# Alignment of Multiple DNA Sequences by Using Improved GA Operators

Manish Kumar*[1]

[1] Department of Computer Science and Engineering, Indian School of Mines,
Dhanbad, Jharkhand, INDIA.
E-mail:- manishism8@gmail.com,
M- +91 9608783395

**ABSTRACT**

One of the most fundamental operations in biological sequence analysis is multiple sequence alignment (MSA). It is a critical tool for biologists to identify the relationships between species and also possibly predict the structure and functionality of biological sequences. The general multiple sequence alignment problem is known to be NP-hard, and hence the problem of finding the best possible multiple sequence alignment is intractable. Therefore, a genetic algorithm based approach has been designed to solve the multiple DNA sequence alignment problem by using different genetic operators. Experimental results with different lengths of DNA sequences has been detailed in this paper . It has also shown that how the increase in length will affect the overall quality of the alignment. The extensive experiment on wide range of datasets and the obtained results has shown the effectiveness of the proposed approach in solving multiple DNA sequences.

**KEYWORDS:** Multiple Sequence Alignment, Genetic Algorithms (GAs), DNA Sequences.

## INTRODUCTION

The main components of the biochemical processes of life are proteins and nucleic acids. There are two types of nucleic acids, deoxyribonucleic acid (DNA) and ribonucleic acid (RNA). DNA sequences are long biomolecular strands composed of four types of nucleotide bases: adenine (A), guanine (G), cytosine (C), and thymine (T). DNA actually occurs as a double strand of such bases. The stands are held together by hydrogen bonds between complementary bases: A-T and G-C. DNA sequences, which consist of hundreds of millions of nucleotides, define the genome of a particular species. Recent advances in bioinformatics have generated volumes of genome data for biomedical research. For example, many immunity genes in the fruit fly genome have nucleotide sequences that are reminiscent of TCGGGGATTTCC[1]. Multiple Sequence Alignments (MSAs) have become highly scrutinized and a fundamental approach in several research domains in molecular biology and bioinformatics such as studies of epidemiology and virulence,[2] drug design,[3] reconstruction of phylogenetic tree, prediction of 3D structure, identifying conserved regions[4-6] and finding molecular function[7-9]. Dozens of algorithms have been developed as a part of an attempt to improve the accuracy of alignments, but still there is not a single MSA method that may generate accurate alignments for all types of test cases[10].

Multiple sequences alignment involves more than two biological sequences, generally protein, DNA, or RNA. Multiple sequence alignment is computationally difficult and is classified as a NP-Hard problem [11]–[13]. Clustal W is a widely used multiple sequence alignment algorithms for DNA or proteins and implements a progressive method for multiple sequence alignment. It calculates the best match for the selected sequences, and lines them up so that the identities, similarities and differences can be seen [14].

Manually refined repositories of MSAs such as BAliBASE,[15] PREFAB,[16] and SABmark[17] are good sources of accurate alignments to gauge performance of various MSA programs, but they have a number of disadvantages such as due to the small size they do not cover the full range of scenarios of protein evolution and due to the uncertain positional homology assessing accuracy of the alignments becomes difficult [18].

It is found that GA is quite suitable for strategy for aligning multiple DNA sequences[19]. The genetic algorithms process starts with an initial population composed of random chromosomes, which form the first generation. Crossover is used to combine genes from the existing chromosomes and create new ones. Then, the best chromosomes are selected to form the next generation. This selection is based on a fitness function which assigns a fitness value to every chromosome. The ones with the best fitness value “survive” to give offspring for the new generation, and the process is repeated until satisfactory solutions evolve.

There are various challenges discovered in the part of genetic algorithm. The first challenge of a genetic algorithm is to determine how the individuals of the population will be encoded and to generate an initial population with some degree of randomness. Literature review [20,21,22] suggests that each individual in the population should be one multiple alignment of all the given sequences, but the way that they came up with the initial population varies. In [21,22,23] the sequence length is increased by a certain percentage and randomly inserted gaps or buffers of gaps into the sequences were considered.

In this study, an approach for aligning multiple DNA sequences has been presented by visualizing the effect of genetic operators. It has been shown that how the length of DNA sequences effect the accuracy of the alignment in longer runs. As the length of DNA sequences increase the alignment quality improves but the success percentage drops down.

The rest of the paper is organised as follow. Next section describes about the materials and methods used for the experiment. Followed by the proposed algorithms section and the experimental section where the results are discussed, The last section summarizes the findings and come to conclusion.

## MATERIALS AND METHODS

### Initial Generation

In this study, first the size of the largest sequence is determined for initializing the DNA sequences. Based on this size, all other sequences are made equivalent to the largest one by inserting gap at different locations of the sequences. It has been taken care that not more than 20% of the gap should be inserted to any of the sequences, as more gap will attract more gap penalty while finding the fitness function and will not give best or accurate alignments in terms of quality. After the population's initialization, the solutions are combined and mutated, producing new individuals through a defined number of generations.

### Fitness Evaluation

Commonly used measure for evaluating the accuracy of MSA programs is to compute *SPS* and *CS* . By counting aligned residue pairs *SPS* can be calculated. It determines MSA tools ability to align some, if not all, of the sequences in an alignment. Let us consider an alignment of $N$ sequences comprise $M$ columns. The $c^{th}$ column can be assigned as $A_{c1}, A_{c2},\ldots, A_{cN}$. For each pair of residues $A_{cj}$ and $A_{ck}$, it is defined $S_{cjk}$ such that $S_{cjk}$ = 1, if $A_{cj}$ and $A_{ck}$ are in the same column of reference alignment. The score for $c$th column ($S_c$) can be defined as follows.

$$S_c = \sum_{j=1}^{N} \sum_{J \neq K}^{N} {}_{K=1} S_{cjk}$$

For full alignment the sum of pair score can be computed as:

$$\text{SPS} = \sum_{c=1}^{M} S_C \quad / \sum_{i=1}^{C_r} S_{rc}$$

$C_r$ denotes number of columns and $S_{rc}$ represents the score of the $c^{th}$ column in reference alignment.

The ability to align all the columns of a given sequences by a MSA tool is determined by column score . It is calculated by dividing the total number of matched columns between test and reference alignments with the total number of "considered" columns in the test alignment. Here, for the experimental analysis it is considered that, $C_c$ = 1 if a column of a (test) alignment matches with the column of reference alignment otherwise it is zero.

$$\text{CS} = \sum_{C=1}^{M} C_c \, / M$$

### Selection Strategies Description

The selection methods used in this research is here under :

Sorting of individuals is done in the mating pool according to their fitness and then every two best individuals are selected for crossover.

### Child Generation

To generate a child population of 10 individuals in any generation, two genetic operators namely Crossover and Mutation are used, which are described below in details.

### Crossover

An overview of the proposed crossover operator used in the experimental study is described hereunder.

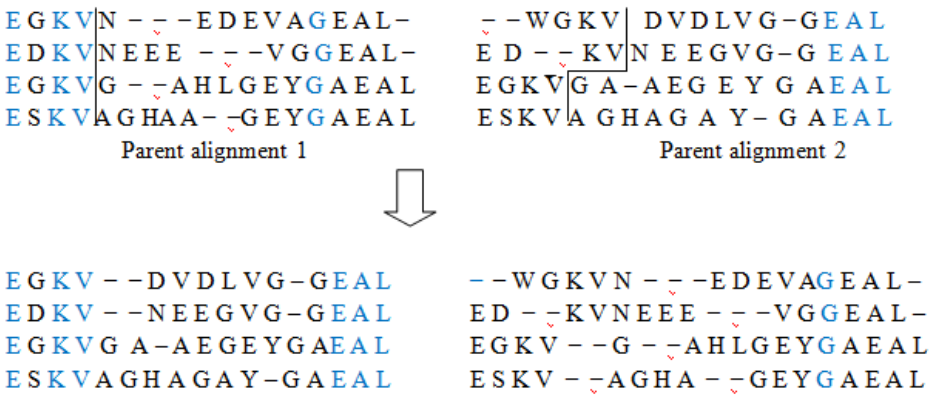


Fig 1. Proposed Crossover operator

### Mutation Operation

A mutation operator is defined and applied for the proposed approach with GA. Mutation operator randomly flips some of the bits in a chromosome. For example, the string 00000100 might be mutated in its second position to yield 01000100. Mutation can occur at each bit position in a string with some probability, usually very small (e.g., 0.001).

Example of mutation operator used in this study.

Here, A is replaced with G and C with T and vice versa. For example,

the parent:

*G* | ***C T A A T*** | *A*

produces an offspring

*G* |***T A G G A*** | *A*

Here ' |' indicates the mutating point.

**Termination condition**

The algorithm is made to be terminated after fixed number of generation or after reaching desire fitness value.

## PROPOSED GENETIC ALGORITHM

The basic steps of the proposed algorithm are summarized as follows.

**Algorithm**

Step I. Calculate the length of each sequence in the alignment and generate the initial alignment of n sequences by inserting the required number of gaps at required positions.

Step II. Evaluate the fitness f(x) of each chromosome that corresponds to the respective sequence alignment.

Step III. Select and save the chromosome with highest fitness value in the current population and perform operations on the remaining chromosomes.

Step IV. Apply crossover and mutation within each populations.

Step V: With a crossover probability (0.8%) crossover the parents to form new offspring (children). If no crossover was performed, offspring is the exact copy of parents.

Step VI. Find the mutation point based on the specified optimal rate for all the selected sequences.

Step VII. Perform mutation (0.2%). Calculate overall alignment fitness value of alignment.

Step VIII. If fitness value of old population is higher, discard changes done in mutation.

Step IX. Save the best chromosome in the alignment and stop after desire number of generations.

## RESULTS AND DISCUSSIONS

**Discussion**

Different genetic operators has been used for implementing four sequences that initially generates ten chromosomes .The genetic operator explained in previous section were then applied on these 10 chromosomes an iteration and it was observed that the overall fitness score for next generation (population) of chromosomes was better than the previous one. Success rate calculated in terms of % is quite significant and the proposed scheme can be used to generate good multiple alignments.

Table 1. Experimental Parameters

| | |
|---|---|
| No. of Sequences | 4 |
| Size of Population | 10 |
| Crossover Rate | 0.8 |
| Mutation Rate | 0.2 |
| Scoring Matrix | PAM250 |
| Gap Penalty | -3 |

**Implementation**

In this section the multiple sequence alignment problems has been solved by Genetic Algorithm using DNA sequences of different lengths. Experiment is performed with the help of selection, crossover and mutation operators with defined number of generations, in order to produce new solution with improved fitness values.

The extensive experiments were performed on the proposed algorithm by using C programming on an Intel Core 2 Duo processor with T9400 chipset, 2.53 GHz CPU and 2 GB RAM running on the platform Microsoft Windows Vista.

Table 2. Comparative study of fitness values in terms of Percentage

| No. of Sequences | Population Size | Fitness Score after. 1st Gen. | Fitness Score after 2nd Gen. | Fitness Score after 3rd Gen. | Success Rate in terms of fitness |
|---|---|---|---|---|---|
| 4 | 10 | - 494 | - 437 | - 401 | 22% |
| 8 | 12 | - 459 | - 440 | - 416 | 11.5% |
| 10 | 14 | - 556 | - 523 | - 489 | 12% |
| 12 | 16 | - 566 | - 550 | - 509 | 11.5% |
| 16 | 20 | - 589 | - 577 | - 553 | 9% |
| 18 | 22 | - 623 | - 607 | - 591 | 8% |

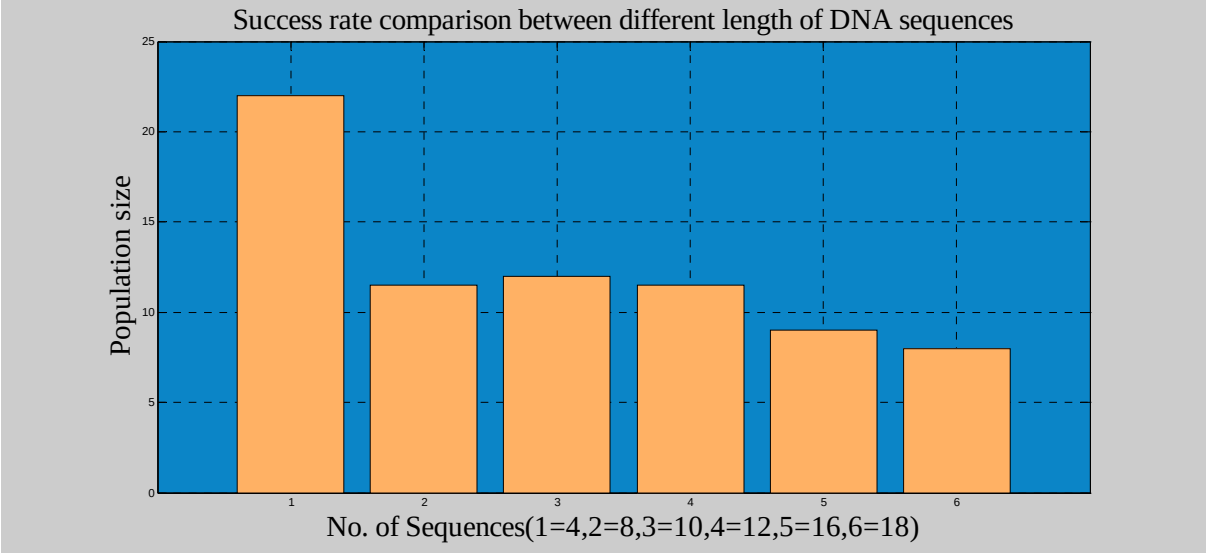


Fig 2. Bar graph comparison of success rate between DNA sequences of different lengths.

The main objective of the research work is to use the proposed operators for solving MSA problem of DNA sequences. Experiment has been performed on six different DNA sequences of different lengths by using the defined operators to solve MSA problem and the result in terms of fitness values has been compared accordingly. Table 2 shows a comparative study of fitness values with changing generation, sequence number and population size.

In the above table, the comparison is made between three generations but, it can be seen that as the generation number increases the fitness values increases accordingly. So, if the generation number is kept at higher side a much more improved fitness value can be expected with a greater rise in success rate in terms of percentage. Here, one more thing which must be taken into consideration from Table 2 is that as the number of sequences increases the success rate decreases. So, when increasing or changing the sequences number or the population size or any other parameter such as the generation number the nature of success rate must be taken into consideration.

## CONCLUSION

In this work, a new approach for genetic operators has been applied for solving MSA problem of DNA sequences. This paper presents an improved approach for the MSA problem by applying standard GA with improved genetic operators. The experimental result shows that the proposed method gives a better scope for multiple sequences alignment, as the results of each alignment tends to improve, which is being shown by the increasing fitness value with increase in number of iterations. But, it can be seen that when the sequences number and the population size increases the success rate of fitness decreases accordingly. Considering this fact as an important issue, the future research work will focus on improving the fitness success rate in accordance with the increase in sequence number and population size so, that the proposed method can be used for sequences of larger length.

## REFERENCES


[1] Rambally, G., A visualization approach to motif discovery in DNA sequences, SoutheastCon, Proceedings IEEE , 2007,pp.348-353.
[2] Bao Y, Bolotov P, Dernovoy D, et al. The influenza virus resource at the National Center for Biotechnology Information. J Virol. 2008,82:596–601.
[3] Kuipers RK, Joosten HJ, van Berkel WJ, et al. 3DM: systematic analysis of heterogeneous superfamily data to discover protein functionalities. Proteins. 2010,78:2101–2113.
[4] Kim J, Ma J. PSAR: measuring multiple sequence alignment reliability by probabilistic sampling. Nucl Acids Res. 2011,39(15):6359–6368.
[5] Siepel A, Bejerano G, Pedersen JS, et al. Evolutionarily conserved elements in vertebrate, insect, worm, and yeast genomes. Genome Res. 2005,15(8):1034–1050.
[6] Roskin KM, Diekhans M, Haussler D. Scoring two-species local alignments to try to statistically separate neutrally evolving from selected DNA segments. In: Proceedings of the seventh annual international conference on computational molecular biology ACM Press. 2003, 257–266.
[7] Levasseur A, Pontarotti P, Poch O, Thompson JD. Strategies for reliable exploitation of evolutionary concepts in high throughput biology. Evol Bioinform Online. 2008,4:121–4137.
[8] Wong KM, Suchard MA, Huelsenbeck JP. Alignment uncertainty and genomic analysis. Science. 2008,319:473–476.
[9] Loytynoja A, Goldman N. Phylogeny-aware gap placement prevents errors in sequence alignment and evolutionary analysis. Science. 2008,320:1632–1635.
[10] Thompson JD, Linard B, Lecompte O, Poch O. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. PLoS One. 2011;6(3):e18093.
[11] L. Wang, T. Jiang, On the complexity of multiple sequence alignment, in Journal Computational Biology, 1994, 1 (4), pp. 337– 348.
[12] W. Just, Computational complexity of multiple sequence alignment with SP-score, in Journal Computational Biology, 2001, 8 (6), pp. 615–623.
[13] S.H. Sze, Y. Lu, Q. Yang, A polynomial time solvable formulation of multiple sequence alignment, in Journal Computational Biology, 2006, 13 (2), pp. 309–319.
[14] Borovska, P.; Gancheva, V.; Landzhev, N., Massively parallel algorithm for multiple biological sequences alignment,(TSP), International Conference on Telecommunications and Signal Processing, 2013,pp.638-642.
[15] Thompson JD, Koehl P, Ripp R, Poch O. BAliBASE 3.0: latest developments of the multiple sequence alignment benchmark. Proteins. 2005,61:127–36.
[16] Edgar RC. MUSCLE: multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004,32(5):1792–1797.
[17] Walle IV, Lasters I, Wyns L. SABmark-a benchmark for sequence alignment that covers the entire known fold space. Bioinformatics. 2005;21(7):1267–8.
[18] Karplus K, Hu B. Evaluation of protein multiple alignments by SAM-T99 using the BAliBASE multiple alignment test set. Bioinformatics. 2001,17(8): 713–720.
[19] Goldberg, D.E.; Genetic algorithms in search, optimization & machine learning. Reading, MA: Addison-Wesley Publishing Company, Inc (1989).
[20] Hernandez, D., Grass, R., & Appel, R. MoDEL: an efficient strategy for ungapped local multiple alignment. Computational Biology and Chemistry, 2004,28, 119-128.
[21] Horng, J.T., Wu, L.C., Lin CM., & Yang, B.H. A genetic algorithm for multiple sequence alignment. Soft Computing, 2005:9, 407-420.
[22] Wang, C, & Lefkowitz, E.J. Genomic multiple sequence alignments: Refinement using a genetic algorithm. BMC Bioinformatics, 2005,6: 200.
[23] Shyu, C, Sheneman, L., & Foster, J.A. Multiple sequence alignment with evolutionary computation. Genetic Programming and Evolvable Machines, 2004:5, 121-14.