Abstract—MicroRNAs (miRNAs) are non-coding RNAs with approximately 22 nucleotides (nt) that are derived from precursor molecules. These precursor molecules or pre-miRNAs often fold into stem-loop hairpin structures. However, a large number of sequences with pre-miRNA-like hairpins can be found in genomes. It is a challenge to distinguish the real pre-miRNAs from other hairpin sequences with similar stem-loops (referred to as pseudo pre-miRNAs). Several computational methods have been developed to tackle this challenge. In this paper we propose a new method, called MirID, for identifying and classifying microRNA precursors. We collect 74 features from the sequences and secondary structures of pre-miRNAs; some of these features are taken from our previous studies on non-coding RNA prediction while others were suggested in the literature. We develop a combinatorial feature mining algorithm to identify suitable feature sets. These feature sets are then used to train support vector machines to obtain classification models, based on which classifier ensemble is constructed. Finally we use a boosting algorithm to further enhance the accuracy of the classifier ensemble. Experimental results on a variety of species demonstrate the good performance of the proposed method, and its superiority over existing tools.

Keywords—miRNA precursor; ensemble method; support vector machine; AdaBoost

I. INTRODUCTION

MicroRNAs (miRNAs) are non-coding RNAs (ncRNAs) of approximately 22 nucleotides that are known to regulate post-transcriptional expression of protein-coding genes [2]. They are derived from pre-miRNAs that often fold into stem-loop hairpin structures. These characteristic stem-loop structures are highly conserved in different species. One challenging research problem is to distinguish pre-miRNAs from other sequences with similar stem-loop structures (referred to as pseudo pre-miRNAs). In this paper we present a novel combinatorial feature mining method for pre-miRNA classification. Our method, named MirID, identifies and classifies an input RNA sequence as a pre-miRNA or not. Experimental results demonstrate the effectiveness of MirID.

II. MATERIALS AND METHODS

A. Datasets

We collected real pre-miRNAs and pseudo pre-miRNAs from eleven species. These RNA sequences were evenly divided into training data and test data. Table I presents a summary of the data. The first column of Table I shows a species or organism name. The second column of Table I shows the number of training sequences followed by the number of test sequences with respect to the organism’s real pre-miRNAs. The third column of Table I shows the number of training sequences followed by the number of test sequences with respect to the organism’s pseudo pre-miRNAs. As an example, referring to Arabidopsis thaliana in Table I, its training set contains 66 real pre-miRNAs and 114 pseudo pre-miRNAs; its test set contains 67 real pre-miRNAs and 114 pseudo pre-miRNAs.

The real pre-miRNAs were downloaded from miRBase available at http://www.mirbase.org/ [4]. We used RNAfold [3] to predict the secondary structures of all the RNA sequences. The lengths of the real pre-miRNAs in the dataset ranged from 60 to 120 nt. The pseudo pre-miRNAs used in this study were collected from GenBank (http://www.ncbi.nlm.nih.gov/genbank/). Following [7], we searched for the protein-coding regions of the genome sequences of the species in Table I, and divided the regions into shorter sequences, each of them having 100 nucleotides. The criteria used in choosing the pseudo pre-miRNAs are: (i) they must contain at least 18 base pairs, including Watson-Crick and GU wobble base pairs, on the stem region of the stem-loop structure, and (ii) their secondary structures have a maximum of -15 kcal/mol free energy without multiple hairpin loops [4]. These criteria
ensure that the stem-loop structures of the pseudo pre-
miRNAs are similar to those of the real pre-miRNAs.

B. Feature Pool

In designing our pre-miRNA classification method, we
examined multiple features extracted from a pre-miRNA
sequence and its secondary structure. Some of these features
were taken from our previous studies on ncRNA prediction
while others were suggested in the literature [5], [7], [9].
These features included the sequence length, the number of
base pairs, GC content, the ratio between the number of base
pairs and the sequence length, the number of nucleotides
contained in the hairpin loop (i.e., the loop size), the free
energy of the sequence’s secondary structure obtained from
RNAfold [3], the number of bulge loops, and the size of the
largest bulge loop in the secondary structure.

In addition, we considered the features described in [9].
These features included the difference of the lengths of the
two tails in the secondary structure where a tail represented
the strand of unpaired bases in the 5’ or 3’ end of the
structure, the number of tails, and the length of the larger tail.
Besides, several combined features were considered. They
included the length difference of two tails plus the larger
tail length, the size of the hairpin loop plus the larger tail
length, the size of the hairpin loop plus the largest bulge
size, the ratio between the larger tail length and the sequence
length, the ratio between the size of the hairpin loop and the
sequence length, the ratio between the largest bulge size and
the sequence length, the ratio between the largest bulge size
and the number of base pairs, the normalized free energy
[6], which is the minimum free energy of the sequence’s
secondary structure divided by the sequence length, and
the ratio between the normalized free energy and the GC
content.

The next set of features included the triplets of structure-
sequence elements described in [7]. Here we used the
dot-bracket notation [3] to represent an RNA secondary
structure. A triplet is composed of three contiguous structure
elements (bases or base pairs) that correspond to three
contiguous nucleotides along with the middle nucleotide.
There are 32 triplets, and hence 32 such features in total.

Finally we considered the symmetric and asymmetric
loops defined in [5]. We refer to the portion of the sequence
from the 5’ end to the hairpin loop as the left arm, and
the portion of the sequence from the hairpin loop to the 3’
end as the right arm. In a symmetric (internal) loop, the
number of nucleotides in the left arm equals the number
of nucleotides in the right arm. In an asymmetric (internal)
loop, the number of nucleotides in the left arm is different
from the number of nucleotides in the right arm. Features
related to these loops included the size of each loop, the
average size of the loops, and the average distance between
the loops. Other features included the proportion of A/C/G/U
in the stem, and the proportion of A-U/C-G/G-U base pairs
in the stem. Totally, there are 74 features in the feature pool.

C. Combinatorial Feature Mining

MirID adopts a novel feature mining algorithm for pre-
miRNA classification. Initially the algorithm randomly gen-
erates N feature sets from the feature pool. (The default
value of N is 100.) Each feature set contains between 1
and 150 features, randomly chosen with replacement from
the feature pool. Some features may repeatedly occur in a
feature set (thus a bagging approach is used here). Duplicate
features have more weights than the other features in the
feature set. The numbers 1 and 150 are chosen, to ensure that
there are enough feature sets containing duplicate features.
We then build a SVM model based on each feature set using
training sequences, and apply the classification model to
test sequences to calculate the accuracy of the model. The
SVM used in this study is the LIBSVM package downloaded
from http://www.csie.ntu.edu.tw/~cjlin/libsvm/. We use the
polynomial kernel provided in the LIBSVM package. The
polynomial kernel achieves the best performance among all
kernel functions included in the package.

Then, we remove the SVM models whose accuracies are
less than a user-determined threshold t. (The default value
of t is 0.8.) The feature sets used to build those removed
SVM models are also eliminated from further consideration.
We construct a classifier ensemble from the remaining SVM
models. The ensemble works by taking the majority vote
from the individual SVM models used to build the classifier
ensemble. This ensemble will be refined through several iter-
ations until its accuracy cannot be enhanced further. In each
iteration, the user-determined threshold t is incremented by
a step value, so that more accurate SVM models are used
to construct a (hopefully) better classifier ensemble in the
next iteration. (The default value of step is 0.005.)

It is likely that different combinations of remaining fea-
tures may yield an even better classifier. Our algorithm then
performs pairwise merge and split operations on the set Ss
of feature sets used to build the best classifier ensemble
obtained so far. In doing so, MirID takes four steps: (1)
picks each pair of feature sets s1 and s2 in Ss; (2) merges
s1 and s2 into a single feature set s3 with, say p, features; (3)
randomly generates a number q, q < p; (4) randomly assigns
q features in s3 into a set s′ 1 and assigns the remaining p − q
features into another set s′ 2. Thus, these four steps take two
feature sets s1 and s2 in Ss as input and produce two new
feature sets s′ 1 and s′ 2 as output.

These pairwise merge and split operations are applied to
the feature sets used to build the best classifier ensemble
obtained so far, to generate new feature sets. The new feature
sets are then used to build new SVM models. Accurate new
SVM models, whose accuracies are greater than or equal to
the newly computed threshold t, are then used to build a new
classifier ensemble. This procedure is repeated several times
to obtain a best classifier ensemble. Figure 1 summarizes our feature mining algorithm.

D. Boosting

The performance of a classification algorithm can be further enhanced through boosting. We apply AdaBoost [1] to the classifier ensemble produced by our feature mining algorithm. Specifically, we treat the classifier ensemble as a weak classifier and continue refining it into a strong classifier through an iterative procedure. Let $X$ be a set of sequences $x_1, x_2, \ldots, x_m$ where $x_i, 1 \leq i \leq m$, is associated with a label $y_i$ such that

$$y_i = \begin{cases} +1 & \text{if } x_i \text{ is a real pre-miRNA} \\ -1 & \text{if } x_i \text{ is a pseudo pre-miRNA} \end{cases}$$

The AdaBoost algorithm works with $K$ iterations. (The default value of $K$ is 20.) In iteration $k$, $1 \leq k \leq K$, the algorithm updates a weight function $W_k$ as explained below, which will be used in selecting training sequences in iteration $k+1$. Initially, every sequence has an equal weight, i.e. $W_0(x_i) = 1/m$, $1 \leq i \leq m$. In iteration $k$, the algorithm samples 1/3 sequences with replacement from $X$ based on the weight function $W_{k-1}$ to form a training set $X_k$. The set $X_k$ is then used to train a weak classifier $H_k$, which classifies each sequence $x_i$ as either a real pre-miRNA or a pseudo pre-miRNA. That is,

$$H_k(x_i) = \begin{cases} +1 & H_k \text{ classifies } x_i \text{ as a real pre-miRNA} \\ -1 & H_k \text{ classifies } x_i \text{ as a pseudo pre-miRNA} \end{cases}$$

Let $E_k = \{x_i | H_k(x_i) \neq y_i\}$. The error rate $\epsilon_k$ of $H_k$ is:

$$\epsilon_k = \sum_{x_i \in E_k} W_{k-1}(x_i)$$

(1)

Let

$$\alpha_k = \frac{1}{Z_k} \ln \left( \frac{1 - \epsilon_k}{\epsilon_k} \right)$$

(2)

The algorithm updates $W_k$ for each sequence $x_i, 1 \leq i \leq m$, as follows:

$$W_k(x_i) = \begin{cases} W_{k-1}(x_i) \times e^{-\alpha_k} & \text{if } H_k(x_i) = y_i \\ W_{k-1}(x_i) \times e^{\alpha_k} & \text{if } H_k(x_i) \neq y_i \end{cases}$$

$$W_k(x_i) = \frac{W_{k-1}(x_i) e^x(-\alpha_k y_i H_k(x_i))}{Z_k}$$

(3)

where $Z_k$ is a normalization factor chosen such that $W_k$ is normally distributed. Thus, the sequences causing classification errors in iteration $k$ will have a greater probability
of being selected as training sequences for constructing the weak classifier $H_{k+1}$ in iteration $k+1$. Using this technique, each weak classifier should have greater accuracy than its predecessor. The final, strong classifier $H$ combines the vote of each individual weak classifier $H_k$, $1 \leq k \leq K$, where the weight of each weak classifier’s vote is a function of its accuracy. Specifically, for an unlabeled test sequence $x$, $H(x)$ is calculated as follows:

$$H(x) = \text{sign}\left(\sum_{k=1}^{K} \alpha_k H_k(x)\right)$$

(4)

The function $\text{sign}$ indicates that if the sum inside the parentheses is greater than or equal to zero, then $H$ classifies $x$ as positive (i.e. a real pre-miRNA); otherwise $H$ classifies $x$ as negative (i.e. a pseudo pre-miRNA).

III. EXPERIMENTS AND RESULTS

We compared our method with two closely related methods, PMirP [8] and TripletSVM [7]. Like our method, both PMirP and TripletSVM were implemented using support vector machines. PMirP adopted a hybrid coding scheme, combining features such as free bases, base pairs, minimum free energy of secondary structure, among others. TripletSVM used triplets of structure-sequence elements, which also were included in our feature pool.

The performance measure used here is accuracy, defined as follows. A method is said to classify a test sequence correctly if the sequence is a real pre-miRNA (pseudo pre-miRNA, respectively) and the method indicates that the sequence is indeed a real pre-miRNA (pseudo pre-miRNA, respectively). A method is said to classify a test sequence incorrectly if the sequence is a real pre-miRNA (pseudo pre-miRNA, respectively) but the method mistakenly indicates that the sequence is a pseudo pre-miRNA (real pre-miRNA, respectively). For each species, the accuracy of a method is defined as the number of correctly classified test sequences of that species divided by the total number of test sequences of that species. Since our feature mining procedure is a randomized algorithm, we ran MirID thirty times and calculated the average.

Table II shows the accuracies of the three methods on the species taken from Table I. These species were used to pre-train PMirP and TripletSVM, and available from their web servers. For each species, the highest accuracy yielded by a tool is in bold. It can be seen that MirID achieves better performance than the related methods.

IV. CONCLUSION

In this paper we present a new method (MirID) for pre-miRNA classification. Experimental results showed that MirID outperforms two closely related methods, PMirP and TripletSVM. Since all the three methods were implemented using support vector machines with similar features, we conclude that the superiority of our method is due to its feature mining and boosting algorithms. Future work includes extending these algorithms for classifying other RNA structures.

REFERENCES


