A Classifier System for Predicting RNA Secondary Structure

Abstract: Finding the secondary structures of Ribonucleic Acid sequences is a very important task. The Secondary structure helps determine their functionalities which in turn plays a role in the proteins production. Manual laboratory methods uses x-ray diffraction to predict secondary structures but it is complex, slow and expensive. Therefore, different computational approaches are used to predict the RNA secondary structure in order to reduce the time and cost associated with the manual process. We propose a system called IsRNA to predict a single element, internal loop, of the RNA secondary structure. IsRNA experiments with different classifiers such as SVM, KNN, Naïve Bayes and simple K means to find the most suitable. Is RNA selects twenty four features classified into five groups are considered and experiments are designed to show the most relevant feature groups. The system is evaluated using Rfam sequences and achieves an overall sensitivity, selectivity, and accuracy of 96.1%, 98%, and 96.1%, respectively.

Keywords: Bioinformatics, RNA, Secondary Structure, Internal Loop, SVM

1 Introduction

Finding the secondary and tertiary structures of Deoxyribonucleic Acid (DNA) and Ribonucleic Acid (RNA) sequences is a very important task in Bioinformatics. The secondary structures of RNA are important in determining their functionality (Campbell, 1999). To form secondary structures, RNA strands fold back on themselves and pair according to Watson and Crick base pairing rules. The strands are folded by pairing Guanine (G) bases with Cytosine (C) bases, and Adenine (A) bases with Uracil (U) bases (Zou, 2007). This secondary structure forms helical areas which are separated by unpaired sections as shown by Figure 1. The main elements of the structure are: stems, bulges, hairpin loops, internal loops, and multi-branched loops. Stems are areas of continuous paired bases. A bulge is small loop separating two stems. The bulge loop is formed by few unpaired bases on one strand only. Hairpin loops on the other hand, are formed from a stem and a closing unpaired loop. Internal loops are a similar to hairpin loops, but they start and end with stems. Finally, a multi-branched loop is a structure similar to internal loop except that it has more than two stems branching off of it. Figure 1 shows two internal loops in the middle and two hairpin loops on both ends.
Because of the high cost and complexity of determining the secondary structures in the laboratory, computational prediction has become very popular. Several approaches have been proposed in the literature and can be summarized as follows. Dynamic Programming utilizes thermodynamic approaches and has the advantage of finding the global minimum free energy model (Durbin et al., 1998). Stochastic Context Free Grammar uses more than one sequence, and captures the common primary and secondary structures of a given sequence (Sakakibara et al., 1994). In general methods can be classified as either minimum energy modelling methods or comparative modelling. Minimum energy modelling folds the RNA molecule and finds the structure with the lowest free energy. The main advantage of this modelling is that it can be used on single sequences and finds the most stable structure. However, it does not resolve all types of structure well (Alkan et al., 2006).

On the other hand comparative modelling works by aligning multiple sequences. The main idea of comparative methods is to look for compensating complementary changes in two regions of an RNA sequence. For instance, if changes at RNA alignment show a strong relationship with complementary changes at another RNA alignment, then the two RNA alignments are candidates for a secondary structure base pair (Bindewald and Shapiro, 2006). Comparative methods are better at finding pseudo-knots than thermodynamic methods which deal with energy transformation that occur in and between structures (Alberty, 2004).

In this paper we propose a new prediction system that looks for internal loops. We experiment with a large number of features and classifiers to select the best combination. The rest of this paper is organized as follows. Section 2 presents the recent advances in RNA secondary structure prediction. Section 3 justifies the use of SVM and discusses the high level framework for IsRNA in details. Section 4 explains the feature vector construction and the feature groups used. Section 5 describes the data sources, performance metrics and details of the experiments used to evaluate the system.

2 State of the Art
There exists a multitude of computational based tools to predict RNA secondary structures. None the less, there is still room for improvement in terms of functionality, speed and accuracy. MiRenSVM by Ding, Zhou, and Guan (2010) is SVM based computational approach for better prediction of miRNA precursors using an ensemble SVM classifier with multi-loop features. The MiRenSVM was first trained on Homo sapiens and Anopheles gambiae genomes, and the results were 93% sensitivity, 96.5% specificity and 96% accuracy. Unfortunately, when performance
evaluation was conducted over 27 additional species from miRBase13.0 the accuracy degrades to 92.8%.

PPfold is stochastic context-free grammars multithreading based comparative prediction aimed at cutting the run time (Sukosd et al., 2011). It is a multithreaded version of Pfold designed to run on multicore processors. Pfold forms a neighbour-joining tree given aligned sequences (Knudsen and Hein, 2003). The algorithm was tested using the BRaliBase1 dataset and reported sensitivity, selectivity, and correlation of 86.4%, 96%, and 91.2% respectively. The time on 8-core to identify structural elements in 24 HIV-1 genomes was 65 minutes.

miRFam is an effective automatic miRNA classification method based on n-grams and a multiclass SVM (Ding, Zhou, and Guan, 2011). It employs n-grams to extract features from known precursor sequences, and then trains a multiclass SVM classifier to classify new miRNAs. The overall classification accuracy is around 90% when testing the algorithm using the entire miRBase15 dataset.

Harmanci, Sharma and Mathews (2011) proposed TurboFold which is an iterative probabilistic estimation of secondary structure for multiple RNA sequences. It combines the information from the comparative analysis between sequences with the thermodynamic folding model, based on the nearest neighbor thermodynamic model, and used homologous RNA sequences as input and estimates of the base pairing probabilities for each sequence as output. TurboFold achieves 80% sensitivity, and 95% positive predictive value.

**Figure 1** RNA secondary structure (Durbin et al., 1998)
Xiao et al. (2011) identify microRNA precursors based on random forest with network-level representation method of stem-loop structure. They translate a pre-miRNA stem-loop secondary structure to a network, consisting of nodes and edges, and use network parameters to construct prediction model. The accuracy reported form independent datasets can dip as low as 91.3% and might reach 97.6%

RNA-SSP is a classification system that uses KNN classifier for predicting RNA hairpin loops. It combines computational approaches and machine learning classifiers to predict individual structure elements using a new search heuristic (Aldwairi, Alqarqaz, and Duwairi, 2009). This work extends RNA-SSP to predict internal loops and uses SVM for better classification.

3 IsRNA Framework
Figure 2 below shows the detailed steps followed in IsRNA framework. The first step is to obtain the sequences from the Rfam databases which are represented in Stockholm format. Because we are interested in internal loops only, the files are processed and internal loops information is extracted and arranged in the format shows by Figure 3. Step two separates the loop from the closing pair at both sides. Each two lines in the input file represent one internal loop. The lines start with the closing pair bases followed by the loop part and the other closing pair bases. The input file might contain multiple lines to indicate multiple internal loops in the same family because an RNA family might contain one or more internal loops.

Step three builds the feature vectors as explained in Section 4. The processed data in the form of vectors may now be used for either training or testing the SVM classifier in steps four and five. The concept is straight forward and flexible enough to add new features or predict other elements.

3.1 Support Vector Machine (SVM)

Support Vector Machine is a supervised learning method used for data mining and was introduced in 1992 by Baser, Guyon, and Vapnik. It is a type of a linear classifier based on machine learning theory to maximize predictive accuracy, and to avoid over fitting of the data (Han and Kamber, 2005).

SVM plots a hyper plane which represents the class label, and points that represent the vectors of the instances to be classified as shown in Figure 4. The classification process is done as follows. The points or instances closest to the hyper plane or class label, most likely belong to that class label. Any points or instances far away from the hyper plane most likely do not belong to that class label (Han and Kamber, 2005).
SVM uses different functions or kernels to determine how to plot or map the data around the hyper plane. Linear, polynomial, radial basis function (RBF), and sigmoid are a few examples. Those kernels play an important role in the classification accuracy and performance. SVM is superior in terms of performance compared to other classifiers used in previous work such as k-nearest neighbour (KNN). IsRNA uses SVM because it has been extensively researched and improved in terms of the speed for training and testing, as well as the variety of available kernels. The experimental results confirm this choice.

**Figure 2 IsRNA Framework**

![IsRNA Framework Diagram]

**Figure 3 IsRNA input sample**

\[
\begin{align*}
    C / G & \text{GAGA} / G \\
    G / A & \text{UGG} / U \\
    G / G & \text{UCC} / C \\
    C / A & \text{AAG} / G \\
\end{align*}
\]
4 Feature Groups

In step 3 the internal loop sequence file will be read and processed to compute the feature vectors. Those vectors represent internal loop identifying properties and will be used to train the classifier. Figure 5 shows the pseudo code used for computing the feature vectors of the extracted internal loop sequences, which have O(n) complexity, where n refers to the size of internal loop sequence. The resulting vector consists of five feature groups and total of 24 features detailed below.

1. **Internal Loop Energy (LE)** is the foundation for RNA secondary structure prediction, because the stability of RNA secondary structure depends on it. Therefore it is used as the main feature for predicting internal loops. There are many different ways to calculate loop energy. IsRNA uses free energy tables for RNA folding from Turner Group, at 37º version 3.0, to calculate internal loop energy (Ding et al., 2010).

![Figure 4 Example of SVM mapping](image)

![Figure 5 Building feature vectors pseudo code](image)
According to Turner's rule the free energy of internal loop is the sum of two terms. The first term depends on the internal loop size. The second term depends on the closing pair's energies and each closing pair's energy depends on the mismatching pairs that come after them. Those energies are taken from internal loop terminal stacking energy tables provided by Turner group (Turner, 2010). The internal loop energy accounts for four features in the IsRNA feature vector. The energy for loop size, the energies for the two closing pairs, and the energy for the internal loop itself, which is the sum of all previous energies.

2. **CG-Content (CG)** is the percentage of Cytosine (C) and Guanine (G) in the internal loop sequence, which is calculated according to Equation (1).

\[
\text{CG-Content} = \frac{G + C}{A + U + C + G} \times 100
\]  

(1)

The CG-Content percentage plays a major role in the stability of RNA secondary structure, because the CG base pair is more stable than the AU base pair. The CG base pair is bound by three hydrogen bonds, while AU pair is bound by two hydrogen bonds. Therefore, RNA with high CG-Content is more stable than RNA with low CG-Content. Consequently, the more stable the RNA sequence the more accurate the internal loop prediction (Chan et al., 2009).

3. **The Compositional Factor (CF)** is the percentage of each of the four nucleotides \{A, U, C, and G\} in the RNA sequence, which is calculated according to Equation (2).

\[
\text{Frequency of } x \text{ over } L
\]  

(2)

Where, \(x\) belongs to set of nucleotides \{A, U, C, G\}, and \(L\) refers to the length of the RNA sequence. This feature indicates the frequency for each nucleotide, which is useful in making comparisons between RNA secondary structure loops and act as a differentiating factor among them. This contributes four additional features to our vector (Chang et al., 2008).

4. **The Potential Base-pairing Factor (PBF)** is the maximal probability of the occurrence of each of the three possible pairs (G-C), (G-U), and (A-U). It is calculated according to Equation (3).

\[
\text{PBF} = \frac{\text{Min} (\text{Num} (x), \text{Num} (y))}{L/2}
\]  

(3)
Where \((x, y)\) refers to (G-C), (G-U), and (A-U) pairs. This contributes three additional features to our vector (Chang et al., 2008).

5. The Nucleotide Proportional Factor (NPF) is the percentage of each nucleotide occurrence to other nucleotides occurrences, which is calculated according to Equation (4).

\[
\frac{\text{Num}(x)}{\text{Num}(y)} \tag{4}
\]

Where \(x, y\) belong to the set \{C, G, A, U\}, and \(x \neq y\). This factor represents the different ratios of nucleotides in the RNA sequence and their affects. It results in 12 additional features (Chang et al., 2008).

The final outcome is a feature vector with total of 24 features and five feature groups listed in Table 1. The vector is organized as follows: \{First closing pair energy, second closing pair energy, loop size energy, internal loop energy, CG-Content, compositional factors, potential base-pairing factors, nucleotide proportional factors, class label\}. The class label indicates the type of the loop and is not a feature. In IsRNA the class label is either zero for non-internal loops or one for internal loop instances.

5. Experimental Results
We perform comprehensive set of simulations to evaluate IsRNA. Subsection 1 explains the data sources used for training and testing. Subsection 2 lists the specification for the simulation environment and Subsection 3 explains the performance metrics used to gauge the performance. Subsection 4 explains the experiments, present the results analysis and explains the trends.

5.1 Dataset Source
To learn and test our system, we used 156 families from Rfam 10.0 (Gardner et al., 2009), which produced 500 instances that formed our dataset. The internal loops are extracted as explained in Section 3 to prepare the final input files.

<table>
<thead>
<tr>
<th>Feature Group</th>
<th>Number of Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal Loop Energy</td>
<td>4</td>
</tr>
<tr>
<td>CG-Content</td>
<td>1</td>
</tr>
<tr>
<td>Compositional Factor</td>
<td>4</td>
</tr>
<tr>
<td>Potential Base-Pairing Factor</td>
<td>3</td>
</tr>
<tr>
<td>Nucleotide Proportional Factor</td>
<td>12</td>
</tr>
<tr>
<td>Total</td>
<td>24</td>
</tr>
</tbody>
</table>

Table 1 Feature groups and number of features
5.2 The Environment
IsRNA uses C++ and Visual Studio 2008 to implement the code for building the feature vectors. In addition, LIBSVM 3.0 (Chang and Lin, 2011) and WEKA 3.7.3 (Hall et al., 2009) are used to run SVM classifier which is implemented in Java. All the experiments are run on a 2.40 GHz Intel Core 2 Duo CPU running on Microsoft Windows 7 Ultimate 32-bit operating system.

5.3 Performance Metrics
Three evaluation criteria are used to measure the performance of the prediction system.

1. Sensitivity measures the proportion of actual positives correctly identified that is all internal loops correctly classified. A sensitivity of 100% means that the test recognizes all actual positives, and it is calculated according to Equation (5).

\[
\text{Sensitivity} = \frac{TP}{TP + FN} \tag{5}
\]

2. Selectivity measures the proportion of actual negatives which are incorrectly identified. A selectivity of 100% means that the test recognizes all actual negatives, and it is calculated according to Equation (6).

\[
\text{Selectivity} = \frac{TP}{TP + FP} \tag{6}
\]

Where,
True Positives (TP) is the number of correctly predicted internal loops.
False Positives (FP) is the number of instances incorrectly predicted as internal loops.
False Negatives (FN) is the number of internal loops in the reference structure that were not predicted.

3. Accuracy measures the overall performance of the system, to see how the system prediction goes for all positives and negatives instances (Chan et al., 2009). WEKA automatically generates this metric, which is calculated according to the following Equation (7).

\[
\times 100 \tag{7}
\]

5.4 Experimental Results and Performance Evaluation
First, we evaluate the performance metrics for all feature groups using SVM with linear kernel because it is simpler than the other kernels. Second, we identify the most important groups: loop energy (LE), CG-
content (CG), the compositional factor (CF), the potential base-pairing factor (PBF) and the nucleotide proportional factor (NPF). Finally, we evaluate the performance for the most important features using different kernels and different classifiers.

Figure 6 shows prediction accuracy for 400 instances when using each feature group separately. We observe that the compositional factor which contains four features results in the highest accuracy compared to the other feature groups. The accuracy with only compositional factor is far from the overall system accuracy with all features groups included, that is 55.1%. This is clear from the compositional factor definition which makes it useful in making comparisons between RNA secondary structure loops. On the other hand, CG-Content results in the lowest accuracy, because it measures the stability of the secondary structure, and does not make a good differentiating factor.

Next we plot the accuracy for different combinations of feature groups to come up with the ideal combination. Figure 7 shows the accuracy versus feature group combinations for 400 instances. The combination of loop energy and CG-Content feature groups results in the highest accuracy, despite the fact that each of the two groups results in a low accuracy when used individually according to Figure 6. However, their combinations results in accuracy equal to the system accuracy with all feature groups which is about 55.1% for 400 instances. This is because the CG-Content makes sure the loop is stable and loop energies determine it is an internal loop. Therefore, loop energy and CG-Content feature groups are considered most significant features combination among all other feature groups.

Figure 8 shows the accuracy when training the system using different combinations between features groups for 400 instances. The testing was done using the extracted features for twenty hairpin loops rather than internal loops. In this case the lower the accuracy the better, because it now represents the number of hairpin loop instances identified as internal loops. The combination of loop energy and the nucleotide proportional factor has the lowest percentage compared to the other combinations. This affirms our belief in that the loop energy is main feature in predicting RNA secondary structure. Nucleotide proportional factor plays a role in distinguishing between secondary structure elements by the occurrence of nucleotides.

IsRNA performance with linear kernel is not promising. It correctly recognizes 66% from all actual positives (TP), 73% from all actual negatives (FP), and 66% from all actual positives and negatives. Table 2 shows the SVM kernels used by IsRNA and their equations used to map the data around the hyper plane. Table 3 shows the accuracy using Linear, Polynomial, Sigmoid and Radial Basis Function (RBF) kernels for 500
instances. It is clear that RBF performs better than other kernels and results in 96.1% accuracy, as opposed to Sigmoid and Polynomial which result in 95% and 92%, respectively.

Table 4 shows the overall performance numbers we obtain for sensitivity, selectivity, and accuracy when testing the system with 500 instances, with 10 cross validation folds with RBF kernel. The instances are split between training and testing with 75% used for training and the other 25% for testing. The table shows that IsRNA correctly recognizes 96.1% from all actual positives (TP), 98% from all actual negatives (FP), and 96.1% from all actual positives and negatives.

Figure 9 shows the accuracy for IsRNA using growing number of instances split as 75% for training and 25% for testing dataset with 10 cross-validation folds. A sharp increase is observed below 200 instances and the curve flattens out indicating slower accuracy increase thereafter. The increase in accuracy is expected because increasing number of instances gives the system larger dataset to learn from, which increases the ability to identify true positive instances, that is internal loops. The sharp increase in accuracy at 200 instances is due to the distribution of instances close to the hyper plane which means that the additional 100 instances are closer to the hyper plane. Therefore, the second 100 instances make the distribution denser and less scattered while adding additional instances (300, 400 and 500) makes the distribution more scattered.

**Table 2** Different SVM kernel with equations

<table>
<thead>
<tr>
<th>Kernel Type</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear</td>
<td>$x_i \times x_j$</td>
</tr>
<tr>
<td>Polynomial</td>
<td>$(\gamma \times x_i \times x_j + \text{coefficient})^d$</td>
</tr>
<tr>
<td>RBF</td>
<td>$\exp(-\gamma</td>
</tr>
<tr>
<td>Sigmoid</td>
<td>$\tanh(\gamma x_i \times x_j + \text{coefficient})$</td>
</tr>
</tbody>
</table>

**Table 3** Accuracy for different SVM kernels

<table>
<thead>
<tr>
<th>Kernel</th>
<th>Linear</th>
<th>RBF</th>
<th>Sigmoid</th>
<th>Polynomial</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>66%</td>
<td>96.1%</td>
<td>95%</td>
<td>92%</td>
</tr>
</tbody>
</table>

**Table 4** Overall Performance for IsRNA with RBF kernel

<table>
<thead>
<tr>
<th>Metric</th>
<th>Sensitivity</th>
<th>Sensitivity</th>
<th>Sensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Percentage</td>
<td>96.1%</td>
<td>96.1%</td>
<td>96.1%</td>
</tr>
</tbody>
</table>
SVM was chosen based on its superior experimental performance compared to other classification algorithms. We experiment with different classification algorithms to be used with IsRNA such as nearest-neighbour classifier (IB1). Which uses normalized Euclidean distance to find the training instance closest to the given test instance, and predicts the same class as this training instance. If multiple instances have the same smallest distance to the test instance, the first one found is used (Hall et al., 2009). K-nearest neighbour classifier (IBk) can select appropriate value of K based on cross-validation. It can also do distance weighting. In addition, we use Naïve Bayes Simple, Naïve Bayes, and Simple K Means (both Manhattan and Euclidean distance). Table 5 shows the accuracy according to the different classification algorithms used for 500 instances. It is clear that SVM outperforms the other algorithms and result in 96.1% accuracy.

**Figure 6** Accuracy using individual feature groups.

**Figure 7** Accuracy using feature group combinations
Figure 8 Accuracy versus the feature group combinations using Hairpin loop data for testing

Figure 9 Overall accuracy versus the number of instances

Table 5 Accuracy for different classification algorithms

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM (RBF)</td>
<td>96.1%</td>
</tr>
<tr>
<td>IB1</td>
<td>94%</td>
</tr>
<tr>
<td>IBk</td>
<td>94%</td>
</tr>
<tr>
<td>Naïve Bayes Simple</td>
<td>87%</td>
</tr>
<tr>
<td>Naïve Bayes</td>
<td>86%</td>
</tr>
<tr>
<td>Simple K Means (Manhattan)</td>
<td>53%</td>
</tr>
<tr>
<td>Simple K Means (Euclidean)</td>
<td>51%</td>
</tr>
</tbody>
</table>

6. Conclusions
We proposed a system, called IsRNA, to predict the internal loop secondary structure based on SVM one class classifier. The system was
trained using internal loop sequences extracted from Rfam 10.0 collection. We experiment with various feature groups and determined that while loop energies hold the most value none of the other feature can be discounted. The overall performance measures we obtained for sensitivity, selectivity, and accuracy when testing the system with about 500 instances, with split 75% as training dataset, and 25% as testing dataset, with 10 Cross-Validation folds are 96.1%, 98%, and 96.1%, respectively.

References


Title


