Multiple Sequence Alignment based on Developed Genetic Algorithm

H. Nabeel Kaghed¹, S. Eman Al-Shamery² and Fanar Emad Khazaal Al-Khuzaie²*

¹Ministry of Higher Education and Scientific Research, Head of Supervision and Scientific Evaluation Apparatus, Hillah, Iraq; nhkaghed@itnet.uobabylon.edu.iq
²Department of Software, College of Information Technology, University of Babylon, Hillah, Iraq; emanalshamery@itnet.uobabylon.edu.iq, fanar@itnet.uobabylon.edu.iq

Abstract

Background: Multiple Sequences Alignment (MSA) is one of the most important research themes in bioinformatics as well known that the Genetic Algorithm (GA) working on finding the optimal solution, but it may take long generations to get to the solution and this is a problem especially in protein. Methods: In this research the dataset has been used in the form of Deoxyribonucleic acid (DNA) sequences, and protein sequences for the purpose of global alignment all these sequences belong to a human. Aligning several sequences cannot be done in polynomial time and therefore, heuristic methods such as GA can be used to found approximate solutions of MSA problems. Several algorithms based on genetic algorithms have been developed for this problem in recent years. In this paper, a new evaluation process, good mutation probability, new recombination operators, and a strong condition for termination have been used by developed genetic algorithm. Findings: The strength of the proposed GA is excellent matching and a good time with less storage. The experimental results show that the proposed algorithm is capable of finding good Multiple Sequence Alignment in contrast to a traditional genetic algorithm while it uses low computational complexity. The weaknesses of the traditional GA have been solved by the developed GA. Applications: Multiple sequence alignment by using genetic algorithm plays a vital role in our day-to-day life for various biological purpose especially in a clinical laboratory.

Keywords: Affine Gap Penalty, Genetic Algorithm, Global Sequence Alignment, Local Sequence Alignment, Multiple Sequence Alignment

1. Introduction

A genetic algorithm belongs to the family of artificial intelligence and specifically to the evolutionary algorithms, it is considered as public search algorithms¹. Genetic algorithm relies on its work on natural selection techniques and natural genes and these algorithms are conducting a random search and parallel to a total of candidate solutions in order to choose the best among them². Genetic algorithms are characterized by high efficiency because of what they offer enormous possibilities including³. Alignment algorithms (heuristic and dynamic) use two different kinds of sequence alignment, local, and global. Local, explore best portion matching when global, explore the best match of sequences in whole⁴. There are many reasons for aligning, to infer homology, and to study the evolutionary relationships between the sequences. The pairwise sequence alignment used to find the best match between two sequences, whether (local or global). There are basic methods which produce pairwise sequence alignment dot matrix method, dynamic programming method and word method, where each of method's has strengths and weaknesses. Multiple sequence alignment is an extension of pairwise, but it is for align all sequences

*Author for correspondence
in the query set. Multiple sequence comparison indicates to search for symmetry in three or more sequences. Alignment of multiple sequences is a hard computational case. There are two main approaches to performing multiple sequence alignment: 1. Progressive method. 2. Iterative method. The progressive method gradually builds up the alignment by aligning the two most similar sequences first, and then one after another by adding the most distant one lastly. This method is very fast and simple, but it has also a disadvantage – mistakes made at an early intermediate stage cannot be corrected later with the addition of more sequences to alignment. On the other hand, iterative methods are trying to optimize a scoring function related to biological decisions such that optimization of the score leads to a correct alignment.

After the introduction, related studies, the proposed system were used, the remaining of the research includes the results and conclusion.

There are some offered methods of the (MSA) problem: evolutionary method SAGA according to GA has been successfully applied to the MSA case. It is used to improve two different objective functions and confirms that they can search huge solution space effectively. But cause of repeated use of the fitness function this may be increasing the time complexity.

Proposed an algorithm according to dynamic programming and GA. It was used with two different distance arrays, there are limitations of great complexity in processing time and limitations of crossover and mutation operations.

New GAs were used to solve MSA case in which many datasets were examined and the practical results were compared with other methods. But after the process of comparison it was clear that this method could obtain a good execution in data sets with a high similarity and long sequences.

Then the effective GARS approach based on GA with the invert selection was proposed. But this method suffers from early convergence in which solutions reach locally in the optimal stage.

At recent, the Cyclic Genetic Approach CGA evolved with the knowledge of problem Parameters by CGA, all values of parameters have been determined based on the problem and the fitness value which got in each generation. But the score value of the column differs for each implementation perhaps not given a relatively the best alignment.

Newly approach of aligning by using affine gap, which are faster than several other ways, but for storage take more space because of the structure of the chromosome which is containing the two locations where one of them represents the number of gaps in the sequence and the other is a field for the weight which is a real value, the support value has been used in the evaluation process, which did not prosper the fitness value.

2. Proposed System

The developed genetic algorithm has been proposed in this research as shown in Figure 1.

**Figure 1.** Block diagram of the proposed system.

In Figure 1 ** refers to proposed step.

1. Input: In this stage DNA/protein sequences are read and stored them for processing.
2. Initial: In this stage variables of the problem is encoded as a chromosome. Structured the vitality group of chromosomes, and fill them by random values (integer code).
3. Evaluation (Sub-Gap fitness): In the evaluation process where using a function which is not one to every problem, but the most difficult part of the algorithm is to choose the appropriate efficiency function, which specifies efficiency or merit or the quality of the chromosome for the problem to be solved.
4. Picking: It is the process of picking the best individual of the generation and choose the best individual depends on the fitness function.
5. Recombine: It is the process of mating between two individuals on crossover point and that produce new
generations. This process done when the parents have crossover probability.

6. Mutation (OP-mutation): Process on the individual to change a series in which the gene (change number or letter according to the chain type), and are used to maintain the diversity of the vital group to the other. Mutation process is done when the individual has mutation probability.

7. Replacement: After mutation process, the children add to the new generation. When reaching to a required number of individuals, then the transmission was to the replacement stage which it is mean this Process is to compensate old individuals by new individuals.

8. Termination: The conditions are specified by the user when he wants to stop the algorithm if true the execution stops else return to the evaluation stage.

9. Output: After many generations the result is found by optimal aligning of sequences.

3. Algorithm of Proposed System

Name: MSA by developed genetic algorithm.

Input: Population with N size, number of generation G, crossover probability Pc, mutation rate Pm.

Output: Aligned sequences.

Begin

1. (Initial) Generate random population of n chromosomes (suitable solutions for the problem).
   a. Select the longest sequence.
   b. Complete the others sequences by gaps until equal with the longest sequence.
   c. Plus 20% of the longest length to all sequences.

2. (Evaluation) Evaluate the fitness f(x) of each chromosome x in the population by

   \[ f(x) = \left( (M_s \ast w_1) + (M_{gap} \ast w_2) \right) \]

   Where:

   \[ M_s = \frac{L_s \ast z}{L_s \ast m} \]

   a. \[ k = \text{no. of rows} \]
   b. \[ M_{gap} = 10^n \] (Affine gap penalty)
   c. \[ W_1: \text{Is first weight.} \]
   d. \[ W_2: \text{Is second weight.} \]

3. (New population) Repeat following steps for new population creating until the completion of a new population.
   a. [Picking] select two individuals (parents) from a population according to their fitness (chance for selection it is depending on better fitness).
   b. [Recombine] with a crossover probability cross over the parents from the crossover pool, and select a particular section in the first chromosome and exchange corresponding intended section with the second chromosome.
   c. [Mutation] mutation operations are applied to each pair after recombining, with a mutation probability, the mutation technique is applied by changing one gene with another randomly.
   d. [Accepting] put new offspring in the new population.

4. (Replace) for a further run of the algorithm, newly generated population have been taken.

5. (Termination) If (the required number of generations has been reached OR state of convergence has been accessed ), stop and return the best solution in the current population.

6. (Loop) Go to step 2.

End

4. Algorithm Explanation

At beginning to choose the longest sequence that have been entered and then adding gaps for each sequence so that they are equal with full-length (the longest sequence +20% of the longest sequence length). The fitness score is calculated after when each chromosome is converted into the corresponding sequences with gaps, for sequences alignment. Fitness function (Sub-Gap fitness) of an evaluation process for each chromosome
in the population through a certain equation rely on the number of transactions (gaps and the frequent values of sub sequence in all sequences), according to the following equation:

\[ f(x) = (M_s \cdot w_1) + (M_{gap} \cdot w_2) \]

Where:

\[ M_s = \frac{L_s \cdot z}{L_s \cdot m} \]

Where, \( L_s \) is the length of sub sequence and it has been fixed during the implementation, searching for similarities among all sequences, see Figure 2.

![Figure 2. Example of sub sequence.](image)

**Length of sub sequence = 3**

**No. of sub sequences = 4**

**Figure 2.** Example of sub sequence.

\[ M_{gap} = 10^n \text{ – (Affine gap penalty)} \]

Where, \( n \) represents the orders value of the gaps in each chromosome, so if the number of one chromosome gaps are 1220 then \( n = 4 \).

Affine gap penalty = \( \sum_{i=1}^{k} Gap_i \)

where, \( Gap = G_{\text{conquest}} + G_{\text{protraction}} \)

where, \( G_{\text{conquest}} \): The penalty of a gap opening,

\( G_{\text{protraction}} \): The penalty of a gap extension.

See Figure 3.

The next example illustrate this cases in more detail, if the:

\[ G_{\text{conquest}} = 4. \]

\[ G_{\text{protraction}} = 3. \]

So the result of affine gap penalty for (a) is:-

\[ 4 + 4 + 4 + 4 = 16 \]

When the result for (b) is:-

\[ 4 + 3 + 3 + 3 = 13 \]

**Figure 3.** Two cases of alignment.

Gap penalty is necessary to avoid inserting lots of gaps in the sequences and it makes the objective function a better approximate of biological alignment fitness. The (Sub-Gap fitness) term refers to (sub-sequence and affine gap penalty) fitness.

After the value of each chromosome has been found then move on to the process of creating a new population through several steps. The most important is the choice of parents according to the supreme value, then mating has been done on crossover probability, where the efficient probability of mating (optimal probability) has been determined in this research, a crossover technique has been illustrated in Figure 4.

The random does not always produce a better solution, this research is reduced the depending on random, resolve this issue of random, and find a better solution, by setting limitations, thus for make sure to give the best solution in terms of time, this shows clearly through the way in which a research has been proposed. In this work mating technique adopted which is a new approach, by specified first part (sequence) in one chromosome and exchange it with the corresponding part (sequence) in another chromosome as shown in Figure 4.

A mutation (OP-mutation) has been done, and then new individuals have been put in the population. The (OP- mutation) term refers to (optimal probability) of mutation. After the completion, the new population has been instead of the old one. Termination condition has been sure if true then the algorithm finish, else move on to Step 2, as well known, that is a sufficient number of generations as a condition to stop the algorithm, consider as an inept condition, so it has been added other condition for the purpose of the termination, which is the absolute value of the difference between the sum of the fitnesses of the current generation with the sum of the fitnesses of the previous generation, and the result is less than the threshold optimal value, which has been reached during the experiments, thus ensuring reduce implementation time and dramatically, there is no need to complete the implementation to the last number generation, as long as required to get the best match.
5. Experimental Results and Discussion

The computer with its specifications (Pentium 4 processor and main memory (3 GB), with Microsoft Windows 7 which has been used as a platform) have been used for implementation. MATLAB language has been used; the information has been extracted from the FASTA format files. The dataset containing DNA sequences of different lengths were as following: 439, 401, 511, 411, 361 which are belong to the human and from the GenBank, as well as we used sequences of protein with several lengths were as following: 239, 381, 264, 239, 163. Many lengths of sequences have been taking through experiments held in this research to see how efficient implementation of the algorithm. While experimenting the number of generations has been fixed, the size of population variable, in order to find out the validity function and taking into account the time during the execution this is clearly shown by the Table 1. The relationship between fitness value and population size of DNA is clearly shown by the Figure 5, and the relationship between fitness value and population size of protein is clearly shown by the Figure 6.

![Figure 4. Crossover technique.](image)

![Figure 5. The relationship between fitness value and population size of DNA.](image)

![Figure 6. The relationship between fitness value and population size of protein.](image)

| Population size | Number of generations | Fitness value | Running time |
|-----------------|-----------------------|---------------|--------------|
| 10              | 100                   | 5334          | 10.535 sec.  |
| 15              | 100                   | 5133          | 15.555 sec.  |
| 20              | 100                   | 5070          | 18.490 sec.  |
| 25              | 100                   | 5055          | 25.409 sec.  |

The most important point in the power of this algorithm is superior speed during the implementation and out the best result, this is evident through the Tables 1 and 2 previously.

The process of selection has been applied randomly taking into account the best choice fitness value of parents, the selection process always guarantees that the
best individual has a higher probability of reproducing and breeding for composing a new generation. The mating has been done according to the probability of crossover (0.8). Mutation process has relied upon selecting randomly of two positions per chromosome and replace them with each other. The mutation has been done according to the probability of mutation which is (0.5). After mutation process, the children add to the new generation. All the past stages have been repeated until stopping by a termination condition (reach the required number of generations or access to state of convergence).

A change has been made in the probability of mating and mutation probability values and ultimately was the best values are:

- The probability of mating = 0.8
- The possibility of mutation = 0.5

As shown in Tables 3 and 4. The relationship between (fitness values) and (probabilities of Crossover and mutation) of DNA is clearly shown by the Figure 7, and the relationship between (fitness values) and (probabilities of Crossover and mutation) of protein is clearly shown by the Figure 8.

### Table 3. The relationship between (fitness values) and (probabilities of Crossover and mutation) of DNA

| Crossover | Mutation | Population size |
|-----------|----------|-----------------|
| 0.3       | 0.1      | 4893            |
| 0.5       | 0.3      | 4808            |
| 0.8       | 0.5      | 4800            |
|           |          | 4760            |

### Table 4. The relationship between (fitness values) and (probabilities of Crossover and mutation) of protein

| Crossover | Mutation | Population size |
|-----------|----------|-----------------|
| 0.3       | 0.1      | 3005            |
| 0.5       | 0.3      | 2856            |
| 0.8       | 0.5      | 2790            |
|           |          | 2701            |

Figure 7. The relationship between (fitness values) and (probabilities of Crossover and mutation) of DNA.

Figure 8. The relationship between (fitness values) and (probabilities of Crossover and mutation) of protein.

A comparative case has been done in this research, between the developed GA, and the traditional GA, for five sequences of DNA, the longest sequence (254), and the shortest sequence (242). Also, five sequences of protein have been taken for a compression where the shortest (242), and the longest (254), as shown in Figure 9.
In the results, the important point is that, for solving complicated problems like MSA, it is not needful to use complicated operators. But it is necessary to define operators precisely which have the ability to improve the population of genetic algorithm.

6. Conclusion

In this paper a simple genetic algorithm has been developed based on many parameters of the algorithm including evaluation, recombine, mutation, termination; as a result the developed algorithm has been improved. The most powerful point of this developed algorithm is the execution time that causes the stop condition so active compared to other developed algorithm techniques. In this research, DNA, and protein sequences have been used, new fitness function has been used in this research.

The optimal probability of crossover and a good probability of mutation have been selected in the proposed system. A comparative case between the proposed system and the traditional genetic algorithm has been done in this research; the results showed the efficiency of the proposed system.

7. References

1. Lin WY, Lee WY, Hong TP. Adapting crossover and mutation rates in genetic algorithms. Journal of Information Science Engineering. 2003; 19(5):889–903.
2. Brindha G, Rohini G, Gnanakousalya C. Genetic algorithm based optimization of single node in reformed-digital micro fluidic biochip. Indian Journal of Science and Technology. 2015; 8(29):1–8.
3. He X, Zhou G. Research of multiple sequence alignment based on genetic algorithm. Journal of Inner Mongolia Agricultural University. 2010; 3:58.
4. Sonali V. Biological sequence alignment for bioinformatics applications using MATLAB. International Journal of Computer Science and Emerging Technologies. 2011; 2(5):310–5.
5. Mojbeak J. Exact multiple sequence alignment using forward dynamic programming. A thesis in Bioinformatics. Bioinformatics Research Center. Aarhus University; 2010.
6. Fatumo S, Akinyemi IO, Adebisi EI. Aligning multiple sequences with genetic algorithm. International Journal of Computer Theory and Engineering. 2009; 1(2):179–82.
7. Deonier RC, Tavaré S, Waterman M. Computational genome analysis: an introduction. Springer Science and Business Media. 2005.
8. Notredame C. Recent progress in multiple sequence alignment: A survey. Pharmacogenomics. 2002; 3(1):131–44.
9. Notredame C, Higgins DG. SAGA: Sequence Alignment by Genetic Algorithm. Nucleic Acids Research. 1996; 24(8):1515–24.
10. Zhang C, Wong AKC. A genetic algorithm for multiple molecular sequence alignment. Computer applications in the biosciences: CABIOS. 1997; 13(6):565–81.
11. Horng JT, Wu LC, Lin CM, Yang BH. A genetic algorithm for multiple sequence alignment. Soft Computing. 2005; 9(6):407–20.
12. Chen Y, Hu J, Hirasesw K, Yu S. Multiple sequence alignment based on genetic algorithms with reserve selection. IEEE International Conference on Networking, Sensing and Control; sanya: 2008. p. 1511–6.
13. Nizam A, Ravi J, Subburaya K. Cyclic genetic algorithm for multiple sequence alignment. International Journal of Research and Reviews in Electrical and Computer Engineering. 2011. p. 1–6.
14. Kaya M, Sarhan A, Alhaji R. Multiple sequence alignment with affine gap by using multi-objective genetic algorithm. Computer methods and programs in biomedicine. 2014; 114(1): 38–49.