Assiut University Journal of Multidisciplinary Scientific Research (AUNJMSR)
Faculty of Science, Assiut University, Assiut, Egypt.
Printed ISSN: 2812-5029
Online ISSN: 2812-5037
Vol. 51(2): 103 – 114 (2022)
https://aunj.journals.ekb.eg/

Brain Storm Optimization for Multiple Sequence Alignment problem

Alaa Fahim [1], Nehad Abdelraheem [2]

Department of Mathematics, Faculty of Science, Assiut University, Egypt.
Faculty of computer and information, Assiut University, Egypt.

* Corresponding author: alaa@aun.edu.eg

ARTICLE INFO

Article History:
Received:30/11/2021
Accepted: 30/3/2022
Online: 1/5/2022

Keywords:
Multiple sequence alignment; Brain storm optimization

ABSTRACT

Brain Storm Optimization (BSO) is one of the most effective swarm intelligence methods for finding optimality in optimization problems by simulating the human brainstorming process. The BSO approach has been effectively used to a wide range of employed in several real-world issues. This study focuses on the use of a hybrid approach in conjunction with the idea of self-organization for multiple sequence alignment (MSA) problems. The term “self-organization” refers to a structure that operates without the need for external intervention. To demonstrate the efficacy of the algorithm, we applied BSO to MSA and evaluated the resulting alignment using the sum-of-pair score (SPS). The efficiency of BSO was evaluated using Benchmark Alignment Database (BAliBASE) reference multiple sequence alignments. The BSO method outperformed some other metaheuristic methods and achieves better alignments than existing MSA techniques.

INTRODUCTION

The alignment of amino acid and DNA sequences is one of the first steps that uses molecular evolutionary analyzes and bioinformatics. One of the most used bioinformatics tools is the pair-wise alignment. It is used to measure the similarities between sequences and all of these are found in the database BLAST similarity search. In a multiple sequence alignment (MSA), the alignment of more than two sequences reflects the genetic evolutionary history between the sequences by assuming that all the sequences that are being compared are originally derived from a common ancestral sequence. The construction MSA process is based mainly on finding homogeneous and similar places between sequences and some of them by inserting gaps in the sequence to find homogeneity in a faster way, and these gaps are the historical development of the sequence. The process of finding homogenates was not limited to inserting gaps only, but these gaps may be the deletion of a nucleotide or an amino acid from the sequence. The purpose of building MSA is to know the historical evaluation of this group of sequences. The alignment of amino acid and DNA sequences is one of the first steps that uses molecular evolutionary analyzes and bioinformatics. One of the most used bioinformatics tools is the pair-wise alignment. It is used to measure the similarities between sequences, and all of these are found in the database BLAST similarity search. In MSA, the alignment of more than two sequences reflects the genetic evolutionary history between the sequences by assuming that all the sequences compared are originally derived from a common ancestral sequence. The MSA construction process is based mainly on finding homogeneous and similar places between sequences and some of them by inserting gaps in the sequence to find homogeneity in a fast way, and such gaps are the historical development of the sequence. The process of finding homogenates is not limited to inserting gaps only, but the gaps may be the deletion of a nucleotide or an amino acid from the sequence. The purpose of building MSA is to know the historical evaluation of this group of sequences.

There are many difficulties in how to evaluate the quality of the algorithm or how the evaluation is conducted despite the importance of forming and constructing good MSA sequences. This is because evaluating sequence quality relies primarily on the following two main components: a benchmark dataset and a scoring method. Although there is a dynamic programming algorithm [5] that assures a mathematically optimal alignment, the method is limited to a few of short sequences since the computational power necessary for bigger alignments is prohibitively expensive. To address this issue, a several heuristic approaches have been developed, resulting in a several programs employing fundamentally distinct strategies (progressive, iterative, mixed, etc.) based on vastly diverse algorithms. Traditionally the most popular strategy is the progressive alignment method. Multiple Alignments are built up progressively by aligning the closest sequences first and successively adding them to the more distant ones. There are several alignment programs based on this method, including MULTALIGN [1], MULTAL [3], DIALIGN [9] and CLUSTALX [16], which provides a graphical interface for CLUSTALW [16]. Simulated annealing [17], Gibbs sampling [10] and genetic algorithms [7] are examples of stochastic optimization techniques. Simulated annealing has been employed for multiple alignments on many occasions, but it is time consuming and generally works effectively only as an alignment improver. This paper presents an application of Brain Storm Optimization (BSO) algorithm [14] in one of the most important biological problems that consider NP hard a problems. BSO is a new type of particle swarm optimization proposed by Shi. It was inspired by the most intelligent organism, humans. After that, many studies have been conducted to develop the algorithm and make it more efficient. BSO depends on the following two basic functions, divergence and convergence. The ability to learn and develop capabilities are the two basic functions that exhibited by BSO. Divergence and convergence correlate with learning and developing respectively. For capacity development, its function is to search for new possibilities, and the solutions are the highest, for learning ability to learn, its main task is to search for new solutions, but from the current solution, which depends on one member of the population. The BSO-MSA algorithm has been applied to examples from Benchmark Alignment Database (BaliBASE) [2], BaliBASE is database that maintains many multiple sequence alignments. The data have been manually pollinated for evaluation, testing and comparison of alignment programs. Whatever the left of the paper is sorted as follows: First, presenting the preliminaries concepts of BSO algorithms. Then, presenting BSO and how to manage it and showing the consequence of the BSOMSA strategy and the correlation between the proposed technique and different strategies. Finally, conclusions are presented.

1. Preliminaries:

This section provides a quick overview of some basic aspects of the BSO algorithm to solve the MSA problem. we also present a brief history about the MSA problem.

1.1. Brain storm optimization Algorithm

The BSO algorithm, like other swarm intelligence optimization algorithms, has been modeled after a brainstorming process. Brainstorming is widely utilized in many companies to improve creativity and thinking and to identify broad-range solutions, and this approach effectively encourages creative thinking [4]. This algorithm, like other algorithms in this family, generates numerous solutions at first, but the name of the idea is known to the solutions created. However, to discover the global solution in this generation, BSO follows several principles, such as clustering ideas, replacing ideas within each team, and building operators. There are five major operations of the BSO algorithm in Figure 1, which include:

- Population Initialization.
- Evaluating Individuals.
- Clustering Individuals.
- Disrupting Cluster Centers.
- Updating Individuals.

First, the search space is randomly seeded with N concepts. Then, each idea is examined, and its fitness value is determined. In the initialization phase, N ideas are produced randomly inside the search space using a normal distribution, and the size of the ideas generated in each iteration must be fixed. Next, similar to other evolutionary algorithms and swarm intelligence algorithms, each idea is assessed in each generation. and cluster centers are established. When concepts are clustered and the number of clusters is smaller than the number of ideas, M is formed. where m is a positive integer smaller than N.

There are several clustering algorithms used in clustering solutions and concepts, but BSO clusters uses K-means. Individual step update consists of two sub-operations expressed as follows:

$$x_{new}^{i} = x_{old}^{i} + \zeta(t) + random(t)$$
$$x_{old}^{i} = \omega_1 * x_{old1}^{i} + \omega_2 * x_{old2}^{i}$$

Where x_{old}^{i} is the summation of i-dimensional ofx_{old1}^{i} and x_{old2}^{i}weights, and ω_1 and ω_2 are the coefficients for weighting two existing individuals ω1 and ω2 2 equal 0 if new individual xi new is generated depending on existing individual x_{old}^{i}. If it depends on two existing individuals x_{old1}^{i} and x_{old2}^{i}, then the coefficient ζ(t) is randomly generated by one possible function:

$$\xi(t) = logsig \left| \frac{\frac{T}{2} - t}{k} \right| * random(t)$$

Where logsig() is a logarithmic sigmoid transfer function, T is the maximum number of iterations, and t is the current iteration number, k changes the slope of logsig(), and random() is a random value within (0,1).

1.2. Multiple Sequence Alignment (MSA)

Encoding of DNA, RNA and protein sequences change over time, and these sequences are developed based on mutations that occur over time. Mutation occurring in nucleotides or amino acids by insertion, deletion, or substitutions are the simplest types of mutation. When such mutations occur in one or two identical sequences, the length of the sequence is altered, which implies that a new number of nucleotides or amino acids is generated. Because observing mutation sites is challenging, it is important to establish the positions of such mutations to determine the original locations of the nucleotides or amino acids. This is known as the alignment process, and while it may be easy for many people, it is a very difficult procedure due to the limited understanding of the units that observed for each aligned position. During the alignment process, gaps in the sequence are created at more than one site with no substantial change in the comparable sequence. Alignment is a biological process that involves the alignment of homogenous residues that have a common evolutionary origin. When the alignment process begins, we presume that evolution will be minimal, and therefore evolutionary modifications needed by the alignment will be minimal. This is done to get the best alignment [6]. The alignment can be either pairwise, consisting of only two sequences, or multiple, which consists of more than two sequences. Several algorithms such as the Needleman-Wunsch [8] and Smith-Waterman [15] algorithms have been employed to process pairwise. In practice, MSA is a general case of pairwise alignment. However, the difficulties emerging from the MSA are not a linear extension of pairwise alignment. In most situations, MSA is handled with by repeated pairwise merging.

MATERIALS AND METHODS

In MSA, the gaps in the representation of encoding ideas considered. The number of gaps to be inserted into chromosome is known so that all of the aligned sequences are of equal length. Integers are used to represent the gaps entered in the sequence.

- Idea Representation

The idea represents an individual solution in the sense that each idea created or generated is a potential solution to the problem and in general the idea is a matrix with the same length and represents the sequence and gaps [13] [11]. In the BSO-MSA algorithm, gaps are considered when building or generating an idea. The idea is represented as binary string using 0 and 1 only, which indicate the absence and presence of a gap respectively, and this representation is used during the mutation process to make mutations in the idea and build a new idea. For the rest of the operations, the idea is represented by knowing the location of the gap. Table 1 lists an example of the sequence and how it is encoded in the BSO-MSA algorithm.

- Population Initialization

The size of the population in any algorithm is an important factor in obtaining optimal results. This is because the BSO algorithm makes a modification to a limited number of ideas generated during one generation or at one time, and the solution is discovered on a small scale to reduce the time used to find the solution and avoid consuming a huge amount of time because of numerous ideas being generating. Therefore, we carefully choose the size of the population, accurate and time-saving users to find a perfect solution. The population size is represented by the number of ideas per generation. The alignment’s search space (N), is determining using the following formula:

$$N = n_{max} * (1 + r)$$

where n_{max} is the maximum length of the sequence and r is chosen according to the probability distribution [18]. The value of N is more critical for limiting the alignment’s maximum length, which is mostly used for chromosomal representation. If the value is too small, no optimum alignment will be obtained, and if it is too large, the process of obtaining a high-quality optimal alignment will take more time.

SPS calculated as the following equations:

$$SPS = \frac{\sum_{i=1}^{M} S_i}{\sum_{i=1}^{M_r} Sr_i}$$

$$whrer, S_i = \sum_{j=1, j\neg k}^{N} \sum_{k=1}^{N} P_{ijk}$$

$$where, P_{ijk} = \begin{cases} 1, & if\ A_{ij} and\ A_{ik}\ are\ aligned \\ 0, & others. \end{cases}$$

Where the number of the tested alignment donated by N, the number of columns for each alignment donated by M, for each ith column in the alignment donate A_{i1}, A_{i2}, A_{i3}, ..., A_{iN}. P_{ijk} indicates the similarity for each pair of residues A_{ij} and A_{ik} and S_i calculate the score for the ith column.

- Clustering Individuals and Disrupting Cluster Centers

Clustering is unsupervised learning. It is a strategy for categorizing data. The primary purpose of clustering algorithms is to divide large data into small groups of items that are similar and related. In the clustering analysis, there are two methods for determining similarity: as follows: finding an intercept between objects; and finding a correlation between objects. A second method of evaluating similarity in clustering is to compute or measure the distance between the items; and the third, is by calculating the distance. In many ways, the clustering process is like the brainstorming process in that it divides ideas into small groups of items that are similar. We decided to employ the K-mean clustering technique [12] some owing its efficiency and precision in computing. The clustering approach is demonstrated in the following procedure 3.1.

Procedure 3.1 Clustering technique

1. Let X = (x_1, x_2, . . ., x_n) be the set of data points and V = (v_1, v_2, . . ., v_c) be the set of centers
2. Randomly select 'c' cluster centers
3. Calculate the distance between each data point and cluster centers using k-means algorithm.
4. Assign the data point to the cluster center whose distance from the cluster center is the minimum of all cluster centers.
5. Recalculate the new cluster center using where, 'ci' represents the number of data points in the ith cluster.
6. Recalculate the distance between each data point and newly obtained cluster centers using the k-means algorithm.
7. If no data point was reassigned then stop, otherwise repeat from Step 3.

- Updating Individuals

We generate new individuals, through one or more group centers, or one or two individuals. To see which one to use, a random value is generated between the range

(0,1). Here we use some of the characteristics of the genetic algorithm, namely the crossover and the mutation The first method for generating individuals from a single cluster is the mutation method, and the following procedure shows how mutation is used to generate a new individual.

Procedure 3.2 Mutation operator

1. Convert the representation of the chromosome to a binary format.
2. The minimal ideal mutation rate is initialized, and the appropriate mutation point is chosen.
3.The genes within each complete point and if any gene appears between the last full point and the mutation point are shuffled independently.

Procedure 3.3 Crossover operator

1.Choose randomly one position point in the chromosome.
2.Take the head of chromosome parent 1 and the tail of the chromosome parent 2 to be child 1 and the head of chromosome parent 2 and the tail of the chromosome parent 1 to be child 2 (Figure 2).

We produce new individuals in the following two ways: through one or more cluster centers, or one or two individuals. To choose which value to use, a random value between (0,1) is produced. There are two methods for generating new individuals. The first is from a single cluster, as described in procedure 3.4:

Procedure 3.4 Updating individual

1. Generate random values in the (0,1) range.
2. If the value generated is less than the predetermined value, select one cluster center and update it by mutation following procedure 3.2
3. Else:

choose a random individual from the cluster group to update by generating a random chromosome.

The second section creates new individuals from two cluster centers or individuals. This approach is referred to as diversification. Procedure 3.5 gives in details the second novel method for individual generation:

Procedure 3.5 Updating individual

1. Generate random values in the (0,1) range.
2. If the value generated is less than the predetermined value, select two random cluster centers and combine them through the crossover process procedure 3.3.
3. Else, choose randomly an individual from two clusters to combine them through the crossover process.

- Selection

When executing the algorithm, the population size does not change; rather, it is fixed. In each cycle, a new human replaces the previous one. The replacement strategy is based on preserving the best by comparing the new and the old individuals in the same index and selecting the best. Finally, the Best-list is updated with the improved individuals.

RESULTS

The proposed approach was validated by using datasets from (BAliBASE), with total lengths of 1000 - 7000. The quantity of input sequence varied from 4 to 16. The resulting score of the final alignment was determined and organized.

1. Parameter Settings

   Some parameters have their values set to those documented in the literature. A preliminary numerical experiment is used to set the other parameters. Table 2 lists the values of the parameters.

2. Performance Analysis

   BSO-GA is programmed in MATLAB. The algorithm terminates when the number of iterations reaches the predetermined number GenMax. The BaliBASE database that contains the alignments strings separated and tested manually, which was specially designed to evaluate the efficiency of the alignment programs and compare them. Two versions of the database are available: mdsa 100s and mdsa all [2].

   - The mdsa 100s version contains the alignments of datasets that TBLASTN found 100% sequence identity for each sequence.
   - The mdsa all version includes all hits with an E-value score above the threshold of 0.001. The primary use of these databases is to benchmark the performance of MSA applications on DNA datasets.

The fitness function used to evaluate the validity and efficiency of multiple sequence alignment is SPS. It is the proportion of definite matches to the alignment score of the sequence. ClustalW, Dialign, Mafft, etc are the tools used for MSA.

The sequences that used to test the algorithm are taken from the BaliBASE database

- Dataset RV 11 BBS11022 from the mdsa all with 4 sequences.
- Dataset RV 11 BB11037 from the mdsa all with 8 sequences.
- Dataset RV 11 BB11002 from the mdsa all with 8 sequences.
- Dataset RV 11 BB11009 from the mdsa all with 4 sequences.

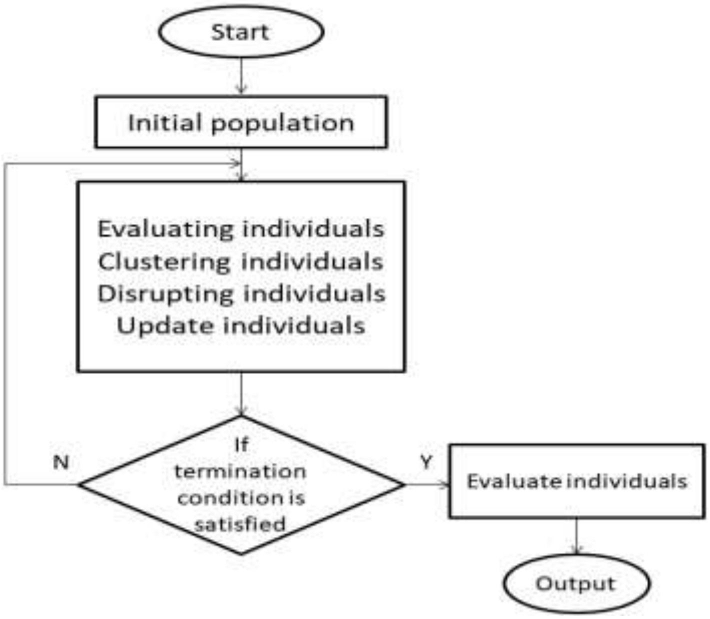

Figure 1: BrainStorm Optimization Flowchart(BSO)

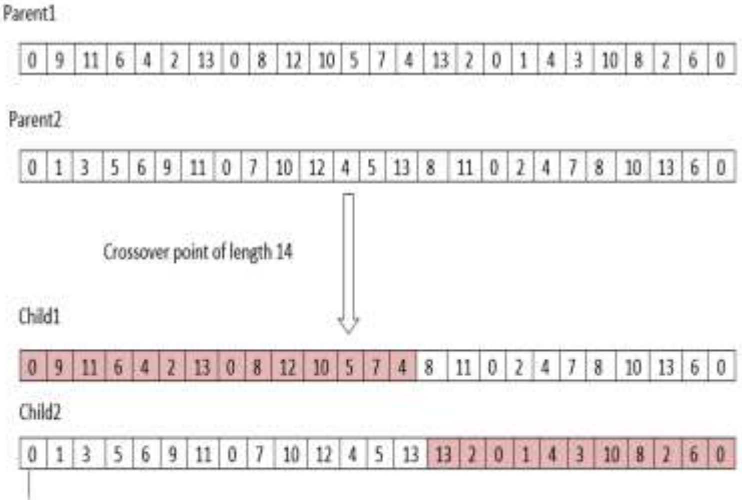

Figure 2: Crossover Flowchart

Table 1: Example for idea representation

Sequence	**_TCGAGT__ GCTTTACAGT_**
Gap position encoding:	0 7 8 19
Binary string encoding:	10000001100000000001

Table 2: BSO-MSA Parameters

Parameter Operator	Parameter Value	Description
cluster num	5	The number of k-means clusters
p_ replace	0.4	The probability of replacing the operator
p _one	0.4	The probability of selecting one cluster
p _one _center	0.3	The probability of selecting the center of one cluster
p _two _center	0.2	The probability of selecting the centers of two clusters
GenMax	5000	The number of generations

Table 3: Comparison of BSO-MSA with other MSA tools.

Dataset	BSO-GA	ClustalW	Dialign	Dialign	Mafft
RV 11 BBS11022	0.53	0.0801	0.0809	0.218	0.0402
RV 11 BB11037	0.015	0.0234	0	0.0038	0.0124
RV 11 BB11002	0.0040	0.0046	0	0.0096	0
RV 11 BB11009	0.035	0.0203	0	0.803	0.0188

DISCUSSION

The dataset was tested using BSO-MSA and compared with the Dialign,Mafft,ClustalW and Multalin tools. The results are listed in Table 3.

The RV11 dataset used consists of sequences with less than 25% identity. As shown in Table 3, there is no similarity among RV 11 BB11037 , RV 11 BB11002 and RV 11 BB11009 using Multalin. Similarly, Mafft could not generate any exact match in the RV 11 BB11002 dataset. BSO-MSA generated scores better than those of Dialign and Mafft in RV 11 BBS11022 and RV 11 BB11037 datasets and ClustalW and Mafft in the RV 11 BB11009 dataset.

CONCLUSION

BSO is a swarm intelligence system currently in its early phase. Swarm intelligence and data mining technologies are combined in the BSO algorithm. Every individual involved in the issue-solving process is not only a solution to the current problem, but also a source of inspiration for future solutions.

In this study, we developed a novel BSO–MSA method to investigate the MSA problems and improve BSOMSA performance. The parameters adopt optimal values during execution, unlike BSO-MSA.

This technique assists non-domain users in avoiding the usage of preset parameter values, which may not be acceptable in all situations. The designed crossover operator encourages the gene to increase exploration to improve working efficiency and investigate the impact of crossover on population diversity. Saving a copy of the best chromosome after each genetic surgery with an elitist selection ensures that the best chromosome is not disrupted. To tackle MSA, BSO-MSA was designed and applied, and it performed better in terms of avoiding premature convergence. The effectiveness and consistency of BSO-MSA in achieving optimum alignment were demonstrated by comparing its performance with those of other frequently used MSA tools.

REFERENCES

[1] Barton, G. J., and Sternberg, M. J. A strategy for the rapid multiple alignment of protein sequences:confidence levels from tertiary structure comparisons. Journal of molecular biology 198, 2 (1987), 327–337.

[2] Carroll, H., Beckstead, W., O'Connor, T., Ebbert, M., Clement, M., Snell, Q., and Mc-Clellan, D. Dna reference alignment benchmarks based on tertiary structure of encoded proteins. Bioinformatics 23, 19 (2007), 2648–2649.

[3] Corpet, F. Multiple sequence alignment with hierarchical clustering. Nucleic acids research 16, 22 (1988), 10881–10890.

[4] Doak, C. K., Jambura, S. M., Knittel, J. A., and Rule, A. C. Analyzing the creative problemsolving process: Inventing a product from a given recyclable item. Creative Education 4, 9 (2013), 592.

[5] Gupta, S. K., Kececioglu, J. D., and Sch¨affer, A. A. Improving the practical space and time efficiency of the shortest-paths approach to sum-of-pairs multiple sequence alignment. Journal of ComputationalBiology 2, 3 (1995), 459–472.

[6] Haubold, B., and Wiehe, T. Introduction to computational biology: an evolutionary approach. Springer Science & Business Media, 2006.

[7] Holland, J. H. Adaptation and artificial systems: An introductory analysis with application to biology, control, and artificial intelligence. USA: University of Michigan (1975).

[8] Kaur, Y., and Sohi, N. Comparison of different sequence alignment methods-a survey. International Journal of Advanced Research in Computer Science 8, 5 (2017).

[9] Kubota, N., Fukuda, T., and Shimojima, K. Virus-evolutionary genetic algorithm for a self-organizing manufacturing system. Computers & industrial engineering 30, 4 (1996), 1015–1026.

[10] Lawrence, C. Altschul. SF, Bogusky, MS, Liu, JS, Neuwald., AF, and Wootton, JC (1993), 208–214.

[11] Liu, D., Xiong, X., Hou, Z.-G., and DasGupta, B. Identification of motifs with insertions and deletions in protein sequences using self-organizing neural networks. Neural Networks 18, 5-6 (2005), 835–842.

[12] MacQueen, J., et al. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability (1967), vol. 1, Oakland, CA, USA, pp. 281–297.

[13] Nizam, A., Ravi, J., and Subburaya, K. Cyclic genetic algorithm for multiple sequence alignment. International Journal of Research and Reviews in Electrical and Computer Engineering (IJRRECE) Vol 1(20).

[14] Shi, Y. An optimization algorithm based on brainstorming process. In Emerging Research on Swarm Intelligence and Algorithm Optimization. IGI Global, 2015, pp. 1–35.

[15] Smith, T. F., and Waterman, M. S. Comparison of biosequences. Advances in applied mathematics 2,4 (1981), 482–489.

[16] Thompson, J. D., Higgins, D. G., and Gibson, T. J. 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, 22 (1994), 4673–4680.

[17] Van Laarhoven, P. J., and Aarts, E. H. Simulated annealing. In Simulated annealing: Theory and applications. Springer, 1987, pp. 7–15.

[18] Wu, S., Lee, M., Lee, Y., and Gatton, T. M. Multiple sequence alignment using ga and nn. International Journal of Signal Processing, Image Processing and Pattern Recognition 1, 1 (2008), 21–30.