Turk J Elec Eng & Comp Sci, Vol.19, No.1, 2011, © TÜBİTAK
doi:10.3906/elk-1002-410

Sequence alignment from the perspective of stochastic optimization: a survey

İhsan Ömür BUCAK, Volkan USLAN
Department of Computer Engineering, Fatih University,
34500 Büyükçekmece, İstanbul-TURKEY
e-mail: ibucak@fatih.edu.tr, vuslan@fatih.edu.tr

Abstract

DNA and protein are the fundamental biological sequences. DNA is a fundamental molecule that plays a vital role in the processes of life. Proteins synthesized by DNA in a cell are the building blocks of every living organism. There is a variety of reasons behind the alignment of biological sequences. Biological sequence alignment helps to discover functional and structural similarity of sequences. Biologists work with these aligned sequences to construct phylogenetic trees, characterize protein families, and predict protein structure. Sequence alignment is an extremely promising field of research that is characterized by very high computational complexity. Stochastic optimization is needed for sequence alignment, as it generates efficient solutions to the problem. The objective of this study is to survey recent trends in stochastic optimization for sequence alignment as means of a guide for researchers who are interested in the sequence alignment problem.

Key Words: *Sequence alignment, stochastic optimization, simulated annealing, genetic algorithms, particle swarm optimization, ant colony optimization*

1. Introduction

A molecular biology experiment is not an easy task. Real experiments have to be carefully carried out using necessary equipment and the results can only be obtained after weeks and months of work. Computational methods are needed, for they generate solutions faster than conventional experimenting methods in order to solve complex problems of biological systems. The solution for a biological problem may be acquired in hours and days instead of weeks and months by using computational methods. Computer science interacts with the biological sciences as in the field of computational biology for solving biological problems characterized by very high computational complexity. Examples of biologically inspired computing include molecular modeling [1], microarray image processing [2], and structural genomics [3]. Each main topic contains a variety of key subtopics. For example, microarray image segmentation [4] is a subtopic that belongs to microarray image processing. One of the most important problems in the area of computational biology is sequence alignment. The task is to align 2 or more DNA or protein sequences and observe the similarity between them. Biologists work with these aligned sequences on many biological problems, such as constructing phylogenetic trees, characterizing protein families, and predicting the structure of proteins.

Quite a number of studies have been conducted on sequence alignment over the last few decades. The computing time and memory space are the main characteristics that indicate the efficiency of the methods applied to the problem. Retrieving and analyzing sequences in a fast and efficient manner is important. However, while the number or size of sequences increases, an exponential growth in computation time could be observed. As in multiple sequence alignment (MSA), the problem may turn into a nondeterministic polynomial-time complete (NP-complete) computation problem [5]. There are several approaches developed in the literature around sequence alignment; however, for MSA, no optimal solution has been found yet.

Stochastic optimization (SO) is an extremely promising field of research, in which several methods have come up recently. The purpose of this study is to survey the sequence alignment problem, organized as a timeline through the literature, from the perspective of stochastic optimization. The well-known techniques that are prominent methods of stochastic optimization have been discussed in detail. The results and distinctive features obtained have been presented. There is no doubt that the application of stochastic optimization to sequence alignment will continue with an upward trend. Hopefully, research will provide new ideas in order to find an effective solution to the problem in the foreseeable future.

This survey aims for a broad discussion of the state of the art in sequence alignment with a main focus on stochastic optimization. Recent advances in the techniques and methods of stochastic optimization for the sequence alignment problem have been presented. This paper is organized as follows: The first section is an introduction to the research area. The second section lays down the description of the problem. The third section summarizes the familiar stochastic optimization methods being researched, while the fourth section discusses the applications of stochastic optimization methods to the problem. The results are reported and discussed in the fifth section. The last section summarizes the study with a conclusion.

2. Sequence alignment problem

2.1. Problem description

As noted in the description of sequence alignment given by Arslan in [6], sequence alignment allows the observing of similarities between biological sequences; if they are highly similar, then they have similar 3D structures or share similar functions. The problem can formally be represented as a set of sequences, $S = \{s_1, \ldots, s_n\}$, and each sequence has its own length. The characters of sequences are defined over an alphabet Σ including a gap symbol denoted by '–', which is a molecular biology term, indel (insertion or deletion). The indels indicate that some parts of a sequence are inserted or deleted. The sequence is either a DNA, ribonucleic acid (RNA), or amino acid (protein) sequence. The nucleotide bases are adenine (A), cytosine (C), guanine (G), thymine (T), and uracil (U). The alphabet is {A, C, G, T} and {A, C, G, U} for DNA and RNA, respectively. On the other hand, 20 letters for amino acid symbols {A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V} constitute the alphabet for protein. There are 2 types of sequence alignment: pairwise and multiple sequence alignment. Pairwise sequence alignment involves only 2 sequences, whereas multiple sequence alignment involves more than 2. The sequence alignment problem has 2 computational approaches: local alignment and global alignment. In global alignment, sequences are aligned as a whole, whereas in local sequence alignment, similarities detected locally between sequences are aligned. Assume that 2 DNA sequences are given as S1 = {ATTGT} and S2 = {AGGACAT} with lengths |S1| and |S2|, respectively. This pair of sequences can be aligned as shown in Figure 1. The efficiency of an alignment is assessed by the application of some techniques. The most often used techniques are sequence editing and calculating the similarity score. The edit operations are insertion, deletion, and substitution. Insertion is the adding of any symbol s in Σ within the sequence. Deletion is the removing of symbol s in Σ from the sequence. Substitution is replacing symbol s in Σ within the sequence.

```
A T T G - - - T
|     |       |
A _ G G A C A T
```

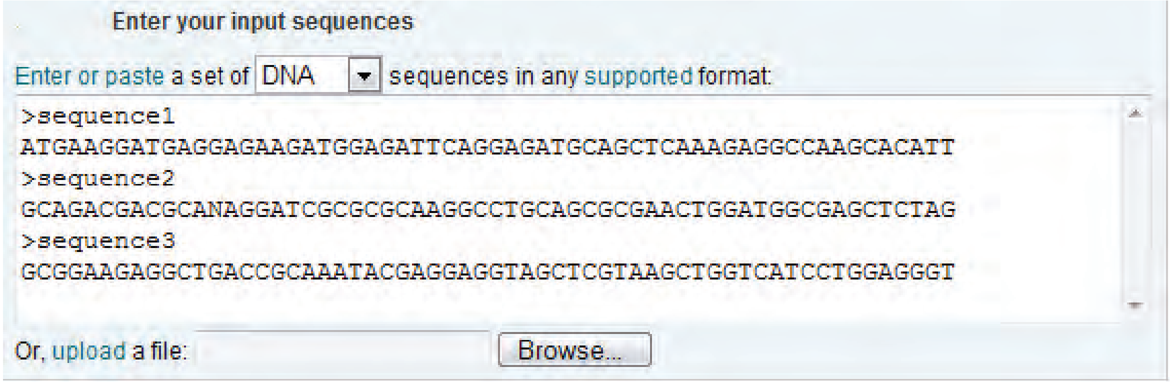
Figure 1. Sequence alignment of 2 DNA sequences.

There exist several scoring schemes for the sequence alignment problem. It is often the case that alignment methods try to minimize the number of gaps by giving a penalty score for their objective functions. The objective function is utilized to maximize the alignment similarity of sequences. One component of a scoring system used in the objective function is penalizing the opening and extending of gaps in the sequence. One familiar approach is using the affine gap penalty model. Affine gap penalty is calculated by

$$f_a(x) = p_o + p_e(x-1), \tag{1}$$

where p_o is the gap opening penalty, p_e is the gap extension penalty, and x is the gap of length. It is observed that a large gap with a small amount is better than a higher amount of small gaps. Therefore, one should consider using a gap extension rather than a new gap opening. There is a novel technique called gap profiling that analyzes the accuracy of indel placement in order to compare the indel positions of 2 MSAs [7]. For protein analysis, a more consistent way to measure the similarity or dissimilarity between sequences is the use of a substitution matrix. The most familiar matrices are based on point accepted mutation (PAM) and the blocks substitution matrix (BLOSUM) [8, 9]. PAM, introduced by Dayhoff in [8], is the first most commonly adapted similarity measure. PAM matrices are derived from closely related proteins. BLOSUM matrices, introduced by Henikoff et al. in [9], on the other hand, are developed by estimating the probabilities from conserved regions of amino acid sequences. BLOSUM matrices are much better in aligning evolutionary divergent sequences. The main difference between these 2 matrices is that PAM matrices are based on global alignments, whereas BLOSUM matrices are based on local alignments. Both similarity matrices and gap penalties contain some specific features. The setting of these parameters in the alignment affects the success of the process [10].

In MSA, one of the most commonly cited examples of the scoring scheme is the sum of pairs method. The sum of pairs method scores all possible combinations of pairs of residues in a column of a multiple sequence alignment. More closely related sequences will have a higher weight in this approach. The MSA program circumvents this by calculating weights to associate to each sequence alignment pair. The weights are then assigned based on the predicted tree of the aligned sequences.



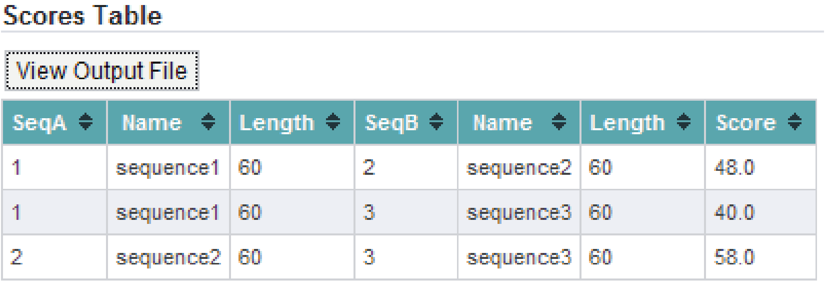
Enter your input sequences

Enter or paste a set of DNA sequences in any supported format:

```
>sequence1
ATGAAGGATGAGGAGAAGATGGAGATTCAGGAGATGCAGCTCAAAGAGGCCAAGCACATT
>sequence2
GCAGACGACGCANAGGATCGCGCGCAAGGCCTGCAGCGCGAACTGGATGGCGAGCTCTAG
>sequence3
GCGGAAGAGGCTGACCGCAAATACGAGGAGGTAGCTCGTAAGCTGGTCATCCTGGAGGGT
```

Figure 2. Entry of the sequences.

Let us give an MSA alignment example from Clustal [11]. Clustal is a common and widely used alignment tool for sequence alignment [12]. The example entries of sequences given in Figure 2 are from the European Bioinformatics Institute (EBI) official website [13]. The process has been carried out usign Clustal 2.0.12 the results obtained with this version have been given in Figures 3 and 4, respectively.



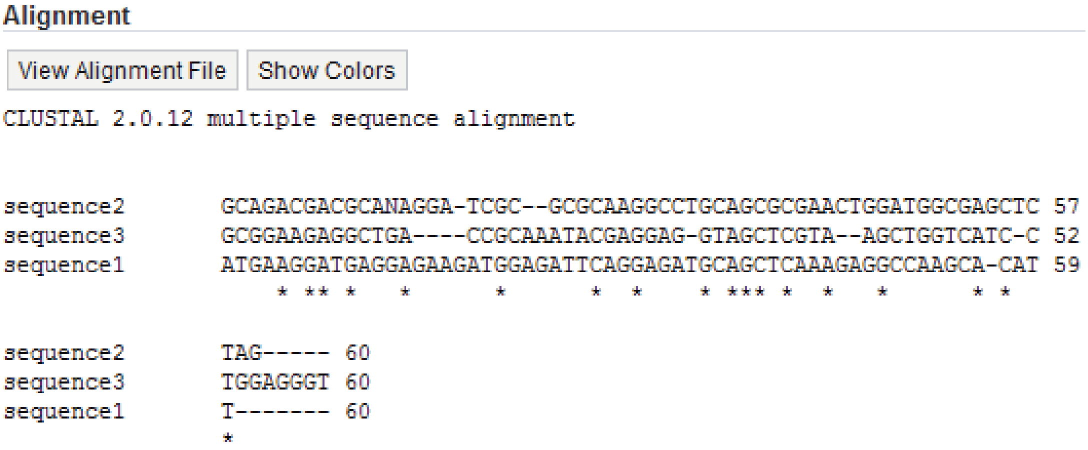
Scores Table

SeqA	Name	Length	SeqB	Name	Length	Score
1	sequence1	60	2	sequence2	60	48.0
1	sequence1	60	3	sequence3	60	40.0
2	sequence2	60	3	sequence3	60	58.0

Figure 3. The score table.

In the example, 3 sequences with an equal length of 60 are the input sequences. When the alignment process has been completed, the outputs are the alignment result and the scores table. The scores table presents the sequence alignment scores, which are 48.0, 40.0, and 58.0, respectively, for this example. The lowest and highest scores represent the lowest and highest similarities between pairs of sequences. Briefly, in Clustal, MSA is performed through the following steps. First, scores of each pair of sequences are calculated and put in a distance matrix. Second, the pairwise alignment scores in a distance matrix are used to construct a hierarchy in the form of a tree. Finally, sequence weights are calculated pairwise based on the tree, and then sequences are progressively aligned using the tree. Aligned sequences, pointed out with a star '*', denote the matching character of these 3 sequences.

Exponential growth may take place with the number of sequences in an MSA problem and, as a consequence, the problem may become intractable. All approaches use some sequence data for performing alignment. There is an international database known as GenBank (Genetic Sequence Data Bank) that contains biological sequences for more than 300,000 named organisms [14]. As new sequences are discovered by scientists, this repository is updated and extended daily. The sequences kept in GenBank provide a useful publicly available repository for all researchers worldwide. Swiss-Prot is another knowledgebase that keeps only protein sequences. Swiss-Prot provides useful information about the protein along with its sequence, such as function description, domain structure, and variations of the protein [15]. Benchmarks assess the success of the methods applied for optimization problems. In sequence alignment, BAliBASE reference alignments have been mostly used in order to assess the performance of the applied methods [55]. This database keeps high quality alignments as reference alignments with their alignment scores.



```
CLUSTAL 2.0.12 multiple sequence alignment


sequence2       GCAGACGACGCANAGGA-TCGC--GCGCAAGGCCTGCAGCGCGAACTGGATGGCGAGCTC 57
sequence3       GCGGAAGAGGCTGA----CCGCAAATACGAGGAG-GTAGCTCGTA--AGCTGGTCATC-C 52
sequence1       ATGAAGGATGAGGAGAAGATGGAGATTCAGGAGATGCAGCTCAAAGAGGCCAAGCA-CAT 59
                    * ** *   *      *      *  *    * *** *  *   *      * *

sequence2       TAG----- 60
sequence3       TGGAGGGT 60
sequence1       T------- 60
                *
```

Figure 4. Alignment result.

2.2. Historical view

In the past decades, several methods were proposed for aligning biological sequences that find the homology among species effectively. The Needleman-Wunsch algorithm has used the dynamic programming approach (DP) for pairwise alignments [16]. The usefulness of DP comes from the fact that it ensures the giving of the maximum score, which means an optimal alignment with any scoring scheme. Another advantage of DP is that it is easy to implement. Smith and Waterman considered the local alignment of sequences instead of aligning them entirely [52]. The Needleman-Wunsch approach was extended by Murata et al. to align 3 protein sequences [17]. Feng et al. utilized the Needleman-Wunsch algorithm for pairwise alignments of sequences and gave scores for each group, and then progressively aligned groups of sequences in order to accomplish multiple alignments [18]. The method rests more on the recently diverged sequences and supports the rule of "once a gap, always a gap." The progressive alignment approach starts by aligning closely related sequences, and then adds the other remaining sequences to the alignment progressively in the order of edit distances. The initial alignment of closely related sequences involves using DP most of the time. This ensures that the initial alignment is more likely an optimal alignment, but the main disadvantage is that once the initial alignment is completed, it will not be possible to modify it again.

3. Stochastic optimization

Stochastic optimization is an extremely promising field of research in which several methods have come up recently, and there is no doubt that this trend will continue and new ideas will be developed in the foreseeable future. Well-known techniques, the prominent methods of stochastic optimization like simulated annealing, evolutionary computation, and swarm intelligence, will be discussed here in detail. There are also many other methods available for stochastic optimization, like basin-hopping, cross-entropy, hill climbing, etc. However, initial versions of these algorithms may need to be improved to fit the sequence alignment problem. This is an open area of research and application of these innovative methods that suit the problem is needed. This paper focuses only on the most familiar ones for the sequence alignment problem that have been researched recently. Swarm intelligence has become the prevailing technique recently and it is utilized as an effective tool for finding solutions for optimization problems. It is inspired by the behavior of swarm insects, like flocks of birds or fish schools.

3.1. Simulated annealing

Simulated annealing, introduced by Kirkpatrick et al., is a stochastic iterative method that attains an optimal solution by performing modifications on existing solutions at each step [19]. It is inspired by the annealing of solids in physics and exposes the similarities between statistical mechanics and combinatorial optimization. The methodology bears analogy with the cooling process of molten metals through annealing. By raising the temperature to a very high level, the atoms of the metal lose their crystalline structure and begin to move freely. The temperature is then gradually declined to a lower level, at which the movements of the atoms are limited. As a consequence, the atoms get arranged and form a crystal structure that yields the minimum energy. The cooling process should be carried out at a slow rate that results in the annealing process; otherwise, the atoms may not achieve the crystal structure and may still be found at a higher energy. Simulated annealing models this slow cooling process of solids to achieve the minimum energy as an analogy, reaching the minimum function value. As a result, it attains an optimal or near-optimal solution by implementing an iterative cooling process from a high temperature, at which solid particles are in the liquid phase. Simulated annealing utilizes a control parameter, temperature t, for the cooling process. The solid is allowed to attain the thermal equilibrium for every t degree that has its energy E probabilistically distributed, as given in equation (2), and kb is the Boltzmann constant. At a high temperature, there is a uniform probability of being at any energy state, whereas a gradual decrease in temperature tends to have only a small probability of being in a high energy state.

$$P(E) = e^{(\frac{-E}{kb^t})} \tag{2}$$

3.2. Evolutionary computation

Evolutionary computation (EC), which is inspired by the genetic mechanisms in nature, is an important field of computational intelligence. EC has received the attention of many researchers over the decades and is applied mainly as an efficient technique for combinatorial optimization problems. A guided search is iteratively applied using biological mechanisms in EC during the optimization process. A genetic algorithm (GA), which is a branch of EC, is perhaps the most familiar and most popular stochastic optimization technique. GA is a heuristic search method inspired by genetic processes in nature [20, 21]. Like in the biology, this method models and uses inheritance, chromosomes, selection, mutation, and crossover concepts. The basic GA algorithm starts initially with a population selected from a set of chromosomes that are potential solutions. Successive individuals within the population are crossed over or mutated so that new individuals are generated. When 2 parent chromosomes are crossed over, the new chromosomes generated are the offspring. Then the best chromosomes from the population are selected and the weak ones are eliminated, and the next generation of the population is formed. A fitness function is utilized to the selection process, which assigns a fitness value to each chromosome. Those that have best fitness value among others in the population survive to give offspring for the new generation. This process goes on until a stopping condition is satisfied. The stopping condition is either an iteration count or an optimal fitness that is achieved by the individuals in the population. The steps of the GA algorithm are as follows:

Step_1: Initialize the population with a set of chromosomes.

Step_2: Calculate the fitness value about each individual in the population according to a fitness function.

Step_3: Select the individuals that have the best fitness values.

Step_4: Generate new offspring for the new generation using genetic operators.

Step_5: Replace weak chromosomes in the population with those newly generated ones.

Step_6: Repeatedly apply the steps from Step_2 until the stopping condition is satisfied.

For small-size problems, GAs can find optimal solutions; however, as the size of the problem grows, it can fall behind the optimal solutions.

3.3. Ant colony optimization

Ant colony optimization is one of the methods used in combinatorial optimization that is inspired from nature, by observing swarm intelligence. It is observed that biological ants can find the shortest path from a food source to the nest [22, 23]. Dorigo and Gambardella described an artificial ant colony in their papers as capable of solving the traveling salesman problem (TSP) [24-26]. The idea is that each artificial ant moves from one city to another with a probabilistic decision on a TSP graph. The probabilistic decision made by an artificial ant is based on the edges that contain a pheromone trail and cities that are close by. Ants modify the pheromone trail of a city when they move to another city. Once the ants complete their tours, the shortest tour travelled by the ant is selected and the edges belonging to this tour are globally modified. ACO is good at escaping from local minimums. This is achieved by the local update rule, which encourages ants to explore.

The equations to model ACO can be described as follows: The decisions made by ant k to select the next city s from city r are shown in equations (3) and (4), respectively. Equation (3) represents a selection of ant k from cities having shorter distances and greater amounts of pheromone, whereas equation (4) allows for ant k to probabilistically select the next city. In other terms, equation (3) is used as exploitation and equation (4) is used as an exploration for ants within the mathematical model. These 2 equations together form the state transition rules for the ACO method.

$$\arg \max_{u \in U_k(r)} = [\tau(r,s)]^{\alpha} [\eta(r,s)]^{\beta}, \text{ if } q \leq q_0, \tag{3}$$

$$p_k(r,s) = \frac{[\tau(r,s)]^{\alpha} [\eta(r,s)]^{\beta}}{\sum_{u \in U_k(r)} [\tau(r,s)]^{\alpha} [\eta(r,s)]^{\beta}}, \text{ if } s \in U_k(r) \text{ and } q > q_0. \tag{4}$$

Here, q is a random value with a uniform distribution over [0,1] and q_0 is a predefined parameter in [0,1]. As q_0 gets smaller, the more probabilistically ants tend to make a random choice. The cities unvisited by ant k located at city r during the travel are denoted by $U_k(r)$. α and β are parameters that indicate the relative importance of the pheromone trail and of the closeness between cities, respectively. τ denotes the amount of pheromone trail and η denotes the ant's heuristic, that is, the inverse of the distance between cities r and s.

The pheromone level on the selected edge is updated according to the local update rule in equation (5). In time, the pheromone amount on the edge weakens. ρ denotes the local pheromone evaporation parameter with a value between 0 and 1. t_0 denotes the initial amount of pheromone deposited on each of the edges.

$$\tau(r,s) = (1-\rho)\tau(r,s) + \rho\tau_0 \tag{5}$$

Once all ants construct a tour that corresponds to a complete iteration, a global update of the pheromone will take place. The shortest tour made by an ant from the beginning is selected. Edges as a function of pairs of $(r,\ s)$ that compose the optimal solution as the shortest path are rewarded with an increase in their pheromone level. This is expressed in equation (6), the global update rule, as follows:

$$\tau(r,s) = (1-\rho)\tau(r,s) + \rho\frac{1}{D_{\min}}, \tag{6}$$

where D_{min} is the global shortest distance found since the beginning. Today, ACO is widely used in optimization problems. Blum has discussed recent trends with ACO and claimed that the research direction offers many possibilities for valuable future research [51].

3.4. Particle swarm optimization

Inspired by social behavior observed in the animal kingdom, some algorithms have been developed in the field of artificial intelligence. Particle swarm optimization, founded by Eberhart et al., is an optimization method that models this social behavior, particularly in terms of fish schooling and bird flocking [27]. It is based on the idea of quickly adapting to the changes in the speed and direction of neighboring particles. PSO contains some similarities to genetic algorithms. The significantly different aspect of PSO is that it does not include the use of genetic operators. The mathematical model of the foraging behavior of the swarm in a PSO algorithm is briefly described below. PSO can formally be defined as a kind of stochastic optimization search technique with the following components:

Step_1: The initial state that includes a population of random solutions.
Step_2: Particles that are potential solutions fly through the search space.
Step_3: An objective function that searches for an optimal solution by updating generations.
Step_4: A stopping criterion that determines when the search is over.

Birds flock together and follow a leader while keeping the distance between them firm. During the movement on search space, each particle moves towards proximity of its best and its neighbors' best, which are called pbest and gbest, respectively. The velocity of a particle is updated as the sum of inertia of its previous velocity, its personal experience, and its cooperation among particles. This equation is given by the following:

$$v_{id}= wv_{id} + c_1 r_1(p_{id} - x_{id}) + c_2 r_2(p_{gd} - x_{id}). \tag{7}$$

Here, w is the inertia weight [28], c_1 and c_2 are the acceleration constants, and r_1 and r_2 are 2 randomly generated values with a uniform distribution over [0,1]. Particles move to their new positions, given by

$$x_{id} = x_{id} + v_{id}. \tag{8}$$

4. Application of stochastic optimization methods to the sequence alignment problem

Numerous algorithms have been proposed for the sequence alignment problem over the past decades. Moreover, over the last few years, there has been a growing upward trend of researching stochastic optimization methods. Notredame pointed out the increasing use of iterative optimization strategies on sequence alignment [29]. This section summarizes recent developments in a number of stochastic optimization methods, such as simulated annealing, evolutionary algorithms, and swarm intelligence, applied to the sequence alignment problem. A timeline of the developments is given for a historical view. Stochastic optimization methods have been in use since the mid-1990s in order to solve the sequence alignment problem. Other than the well-known techniques, stochastic optimization consists of many other algorithms. The solution is being pursued with an iterative alignment approach; first, an initial alignment of sequences is created, and then, in each iteration, the alignment of sequences is modified in order to minimize the error and maximize the overall alignment.

Simulated annealing (SA) was one of the first heuristics applied to sequence alignment [30]. Briefly, the system works as follows. Initially, the sequences are randomly aligned with the control parameter, temperature, which is set at a very high level. The system tends to move at low energy states with an analogy of altering from one solution to other neighboring solutions in order to find the optimal one. Next, depending on the move, the score is calculated and the move is either accepted or rejected by evaluating the score of the alignment. Subsequently, the temperature is slowly decreased. The process continues until it reaches a stopping criterion or until a predefined iteration count with an analogy that the freezing point has been reached. Variants of this method have been studied recently. Kim et al. suggested the use of simulated annealing to overcome some limitations of dynamic programming that require long computation time [31]. Keith et al. have utilized simulated annealing to find a consensus sequence that represents common features shared by most family members of the related sequences [32]. This algorithm searches pairs of sequences in order to find a resulting sequence that has the shortest distance from each of the sequences in the family. Hernandez-Guia et al. proposed a probabilistic algorithm to solve the multiple sequence alignment problem based on simulated annealing that exploits the representation of the multiple alignment between D sequences as a directed polymer in D dimensions [33].

Researchers found that evolutionary computation could be a good alternative to SA for the sequence alignment problem. EC, inspired by genetic processes, is a stochastic approach that is iteratively applied to find an optimal solution in the search space. Hence, EC began to be used in sequence alignment problems in the 1990s. The system is basically derived from the simple genetic algorithm. Over time, GA and many variants of it have been applied to the sequence alignment problem. Shyu et al. reviewed and presented the strengths and weaknesses of their recent work for sequence alignment using evolutionary computation [34]. The objective of genetic algorithms applied to the sequence alignment problem is to generate as many different multiple sequence alignments as possible and to select and iterate on the alignments that have good fitness values through the use of genetic operators, mutation, and crossover concepts. It starts initially with a population selected from a set of alignments. These alignments are the models of chromosomes that can be stated as candidate solutions. Crossover ensures 2 different alignments combined together to form a new one. A cutoff position on parent alignments is arbitrarily chosen and the left side of a parent combines with the right side of the other parent to produce a child. Each candidate solution has a fitness value. In a multiple sequence alignment problem, those that have good fitness values are the ones that have better sums of pair scores. The best alignments within the population are then crossed over or mutated so that new alignments are generated. Moreover, a selection from the population is performed; weak alignments are eliminated from the population and newly generated alignments are added to the population. This process continues until the stopping condition is satisfied. In GA, the stopping condition can be an iteration count or an appearance to attain the optimal fitness of alignment. Wayama et al. proposed GA for protein sequence alignment in which only up to a small number of residues could optimally be aligned [35]. Serial alignment by genetic algorithm (SAGA), introduced by Notredame, is one such approach that achieves very promising results but becomes slow when more than 20 sequences are used [36]. Zhang et al. implemented GA for sequence alignment to identify matches and mismatches, and the average computing time was much more efficient as compared with pairwise dynamic programming [37]. Chellapilla et al. proposed an evolutionary programming approach for sequence alignment, and the offspring generated by parents were probabilistically varied using several predefined variation operators [38]. Horng et al. used GA and reported that good performance and efficiency were achieved in the majority of datasets with high similar sequences with long lengths [39]. Omar et al. proposed an algorithm using GA for the multiple sequence alignment with simulated annealing and considered simulated annealing as an alignment improver [53]. Lee et al. presented a genetic algorithm by incorporating a local search with ant colony optimization [54].

Swarm intelligence, which is a branch of artificial intelligence, is another area becoming prominent in the field of optimization. Ant colony optimization (ACO) and particle swarm optimization (PSO) are the main swarm intelligence methods. ACO is inspired by the foraging behavior of biological ants that find the shortest path from a food source to the nest. PSO is also another major swarm intelligence method that models the swarm's social behavior, particularly in terms of fish schooling and bird flocking. A particle adapts its speed and direction to the changes in the speed and direction of neighboring particles within the swarm. ACO was one of the first swarm intelligence methods applied to the sequence alignment problem. It is one of the problems residing within the large range of discrete combinatorial optimization problems. Briefly, ants search for a pairwise alignment of 2 sequences in the following way. They move on the sequences to choose the matching characters. The selected probability of characters or gaps is determined by the matching score and pheromone deposited on matching characters. The ants ultimately tend to pursue finding the highest alignment score with an analogy of the shortest route attained. Once the optimal route is determined by the ants, then the stopping criterion is reached. In recent years, ACO has been applied to the sequence alignment problem in many ways. Moss et al. applied one of the first examples of ACO to sequence alignment and found that the results were efficient when sequences were similar [40]. Karpenko et al. considered applying ACO for finding the optimal multiple peptide alignment [41]. Multiple peptide alignment is considered to be ungapped and is used for the derivation of a position-specific scoring matrix for a given set of short protein peptides. Mikami et al. focused on improving multiple peptides by determining a better starting point for each sequence [42]. Y. Chen et al. proposed a divide-and-conquer approach based on ACO in order to solve multiple sequence alignment [43]. In this algorithm, a set of sequences are divided into several subsections vertically by splitting the sequences. Next, ACO is used to align sequences for each subsection. The alignment of original sequences can then be obtained by assembling the result of each subsection. Avoidance of the local optimum was achieved by adaptively adjusting the parameters and updating the pheromones. W. Chen et al. proposed a new method for pairwise alignment [44]. This method can find the optimal alignment without the use of a scoring matrix. The aligning process employs a plan by the modified dot plans. The next position will be selected by the amount of pheromone and the matching score of candidates. W. Chen et al. further studied the sequence alignment problem, this time, the multiple alignment case [45], by taking every possible aligning result into account by defining the representation of gap insertion and the scoring rule and determining the value of heuristic information in every optional path. The study employed ACO for finding optimal paths by using the multidimensional graph. PSO is the next swarm intelligence method applied to the sequence alignment problem. In recent years, due to its simple, fast, cost-effective properties, PSO has attracted the attention of many researchers and has been applied to many computational problems. In short, the system works for the pairwise alignment case as follows. A motion space for the particles is constructed as a matrix based on 2 sequences. Each particle starts from the left-top corner of the matrix and takes a path to reach the right-bottom corner in order to build the alignment. At each step, the particle can choose from 3 directions to move: down, right, and lower diagonal. During the moves on the search space, each particle moves toward the proximity of its best alignment and global best alignment. The velocity and position of the particles are updated at each iteration until a stopping criterion that determines the search is over. Ge et al. described an immune particle swarm optimization (IPSO), which was based on the models of the vaccination and receptor editing in immune systems [46]. First, hidden Markov models (HMMs) were trained and then integrated with IPSO. The tests were performed by benchmark alignment database (BAliBASE) in order to measure the performance of the proposed algorithm. Rodriguez et al. used PSO as an alignment improver [50]. The alignment was first obtained by using Clustal. They designed special representation and operators for PSO and performed their tests over families of proteins. Juang et al. presented an algorithm that consisted of DP and PSO working together [47]. This approach considers some drawbacks that arise from the inefficient computational and memory capacity, and the trapping local optimum problem in DP when dealing with more than 2 sequences. PSO is used for escaping the local optimum problem, and sequences are aligned progressively and iteratively by the pairwise DP. Xu et al. proposed a PSO method along with designing 3 operators, which were gap deletion, gap insertion, and local search, in order to solve MSA [48]. Lei et al. proposed an algorithm that used the idea of chaos systems with PSO [49]. The particle population was initialized using the chaotic variables with a uniform distribution over [0,1] and a logistic map was used to generate the sequences; the diversity of the population became stronger.

5. Results and discussion

Today, popular MSA tools still use DP, but the drawback is the excessive need for increase in computational capacity and memory in proportion to the increase in the number of sequences and sequence lengths. An increase in problem size can cause exponential growth in computation time. Figure 5 illustrates how exponential growth occurs in computational resources with an increase in the number of sequences. This is where stochastic optimization may take place in order to efficiently solve the sequence alignment problem.

In this study, we have surveyed several recent and past approaches of the existing literature for sequence alignment from the perspective of stochastic optimization. Due to the NP-hard nature of MSA, no optimal solution has yet been found. On the other hand, pairwise alignment may need excessive computational resources as the sequence size increase. Hence, there have been many methods proposed by researchers in order to retrieve and analyze sequences in a fast and efficient manner. Researchers have to consider 3 basic parameters in particular, the number of sequences, the average length of sequences, and the overall similarity of sequences, when developing an algorithm in order to efficiently solve the multiple sequence alignment problem. The computing time and memory space are the main characteristics that indicate the efficiency of the methods applied to the problem. A wide variety of opportunities can appear when researching the sequence alignment problem for finding an optimal solution to the problem. Research will hopefully allow the overcoming of the dramatic increase in computation time. Consequently, as researchers develop new or existing algorithms, new horizons will be opened and this might result in significant improvements in complex sequence alignments. In the following tables, we present the methods surveyed in this paper, aiming to find an optimal solution to the problem from the perspective of stochastic optimization. Table 1 gives further details of the indicative timeline for this work. Table 2 summarizes the performance evolutions of the references given in Table 1.

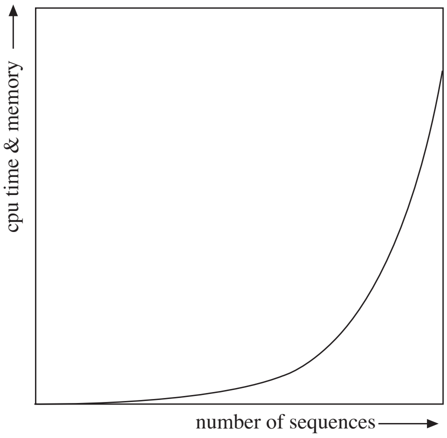


Figure 5. Exponential growth in computational resources with an increase in the number of sequences.

Table 1. Stochastic optimization methods applied for sequence alignment problem.

SO Method	Authors	Characteristics	Year	Ref.
SA	Ishikawa et al.	One of the first heuristics for the problem	1993	[30]
SA	Kim et al.	Simulated annealing based	1994	[31]
GA	Wayama et al.	Simple genetic algorithm employed	1995	[35]
GA	Notredame et al.	The use of operators with a scheduling scheme	1996	[36]
GA	Zhang et al.	Use of a genetic algorithm	1997	[37]
EP	Chellapilla et al.	Evolutionary programming-based	1999	[38]
SA	Keith et al.	Finding a consensus sequence	2002	[32]
ACO	Moss et al.	Multiple sequence alignment	2003	[40]
GA	Shyu et al.	Finding a consensus sequence	2004	[34]
ACO	Karpenko et al.	Applied on MHC class II molecules	2005	[41]
SA	Hernandez-Guia et al.	Indels without extra computational cost	2005	[33]
PSO	Ge et al.	Immune with HMM	2005	[46]
GA	Omar et al.	With simulated annealing	2005	[53]
ACO	Y. Chen et al.	Divide-and-conquer	2006	[43]
PSO	Rodriguez et al.	Alignment improver	2007	[50]
ACO	W. Chen et al.	Pairwise alignment	2008	[44]
PSO	Juang et al.	With DP	2008	[47]
GA	Lee et al.	With ACO	2008	[54]
PSO	Xu et al.	With the design of 3 operators	2009	[48]
ACO	W. Chen et al.	Multiple alignment	2009	[45]
PSO	Lei et al.	Chaotic approach	2009	[49]

Table 2. Performance evaluations of the applied stochastic optimization methods.

SO Method	Performance Evaluation	Ref.
SA	Utilized parallel algorithm to achieve reasonable solution times	[30]
SA	Overcomes some limitations of dynamic programming	[31]
GA	Succeeds to find optimal values of the genetic algorithms parameters	[35]
GA	Becomes slow when more than 20 sequences are used	[36]
GA	More efficient when compared with pairwise dynamic programming	[37]
EP	The quality of solutions is comparable to those obtained with Clustal	[38]
SA	Results are similar to and in many cases better than Clustal	[32]
ACO	Efficient when sequences are similar	[40]
GA	Iterations do not depend on the number of sequences aligned	[34]
ACO	More versatile in finding the best value prompter	[41]
SA	Satisfactory for small numbers of sequences	[33]
PSO	Faster alignment	[46]
GA	Simulated annealing as an alignment improver	[53]
ACO	Quality solution and reduced running time	[43]
PSO	Improves a sequence alignment previously obtained using Clustal	[50]
ACO	No use of scoring matrix	[44]
PSO	Good at escaping the local optimum	[47]
GA	Better performance than other algorithms when similarity of dataset is low	[54]
GA	More efficient compared with Clustal	[48]
PSO	Gives good results on MSA benchmarks and can improve the solution quality	[45]
ACO	Promising performance on datasets with different similarity	[49]

Simulated annealing was one of the first stochastic optimization methods applied to the sequence alignment problem. One of the major drawbacks of simulated annealing is that it can be trapped in a local optimal alignment, although there could exist a globally optimal aligned sequence. SA is also considered too slow to converge, and some researchers say that it can be considered as an alignment improver. On the other hand, for small numbers of sequences, GA is a good alternative for finding the optimal solution. However, as the number of sequences increases, it can fall behind optimal solutions and exponential growth in time may be observed. Swarm intelligence methods have some advantages, such as self-organization, robustness, and flexibility. Self-organization is the cooperation of individuals to accomplish difficult tasks without any strict top-down control. It is robust because the swarm can sustain its tasks even if some individuals fail to fulfill their tasks. Flexibility is the adaptation of individuals in the changing environment. Being equipped with these properties, swarm intelligence could be employed for solving relatively complex problems, as in sequence alignment.

6. Conclusion

This paper is a survey, organized as a timeline through the literature, aiming to provide a guide for researchers on the sequence alignment problem from the perspective of stochastic optimization and swarm intelligence. Biology is increasingly becoming an entirely data-driven science, making computation very essential for research. Sequence alignment is an essential task in computational biology. Alignment problems are considered to be NP-hard. An increase in problem size can cause exponential growth in computation time. Researchers develop new or existing algorithms for improving speed and sensitivity as the main goal. Although dynamic programming gives us the exact and optimal solutions, they are very slow. In order to gain speed, heuristics is considered as an option. More often than not, finding a near optimal sequence alignment would have a higher degree of importance than finding the exact or optimal solution. This is where stochastic optimization methods gain importance, and they first come to mind as a methodic approach to the problem. In conclusion, this survey lays down the sequence alignment problem from the point of view of stochastic optimization and swarm intelligence and presents distinctive features of each investigated method.

References

[1] A.R. Leach, Molecular Modelling: Principles and Application, Prentice Hall, 2001.

[2] M. Schena, D. Shalon, R.W. Davis, P.O. Brown, "Quantitative monitoring of gene expression patterns with a complementary DNA microarray", Science, Vol. 270, pp. 467-470, 1995.

[3] J. Skolnick, J.S. Fetrow, A. Kolinski, "Structural genomics and its importance for gene function analysis", Nature Biotechnology, Vol. 18, pp. 283-287, 2000.

[4] V. Uslan, İ.Ö. Bucak, "Microarray image segmentation using clustering methods", Mathematical and Computational Applications, Vol. 15, pp. 240-247, 2010.

[5] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, New York, W.H. Freeman, 1990.

[6] A.N. Arslan, "Sequence Alignment", Biyoinformatik-II, pp. 101-114, 2004.

[7] C.L. Strope, S.D. Scott, E.N. Moriyama, "Gap profiling: scoring indels in multiple sequence alignment", Biotechnology and Bioinformatics Symposium, 2009.

[8] M.O. Dayhoff, R.M. Schwartz, B.C. Orcutt, "A model of evolutionary change in proteins", in: Atlas of Protein Sequence and Structure, Vol. 5, pp. 345-352, 1978.

[9] S. Henikoff, J.G. Henikoff, "Amino acid substitution matrices from protein blocks", Proc. Nat. Acad. Sci., pp. 10915-10919, 1992.

[10] M. Vingron, M.S. Waterman, "Sequence alignment and penalty choice. Review of concepts, case studies and implications", J. Mol. Biol., Vol. 235, pp. 1-12, 1994.

[11] D.G. Higgins, P.M. Sharp, "CLUSTAL: a package for performing multiple sequence alignment on a microcomputer", Gene, Vol. 73, pp. 237-244, 1988.

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

[13] European Bioinformatics Institute, http://www.ebi.ac.uk/Tools/clustalw2/index.html.

[14] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, E.W. Sayers, "GenBank", Nucleic Acids Research, Vol. 38, database issue D46-D51, 2010.

[15] B. Boeckmann, A. Bairoch, R. Apweiler, M.C. Blatter, A. Estreicher, E. Gasteiger, M.J. Martin, K. Michoud, C. O'Donovan, I. Phan, S. Pilbout, M. Schneider, "The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003", Nucleic Acids Research, Vol. 31, pp. 365-370, 2003.

[16] S.B. Needleman, C.D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequences of two proteins," J. Mol. Biol, Vol. 48, pp. 443-453, 1970.

[17] M. Murata, J.S. Richardson, J.L. Susman, "Simultaneous comparison of three protein sequences", Proc. Natl. Acad. Sci. USA, Vol. 82, pp. 3073-3077, 1985.

[18] D.F. Feng, R.F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees", J. Mol. Evol., Vol. 25, pp. 351-360, 1987.

[19] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimization by simulated annealing", Science, Vol. 220, pp. 671-680, 1983.

[20] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, 1989.

[21] J.H. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975.

[22] R. Beckers, J.L. Deneubourg, S. Goss, "Trails and u-turns in the selection of a path by the ant *Lasius niger*", Journal of Theoretical Biology, Vol. 159, pp. 397-415, 1992.

[23] S. Goss, S. Aron, J.L. Deneubourg, J.M. Pasteels, "Self-organized shortcuts in the Argentine ant", Naturwissensschaften, Vol. 76, pp. 579-581, 1989.

[24] M. Dorigo, L.M. Gambardella, "Ant colonies for the traveling salesman problem", Biosystems, Vol. 43, pp. 73-81,1997.

[25] M. Dorigo, L.M. Gambardella, "Ant Colony System: A cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computing, Vol. 1, pp. 53-66, 1997.

[26] L.M. Gambardella, M. Dorigo, "Solving symmetric and asymmetric TSPs by ant colonies", in: Proceedings of the IEEE Conference on Evolutionary Computing, 1996.

[27] R.C. Eberhart, J. Kennedy, "Particle swarm optimization", in: Proc. of the IEEE Int. Conf. on Neural Networks, Piscataway, NJ, pp. 1942-1948, 1995.

[28] Y. Shi, R.C. Eberhart, "Empirical study of particle swarm optimization", in: Proc. of the Evolutionary Comp., 1999.

[29] C. Notredame, "Recent progress in multiple sequence alignment: a survey", Pharmacogenomics, Vol. 3, pp. 131-144, 2002.

[30] M. Ishikawa, T. Toya, M. Hoshida, K. Nitta, A. Ogiwara, M. Kanehisa, "Multiple sequence alignment by parallel simulated annealing", Comp. Applic. Biosci., Vol. 9, pp. 267-273, 1993.

[31] J. Kim, S. Pramanik, M.J. Chung, "Multiple sequence alignment using simulated annealing", Comp. Appl. Biosci., Vol. 10, pp. 419-426, 1994.

[32] J.M. Keith, P. Adams, D. Bryant, D.P. Kroese, K.R. Mitchelson, D.A.E. Cochran, G.H. Lala, "A simulated annealing algorithm for finding consensus sequences", Bioinformatics, Vol. 18, pp. 1494-1499, 2002.

[33] M. Hernandez-Guia, R. Mulet, S. Rodriguez-Perez, "Simulated annealing algorithm for the multiple sequence alignment problem: the approach of polymers in a random medium", Phys. Rev. E. Stat. Nonlin. Soft Matter Phys., Vol. 72, 031915 (epub), 2005.

[34] C. Shyu, L. Sheneman, J.A. Foster, "Multiple sequence alignment with evolutionary computation", Genetic Programming and Evolvable Machines, Vol. 5, pp. 121-144, 2004.

[35] M. Wayama, K. Takahashi, T. Shimizu, "An approach to amino acid sequence alignment using a genetic algorithm", Genome Informatics, Vol. 6, pp. 122-123, 1995.

[36] C. Notredame, D.G. Higgins, "SAGA: sequence alignment by genetic algorithm", Nuc. Acids Res., Vol. 24, pp. 1515-1524, 1996.

[37] C. Zhang, A.K. Wong, "A genetic algorithm for multiple molecular sequence alignment", Comput. Applic. Biosci., Vol. 13, pp. 565-581, 1997.

[38] K. Chellapilla, G.B. Fogel, "Multiple sequence alignment using evolutionary programming", in: Proceedings of the IEEE Congress on Evolutionary Computation, Vol. 1, pp. 445-452, 1999.

[39] J.T. Horng, L.C. Wu, C.M. Lin, B.H. Yang, "A genetic algorithm for multiple sequence alignment", Soft Comput., Vol. 9, pp. 407-420, 2005.

[40] J.D. Moss, C.G. Johnson, "An ant colony algorithm for multiple sequence alignment in bioinformatics", Springer, Artificial Neural Networks and Genetic Algorithms, pp. 182-186, 2003.

[41] O. Karpenko, J. Shi, Y. Dai, "Prediction of MHC class II binders using the ant colony search strategy", Artificial Intelligence in Medicine, Vol. 35, pp. 147-156, 2005.

[42] A. Mikami, J. Shi, "A modified algorithm for sequence alignment using ant colony system", IPSJ Transactions on Bioinformatics, Vol. 2, pp. 63-73, 2009.

[43] Y. Chen, Y. Pan, J. Chen, W. Liu, L. Chen, "Multiple sequence alignment by ant colony optimization and divide-and-conquer", Springer, Computational Science, Vol. 3992, pp. 646-653, 2006.

[44] W. Chen, B. Liao, W. Zhu, H. Liu, Q. Zeng, "An ant colony pairwise alignment based on the dot plots", Journal of Computational Chemistry, Vol. 30, pp. 93-97, 2008.

[45] W. Chen, B. Liao, W. Zhu, X. Xiang, "Multiple sequence alignment algorithm based on a dispersion graph and ant colony algorithm", J. Comput. Chem., Vol. 30, pp. 2031-2038, 2009.

[46] H.W. Ge, Y.C. Liang, "A hidden Markov model and immune particle swarm optimization-based algorithm for multiple sequence alignment", Proc. Advances in Artificial Intelligence, pp. 756-765, 2005.

[47] W.S. Juang, S.F. Su, "Multiple sequence alignment using modified dynamic programming and particle swarm optimization", Journal of the Chinese Institute of Engineers, Vol. 31, pp. 659-673, 2008.

[48] F. Xu, Y. Chen, "A method for multiple sequence alignment based on particle swarm optimization", Springer, Vol. 5755, pp. 965-973, 2009.

[49] X.J. Lei, J.J. Sun, Q.Z. Ma, "Multiple sequence alignment based on chaotic PSO", in: Proc. of the Computational Intelligence and Intelligent Systems, Springer, Vol. 51, pp. 351-360, 2009.

[50] P.F. Rodriguez, L.F. Nino, O.M. Alonso, “Multiple sequence alignment using swarm intelligence”, Inter. J. of Comp. Int. Research, 2007.

[51] C. Blum, “Ant colony optimization: introduction and recent trends”, Physics of Life Reviews, Vol. 2, pp. 353-373, 2005.

[52] T.F. Smith, M. Waterman, “Identification of common molecular subsequences”, J. Mol. Biol., Vol. 147, pp. 195-197, 1981.

[53] M.F. Omar, R.A. Salam, R. Abdullah, N.A. Rashid, “Multiple sequence alignment using optimization algorithms”, International Journal of Computational Intelligence, Vol. 1, pp. 81-89, 2005.

[54] Z.Y. Lee, S.F. Su, C.C. Chuang, K.H. Liu, “Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment”, Applied Soft Computing, Vol. 8, pp. 55-78, 2008.

[55] J.D. Thomson, F. Plewniak, O. Poch. “BAliBASE: A benchmark alignment database for the evaluation of multiple alignment programs,” Bioinformatics, Vol. 15, pp. 87-88, 1999.