Swarm Intelligence-based Hierarchical Clustering for Identification of ncRNA using Covariance Search Model.

PRATIWI, Lustiana, CHOO, Yun-Huoy, MUDA, Azah Kamilah and PRATAMA, Satrya Fajri

Swarm Intelligence-based Hierarchical Clustering for Identification of ncRNA using Covariance Search Model

Lustiana Pratiwi[1], Yun-Huoy Choo[2] and Azah Kamilah Muda[3]
Computational Intelligence and Technologies Lab Research Group
Fakulti Teknologi dan Maklumat Komunikasi
Universiti Teknikal Malaysia Melaka
Melaka, Malaysia

Satrya Fajri Pratama[4]
Department of Computing
College of Business, Technology and Engineering
Sheffield Hallam University
Sheffield, United Kingdom

***Abstract*—Covariance Model (CM) has been quite effective in finding potential members of existing families of non-coding Ribonucleic Acid (ncRNA) identification and has provided excellent accuracy in genome sequence database. However, it has significant drawbacks with family-specific search. An existing Hierarchical Agglomerative Clustering (HAC) technique merged overlapping sequences which is known as combined CM (CCM). However, the structural information will be discarded, and the sequence features of each family will be significantly diluted as the number of original structures increases. Additionally, it can only find members of the existing families and is not useful in finding potential members of novel ncRNA families. Furthermore, it is also important to construct generic sequence models which can be used to recognize new potential members of novel ncRNA families and define unknown ncRNA sequence as the potential members for known families. To achieve these objectives, this study proposes to implement Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) to ensure the CCMs have the best quality for every level of dendrogram hierarchy. This study will also apply distance matrix as the criteria to measure the compatibility between two CMs. The proposed techniques will be using five gene families with fifty sequences from each family from Rfam database which will be divided into training and testing dataset to test CMs combination method. The proposed techniques will be compared to the existing HAC in terms of identification accuracy, sum of bit-scores, and processing time, where each of these performance measurements will be statistically validated.**

***Keywords*—Covariance model; ncRNA identification; swarm intelligence; hierarchical clustering**

I. Introduction

Distinguishing many different classes of noncoding (nc) ribonucleic acids (RNA) according to the performance of its variety roles has been an prevalent area in bio-computational technology [1], [2], [3], [4], [5]. For example, after the relationship between RNA structure and its function is discovered, it is desirable to know the common structure of homologous RNAs to find out the functional signatures. It is also desirable to scan a genome looking for ncRNAs. Strategies employed in protein coding gene identification are not commonly applicable for ncRNAs. Therefore, the identification of ncRNA remains an open problem in bioinformatics. However, two-bases are not necessarily covary, since some point mutations, such as G–C to G–U, are still considered as base pairing [6].

Thus, methods searching for covariation may miss valuable information. The main drawback of covariance model (CM) is the use of information of a specific gene family to gain in accuracy [7]. In areas of ncRNA identification, CM has been quite effective in finding potential members of existing families and has provided excellent accuracy in genome sequence database [8], [9], [10]. Representation of multiple secondary structure alignment using a hairpin loop based on an ordered tree are constructed automatically from existing sequence alignments or even from unaligned example sequence [11].

However, it also has considerable disadvantage, such as computationally expensive and thus hindering its application in practice [6], [12]. Apart from having problems with family-specific search that includes large processing requirements, ambiguity in defining which sequences form a family and insufficient numbers of known sequences to properly estimate model parameters are known to be a big challenge in identification of ncRNA. To improve CM performance, hierarchical clustering, as the most frequently used mathematical technique, tries to group genes into small clusters and to group clusters into higher-level systems [13]. Hierarchical clustering provides a series of nested partitions of the dataset. It splits the data into a nested tree structure, where the levels of the tree show similarity or dissimilarity among the clusters at different levels [5], [7].

In regards to this issue, hierarchical clustering has been known as a efficient and useful technique for analyzing genome data and can be applied to group known ncRNA gene families [13], [14], [15], [16]. Past researchers have applied hierarchical clustering to support the identification process in combining and clustering group of families [15]. Lessening the computational cost imposed by covariance model (CM) based noncoding RNA (ncRNA) gene finding is desirable to search the sequence data using a large number of ncRNA families [14], [17], [18]. The main consideration of CM is searching a gigabyte database of sequences for all known ncRNA gene families, which will take quite a long time, and thus is not practical [19], [22].

Hierarchical clustering has successfully reduced the search time to find members from all original ncRNA families using dot-bracket notations [13]. However, its performance continuously declining when additional families of CMs are introduced into the CCM. This is because more structural information will be excluded, and the sequence features of each family will be significantly less apparent as the number of original CMs increase [5]. Furthermore, it is not sufficiently covering all the problem spaces. A swarm intelligence-based hierarchical algorithm is proposed in this study to select base pairs from two or more CMs of ncRNA families, such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). The general idea is to select base pairs from one sequence feature that has fewest conflicts with the base pairs in another structure and construct new CCM structures, which will increase the discernibility performance of the CCMs. Thus, the paper is organized as follows: the next section further describes the problem which motivated this study, while Sections III and IV elaborate the proposed method and present the results, respectively, followed by the discussion of the results in Section V, and the last section concludes this work.

II. Hierarchical Clustering for Covariance Model

In the previous study conducted by [14], a technique was introduced to reduce the search complexity when a target genome is searched for more than one known ncRNA gene family. The basic construct of the technique is to combine different CMs into one single CM which captures part of both sequence and structure features of each CM by selecting randomly three sequences, known as cluster, from each family by using hierarchical clustering. Hierarchical clustering is a powerful and useful technique for analyzing genome data because hierarchical clustering can generate a dendrogram which helps organize the CCMs. Each non-leaf node of the dendrogram is a CCM of its child nodes, and each leaf node represents a single CM of a ncRNA gene family [5]. Fig. 1 shows an example of possible structure of a dendrogram.

The existing technique applied the agglomerative approach to cluster ncRNA genes [5]. After the clustering of ncRNA gene families, the existing technique built the CCM for any two CMs in the dendrogram that share a common parent which combined the two selected CMs such that the new CCM captured features of both original CMs. CMs are built from multiple sequence alignment with annotated secondary structure. Thus, the general idea of combining two CMs of the existing technique was to select part of the multiple sequence alignment columns of each CM and connect them together to create a new multiple sequence alignment, which the new CM is built from [5].

Since the multiple sequence alignment is annotated with secondary structure, and each column of multiple alignment corresponds to a secondary structure symbol, either a dot or a bracket, indicating whether it is a paired base or not, combining the multiple sequence alignment is equivalent to combining the annotated secondary structure [18]. To combine secondary structures, the key point is to determine how to select base pairs from the two original secondary structures and put them into the new structure. It is obvious that the algorithm cannot just select all the base pairs from both original structures and simply connect them together, since it will significantly increase the complexity of the CM and make no difference to searching with the two original CMs separately [4].

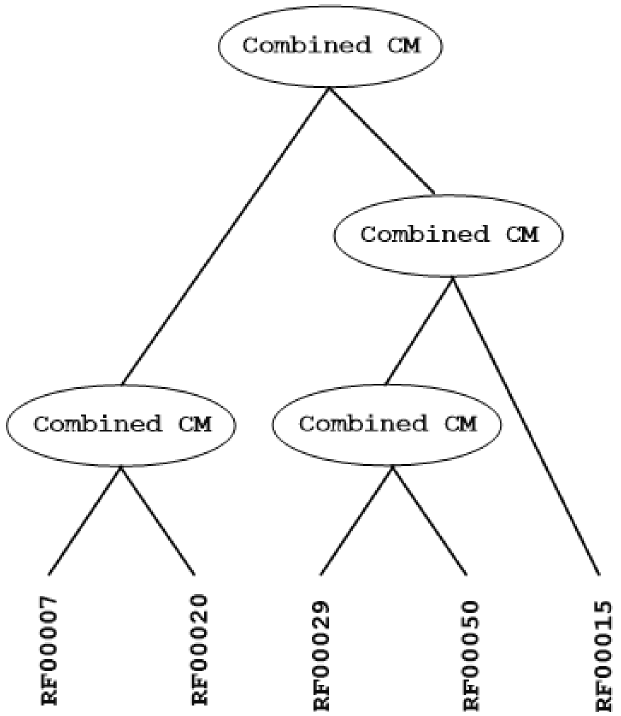

Fig. 1. A Sample of Hierarchical Clustering Result of Five ncRNA Families from Rfam Database.

In most methods of hierarchical agglomerative clustering (HAC), a measure of dissimilarity between clusters is required to decide which clusters should be combined [20], [21]. In the existing technique [23], the members of the cluster are ncRNA gene families, each of which is represented by its secondary structure in dot-bracket notation. Dot-bracket notation is widely used in describing RNA secondary structure. It uses matching brackets to indicate paired bases and dots to denote unpaired bases. This study defines a different distance function that is particularly suitable to deal with the secondary structure data in dot-bracket notation. However, there are two definitions that need to be introduced [5].

Definition 1. *Base pair conflict: Given two RNA secondary structures, a base pair* (m, n) *from one secondary structure is called base pair conflict if there exists a base pair* (i, j) *in the other secondary structure such that m¡i¡n¡j, or i¡m¡j¡n.*

Definition 2. *Structure distance: Given two RNA secondary structures, the structure distance between them is the average of the number of base pair conflicts in each secondary structure.*

The definition of base pair conflict is similar to the definition of pseudoknots, but the difference is that base pair conflict is for base pairs in two different structures while pseudoknots is for those in one structure [14], [24], [25]. Since CM cannot deal with pseudoknots which should be avoided in the combined structure, only one of the two conflict base pairs can be retained in the combined structure [5], [23].

Unlike the distance functions used in most clustering algorithms that measure dissimilarity between observations, structure distance, the distance function applied in the existing technique, is a measurement of compatibility of two RNA secondary structures [5]. The smaller this value is, the more compatible the two secondary structures are. The compatibility between two secondary structures shows how much structural information can be retained when they are combined. Since the objective is to build a CCM that can capture as much information as possible about both original CMs, two CMs with more compatible secondary structure components would be perfect to be combined. The basic process of HAC [5] is as shown in Algorithm 1.

Algorithm 1 HAC Process

```
1: Start
2: Assign each ncRNA gene family to its own cluster, then build the distance matrix by computing the structure distance between each family.
3: Find the closest pair of clusters (the minimal element in distance matrix) and combine their secondary structures to create a new family, which corresponds to a non-leaf node in the dendrogram, and then remove the cluster pair from dendrogram.
4: Compute structure distance between the new cluster and each of the old clusters.
5: Repeat steps 2 and 3 until there is only one cluster, which corresponds to the root of the dendrogram.
6: End
```

Ref. [5] proposed three criteria that should be followed for selecting base pairs from two secondary structures. First, the set of selected base pairs should contain as many base pairs as possible. Since the greater number of base pairs are selected, the more secondary structure components are retained, which also means the more likely a target sequence will be found when searching the genome database. Second, the base pairs selected from one CM should not conflict with those from the other CM. This means there are no pseudoknots in the combined secondary structure due to the reason that CM cannot deal with pseudoknots. Third, each CM should have roughly the same number of base pairs selected, which means this study wants to make a balance between the two original secondary structures.

A greedy algorithm to select base pairs from two secondary structures to form a new secondary structure that satisfies the above three criteria is outlined in Algorithm 2, which is termed as Hierarchical Agglomerative Clustered Covariance Model (HACCM) by [5]. The general idea is to select a base pair from one structure that has fewest conflicts (pseudoknots) with the base pairs in the other structure, which means selecting this base pair will cause fewest deletions of base pairs in the other structure, and such selection takes turns between the two structures.

This existing technique of [5] performs rather well, in most cases, when not too many CMs are combined, where the CCM can successfully represent members from all original families and hardly provide any false alarms. However, its performance gradually deteriorate as more families of CMs are added into the CCM, since more structural information will be discarded, and the sequence features of each family will be significantly diluted as the number of original CMs increase [5]. Furthermore, the original technique is not sufficiently covering all the problem spaces, which means the solution provided may not be the best solution. The frequently used strategy to explore wider problem spaces is by employing evolutionary and swarm intelligence, where each member of the swarm represents one possible solution. By adjusting the number of unique members based on the resource availability, the problem spaces can be further explored.

Algorithm 2 Pseudocode to Construct CCM using HACCM [5]

```
1: Start
2: Generate a list of $N_f$ gene families (each family consists of five randomly selected clusters, one cluster from each original family)
3: Calibrate the gene families using *cmcalibrate* tool from Infernal package
4: while $N_f > 0$ do
5:     Generate distance matrix between gene families
6:     Select two nearest gene families
7:     Construct CCM from two nearest families by using base pair conflicts
8:     Exclude the two families from future distance matrix calculation
9:     Calculate the TP, TN, FP, and FN from the bit-score of the CCM using members of gene families
10: end while
11: End
```

To achieve this objective, this study proposes to implement swarm intelligence to ensure the CCMs have the best quality for every level of dendrogram hierarchy. On the initialization phase of the swarm intelligence technique, several possible combinations of the CMs will be generated and considered as a single member of swarm intelligence. The selected number of sequences will be unique and the number of sequences for these particles will vary to ensure the maximum coverage. This study will also apply distance matrix as the criteria to measure the compatibility between two CMs.

However, the difference between this study and the previous one is that the selected CCMs will be directly evaluated by CM scoring model, which will be assigned as one of the components of fitness value of the member of swarm intelligence instead of producing the complete dendrogram before evaluating the CM scoring model. The CM from the best member in the current iteration will be then combined with other CMs to generate the new CMs in the subsequent iterations. Furthermore, this study aims to generate reliable cluster pool by having proper selection approach to reduce sequence features dilution in CCMs and to optimize the CCMs selection models which is based on Hierarchical Agglomerative Clustering using Swarm Optimization in finding potential members of novel ncRNA. By the end of the optimization process, this study will construct generic sequence models which represent multiple families to identify unknown family members of ncRNA.

III. Proposed Swarm Intelligence-Based HACCM

This section discusses the tasks undertaken to develop the proposed techniques by hybridizing them with swarm intelligence (SI) techniques, such as Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). In Algorithm 3, the hybridization with SI techniques is primarily to prevent the existing HACCM technique [5] selects the local optima and forces it to reevaluate the candidates with the same merit in every iteration to find the global optima.

Algorithm 3 Optimization Process using SI for HACCM

```
Start
Step 1
    Generate balanced cluster pool based on minimum number
Step 2
    Generate similarity matrix between clusters
    Find the closest pair of clusters using greedy algorithm and combine it
    Construct tree-based CCM structures
Step 3
    Measure the fitness value of each combined clusters using CM search score and confusion matrix
Step 4
    Compare and get best hierarchical CM structure with highest fitness value
    Update clusters combination with PSO's velocity composite values
Step 5
    Set threshold Fit = 1 or stagnant fitness for 5 iterations
    Proceed to Step 2 if threshold not fulfilled
End
```

The fitness function for the PSO and GA should be identified beforehand. Although there are several possible performance measurements available, such as sensitivity or recall, precision, F_1-score, specificity, and accuracy, this study prefer to use accuracy as the parameter of fitness function because they do not solely rely on the true and false positives. However, in case of the similar fitness value between two or more PSO particles or GA individuals, other criteria should be taken into consideration, which is the sum of bit-scores, which is obtained by evaluating the CM using the *cmsearch* program from Infernal package [3]. In this study, the sum of literal value of bit-scores or similarity score of the families will be used in the fitness function as the tiebreaker parameter. Thus, the proposed fitness function of this study is given in Eq. 1.

$$F_i = \alpha \times \frac{Acc_i(t) - Acc_i(0)}{Acc_i(0)} + \beta \times \frac{SS_i(t) - SS_i(0)}{SS_i(0)}, \quad (1)$$

where Acc_i is the accuracy of ith member (particle in case of PSO and individual in case of GA) in the tth iteration ($t = 0, 1, ...$), and SS_i is the sum of bit-scores from the families of ith member, with $Acc_i(0)$ and $SS_i(0)$ are the accuracy and sum of bit-scores of the original HACCM, which is inspired from the fitness function of HelixPSO for finding RNA secondary structure [28], given in Eq. 2.

$$F = \alpha \times \frac{|S| \cap |C|}{|C|} + \beta \times \min\left(0, \frac{E(S)}{mfE}\right), \quad (2)$$

where α and β in Eqs. 1 and 2 are the parameters used to determine the importance of classification accuracy and the subset size, where $\alpha \in [0, 1]$ and $\beta = 1-\alpha$. In this study, the α is set to 0.9, while β is set to 0.1. By setting the accuracy and sum of bit-scores of the current PSO particle or GA individual relative to the original HACCM, the values of fitness value can be guaranteed to not deviate too much to the original HACCM performance.

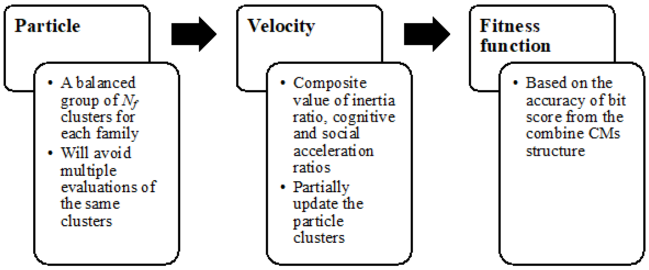
Particle
• A balanced group of N_f clusters for each family
• Will avoid multiple evaluations of the same clusters
Velocity
• Composite value of inertia ratio, cognitive and social acceleration ratios
• Partially update the particle clusters
Fitness function
• Based on the accuracy of bit score from the combine CMs structure

Fig. 2. Optimization Process using PSO.

A. PSO and HACCM Hybridization

In this study, Particle Swarm Optimization (PSO) [26], [27] is selected as one of swarm intelligence techniques to optimize HACCM. One of the main considerations taken for selecting PSO is due to its simple yet effective implementation. The main idea of Particle Swarm Optimized HACCM (PSO-HACCM) is that fitness function of PSO is modified by implementing confusion matrix-based performance measurement techniques calculated from the results of bit-score obtained from *cmsearch* of Infernal package. This is to allow the most optimal interaction between PSO and HACCM, and thus allow for wider search space exploration. Furthermore, there are multiple instances of HACCM executed concurrently; each of them is executed in PSO particle.

This study uses the sum of bit-scores value to be assigned as the particle position. Meanwhile, the fitness value must be identified beforehand since there are several possible confusion matrix-based performance measurement techniques, however this study found the accuracy. Each particle will examine diverse set of CM family clusters, and thus produce unique results, this is because the examined CM family clusters set and its results are recorded, to prevent different particles from examining the same set multiple times.

Apart from the modification to the fitness function, the particle velocity update strategy is also modified which is shown in Fig. 2. Like original PSO, the velocity of the PSO v_i in the $(t+1)$th iteration is affected by inertia ratio I_i, cognitive acceleration ratio C_i, and social acceleration ratio S_i, although is it slightly modified such that

$$v_i(t+1) = I_r + C_r + S_r, \quad (3)$$

$$I_i = I \times v_i(t), \quad (4)$$

$$C_i = C \times rand() \times (p_i - x_i(t)), \quad (5)$$

$$S_i = S \times rand() \times (p_{best} - x_i(t)), \quad (6)$$

where I, C, and S are the inertia weightage, cognitive acceleration coefficient, and social acceleration coefficient, respectively. In this study, the constants I is set to 0.729844 while C and S is set to 1.49618 [29]. Since the particle is made up N_f clusters, the implementation of these ratios is that there will be maximum $\lfloor \frac{I_r \times N_f}{v_i(t)} \rfloor$ numbers of clusters are reselected from clusters pool, maximum $\lfloor \frac{C_r \times N_f}{v_i(t)} \rfloor$ numbers of clusters are reselected from the particle's personal best, and maximum $\lfloor \frac{S_r \times N_f}{v_i(t)} \rfloor$ numbers of clusters are reselected from the global best particle. The algorithm of PSO-HACCM is illustrated in Algorithm 4.

Algorithm 4 The Pseudocode to Construct Hybrid PSO and HACCM

```
Start
Set number of particles $N_p = 3$
Set number of iterations $N_t = 100$
for $i = 1...N_p$ do
    Generate particle $P_i$, which is the CCM
    Calculate initial sum of bit-scores of CCM as position $X_i$
    Calculate fitness value $F_i$
    Calculate inertia ratio $I_i$, cognitive acceleration ratio $C_i$, and social acceleration ratio $S_i$
    Initialize velocity $V_i = I_i + C_i + S_i$
    Select global best $G_{best}$ based on maximum fitness value
end for
for $t = 1...N_t$ do
    for $i = 1...N_p$ where $P_i! = G_{best}$ do
        $NumInert = I_i * N_f/V_i$
        $NumCog = C_i \times N_f/V_i$
        $NumSoc = S_i \times N_f/V_i$
        Randomly select $n1 \leq NumInert$ clusters from cluster pool
        Randomly select $n2 \leq NumCog$ clusters from $P_{best}$
        Randomly select $n3 \leq NumSoc$ clusters from $G_{best}$
        Generate hierarchical CCM from new clusters
        Update $X_i$, $F_i$, $I_i$, $C_i$, $S_i$, and $V_i$
    end for
    Select new Gbest
    if $F_{G_{best}} == 1$ OR is stagnant after 5 iterations then
        Stop iteration
    end if
end for
End
```

B. GA and HACCM Hybridization

In this study, Genetic Algorithm (GA) [31] is selected as another swarm intelligence techniques to optimize HACCM. The main idea of GA-based HACCM (GA-HACCM) is similar to PSO-HACCM, where the fitness function of GA is modified by implementing well-known performance measurement techniques based on the results of bit-score calculation. This is also to allow the most optimal interaction between GA and HACCM, and thus allow for wider search space exploration based on the modified parameters of the optimization process in Fig. 3.

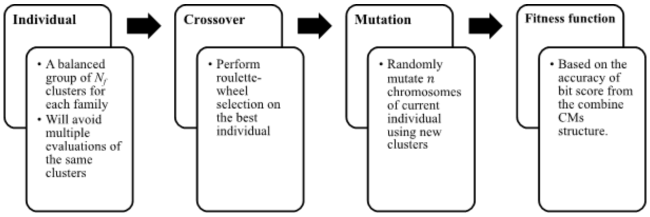
Individual
• A balanced group of N_f clusters for each family
• Will avoid multiple evaluations of the same clusters
Crossover
• Perform roulette-wheel selection on the best individual
Mutation
• Randomly mutate n chromosomes of current individual using new clusters
Fitness function
• Based on the accuracy of bit score from the combine CMs structure.

Fig. 3. Optimization Process using GA.

Similarly, there are multiple instances of HACCM executed concurrently; each of it is executed in GA individual. Each individual I_i will examine different set of CM family clusters, and thus produce unique results, this is because the examined CM family clusters set and its results are recorded, to prevent different individuals examine the same set multiple times. However, unlike PSO-HACCM, there are not much changes modification required to hybridize GA and HACCM. The algorithm of GA-HACCM is illustrated in Algorithm 5.

Algorithm 5 The Pseudocode to Construct Hybrid GA and HACCM

```
Start
Set number of individuals $N_i = 3$
Set number of iterations $N_t = 100$
for $i = 1...N_p$ do
    Generate individual $I_i$, which is the CCM
    Calculate initial sum of bit-scores of CCM as $X_i$
    Calculate fitness value $F_i$
    Select global best individual $G_{best}$ based on fitness value
end for
for $t = 1...N_t$ do
    for $i = 1...N_i$ where $I_i! = G_{best}$ do
        if Randomly selected for crossover then
            Crossover with $G_{best}$ using roulette-wheel selection
            Randomly mutate $n$ chromosomes of $I_i$ using new clusters from pool
            Generate hierarchical CCM from new clusters
            Update $X_i$ and $F_i$
        end if
    end for
    Select new Gbest
    if $F_{G_{best}} == 1$ OR is stagnant after 5 iterations then
        Stop iteration
    end if
end for
End
```

IV. Experimental Setup

With the goal stated in the section above, an extensive and rigorous empirical comparative study is designed and conducted. In this section, a detailed description of the experimental method is provided.

A. Dataset Collection and Preparation

This study obtained the dataset from Rfam database, the most commonly used database that store ncRNA gene

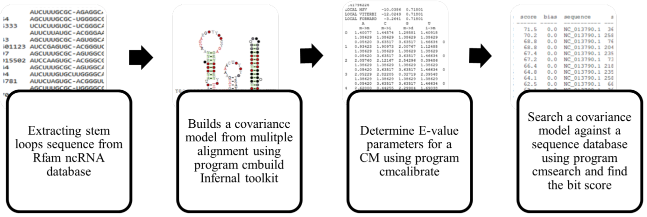
Extracting stem loops sequence from Rfam ncRNA database
Builds a covariance model from mulitple alignment using program cmbuild Infernal toolkit
Determine E-value parameters for a CM using program cmcalibrate
Search a covariance model against a sequence database using program cmsearch and find the bit score

Fig. 4. Steps of Data Collection and Preparation.

TABLE I. Summary of Training and Testing Dataset Preparation

Family	Number of Sequences	Number of Selected Sequences	Number of Unselected Sequences
RF00007	51	51	0
RF00015	137	51	86
RF00020	131	51	80
RF00029	86	51	35
RF00050	144	51	93

families from various species [30]. This database is a collection of ncRNA families represented by manually curated sequence alignments, consensus secondary structures and annotation gathered from corresponding taxonomy and ontology resources. The summary of the process of collecting and preparing the dataset is as shown in Fig. 4.

In this study, five sets of ncRNA gene families were selected to test CMs combination method. The selected gene families have roughly similar average length so that their CM combination would not have bias towards either of them. Thus, the number of selected sequences will be set to the minimum number of sequences available, which is fifty-one sequences. This study proposes to randomly divide the selected sequences within one family into groups of three sequences for each gene families by following steps of dataset collection and preparation in Fig. 4 then generate a balanced cluster pool from selected sequences for each family, while the rest of the unselected sequences will be used to form the testing dataset and validate the proposed technique. The process to construct from the generic CCM using existing and proposed techniques is as shown in Algorithm 6, while the summary of the training and testing dataset is shown in Table I.

Algorithm 6 The Pseudocode for the Dataset Preparation

```
Start
Set number of families $N_f = 5$
Identify minimum number sequences $N_s$ for each family
Set number of sequences per cluster $N_{sc} = 3$
Set number of clusters per family $N_c = FLOOR(N_s/N_{sc})$
Generate cluster pool containing $N_c$ clusters by randomly selecting sequences for each family
Set unselected sequences as testing dataset
End
```

Each family obtained from Rfam database is stored in one plain-text file. Each file consists of multiple lines, where each line represents the known ncRNA sequence from that family. There are two data contained in each line separated with tab space: the first data is the name of the sequence, while the second data is the ncRNA sequence. The unique pattern of each family is stored in the last line of the plain-text file. Apart from the plain-text file, it is also possible to visualize the ncRNA sequence in the Rfam database.

After the ncRNA family dataset has been successfully obtained from Rfam database, the dataset must be prepared and processed so that it can be used by the tools in the Infernal package. Prior to processing by Infernal package tools, the dataset must be stored in .CM file format, which is constructed from .STO file format. The data format in .STO file format is almost similar to the format obtained from Rfam database, with only minor differences, such as the first line of the file in .STO file format must be annotated with *# STOCKHOLM 1.0* syntax and the gene family pattern of .STO file format is only limited to the dot-bracket notation, which consists of left-angle bracket (<), right-angle bracket (>), and dot symbol (.).

As mentioned earlier, the dataset must be stored in .CM file format. To generate .CM file format, *cmbuild* program from Infernal package must be used. After the .CM file is generated, it is also necessary to calibrate the data using *cmcalibrate* program so that the dataset can be used by other programs in Infernal package. However, calibrating the dataset to .CM format takes a rather long time; thus, this study needs to adjust the parameters of *cmcalibrate* program to reduce the calibration time. The *cmcalibrate* is executed without the forecasting capability, which estimates the execution time by running a small sample of data, reduced its tail loss probability to 10-2 instead of the default 10-15, and the sample size to 0.5 MB instead of the default 1.6 MB.

B. Experimental Design

The quality of the proposed technique must be validated using various performance measurement criteria, because evaluating the performance of learning algorithms is a fundamental aspect of machine learning. In this study, the performance measurements for evaluating the performance of the existing and proposed techniques are the sum of bit-scores, the processing time, and the value of confusion matrix-based performance measurement technique. Despite there are several confusion matrix-based performance measurement techniques available, such as sensitivity, specificity, precision, F_1-score, accuracy, prevalence, phi coefficient, Fowlkes–Mallows index, informedness, markedness, and a few others, this study will only focus on the accuracy as the primary performance measurement by comparing the most commonly used techniques. The sum of bit-scores and processing time are considered as supplementary performance measurements to support the results of the primary measurement.

The bit-score indicates the performance measurement for selected ncRNA sequence. The bit-score is obtained from the *cmsearch* program from Infernal package, and it is used to measure the probability of similar sequence to be found on a given set on covariance model gene family. The processing time can be measured directly while executing the existing and proposed techniques in the clean room environment.

The proposed ncRNA search model techniques are developed to construct generic sequence models which can be used to recognize new potential members of novel ncRNA

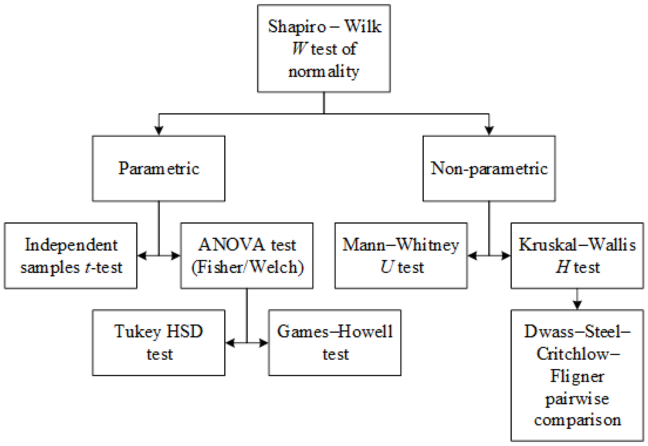

Fig. 5. Summary of Statistical Validation

families. Therefore, it is important to construct a performance measurement to assess this capability. However, since it is practically difficult to obtain the novel and unknown sequences of ncRNA to assess the existing and proposed techniques, this study instead proposed a scenario to simulate this situation.

The scenario of identifying the unknown ncRNA sequences is similar to identifying a set of testing instances by using supervised learning method. By using the datasets mentioned earlier, a portion of the data from the datasets must be set aside to form the testing dataset and the remaining data will be used to form the training dataset, which will be used to construct the CCM. After the CCM has been completely constructed, the testing dataset will be used to validate the CCM, where the result is either the sequence can be found or not found in the combined covariance model. The result will be used to validate the generic sequence models.

The performance measurements are conducted fifty times to ensure the statistical consistency of the results using the datasets mentioned earlier. These performance measurements are validated using various statistical validation techniques available in jamovi statistical package software. The summary of the statistical validation is depicted in Fig. 5.

V. Results and Discussion

This section provides the results of the comparative study used to identify the most suitable technique to improve the quality of original HACCM proposed by [5]. Table II presents the descriptive test results for the accuracies and the sum of bit-scores of the best member from 50 executions using 3 members for each technique, as well as the total processing time to complete these 50 executions.

Based on the results shown in Table II, the PSO-HACCM produces the best accuracy, sum of bit-scores, and processing time compared to the original HACCM and GA-HACCM. The percentage of the differences between the PSO-HACCM compared to the GA-HACCM and original HACCM is shown in Table III.

The PSO-HACCM results shown in Table III are higher than other techniques are due to its cognitive and social accelerations capabilities, which are not present in the original HACCM and GA-HACCM. However, to validate whether the difference between PSO-HACCM, GA-HACCM, and original HACCM is statistically significant, in-depth comparison of these techniques must be conducted. The results summarized in Table II are then validated to determine the significance of PSO-HACCM and GA-HACCM compared to original HACCM using either ANOVA or Kruskal–Wallis H test, because it is suitable for comparing the means from more than two distinct groups of data. However, before either ANOVA or Kruskal–Wallis H test is conducted, all techniques must be tested for normality to determine whether its data is normally distributed. Table IV presents the result of Shapiro–Wilk W tests of normality.

Table IV presents the results from Shapiro–Wilk W test of normality. It is concluded that the accuracies of PSO-HACCM,

TABLE II. Descriptive Test Results

Descriptive	Technique	Accuracy	Sum of Bit-Score	Processing Time
Valid cases	HACCM PSO-HACCM GA-HACCM	50 50 50	50 50 50	50 50 50
Missing cases	HACCM PSO-HACCM GA-HACCM	0 0 0	0 0 0	0 0 0
Mean	HACCM PSO-HACCM GA-HACCM	82.8% 91.0% 86.2%	316 359 330	391 357 392
Median	HACCM PSO-HACCM GA-HACCM	80.0% 90.0% 85.0%	313 362 325	391 355 391
Mode	HACCM PSO-HACCM GA-HACCM	80.0% 95.0% 85.0%	302 306 243	328 314 310
Standard deviation	HACCM PSO-HACCM GA-HACCM	4.97% 4.95% 5.11%	37.4 45.8 37.7	31.4 21.1 42.2
Variance	HACCM PSO-HACCM GA-HACCM	0.247% 0.245% 0.261%	1400 2098 1422	987 447 1780
Range	HACCM PSO-HACCM GA-HACCM	25.0% 15.0% 25.0%	178 173 200	148 109 229
Skewness	HACCM PSO-HACCM GA-HACCM	1.26 -0.0263 0.329	0.177 -0.224 0.394	0.142 0.847 0.735
Kurtosis	HACCM PSO-HACCM GA-HACCM	2.65 -1.38 0.544	0.126 -0.546 0.867	-0.151 1.49 1.76

TABLE III. Differences between Techniques

Compared Technique	Accuracy		Sum of Bit-Score		Processing Time	
	Value	%	Value	%	Value	%
HACCM	4.8%	10.2% higher	43	13.6% higher	34	8.7% faster
GA-HACCM	8.2%	5.6% higher	29	8.8% higher	35	8.9% faster

TABLE IV. Preliminary Shapiro–Wilk W Tests of Normality

Criteria	Technique	Statistic	*df*	Sig.
Accuracy	HACCM PSO-HACCM GA-HACCM	0.823 0.816 0.897	50 50 50	< 0.001 < 0.001 < 0.001
Sum of bit-scores	HACCM PSO-HACCM GA-HACCM	0.975 0.966 0.979	50 50 50	0.364 0.159 0.522
Processing time	HACCM PSO-HACCM GA-HACCM	0.981 0.957 0.959	50 50 50	0.574 0.064 0.083

TABLE V. KRUSKAL–WALLIS H TEST RESULTS FOR THE ACCURACY

Chi-Square	df	Asymp. Sig.
50.9	2	< 0.001

TABLE VI. POST-HOC TEST RESULTS FOR THE ACCURACY

Compared Measurements	W	p
HACCM vs PSO-HACCM	9.42	< 0.001
HACCM vs GA-HACCM	5.29	< 0.001
PSO-HACCM vs GA-HACCM	-6.07	< 0.001

TABLE VII. SHAPIRO–WILK W TESTS OF NORMALITY FOR ASSUMPTION CHECKS OF THE SUM OF BIT-SCORES AND PROCESSING TIME

Criteria	W	Sig.
Sum of bit-scores	0.993	0.655
Processing time	0.972	0.004

TABLE VIII. TEST OF HOMOGENEITY OF VARIANCES OF THE SUM OF BIT-SCORES AND PROCESSING TIME

Criteria	Levene Statistic	$df1$	$df2$	Sig.
Sum of bit-scores	1.87	2	147	0.157
Processing time	9.80	2	147	< 0.001

GA-HACCM, and original HACCM techniques are not normally distributed since $p < 0.05$, therefore non-parametric test such as Kruskal–Wallis H test should be conducted instead. On the other hand, the sum of bit-scores and processing time criteria for these techniques are normally distributed, and thus the ANOVA can be conducted for testing the score of these techniques.

The Kruskal–Wallis H test results for the accuracy of PSO-HACCM, GA-HACCM, and original HACCM are shown in Table V, with the post-hoc results using Dwass–Steel–Critchlow–Fligner pairwise comparison is shown in Table VI. Meanwhile, the assumption checks and test of homogeneity of variances for the sum of bit-scores and processing time of PSO-HACCM, GA-HACCM, and original HACCM are shown in Tables VII and VIII, respectively.

Based on the results shown in Table V, there are a statistically significant effect of three techniques $[H(2) = 50.9, p < 0.001]$. Post-hoc comparisons using Dwass–Steel–Critchlow–Fligner pairwise comparison on each pair of groups which is shown in Table VI indicated that there is a statistically significant difference between the accuracy of HACCM and PSO-HACCM $(W = 9.42, p < 0.001)$, HACCM and GA-HACCM $(W = 5.29, p < 0.001)$, and PSO-HACCM and GA-HACCM $(W = -6.07, p < 0.001)$.

On the other hand, based on the results shown in Tables VII and VIII, the sum of bit-scores is normally distributed and there is homogeneity of variances for the sum of bit-scores between groups of techniques, therefore the assumption of ANOVA has been validated, and thus, the Fisher's test must be conducted and Tukey HSD post-hoc tests must be used consequently. On the contrary, the processing time is not normally distributed and there are no there are homogeneity of variances for the processing time between groups of techniques, therefore the assumption of ANOVA has been violated, and the Kruskal–Wallis H test must be conducted instead.

TABLE IX. FISHER'S TEST ANOVA RESULTS OF THE SUM OF BIT-SCORES

F	$df1$	$df2$	Sig.
15.0	2	147	< 0.001

TABLE X. POST-HOC TEST RESULTS USING TUKEY HSD TEST FOR THE SUM OF BIT-SCORES

Compared Techniques	Mean Difference	t-value	df	Sig.
HACCM vs PSO-HACCM	-43.3	-5.35	147	< 0.001
HACCM vs GA-HACCM	-13.5	-.167	147	0.222
PSO-HACCM vs GA-HACCM	29.8	3.68	147	< 0.001

TABLE XI. KRUSKAL–WALLIS H TEST RESULTS FOR THE PROCESSING TIME

Chi-Square	df	Asymp. Sig.
35.3	2	< 0.001

TABLE XII. POST-HOC TEST RESULTS FOR THE PROCESSING TIME

Compared Measurements	W	p
HACCM vs PSO-HACCM	-7.72	< 0.001
HACCM vs GA-HACCM	-0.04	0.999
PSO-HACCM vs GA-HACCM	6.80	< 0.001

Thus, the ANOVA for the sum of bit-scores of PSO-HACCM, GA-HACCM, and original HACCM are shown in Table IX with the post-hoc results using Tukey HSD and Games–Howell are shown in Table X for the sum of bit-scores, while the Kruskal–Wallis H test results for the processing time of PSO-HACCM, GA-HACCM, and original HACCM are shown in Table XI, with the post-hoc results using Dwass–Steel–Critchlow–Fligner pairwise comparison is shown in Table XII.

Based on the results shown in Table IX, there is a statistically significant effect of the sum of bit-scores between PSO-HACCM, GA-HACCM, and original HACCM techniques $[F(2, 147) = 15.0, p < 0.001]$ at the $p < 0.05$ level. Post-hoc comparisons using the Tukey HSD test shown in Table X indicated that the mean score for the sum of bit-scores of PSO-HACCM is statistically significantly better than HACCM $[t(147) = -5.35, p < 0.001]$ and GA-HACCM $[t(147) = 3.68, p < 0.001]$, while there is no statistically significant difference between HACCM and GA-HACCM $[t(147) = -0.167, p = 0.222]$.

Meanwhile, based on the results shown in Table XI, there are a statistically significant effect of three techniques $[H(2) = 35.3, p < 0.001]$. Post-hoc comparisons using Dwass–Steel–Critchlow–Fligner pairwise comparison on each pair of groups which is shown in Table XII indicated that the mean score for the processing time of PSO-HACCM is statistically significantly better than HACCM $(W = -7.72, p < 0.001)$, and GA-HACCM $(W = 6.80, p < 0.001)$. These statistical test results of accuracy, sum of bit-scores, and processing time confirm that PSO-HACCM technique is indeed improving the performance of the original HACCM.

To further validate the performance of the proposed techniques, these techniques should be validated by simulating the identification of unknown ncRNA sequences using the combined covariance model constructed by these techniques.

TABLE XIII. Descriptive Test Results of the Identification Accuracy

Descriptive	Technique	Accuracy
Valid cases	HACCM	50
	PSO-HACCM	50
	GA-HACCM	50
Missing cases	HACCM	0
	PSO-HACCM	0
	GA-HACCM	0
Minimum	HACCM	40.0%
	PSO-HACCM	66.67%
	GA-HACCM	32.16%
Maximum	HACCM	99.61%
	PSO-HACCM	91.76%
	GA-HACCM	90.20%
Mean	HACCM	67.6%
	PSO-HACCM	76.6%
	GA-HACCM	66.%
Median	HACCM	67.4%
	PSO-HACCM	75.3%
	GA-HACCM	67.4%
Mode	HACCM	67.4%
	PSO-HACCM	73.7%
	GA-HACCM	60.4%
Standard deviation	HACCM	12.3%
	PSO-HACCM	6.21%
	GA-HACCM	11.9%
Variance	HACCM	1.52%
	PSO-HACCM	0.39%
	GA-HACCM	1.41%
Range	HACCM	59.6%
	PSO-HACCM	25.1%
	GA-HACCM	58.0%
Skewness	HACCM	0.0769
	PSO-HACCM	0.465
	GA-HACCM	-0.379
Kurtosis	HACCM	0.226
	PSO-HACCM	-0.658
	GA-HACCM	0.526

TABLE XIV. Preliminary Shapiro–Wilk W Tests of Normality of the Identification Accuracy

Technique	Statistic	df	Sig.
HACCM	0.989	50	0.917
PSO-HACCM	0.955	50	0.055
GA-HACCM	0.980	50	0.534

The identification will be using the testing dataset discussed in Section IV which has been set aside prior to the construction of the generic sequence models that simulates the identification of unknown ncRNA sequences. The result of this identification simulation is summarized in Table XIII.

Based on the summarized results in Table XIII, it can be seen that the mean accuracy of PSO-HACCM is higher compared to the GA-HACCM and original HACCM. This is due to PSO-HACCM successfully constructed the generic sequence models which can be used to recognize potential members of novel ncRNA families. To determine the significance of the PSO-HACCM results presented in Table XIII compared to GA-HACCM and original HACCM, another statistical validation using ANOVA or Kruskal–Wallis H test must be conducted. As mentioned earlier, all techniques must be tested for normality to determine whether its data is normally distributed. Table XIV presents the result of Shapiro–Wilk W tests of normality for these techniques.

Based on the results presented in Table XIV, it is concluded that the identification accuracies of PSO-HACCM, GA-HACCM, and original HACCM techniques are normally distributed since $p \geq 0.05$, and thus the ANOVA can be conducted for validating the identification accuracy of these techniques. The assumption checks and test of homogeneity of variances for the identification accuracy of PSO-HACCM, GA-HACCM, and original HACCM are shown in Tables XV and XVI, respectively.

TABLE XV. Shapiro–Wilk W Tests of Normality for Assumption Checks of the Identification Accuracy

W	Sig.
0.988	0.222

TABLE XVI. Test of Homogeneity of Variances Results of the Identification Accuracy

Levene Statistic	$df1$	$df2$	Sig.
6.82	2	147	0.001

TABLE XVII. ANOVA Results of the Identification Accuracy

Test	F	$df1$	$df2$	Sig.
Fisher's	13.6	2	147	< 0.001

TABLE XVIII. Post-Hoc Test Results of the Identification Accuracy

Compared Techniques	Mean Difference	t-value	df	Sig.
HACCM vs PSO-HACCM	-9.07	-4.31	147	< 0.001
HACCM vs GA-HACCM	0.784	0.373	147	0.926
PSO-HACCM vs GA-HACCM	9.851	4.685	147	< 0.001

Based on the results shown in Tables XV and XVI, the identification accuracy is normally distributed and there is homogeneity of variances for the identification accuracy between groups of techniques, therefore the assumption of ANOVA has been validated and the Fisher's test results must be considered and Tukey HSD post-hoc tests must be used consequently. The ANOVA for the identification accuracy of PSO-HACCM, GA-HACCM, and original HACCM is shown in Table XVII with the post-hoc results using Tukey HSD is shown in Table XVIII.

Based on the results shown in Table XVII, there is a statistically significant effect of the identification accuracy between PSO-HACCM, GA-HACCM, and original HACCM techniques $[F(2, 147) = 13.6, p < 0.001]$ at the $p < 0.05$ level. Post-hoc comparisons using the Tukey HSD test shown in Table XVIII indicated that the mean score for the identification accuracy of PSO-HACCM is statistically significantly better than HACCM $[t(147) = -4.31, p < 0.001]$ and GA-HACCM $[t(147) = 4.685, p < 0.001]$, while there is no statistically significant difference between HACCM and GA-HACCM $[t(147) = 0.373, p = 0.926]$.

From the comparative performance measurements and statistical validations conducted throughout this section, it can be concluded that PSO-HACCM is indeed performing better compared to the original HACCM as well as GA-HACCM, thanks to its generic sequence model construction.

VI. Conclusion

The comparison of accuracy, sum of bit-scores, and processing time have been conducted between original HACCM, PSO-HACCM, and GA-HACCM to demonstrate the capability of the proposed techniques in constructing generic sequence models, which can be used to recognize potential members of novel ncRNA families. Performance measurement results show that proposed PSO-HACCM performs better than the original HACCM and the proposed GA-HACCM technique, in terms of optimization performance, identification accuracy, sum of bit-scores, and processing time. This finding opens up the possibilities of future works, such as leveraging Graphical Processing Unit (GPU) to speed the computation process, explorations of other swarm intelligence techniques, and inclusion of more gene families for the dataset.

Acknowledgment

Authors would also wish to express their gratitude towards Skim Zamalah provided by Universiti Teknikal Malaysia Melaka (UTeM).

References

[1] C. Biology and T. B. Laboratories, *Introns and the RNA World*, RNA World, pp. 221–232, 1999.

[2] H. -H. Tseng, Z. Weinberg, J. Gore, R. R. Breaker, and W. L. Ruzzo, *Finding non-coding RNAs through genome-scale clustering*, J. Bioinform. Comput. Biol., vol. 7, no. 2, pp. 373–88, 2009.

[3] S. R. Eddy and R. Durbin, *RNA sequence analysis using covariance models*, Nucleic Acids Res., vol. 22, no. 11, pp. 2079–2088, 1994.

[4] S. F. Smith, *Covariance searches for ncRNA gene finding*, Proc. 2006 IEEE Symp. Comput. Intell. Bioinforma. Comput. Biol. CIBCB'06, pp. 320–326, 2006.

[5] W. Jiang and K. C. Wiese, *Combined covariance model for non-coding RNA gene finding*, IEEE SSCI 2011 - Symp. Ser. Comput. Intell. - CIBCB 2011 2011 IEEE Symp. Comput. Intell. Bioinforma. Comput. Biol., pp. 22–26, 2011.

[6] S. Will, K. Reiche, I. L. Hofacker, P. F. Stadler, and R. Backofen, *Inferring noncoding RNA families and classes by means of genome-scale structure-based clustering*, PLoS Comput. Biol., vol. 3, no. 4, pp. 680–691, 2007.

[7] Y. Saito, K. Sato, and Y. Sakakibara, *Fast and accurate clustering of noncoding RNAs using ensembles of sequence alignments and secondary structures*, BMC Bioinformatics, vol. 12 Suppl 1, p. S48, 2011.

[8] T. Hermann and E. Westhof, *Non-Watson-Crick base pairs in RNA-protein recognition*, Chemistry and Biology, vol. 6, no. 12. 1999.

[9] A. MacHado-Lima, H. A. Del Portillo, and A. M. Durham, *Computational methods in noncoding RNA research*, J. Math. Biol., vol. 56, no. 1–2, pp. 15–49, 2008.

[10] S. Zhang, I. Borovok, Y. Aharonowitz, R. Sharan, and V. Bafna, *A sequence-based filtering method for ncRNA identification and its application to searching for riboswitch elements*, Bioinformatics, vol. 22, no. 14, pp. 1–11, 2006.

[11] S. E. Butcher and A. M. Pyle, *The molecular interactions that stabilize RNA tertiary structure: RNA motifs, patterns, and networks*, Acc. Chem. Res., vol. 44, no. 12, pp. 1302–1311, 2011.

[12] S. Crowder, J. Holton, and T. Alber, *Covariance analysis of RNA recognition motifs identifies functionally linked amino acids*, J. Mol. Biol., vol. 310, no. 4, pp. 793–800, 2001.

[13] Z. Yao, Z. Weinberg, and W. L. Ruzzo, *CMfinder - A covariance model based RNA motif finding algorithm*, Bioinformatics, vol. 22, no. 4, pp. 445–452, 2006.

[14] S. R. Eddy, *A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure*, BMC Bioinformatics, vol. 3, p. 18, 2002.

[15] S. C. Johnson, *Hierarchical clustering schemes*, Psychometrika, vol. 32, no. 3, pp. 241–254, 1967.

[16] S. Alam, G. Dobbie, P. Riddle, and M. A. Naeem, *Particle Swarm Optimization Based Hierarchical Agglomerative Clustering*, Web Intell. Intell. Agent Technol. (WI-IAT), 2010 IEEE/WIC/ACM Int. Conf., vol. 2, pp. 64–68, 2010.

[17] G. Nowak and R. Tibshirani, *Complementary hierarchical clustering*, Biostatistics, vol. 9, no. 3, pp. 467–483, 2008.

[18] J. A. Smith, *RNA search with decision trees and partial covariance models*, IEEE/ACM Trans. Comput. Biol. Bioinforma., vol. 6, no. 3, pp. 517–527, 2009.

[19] F. Murtagh and P. Contreras, *Methods of Hierarchical Clustering*, Computer (Long. Beach. Calif)., vol. 38, no. 2, pp. 1–21, 2011.

[20] S. Savaresi, D. Boley, S. Bittanti, and G. Gazzaniga,*Cluster Selection in Divisive Clustering Algorithms*, 2nd SIAM International Conference on Data Mining, 2002.

[21] M. L. Zepeda-Mendoza and O. Resendis-Antonio, *Hierarchical Agglomerative Clustering*, Encyclopedia of Systems Biology, Springer, New York, NY, 2013.

[22] J. Augen, *Bioinformatics and Transcription*, in Bioinformatics in the Post-Genomic Era: Genome, Transcriptome, Proteome, and Information-Based Medicine, 2005, p. 408.

[23] S. Wang, S. Hou, J. Wu, and J. Wei, *Clustering of ncRNA based on structural and semantic similarity*, J. Bionanoscience, vol. 7, no. 1, pp. 20–25, 2013.

[24] D. Li et al., *Experimental RNomics and genomic comparative analysis reveal a large group of species-specific small non-message RNAs in the silkworm Bombyx mori*, Nucleic Acids Res., vol. 39, no. 9, pp. 3792–3805, 2011.

[25] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy, *Rfam: An RNA family database*, Nucleic Acids Research, vol. 31, no. 1. pp. 439–441, 2003.

[26] R. C. Eberhart and J. Kennedy, *A New Optimizer using Particle Swarm Theory*, In: Proceedings of 6th International Symposium on Micro Machine and Human Science, 1995.

[27] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proceedings of IEEE International Conference on Neural Networks, pp. 1942–1948 vol. 4, 1995.

[28] M. Geis and M. Middendorf, *Particle swarm optimization for finding RNA secondary structures*, International Journal of Intelligent Computing and Cybernetics, vol. 4 no. 2, pp. 160-186, 2011.

[29] R. C. Eberhart and Y. Shi, *Comparing inertia weights and constriction factors in particle swarm optimization*, Proceedings of the 2000 Congress on Evolutionary Computation, vol. 1, pp. 84-88, 2000.

[30] I. Kalvari, E. P. Nawrocki, N. Ontiveros-Palacios, J. Argasinska, K. Lamkiewicz, M. Marz, S. Griffiths-Jones, C. Toffano-Nioche, D. Gautheret, Z. Weinberg, E. Rivas, S. R. Eddy, R. D. Finn, A. Bateman, and A. I. Petrov, *Rfam 14: expanded coverage of metagenomic, viral and microRNA families*, Nucleic Acids Research, vol. 49(D1), pp. D192–D200, 2021.

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