Published in final edited form as:

IEEE/ACM Trans Comput Biol Bioinform. 2015 ; 12(3): 551–559. doi:10.1109/TCBB.2014.2366758.

A Machine Learning Approach for Accurate Annotation of Noncoding RNAs

Yinglei Song,

School of Electronics and Information Science, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, China

Chunmei Liu, and

Department of Systems and Computer Sciences, Howard University, Washington D.C. 20059, U.S.A

Zhi Wang

School of Computer Science and Engineering, Jiangsu University of Science and Technology, Zhenjiang, Jiangsu 212003, U.S.A

Yinglei Song: yingleisong@gmail.com; Chunmei Liu: chunmei@scs.howard.edu; Zhi Wang: cw@just.edu.cn

Abstract

Searching genomes to locate noncoding RNA genes with known secondary structure is an important problem in bioinformatics. In general, the secondary structure of a searched noncoding RNA is defined with a structure model constructed from the structural alignment of a set of sequences from its family. Computing the optimal alignment between a sequence and a structure model is the core part of an algorithm that can search genomes for noncoding RNAs. In practice, a single structure model may not be sufficient to capture all crucial features important for a noncoding RNA family. In this paper, we develop a novel machine learning approach that can efficiently search genomes for noncoding RNAs with high accuracy. During the search procedure, a sequence segment in the searched genome sequence is processed and a feature vector is extracted to represent it. Based on the feature vector, a classifier is used to determine whether the sequence segment is the searched ncRNA or not. Our testing results show that this approach is able to efficiently capture crucial features of a noncoding RNA family. Compared with existing search tools, it significantly improves the accuracy of genome annotation.

Keywords

noncoding RNAs; secondary structure; genome annotation; feature vector; classifier

1 Introduction

Noncoding RNAs (ncRNAs) are crucial for a variety of biological processes including RNA modification, gene regulation, and chromosome replication [19, 29]. In particular, recent research in molecular biology has revealed that ncRNAs such as microRNAs are associated

with the development of certain types of cancer [15, 21]. The biological functions of many ncRNAs are determined by their secondary structures. Recently, due to the large amount of available genome data, computational approaches have been developed to search genomes to identify ncRNA genes [5, 9–11, 16–18, 22–26, 28, 30]. Most of these approaches use a structural model to describe the secondary structure of a searched ncRNA family and use sequence-structure alignment to determine whether a given sequence segment belongs to it or not.

Most software tools that can search genomes for ncRNAs use a Covariance Model (CM) [5] to model the secondary structure of an ncRNA family. Similar to a Hidden Markov Model (HMM), a CM uses the statistical profiles of all single and paired positions in the sequences to describe both the primary sequence and the secondary structure of an ncRNA family. The sequence-structure alignment between a sequence and a structure model can be computed with a dynamic programming algorithm [5]. Based on the statistical profiles of single and paired positions, the algorithm can find the alignment of the maximum probability. The statistical significance of this probability is then used to determine whether the sequence is a member of the family or not. CM based sequence-structure alignment has been successfully used to identify a variety of ncRNAs in the genomes of many species [13, 18, 20].

In theory, the computation time needed to optimally align a sequence segment of length N to a structure model that contains $O(N)$ states is $O(lN)^3$, where l is the number of bifurcation rules in the model. Since a genome sequence usually contains at least 10^6 nucleotides, the computational efficiency of sequence-structure alignment becomes an important issue when the searched structure contains more than 300 nucleotides. To improve the computational efficiency for searching long genomes or large sequence databases, a preprocessing step can be used to remove portions of a genome that are unlikely to contain the desired pattern [1, 10, 12, 28]. In [22], an approach based on partial covariance models is developed for ncRNA search. A binary decision-tree is constructed to determine the order to apply the partial models and the score thresholds associated with these models. Recently, Infernal combines a pipeline of filtering approaches with a search space reduction technique to speed up the search procedure. These filtration based techniques can significantly reduce the search time but the accuracy of the search might also be adversely affected.

Structator is an index-based search tool that can elegantly and efficiently match RNA sequence-structure patterns with affix arrays [6]. However, it does not fully utilize the statistical information of individual or paired positions in the secondary structure of a searched family and thus may miss important homologs.

Our previous work developed a new graph theoretic approach to model the secondary structure of noncoding RNAs [23, 25, 26]. This approach uses a conformational graph to represent the secondary structure of an ncRNA family and an image graph to represent a sequence. The alignment between a sequence and a structure can be computed by solving a maximum valued subgraph isomorphism problem. Based on a tree decomposition of the conformational graph, the problem can be efficiently solved with a dynamic programming based approach in time $O(k^tN)$, where t is the tree width of the tree decomposition and k is a small integer parameter that is usually at most 7 [23, 26]. This approach can perform the

sequence-structure alignment in linear time since the tree widths of most conformational graphs are small integers. However, the construction of the image graph is based on the assumption that each stem must be in a restricted region in the sequence, which may not be the case when certain structural units are missing in some sequences of the family. In addition, the exponential term in the time complexity of the algorithm may become a large factor when the tree width t or the parameter k is a large integer.

Recent work has shown that some highly conserved regions in the secondary structure of an ncRNA family might be important for its biological functions [8]. Recognizing these regions during the search thus may significantly improve the search accuracy. However, a CM based search usually uses the overall alignment score between a sequence segment and a structure profile as the basis for decision and thus may ignore the contributions from such structural units. Although a few filtering based approaches have been developed to integrate some of these structural units into their filters, a systematic method that can evaluate the relative importance of these structure units to achieve maximum search accuracy is still unavailable. A software tool that is able to recognize these structural units and correctly estimates their contributions to the overall probability that a sequence segment belongs to the searched family thus may significantly improve the search accuracy.

While previous approaches have developed accurate structural models for search, they did not consider the relative importance of nucleotides and base pairs with different levels of conservation for the biological functions of an ncRNA family. Our main goal in this paper is to combine previous approaches with the vital biological information of nucleotides and base pairs that are preserved with different levels of conservation during evolution and test the performance of such a classifier system against previous approaches.

In this paper, we develop a novel machine learning approach that can efficiently determine whether a sequence segment belongs to an ncRNA family or not. We develop a new approach that can extract a large number of features to represent a sequence segment in $O(N^3)$ time. To reduce the computation time needed for feature extraction, we partition the stems in a secondary structure into a few disjoint sets such that each set only contains a set of nested stems. To recognize the structure units that have different levels of conservation in a family, a set of CMs are constructed to represent the stems in each set. Specifically, for a particular CM in the set, only matching states in positions whose information content falls within a certain interval generate nucleotides. In other words, only the sequence content of positions where the degree of conservation in sequence content falls within a certain interval is modeled in a particular CM in the set.

We then align the sequence segment to each CM we have created, an alignment score can be obtained for each CM. All these alignment scores collectively form a feature vector that represents the sequence segment. To determine the relative importance of each component of a feature vector and remove the redundancy introduced during the alignments, we use a linear discriminant analysis (LDA) based approach to reduce the dimensionality of a feature vector. As the last step, a Bayesian classifier is used to determine the relationship between the sequence segment and the family.

Based on this approach, the features of all types of ncRNAs including those that contain pseudoknots can be accurately computed in time $O(N^3)$, where N is the length of the searched ncRNA. Moreover, the time complexity can be further reduced to $O(N^2)$ if the search space reduction technique used in Infernal is applied to the sequence-structure alignments for feature extraction. We have implemented this approach into a computer program RNASC and compared its search accuracy and computational efficiency with a few other search tools, including Infernal 1.1 [17], Structator [6] and RNATOPS [9]. Our testing results show that, on average, RNASC is effective for capturing important features of ncRNA families and is able to outperform existing tools in search accuracy.

2 Approach

2.1 Problem Formulation

Given the structural alignment of a set of sequences in an ncRNA family, the goal of the problem is to determine the location of an ncRNA from the family in a genome. Most of the existing approaches perform the search by aligning each sequence segment to a structural model that describes the secondary structure of the family, the alignment score is then used to determine whether the sequence segment is a member of the family or not. A structure model contains the statistical information of each position or pair of positions that form a base pair in the structure.

We solve the ncRNA search problem with a machine learning approach. Specifically, a sequence segment is represented by a number of features. A classifier then processes these features and labels the sequence segment as “in the family” or “not in the family”. The features are obtained by aligning the sequence segment to a set of CMs. Each CM in the set models the sequence content with a certain degree of conservation in a set of nested stems in the secondary structure.

To obtain the sets of nested stems modeled by the CMs, we partition the stems in the secondary structure of an ncRNA into disjoint sets of nested stems. For a secondary structure that contains s disjoint sets of nested stems, if k levels of conservation are selected to construct CMs, a feature vector contains ks components. The number of components may become a large number when k or s is large. In addition, the sequence content modeled by some of the CMs may overlap and the corresponding components thus contain redundant information.

We use an LDA based approach to remove the redundancy contained in a feature vector and reduce its dimensionality. A Bayesian classifier is then used to determine whether the sequence segment is a member of the family or not. The classifier needs to be trained with some training data sets. The training data sets can be obtained by applying the feature extraction procedure to the sequences in the family and a number of randomly generated sequences that are certainly not in the family. A Bayesian classifier computes the posterior probability for a sequence segment to be a member of the family and labels it based on the computed posterior probability.

A pseudoknot structure contains at least two crossing stems and cannot be modeled with a single CM. By partitioning the stems in the secondary structure of an ncRNA family into a few disjoint sets of nested stems, this approach can be easily applied to efficiently model and search for pseudoknot structures.

2.2 Partition the Secondary Structure

We construct a directed graph to model the secondary structure of an ncRNA. Each stem in the secondary structure is represented by a single vertex in the graph and two vertices are joined with a directed edge if one of the corresponding stems is contained by the other one in the secondary structure. For instance, if u and v are the two vertices that represent two stems S_u and S_v and S_u contains S_v in the secondary structure, a directed edge with a direction from u and v is created to join u and v .

To partition the stems in the secondary structure into disjoint sets of nested stems, we arbitrarily choose a vertex s_1 that has an in-degree of zero and mark s_1 as selected, we then check whether there is an edge that points from s_1 to an unselected vertex s_2 or not, if it is the case we mark s_2 as selected and repeat the procedure until no more vertices can be selected, otherwise the procedure is terminated. All selected vertices are included in one set and output by the algorithm. Those selected vertices are then removed from the graph and the above procedure is repeated until the graph becomes empty. It is not difficult to see that the sets output by the algorithm are disjoint and each set contains vertices that correspond to nested stems. Figure 1 provides an example to illustrate the procedure. Specifically, Figure 1 a) shows stems a, b and c in the secondary structure of a ncRNA family; Figure 1 b) shows the directed graph constructed to represent the relationships among stems a, b and c; Figure 3 c) shows the two disjoint sets of nested stems obtained from the directed graph as the result of partition.

2.3 Feature Extraction

The information content of each single unpaired column in the structural alignment is computed as I_c as follows

$$I_c = -\sum_{i=1}^5 p_i \log_2 p_i \quad (1)$$

where p_i is the probability for each nucleotide and a gap to appear in the column. For a given structural alignment, we use the frequency of each nucleotide and gap in the column to approximate p_i . Specifically, p_i can be computed as shown in equation (2).

$$p_i \approx \frac{C_i}{\sum_{i=1}^5 C_i} \quad (2)$$

where C_i is the number of times that nucleotide i ($i>1$) or gap ($i=1$) appears in the column. Similarly, for a pair of positions that form a base pair, the information content I_p of the pair is computed as follows.

$$I_p = \sum_{i=1}^5 \sum_{j=1}^5 p_{ij} \log_2 p_{ij} \quad (3)$$

where p_{ij} is the probability for each possible combination of nucleotides and gaps to appear in the position pair. For a given pair of positions that form a base pair in the structural alignment, we use the frequency of each pair of nucleotides or gaps to approximate the value of, as shown in equation (4).

$$p_{ij} \approx \frac{C_{ij}}{\sum_{i=1}^5 \sum_{j=1}^5 C_{ij}} \quad (4)$$

where C_{ij} is the number of times that pair (i, j) appears in the position pair.

The information content is a value that suggests how well the sequence content in a column or pair of columns is conserved. Biologically, those columns or pairs of columns that contain highly conserved sequence content is more likely to be crucial for the biological function of the ncRNA than those that are less conserved in sequence content [8]. Based on this criterion, we can select a few values within the range of possible values of the information content and use these values to construct a set of CMs.

A classical CM contains three types of states. They are matching states, insertion states and deletion states. A *matching state* is able to emit a single nucleotide or a base pair with certain probability. Matching states generate the sequence content that appears in most of the sequences in a family. By contrast, although an *insertion state* is also able to emit a single nucleotide or a base pair with certain probability, it is responsible for generating the sequence content that only appears in a small fraction of the sequences in a family. A deletion state is unable to emit any nucleotide and is responsible for modeling the gaps in the structural alignment of the sequences in a family. A column or a pair of columns in the structural alignment can be modeled by combining a deletion state with a matching state or an insertion state. In addition, transitions only occur between states for columns or pairs of columns that are consecutive in the alignment. Figure 2 a) shows four consecutive columns in a structural alignment and Figure 2 b) illustrates the states in a CM constructed based on the structural alignment in Figure 2 a), where M denotes a matching state, I represents an insertion state and D represents a deletion state.

We change the definition of a traditional CM to include a fourth type of states, *intermediate states*. An intermediate state generates a single nucleotide of any type or a pair of any combination of nucleotides and gaps with a probability of 1.0. In other words, an intermediate state “occupies” a position or a pair of positions in an alignment but does not check the content in the position or the pair of positions.

We then divide the interval $[I_{\min}, I_{\max}]$ of all possible values of information content into k intervals of equal length. The intermediate values $c_1, c_2, \dots, c_{k-1}$, are used as thresholds to select columns to be included in a CM. In particular, we use $M_0, M_1, \dots, M_{k-1}$ to denote the k CMs generated based on $c_0 = I_{\min}, c_1, c_2, \dots, c_{k-1}$. A column is included in M_i ($0 \leq i \leq k-1$) as a matching state or an insertion state if the information content of the column is between c_i and c_{i+1} otherwise it is included in M_i as an intermediate state.

It is not difficult to see that M_0 contains the statistical information of sequence content of the columns that are least conserved in an alignment, while M_{k-1} only contains the statistical information of those that are highly conserved, M_{k-1} thus contains the columns that are most important for the biological functions of the ncRNA. By aligning a sequence segment to the k CMs, we are able to obtain k different alignment scores. These alignment scores reflect the fitness of the sequence segment to structural units of different levels of conservation. Those fitness values together should determine whether the sequence segment is a member of the family or not.

In theory, the alignment score between a sequence segment and a CM is the maximum probability for the CM to generate the sequence segment. However, since the probability is usually a small value, the logarithm of the probability is computed in practice to represent the probability itself. In addition, since the alignment score depends on the length of the sequence segment, which may adversely affect the accuracy of the search, we use log-odd scores as the alignment score, the log-odd score is computed by subtracting the logarithm of the probability to randomly generate the sequence segment from that of the maximum probability for a CM to generate it.

The optimal alignment between a sequence segment and a CM can be computed with a CYK-style dynamic programming algorithm. Since each CM only contains nested stems, the dynamic programming needs only $O(N^3)$ computation time, where N is the number of nucleotides in the sequence segment. We use l to denote the number of sets of nested stems in the secondary structure, since k CMs are constructed from each set of nested stems, the total number of features we can extract from the sequence segment is kl and the time needed to compute all these features is $O(klN^3)$.

2.4 Dimensionality Reduction

The features generated from a sequence segment may contain redundant information, the accuracy of classification can be adversely affected by the redundancy in the features. We thus need an approach to remove as much redundancy as possible from the features while retaining the information crucial for the classification.

We use Linear Discriminant Analysis (LDA) to reduce the dimensionality of the feature vector we have obtained. Specifically, LDA computes a $lk \times r$ matrix W the following ratio ρ can be maximized

$$\rho = \text{tr}((W^T S_w W)^{-1} (W^T S_b W)) \quad (5)$$

where $tr(.)$ denotes the trace of a matrix, S_w is the *within-class scatter matrix* and S_b is the *between-class scatter matrix*. S_w and S_b can be computed as shown in equations (6) and (7).

$$S_w = \sum_{i=1}^N \sum_{j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T + C\lambda I \quad (6)$$

$$S_b = \sum_{i=1}^N m_i (\mu - \mu_i)(\mu - \mu_i)^T + \lambda I \quad (7)$$

where N is the number of classes in the training data set, C_i is the i th class, m_i is the number of data points in class C_i , x_j is the j th data point in class C_i , μ_i is the mean of all data points in class C_i and μ is the mean of all data points in the training set. C and λ , in equations (2) and (3) represent two regularization constants that increase the rank of S_b and I is the identity matrix. The value of λ , is set to be 0.1 and C is computed as follows.

$$C = \frac{tr(\sum_{i=1}^N \sum_{j \in C_i} (x_j - \mu_i)(x_j - \mu_i)^T)}{tr(\sum_{i=1}^N m_i (\mu - \mu_i)(\mu - \mu_i)^T)} \quad (8)$$

It has been shown that [7], when S_w is nonsingular, the right hand side of equation (5) can be maximized by solving the following eigenvalue problem

$$S_w^{-1} S_b p = \lambda p \quad (9)$$

where λ , is a constant, p is an eigenvector that needs to be determined.

The eigenvectors that corresponds to the r largest eigenvalues of the matrix $S_w^{-1} S_b$ are selected to form W . The dimensionality of a feature vector f can be reduced by applying a linear transformation to f as shown in equation (10).

$$s = W^T f \quad (10)$$

Figure 3 shows examples of two-dimensional feature vectors obtained from sequences in two ncRNA families, *SNORA73* and *Entero_OriR*, and those that are not in the families. Specifically, Figure 3 a) shows the feature vectors for sequences that are in *SNORA73* and these that are not in it. Figure 3 b) shows the feature vectors for sequences in *Entero_OriR* and these that are not in it.

2.5 Classification

Given a feature vector that represents a sequence segment, a Bayesian classifier can be used to determine whether it is a member of the searched family or not. We assume that there are

two classes in a training data set. One class contains the feature vectors obtained from the sequences in the family. These vectors are labeled as being in the family. The other one contains the features vectors obtained from some randomly generated sequences, and these feature vectors are labeled as not being in the family. In addition, we assume the Euclidean distance of a feature vector to the mean of its class follows Gaussian distribution. Based on this assumption, we can compute the posterior probability using Bayesian Theorem as follows.

$$P(x \in C_i|x) = \frac{P(x, C_i)}{P(x)} = \frac{P(x|C_i)P(C_i)}{\sum_{t=1}^2 P(x|C_t)P(C_t)} = \frac{P(x|C_i)}{\sum_{t=1}^2 P(x|C_t)} \quad (11)$$

where the last equality is based on the assumption that $P(C_1)$ is equal to $P(C_2)$. The values of $P(x|C_1)$ and $P(x|C_2)$ can be computed from x as shown in equation (12).

$$p(x|C_i) = \frac{1}{\sqrt{2\pi}\sigma_i} \exp\left(-\frac{|x-\mu_i|^2}{\sigma_i^2}\right) \quad (12)$$

where σ_i is the standard deviation of the Euclidean distances of the feature vectors in class C_i to their mean μ_i .

2.6 Generating Training Data Sets

The training data set is generated by randomly generating a number of sequences and extract features from them. The feature vectors extracted from them are then labeled as not being in the family. The sequences in the ncRNA family are processed by the same feature extraction procedure and the features extracted from them are labeled as being in the family. To guarantee that our approach is able to obtain the features that are most important for sequences in the searched family, the random sequences are generated with the same base composition as these that are in the family and are of the same lengths as them.

3 Testing Results

3.1 The Recognition Ability

We evaluate the recognition ability of our approach on all of the 2208 noncoding RNA families available in the Rfam 11.0 database. From the database, we are able to obtain the sequence content and structural information of the sequences in these families. The database can be accessed online at <http://rfam.xfam.org/> [4]. For each family, we download the structural alignment of a number of seed sequences and construct the set of CMs that describe the family based on the alignment.

We then apply the steps of feature extraction we have described to process the structural alignment of seed sequences in each family. To obtain feature vectors for sequences that are not in a family, we randomly generate sequences based on the base composition of all sequences in the family, these sequences are of the same lengths as those in the family. Feature vectors that are not in the family can be obtained by processing these randomly

generated sequences with the same feature extraction procedure. We vary the value of parameter k . For each given value of k , we measure the ratio of between-class and within-class distances and the computation time needed to obtain the feature vector for each family. Table 1 shows the means and standard deviations of the ratio and the computation time for all 2208 families. From the table, it is clear that the recognition ability and computational efficiency of the approach can be well balanced when the parameter k is 4. The parameter k is thus set to be 4 in all of our experiments.

To determine the best value for the dimensionality of a feature vector after dimensionality reduction is applied, we choose different values for the dimensionality r and measure the ratio of between-class and within-class distances on all 2208 families for each given value of r . Table 2 shows the mean and standard deviation of the ratios on all families. It can be seen 1545–5963 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See from the table that when $r = 2$, the approach achieves the best average recognition ability. It is also clear from the table that a dimensionality reductions step can indeed improve the recognition ability of our approach. We thus set the value of parameter r to be 2 in all experiments that evaluate the search accuracy of the approach. Specifically, we project all feature vectors along the directions of the two eigenvectors of $S_w^{-1} S_b$ that correspond to its two largest eigenvalues.

We then compare the recognition ability of our approach with that of a traditional CM. For a fair comparison, we compute the ratio of the projected between-class distance and the projected within-class distance along the direction of the eigenvector of the largest eigenvector of $S_w^{-1} S_b$ for each family. In addition, we construct a traditional CM for each family and treat the alignment score obtained by aligning a single sequence to this traditional CM as a single feature for the sequence and a ratio of between-class distance and within-class distance can also be computed. Table 3 shows the results of the comparison.

Table 3 shows that, for 74.86% of the 2208 families, the projection along the direction of the largest eigenvector of $S_w^{-1} S_b$ leads to a higher ratio of between-class distance and within-class distance than that of the traditional CM. It is clear from the table that our approach has the potential to further improve the search accuracy for most of the ncRNA families available in Rfam 11.0.

The classification by either classifier is based on the computed posterior probability for the window to contain a sequence segment from the family. For a traditional Bayesian classifier, a threshold value of 0.5 is generally used to determine which class a feature vector belongs to [2]. In our experiments, we also use a threshold value of 0.5 to determine whether a window contains a sequence segment from the family or not. In other words, if the computed posterior probability is a value larger than 0.5, the window is considered to contain a sequence segment from the family and otherwise not.

3.2 Search Accuracy

A software package RNASC (RNA Search by Classification) has been developed to implement our approach. The source code of RNASC can be downloaded from the web page

at <http://www.rworldpublishing.org/software.html>. RNASC is written in both ANSI C and MATLAB programming languages, where the steps in feature extraction and classification are developed with ANSI C and the procedure of dimensionality reduction is implemented with MATLAB.

To test the search accuracy of our approach, we randomly generate a sequence of 10^5 nucleotides for each family. This long sequence is used as a random background and we then insert a number of sequences from the family into the background. Specifically, we compute the average similarity of each sequence that is not a seed sequence to all seed sequences and insert the 30 sequences that have the lowest average similarity values into the random background. For families that have less than 30 sequences, we insert all sequences in the family into the random background. Again, the value of the parameter k is set to be 4 and the LDA based approach reduces the dimensionality of a feature vector to 2. A hit is reported by RNASC if the computed posterior probability for a window to contain a member of the searched family is larger than 0.5.

We then use a window to slide through the sequence and align the sequence segments in each window to the sets of CMs we have constructed to extract a feature vector for the window. To allow for additional insertions that may occur during the evolution, the size of the window is set to be 1.5 times of the length of the longest sequence in the family.

The search accuracy is evaluated in terms of both sensitivity and specificity. Sensitivity is the percentage of inserted sequences that are correctly recognized by the program and specificity is the percentage of recognized sequences that are members of the family. We compare the search accuracy of RNASC with that of a few other search tools including Infernal 1.1 [17], Structator [6], and RNATOPS [9]. Tables 4 and 5 show the distribution of sensitivity and specificity obtained with all four search tools on the 2208 families. We also perform pair-wise χ^2 tests to compare the results obtained with RNASC and other tools, the p-values of the tests are also shown in the tables. It is clear from the tables that the result of RNASC is significantly different from those of others and it achieves higher average sensitivity and specificity than all three other tools.

In addition, we define the *overall accuracy* as the average value of sensitivity and specificity and compare the overall accuracy achieved by RNASC and all three other tools on each family. Table 6 shows the result of comparison. It can be seen from the table that, for 1513 of the 2208 tested families, RNASC achieves higher overall accuracy and for only 119 families, the overall accuracy of RNASC is lower than all of the three other tools. Testing results in tables 4, 5, and 6 show that our approach is able to capture additional features that are important for the recognition of most ncRNA families and the search accuracy is thus significantly improved.

3.3 Computation Time and Efficiency

As we have analyzed, the computation time needed to extract features from a sequence segment is $O(kln^3)$, where l is the number of sets of nested stems obtained from the secondary structure, k is the number of CMs generated for each set of nested stems, and N is the length of the sequence segment. The training data set in total contains the features

extracted from $2m$ sequences, where m is the number of sequences in the family. Computing matrices S_w and S_b needs time $O(k^2l^2m)$, the computation time needed to solve the eigenvalue problem in LDA is $O(k^3l^3)$. The preprocessing stage thus needs $O(kln^3 + k^2l^2m + k^3l^3)$ time.

When the search starts, we use a window to slide through the genome and align the sequence segment within the window to the set of kl CMs we have constructed during the preprocessing stage, for each particular position in the genome, the sequence-structure alignment between the sequence segments in the window and a CM can be performed in time $O(N^2)$, where N is the length of nucleotides contained in the ncRNA. This is due to the fact that the alignment algorithm can use the results that have been computed and stored in the dynamic programming tables. The total computation time is thus at most $O(kln^3 + k^2l^2m + k^3l^3 + kln^2L)$, where L is the length of the genome. By applying the technique that was proposed in [3] and later used in [17], the total computation time can be further improved to $O(kln^3 + k^2l^2m + k^3l^3 + klnL)$.

Based on the above analysis, the asymptotic time complexity of our approach is higher than that of a CM based optimal sequence-structure alignment. However, most of the CM-based search tools have employed filters to screen out the part of the genome that is unlikely to contain the searched ncRNA. A CM based optimal sequence-structure alignment is only performed on these sequence segments that have passed the filtration process. Therefore, computational efficiency is no longer an important issue for search. To check whether our approach can search with satisfactory computational efficiency or not, we measure the average computation time needed by RNASC to complete the search of a sequence of 10^5 nucleotides and compare it with that of Structator, Infernal, and RNATOPS. Table 7 shows the average amount of search time in minutes needed by all four search tools. It can be seen from the table that, although the expected asymptotic computation time is greater than that of previous approaches, the practical run time of RNASC is still as feasible as that of other search tools.

3.4 On Benchmark Data Sets

In addition to the search we have performed on sequences generated by ourselves, we also test the performance of RNASC, Structator, Infernal and RNATOPS on the RMARK3 benchmark developed in [17]. The benchmark contains both a training data set and a testing data set for 106 ncRNA families. The sequences in a training data set are selected such that the mutual similarity is at most 60% and the mutual similarity between any pair of sequences in a testing data set is at most 70%. The benchmark also contains 10 pseudo genomes, each of which contains 10^6 nucleotides and 780 testing sequences from the 106 families are inserted into these pseudo genomes. We apply all four search tools to recognize the inserted sequences in the 10 pseudo genomes and use pair-wise χ^2 tests to evaluate the statistical significance of the result obtained with RNASC. Table 8 shows the distribution of the overall accuracy achieved by all four search tools and the p-values of the pair-wise χ^2 tests by comparing the result of RNASC with those of other tools.

We also compare the overall accuracy of all four search tools on each of the 106 families in the RMARK3 benchmark. Table 9 shows the result of comparison. It can be seen from the

table that RNASC achieves higher overall accuracy than all three other tools on 72 of the 106 families in the benchmark. Only on 6 families, its overall accuracy is lower than all three other tools. Our testing results on RMARK3 benchmark clearly show that, on average, RNASC is able to achieve higher search accuracy than all three other tools.

4 Conclusions

In this paper, we developed a new machine learning approach for searching genomes for ncRNA genes. This new approach models a sequence segment with a feature vector and a classifier is used to determine whether it is a member of an ncRNA family or not. Our testing results also show that this machine learning based approach can capture important features of an ncRNA family and thus achieve improved search accuracy compared with existing tools.

So far, the extracted features do not contain information that represents the interaction of structural units in the secondary structure of a family. We believe that including such information in the features [27] can further improve the search accuracy and it thus constitutes one important direction for our future work.

References

1. Bafna, V.; Zhang, S. FASTR: Fast database search tool for non-coding RNA. Proceedings of the 3rd IEEE Computational Systems Bioinformatics Conference; 2004. p. 52-61.
2. Bishop, CM. Pattern Recognition and Machine Learning (Information Science and Statistics). Springer Verlag; 2007.
3. Brown, MP. Small subunit ribosomal RNA modeling using stochastic context free grammars. Proceedings of the 8th International Conference on Intelligent System Computational Biology; 2000. p. 57-66.
4. Burge SW, et al. Rfam 11.0: 10 years of RNA families. Nucleic Acids Research. 2013; 41:D226–D232. [PubMed: 23125362]
5. Eddy S, Durbin R. RNA sequence analysis using covariance models. Nucleic Acids Research. 1994; 22:2079–2088. [PubMed: 8029015]
6. Fernando M, et al. Structator: fast index-based search for RNA sequence-structure patterns. BMC Bioinformatics. 2011; 12:214. [PubMed: 21619640]
7. Fukunaga, K. Introduction to Statistical Pattern Classification. Academic Press; San Diego, California, USA: 1990.
8. Geissmann T. A search for small noncoding RNAs in *Staphylococcus aureus* reveals a conserved sequence motif for regulation. Nucleic Acids Research. 2009; 37(21):7239–7257. [PubMed: 19786493]
9. Huang Z, et al. Fast and accurate search for non-coding RNA pseudoknot structures in genomes. Bioinformatics. 2008; 24(20):2281–2287. [PubMed: 18687694]
10. Janssen S, et al. Shape based indexing for faster search of RNA family databases. BMC Bioinformatics. 2008; 9:131. [PubMed: 18312625]
11. Klein RJ, Eddy SR. Rsearch: finding homologs of single structured RNA sequences. BMC Bioinformatics. 2003; 4:44. [PubMed: 14499004]
12. Kolbe DL, Eddy SR. Fast filtering for RNA homology search. Bioinformatics. 2011; 27:3102–3109. [PubMed: 21965818]
13. Lowe TM, Eddy SR. TRNASCAN-SE: A program for improved detection of transfer RNA genes in genomic sequences. Nucleic Acids Research. 1997; 25:955–964. [PubMed: 9023104]
14. Lu J. MicroRNA expression profiles classify human cancers. Nature. 2005; 435:834–838. [PubMed: 15944708]

15. Meltzer PS. Small RNAs with big impacts. *Nature*. 2005; 435:745–746. [PubMed: 15944682]
16. Nawrocki EP, et al. Infernal 1.0: inference of RNA alignments. *Bioinformatics*. 2009; 25(10): 1335–1337. [PubMed: 19307242]
17. Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology searches. *Bioinformatics*. 2013; 29:2933–2935. [PubMed: 24008419]
18. Nawrocki EP, Eddy SR. Computational identification of functional RNA homologs in metagenomic data. *RNA Biology*. 2013; 10(7):1170–1179. [PubMed: 23722291]
19. Nguyen VT, et al. 7sk small nuclear RNA binds to and inhibits the activity of cdk9/cyclin t complexes. *Nature*. 2001; 414:322–325. [PubMed: 11713533]
20. Rivas E, et al. Computational identification of noncoding RNAs in *E. Coli* by comparative genomics. *Current Biology*. 2001; 11:1369–1373. [PubMed: 11553332]
21. Sassen, Sv, et al. MicroRNA-implications of cancer. *Virchows Arch*. 2008; 452(1):1–10. [PubMed: 18040713]
22. Smith JA. RNA search with decision trees and partial covariance models. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2009; 6(3):517–527. [PubMed: 19644178]
23. Song, Y., et al. Tree decomposition based fast search of RNA structures including pseudoknots in genomes. *Proceedings of the Fourth International IEEE Computer Society Computational Systems Bioinformatics Conference*; 2005. p. 223–234.
24. Song Y, et al. RNA Structural Homology Search with a Succinct Stochastic Grammar Model. *Journal of Computer Science and Technology*. 2005; 20(4):454–464.
25. Song, Y., et al. Efficient Parameterized Algorithm for Biopolymer Structure-Sequence Alignment. *Proceedings of the Fifth International Workshop on Algorithms in Bioinformatics*; 2005. p. 376–388.
26. Song Y, et al. Efficient parameterized algorithm for biopolymer structure sequence alignment. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*. 2006; 3(4):423–432. [PubMed: 17085850]
27. Song Y, Chi AY. A new approach for parameter estimation in sequence-structure alignment of non-coding RNAs. *Journal of Information Science and Engineering*. 2014 in press.
28. Weinberg Z, Ruzzo WL. Sequence-based heuristics for faster annotation of non-coding RNA families. *Bioinformatics*. 2006; 22(1):35–39. [PubMed: 16267089]
29. Yang Z, et al. The 7sk small nuclear RNA inhibits the cdk9/cyclin t1 kinase to control transcription. *Nature*. 2001; 414:317–322. [PubMed: 11713532]
30. Yao Z, et al. CM Finder—A covariance model based RNA motif finding algorithm. *Bioinformatics*. 2006; 22(4):445–452. [PubMed: 16357030]

Biographies

Chunmei Liu received her Ph.D. degree in Computer Science from the University of Georgia, USA in 2006. She worked as an assistant professor of Computer Science at Howard University from 2007 to 2010, and an associate professor of Computer Science at Howard University from 2010 to 2014. She is now a professor of Computer Science at Howard University, USA. Her research interests include proteomics, structural bioinformatics, and algorithm design.

Yinglei Song received his Ph.D. in Computer Science from the University of Georgia, USA in 2006. He worked as an assistant professor of Computer Science in the Department of Mathematics and Computer Science at the University of Maryland Eastern Shore, USA from 2007 to 2013. He is now a professor of Computer Science in the School of Computer Science and Engineering at Jiangsu University of Science and Technology, China. His research interests include bioinformatics, algorithm design, and machine learning.

Zhi Wang received his Master degree in Electrical Engineering from Harbin Engineering University, China in 1987. He worked as an associate professor of Electrical Engineering and Computer Science at Jiangsu University of Science and Technology, China from 1997 to 2007. He is now a professor of Computer Science at Jiangsu University of Science and Technology, China. His research interests include machine learning, data mining, data fusion and automatic control.

Figure 1.

The procedure to partition the stems in the secondary structure of an ncRNA family. (a) the stems in a secondary structure; (b) a directed graph constructed to represent the relationships of stems in the secondary structure in a); c) the sets of nested stems obtained from the partition.

Figure 2.

The construction of a CM from an alignment. a) columns i , $i+1$, $i+2$, and $i+3$ in a structural alignment; b) For each column in a), one or two states are created and arrows represent the transitions between states for consecutive columns in the alignment.

Figure 3.

The extracted two dimensional feature vectors for two ncRNA families. (a) the feature vectors obtained on sequences from family SNORA73 and randomly generated sequences; (b) the feature vectors obtained on sequences from family Entero_OriR and randomly generated sequences.

The means and standard deviations of computation time (in seconds) and the ratios of and between-class and within-class distances on 2208 RNA families.

Table 1

The value of k	1	2	3	4	5	6	7	8
Mean of the ratios	14.3	18.5	20.7	22.4	22.8	22.9	23.1	23.2
Standard deviation of the ratios	5.1	6.2	5.6	6.7	6.4	5.8	6.3	6.1
Mean of time (sec)	4.7	8.9	15.2	17.6	24.1	27.8	34.5	36.5
Standard deviation of time (sec)	3.2	5.8	10.3	11.5	14.7	18.3	23.3	24.1

Table 2

The means and standard deviations of the ratios of and between-class and within-class distances on 2208 RNA families. The last column shows the recognition ability based on the original feature vectors.

The value of r	1	2	3	4	5	Original
Mean of the ratios	23.7	26.5	25.3	23.6	22.7	22.4
Standard deviations of the ratios	5.4	5.3	6.4	7.2	6.1	6.7

Table 3

The comparison of the ratios of between-class distance and within-class distance for our approach and traditional CM.

	Our Approach has a higher ratio	Traditional CM has a higher ratio
Percentage	74.86%	25.14%

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 4

The distribution of sensitivity of RNASC, Structator, Infernal and RNATOPS on the 2208 families in Rfam 11.0.

Distribution	RNASC	Structator	Infernal	RNATOPS
Above 90%	1321	831	1165	1032
80%–90%	695	752	635	617
70%–80%	103	291	317	352
Below 70%	89	334	91	207
Average	89.4%	78.7%	84.6%	81.3%
Std	9.5%	22.2%	11.5%	16.4%
p-value	–	$<1.0 \times 10^{-5}$	$<1.0 \times 10^{-5}$	$<1.0 \times 10^{-5}$

Table 5

The distribution of specificity of RNASC, Structator, Infernal and RNATOPS on the 2208 families in Rfam 11.0.

Distribution	RNASC	Structator	Infernal	RNATOPS
Above 90%	1417	1052	1290	1132
80%–90%	651	452	632	598
70%–80%	82	462	75	343
Below 70%	58	242	211	135
Average	92.3%	84.3%	88.7%	85.9%
Std	7.2%	11.4%	8.9%	13.5%
p-value	-	$<1.0 \times 10^{-5}$	$<1.0 \times 10^{-5}$	$<1.0 \times 10^{-5}$

Table 6

The numbers of families on which RNASC achieves higher overall accuracy than all of the three other tools, two of them, one of them and none of them.

Number of tools that outperform RNASC	3	2	1	0
Number of Families	119	214	362	1513

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 7

The average amount of time needed by each search tool to complete a sequence of 10^5 nucleotides.

Search Tools	RNASC	Structator	Infernal	RNATOPS
Time (mins)	7.32	1.53	5.51	4.65

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

Table 8

The distribution of overall accuracy of RNASC, Structator, Infernal and RNATOPS on RMARK3 benchmark.

Search Tools	RNASC	Structator	Infernal	RNATOPS
Above 90%	65	45	61	56
80%–90%	38	30	34	31
70%–80%	3	17	10	15
Below 70%	0	14	1	4
Average	92.7%	84.5%	88.7%	86.3%
Std	7.6%	10.6%	8.0%	10.3%
p-value	-	$<1.0 \times 10^{-4}$	<0.1	<0.002

Table 9

The numbers of families on which RNASC achieves higher overall accuracy than all of the three other tools, two of them, one of them and none of them for the 106 families in RMARK3 benchmark.

Number of tools that outperform RNASC	3	2	1	0
Number of families	6	16	12	72