A Hybrid-Based Harmony Search Algorithm for RNA Multiple Sequence Alignment

Mubarak S. Mohsen, Rosni Abdullah, Mohd. Adib Omar

School of Computer Sciences, Universiti Sains Malaysia, Penang, 11800, Malaysia
mobarak_seif@yahoo.com, rosni@cs.usm.my, adib@cs.usm.my

Abstract: Multiple sequence alignment (MSA) is an important topic in the field of computational molecular biology. Scalability, computational complexity as well as biological accuracy, are currently required for MSA methods. Finding an accurate alignment from primary sequences is computationally NP-hard problem. The aim of this study is to investigate and explore the capability of meta-heuristic technique known as harmony search algorithm (HS) to address MSA problem. This study introduces BHS-MSA, MHS-MSA, and HHS-MSA methods based on harmony search algorithm. Using a well-known benchmark (BRAliBase), the proposed method HHS-MSA showed better results compare to (BHS-MSA and MHS-MSA). HHS-MSA produced comparative results in terms of quality (Q) and total column (TC) and achieved good results in certain test groups especially alignments with large sequences.

[Mubarak. S. M., Rosni A., Adib O. **A Hybrid-Based Harmony Search Algorithm for RNA Multiple Sequence Alignment.** *Life Sci J* 2018;15(11):19-31]. ISSN: 1097-8135 (Print) / ISSN: 2372-613X (Online). http://www.lifesciencesite.com. 4. doi:10.7537/marslsj151118.04.

Keywords: RNA, Harmony search algorithm, Multiple sequence alignment, DCA, Conserved Blocks.

1. Introduction

Multiple sequence alignment (MSA) has become widely used in many different areas in bioinformatics such as biological sequence analysis, gene regulation and polymerase chain reaction primer design (Anbarasu et al., 2000). Moreover, MSA has significantly improved the accuracy of RNAs structures prediction (Hickson et al., 2000). For example, current RNA secondary structure prediction methods which use aligned sequences are more successful in gaining higher prediction accuracy than single sequence (Bernhart et al., 2008). Identifying the secondary structure of RNA molecules is the fundamental key to understand its biological function (Tsang and Wiese, 2007).

Finding an accurate MSA from sequences is a challenging task. It is a time consuming and computationally NP-hard problem (Bonizzoni and Della Vedova, 2001) which motivates the researchers for heuristics (Thompson et al., 1994). The MSA problem can be considered as an optimization problem which aims to maximize or minimize the scoring function (Xu and Chen, 2009) by improving the quality of initial alignment (Zhang and Kahveci, 2007).

Over the last decade, the evolutionary and meta-heuristic are the recent approaches to solve the optimization problem. Most of MSA methods are based on heuristics to produce near optimal alignment with high accurate within moderate computational time. In this context, many researches address MSA problem as optimization problem by using Genetic Algorithm (Wang and Lefkowitz, 2005; Notredame and Higgins, 1996), Particle Swarm (Xu and Chen, 2009), Ant Colony (Liu et al., 2007), and Simulated Annealing (Roc, 2007).

Recently, the number of discovered biological functions of RNA has increased. In addition, the RNA function scope has expanded, and thus RNA is not a passive messenger of genetic information from DNA to protein manufacturer as had been thought before. It has been found that RNA plays important roles in all molecular biology such as carrying genetic information (mRNA), interpreting the code (ribosomal RNA), and transferring genetic code (tRNA). It also performs different functions which include: catalyzing chemical reactions (Hansen et al., 2002), directing the site specific modification of RNA nucleotides and controlling gene expression.

Harmony Search (HS) algorithm (Geem et al., 2001) is a new meta-heuristic algorithm for solving optimization problem. HS is deemed one of the most efficient alternatives to seek and determine a near-optimal solution without relying on exact yet computationally demanding algorithms. HS algorithm has shown to achieve good result in a wide range of optimization problems. HS algorithm has innovative aspects features in its operational procedure that foster its utilization in diverse fields such as construction, engineering, robotics, health and energy (Manjarres et al., 2013). Specifically, HS has adapted to address bioinformatics problems such as Mohsen et al. (Mohsen et al., 2010) for RNA secondary structure prediction and Abual-Rub et al. (Abual-Rub et al., 2012) for protein tertiary structure prediction. Based on the HS success in several scientific and engineering optimization problems, HS might possess the potential to be used for the MSA.

The main objectives of this paper are: (1) to adapt the harmony search algorithm for MSA problem (henceforth called basic harmony search algorithm for MSA, BHS-MSA), (2) to improve the harmony search algorithm by modifying the harmony search improvisation stage (henceforth called modified harmony search algorithm, MHS-MSA), (3) to investigate the effectiveness of hybridization of the harmony search algorithm with divide-and-conquer approach (henceforth called hybrid harmony search algorithm, HHS-MSA).

This paper is organized as follows: Section 2 reviews the current MSA methods. Section 3 explains the research design and methodology used in this research. Section 4 discusses the experimental results and comparative analysis to assess the alignment quality. Lastly, the conclusion of the paper and future work are provided in section 5.

2. Review of Current MSA

Over the past years, researchers have developed various multiple sequence alignment methods. These methods differ in sequence orders, procedures and sequences score. Existing methods can be classified into the following approaches (Wallace et al., 2005; Edgar and Batzoglou, 2006; Mohsen and Abdullah, 2011): exact, progressive, iterative, block-based, consistency-based, probabilistic, computational intelligence, and heuristic.

This section provides a brief overview of meta-heuristic optimization approaches. Operating several solutions simultaneously is the advantage of Evolutionary Algorithm (EA) and meta-heuristic. Moreover, Meta-heuristic combining exploratory search through the solution space with exploitation of current results (Bi, 2008). There are no restrictions on the number of sequences or their length. Hence, evolutionary algorithm is very flexible in the optimization with low complexity.

Different GA methods are suggested to address the MSA problem (Notredame and Higgins, 1996; Isokawa et al., 1996; Zhang and Wong, 1997a; Jorng-Tzong Horng et al., 2000; da Silva et al., 2008 ; Gondro and Kinghorn, 2007; Lai et al., 2009). For example, Sequence Alignment by Genetic Algorithm (SAGA) (Notredame and Higgins, 1996) is the earliest GA method applied to tackle the MSA.

Furthermore, hybrid methods of GA are also used to address the MSA problem. For instance, Zhang and Wong (1997b) present a pairwise dynamic programming (DP) technique based on GA. Similarly, utilization of GA in progressive-based approach has developed by (Cai et al., 2000). Later, Wang and Lefkowitz (2005) produce Gen Align Refine algorithm, which uses a genetic algorithm to improve the local regions alignment leading to the enhancement the overall quality of global multiple alignments. Meanwhile, Ergezer and Leblebicioglu (2006) use GA as iterative approach to refine the alignment score obtained by progressive method. Using the GA to locate the cut-point in divide-and-conquer approach has developed by (Chen et al., 2005). With the similar combination, the genetic algorithm is incorporated with ant colony optimization which called GA-ACO (Lee et al., 2008). Likewise, Taheri and Zomaya (2009) have introduced the RBT-GA which is a combination between the Rubber Band Technique (RBT) and the Genetic Algorithm (GA) meta-heuristic. An interesting attempt is the proposed of PASA algorithm (Jeevitesh. M.S et al., 2010) which uses the GA model to combine the alignment outputs from two MSA programs (MCoffee and ProbCons).

Furthermore, Ant colony optimization (ACO) is used to address the MSA. Zhao (Zhao et al., 2008) developed an improved ant colony algorithm yet is more complicated. Likewise, Particle swarm optimization (PSO) is employed widely in addressing the MSA. For instance, Rasmussen and Krink (2003) used a combination between PSO with evolutionary algorithms to train the hidden Markov model (HMMs). Similarly, Pedro et al. (2007) presented a modified algorithm based on PSO to improve the sequence alignment previously generated by Clustal X. Likewise, Juang and Su (2008) produce an algorithm which combines the pairwise DP and the PSO techniques to overcome the local optimum problems. Based on the idea of chaos optimization, Lei et al. (2009) have produced chaotic PSO (CPSO). In the similar context, a methods of mutation-based binary particle swarm optimization (M-BPSO) is designed by (Hai-Xia et al., 2009).

Similarly, simulated annealing (SA) algorithms have suggested tackling the MSA from different aspects. For example, Kim et al. (1994) have developed the multiple sequence alignment simulated annealing (MSASA) method. In another example, Uren et al. (2007) presented the MAUSA which use the simulated annealing to perform searching through the space of possible guide trees. Furthermore, SA algorithm is modified and combined in various aspects including finding a consensus sequence (Keith et al., 2002), selecting the cut-points (Roc, 2007), combining with GA algorithm (Omar et al., 2005), and combining with SA, GA and Monte Carlo methods(Joo et al., 2008). In the same way, Tabu Search (TS) has applied to tackle the MSA by implementing adaptive memory features to refine the MSA (Riaz et al., 2005) and exploring the iterative refinement techniques such as a hidden Markov model and an intensification heuristic (Lightner, 2008).

Although there are many attempts to apply the meta-heuristic in bioinformatics. Available meta-heuristic methods can not solve the MSA problem completely and professionally. Harmony Search (HS) algorithm (Geem et al., 2001) is a new meta-heuristic algorithm for solving optimization problem. HS is deemed one of the most efficient alternatives to seek and determine a near-optimal solution without relying on exact yet computationally demanding algorithms. HS algorithm has shown to achieve good result in a wide range of optimization problems. Specifically, HS has adapted to address bioinformatics problems such as Mohsen et al. (Mohsen et al., 2010) for RNA secondary structure prediction and Abual-Rub et al. (Abual-Rub et al., 2012) for protein tertiary structure prediction. Based on the HS success in several scientific and engineering optimization problems, HS might possess the potential to be used for the MSA.

3. Material and Methods

This section explains the MSA problem, MSA problem modeling, RNA dataset, evaluation design, and the proposed methods. It contains details of problem representation, and objective function. BHS-MSA, MHS-MSA, HHS-MSA method are proposed and explained.

3.1 Problem Description

MSA is the natural way to see the relation between sequences by making alignment between the primary sequences. Alignment is a method to arrange the sequences one over the other in a way to show the matching and mismatching between residues. To improve the alignment score, the character "–" is used to correspond to a space introduced in the sequence. This space is usually called a gap. The gap is viewed as insertion in one sequence and deletion in the other. A score value is used to measure the alignment performance. For clarity's sake, the generic MSA problem is expressed with the following declaration: "*Insert gaps within a given set of sequences in order to maximize a similarity criterion*" (Zablocki, 2007). The MSA problem can be divided into three challenges, which are scalability, optimization, and objective function. In fact, the complexity comes from that all three problems must be solved simultaneously. The first challenge is the scalability, which is to find the alignment of many long sequences. The second challenge is the optimization, which is to find the alignment with the highest score based on a given objective function among sequences. Optimization of even a simple objective function is an NP-hard problem. The third challenge is the objective function (OF), which is to speed up the calculation in order to measure the alignment.

3.2 Problem Modeling

Most optimization programs that produce multiple sequence alignments contain two parts: an objective function for score the alignment, and an optimization procedure for improving the alignment with respect to the chosen objective function (Notredame, 2002). The MSA problem representation and mathematical model with the objective function are presented as follows:

3.2.1 Problem Representation

Alignment of N sequences with different lengths from L_1 to L_N, are represented as a matrix of N x M where each row contains gap positions. The maximum number of columns in the matrix is $M = [\alpha L_{max}]$, where $L_{max} = max\{L_1, L_2, \dots, L_N\}$, and $[x]$ is the smallest integer greater than or equal to x and the parameter α is the scaling factor (Chellapilla and Fogel, 1999). The value of α may vary according to the alignment sequence length. It is widely recommended that increasing gaps number affects the alignment quality; hence, MSA tries to maximize the number of matching characters and minimize the number of inserted gaps (Zablocki, 2007). In the present study, the value of α is assigned 1 which allows alignment to be as long as the longest sequence. Thus, only necessary gaps are added.

3.2.2 The Mathematical Model

In order to optimize MSA, there are three basic components (Tsang and Wiese, 2007): (i) an objective function (or score function), which is the object of minimization or maximization; (ii) a set of variables which affect the value of the objective function; (iii) a set of constraints that allow the variables to take on certain values while excluding others. MSA can be formulated mathematically(Lei et al., 2009; Xu and Lei, 2010) as follows: Given group of sequences $s = (s_1, s_2, \dots, s_N)$ where $s_i = s_{il1}, s_{il2}, \dots\dots, s_i l_i$, $1 \le i \le N$, $s_{ij} \in \sum [A,C,G,U]$, $1 \le j \le l_i$, l_j is equal to the length of *i-th* sequence. Alignment of S is defined as a matrix $A = (a_{ij})$, where $1 \le i \le N$, $1 \le j \le l$, where i is the row, j is the column positions, and l is the length of matrix (number of columns). The characteristics of matrix A are subjected to the following: (i) $a_{ij} \in \sum \cup \{-\}$ where the symbol $\{-\}$ denotes a gap. (ii) The i^{th} sequence of A is identical with the i^{th} of S when the gaps are deleted; (ii) There is no row or column formed by only $\{-\}$ in matrix A.

Thus, the MSA optimization problem is $P = (S, f)$, where S (also called search space) is a finite set of solutions and the f *is* the score function. The score function maps the solution S to the real numbers value R, $f: S \rightarrow R$. It assigns a cost value to each solution. Thus, the goal is either to find a good enough solution in reasonable amount of time(Dorigo and Blum, 2005). The objective function is formulated as the following form:

Optimize f (x),

Subject to x_{ij} X_i, $1 \le i \le N$, $1 \le j \le 1$ (1)

where $f(x)$ is an objective function, $\boldsymbol{x}$ is a solution or harmony composed of each decision variable x_{ij} (each sequence), X_i is a set of candidate values (gap position) for each decision variable x_i, which becomes $X_i = \{ x_i(1), x_i(2), \ldots, x_i(K)\}$, $x_i(1) < x_i(2) < \ldots < x_i(K)$ for discrete decision variables; i is the number of decision variables (sequence number) and K is the number of candidate values for the discrete decision variables (number of gap position for each sequence).

3.2.3 Objective Function

Sum of pair (SP) is the standard method of scoring the MSA (Trystram and Zola, 2007) which is introduced by (Carrillo and Lipman, 1988). In the SP-score, it is assumed that the columns of the alignment matrix are statistically independent by ignoring the phylogenetic tree. The SP objective function defines the scores of n sequences by the sum of the scores of $n(n-1)/2$ pairwise alignments. Currently, SP objective function is the most widely used (Nicholas et al., 2002). Generally, the SP is used without any prior knowledge of the reference alignment and it does not really provide any biological or probabilistic justification (Durbin, 1998; Shyu and Foster, 2003). The general form of objective function for the alignment consists of n sequences and m columns as follows:

$$OF = \sum_{i=1}^{l} \{S_n(m_i) - G_n(m_i)\} \quad (2)$$

Where OF is the objective function, $S_n(m_i)$ is the similarity score of the column m_i, $G_n(m_i)$ is the gap penalty of the column m_i and l is the sequence length. The similarity score of the column m_i can be measured by the sum-of-pairs (SP). The SP-score $S(m_i)$ for the i-th column m_i is calculated as follows:

$$S(mi) = \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} \boldsymbol{s}(m_i^j, m_i^k) \quad (3)$$

where m_i^j is the j-th row in the i-th column. For aligning two residues x and y, the substitution matrix $s(x,y)$ is used to give the similarity score.

3.3 RNA Dataset

Table 1. The RNA Families Dataset Used in the Present Study

Test Group		GcvT	THI	Yybp -ykoy	Total
BRALiBase 2.1 (232 datasets)	k5	22	69	33	124
	k7	12	32	18	62
	k10	3	17	12	32
	k15	1	5	8	14
LocalExtR (90 datasets)	k20	10	10	10	30
	k40	10	10	5	25
	k60	10	10	0	20
	k80	5	10	0	15
Total		73	163	86	322

Table 2. Sequence Length of Each Test Group

Test Group		Sequence length		
		Avg.	Min	Max
BRALiBase 2.1 (232 datasets)	k5	109	96	125
	k7	110	94	131
	k10	108	94	129
	k15	110	88	137
LocalExtR (90 datasets)	k20	115	90	172
	k40	114	87	180
	k60	107	81	189
	k80	106	77	204

The BRAliBase (Benchmark RNA Alignment dataBase) dataset (Gardner et al., 2005) is constructed using alignments from release 5.0 of the Rfam database (Griffiths-Jones et al., 2005). It is a large collection of hand-curated multiple RNA sequence alignments of 36 RNA families. The BRAliBase 2.1 contains in total 18,990 aligned sets of sequences each consists of 2, 3, 5, 7, 10 and 15 sequences (categorized into k2, k3, k5, k7, k10 and k15 reference sets). The BraliBase 2.1 benchmark has reference alignments with almost 15 sequences. Thus, Wang et al. (Wang et al., 2007) have designed two

types of datasets to verify the potential of RNA sequence aligners: (i) Subset of BRAliBase which is highly variable and suitable for local MSA (contains 232 reference alignments) and categorized into k5, k7, k10 and k15 with 5, 7, 10 and 15 sequences, respectively. (ii) LocalEXtR which is an extension of BRAliBase 2.1 and comprises large-scale test groups with total of 90 large-scale reference alignments categorized into k20, k40, k60 and k80 with 20, 40, 60 and 80 sequences reference sets, respectively. The subset of BRAliBase 2.1 is selected from the most variable dataset within the suite. They are from THI, Glycine riboswitch and Yybp-Tkoy RNA families. Table 1 and Table 2 show the details of the test groups used in this study and the information about each test group.

3.4 Evaluation Design

To assess the quality of the produced alignment, a reference alignment from database benchmark is required. Two different measures are used to compare the accuracy of the produced alignment with the reference alignment. These two measures are the quality (Q) and the total column (TC) scores (Edgar, 2004). The Q score is the number of correctly aligned residue pairs in the test or predicted alignment divided by the number of residue pairs in the reference alignment. Other names of the Q score are also suggested such as developer score (Sauder et al., 2000) and sum of pairs (SPS) (Thompson et al., 1999). It should be noted, that this"sum of pairs" score is totally different from the SP score in objective function (Ahola et al., 2006). However, the TC is the number of the correctly aligned columns in the produced alignment divided by the number of columns in the reference alignment.

In any given alignment which consists of M columns, the i^{th} column is denoted by $A_{i1},A_{i2},...,A_{iN}$ where N is the number of sequences. For each pair of residues A_{ij} and A_{ik}, $p_i\,(j,k)$ is defined such that $p_i\,(j,k) = 1$ if residues A_{ij} and A_{ik} from the test alignment are aligned with each other in the reference alignment, otherwise $p_i\,(j,k) = 0$. The Score of i^{th} column can be calculated as follows:

$$Si = \sum_{j=1}^{N} \sum_{k=1,k\neq j}^{N} P_i(j,k) \qquad (4)$$

Then, the Q score for a given alignment can be calculated as follows:

$$Q = \frac{\sum_{i=1}^{M} S_i}{\sum_{i=1}^{Mr} S_{ri}} \qquad (5)$$

where M_r is the number of columns in the reference alignment and S_{ri} is the score S_i for the i^{th} column in the reference alignment. On the other hand, the score C_i of the i^{th} column is equal to 1 if all the residues in that column are aligned in the reference alignment, otherwise it is equal to 0. Therefore the TC score is calculated as follows:

$$TC = \sum_{i=1}^{M} \frac{C_i}{M} \qquad (6)$$

3.5 Harmony Search-based Methods for MSA

To address the MSA in this study, the MSA problem is designed as an optimization model by using a meta-heuristic approach. Thus, several methods are proposed to deal with the MSA through adapting the harmony search (HS) algorithm (Geem et al., 2001). The first method is the basic HS adaptation which adapts the HS for MSA (BHS-MSA). The second method modifies the improvisation stage of the harmony search algorithm and is called the modified harmony search algorithm for MSA (MHS-MSA). The third method hybridizes the harmony search algorithm with divide-and-conquer and blocks-based approaches and is called the hybrid harmony search algorithm for MSA (HHS-MSA).

3.5.1 Adapting the Harmony Search Algorithm

A framework is developed to adapt the HS algorithm for addressing the MSA problem. The HS components are harmony memory, fitness function, improvising new harmony solution and termination condition (Yang, 2009). These components of the HS algorithm should be adapted to satisfy the requirements of the MSA process. Herein, the BHS-MSA method is explained intensively. Basically, the Figure 1 describes the flowchart of the proposed BHS-MSA method and the adaptation of the HS algorithm.

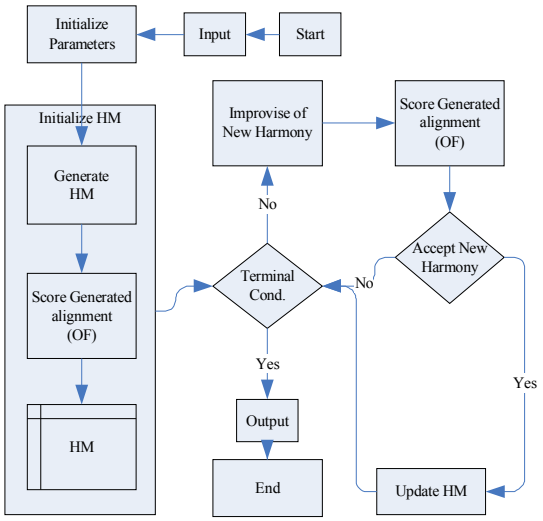

Figure 1. The Proposed BHS-MSA Method

Additionally, the BHS-MSA consists of the following steps:

Step 1: Initializing the HSA and MSA Parameters

It is evident that the balance between exploration and exploitation (diversification and intensification) is an important factor controlling the success of HS algorithm over other algorithms (Yang, 2009). It is well known that the performance of many population based methods is highly dependent on their parameter values (Eiben et al., 1999). HAS and MSA parameter are initialized based on the tuning procedures.

Step 2: Harmony Memory Initialization

The harmony memory is a location vector which contains a set of solutions (individuals) determined by Harmony Memory Size (HMS) as shown in Figure 2. The harmony itself is a matrix with *n* x *m* of gap positions where *n* is the sequences number (rows numbers) and *m* is the maximum length of gap positions (column numbers). Each harmony (individual) in the HM encodes into a RNA feasible alignment.

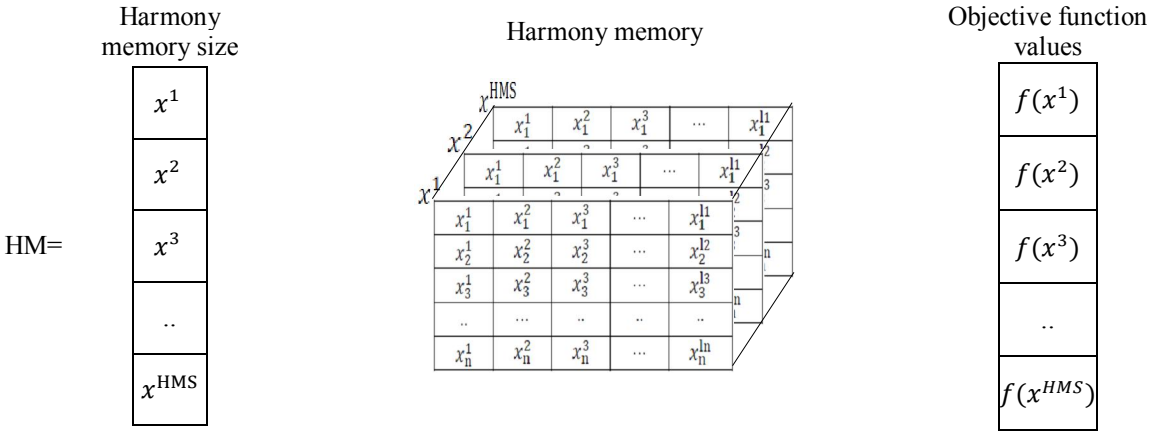

Figure 2: Harmony Memory Representation

Step 3: Improvising the New Harmony Solution

The simplest way to optimize the objective function score is to re-arrange gaps in the alignment (Wang and Lefkowitz, 2005). Improvising a new harmony is the essence of the HS algorithm and the cornerstone for building this algorithm. A new harmony (solution) is generated (improvised) from scratch $x' = (x_1', x_2', \dots, x_N')$ based on these mechanisms (memory consideration, random consideration and pitch adjustment).

Memory Consideration: In the memory consideration, the values (gap position) of the new harmony vector (new alignment solution) are randomly assigned by a value inherited from the historical values stored in the HM with a probability of HMCR$\in [0,1]$. Therefore, the value of decision variable (x_i') is chosen consecutively from $(x_i^1, x_i^2, \dots, x_i^{HMS})$.

Random Consideration: The decision variables (gap positions) that are not assigned with values according to memory consideration mechanism are randomly assigned according to their possible range with a probability of (1-HMCR). This increases the diversity of the solution and drives the system further to explore various diverse solutions so that global optimality can be attained. Equation 7 summaries these two steps of memory consideration and random consideration as follow:

$$x_i' \leftarrow \begin{cases} x_i' \in (x_i^1, x_i^2, \dots, x_i^{HMS})\ w.p\ HMCR \\ x_i' \qquad \in X_i\ w.p\ (1 - HMCR) \end{cases} \tag{7}$$

Pitch Adjustment: It is an additional search for a good solution in the search space. It is achieved through tuning a chosen decision variable (gap positions) in the new harmony vector (new alignment solution) $(x_1', x_2', \dots, x_N')$ inherited from the HM. Every decision variable of the new harmony vector is examined to decide whether it should or should not be pitch adjusted with the probability of PAR [0,1] as in Equations (8, 9).

Pitch adjustment

$$x_i' \leftarrow \begin{cases} Yes \in (x_i^1, x_i^2, \dots, x_i^{HMS})\ w.p\ PAR \\ No \qquad\qquad w.p\ (1 - PAR) \end{cases}$$ **Error! No text of specified style in document.** (8)

The new decision variable x_i' is modified based on the following equation:

$$x_i' = x_i' \pm bw \tag{9}$$

Here, *bw* is an arbitrary distance bandwidth used to determine the amount of movement or change that may occur to the components of the new variable (gap position). The way that the parameter PAR modifies the components of the new harmony is similar to the

musicians' behavior when they slightly change their tone frequencies in order to get better harmonies. It explores more solutions in the search space and improves the searching abilities. The bandwidth {1,5} is applied to change the position of the gap.

Step 4: Updating the Harmony Memory

In order to update the HM with a newly generated vector (alignment solution) $x' = (x'_1, x'_2, \dots, x'_N)$, the objective function is calculated for each new harmony vector x'. The solutions in the HM are sorted in an ascending order based on their objective function values. If the objective function value of the new vector is better than the worst harmony vector, the worst harmony vector is replaced by the new vector. Otherwise, this new vector is ignored.

Step 5: Checking the Stop Condition

The improvisation and the HM update processes are terminated when the maximum number of iterations is reached. Finally, the best harmony vector in the HM is selected and considered to be the best solution to the problem under investigation.

3.5.2 Modified Harmony Search Algorithm

This method is called the modified harmony search algorithm for MSA (MHS-MSA). The modified is inspired from the nested genetic algorithm as in (De Pauw and Vanrolleghema, 2006) and dividing the pitch adjustment operator as in (Al-Betar et al., 2010). The performance and the alignment quality of the earlier BHS-MSA method might be affected due to: large *search space* and the *feasible solutions* are far away from the optimal. In this method, the harmony search algorithm is modified during the improvisation stage (Step3). The modification strategy is designed to improve the acceptance rule by modifying the memory consideration and the random consideration. The acceptance rule of memory consideration is modified by adding another level of memory selection to the improvisation stage based on two levels: the sequences level and the gap positions level. Memory consideration operator (HMCR) is divided into two procedures: (i) assigning a feasible value of sequence with probability of HMCR1, and (ii) assigning a feasible value of gap position with probability of HMCR2.

(A)

$$s'_i \leftarrow \begin{cases} s'_i \in (s_i^1, s_i^2, \dots, sr_i^{HMS})\ w.p\ HMCR1\ for\ each\ x' \rightarrow (B) \\ s'_i \in X_i(profile)\ w.p\ (1 - HMCR1) \end{cases}$$

(B)

$$x'_i \leftarrow \begin{cases} x'_i \in (x_i^1, x_i^2, \dots, x_i^{HMS})\ w.p\ HMCR2 \\ x'_i \in X_i\ w.p\ (1 - HMCR2) \end{cases} \quad (10)$$

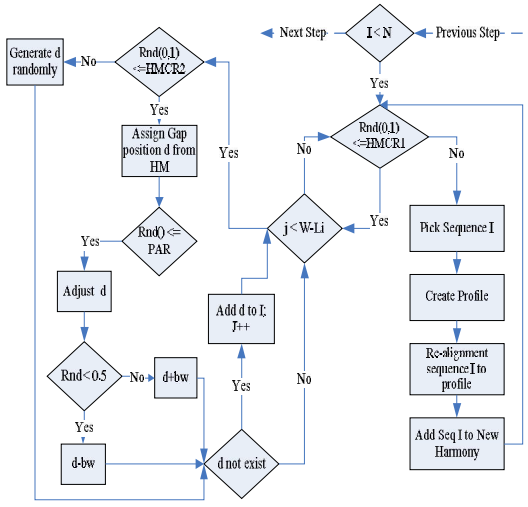

Figure 3. Flowchart of the Proposed MHS-MSA Method in the Improvise Stage.

Consequently, the random consideration is modified in two levels based on changing in memory consideration. In the first level, a sequence is randomly assigned by a feasible value from HM with probability of 1-HMCR1. The selected sequence is profiling to the rest of the aligned sequences. Profile approach is used to align individual sequence to the group of sequences. The gap positions are added as the outcome of this profile procedure. In the second level, gap positions are randomly assigned by a feasible value from HM with probability of 1-HMCR2. Moreover, these modification strategies are shown in the follows Figure 3.

3.5.3 Hybridization of HS and DCA methods

For highly NP-hard problems where the solution space is very huge and the possible solutions vary extensively. One way to deal with this huge solution search space is to divide and conquer, i.e. decompose the problem into sub-problems and treat each part separately and then combine these parts in the final stage (Hadwan et al., 2013). Therefore, the search space in multiple sequence alignment are narrowed down to a number of possible regions per sequence (Chen et al., 2005). This leads to simplify the search process by dividing the search space into small searching spaces instead of using the full search space. A hybrid method is proposed to reduce the

search space and improve the alignment quality by dividing the sequences into sub-regions. Based on the divide-and-conquer (DCA) approach (Stoye et al., 1997; Bai and Rezael, 2005; Chen et al., 2005; Liu et al., 2007), conserved regions are identified to find the best cut-points in the sequences set. Then, the sequences are cut into portions to form many subsequence sets. Thereafter, the MHS-MSA method is used to align the sub-sequence within those regions.

The HHS-MSA method is a combination of MHS-MSA, divide-and-conquer, and blocked-based methods. The aim of HHS-MSA is to identify the areas of local conservations before finding the global alignment. Conserved blocks can help to guide the identification of the homology and to assist the process of MSA. The ability to identify the conserved blocks has at least two advantages. It prevents the same blocks to be changed in the later process. Additionally, it speeds up the optimization process (Zhao and Jiang, 2001). The diagram of this method is illustrated in Figure 4.

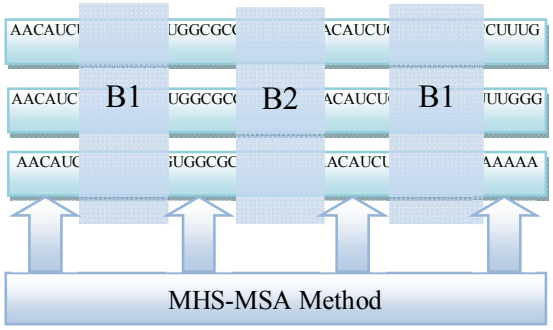

Figure 4: A hybrid Harmony Search for MSA (HHS-MSA).B1-B3 Conserved Regions (Blocks)

The steps for developing the HHS-MSA method are: (i) Building-Blocks (ii) Dividing the sequences at cut-points based on the blocks into conserved and non-conserved regions; (iii) Aligning the obtained non-conserved regions independently by using the MHS-MSA method; (iv) Construct the final alignment by combining the conserved and non-conserved regions. The processes of building-blocks consists of four steps: (a) finding all possible matched segments in each sequence pairs by using pairwise algorithm; (b) Using consistency concept to find all possible blocks that are acceptable; (c) Calculating the score value for each block by using the sum-of-pairs objective function; (d) Identifying and analysing the potentially useful block and selecting those are more consistent with each other;

4. Results and Discussion

This section contains experimental design, experimental results and comparative discussion. Results and comparisons have conducted to measure the behaviour of the proposed methods on addressing the MSA problem.

4.1 Experimental Design

The MSA and HS parameters are initialized based on the tuning procedure which provides the best setting values. Therefore, the parameters values are selected as: substitute matrix = RiboSum matrix (Klein and Eddy, 2003) and Gap open= -10, Gap extend= -0.5. Similar, The HS parameters are selected as: HMS=30, HMCR2 = 0.95, PAR=0.10 and HMCR1=0.55. Moreover, the fragment length in the building conserved blocks method is selected as 11.

4.2 Experimental Results

Table 3. Comparison results of the Alignment Quality between HHS-MSA with MHS-MSA and BHS-MSA Methods

Test Group	BHS-MSA		MHS-MSA		HHS-MSA	
	Alignment Accuracy					
	Q	TC	Q	TC	Q	TC
k5	0.407	0.201	0.592	0.385	**0.694**	**0.499**
k7	0.336	0.110	0.555	0.274	**0.687**	**0.428**
k10	0.308	0.088	0.542	0.221	**0.676**	**0.368**
k15	0.254	0.046	0.498	0.157	**0.665**	**0.365**
k20	0.104	0.007	0.406	0.059	**0.639**	**0.205**
k40	0.076	0.006	0.342	0.024	**0.697**	**0.159**
k60	0.064	0.006	0.293	0.015	**0.727**	**0.082**
k80	0.053	0.001	0.298	0.008	**0.754**	**0.075**

Table 3 summarizes the comparison results of alignment quality (Q and TC) among the HHS-MSA with previously proposed methods of BHS-MSA and MHS-MSA. Comparative analyses are performed and verified according to the references alignment provided with the dataset. It is clearly shown that the HHS-MSA has considerable improvement compared to previous methods of BHS-MSA and MHS-MSA. Therefore, the HHS-MSA aligns the sequences more effective and more accuracy due to many reasons.

HHS-MSA is first finding the most promising blocks, thus leading to the most accurate results. Conserved blocks can be considered as a useful guide in identifying the homology and assisting in improving the quality of the MSA. The ability to determine the well-aligned blocks has three main advantages: (i) it explores all possible blocks which are correct and consistent, (ii) it prevents the same block to be changed in the later process and (iii) it speeds up the optimization process. Additionally, HHS-MSA decomposes the problem and search space into multiple search space, this lead to improve the alignment accuracy as well as speed up the computation of MSA. The divide-and-conquer techniques can effectively reduce the space complexity for multiple sequence alignment(Chen et al., 2005). Dealing with the divided part of the MSA separately is a good direction to parallelism strategy.

4.3 Comparative Discussion

This section provides the comparative discussion among proposed method HHS-MSA and other 15 commonly used methods. As shown in Table 4, the alignment accuracy is expressed as quality (Q) and Total Column (TC) values. The result of quality (Q) outperforms *MAFFT* in the test groups greater than 20 sequences, and outperforms the other remaining methods. The result of the proposed HHS-MSA method shows a good quality in test groups k40, k60, and k80 whereas the *MAFFT* shows the good quality in test group k5, k7, and k10. However, the results of total column (TC) for the proposed HHS-MSA method are comparable to the *MAFFT* and *MUSCLE* whereas it outperforms other remaining methods.

Table 4. Comparison of the Alignment Quality (Q) Produced by the Proposed HHS-MSA and Commonly Used Methods

	k5		k7		k10		k15		k20		k40		k60		k80	
	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC	Q	TC
HHS-MSA	0.694	0.499	0.687	0.428	0.676	0.368	0.665	**0.365**	0.639	0.205	**0.697**	0.159	**0.727**	0.082	**0.754**	0.075
ClustalW	0.545	0.326	0.501	0.228	0.506	0.185	0.491	0.157	0.413	0.068	0.417	0.071	0.423	0.044	0.447	0.039
MAFFT	**0.711**	**0.525**	**0.712**	**0.476**	**0.685**	**0.400**	0.647	0.342	**0.647**	0.209	0.692	0.167	0.715	**0.126**	0.730	**0.133**
MUSCLE	0.682	0.490	0.672	0.418	0.660	0.361	0.611	0.342	0.606	**0.214**	0.678	**0.208**	0.694	0.101	0.716	0.106
ProbCons	0.422	0.217	0.414	0.162	0.395	0.127	0.355	0.088	0.381	0.065	0.410	0.050	0.447	0.027	0.453	0.019
T-Coffee	0.592	0.384	0.566	0.292	0.555	0.253	0.535	0.236	0.514	0.134	0.551	0.097	0.569	0.070	0.564	0.066
DIALIGN	0.619	0.419	0.586	0.316	0.571	0.250	0.569	0.218	0.566	0.164	0.567	0.097	0.580	0.075	0.578	0.065
DIALIGN -TX	0.583	0.365	0.563	0.287	0.555	0.237	0.555	0.214	0.502	0.110	0.496	0.070	0.488	0.055	0.508	0.057
SAGA	0.656	0.457	0.685	0.417	0.664	0.360	0.648	0.318	0.587	0.186	0.653	0.136	0.713	0.090	0.730	0.112
Kalign	0.691	0.496	0.665	0.418	0.646	0.360	0.627	0.333	0.591	0.160	0.582	0.097	0.607	0.062	0.650	0.041
PicXAA -pf	0.669	0.461	0.661	0.377	0.657	0.331	**0.681**	0.335	0.643	0.175	0.680	0.138	0.691	0.098	0.691	0.077
MSAProbs	0.414	0.207	0.407	0.155	0.393	0.126	0.366	0.101	0.393	0.073	0.428	0.058	0.467	0.036	0.468	0.022
PCMA	0.533	0.315	0.480	0.209	0.498	0.198	0.503	0.172	0.435	0.073	0.453	0.070	0.438	0.043	0.472	0.044
Align-m	0.614	0.429	0.605	0.343	0.613	0.317	0.595	0.286	0.585	0.197	0.624	0.165	0.672	0.136	0.658	0.103
PRIME	0.574	0.384	0.556	0.322	0.544	0.287	0.528	0.246	0.545	0.173	0.606	0.147	0.657	0.122	0.671	0.103
ProAlign	0.631	0.418	0.596	0.327	0.582	0.283	0.563	0.239	0.491	0.120	0.510	0.083	0.490	0.046	0.516	0.026

The best case of HHS-MSA performance, compared to *ProbCons* shows improvement with 69% in alignment accuracy (Q) and 190% in alignment accuracy (TC) (Table 5), whereas the performance is slightly similar compared to the alignment method in *MAFFT*. This finding can be generalized to other alignment methods within this performance ranges.

The results obtained by some group test have less than 20 sequences are slightly less accurate. This might be due to the objective function. To obtain results that are still near to biologically correct alignment, it seems that more sophisticated objective functions incorporating further biological criteria have to be considered. Additionally, the cut points are fixed and chosen based on the found blocks and this might not be a suitable point for some alignment test groups. The choice of the cut positions is critical for the success of the divide-and-conquer (DCA) procedure and inadequate cut positions in a division step can deteriorate the whole alignment.

Additionally, the search space of the MSA is simplified and divided to k blocks. The search space of the MSA can be calculated by accumulating all different permutation ways for each alignment and subtracting the search space of the blocks. In this context, the search space of aligning n sequences of maximum length L, and k blocks are shown in the following formula:

$$\sum_{i=1}^{n} \frac{L!}{L_i!} - k \times \sum_{i=1}^{n} \frac{L!}{(L-L_{bi})!} \qquad (11)$$

The overall search space is reduced by $k \times \sum_{i=1}^{n} \frac{L!}{(L-L_{bi})!}$ where k is the number of blocks and L_{bi} is the length of row i in the block k. This restricts the search space, leading to improve alignment quality. Furthermore, the results from the comparison analysis indicate that alignment accuracy (Q and TC) is

significantly difference among all commonly used methods (P<0.001, x^2=3123.971, df=15) and (P<0.001, x^2=2334.531, df=15) respectively.

Table 5. A Summary of the Comparison between HHS-MSA and Other Common Used Methods

	Alignment Accuracy		Improvement Percentage	
Method	Q	TC	Q	TC
HHS-MSA	**0.692**	**0.273**	-	-
ClustalW	0.468	0.140	48	95
MAFFT	**0.692**	**0.297**	0	-8
Muscle	0.665	**0.280**	4	-3
ProbCons	0.410	0.094	**69**	**190**
Tcoffee	0.556	0.191	24	43
Dialign	0.579	0.200	20	37
Dialign-tx	0.531	0.174	30	57
SAGA	0.667	0.259	4	5
Kalign	0.632	0.246	9	11
Picxaa-pf	0.672	0.249	3	10
MSAProbs	0.417	0.097	66	181
PCMA	0.476	0.141	45	94
Align-m	0.621	0.247	11	11
PRIME	0.585	0.223	18	22
ProAlign	0.547	0.193	27	41

5. Conclusions and Future Work

The overall goal of this paper is to provide an alternative approach to address the multiple sequence alignment (MSA) problem. Those methods (BHS-MSA, MHS-MSA, HHS-MSA and BHS-LMSA) are inspired from the harmony search (HS) algorithm. It is the first attempt which has not applied previously according to the available literature. The HHS-MSA method is proposed as a strategy to improve the alignment quality and to overcome the shortcoming of searching space problem by breaking the MSA into sub-regions based on the conserved blocks. Using the harmony search algorithm, align these sub-alignments separately and combine them to construct the final alignment. Interestingly, the final produced alignment of HHS-MSA shows better quality solutions compared to (BHS-MSA and MHS-MSA) methods. HHS-MSA has high efficient search space. It is also good for parallelization where it can implement each sub-region in separate node or processor using parallel architecture. In comparison with other common used methods, the HHS-MSA method is able to achieve a good ranking to the most common used methods and has remarkable competition in terms of quality (Q) especially when the alignment contains more than 20 sequences. On the other hand, it also achieves a high rank to the most common used methods in terms of total column (TC) and on the edge with the best alignment method (*MAFFT* and *MUSCLE*).

HHS-MSA brightens alignment quality versus iteration number. The larger iteration number; is the better alignment score. This might be due to the larger optimal alignments to be computed. This motivates to investigate effectiveness of adapting the harmony search algorithm on the local MSA.

Acknowledgements:

The authors would like to appreciate School of Computer Sciences as well as Universiti Sains Malaysia for their facilities and assistance. The authors are appreciative of the efforts of Parallel and Distributed Computing Centre (PDCC) academic and staffs for discussions and encouragement. The authors are appreciative of the reviewer's efforts for their helpful comments.

Corresponding Author:
Mubarak Saif Mohsen
School of Computer Sciences
Universiti Sains Malaysia
Penang, 11800, Malaysia
E-mail: mobarak_seif@yahoo.com

References

1. Abual-Rub, M. S., Al-Betar, M. A., Abdullah, R. & Khader, A. T. 2012. A hybrid harmony search algorithm for ab initio protein tertiary structure prediction. *Network Modeling and Analysis in Health Informatics and Bioinformatics*, 1-17.
2. Ahola, V., Aittokallio, T., Vihinen, M. & Uusipaikka, E. 2006. A statistical score for assessing the quality of multiple sequence alignments. *Bmc Bioinformatics,* 7, 484.
3. Al-Betar, M., Khader, A. & Liao, I. 2010. A harmony search with multi-pitch adjusting rate for the university course timetabling. *Recent*

Advances In Harmony Search Algorithm, 147-161.
4. Anbarasu, L., Narayanasamy, P. & Sundararajan, V. 2000. Multiple molecular sequence alignment by island parallel genetic algorithm. *Current Science,* 78, 858-863.
5. Bai, J. & Rezael, S. 2005. Multithreaded multiple sequence alignments. *2005 27th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Vols 1-7*, 2863-2866.
6. Bernhart, S. H., Hofacker, I. L., Will, S., Gruber, A. R. & Stadler, P. F. 2008. RNAalifold: improved consensus structure prediction for RNA alignments. *Bmc Bioinformatics,* 9, -.
7. Bi, C. 2008. Computational intelligence in multiple sequence alignment. *International Journal of Intelligent Computing and Cybernetics,* 1, 8-24.
8. Bonizzoni, P. & Della Vedova, G. 2001. The complexity of multiple sequence alignment with SP-score that is a metric. *Theoretical Computer Science,* 259, 63-79.
9. Cai, L. M., Juedes, D. & Liakhovitch, E. 2000. Evolutionary computation techniques for multiple sequence alignment. *Proceedings of the 2000 Congress on Evolutionary Computation, Vols 1 and 2*, 829-835.
10. Carrillo, H. & Lipman, D. 1988. The Multiple Sequence Alignment Problem in Biology. *Siam Journal on Applied Mathematics,* 48, 1073-1082.
11. Chellapilla, K. & Fogel, G. B. Year. Multiple sequence alignment using evolutionary programming. *In*, 1999.
12. Chen, S. M., Lin, C. H. & Chen, S. J. 2005. Multiple DNA sequence alignment based on genetic algorithms and divide-and-conquer techniques. *International Journal of Applied Science and Engineering,* 3, 89-100.
13. Da Silva, F. J. M., Perez, J. M. S., Pulido, J. a. G. & Rodriguez, M. a. V. 2008. Alinea GA: A Genetic Algorithm for Multiple Sequence Alignment. *New Challenges in Applied Intelligence Technologies,* 134, 309-318.
14. De Pauw, D. J. W. & Vanrolleghema, P. 2006. Nesting genetic algorithms to solve a robust optimal experimental design problem.
15. Dorigo, M. & Blum, C. 2005. Ant colony optimization theory: A survey. *Theoretical Computer Science,* 344, 243-278.
16. Durbin, R. 1998. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*, Cambridge Univ Pr.
17. Edgar, R. C. 2004. MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research,* 32, 1792-1797.
18. Edgar, R. C. & Batzoglou, S. 2006. Multiple sequence alignment. *Current Opinion in Structural Biology,* 16, 368-373.
19. Eiben, A. E., Hinterding, R. & Michalewicz, Z. 1999. Parameter control in evolutionary algorithms. *Evolutionary Computation, IEEE Transactions on,* 3, 124-141.
20. Ergezer, H. & Leblebicioglu, K. 2006. Refining the progressive multiple sequence alignment score using genetic algorithms. *Artificial Intelligence and Neural Networks,* 3949, 177-184.
21. Gardner, P. P., Wilm, A. & Washietl, S. 2005. A benchmark of multiple sequence alignment programs upon structural RNAs. *Nucleic Acids Research,* 33, 2433-2439.
22. Geem, Z. W., Kim, J. H. & Loganathan, G. V. 2001. A new heuristic optimization algorithm: Harmony search. *Simulation,* 76, 60-68.
23. Gondro, C. & Kinghorn, B. P. 2007. A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research,* 6, 964-982.
24. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S. R. & Bateman, A. 2005. Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Research,* 33, 121-124.
25. Hadwan, M., Ayob, M., Sabar, N. R. & Qu, R. 2013. A harmony search algorithm for nurse rostering problems. *Information Sciences*.
26. Hai-Xia, L., Wen-Bo, X., Jun, S. & Wen-Juan, J. 2009. Multiple Sequence Alignment Based on a Binary Particle Swarm Optimization Algorithm. *Proceedings of the 2009 Fifth International Conference on Natural Computation - Volume 03*. IEEE Computer Society.
27. Hansen, J. L., Schmeing, T. M., Moore, P. B. & Steitz, T. A. 2002. Structural insights into peptide bond formation. *Proceedings of the National Academy of Sciences,* 99, 11670.
28. Hickson, R. E., Simon, C. & Perrey, S. W. 2000. The performance of several multiple-sequence alignment programs in relation to secondary-structure features for an rRNA sequence. *Molecular Biology and Evolution,* 17, 530-539.
29. Isokawa, M., Wayama, M. & Shimizu, T. 1996. Multiple sequence alignment using a genetic algorithm. *Genome Informatics,* 7, 176-177.
30. Jeevitesh. M.S, Behera, N., Jose, J., Kant, K. & Dey, A. 2010. Higher accuracy protein Multiple Sequence Alignment by Stochastic Algorithm.
31. Joo, K., Lee, J., Kim, I., Lee, S. J. & Lee, J. 2008. Multiple Sequence Alignment by

Conformational Space Annealing. *Biophysical Journal,* 95, 4813-4819.
32. Jorng-Tzong Horng, Ching-Mei Lin, Baw-Jhiune Liu & Kao, C.-Y. Year. Using Genetic Algorithms to Solve Multiple Sequence Alignments. *In:* Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2000), 10-12 Jul. 2000 2000 Morgan Kaufmann, Las Vegas, Nevada, USA. 883–890.
33. Juang, W. S. & Su, S. F. 2008. Multiple sequence alignment using modified dynamic programming and particle swarm optimization. *Journal of the Chinese Institute of Engineers,* 31, 659-673.
34. Keith, J. M., Adams, P., Bryant, D., Kroese, D. P., Mitchelson, K. R., Cochran, D. a. E. & Lala, G. H. 2002. A simulated annealing algorithm for finding consensus sequences. *Bioinformatics,* 18, 1494-1499.
35. Kim, J., Pramanik, S. & Chung, M. J. 1994. Multiple Sequence Alignment Using Simulated Annealing. *Computer Applications in the Biosciences,* 10, 419-426.
36. Klein, R. J. & Eddy, S. R. 2003. RSEARCH: finding homologs of single structured RNA sequences. *Bmc Bioinformatics,* 4, 44.
37. Lai, C. C., Wu, C. H. & Ho, C. C. 2009. Using Genetic Algorithm to Solve Multiple Sequence Alignment Problem. *International Journal of Software Engineering and Knowledge Engineering,* 19, 871-888.
38. Lee, Z. J., Su, S. F., Chuang, C. C. & Liu, K. H. 2008. Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment. *Applied Soft Computing,* 8, 55-78.
39. Lei, X. J., Sun, J. J. & Ma, Q. Z. 2009. Multiple Sequence Alignment Based on Chaotic PSO. *Computational Intelligence and Intelligent Systems,* 51, 351-360.
40. Lightner, C. A. 2008. *A Tabu Search Approach to Multiple Sequence Alignment.* Ph.D.
41. Liu, W., Chen, L. & Chen, J. 2007. An efficient algorithm for multiple sequence alignment based on ant colony optimisation and divide-and-conquer method. *New Zealand Journal of Agricultural Research,* 50, 617-626.
42. Manjarres, D., Landa-Torres, I., Gil-Lopez, S., Del Ser, J., Bilbao, M., Salcedo-Sanz, S. & Geem, Z. 2013. A survey on applications of the harmony search algorithm. *Engineering Applications of Artificial Intelligence*.
43. Mohsen, A. M., Khader, A. T. & Ramachandram, D. 2010. An optimization algorithm based on harmony search for rna secondary structure prediction. *Recent Advances In Harmony Search Algorithm.* Springer.
44. Mohsen, M. S. & Abdullah, R. 2011. HS-MSA: New algorithm based on Meta-heuristic harmony search for solving multiple sequence alignment. *International Journal of Computer Science,* 9.
45. Nicholas, H. B., Ropelewski, A. J. & Deerfield, D. W. 2002. Strategies for multiple sequence alignment. *Biotechniques,* 32, 572-591.
46. Notredame, C. 2002. Recent progress in multiple sequence alignment: a survey. *Pharmacogenomics,* 3, 131-144.
47. Notredame, C. & Higgins, D. G. 1996. SAGA: Sequence alignment by genetic algorithm. *Nucleic Acids Research,* 24, 1515-1524.
48. Omar, M. F., Salam, R. A., Abdullah, R. & Rashid, N. A. 2005. Multiple Sequence Alignment Using Optimization Algorithms. *International Journal of Computational Intelligence,* 1, 2.
49. Pedro F. Rodriguez, Nino, L. F. & Alonso, O. M. 2007. Multiple sequence alignment using swarm intelligence. *International Journal of Computational Intelligence Research* 3, 123-130.
50. Rasmussen, T. K. & Krink, T. 2003. Improved Hidden Markov Model training for multiple sequence alignment by a particle swarm optimization - evolutionary algorithm hybrid. *Biosystems,* 72, 5-17.
51. Riaz, T., Wang, Y. & Kuo-Bin, L. 2005. A TABU SEARCH ALGORITHM FOR POST-PROCESSING MULTIPLE SEQUENCE ALIGNMENT. *Journal of Bioinformatics & Computational Biology,* 3, 145-156.
52. Roc, R. O. C. 2007. Multiple DNA Sequence Alignment Based on Genetic Simulated Annealing Techniques. *Information and Management,* 18, 97-111.
53. Sauder, J. M., Arthur, J. W. & Dunbrack Jr, R. L. 2000. Large-scale comparison of protein sequence alignment algorithms with structure alignments. *Proteins: Structure, Function, and Bioinformatics,* 40, 6-22.
54. Shyu, C. & Foster, J. A. 2003. Evolving consensus sequence for multiple sequence alignment with a genetic algorithm. *Genetic and Evolutionary Computation - Gecco 2003, Pt Ii, Proceedings,* 2724, 2313-2324.
55. Stoye, J., Moulton, V. & Dress, A. W. M. 1997. DCA: An efficient implementation of the divide-and-conquer approach to simultaneous multiple sequence alignment. *Computer Applications in the Biosciences,* 13, 625-626.
56. Taheri, J. & Zomaya, A. Y. 2009. RBT-GA: a novel metaheuristic for solving the multiple sequence alignment problem. *Bmc Genomics,* 10, -.

57. 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.
58. Thompson, J. D., Plewniak, F. & Poch, O. 1999. A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research,* 27, 2682-2690.
59. Trystram, D. & Zola, J. 2007. Multiple Sequence Alignment and Phylogenetic Inference. *Grid Computing for Bioinformatics and Computational Biology,* 1, 123.
60. Tsang, H. H. & Wiese, K. C. Year. SARNA-Predict: A study of RNA secondary structure prediction using different annealing schedules. *In*, 2007. IEEE, 239-246.
61. Uren, P. J., Cameron-Jones, R. M. & Sale, A. H. J. 2007. MAUSA: Using simulated annealing for guide tree construction in multiple sequence alignment. *Ai 2007: Advances in Artificial Intelligence, Proceedings,* 4830, 599-608.
62. Wallace, I. M., Blackshields, G. & Higgins, D. G. 2005. Multiple sequence alignments. *Current Opinion in Structural Biology,* 15, 261-266.
63. Wang, C. L. & Lefkowitz, E. J. 2005. Genomic multiple sequence alignments: refinement using a genetic algorithm. *Bmc Bioinformatics,* 6, -.
64. Wang, S., Gutell, R. R. & Miranker, D. P. 2007. Biclustering as a method for RNA local multiple sequence alignment. *Bioinformatics,* 23, 3289-3296.
65. Xu, F. S. & Chen, Y. H. 2009. A Method for Multiple Sequence Alignment Based on Particle Swarm Optimization. *Emerging Intelligent Computing Technology and Applications: With Aspects of Artificial Intelligence,* 5755, 965-973.
66. Xu, X. & Lei, X. 2010. Multiple sequence alignment based on abc_sa. *Artificial Intelligence and Computational Intelligence.* Springer.
67. Yang, X.-S. 2009. Harmony Search as a Metaheuristic Algorithm. *Music-Inspired Harmony Search Algorithm.*
68. Zablocki, F. B. R. 2007. *Multiple Sequence Alignment using Particle Swarm Optimization.* Master Thesis, University of Pretoria.
69. Zhang, C. & Wong, A. K. C. 1997a. A genetic algorithm for multiple molecular sequence alignment. *Computer Applications in the Biosciences,* 13, 565-581.
70. Zhang, C. & Wong, A. K. C. 1997b. Toward efficient multiple molecular sequence alignment: A system of genetic algorithm and dynamic programming. *Ieee Transactions on Systems Man and Cybernetics Part B-Cybernetics,* 27, 918-932.
71. Zhang, X. & Kahveci, T. 2007. QOMA: quasi-optimal multiple alignment of protein sequences. *Bioinformatics,* 23, 162-168.
72. Zhao, P. & Jiang, T. 2001. A heuristic algorithm for multiple sequence alignment based on blocks. *Journal of Combinatorial Optimization,* 5, 95-115.
73. Zhao, Y. D., Ma, P., Lan, J., Liang, C. & Ji, G. L. 2008. An Improved Ant Colony Algorithm for DNA Sequence Alignment. *Isise 2008: International Symposium on Information Science and Engineering, Vol 2*, 683-688.

11/18/2018