MSAGA: Multiple Sequence Alignment Using Genetic Algorithms

P. Y. Yin and S. J. Shyu

Department of Computer Science
Ming Chuan University
Taoyuan 333, Taiwan

Abstract

Multiple nucleic acid or amino acid sequences alignment is one of the most commonly used techniques in bioinformatics. It helps find out homology between new sequences and existing ones. In this paper, we propose a new approach for multiple sequence alignment using genetic algorithms. The aligning order of the sequences is represented by a spanning tree, and the optimum (or very close to it) alignment is yielded by interchanging information between fitter spanning trees. The experimental results manifest that the performance of the proposed method is better than those of existing methods.

Keyword: *Bininformatics; Computational molecular biology; Genetic algorithms; Spanning trees; deoxyribonucleic acid (DNA); Multiple sequence alignment;*

1. Introduction

Multiple sequence alignment (MSA) is a fundamental and challenging problem in computational molecular biology. It plays a key role in computing similarity and in finding highly conserved subsequences among a set of deoxyribonucleic acid (DNA) sequences. Similar subsequences often implicitly imply a significant functional or structural resemblance in biology. The DNA sequences are reordered and mutated during evolution, and thus show up pattern repeats either within a single genome or across the genomes of a variety of species. As a result, the MSA is one of the most commonly used methods for inferring biological structures and functions. More often, it is the first step of many tasks in computational biology involving fragment assembly, evolutionary tree reconstruction, and genome analysis. For a comprehensive review on the MSA, please refer to (Chan *et al.*, 1992; Gusfield, 1997; Waterman, 1995).

Though the optimal alignment of two sequences could be found via the dynamic programming approach (Needleman and Wunsch, 1970; Waterman, 1995), the problem of computing the minimum cost for the MSA has been shown to be NP-hard (Wang and Jiang, 1994). Solutions thus rely on heuristics for practical consideration. Meanwhile many criteria were proposed to measure the goodness of the MSA result. One of the most popular criteria is the *sum-of-pairs* (Gusfield, 1993; Pevzner, 1992). It finds an alignment which minimizes the total sum of all distances between every pair of aligned sequences induced by this alignment.

The heuristics proposed to approximate the MSA can be classified into two main categories: (1) iterative 2-way alignment, and (2) stochastic methods. Many approaches belonging to the first category were presented in the literature (Bains, 1986; Feng and Doolittle, 1987; Corpet., 1988; Chan, *et al.*, 1992; Higgins, *et al.*, 1996; Shyu, *et al.*, 2001; Genetic Computer Group (GCG)). Some of them provide the theoretical analysis on performance guarantee (Gusfield, 1993; Bafna, Lawler and Pevzner, 1997). However, quite few in the literature for the latter category have shown their effectiveness (Ishikawa *et al.*, 1993; Kim *et al*., 1994; Notredame and Higgins,

1996; Notredame *et al*., 1997).

In this paper, we shall present an algorithm for the MSA based on the genetic algorithms. Also, experiments are conducted to test the effectiveness of the proposed approach.

2. Iterative 2-way Alignment and Progressive Alignment

2.1 Iterative 2-way Alignment

Let $S_1$and S_2 be two sequences of length n and m, and s_{1i} and s_{2j} be the ith and jth character of the corresponding sequences, respectively. Let $\sigma(s_{1i}, s_{2j})$ denote the dissimilarity score between characters s_{1i} and s_{2j}. The dynamic programming algorithm for deriving the optimal alignment of the two sequences (2-way alignment) can be illustrated by the following formulas:

$$
\begin{aligned}
&A(0, 0) = 0\\
&A(i, 0) = A(i\text{-}1, 0) + \sigma(s_{1i}, \sim)\\
&A(0, j) = A(0, j\text{-}1) + \sigma(\sim, s_{2j})\\
&A(i, j) = \min\begin{cases} A(i-1, j-1) + \sigma(s_{1i}, s_{2j}) \\ A(i-1, j) + \sigma(s_{1i}, \sim) \\ A(i, j-1) + \sigma(\sim, s_{2j}) \end{cases}
\end{aligned}
$$

where "~" denotes a gap within the aligned sequence and $A(n, m)$ is the optimal alignment score.

Now, suppose that k sequences $S_1, S_2, \ldots, S_k$ are considered simultaneously. A straightforward extension of the above algorithm can be easily performed to find an optimal alignment of the k sequences (Sankoff, 1975; Waterman *et al.*, 1976). However, the computational time involved is proportional to $2^k \prod_{i=1}^{k} |S_i|$ where $|S_i|$ denotes the length of sequence S_i, which is unbearable for practical usage. Therefore, many efficient heuristics are proposed based on applying the 2-way alignment iteratively (Gusfield, 1993; Shyu *et al*., 2001).

2.2 Progressive Alignment

When the result of a 2-way alignment is treated as a single entity, the idea of the progressive alignment arises (Hogeweg and Hesper, 1984; Feng and Doolittle, 1987; Taylor, 1988; Corpet, 1988; Higgins and Sharp, 1989). The algorithm proceeds as follows. First, two of the k sequences are aligned optimally, and the resulting sequence of the commonly aligned characters replaces the two original sequences. This reduces the problem of aligning k sequences to a problem of aligning k-1 sequences. The process is applied iteratively until only one aligned sequence remains. When dealing with the gaps, we follow the '*once a gap, always a gap*' principle (Feng and Doolittle, 1987).

What makes the progressive alignment possible is that the dynamic programming algorithm for aligning two sequences can be well performed when one or both of the sequences are the alignments themselves of the original sequences. Let X and Y be the two alignments to be aligned. Let $x_{i,j}$ and $y_{i,j}$ denote the character at the jth position of the ith sequences in X and Y, respectively. We give the formulation of the dynamic programming for the progressive alignment as follows.

$$A(0, 0) = 0$$

$$A(i, 0) = A(i\text{-}1, 0) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(x_{r,i}, \sim) = A(i\text{-}1, 0) + |Y| \times \sum_{r=1}^{|X|}\sigma(x_{r,i}, \sim)$$

$$A(0, j) = A(0, j\text{-}1) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(\sim, y_{s,j}) = A(0, j\text{-}1) + |X| \times \sum_{s=1}^{|Y|}\sigma(\sim, y_{s,j})$$

$$A(i, j) = \min\begin{cases} A(i-1, j-1) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(x_{r,i}, y_{s,j}) \\ A(i-1, j) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(x_{r,i}, \sim) \\ A(i, j-1) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(\sim, y_{s,j}) \end{cases}$$

$$= \min\begin{cases} A(i-1, j-1) + \sum_{r=1}^{|X|}\sum_{s=1}^{|Y|}\sigma(x_{r,i}, y_{s,j}) \\ A(i-1, j) + |Y| \times \sum_{r=1}^{|X|}\sigma(x_{r,i}, \sim) \\ A(i, j-1) + |X| \times \sum_{s=1}^{|Y|}\sigma(\sim, y_{s,j}) \end{cases}$$

Note that all of the characters in each column of an alignment would be aligned with all of the characters in each column of another alignment.

Once an aligning order of the sequences is given, the progressive alignment approach can deliver an approximate result for the local optimum. However, applying the progressive alignment approach for each of the possible aligning orders is not practical. In the next section, we propose to use a genetic algorithm to effectively explore the aligning order space and enable the progressive alignment approach to approximate the global optimum.

3. The Proposed Algorithm: MSAGA

Here, we realize the MSA by another perspective. Consider a complete graph, if we annotate each node as a sequence and every two nodes are connected by an edge with a weight representing the alignment score between the corresponding sequences, the iterative 2-way alignment can be actually illustrated by a spanning tree. A spanning tree is a tree that spans all the nodes. We propose to explore the space of all

possible spanning trees using a genetic algorithm (GA). GAs are stochastic search algorithms which simulate the biology evolution (Goldberg, 1989). They perform parallel search by a population of randomly generated solutions, called chromosomes, to the solution space. The average fitness value of the population is improved by conducting reproduction that is consisting of two operators: crossover and mutation. Reproduction mimics the natural selection based on Darwinian survival of the fittest. These selected individuals interchange parts of their information by crossover and occasionally alter the gene allele by mutation.

Our idea proceeds as follows. Initially, a population of randomly generated spanning trees over the sequence nodes is established. A random spanning tree can be generated by adding a new edge to the tree one by one until the spanning tree is obtained in such a way that the insertion of the edge does not create a cycle. The fitness value of a spanning tree is defined as the inverse of the alignment score of the sequences according to the weight-ascending order on the edges using the progressive approach. Then, the spanning trees are selected to form a mating pool for reproduction with the probability proportional to their fitness values. There are two genetic operators, namely the crossover and the mutation, to fulfill the reproduction process. In particular, we design the genetic operators for spanning trees as follows.

Crossover. Randomly choose two spanning trees T_1 and T_2 as parents from the mating pool as illustrated in Fig. 1(a). The common edges of T_1 and T_2 are firstly extracted to establish the initial form of their offsprings S_1 and S_2 which could be a single tree or a forest (see Fig. 1(b)). Then, S_1 and S_2 iteratively select the next edges from their parents until they become spanning trees. At each iteration, a random number q is drawn within (0, 1). If $q < 0.5$ then S_1 chooses an edge from T_1 and S_2 chooses an edge from T_2. Otherwise, S_1 chooses an edge from T_2 and S_2 chooses an edge from T_1. However, the insertion of the chosen edge can not create a cycle in the offspring (see Figs. 1(c)-1(d)). As such, the offspring inherits mixed information from their parents in the hope that it will deliver a better alignment score.

Mutation. Every reproduced offspring has a very low probability of undergoing mutation. Let offspring S_1 in the previous example is determined for mutation. First, arbitrarily remove an edge from S_1 and yield two disjoint node sets denoted by V_1 and V_2 (see Fig. 2(a)). Second, create a new edge (i, j) with weight $w_{i,j}$ such that $i \in V_1$ and $j \in V_2$ as shown in Fig. 2(b), that is, edge (i, j) will connect V_1 and V_2 and reproduce a spanning tree again. Mutation is a necessary evolution step which involves new information not attainable from the biological parents.

The population of spanning trees is reproduced from generation to generation. After a given number of generations, the spanning tree with the best alignment score over the entire evolution is output as the final solution. We present the MSAGA approach in Table 1.

Table 1	The multiple sequence alignment by genetic algorithms (MSAGA).
Step 1:	Construct a complete graph $G(V, E)$ where V is a set of nodes and E is a set of edges. Each node represents a sequence and each edge is associated with a weight representing the alignment score of the corresponding sequences.
Step 2:	Generate a population of random spanning trees of G.
Step 3:	Select spanning trees from the current population to form a mating pool according to their fitness values.
Step 4:	Reproduce the next generation by applying crossovers and mutations.
Step 5:	Repeatedly performing Steps 3-4 until a given number of generations is reached.
Step 6:	Output the spanning tree with the best alignment score over the entire evolution.

3. Results

In this section, we will give the comparative performance of the proposed MSAGA approach with the Gusfield's method and the GCG method. Several test cases of the MSA of DNA sequences are obtained from the public database (ftp://ftp.ebi.ac.uk/pub/databases/embl/align). The number of sequences in the test case ranges from 12 to 44, and the maximum length of the sequences is 714. Table 2 lists the obtained alignment scores by using the competing approaches. Remind that the smaller the score, the better the alignment. It is seen that the GCG method produces the worst results, the Gusfield's solutions are moderate, and the proposed MSAGA yields the best alignment scores for almost all the test cases. To visualize the relative quality of the delivered alignment score, we set the GCG result as a base line and compute the ratio to the others. Fig. 3 illustrates that the proposed MSAGA method provides a substantial improvement over the existing methods.

Table 2 The alignment scores by using GCG method, Gusfield's method, and the proposed MSAGA approach.

Test cases	GCG method	Gusfield's method	MSAGA
1	12238	12559	12068
2	140624	105935	102754
3	16719	10393	9970
4	6754	6182	5920
5	17283	16021	14943
6	13345	14345	13347
7	3817	3817	3817

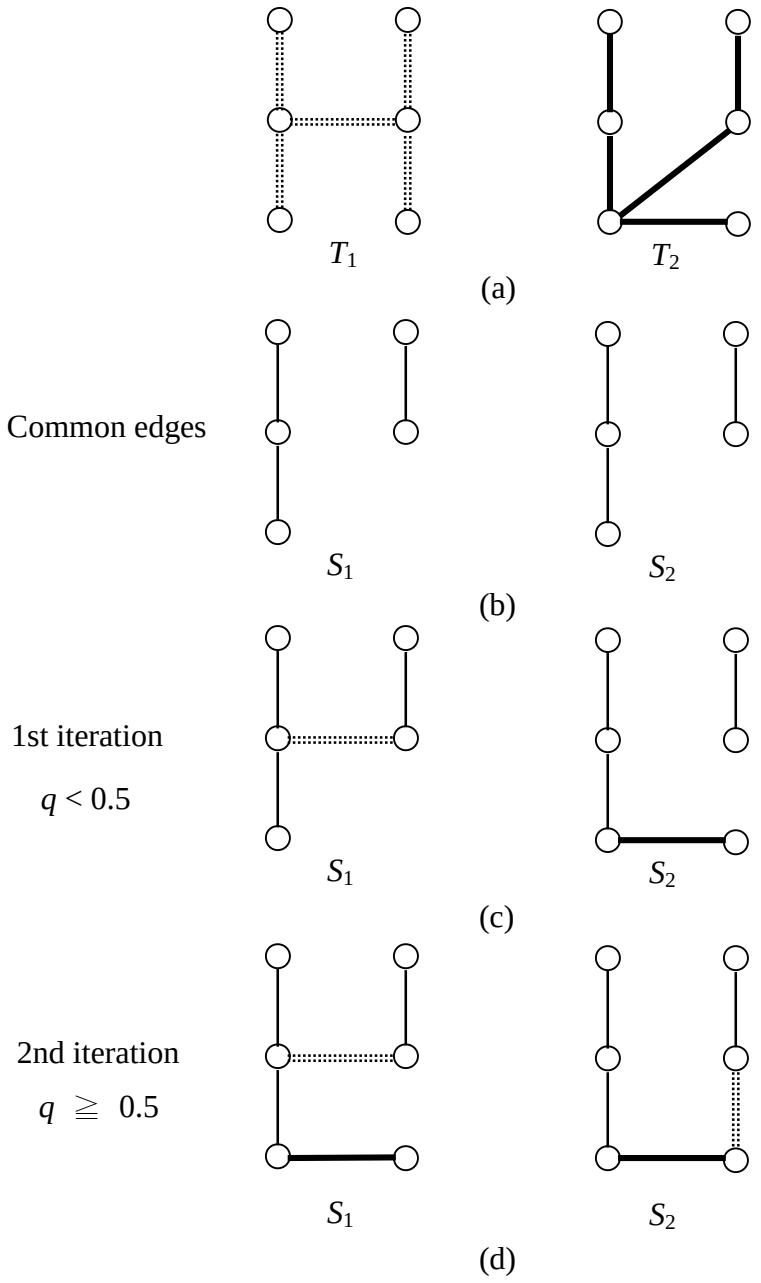

Fig. 1 Crossover of two spanning trees.

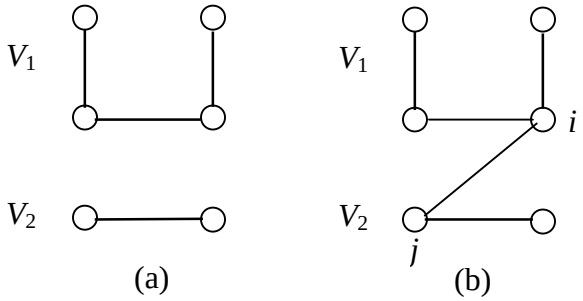

Fig. 2 Mutation of a spanning tree.

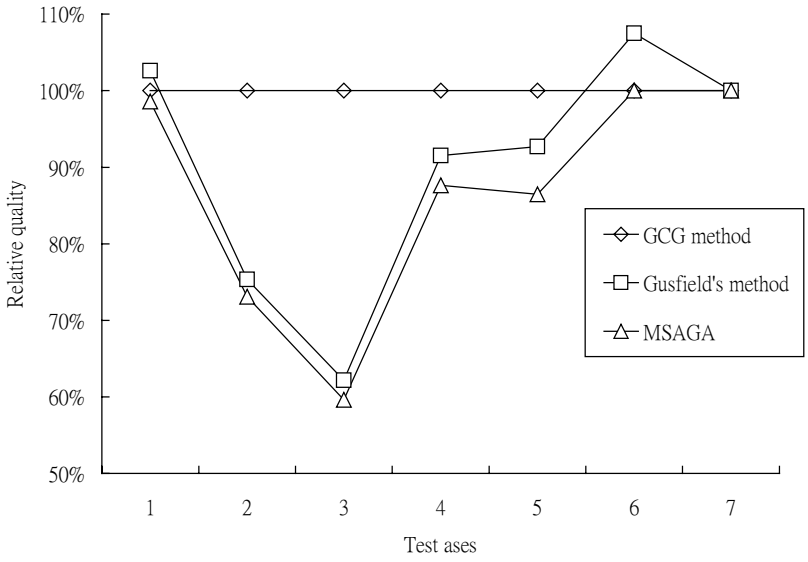

Fig. 3 The relative quality ratio of the delivered alignment score by setting the GCG result as a base line.

Reference

Bains, W. (1986). Multan: a program to align multiple DNA sequences. Nucleic Acids Research, 14:159-177.

Chan, S. C., Wong, A. K. C. and Chiu, D. K. T. (1992) A survey of multiple sequence comparison methods. *Bulletin of Mathematical Biology*, 4, 563-598.

Corpet, F. (1988). Multiple sequence alignment with hierarchical clustering. *Nucleic Acids Research*, 16, 22, 10881-10890.

Feng, D.-F. and Doolittle, R. F. (1987). Progressive alignment as a prerequisite to correct phylogenetic trees. *Journal of Molecular Evolution*, 25, 351-360.

Genetic Computer Group (GCG), http://helix.nih.gov/science/bioinfo/gcg.html.

Goldberg, D. E. (1989). Genetic Algorithms: Search, Optimization and Machine Learning. Addison-Wesley, Reading, MA.

Higgins, D. and Sharp, P. (1989). Fast and sensitive multiple sequence alignments on a microcomputer. *CABIOS*, 5,151-153.

Higgins, D. G. , Thompson, J. D. and Gibson, T. J. (1996). Using CLUSTAL for multiple sequence alignments. Methods in Enzymology, 266:383-402, 1996.

Hogeweg, P. and Hesper, B. (1984). The alignments of sets of sequences and the construction of phylogenetic trees: an integrated method. *Journal of Molecular Evolution*, 20, 175-186.

Ishikawa, M. and Yoya, T., Hoshida, M., Nitta, K., Ogiwara, A. and Kanehisa, M. (1994). Multiple sequence alignment by parallel simulated annealing. *CABIOS*, 9(3), 267-273.

Kim, J., Pramanik, S. and Chung, M. J. (1994). Multiple sequence alignment using simulated annealing. *CABIOS*, 10(4), 419-426.

Sankoff, D. (1975). Minimal mutation trees of sequences. *SIAM Journal of Applied Mathematics*, 28(1), 35-42.

Shyu, S. J., Tsai, Y. T. and Lee, R. C. T. (2001). The minimal spanning tree preservation approaches for DNA multiple sequence alignment and evolution tree construction, Submitted for publication.

Taylor, W. R. (1988). A flexible methods to align large numbers of biological sequences. *Journal of Molecular Evolution*, 28, 161-169.

Thompson J.D., Higgins D.G., Gibson T.J. (1994). CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673-4680. (http://www.ebi.ac.uk/clustalw/)

Thompson, J., Plewniak, F. and Poch, O. (1999). BAliBASE: A benchmark alignments database for the Evaluation of multiple sequence alignment programs. *Bioinformatics*, 15, 87-88.

Waterman, M., S. (1995). *Introduction to Computational Biology. Maps, Sequence and Genomes*. Chapman & Hall, London, UK.

Waterman, M., S., Smith, T. F. and Beyer, W. A. (1976). Some biological sequence metrics, *Advance Mathematics*, 20, 367-387.