Available online at www.elixirpublishers.com (Elixir International Journal)

Information Technology

Elixir Inform. Tech. 97 (2016) 42380-42383

ISSN: 2229-712X

Using Genetic Algorithm to Align Multiple Sequences

Seyed Mostafa Javadi Moaghaddam and Zohreh vahedi

Islamic Azad University of Ferdows, Department of Computer Engineering, Ferdows, South Khorasan, Iran.

ARTICLE INFO

Article history:
Received: 10 September 2015;
Received in revised form:
21 August 2016;
Accepted: 31 August 2016;

Keywords
Bioinformatics,
Sequence Alignment,
Evolutionary Algorithms,
Genetic Algorithm.

ABSTRACT

The problem of Multiple Sequence Alignment (MSA) is one of the most significant problems in bio-informatics world. Solving this problem helps to reconstruct the phylogenetic tree, predict protein structure and its function. Several algorithms are proposed to solve this problem. Genetic algorithm is one of them which has been proposed in different versions to solve the MSA problem. In this article we provide a specific type of genetic algorithm to solve the MSA problem and we will explain it in detail. Once the problem is described, it will also be explained how to formulate the problem and how to define crossover and mutation operators. Finally, using the BALiBASE 3.0 database the performance of the algorithm is evaluated and the results are reported.

Introduction

Multiple Sequence Alignment (MSA) is one of the most significant problems in bio-informatics which is used to define the degree of similarity between sequences of DNA, RNA, and protein [5]. The term sequence refers to a string of characters (nucleic acids or amino acids), and sequence alignment refers to the alignment of strings in a way that their similarity is maximized [1]. Solving this problem will help to reconstruct the phylogenetic tree, predict the structure and/or function of unknown proteins, and search sequence databases. Several algorithms are proposed to solve this problem [3].

Dynamic Programming (DP) Method was first used to solve the MSA problem and some algorithms such as Needleman-Wunsch and Smith-Waterman were proposed [5]. The sequence alignment algorithms provide the best answer to the problem but their problem is to have high computational load. Later, Progressive Algorithm were proposed in order to overcome the problems of dynamic programming method. Today, using progressive algorithms such as T-Coffee and Clustal W are considered as the main approach to solve the MSA problem [5].

It is noticeable that several versions of GA has been proposed to solve the MSA problem including SAGA ،MSA-GA ،VDGA [5]. In this article we are going to apply some changes to the crossover and mutation operators, and offer a new approach to solve the MSA problem [1].

Sequence alignment is defined in section 2. A summary of the genetic algorithm is given in section 3 and in section 4 problem solving method of sequence alignment using genetic algorithm is described. Finally, in section 5 the results are presented [3].

This algorithms are very fast and their answers are the same in different executions, but their common disadvantage is that they may get trapped in local optimal. Repetitive algorithms are a second group of algorithms proposed to solve the MSA problem [4]. These algorithms first perform a preliminary alignment of sequences, and then try to improve the initial alignment within successive iterations until the desired result is achieved. Proposed repetitive algorithms includes Taboo Search, Simulated Annealing, Particle Swarm Optimization, and Ant Colony Optimization. This algorithms are suitable for alignment of complex sequences and produce different answers in different executions [2].

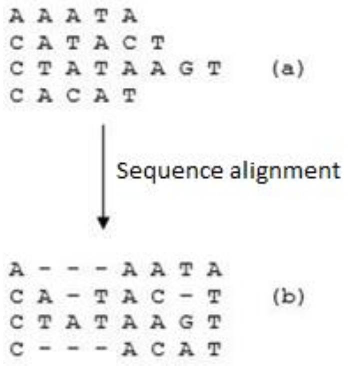
Genetic algorithm is one of them which has been proposed in different versions to solve the MSA problem. The advantage of GA is that the fitness function – which defines the degree of similarity between sequences – can be described totally independent of the body of the main algorithm, or several fitness functions can be defined and their results can be compared. Another advantage of GA is that it can be implanted in way that it is executed in parallel on multiple processors, which in turn results in a reduction in computation time [5].

Problem Definition

The problem of sequence alignment

Suppose the set S includes N sequences as S= (S1… S_N), N>2, each sequence consists of a string of characters (nucleic acids or amino acids), and sequences have different lengths. The sequences of set S can only be DNA or protein sequences. It should be mentioned that DNA sequences can be from a set of four characters nucleic acids (A, T, G and C), and protein sequences can be built from a set of twenty character of amino acids [1].

Now, if we generate the set S' = (S_1'… S_N') by putting gap (empty space) between sequences of set S in a way that the length of all sequences of S' are the same, it is said that the sequences of set S are aligned. Figure 1 shows an alignment of sequences.



```
A A A T A
C A T A C T
C T A T A A G T    (a)
C A C A T

A - - - A A T A
C A - T A C - T    (b)
C T A T A A G T
C - - - A C A T
```

Figure 1. Initial sequences (a) Initial sequences after alignment (b)

Tele:
E-mail address: mahdidarbandi@hotmail.com

As seen in (1-a) with the gaps in different places different alignment can be achieved. A criterion should be used for evaluation in results can be compared. Another advantage of GA is that it can be implanted in way that it is executed in parallel on multiple processors, which in turn results in a reduction in computation time [5].

SP Evaluation Function

Suppose that the length of each sequence is L after alignment, and the j th character of the i th sequence is presented by C_{ij} ($1 \leq j \leq L$). Then the alignment score for characters of j th column is defined as SP – Score (j) which its calculation method is mentioned in formula 1 [5].

$$SP - Score(j) = \sum_{i=1}^{N-1} \sum_{k=i+1}^{N} match(C_{ij}, C_{kj}) \qquad (1)$$

Where *match*(C_{ij}, C_{kj}) represents the degree of similarity between characters C_{ij} and C_{kj} which can be defined based PAM, BLOSUM, or any other valid similarity criteria. The total score of an alignment is finally calculated via the formula 2 [4].

$$Sum(S') = \sum_{j=1}^{L} SP - Score(j) \qquad (2)$$

As seen in (1-a) with the gaps in different places different alignment can be achieved. A criterion should be used for evaluation in order to determine which of the alignments maximizes the similarities between the sequences [1]. Using SP function (sum-of-pairs) is the most common methods to evaluate and score the alignments in bio-informatics, however there are other functions such as COFFEE for this purpose. We use SP function to evaluate various alignments [3].

chromosomes. The principle of genetic algorithm function is to Move from one population to a new one using biological operators such as selection, recombination, mutation, etc. [3]. Genetic algorithm flowchart is shown in figure 2 [2].

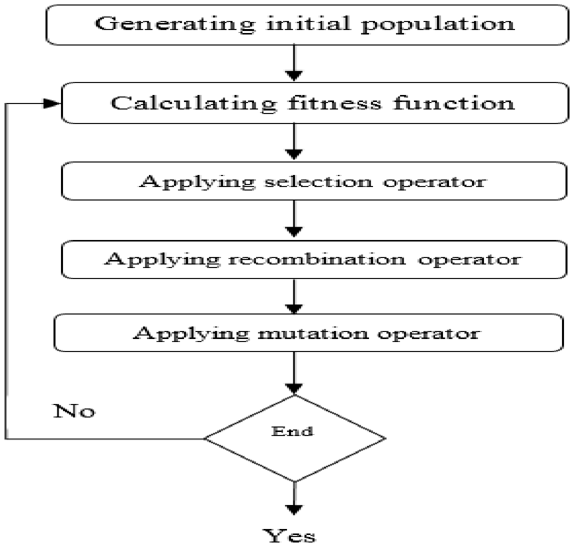

Generating initial population
Calculating fitness function
Applying selection operator
Applying recombination operator
Applying mutation operator
No
End
Yes

Figure 2. Genetic Algorithm Flowchart

Genetic Algorithm

Genetic algorithm is a random search method that uses the natural evolution for solving optimization problems [4]. The algorithm begins the search with a random population in the decision-making space. Each individual is an answer to the question and is so called $L = \alpha * l_{max}$ (3)

Where l_{max} is the length of the greatest sequence (from initial sequences) and we consider the value of $\alpha = 1.2$ in this article Now, if the set S is considered as the equivalent of an individual of the population (chromosomes), different S' sets (different individuals of a population) can be generated by randomly placing gaps between characters of sequences of set S, thus generating the initial population [5].

Solving the sequence alignment problrm using genetic algorithm

Generating initial population

The problem must first be coded in the form of chromosomes in order to solve the MSA problem using genetic algorithm [1]. As explained in section 1-2, set S' = (S_1'… S_N') can be generated by putting gaps (empty space) between characters of sequences of set S in a way that the length of all its sequences equal L. The appropriate value for L can be calculated via the formula 3.

Fitness Function

After creating a generation of people (different alignment), the fitness of each individual should be calculated to compare them. In this problem, formula (2) can be used to calculate fitness function for each individual [4]. By calculating the function for every single individuals of the population, the degree of similarity between sequences can be determined by different alignment. It is worth mentioning that the fitness of each alignment affects its selection for creating the next generation [3].

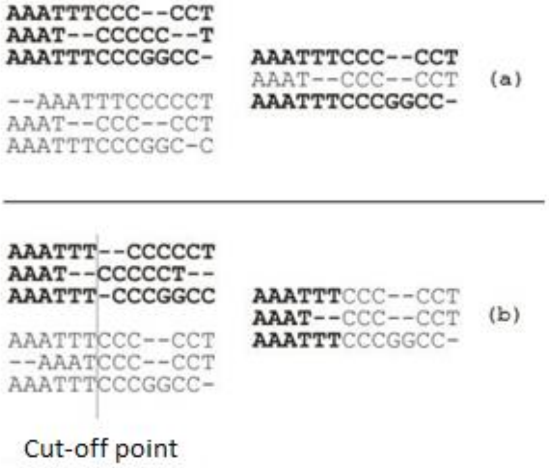
Selection Operator

The selection operator is usually the first operator being applied on the population. Using this operator, some individuals are selected to participate in generating the next generation. There are several ways to select individuals from the population. In this paper w use Roulette Wheel method for this purpose [1]. According to the Roulette Wheel method, the probability for each individual to be selected for participating in the creation of the next generation is based on the fitness of that individual [3]. Successful search strategy used by our group to the protein, ie, genetic algorithm was established [4].

The performance of the new implementation and subroutines which have been for fast computing is tested and the gained results are given here. We can show that the GA approved a strong and efficient search strategy. The second challenge is to identify proper fitness and weight parameters. After the exploitation, in addition to the second strategy presented in this thesis, the approach based on the description is explained. In order to achieve a rapid conformational search for our structure, the prediction system minimizes the computation time for an algorithm (take a look at linear search) [3]. Systematic algorithms cover a long time for a structural space in term of duration. However, this principle is unsuitable for our purpose. We would like to prefer the protein structure to the computer modeling only for a limited amount of computation at a time. In addition to our standard GA, a farming method (see below) is applied on our version [2]. Before that, we are here to test the efficiency of new and rapid implementation of the genetic algorithm (GA) which is used in many difficult optimization problems. Such problems involved in the trial optimization of the gas flow in jet engines (Ingo Rechenberg, 1969, and then Professor Paul Schwefel, University of Dortmund) and flow through different gas pipelines (John Holland). John Holland, a researcher in genetic algorithms (1975) called them as "the optimization of nature". Subsequently, David Goldberg (1989) showed efficiency and good improvement of the GA in his detailed monograph in a number of problems in mathematics, engineering, biology and medicine, including the cost of hospital. In addition to Holland and Goldberg basic theory for GA, they explain the optimization process and different solutions, particularly identifying GA issues by systematic [5].

Recombination Operator

After selecting the alignments in order to contribute to the next generation, they should be remixed two by two using the recombination operator. In this paper, both vertical and horizontal recombination methods are used. Figure 3 shows how they work. It is worth mentioning that the new genration is created once the recombination operator is applied [2].



Mutation Operator

In this problem the mutation operator is defined as a random gap being added to the sequence of an alignment. It is worth mentioning that we have considered the possibility of the mutation operator 1.0 in this article. Multiple sequence alignments can be used in the construction of a phylogenetic tree. [30]. It is possible for two reasons. The first is because the functional areas that are known in annotated sequences can be used for alignment in non-annotated sequences. Another reason is that conserved regions known to be functionally important can be found. So this makes it possible for multiple sequence alignments to be used to analyze and to find evolutionary relationships between the sequences. Insertions and deletions and point mutations can be found. During the project we implemented a [4, 5] recombination. GA is the simulation of an evolutionary process and using it to find solutions that are appropriate to a given problem [3]. About the released protein in the network, modeling problem is to find the next of the three protein, and the solution consists of a linear array vector which is the representative of all conformational elements. The easiest way to understand the GA is to compare it with nature as it was before John Holland. In summary, we have brought this comparison in Table II.2.1..:

Nature	GA
Individuals/ Species	Chromosomes/ individuals = testing solution
Environment, i. e. temperature, humidity, nutrients, etc.	The objective function to judge the fitness of individuals
Mutation = random base variations, delete, add	Mutation = random number of tests in the range of programming
Generation = duration between the publication of sex, i. e. genetic recombination	A generation cycle = A program that includes chromosomal recombination judgment and objective function.
Cross = pairwise currency of chromosomal material	Crossover: Changes in pairs of data between two chromosomes
Reproduction	The number of copies of a chromosome only in the next generation

Because we try to make an easier program for a computer to provide processing of a one-byte secondary structure, we choose to use 9 different characters or numbers. A 0-8 integer number represents each amino acid composition. An integer number in computer random access memory is nothing but a 4 byte = 32 bit string (based on Power PC processor) [1].

GA is now using a subroutine, the objective function, to judge all people and access chromosome. The objective function will be discussed later [3]. We are now in GA and will take a look at core functions. GA will decide how chromosome replicate. It determines the number of copies of each chromosome that will help the new population according to the fitness value. In contrast to nature, the population size can neither be increased nor reduced [4]. Chromosomes with low fitness value will be replaced with the version of the specification. In analogy with nature a low fitness value should set the chance of survival to zero. We would do it using a roulette wheel selection. Chromosomes with high fitness value win over the Chromosomes with low fitness value. In most cases the chromosome with worse characteristic is the alternative of the random number generator, however there can be exceptions to the roulette wheel. The next step is mating. Chromosomes crossed over with random partners. Figure II.2.1. Shows that compared to nature [4].

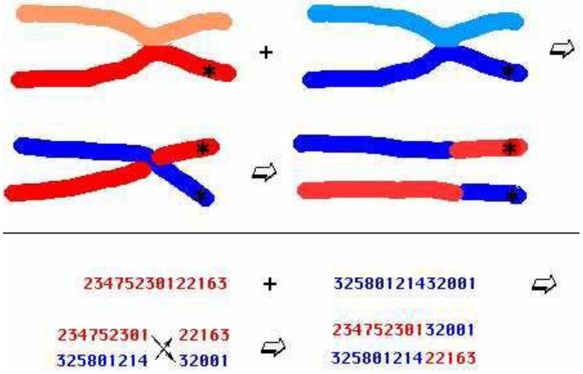


Figure II.2.1. Crossing Over Nature in GA

Now, generation and crossing over alone can lead to results further than random products. It cannot be efficient in the long term depending on the population of all the solutions that have been produced. We introduce natural mutations as similar to those that occur in the genome [2]. On a computer that changes in individual random W has better performance in term of speed.

Suggestions for future work

As mentioned in the previous sections, evolutionary algorithms are one of the ways in which they are used to solve the problem of sequence alignment. Evolutionary algorithms proposed to solve this problem up to now include Taboo Search, Simulated Annealing, Particle Swarm Optimization, and Ant Colony Optimization. Researchers' next plan is to study this problem using ICA algorithm developed by researchers of our country and compare its results with other evolutionary algorithms.

Integers are shown by population, we still use a random number generator to define the occurrence of these changes. The overall mutation rate at each integer is set by the user. An artificial breeding method is applied to GA in addition to the traditional functions [2]. By doing this we try to save calculation time as much as possible. Reproductive realized by a routine is the correction of protected children and results in crossing population with randomly selected individuals. Backcrossing position is randomly determined with the user-defined possibility [1]. To understand the relevance of the GA it is important to know that GA is the common search core of the program. Functions are the most sophisticated part of fitness. Figure II.2.2 depicts the object hierarchy and show that GA is a master function. Size of the box indicates the complexity of the procedures listed.

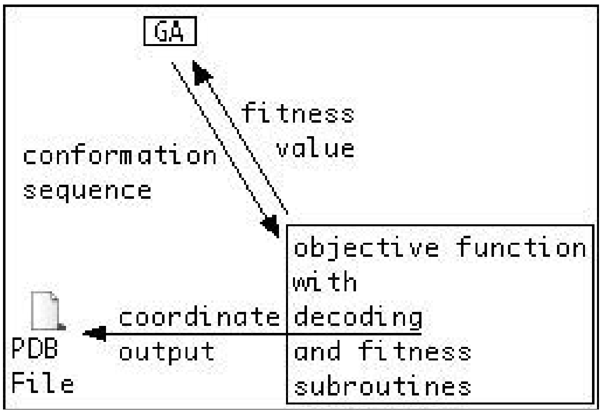


Figure II.2.2. GA Hierarchy and Comparing the Complexity of de Routines

Results and Conclusion

After the implementation of a genetic algorithm to solve the problem of alignment, the performance of the algorithm was evaluated using BALiBASE 3.0 database, and results were compared with Clustal W results. The results showed that in some sequences, alignments derived from genetic algorithm win more points compared to Clustal W, and that the genetic algorithm provides better alignment, while the Clustal.

References

1. Xiujuan, L., Jingjing, S., Xiaojun, X., Ling, G., *Artificial Bee Colony for Solving Multiple Sequence Alignment*, IEEE, 2010.

2. Ling, C., Lingjun, Z., Juan, C., *An Efficient Ant Colony Algorithm for Multiple Sequence Alignment*, Third International Conference on Natural Computation (ICNC), 2007.

3. Radenbaugh, A. J., *Applications of genetic algorithms in bioinformatics*, Master’s Thesis, San Jose State University, 2008.

4. Masamichi, I., Masato, W., Toshio, S., *"Multiple Sequence Alignment Using a Genetic Algoirthm"*,.

5. Gondro, C., Kinghorn, B. P., *A Simple genetic algorithm fo multiple sequence alignment*, Genetics and Molecular Research Jouranl, 2007.