Differential evolution-simulated annealing for multiple sequence alignment

R C Addawe¹, J M Addawe¹, M R K Sueño², and J C Magadia³

¹ Department of Mathematics and Computer Science
University of the Philippines Baguio, Philippines
² College of Science, University of the Philippines Baguio, Philippines
³ School of Statistics, University of the Philippines Diliman, Philippines
E-mail: {rcaddawe, jmaddawe, mksueno, jcmagadia}@up.edu.ph

Abstract. Multiple sequence alignments (MSA) are used in the analysis of molecular evolution and sequence structure relationships. In this paper, a hybrid algorithm, Differential Evolution - Simulated Annealing (DESA) is applied in optimizing multiple sequence alignments (MSAs) based on structural information, non-gaps percentage and totally conserved columns. DESA is a robust algorithm characterized by self-organization, mutation, crossover, and SA-like selection scheme of the strategy parameters. Here, the MSA problem is treated as a multi-objective optimization problem of the hybrid evolutionary algorithm, DESA. Thus, we name the algorithm as DESA-MSA. Simulated sequences and alignments were generated to evaluate the accuracy and efficiency of DESA-MSA using different indel sizes, sequence lengths, deletion rates and insertion rates. The proposed hybrid algorithm obtained acceptable solutions particularly for the MSA problem evaluated based on the three objectives.

1. Introduction

Multiple sequence alignments (MSAs) are strategies in molecular biology for homology transfer, phylogenetic analyses, structural modeling, functional predictions and protein structure and interaction predictions[1, 2, 3]. The development novel experimental techniques such as the next-generation sequencing and high-throughput experiments motivated the need for MSA tools that extract biological meanings from the massive information generated by novel experimental techniques such as the next-generation sequencing and high-throughput experiments. Efficient MSA tools are capable of dealing massive information by using advanced computational approaches based on artificial intelligence and machine learning algorithms. Differential Evolution (DE) [4, 5], developed by Ken Price in 1994, is a simple population-based stochastic function minimizer which is very powerful. It basically adds the weighted difference between two population vectors to a third vector and uses a way to separate probability distributions to make a completely self-organizing algorithm. The development of DE and its improvements included hybridization of DE and its applications [6]. Simulated annealing (SA) [7, 8], on the other hand, is a probabilistic method for finding the global minimum among several local minima of an objective function. SA is based on an analogy with a physical phenomenon of the annealing process of metals [9].

In this paper, we applied the DESA algorithm [10] with the MSA problem based on structural information [11], non-gaps percentage and totally conserved columns. In the next section we
briefly discuss the algorithms and multiple sequence alignment problem. The next sections present the results and conclusions.

2. Related Methodologies

In this section, DE, SA, DESA, the MSA problem, and the objective functions are summarized as the foundation to construct the proposed hybrid algorithm to solve the multiple sequence alignment problem.

2.1. Differential Evolution (DE)

Differential Evolution (DE) [4] is a robust global optimization tool for NP-hard and complex optimization problems. It has been developed with several working variants. In the DE/rand/1/bin variant, we minimize the objective function, \( f(X) : \mathbb{R}^n \rightarrow \mathbb{R} \) by optimizing the values of \( X = (x_1, \ldots, x_D) : X \in \mathbb{R}^D \), where \( X \) is a vector of \( D \) parameters subject to lower and upper constraints, \( x^{(L)} \) and \( x^{(U)} \), such that \( x_j^{(L)} \leq x_j \leq x_j^{(U)} \), where \( j = 1, 2, \ldots, D \). A potential solution under population \( i \) and generation \( G \) is denoted by \( X_{i,G} \), such that, \( P_{G} = x_{i,G} = x_{i,j,G} \), where \( i = 1, 2, \ldots, NP \). For a starting solution, let \( P_G = 0 : P_0 = X_{i,j,0} = r_{i,j}(x_j^{(U)} - x_j^{(L)}) + x_j^{(L)} \) where \( r \) is a uniformly distributed random value in \([0,1]\) for each \( j \). The next population \( P_{G+1} \) is created from the current population \( P_G \). A trial population for the next generation, \( P'_{G+1} \), is \( x'_{i,j,G+1} = x_{C_{i},j,G} + F(x_{A_{i},j,G} - x_{B_{i},j,G}) \) if \( (r_{i,j} \leq CR) \lor (j = D_i) \); \( x_{i,j,G} \) otherwise. \( A_i \neq B_i \neq C_i \neq i = 1, 2, \ldots, NP \), \( CR \in [0,1] \), \( F \in [0,2] \), and \( r \in [0,1] \) is a uniformly distributed random value. Then new random values of \( A, B, C \) are assigned for each \( i \) and a new random number \( r \) is assigned for each \( j \). \( D \) is a randomly chosen chromosome of each vector \( X_{G+1} \) different from each counterpart in the previous generation, \( X_G \). A new random integer is assigned to \( D \) for \( i \). Hence, from the initial population \( P_G \) and the temporary population \( P'_{G+1} \), the next generation \( P_{G+1} \) is \( X_{i,G+1} = X'_{i,G} \) if \( f_{\text{cost}}(X'_{i,G+1}) \leq f_{\text{cost}}(X_{i,G}) \); \( X_{i,G} \) otherwise.

2.2. Simulated Annealing (SA)

Simulated Annealing (SA) [8, 9] starts with a set of random values of the parameters and a temperature value \( T \). For each iteration, this set of parameters is randomly varied proportional to \( T \), and the objective function, \( f(X) \), is calculated. The change in \( f(X) \) may be denoted by \( \Delta E \). If the set of parameters results in a better value of \( f(X) \), the parameters are retained and forms the basis of the next variation of the potential parameter values. SA allows a deterioration in the objective function with a known probability, given by Boltzmann distribution: \( P(\Delta E) = e^{(-\frac{\Delta E}{T})} \), where \( P(\Delta E) \) is a probability distribution giving the probability of making an energy increasing step having magnitude \( \Delta E \). This can be done by drawing a random number in \([0,1]\) and checking if it is less than \( P(\Delta E) \); returning true, if so. Then, the value of \( T \) is decreased by an acceptably small amount and the whole set of iterations is repeated. SA surely obtains the global minimum \( f(X) \). SA surely obtains the global minimum of \( f(X) \), after a number of iterations and when \( T \) is sufficiently small.

2.3. Differential Evolution - Simulated Annealing (DESA)

Differential Evolution - Simulated Annealing (DESA) [10, 12] is characterized by self-organization, mutation, crossover, and SA-like selection scheme of the strategy parameters. From the current population \( P_G \) and the trial population \( P'_{G+1} \), the population of the next generation \( P_{G+1} \) is \( P_{i,G+1} = P'_{i,G} \) if \( R(\Omega_i'_{G+1}) \leq R(\Omega_i,G) \) or \( e^{-[\frac{R(\Omega_i'_{G+1}) - R(\Omega_i,G)}{T_{\text{redact}}}] > \text{rand}[0,1]} \); \( P_{i,G} \) otherwise, where \( \Omega \) is the parameter vector and \( R \) is the fitness function. In this SA-like selection scheme of DESA, a trial vector is compared only against one individual in the current population; and
that besides accepting improvements in cost, it also accepts deteriorations in cost of the objective function.

2.4. The Multiple Sequence Alignment (MSA) Problem

In a multiple sequence alignment, the idea is to maximize the number of similar residues in a given column of the alignment. An alignment can be easy or difficult depending on the degree of relationship among the sequences. More related sequences are easier to align than less related ones since insertions and deletions of gaps are greatly needed in aligning these kinds of sequence [13]. We use DESA to compute sequence alignment using the structural information as the scoring scheme. To test the performance of DESA-MSA, numerical simulations were carried out on data sets obtained from benchmark alignment data (BAliBASE) in deciding the appropriate parameters for the proposed algorithm DESA.

Given an MSA $A$, the score of $A$ is the sum of the scores of each target sequence $i$ when they are aligned with a sequence of known structure called the template. The score of all the target sequences, $\text{Score} (\text{Target})$, normalized by the total number of contacts $|c|$ within the template is given by:

$$\text{Score} (\text{Target}) = \frac{\sum_{ij} \text{STRIKE}(c_i c_j) \times IsContact(c_i c_j)}{|c|} \tag{1}$$

with

$$IsContact(c_i c_j) = \begin{cases} 1, & \text{if } c_i c_j \text{ are in contact} \\ 0, & \text{else} \end{cases}$$

The STRIKE score is denoted by $\text{STRIKE}(c_i c_j)$ which is the total score of the contact between $c_i$ and $c_j$ given by the STRIKE matrix. Therefore, the score of MSA $A$, where $N$ is the number of sequences is given by:

$$\text{Score}(A) = \sum_{i}^N \text{Score(Target)} \tag{2}$$

The Non-Gaps Percentage (NGP) $g$ is the number of gaps in the alignment, $aa$ is the number of amino acids in the alignment; such that the total number of gaps and amino acids in the alignment equal to $g + aa$. The Percentage of Totally Conserved Columns, is denoted as $TCP = \frac{c}{l} \times 100\%$, where $c$ is the number of columns aligned with exactly the same amino acid and $l$ is the length of the matrix.

3. Results

To find the appropriate parameters for DESA-MSA, two sets of experiments were done on five datasets obtained from BAliBASE. First, we optimized the three objective functions on STRIKE, NGP and TCP scores. Next, we maximize the STRIKE scores only to improve structural similarities between the aligned sequences. In addition, different combinations of the crossover probabilities, $P_c$ and mutation probabilities, $P_m$, were investigated. The alignment with the highest STRIKE score, which gives an alignment of more quality according to the sequence structures is considered as the best alignment. The average STRIKE scores, NGP scores, TCP scores, and computer runtimes are shown in Fig. 1 to 4. A graphical comparison between the average STRIKE scores of the best alignments obtained from the two sets of simulations using different combinations of $P_c$ and $P_m$ is shown in Fig. 1.

The 30%-70% combination, which obtained the highest average STRIKE score of 2.2011, is just about 2% higher than the worst average STRIKE score obtained from the 20%-80% combination which is 2.1569. However, the scores from the different $P_c - P_m$ combination are not significantly different from the average STRIKE score of 2.1835, with standard deviation of
0.152. This imply that the STRIKE score cannot be improved further by the different combination of $P_c - P_m$. On the other hand, the STRIKE scores obtained by optimizing the three objective functions on SRIKE, NGP and TCP scores had an average of 1.6224 with standard deviation 0.9238. These STRIKE scores are significantly lower than the average score of 2.1835 obtained by maximizing the SRIKE score only. However, maximizing the three objectives and using $P_c - P_m$ of 90%-10%, 80%-20%, 70%-30%, or 60%-40%, obtained significantly higher STRIKE scores with 80%–20% combination being best. Thus, SRIKE scores may be improved considering these $P_c - P_m$ combinations. However, including TCP and NGP scores with the STRIKE scores in maximizing alignment resulted to about 25% decreased in the STRIKE scores.

For the average percentage of non-gaps (NGP) obtained from the two sets of simulations presented in Fig. 2, the 80%-20% combination yielded the highest which is 63.9912% followed by the 90%-10% combination. The NGP scores with a mean of 62.9820 and standard deviation of 0.5654 obtained by maximizing NGP, TCP and STRIKE scores are significantly higher than the NGP scores with mean of 61.1745 with a standard deviation of 0.9238, obtained by maximizing STRIKE score only. Thus, higher NGP scores may be obtained when using either 90%-10% or 80%-20% combination in optimizing the function with the NGP, TCP and STRIKE scores.
two sets of simulations are also presented in Fig. 3. The 40%-60% and 10%-90% combinations gave the highest average percentages with the 40%-60% combination yielding better average TCP equal to 4.4086% which only differs by 0.0248% from the 10% - 90% combination. These TCP scores using 40%-60% and 10% - 90% $P_c - P_m$ combinations are significantly higher than the rest of the combinations used.

On the average, TCP scores obtained by maximizing the three objectives with an average of 3.55446% and a standard deviation of 0.7833, is significantly higher than the TCP scores obtained by maximizing the STRIKE score only having an average of 0.9508% and a standard deviation of 0.0745. The average difference of 2.5937 on the TCP scores contributed to an improvement of about 273% on the TCP scores when the three objective functions were maximized. Hence, the optimizing the three objectives using 40%-60% or 10% - 90% combinations yielded an improved TCP scores.

For the runtime, data are presented in Fig. 4 where simulations using the 90%-10% combination had the shortest average runtime of 0.7085 minutes.

This runtime average is significantly lower than the average runtimes obtained by the rest of the $P_c - P_m$ combinations with mean of 0.7776 minutes and a standard deviation of 0.0363 minutes. On the other hand, optimizing the three objectives obtained significantly longer average runtimes. In addition, simulations maximizing only the STRIKE score are twice faster than the
simulations maximizing the three objectives. Nevertheless, in maximizing the three objectives, simulations using the 20%-80% combination had the shortest average runtime of 1.0551 minutes followed by the 10% - 90% combination with an average runtime of 1.0483 minutes, with 20%-80% combination being the fastest. These two runtime averages obtained from 10%-90% and 20% - 80% are significantly lower than the average runtimes obtained by the rest of the \( P_c - P_m \) combinations with mean of 1.992 minutes and a standard deviation of 0.1285. In addition, simulations maximizing only the STRIKE score are twice faster than the simulations maximizing the three objectives.

Results of the simulations obtained from DESA-MSA using multi-objective functions have shown that NGP and TCP scores may be improved by using different \( P_c - P_m \) combinations. However, as the NGP and TCP scores the STRIKE score may be decreased and the average runtime is increased. Higher NGP scores may be obtained when using either 90%-10% or 80%-20% combination in optimizing the function with the NGP, TCP and STRIKE scores. The TCP scores were improved up to 273% when using \( P_c - P_m \) combination of either 40%-60% or 10% - 90%. As expected, simulations in maximizing the three objectives, yielded longer runtime than the simulations using STRIKE score only. Based on the data presented, the 80%-20% combination ranked first in the average NGP score computations but obtained fairly low TCP scores. The 40%-60% combination ranked first in the average TCP score computations and third in the NGP score computations, respectively. In particular, the 40%-60% combination is more appropriate than the 80%-20%. Results show that the experiments on maximizing the STRIKE, NGP and TCP together with the 40%-60% combination obtained higher NGP and TCP values; but lower STRIKE score and longer runtime than maximizing only the STRIKE score. Though the simulations in maximizing the three objectives obtained longer runtime of about 52% longer runtime and lower STRIKE score of about 32%, it provided an NGP and TCP scores better by 3% and 273%, respectively. The 80%-20% combination is an alternative, yielding an average of about 5.5% higher NGP score, 133% higher TCP score, 32% lower STRIKE score and 75% longer runtime.

4. Conclusions and Recommendations
In this study, we proposed Differential Evolution - Simulated Annealing (DESA), a hybrid algorithm for optimizing multiple sequence alignments (MSAs) based on three objectives: (1) structural information (2) non-gaps percentage and (3) totally conserve columns. DESA algorithm obtained acceptable and even better solutions particularly for the MSA problem evaluated based on the three objectives. DESA-MSA, a hybrid algorithm for multiple sequence alignment has been shown to be efficient in achieving optimized alignments. This paper aimed to find the best parameter by testing varying combinations of crossover and mutation probabilities. Numerical experiments were performed on five datasets obtained from the Benchmark Alignment DATABASE (BAliBASE) to decide the appropriate parameters for DESA-MSA. Results obtained from the simulations have showed that the combination 40% crossover and 60% mutation gave alignments with the best qualities in terms of sequence structures and evolutionary homologies. Thus, in this study, the best combination of parameters is the 40% crossover and 60% mutation probabilities.

Acknowledgments
The author RCA would like to thank Japan Mathematical Society and the University of the Philippines for the conference financial support.

References
[1] Doolittle R F 1981 Science 214 149
[2] Fitch W 1966 J. Mol. Biol. 16 9
[3] Bacon D and Anderson W 1986 *J. Mol. Biol.* **191** 153
[4] Storn R and Price K 1997 *J. Global Optim.* **11** 341
[5] Storn R and Price K 1997 *Dr. Dobb’s Journal* **22** 18
[6] Das S and Suganthan P N 2011 *IEEE Transactions on Evolutionary Computation* **15** 4
[7] Kirkpatrick S, Gelatt C D and Vecchi M P 1983 *Science* **220** 671
[8] Cerny V 1985 *J. Optim. Theory Appl.* **45** 41
[9] Bertsimas D and Tsitsiklis J 1993 *Statistical Science* **8** 10
[10] Addawe R, Adorio E, Addawe J and Magadia J 2006 *DESA: A hybrid optimization algorithm for high dimensional functions, in Proc of the 8th IASTED Int Conf on Control and Applications (ACTA Press)* pp 316–321
[11] Sueno M and Addawe J 2016 *Advanced Studies in Biology* **8** 9
[12] Addawe R and Magadia J 2014 *Differential Evolution-Simulated Annealing (DESA) algorithm for fitting autoregressive models to data, Proc of OPT-i2014 - Int Conf on Engineering and Applied Sciences Optimization* vol eds. M. Papadrakakis et al pp 434–451
[13] Florido J, Ortuno F, Pomares H, Rojas F, Urguisia J and Valenzuela O 2013 *BMC Bioinformatics* **29** 2112