International Journal of Engineering & Technology, 7 (4.5) (2018) 481-484

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

SOGA: space oriented genetic algorithm for multiple sequence alignment

Ruchi Gupta [1] *, Pankaj Agarwal [2]

[1] AKGEC Ghaziabad, Dr. A.P.J. Abdul Kalam University
[2] IMS Ghaziabad, Dr. A.P.J. Abdul Kalam University
**Corresponding author E-mail: 80ruchi@gmail.com*

Abstract

Multiple sequence alignment is one of the recurrent assignments in Bioinformatics. This method allows organizing a set of molecular sequences in order to expose their similarities and their differences. Although several applicable techniques were observed in this re- search, from traditional method such as dynamic programming to the extent of widely used stochastic optimization method such as Simu- lated Annealing and motif finding for solving this problem, their use is limited by the computing demands which are necessary for ex- ploring such a large and complex search space. This paper presents a new genetic algorithm, namely SOGA (Space Oriented Genetic Algorithm for Multiple Sequence Alignment), which has two new mechanisms: the first generates the population with randomly inserting the space between the selected sequences and the second applying new crossover and mutation operator, within an iterative process, to generate new and better solutions. This method is simple and fast. Its performance will further be tested on standard benchmark databas- es and will be compared with well-known algorithms. However, as its solutions clears that there is scope for further improvement.

Keywords: *Genetic Algorithm; Crossover; Mutation; Selection; Multiple Sequence Alignment.*

1. Introduction

One of the most recognized issues in biological information pro- cessing of computer science is sequence comparison. Bimole- cular sequences for computer scientist are another form of a data source. The Multiple Sequence Alignment (MSA) is a fundamen- tal approach to examine a set of sequence. The procedure of diffe- rentiating various sequences for matching maximum characters is re- ferred as a multiple alignment of protein or DNA sequence [1]. Although several applicable techniques were observed in this re- search, from the traditional methods (such as dynamic program- ming) to the extent of widely used stochastic optimization me- thods (such as Simulated Annealing and motif finding) for solving this problem, their use is limited by the computing demands which are necessary for exploring such a broad and complex search space. It can be noted that if more sequences are added to the input se- quence, then the problem complexity increases. Therefore, the clas- sical methods such as DP and Needleman's algorithm [2] are re- stricted to take into account a few short sequences. Bearing in mind that these requirements are conflicting, an optimal alignment solu- tion cannot always be found Genetic Algorithms are adaptive search methods, which perform well in large and complex areas.

This paper shows that Genetic Algorithms can be used to solve the multiple Sequences Alignment problems.

A GA based approach, known as SOGA, is being discussed in this paper. This approach begins with a random initialization of popu- lation. Initially, a population is initialized and then chromosomes are encoded Isokawa et al. [3] and Wayama et al. [4]. The encoded chromosomes have a bit matrix comprising of 0 and 1, where the position of 1 represents to a space, and the position of 0 relates to a base symbol. Encoding chromosomes is a concept similar to the bit matrix but space positions are just recorded here. To provide for mutation and crossover operators and guiding the spaces to the ap- propriate position, additional information for each chromosome is maintained apart from the space positions of the actual chromo- somes of the population. The computational efficiency is contri- buted by the choice of operator and parameter. This is done by min- imizing the time required to find a solution in a large space on the basis of empirical values of previous calculation. To observe the effect of the proposed mechanisms and to test the proposed algo- rithm, several experiments were conducted using a benchmark da- tabase . Also, to compare GAMS with some of the well-known methods, namely SAGA [5], MSA-GA [6], SGA [7], CLUSTAL W [8], ATS [9] and TSSA [10], the best fitness score corresponding to Bali score were evaluated

Genetic algorithms based approach (SOGA) is used with new se- lection and crossover scheme. Such an algorithm is useful in pro- ducing the best population on local schema for achieving a good alignment. The complete structure that solves the MSA prob- lem has been shown in Figure 1.1.

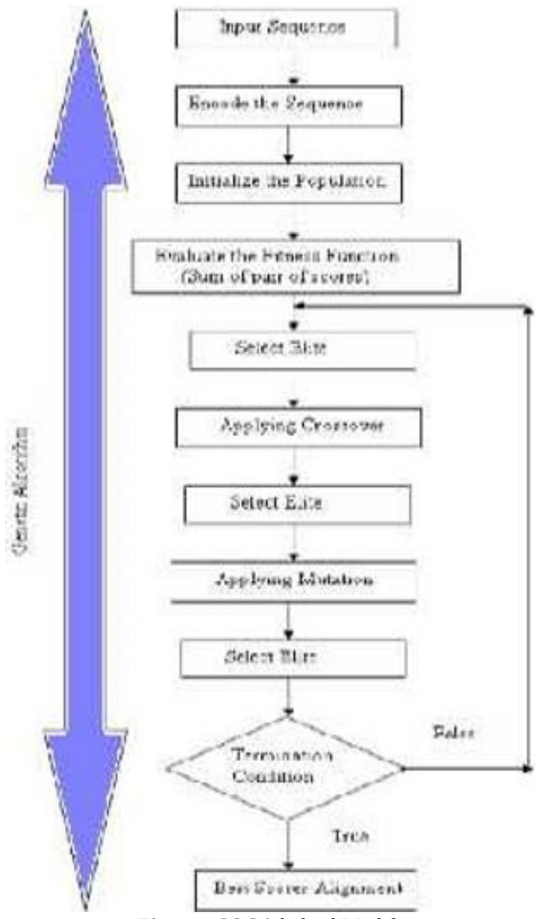


Fig. 1.1: SOGA hybrid Model.

2. Development of SOGA for solving MSA

The gap position represents the chromosomes encoding for MSA. When calculating the number of space that is inserted in each sequence, the lengths of all the aligned sequences are kept equal. The total number of gap positions of all the input sequences is represented by integers while constituting a chromosome. The processes such as the formulation of chromosome representation, population initialization, and fitness evaluation and selection pro- cedure are common for the proposed variants of SOGA. They are explained below:

2.1. Formulation of chromosome representation

The chromosome should have information related to a solution. The encoding is be implemented through binary string. In general, chromosome is represented as a matrix with fixed lengths and sequences with spaces [11] [12]. The gap positions are used to en- code the chromosomes for an MSA problem. A single chromo- some comprises of gap positions of all the sequences in order. The binary string is encoded as a bit matrix that constitutes of 0 and 1 and was encoded by Wayama et al [13]. Here, the position of [1] and 0 relates to a space and a symbol respectively. The present work utilizes the encoding chromosomes such as the bit matrix and records only the space position. The chromosomes are encoded as binary digits (1, 0) in the mutation process where the presence and absence of the gap in sequence is represented. In SOGA, based on the number of sequences and its length, the chromosome length is adaptively changed [14]. Chromosome is defined as the multiple number-strings with fixed lengths represented for sequences and spaces, in an alignment [15].
The longest length of the alignments represented by a chromosome is limited by the N value. The N value which depicts the search space size of alignments has to be selected. The significance of N for finding the best alignments of similar sequences that fail to originate is quite small. If the value of N is big, then it requests for additional time to discover the optimal alignment of greatest likeness of the sequences. Therefore, evaluate the length of sequence by multiplying the highest length of particular ele- ment of sequences with rsp1.2. In a set of sequence S = {S1, S2, S3 ….Sn}, the greatest length of the column can be generated by multiplying the sequence with rsp by greatest column length. The value of scaling factor rsp defines the alignment that needs to be 20% longer than the sequence as it is based on the observations. MSA problem contains more than 20% gaps for which a particular solution is founded. rsp referred as space ratio determines the M value. If the longest length of the sequence to be aligned is mmax, then

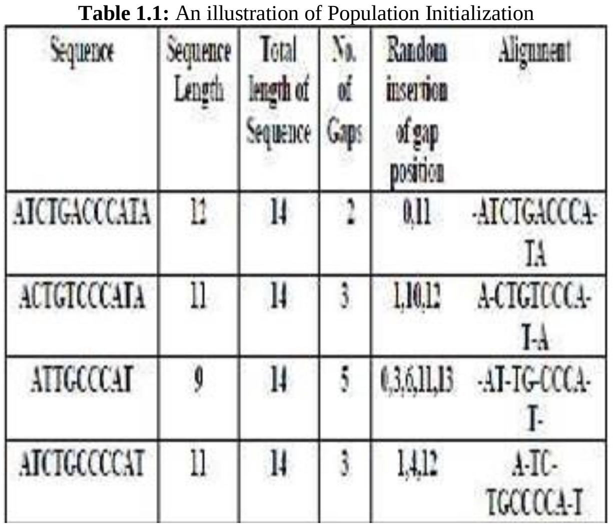
M = mmax × (1 + rsp)
Space column is the column that contains the space symbols. The value of mmax is 12 as shown in figure 1.2. If rsp be 0.2, then L is 12× (1+. 2) =14. For example, a chromosome (0,11), (1,10,12), (0,3,6,11,13), (1,4,12) is encoded in the form of the alignment:
-ATCTGACCCA
-TA A-CTGTCCCA
-T-A- AT-TG-CCCA
-TA- TC-TGCCCCA-T

Fig. 1.2: An Example of an Alignment

2.2. Population initialization

The number of chromosomes in a generation is indicated by the population size that should be optimal for a particular problem. It considers 4 sequences that are aligned with the length (12, 11, 9, 11) and the space ratio rsp is = 0.2 .Chromosomes can be transformed to actual alignments by inserting gaps in the appropriate positions. The first population is generated randomly in the form of steps which are further discussed. In the first step, the length of alignment N is calculated according to the given sequences and space ratio. In the second step, a random permutation is generated from the set {0, 1, 2 N}. Each row shortens its elements to fix the length similar to the space symbol in the corresponding row's original alignment. Finally, the space symbol occupies the positions of all the rows in the matrix. Table 1.1 shows an example of using the random initialization procedure to generate the initial population.

Table 1.1: An illustration of Population Initialization



Sequence	Sequence Length	Total length of Sequence	No. of Gaps	Random insertion of gap position	Alignment
ATCTGACCCATA	12	14	2	0,11	-ATCTGACCCA-TA
ACTGTCCCATA	11	14	3	1,10,12	A-CTGTCCCA-T-A
ATTGCCCAT	9	14	5	0,3,6,11,13	-AT-TG-CCCA-T-
ATCTGCCCCAT	11	14	3	1,4,12	A-TC-TGCCCCA-T

International Journal of Engineering & Technology 151

2.3. Estimation of fitness function

The chromosomes must be converted to the alignment form to evaluate their fitness and therefore, they are applied with the sum- of-pairs function [16]. The sum-of-pairs function is described as the sum of scores of each pair of symbols in a column.
Sum-of-pairs Score:

The sum-of-pairs score evaluates the ratio of correctly aligned residue-pairs with the length of the generated alignment. By ob- serving the matches, mismatches, and the gaps in the two se- quences, each of the columns are scored. The values that are as- sumed are: a match = 1, a mismatch = 0 and a gap = -1. Sum-of- pairs function is defined as the sum of scores of all the pairs of symbols in the column. A pair of symbol might get zero score, match score, mismatch score or space score. To assess the perfor- mances of the algorithm, two different scores are calculated [17]
[18] with which the quality of an alignment is estimated. For this, a test alignment is assumed consisting of N sequences of M columns. The ith column in the alignment can be referred to as Ai1, Ai2, AiN. Aij and Aik, pijk for each pair of residues and is de- fined such that pijk = 1.The score Si for the ith column is represented as:

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

2.4. Selection procedure

After the fitness score of all the population is calculated, SOGA implements a larger tournament method where n individuals are randomly chosen. In this case, the fitter individual is selected by the following process:-

i) Tournament method is applied on the current population based on the fitness value of chromosomes.
ii) Two optimal DNA sequences are selected randomly, based on their fitness score.
iii) Two new offspring's are selected with their maximum Fitness value.

2.5. Crossover

Crossover is the process of exchanging the information for creating two new off springs. Good structures of parent chromosomes might be preserved and the child chromosomes with greater performance might be formed. In the crossover procedure, two parent chromosomes, represented as X and Y, are randomly selected and after exchanging their information, each of them produce two child or off-springs. These off-springs are called optimal random child or off-springs respectively. In the crossover method, random cut point is selected between the two parents. The crossover block is generated after selecting the cut point area between the parents. Then the crossover blocks of random child are randomly selected.

2.6. Elite is selected

After calculating the cutting point, the first parent sequence is divided vertically into two segments. The second parent is also divided into two segments in such a way that every row of the first and second segment has equal number of essentials as that of the first parent. The steps in the division are presented with solid arrows for Parent P and Parent Q as depicted in figure 1.3. Moreover, the segment of both the parents are swapped and combined simultaneously between two individuals. There are two off springs that are produced by crossover operators, only to generate two new offsprings. With the use of this crossover better offspring can be achieved.

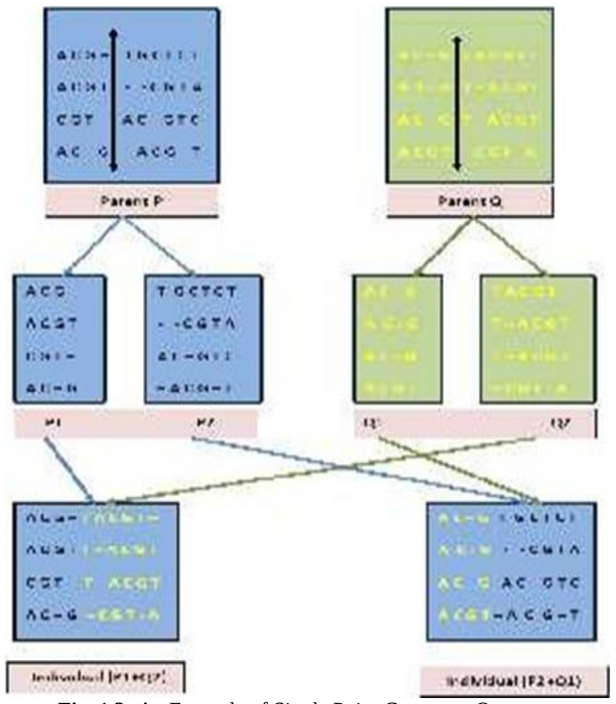


Fig. 1.3: An Example of Single Point Crossover Operator.

2.7. Mutation process

After the crossover takes place, two new offspring are created the mutation process is then started. A DNA chromosome from the mating pool is selected randomly from the system. A bit flip or binary method is applied on individual. In this, the space mutation in an individual bit of a binary string is replaced [13] (for instance, 0 is converted to 1, and vice versa).The new offspring changes randomly due to mutation. Chromosome representation is con- verted to binary digits (1, 0) by proposed space mutation operator. It represents the presence and absence of gaps.

2.8. New generation

For evaluating the new generation, the best 50% of the parents and children were combined together. It is noticed that there are no replication of offspring. Another criterion for splitting the genera- tion like (40 % parent, 60 % child) and (60 % parent, 40% child) is performed. This mix (50-50 percent parent-child) was chosen on the basis of experimental observations to ensure the balance be- tween exploration and exploitation. Figure 4.6 illustrates the process of forming a new generation.

3. Analysis of GAMS for MSA

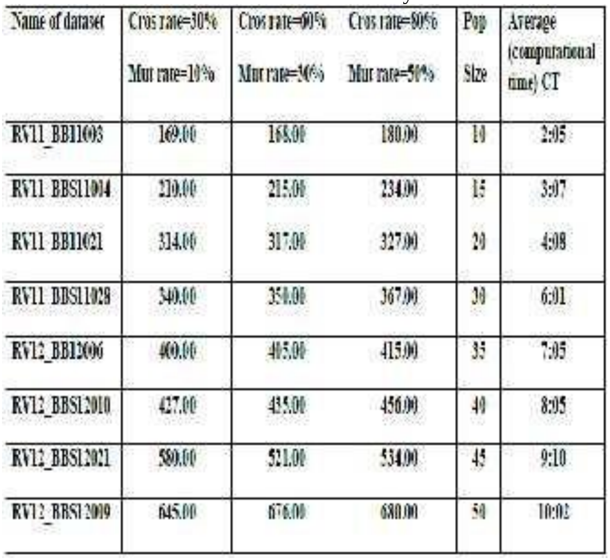
The performance of GAMS algorithm is analyzed based on the results of 100 independent runs. This is further compared with other well-known methods. In the proposed SOGA algorithm, two elementary search operators are used to align the sequence. To find the optimal solution for aligning the sequences; different experiments were carried out using 08 randomly selected BaliBase datasets [19] (version 2.0). With respect to each of the ten datasets, the algorithm is executed 100 times on GA in the present experi- ment.
There are two basic operators that are used in the proposed GAMS algorithm, namely, crossover and mutation. The crossover and mutation values are determined with the help of the following three different experiments that were carried out using 10 random- ly selected BaliBase datasets (version 2.0):

i) 30% crossover and 10% mutation
ii) 60% crossover and 30% mutation
iii) 80% crossover and 50% mutation

This algorithm is performed for each of the 10 datasets with different parameter settings. They are performed for 50 independent

runs, 50 runs (considered as the best score) are recorded and BAliscore is reflected in Table 1.2. Simultaneously, figure1.4 presents the related plot. It is observed that our algorithm is carried out
152 International Journal of Engineering & Technology
with best solutions with some specific options such as for 30 % crossover with 10 % mutation, 50% mutation option with 80% crossover and 60% crossover with 30% mutation. It has been observed that 60 % crossover with 30 % mutation provide solutions close to the best scores. 10% mutation with 30% crossover option doesn‟t produces better solutions. Thus, it might be observed that better performance is achieved by GA for these test datasets when the crossover and mutation rates are selected as 80% and 50%. Statistical and experimental analyses states that 80% crossover and 50% mutation options provide better combination of the ge- netic operators. A single point crossover for the 80% crossovers is selected for half of them. Based on the fitness values, the popula- tion is divided into two groups to form the next child population. An individual is chosen from the first half of 80% and the rest from the second half of 20% for the crossover option. The new population is formed by selecting the best chromosomes from the parents and children that were combined previously. The popula- tion size of 50 is chosen which is used in SAGA [5]. Experiments are conducted using population sizes of 30, 100 and 200 with 80% crossover option & 50% mutation. GAMS performance with a population size of 50 is significantly better than a population size of 30 but it does not shows any relevant difference with the popu- lation size of 200.

Table 1.2: Parameter Analysis



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) CT
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
RV12_BB12006	400.00	405.00	415.00	35	7:05
RV12_BBS12010	427.00	435.00	456.00	40	8:05
RV12_BBS12021	580.00	521.00	534.00	45	9:10
RV12_BBS12009	645.00	676.00	680.00	50	10:02

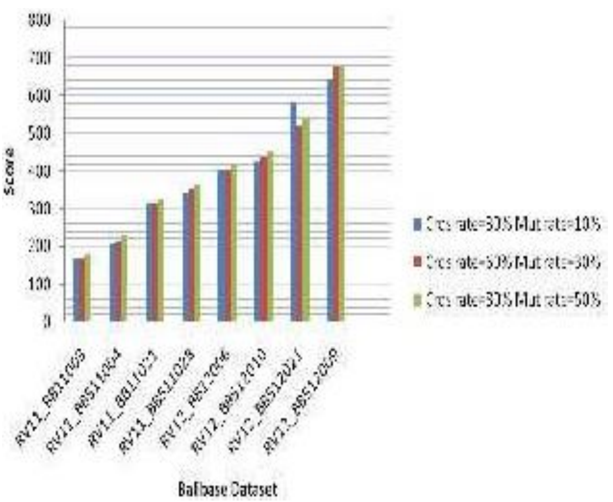


Fig. 1.4: Fitness Scores with Selected Crossover and Mutation Rate Options.

4. Conclusion

Genetic Algorithm is an iterative approach that is beneficial for searching solutions for problems associated with combinatorial optimization, where the traditional methods fail to explain. The limitations of evolutionary algorithms of SA overcome with its adaption to the multiple sequence alignment. It is able to solve the MSA problem with a reasonable level of accuracy. In convention- al GAs, optimal parameter value for a particular problem can be found, by executing the algorithm with all possible values and all possible combinations with other parameter values. This process of optimization requires more time. This time requirement is elim- inated in SOGA. SOGA can also be made to undergo learning process, so that time required to solve similar type of problems in future can be further reduced.

References

[1] Mount, D. W.: Bioinformatics: Sequence and Genome Analysis.*ColdSpring Harbor Laboratory Press*, vol.23. , 2001, pp.134-156.

[2] Needleman, S. B. and Wunsch, C. D.: A general method applicable to the search for similarity in the amino acid sequences of two pro- teins. *Journal of Molecular Biology*, vol. 48, 1970, pp. 443-453.

[3] Isokawa, M., Wayama, M. and Shimizu, T.: Multiple sequence alignment using a genetic algorithm. Genome *Informatics*, vol. 7, 1997, pp. 176–177.

[4] Wayama, W., Takahashi, K. and Shimizu, T.: An approach to amino acid sequence alignment using a genetic algorithm. *Genome Infor- matics*, vol. 6, 1995, pp. 122–123.

[5] Notredame, C. and Higgins, D. G.: SAGA: Sequence alignment by genetic algorithm. *Nucleic Acids Res.*, vol. 24, 1996, pp.1515–1524.

[6] Agarwal, P. and Gupta, R.: A Genetic Algorithm for Alignment of Multiple DNA *Sequences*. *CNC*, 2012, pp. 437–443.

[7] Gondro, C. and Kinghorn, B. P.: A simple genetic algorithm for multiple sequence alignment. *Genetic and Molecular Res.* vol. 6, 2007, pp. 964- 982.

[8] Thompson, J. D., Higgins, D.G. and Gibson, T. J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* vol. 22, 1994, pp. 4673- 4680.

[9] Lamiche, C and Moussaoui A.: An Adaptive Tabu Search Algorithm for Obtaining Alignment of Multiple Sequences. *Baltic J. Modern Computing*, Vol. 2, No-2, 2014, pp.75-83.

[10] Lamiche, C and Moussaoui A.: A Hybrid Method Applied to Mul- tiple Sequence Alignment Problem. *WSEAS Transactions on com- puters*, Vo.14, 2014, pp.2224-2872.

[11] Horng, J.T., Lin, E.M., Yang, B.H., Kao, E.Y.: A Genetic Algorithm for Multiple Sequence Alignment. *In Proceeding of GCB.,*2001.

[12] Liu, D., Xiong, X., Hou, Z., Gupta B.D.: Identification of Motifs with Insertions and Deletions in Protein Sequences using Self- organizing Neural Networks. *Neural Networks*, vol. 18, 2005, pp. 835-842.

[13] Wayama M., Takahashi K., and Shimizu T.: An Approach to Amino Acid Sequence Alignment Using A Genetic Algorithm. *Genome In- formatics*, vol. 6, 1995, pp 122–123.

[14] Wu S, Lee M, Gatton, TM. : Multiple Sequence Alignment using GA and NN. International Journal of Signal Processing. *Image Processing and Pattern Recognition*, 2002, pp. 21-30.

[15] Hong, T., Wang, H., Lin, W., Lee, W.: Evolution of Appropriate Crossover and Mutation Operators in a Genetic Process. *Applied In- telligence,* 16, vol. 16, 2002, pp.1-7.

[16] Sivanandam, S.N., Deepa, S.N.: Introduction to Genetic Algorithms.Springer, 2008.

[17] Sankar, K. and Paul, P.: Genetic algorithms for pattern recognition. CRC Press, Boca Raton, 1996.