



Two New Genetic Algorithm Based Methods for Obtaining Alignment of Multiple Sequences

Ruchi Gupta¹, Dr. A. K. Soni², Dr. Pankaj Agarwal³

¹ Assistant Professor, AKGEC Ghaziabad, India, Ph.D Research Scholar

² Professor & Head, Department of Computer Science & Engineering,
Sharda University, Greater Noida, India

³ Professor & Head, Department of Computer Science & Engineering,
IMS Engineering College Ghaziabad, India

Abstract: Multiple Sequence Alignment (MSA) is considered as one of the computationally challenging problem in the field of molecular biology. In the recent past, Genetic Algorithm (GA) based solutions have emerged as useful tool for solving MSA problem. Here we have proposed two new GA based methods to target the MSA problem. This is in continuation to our earlier published work [1] on solving MSA using GA based approach. BALiBASE dataset has been considered for experimental work & analysis. We have also compared for the accuracy of alignment scores with our previous proposed solutions as well as with few standard & known MSA tools like CLUSTAL-W, DALIGN, T-COFFEE. Experimental results have shown some encouraging results in terms of obtaining better alignment scores using our proposed solution.

Keywords: DNA Sequences, alignment, Genetic Algorithm, Crossover, Mutation, Selection, Multiple Sequence Alignment etc.

I. Introduction

Multiple sequence alignment (MSA) can be seen as a generalization of Pairwise Sequence alignment where instead of aligning two sequences, n sequences are aligned simultaneously, where n is > 2 . It arranges a set of sequences in a scheme where positions believed to be homologous are written in a common column. In one approach, one simply adds the scores of all the induced pairwise alignments contained in a multiple alignment [2]. For a linear gap penalty this amounts to scoring each column of the alignment by the sum of the amino acid pair scores in this column. The corresponding score is called the sum of pairs (SP-) score. Although it would be biologically meaningful, the distinctions between global, local and other forms of alignment are rarely made in a multiple alignment. In particular, pairwise alignments can be merged when they align one sequence to all others, when a linear order of the given sequence is maintained, or when sequences pairs with pairwise alignments form a tree. While all these schemes allow for the ready definition of algorithms that output multiply aligned sequences, they do not include any information stemming from the simultaneous analysis of several sequences. An example of an alignment of four hypothetical DNA sequences is shown in figure 1.

```
- G C T G A T A T A G C T
G G G T G A T - T A G C T
- G C T - A T - - C G C -
A G C G G A - A C A C C T
```

Figure1: An Example of an Alignment

The basic idea is that the sequences are aligned on top of each other, so that a coordinate system is set up, where each row is the sequence for one DNA, and each column is the 'same' position in each sequence. Each column corresponds to a specific residue in the 'prototypical' DNA [3].

Genetic algorithms appear to find good solutions for the multiple sequence alignment problems [4], however it depends very much on the way the problem is encoded and which crossover and mutation methods are used. The different forms of encoding, crossover and mutation that we have seen so far can be combined to give various genetic algorithms that can be used to solve the multiple sequence alignment problems.

The method works by breaking a series of possible MSAs into fragments and repeatedly rearranging those fragments with the introduction of gaps at varying positions. This paper also explores the possibility of applying GA based solution for MSA problem. One such proposed & developed solution is also presented.

II. Proposed Approach

We propose Genetic Oriented Hybrid Model to solve the MSA as shown in Figure 2. In order to reach quality solutions, we developed two new approaches to aim at improving the quality solution. The first approach 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 approach 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.

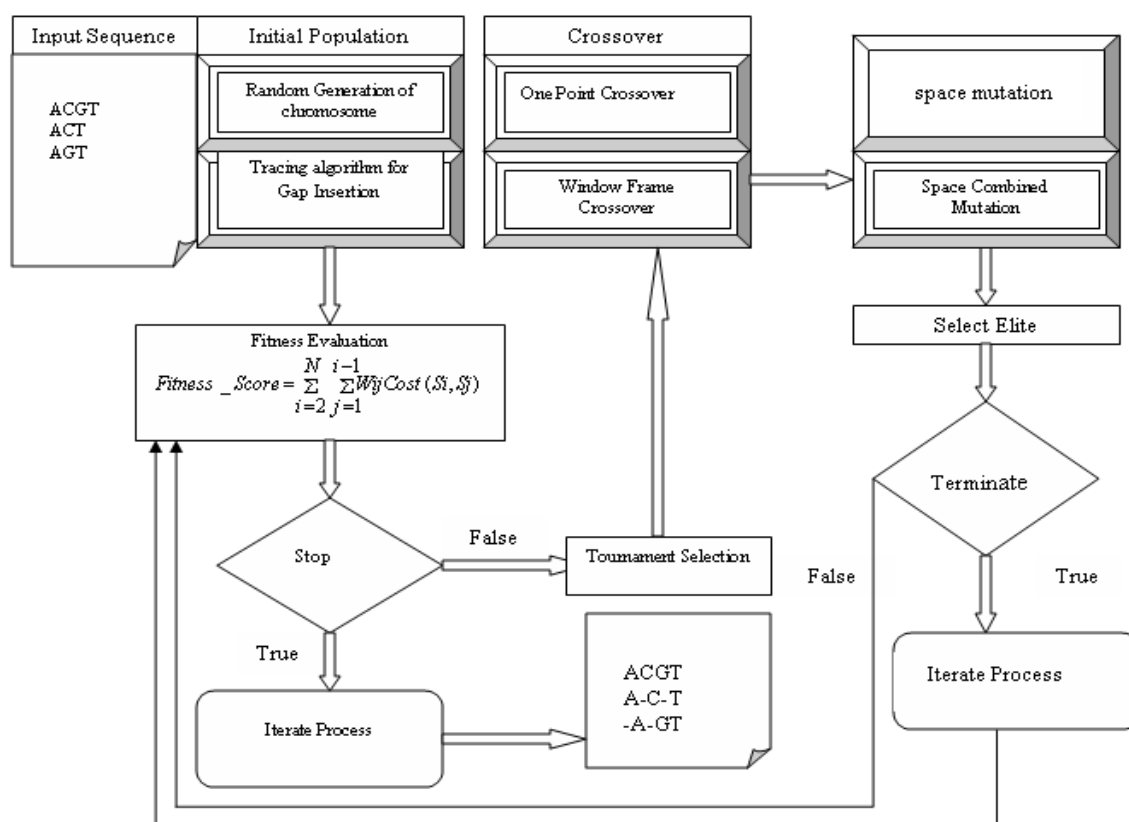


Fig 2: Genetic Oriented Hybrid Model

Approaceh-1

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} . Let's say we have a set of sequence $S = \{S_1, S_2, S_3 \dots S_n\}$. 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.

B. Evaluation of Fitness Function

To evaluate their fitness, the chromosomes must be converted to the alignment form to be applied sum-of-pairs function [5]. 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:-

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

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.

C. Selection Procedure

After calculating the fitness score of all the population applying 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 times.

D. 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:-

$$Crossover \dots point = total \dots no \dots of \dots gaps \times 0.5$$

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 mutation operation.

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 from 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:-

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

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.

Approach-2

A. Chromosome Representation

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

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

Figure 3: Equal length of sequence matrix

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.

In the following example depicted in figure 4, we will consider a novel approach to computing an MSA based on “integer linear programming”. Suppose we are given two sequences $a1 = A G C T$ and $a2 = A G T$

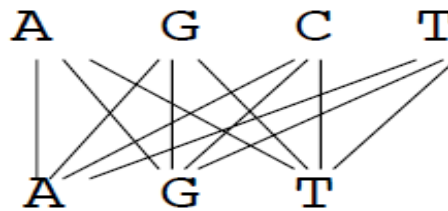


Figure 4: Equal length of sequence matrix

The set of realized edges is called the *trace* of the alignment. An arbitrary subset $T \subseteq E$ of edges is called a *trace*, if there exists some alignment that realizes precisely the edges in E . 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 $a1 = A G C T$, $a2 = A G T$ and $a3 = A C T$ that are shown in figure 5.

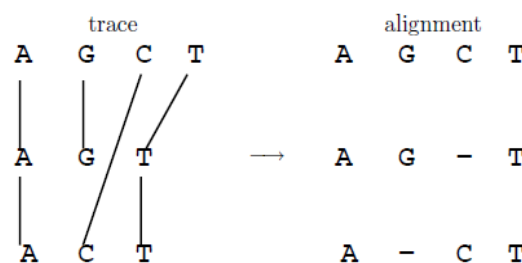


Figure 5: matrix after inserting the space

The encoding process is often the most difficult aspect of solving a problem using genetic algorithms. When applying them to a specific problem it is often hard to find an appropriate representation of the solution that will be easy to use in the crossover process.

B. Evaluation

We use the evaluation function to decide how 'good' a chromosome is. The evaluation function usually comes straight from the problem. We assume that a match = 1, a mismatch = 0, and a gap = -2. W_{ij} is considered as weight that may be strong biases in the sequence set.

$$Fitness_Score = \sum_{i=2}^N \sum_{j=1}^{i-1} W_{ij} Cost(S_i, S_j)$$

The fitness Score for each alignment is calculated by summing the individual score for each column in the matrix. After calculating the overall score of matrix, it is multiplied by the weight factor W_{ij} . 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. This penalty is only employed when aligning a residue with a gap.

C. Selection Procedure

In this case we are applying the 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. 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.

E. Combine Space Mutation Operator

The purpose of the combine Space operator is to merge two or three spaces together. In the combine space operator we combine space of 20% rows of all the parent chromosomes after crossover.

III. Implementation and Test configuration

The algorithm 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 [8] 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, Nos = Total number of sequences.

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
RV11_BB11001	>1aab_.dna ref NM_010439.3 >1j46_A.dna ref NM_003140.1 >1k99_A.dna ref NM_001044383.1 >2lef_A.dna ref NM_010703.2	4
RV_11BB11024	>1evk_A.dna gb U00096.2 .revcom >1h4q_A.dna gb AF384553.1 AF384553 >1qe0_A.dna gb CP000255.1 .revcom >1ser_A.dna gb AE017221.1 .revcom	4

Table1:BALiBASE_mdsa_alland BALiBASE_mdsa_100s Dataset

IV. Experimental Results

In order to examine our algorithm validity, we test number of series with DNA sequence. In our first approach, the number of generations, the population size, the crossover rate and the mutation rate are 100, 10, (0.8, 0.6, 0.3) and (0.5, 0.3, 0.1) respectively. We have calculated the fitness score of each one of these datasets. For each of four datasets, the algorithm was executed for 10 independent runs. For each dataset, the best score out of the 100 runs was recorded and the corresponding BALIScore was reported in Table 2 and corresponding plots are to be presented in figure 6. Our algorithm has to be run with 30% crossover & 10% mutation option, 60% crossover & 30% mutation option and 80% crossover & 50% mutation option .it is observed that our algorithm

obtained the best solutions for 80% crossover & 50% mutation option .The solutions obtained by the 60% crossover and 30% mutation for the same datasets are close to the best scores, however the option 30% crossover & 10% mutation has not achieved any good quality solutions.

Name of dataset	Cros rate=30% Mut rate=10%	Cros rate=60% Mut rate=30%	Cros rate=80% Mut rate=50%	Pop Size	Average computational Time
RV11_BB11003	169.00	168.00	180.00	10	2:05
RV11_BBS11004	210.00	215.00	234.00	15	3:07
RV11_BB11021	314.00	317.00	327.00	20	4:08
RV11_BBS11028	340.00	350.00	367.00	30	6:01

Table 2: fitness scores with selected Crossover and mutation rate options

The basic concept of second approach is to insert the gap with trace sequence algorithm rather than random generation of gaps after that it used window frame crossover with two parents are selected, one from the top 50% and another from the bottom 50% of the parent generation. For window frame crossover, each parent gets middle 20% and create 20% frame window for crossover operation. The middle 20% piece 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 purpose of the combine Space operator is to merge 20 % spaces together. In order to examine our algorithm validity, we will compare the fitness scores of approach1 with approach 2 by taking BALiBASE_mdsa_100s Dataset; we observed that fitness score calculated by approach2 is having significant amount of improvement while increasing the number of population, running time is also increased accordingly and indicate by a notable rise about 100% after increasing each size of population. A more comprehensive test needs to be carried out to prove our algorithm. It will require a larger dataset to prove that second approach will produce better results. Table 2 and figure 7 shows comparison of the evolution of the fitness of the best individual and the average fitness of the entire population throughout the generations.

The last set of experiments we analyzed the results of our approach by comparing the alignments produced by our algorithm with those obtained through other leading alignment techniques including ClustalW, DiAlign, TCOFFEE and MAFFT. 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 3 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. Results are shown in table 3:-

SP_Score	RV11_BB11025	RV12_BBS12009
ClustalW	324	101
DiAlign	118	87
Tcoffe	356	128
Maft	284	185
Our Algo	183	116

Table 3: Fitness value comparison of different method with our algorithm

V. Conclusion

The current experiments showed that GA itself is sufficient to solve the problem. Our algorithm uses modified GA with newly designed mutation and operator and and trace sequence method for gap insertion with local search optimization for solving multiple sequence alignment process where as some of the existing tools uses it for optimization of functioning algorithm. Moreover it also finds the optimal pattern and fitness scores which may have significant role in deducing evolutionary relationship among different organisms. Our algorithm is still fairly slow for large test cases (e.g. with >50 or so sequences and 100 generation). In future, it is desirable

to use an optimization process with that of progressive approach in order to combine the speed of the former with the accuracy of the latter.

Reference

- [1] Ruchi Gupta ,Dr. Pankaj Agarwal , Dr. A. K. Soni, “Genetic Algorithm Based Approach for Obtaining Alignment of Multiple Sequences”, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 3, No. 12, **2012**
- [2] C. Gondro and B.P. Kinghorn, “A simple genetic algorithm for multiple Sequence alignment” , Genetics and Molecular Research 6 (4): 964-982 (**2007**)
- [3] J.D. Thompson, J.e. Thierry, O. Poch. (**2003**). “RASCAL: rapid scanning and correction of multiple sequence alignments”, Bioinformatics, Vol. 19, No. 9.
- [4] Jorng-Tzong Horng, Ching-Mei Lin, Baw-Jhiune Liu, Cheng-Yen KaoUsing “Genetic Algorithms to Solve Multiple Sequence Alignments”
- [5] C. Notredame, “Recent progresses in MSA a survey”, pharmacogenomic, volume 3, pages 1–14, **2002**.
- [6] Amouda Nizam,Jeyakodi Ravi1, and Kuppuswami Subburaya2, “Cyclic Genetic Algorithm for Multiple Sequence Alignment”, International Journal of Research and Reviews in Electrical and Computer Engineering (IJRRECE) Vol. 1, No. 2, June **2011**
- [7] 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
- [8] 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>.’