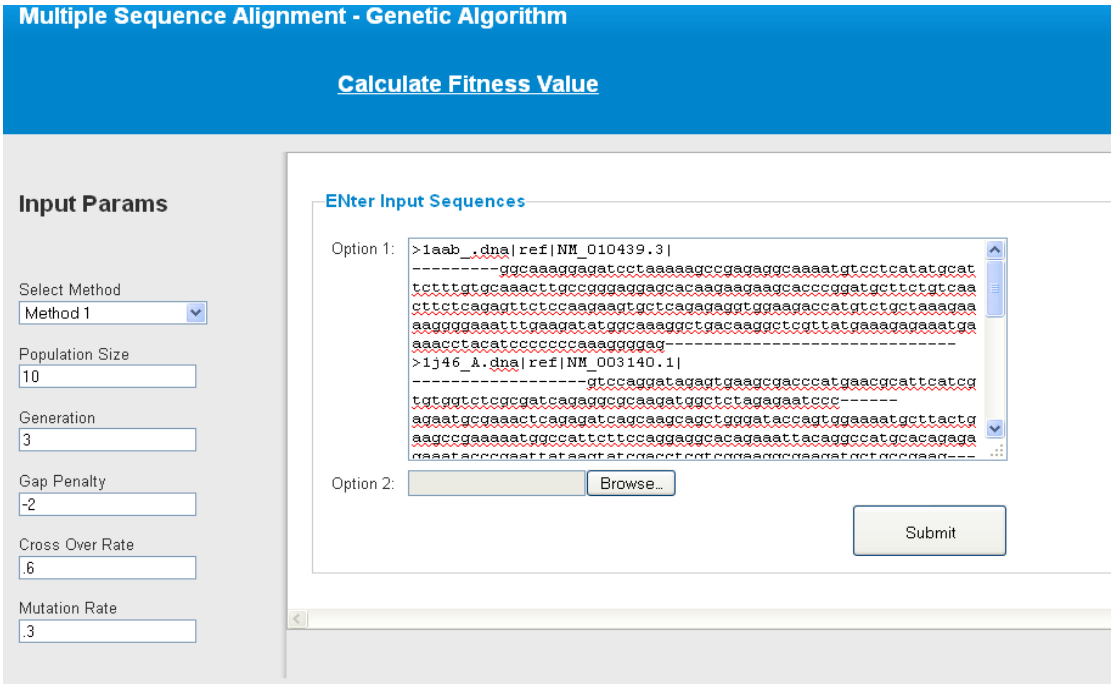

Figure 2. Screenshot of multiple sequence alignment-genetic algorithm (MSA-GA). The right pane shows BAliBASE format sequences to be aligned. The left displays the input param for sequence. The MSA-GA Settings window allows selecting the run parameters of the GA and the alignment settings.

A. Chromosome Encoding

The most popular way of encoding is a binary string. Each chromosome has one binary string. Each bit in this string can represent some characteristic of the solution or the whole string can represent a number. Each sequence has its own length. The number of gaps in the sequence is to be inserted in each sequence. It is calculated in a way that the length of all sequence remains the same. Therefore we have to generate the maximum length of sequence by multiplying the maximum length of particular element of sequences with $r_{sp}1.2$. Let's say we have a set of sequence S = {S1, S2, S3 ….Sn}. So the maximum length of the column has to be found out by multiplying the sequence with r_{sp} by maximum length column. The value of scaling factor r_{sp} defines that the alignment to be 20% longer then the sequence which is based on the observations that solution to common MSA problem really contains more than 20% gaps.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

A multiple sequence alignment (MSA) of A is also obtained by inserting gaps ('-') into the original sequences such that all resulting sequences A*i have equal length L ≥max {ni | i = 1, . . . , r}, Figure 3 shown the matrix that we can get back the sequence Ai by removing all gaps from A*i , and no column consists of gaps only:

$$\mathbf{A}^* := \begin{cases} A_1^* = (a_{11}^*, a_{12}^*, \ldots, a_{1L}^*) \\ A_2^* = (a_{21}^*, a_{22}^*, \ldots, a_{2L}^*) \\ \vdots \\ A_r^* = (a_{r1}^*, a_{r2}^*, \ldots, a_{rL}^*), \end{cases}$$

$$\mathbf{A}^* := \begin{cases} A_1^* = (a_{11}^*, a_{12}^*, \ldots, a_{1L}^*) \\ A_2^* = (a_{21}^*, a_{22}^*, \ldots, a_{2L}^*) \\ \vdots \\ A_r^* = (a_{r1}^*, a_{r2}^*, \ldots, a_{rL}^*), \end{cases}$$

Figure 3: Equal length of sequence matrix

Our goal is to characterize all legal traces. So firstly we have trace all the sequence with same pattern and put these symbols in same column then insert the gap in that place where the same symbol is not matched. Therefore we have to generate the maximum length of sequence by inserting the spaces in a matrix. Suppose we are given three sequences *a*1 = A G C T, *a*2 = A G T and A C T. Figure 4 shows the trace sequence method for inserting the gaps in the sequence.

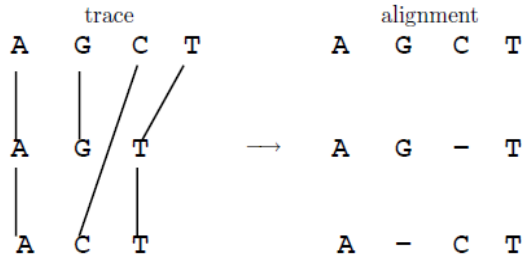

Figure 4: sequence after inserting the gaps by trace algorithm

Our possible solutions are obviously just numbers, so our representation is simply the binary form of each number. For example, one can encode directly integer or real numbers; sometimes it is useful to encode some permutations and so on. Each sequence has its own length. The number of gaps in the sequence is to be inserted in each sequence.

B. Evaluation of Fitness Function

To evaluate their fitness scores, we evaluate two functions for the chromosomes. Firstly we scored each column by looking at matches, mismatches, and gaps in the two sequences. We assume that a match = 1, a mismatch = 0, and a gap = -1. The fitness or scoring function of each individual is calculated by the formula has been shown by figure 5:-

$$Fitness\_Score = \sum_{i=1}^{p-1} \sum_{j=j+1}^{p} ScoringMatrix(A_i, A_j)$$

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

Figure 5: Fitness score Function

Secondly from figure 6 the fitness Score for each alignment is calculated by summing the individual score for each column in the matrix. Scoring matrix is needed to determine the cost of aligning a residue with another. Also, a gap penalty value must be settled for determining the cost of aligning an amino acid with a gap. Secondly we scored each column by looking at matches, mismatches, and gaps in the two sequences. We assume that a match = 1, a mismatch = 0, and a gap = -2. Wij is considered as weight that may be strong biases in the sequence set.

$$Fitness\_Score = \sum_{i=2}^{N} \sum_{j=1}^{i-1} WijCost(Si,Sj)$$

Figure 6: Fitness score Function

C. Selection Procedure

The proposed method use two popular section strategies for obtaining the fitness score of the population.

(i) Roulette Wheel Selection method where fitness level is used to associate a probability of selection with each individual solution.
- We first calculate the fitness for each input and then represent it on the wheel in terms of percentages.
- In a search space of 'N' chromosomes, we spin the roulette wheel.
- Chromosome with bigger fitness will be selected more

(ii) Tournament selection procedure for finding out the best solution. Two solutions are picked out of the pool of possible solutions, their fitness is compared, and the better is permitted to reproduce.
- Deterministic tournament selection selects the best individual in each tournament.
- Can take advantage of parallel architecture

D. Crossover

Single Point Crossover

In the single point crossover process, Crossover selects sequence from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent .we select crossover point at the rate of 0.5 and count the entire gap in each population then multiply it with crossover rate and take ceiling of crossover rate. The crossover point is selected by the formula:-

$$Crossover..point = total...no...of\cdots gaps \times 0.5$$

After selecting point, copy the chromosome of first parent exact at the crossover point value then copy all chromosome of second parent and vice versa [6] so there are two offspring has to be generated after applying the crossover function. Calculate the fitness score of current population and select the best individual for performing crossover operation. The flow of one point crossover is shown in figure 7.

```
CrossOver (b,p)
{
//let cr be the cross rate
//let crosspoint be the cp
```

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

```
//cp=sequencecount+cr
//pick an array b[] in the range crossover from random
//Declare p as point =b[]
for i=1 to  Do cp
            {
            if b[i]=p[p+cp]Go to first step
            else
            b[p+cp]=p[p+cp-i]
            }
//let position of matrix pos
pos=b[i]
if b[i]>max_length
max_length +=max_length
pos=pos-(max_length*(sequencecount-1))
}
```

Figure 7: flow of one point crossover

Window frame Combine Crossover Operator

In window frame combine operator, both parents are selected, from the middle of 20% of the parent generation. For window frame combine crossovers, each parent is divided into three parts. The different part from these parents are then exchanged and merged together to generate two new individuals. However, the better one will be taken as a child. The crossover is implemented in two steps as described below.

Step 1: To create a 20% window frame from the overall length of both parents and calculate the score scores of the first 20% of columns for both parents. The parent having the better score is divided vertically at that column. The other parent is divided using the same mechanism.

Step 2: We now have cut the 20% window frame of first parent and it combine with second to create new individual and vice versa. To complete the crossover, the middle part of both parents are to be exchanged and then all three pieces are merged together to generate two new individuals. MSA-GA display fitness score after applying crossover operator in figure 8.

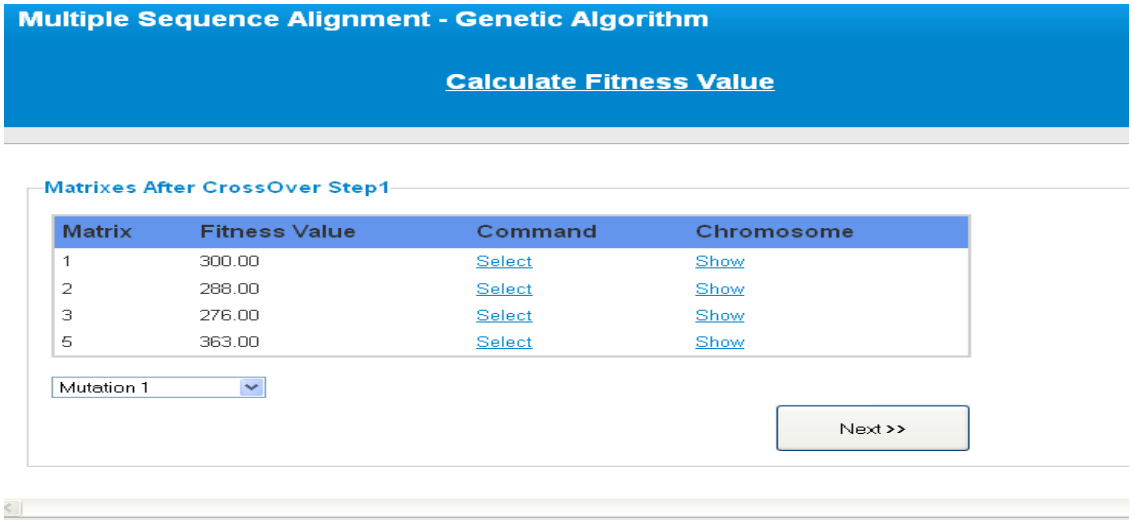

Figure 8: The MSA-GA displays fitness scores after performing crossover operator

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

E. Mutation

After a crossover is performed, mutation takes place. This is to prevent falling all solutions in population into a local optimum of solved problem. The system randomly chooses a gene of a chromosome form the mating pool randomly and applying binary mutation. Mutation changes randomly the new offspring. For binary encoding we can switch a few randomly chosen bits from 1 to 0 or from 0 to 1 [7]. where all the gaps are represented by 0's and all the base symbols are represented by 1's and mutation takes place separately in each sequence up to the mutation point rate of 0.2 [7] is initialized and corresponding mutation point is selected. The mutation point is to be selected by the formula:-

$$Mutation..point = 0.2 \times total...length...of...bit \cdots string$$

First the mutation operator converts the total sequence in to bit string then calculate the mutation point after calculating the mutation point every picks a random amino acid from a randomly chosen row (sequence) in the alignment and checks whether one of its neighbors has a gap. If this is the case, the algorithms swap the symbols. The flow of one point crossover is shown in figure 9.

```
Mutation (mp,i)
{
//Declare mutation point mp
//declare sequence row count count
mp=count* mutation rate
//declare string symbol
 for i=1 to mp do
            {
            symbol=removeSymbols.SubString(i,1)
             if(matrix row='-')
            matrix row =symbol
            else
             matrix row = '-'
            }
}
```

Figure 9: flow of space mutation

Combine Space Mutation Operator

The purpose of the combine Space operator is to merge two or three spaces together. In the combine space all the parent chromosomes after crossover. The flow of one point crossover is shown in figure 10. MSA-GA display fitness score after applying mutation operator in figure 11.After completing all mutation and crossover operator final fitness score is to be obtained using MSA-GA tool which can be displayed in figure 12.

```
Mutation()
{
// Get 20% percent of matrix for crossover
// Get value when data in table
 // Declare string blankvalue,value,totalvalue
 // Get 20% coloumn data as 2p
for i=1; to 2p
 blankvalue = Get all blank value
 value = Get all withoutblank value
 if(2p==i)
 totalvalue = blankvalue + value;
 array[k1]=totalvalue
```

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

```
foreach k in array[k1]
j++;
for ij=1 to 2p
if(j==ij)
// Value set in table
}
```

Figure 10: flow of combine space mutation

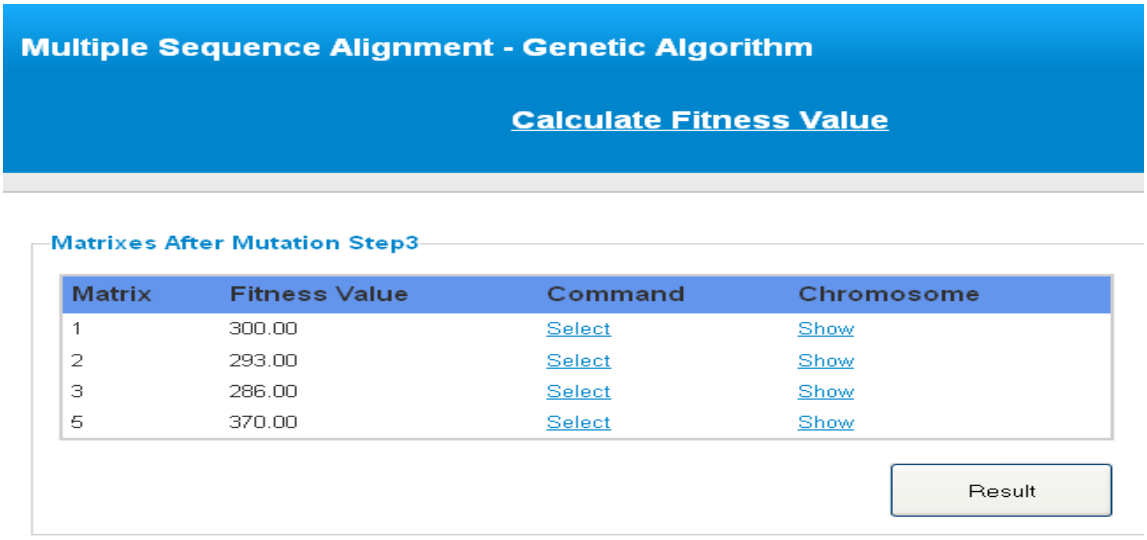

Figure 11: The MSA-GA displays fitness scores after performing mutation operator

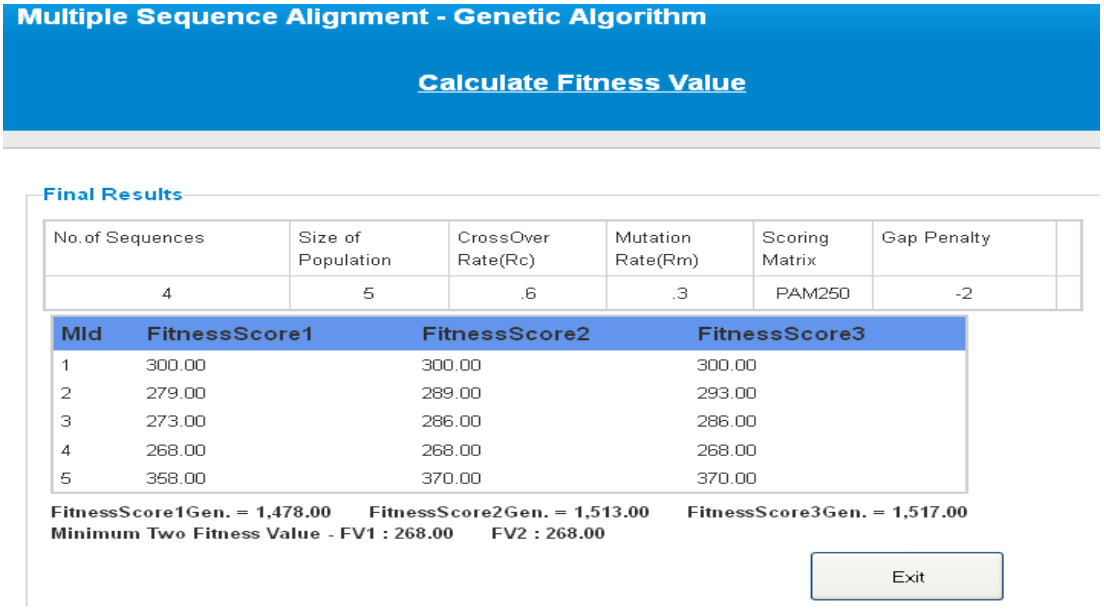

Figure 12: The MSA-GA displays fitness scores final results

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: 10.7321/jscse.v3.n8.1

3. Implementation

The Web based Tool MSA-GA is implemented using Microsoft visual studio and the machine for this research is a personnel computer with Intel Pentium III processor .The main memory is 4 gigabyte and Microsoft XP was used as a platform for the implementation. We test our algorithm with seven datasets from the Reference 1 alignments of BAliBASE [16] as the benchmark for the experiment. BaliBase is a standardized set of benchmark reference 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 are shown in Table 1.
Seq ID = sequences identification, No's = Total number of sequences.

Table1:BAliBASE_mdsa_alland BAliBASE_mdsa_100s Dataset

Seq ID	Sequence Specification	No of Sequence
RV11_BB11025	>1tvx_A.dna|gb|M88539.1|SYNCLCTAPA >1prt_F.dna|emb|BX640422.1| >1sap_.dna|gb|CP000077.1| >1lt5_D.dna|gb|DQ778054.1|	4
RV12_BBS12009	>1csp.dna|emb|Z99108.2|BSUB0005.revcom >CSPD_HAEIN.dna|gb|CP000057.1| >GR2B_ARATH.dna|ref|NM_127676.2| >CSPF_ECOLI.dna|gb|U00096.2| >PIPI_HUMAN.dna|ref|NM_014460.2|	5

We analyzed the results of our approach by comparing the alignments produced by MSA-GA tool with those obtained through other leading alignment techniques including ClustalW, DiAlign, TCoffe and Maft. The results of these programs are collected by running their programs on our system. To estimate the biological quality of each alignment, we have measure SPS (Sum of pair score) given by the program *Bali Score* and the core bloc annotation file which are available on the BALiBASE site. Moreover, we have performed another test in order to measure the accuracy of our approach in terms of secondary structure information. Table 2 summarizes the performance of different alignment methods including our approach, CLUSTAL, DIALIGN, Tcoffe and Maft upon structural DNA. The results state clearly that our approach can be useful in the alignment of sequence. In the most cases, it ranks high on SCI results.

Table2: Comparison of MSA-GA with standard tools

SP_Score	RV11_BB11025	RV12_BBS12009
ClustalW	324	101
DiAlign	118	87
Tcoffee	356	128
Mafft	284	185
MSA-GA	268	116

4. Conclusion

Multiple alignments of biological sequences provide a valuable source of information for investigating the properties, characteristics, and functions of novel sequences. MSA-GA is an evolutionary tool to this assignment that uses a GA with enhanced exploitation capabilities present on its crossover and mutation operators. The algorithm's robustness is improved by the use of a new crossover operator and by adopting an elitist strategy that combines the preservation of the fittest individuals with the diversity originated by the exploring capability of its crossover operators.

International Journal of
Soft Computing And Software Engineering (JSCSE)
e-ISSN: 2251-7545
Vol.3,No.8, 2013
Published online: Aug 25, 2013
DOI: **10.7321/jscse.v3.n8.1**

5. Acknowledgement

We would like to thank Dr. Pankaj Agrawal and Dr A.K Soni for his contribution to the design work of MSA-GA software.

References

[1] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts, J. Watson, "Molecular Biology of the Cell" ,Garland Publishing, Inc., 1994.
[2] J.D. Thompson, J.C. Thierry, O. Poch, RASCAL "rapid scanning and correction of multiple sequence alignments", Bioinformatics 19 (9) (2003) 1155–1162.
[3] Vision, T & McLysaght, A 2004, "Computational Tools and Resources n Plant Genome Informatics" ,in P Christou & H Klee (eds), *Handbook of Plant Biotechnology*, 1 edn, John Wiley & Sons., p. 1552.
[4] Chengpeng Bi, "Deterministic local alignment methods improved by a simple genetic algorithm" , ELESVIER Neurocomputing 73 (2010) 2394–2406
[5] C. Notredame and D. Higgins, "SAGA: Sequence alignment by genetic algorithm", *Nucleic Acids Research*, 24(8):1515–1524, 1996.
[6] K. Chellapilla and G. Fogel, "Multiple sequence alignment using evolutionary programming" In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 445–452, 1999.
[7] L. Cai, D. Juedes, and E. Liakhovitc, "Evolutionary computation techniques for multiple sequence alignments" In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 829–835, 2000.
[8] R. Thomsen, G. Fogel, and T. Krink, "A Clustal alignment improver using evolutionary algorithms" ,In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 121–126, 2002.
[9] Nguyen HD, Yamamori K, Yoshihara I, Yasunaga M, "Improved GA-based method for multiple protein sequence alignment", *The 2003 Congress on Evolutionary Computation (CEC '03)* 2003, 3:1826 - 1832.
[10] 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
[11] Jorng-Tzong Horng, Li-Cheng Wu, Ching-Mei Lin, Bing-He Yang, "A Genetic Algorithm For Multiple Sequence Alignment", Soft Computing-A Fusion of Foundations, Methodologies and Applications, Vol. 9, Issue 6, pp 407 – 420. (2005)
[12] Yang Chen, Jinglu Hu, Member, IEEE, Kotaro Hirasawa, Member, IEEE, Songnian Yu. "Multiple Sequence Alignment Based on Genetic Algorithms with Reserve Selection" , ICNSC, pp 1511-1516 (2008).
[13] Fernando José Mateus Silva, *Member, IEEE*, Juan Manuel Sánchez-Pérez, Juan Antonio Gómez-Pulido and Miguel A. Vega-Rodríguez , "An Evolutionary Approach for Performing Multiple Sequence Alignment", 978-1-4244-8126-2/10/$26.00 ©2010 IEEE
[14] V.Amouda,VSeraj,S.Kuppuswami,S.Buvanswari, "iMAGA:Intron Multiple sequence alignmentnusing genetic algorithm", International Journal of Engineering Science and Technology Vol. 2(11), 2010, 6361-6370
[15] Amouda Nizam,Jeyakodi Ravi1, and Kuppuswami Subburaya, "Cyclic Genetic Algorithm for Multiple Sequence Alignment", International Journal of Research and Reviews in Electrical and Computer Engineering (IJRRECE) Vol. 1, No. 2, June 20
[16] 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.'