Accuracy Improvement for RNA Secondary Structure Prediction with SVM [*]

Chia-Hung Chang, Chang-Biau Yang[†], Yung-Hsing Peng and Chiou-Yi Hor
Department of Computer Science and Engineering
National Sun Yat-sen University, Kaohsiung, Taiwan
[†]cbyang@cse.nsysu.edu.tw

Abstract

Prediction of RNA secondary structures has drawn much attention from both biologists and computer scientists. Consequently, many useful tools have been developed for this purpose, with or without pseudoknots. These tools have their individual strength and weakness. As a result, we propose a hybrid feature extraction method which integrates two prediction tools pknotsRG and NUPACK with a support vector machine (SVM). We first extract some useful features from the target RNA sequence, and then decide its prediction tool preference with SVM classification. Our test data set contains 723 RNA sequences, where 202 pseudoknotted RNA sequences are obtained from PseudoBase, and 521 nested RNA sequences are obtained from RNA SSTRAND. Experimental results show that our method improves not only the overall accuracy, but also the sensitivity and the selectivity of the prediction results. Our method serves as a preprocessing way in analyzing RNA sequences before employing the RNA secondary structure prediction tools. The ability to combine the existing methods and make the prediction tools more accurate is our main contribution.

Key words: RNA, secondary structure, SVM, machine learning, classification

1 Introduction

An RNA secondary structure is the *fold* of the given strand so that non-adjacent nucleotides form base pairs. There are total three possible combinations that nucleotides can bond to form a base pair: A-U, G-C, and G-U, where A-U and G-C are called *Watson-Crick pairs* and G-U is called the *Wobble pair*. Generally, an RNA secondary structure can be deemed as a sequence S with a set S' of base pairs (i, j), where $1 \leq i < j \leq |S|$ and $\forall(i, j)$, $(i', j') \in S'$, $i = i'$ if and only if $j = j'$. By this definition, no base can belong to more than one base pair. With the constraint, the RNA secondary structure prediction problem is to find the optimal fold of a given (target) RNA sequence.

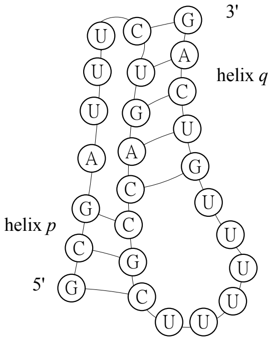

Figure 1: An example of pseudoknots. The sequence starts at 5' end and terminates at 3' end. There are two helixes in this structure.

There are mainly two kinds of secondary structures: *nested* and *pseudoknotted* structures. Predictions on nested structures are well-developed while those on pseudoknotted ones are always limited because of high computational requirements [9, 15]. A *pseudoknotted structure* is a structure in which the bases outside a helix hold hydrogen bonds to form another helix, as shown in Figure 1. Given an RNA sequence S with a set S' of base pairs, we say that S is *pseudoknotted* if in S' there exist two pairs (i', j') and (i, j) that $i < i' < j < j'$. In the past decades, various algorithms have been proposed to predict the secondary structure of a target RNA sequence. However, for computational reasons, most of them do not take the pseudoknotted structures into account. Though the prediction of pseudoknots is a problem of high interest, some well-known prediction models have proved to be NP-hard [8].

[*]This research work was partially supported by the National Science Council of Taiwan under contract NSC-96-2221-E-110-010.

The methods for predicting RNA secondary structure can be categorized into two main types. One is based on thermodynamics [14, 15, 19], and the other requires comparative approaches [3, 16]. For the approaches based on thermodynamics, some experimentally determined parameters can be employed to predict the structures with rules of minimum free energy. One comparative approach requires a set of sequences, including one target sequence with unknown structure, and a set of aligned sequences whose secondary structures are given.

For predicting RNA secondary structures, pknotsRG [14] and NUPACK [6] are two well-developed, thermodynamic-based [14, 15, 19] software tools. However, each of them still has its individual strength and weakness. In this paper, we propose a way to combine the strength of both prediction tools. Based on *SVM (support vector machine)* [2, 5], we propose a new method for feature extraction, by which we can effectively integrate these two prediction tools. Briefly, our method is to determine, before prediction, which of the two tools is preferred to be applied. Experimental results show that our approach improves the average *accuracy*, *sensitivity*, and *selectivity*. Our method can be further used to integrate various kinds of RNA secondary structure prediction tools.

The rest of this paper is organized as follows. In Section 2, we will give more detailed description for pknotsRG and NUPACK, which are the tools integrated with our method. In addition, we give a brief introduction to SVM, which is an important tool for the integration. We introduce our feature extraction method in Section 3. In Section 4, we explain how to integrate pknotsRG and NUPACK. Our experimental results and conclusions are given in Sections 5 and 6, respectively.

2 Preliminary

2.1 pknotsRG

Based on thermodynamics, some researchers propose algorithms for predicting pseudoknotted structure [9, 19]. However, it has been proved [13] that predicting arbitrary pseudoknotted structures in thermodynamic way is NP-complete. Rivas and Eddy [15] thus propose a dynamic programming algorithm, which only focuses on some general classes of pseudoknots, with $O(n^6)$ time and $O(n^4)$ space in the worst case.

Based on Rivas's system (pknots) [15], a prediction software tool, *pknotsRG* [14], has been developed. By setting proper restrictions, pknotsRG efficiently predicts structures with pseudoknots. With the simplification of focusing on H-type pseudoknots, pknotsRG reduces the required time to $O(n^4)$ to report the predicted structures by considering the starting and the ending positions of a pseudoknot. The H-type pseudoknot is a common pseudoknot in PseudoBase [17]. Thus, pknotsRG has good performance when predicting RNAs in PseudoBase.

Though pknotsRG is such a powerful tool in predicting RNA secondary structure with pseudoknots, it has lower accuracy for predicting secondary structures of tRNAs in the database RNA SSTRAND [1]. In the following section, we consider another software tool, NUPACK, which is more suitable for sequences in RNA SSTRAND.

2.2 NUPACK

Dirks and Pierce [6] present an alternative algorithm which considers the partition function involved in physically related pseudoknots. The partition function gives information about melting behavior for the secondary structure in the consideration of temperature [10]. The main idea of this software tool is to compute the base-pairing probabilities. Through dynamic programming, a transformation is applicable to any algorithm based on any partition function. Conceptually, this algorithm is a dynamic programming method which considers each probability by tracing the partition function.

Generally, NUPACK is more suitable for predicting structures in RNA SSTRAND, while pknotsRG is for PseudoBase. In Section 4, we show how to integrate pknotsRG and NUPACK, by which we obtain a more accurate tool for structure prediction.

2.3 The Support Vector Machine (SVM)

A *support vector machine (SVM)* is a machine learning tool which minimizes the structural risk in statistical learning theory [2]. Given two groups of points, SVMs can perform the binary classification by finding a hyperplane that best separates these two groups. In addition to binary classification problems, SVM can also be extended to multi-class classification by applying $\frac{k(k-1)}{2}$ times of classifications if there are k classes. The hyperplane in an SVM may not be linear, since the given points are not necessarily linearly separable.

Let $P = \{(x_1,y_1),(x_2,y_2),(x_3,y_3),\ldots,(x_n,y_n)\}$, $x_i \in R^d, y_i \in \{+1,-1\}, 1 \leq i \leq n$, be the training data set, where n and d denote the number of samples and the dimension of the vectors in P, respectively. Figure 2 gives an example of hyperplane. Conceptually, for any x_i and x_j with $y_i \neq y_j$, (x_i,y_i) and (x_j,y_j) should be separated into different classes. Here we assume that there exists a hyperplane $f_H(x) = w^T x + b$ between the

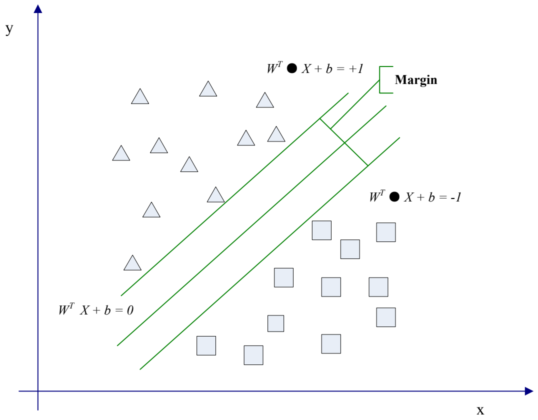

Figure 2: An example for illustrating how the hyperplane divides the data set into two disjoint subsets.

two classes, as shown in Figure 2. To divide P into two subsets, the separating function can be described as follows.

$$f_H(x_i) = sign(w^T x_i + b) = \begin{cases} \geq +1 & if \quad y_i = +1 \\ \leq -1 & if \quad y_i = -1 \end{cases}, 1 \leq i \leq n.$$

where $w^T \in R^d$ and $b \in R$. If there exist w and b such that all data in P meet the inequality in the above formula, then we say that P is *linearly separable*.

Many real world classification problems can be solved with a linear function. Nevertheless, there are still some nonlinearly separable cases that cannot be divided into two classes with a linear function. To handle this problem, there are two main approaches: nonlinear kernel functions and soft margins. A kernel function can transform the space of G into a linear one with higher dimension, which usually makes the groups linearly separable. The soft margin technique allows the data not to be restrictively separated for getting a better classification later.

3 Effective Features for RNA Structure Prediction

A high accuracy of classification in SVM requires a proper set of features. Good features have been considered for classifying proteins effectively [7, 18]. Since RNA plays the intermediate role between DNA and proteins, it is reasonable to apply similar ways to extract features from RNA sequences. In addition, we find some new feature factors based on our observation. There are nine feature factors involved in our method, and each of them includes a number of features (size), as listed in Table 1. There are 143 features in total, which will be explained in detail in the following subsections.

3.1 The Compositional Factor

The *compositional factor*, as its name indicates, stands for the appearance percentage of each of the four nucleotides (A, C, G and U) in a target RNA sequence. Let $Num(X)$ denote the number of nucleotide X in the given sequence S, and let $|S|$ denote the length of S.

Fraction of the compositional factor for nucleotide X =

$$\frac{Num(X)}{|S|}, \tag{1}$$

where $X \in \{$A, U, G, C$\}$.

3.2 The Bi-transitional Factor

In addition, the *bi-transitional factor* demonstrates the frequency of transitions, variant or invariant, between two consecutive nucleotides. Let $BT(X,Y)$ represent the number of transitions from nucleotide X to Y in the sequence.

Fraction of the bi-transitional factor for nucleotide X and Y =

$$\frac{BT(X,Y)}{|S| \text{ - } 1}, \tag{2}$$

where X, $Y \in \{$A, U, G, C$\}$. Since there are four nucleotides (A, C, G and U), we have 16 permutations of two consecutive nucleotides, such as AA, AC, AG, AU, CA, and so on.

3.3 The Distributional Factor

Furthermore, the *distributional factor* shows how the four nucleotides distribute in the sequence. We calculate the position at the accumulated amount of 0, 1/4, 1/2, 3/4, 1 in the sequence for the four nucleotides. Let $POS_X(acc)$ denote the position at the accumulated amount acc of nucleotide X.

Fraction of the distributional factor of nucleotide X =

$$\frac{POS_X(acc)}{|S|}, \tag{3}$$

where $acc \in \{$0, 1/4, 1/2, 3/4, 1$\}$. We check each nucleotide in 5 accumulated amounts, thus we have 20 features in this factor.

Table 1: The nine feature factors for prediction tool preference.

Factor	Explanation	Size	Abbrev.
Composition	The percentage of each nucleotide	4	C
Bi-transition	The frequency of transitions in two consecutive nucleotides	16	B
Distribution	Distributions of four nucleotides	20	D
Tri-transition	The frequency of transitions in three consecutive nucleotides	64	T
Potential base-pairing	The information of possible helixes	3	P
Nucleotide proportion	The proportion of nucleotides	12	N
Potential single-strand	The information of possible single strand	3	S
Sequence specific score	A score based on its primary and potential secondary structure	1	R
Segmental information	Quaternary codes	20	G

3.4 The Tri-transitional Factor

The *tri-transitional factor* specifies the contents of triplets in a sequence. Let $TT(X,Y,Z)$ represent the number of transitions for three consecutive nucleotides X, Y, and Z in the sequence.

Fraction of the tri-transitional factor among nucleotide X, Y, and Z =

$$\frac{TT(X,Y,Z)}{|S|\text{ - }2}, \tag{4}$$

where X, Y, $Z \in \{\text{A, U, G, C}\}$. There are $4 \times 4 \times 4$ permutations of the transitions. Therefore, we have 64 features in this factor.

3.5 The Potential Base-pairing Factor

The *potential base-pairing* information calculates the maximal probability of the occurrence of each of the three possible pairings (A-U), (G-C) and (G-U). We obtain the values by first looking for the minimum number of nucleotides involved in each sort of pairings, and then dividing it by the half length of that sequence. Let $min(i, j)$ be the minimum number of i and j.

Fraction of the potential base-pairing factor of base-pair (X,Y) =

$$\frac{min(Num(X),Num(Y))}{|S|\text{ / }2}, \tag{5}$$

where base-pair (X,Y) refers to any kind of (A-U), (G-C),and (G-U) pairs. Therefore, there are 3 features in this factor.

3.6 The Nucleotide Proportional Factor

Nucleotide proportion is the collection of the four nucleotides divided by each other. This factor examines the effect of different ratios of nucleotides in the sequence.

Fraction of the nucleotide proportional factor =

$$\frac{Num(X)}{Num(Y)}, \tag{6}$$

where X, $Y \in \{\text{A, U, G, C}\}$, and $X \neq Y$. We obtain 12 features by this encoding rule.

3.7 The Potential Single-stranded Factor

The complementary factor of the information in *potential base-pairs* is that in *potential single-strand*. The potential single-stranded information is the difference between the occurrence of each of the three possible pairings (A-U), (G-C) and (G-U). The rest of nucleotides which remain unpaired form the single strand in a sequence. As a result, we have 3 features from this factor. Let $abs(i, j)$ be the absolute value of the difference of i and j.

Fraction of the potential single-stranded factor corresponding to base-pair (X,Y) =

$$\frac{abs(Num(X),Num(Y))}{|S|}, \tag{7}$$

where base-pair (X,Y) refers to any kind of (A-U), (G-C),and (G-U) pairs.

3.8 The Sequence Specific Score

Each base, paired or unpaired, has its individual free energy. The *sequence specific score* contains both the minimum free energy (MFE) rules and the information of sequence combination, including single position, double consecutive position, and triple consecutive position. With the spirit of MFE, we score the sequences according to the distribution of base pairs. We now employ three rules for scoring. The rules are given as follows.

Rule 1. The bi-transitional factor is reused here. When nucleotides A-U, G-C, and G-U appear in consecutive positions, they cannot form pairs. We take them as unprofitable to minimize the free energy. According to the number of hydrogen bounds supporting the three kinds of pairs, we subtract 1 for consecutive G-U, 2 for consecutive A-U, and 3 for consecutive G-C. For other types of consecutive nucleotides, the subtraction is not performed.

Rule 2. Similarly, the tri-transitional factor is considered. Once we detect that the triplets AXU, UXA, GXC, CXG, UXG, and GXU exist in the sequence, we plus an amount of 1 for triplets with nucleotide GXU or UXG, 2 for AXU or UXA, and 3 for GXC or CXG (X stands for an arbitrary nucleotide). In this case, we assume that nucleotides would fold to each other with an extreme sharp shape. For other triplets, the subtraction is not performed.

The situations described in Rule 1 and Rule 2 are called *pairing transitions*. Let $S[i,j]$ denote the substring of S that starts with the ith character and ends with the jth character. The mathematical notation of the sequence specific score is given as follows:

Sequence specific score =

$$\sum_{l=1}^{|S|-1} f_1(l) + \sum_{l=1}^{|S|-2} f_2(l), \tag{8}$$

where $f_1(l) \in \{0,-1,-2,-3\}$ and $f_2(l) \in \{0,1,2,3\}$ denote the energy obtained with Rule 1 on $S[l,l+1]$ and Rule 2 on $S[l,l+2]$, respectively.

3.9 The Segmental Factor

The last factor, *segmental* information, views one sequence as a constitution of 20 non-overlapping segments. Each of the 20 segments is numbered with a quaternary encoding scheme. In our method, the quaternary encoding technique encodes (A) to 0, (U) to 1, (G) to 2, and (C) to 3. The quaternary number of each segments is represented by

$$\sum_{u=1}^{|S|/20} v * 4^{u-1}, \tag{9}$$

where v is the quaternary number for each nucleotides.

To eliminate the effect of domination caused among features with larger numerical values and those with less numerical values, we scale our data to the range [-1,+1]. LIBSVM [4] package works as a good tool in scaling as well. With these nine factors, we can obtain 511 ($2^9 - 1 = 511$) distinct types of combinations. The size of feature vectors of input data varies from 1 (only *sequence specific score* is selected) to 143 (all nine factors are included).

4 Our Method with SVM

Briefly, our goal is to improve the overall accuracy in RNA secondary structure prediction. We examine the performance of pknotsRG [14] and NUPACK [6] from the database, PseudoBase and RNA SSTRAND, and obtain the result that the strength of each tool is distinguishable. That is, combining their individual strength can bring us a better overall performance. Our method works as a preprocessing way in the prediction of RNA secondary structure. The main steps in our method are explained as follows.

Step 1. Collect RNA sequences as training data set.

Step 2. Extract the features from the training data set.

Step 3. Use SVM to build the training model for the classification of software preference.

Step 4. Extract features from the input (target) RNA sequence and obtain the software preference by the model obtained in Step 3.

Step 5. According to the software preference, execute either pknotsRG or NUPACK to obtain the secondary structure of the target RNA.

The principle of our method is simple. However, one should note that the method is flexible for adding any new feature factor, or any new prediction software tool in the future.

5 Experimental Results

5.1 The Source of Our Data and the Evaluation Criteria

To test our method, we download 247 pseudoknotted RNA sequences from PseudoBase [17] and 526 nested tRNA sequences from RNA SSTRAND [1]. In this paper, LIBSVM-2.85 package [4] is used as our SVM. In addition, the prediction of RNAs is powered by pknotsRG-1.3 [14] and NUPACK-2.1 [6]. We employ three criteria, *sensitivity*, *selectivity* [12], and *accuracy* to measure the performance of the prediction. The definitions of these criteria are described as follows.

Sensitivity $= \dfrac{100 \times TP}{TP+FN}$,

Selectivity $= \dfrac{100 \times TP}{TP+FP}$,

Accuracy $= \dfrac{100 \times TP}{TP+FP+FN}$, where TP (*true positive*) means the number of the correctly predicted base-pairs in the predicted structure, FN (*false negative*) is the number of the real base-pairs that are not predicted, and FP (*false positive*) indicates the number of the incorrectly predicted base-pairs in the predicted structure. Note that in our experiments, we do not calculate TN (*true negative*), because the number of TN in our data is large, which may marginalize the effects of other criteria. As a result, the well-known formula for *accuracy* is modified by omitting TN. Note that the original formula for *accuracy* is $\frac{100\times(TP+TN)}{TP+TN+FP+FN}$.

Table 2 shows the sensitivity, selectivity and accuracy of pknotsRG and NUPACK, for sequences from PseudoBase and RNA SSTRAND. In Table 2, one can see that for those sequences in PseudoBase, pknotsRG has better performance (accuracy of the predicted structure). For tRNAs in RNA SSTRAND, NUPACK has better results. To obtain more feasible models, we remove 50 sequences which have the same accuracy for both pknotsRG and NUPACK. Therefore, there are 723 sequences used in our experiment.

5.2 The Performance

To obtain an ideal combination of features proposed in Section 4, we examine our models with the self-consistency test and jackknife test [18]. In the self-consistency test, an SVM model is trained and tested with the same data set, whereas the jackknife test is the leave-one-out cross validation.

Table 3 shows the result of tool preference classification with 10-fold cross validation and self-consistency tests. The upper part of Table 3 shows the accuracy of classification with each feature factor. The principle of combining the factors is quite greedy. We select factors B, D, T, and G since they have higher accuracy. The combinations with higher accuracy of classification (AOC) can be further combined to obtain better combinations as shown in the lower part of Table 3. Combinations with low AOC are definitely inappropriate.

The jackknife test is an objective method [11, 18] for cross validation. The fourth column of Table 4 shows the AOC. In this table, we can find that the combinations of 000100000 has the highest AOC.

With the software preference, we can invoke either pknotsRG or NUPACK to predict the secondary structure of a target RNA. The prediction results are shown in Table 5. In the left part, the preference classification is trained and tested with the self-consistency scheme, while the right part is obtained by jackknife test. Since the jackknife test is more convincing than the self-consistency test, the results obtained by the jackknife test serve as the final reports of our method

Table 4: The result of jackknife tests and the optimal parameters. Here the term *AOC* stands for the accuracy of classification.

Factor combinations	$\log_2$(Cost)	$\log_2$(Gamma)	AOC
000000001	9	-5	64.73
000000010	9	3	61.41
000000100	9	1	58.92
000001000	13	-3	63.21
000010000	9	1	59.75
000100000	3	-1	68.33
000100001	3	-1	68.19
001000000	15	-11	66.53
001000001	7	-7	66.11
001100000	15	-13	67.63
001100001	3	-3	68.05
010000000	3	-3	65.01
010000001	11	-7	65.56
010100000	1	-3	67.22
010100001	5	-1	67.91
011000000	1	-1	67.22
011000001	1	-1	66.81
011100000	3	-3	66.53
011100001	3	-3	66.39
100000000	11	-1	62.24
111111111	13	-15	68.19

Table 6: The performance improvement in sensitivity, selectivity, and accuracy against pknotsRG and NUPACK.

	Sensitivity	Selectivity	Accuracy
pknotRG	82.30	76.02	66.79
000100000	87.57	78.92	71.56
Increment against pknotRG	5.27	2.91	4.77
NUPACK	85.03	76.52	68.04
000100000	87.57	78.92	71.56
Increment against NUPACK	2.54	2.41	3.52

From Table 4, we obtain that 000100000 is our final feature combination. In other words, *tri-transitional factor* is the most effective. It is not so surprising because the factor contains 64 features. From 5, the prediction accuracy of our method with 000100000 is 71.56%. Table 6 shows our improvement measured with the three criteria. The improvement of the average sensitivity, selectivity, and accuracy to NUPACK are 2.54%, 2.41%, and 3.52%. Besides, the corresponding improvement to pknotsRG are 5.27%, 2.91%, and 4.77%.

6 Conclusion

With the aid of SVM, we integrate two famous software tools, pknotsRG and NUPACK for predicting RNA secondary structures. Our integration is tested with 723 RNA sequences from PseudoBase and RNA SSTRAND. Experimental results show that the *tri-*

Table 2: The analysis of RNA sequences from PseudoBase and RNA SSTRAND. *Sens.*: sensitivity; *Sele.*: selectivity; *Accu.*: accuracy.

	pknotsRG				NUPACK					
	Sens.	Sele.	Accu.	Win	Sens.	Sele.	Accu.	Win	Draw	Total
PseudoBase	86.31	83.08	73.74	129	77.03	78.32	65.40	73	45	247
RNA SSTRAND	80.89	73.95	64.67	193	88.98	76.89	70.31	328	5	526
Overall	82.62	76.87	67.57	322	85.16	77.27	68.67	401	50	773

Table 3: The result of 10-fold cross validation and self-consistency tests. Here the term *AOC* stands for the accuracy of classification.

	Parameter		AOC	
Factor combinations	$\log_2$(Cost)	$\log_2$(Gamma)	10-fold	Self-consistency
000000001	11	-5	64.45	66.39
000000010	3	3	61.00	61.27
000000100	1	-1	58.78	59.47
000001000	7	-1	63.35	65.84
000010000	11	1	59.34	65.98
000100000	3	-1	68.88	99.59
001000000	15	-11	65.70	67.50
010000000	3	-1	66.39	83.13
100000000	13	-1	62.24	67.22
000100001	7	-1	68.46	100.00
001000001	15	-11	65.84	69.16
001100000	3	-1	67.63	99.72
001100001	3	-1	67.63	99.72
010000001	3	-1	66.25	85.62
010100000	3	-1	69.57	99.72
010100001	5	-3	68.05	99.59
011000000	1	-1	67.08	89.63
011000001	9	-9	66.67	68.74
011100000	5	-3	65.98	99.59
011100001	13	-15	66.25	72.34
111111111	1	-3	67.22	96.54

Table 5: The results of RNA secondary structure prediction. Here, the sensitivity, selectivity and accuracy are measured by the predicted structures.

	Self-consistency			Jackknife		
	Sensitivity	Selectivity	Accuracy	Sensitivity	Selectivity	Accuracy
000000001	88.25	78.43	71.28	87.98	77.86	70.67
000000010	86.20	78.22	70.19	86.77	77.30	69.69
000000100	85.71	77.05	69.00	85.11	76.87	68.66
000001000	87.13	78.68	70.74	86.68	78.34	70.17
000010000	86.84	78.19	70.49	85.91	77.57	69.52
000100000	91.09	82.72	76.83	87.57	78.92	71.56
000100001	91.15	82.75	76.89	87.55	78.95	71.51
001000000	88.48	78.64	71.57	87.85	78.73	71.39
001000001	88.65	78.75	71.77	88.06	78.33	71.05
001100000	91.11	82.71	76.83	86.88	78.72	70.73
001100001	91.11	82.71	76.83	86.36	78.36	70.42
010000000	89.71	80.64	74.14	87.21	78.08	70.58
010000001	89.61	80.78	74.32	87.41	78.21	70.76
010100000	91.11	82.71	76.83	87.46	78.89	71.43
010100001	91.09	82.72	76.83	87.34	78.68	71.16
011000000	90.29	81.42	75.17	87.60	78.32	71.06
011000001	88.39	78.56	71.48	87.60	78.50	71.18
011100000	91.09	82.72	76.83	87.16	78.61	70.97
011100001	88.10	79.18	72.02	87.27	78.54	70.95
100000000	87.55	78.87	71.08	87.03	78.19	70.17
111111111	90.86	82.35	76.35	87.09	78.95	71.27

transitional factor has the highest power. The advantage of our method is the flexibility. One can see that our method can be used to integrate various tools designed for different target RNAs. Through our integration, the average accuracy of RNA secondary structure prediction can be improved against NUPACK with 3.52%, and against pknotsRG with 4.77%. This means that our method is a useful strategy for integrating various competitive algorithms.

References

[1] M. Andronescu, V. Bereg, H. Hoos, and A. Condon, "RNA SSTRAND," 2004.

[2] A. Ben-Hur, D. Horn, H.T.Siegelmann, and V. Vapnik, "Support vector clustering," *Machine Learning*, Vol. 2, pp. 125–137, 2001.

[3] L. Cai, R. L. Malmberg, and Y. Wu, "Stochastic modeling of RNA pseudoknotted structures: a grammatical approach," *Bioinformatics*, Vol. 19, pp. i66–i73, 2003.

[4] C. C. Chang and C. J. Lin, *LIBSVM: A library for support vector machines*. National Taiwan University, No. 1, Roosevelt Rd. Sec. 4, Taipei, Taiwan 106, ROC, 2001. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[5] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

[6] R. M. Dirks and N. A. Pierce, "An algorithm for computing nucleic acid base-pairing probabilities including pseudoknots." Wiley InterScience (www.interscience.wiley.com), Wiley Periodicals, Inc., 2004.

[7] J. S. Lin, "An effective feature selection for protein fold recognition," master's thesis, Department of Computer Science and Engineering, National Sun Yat-sen University, Taiwan, No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan, R.O.C, Oct. 2007.

[8] R. B. Lyngsϕ and C. N. S. Pedersen, "Pseudoknots in RNA secondary structures," *Research in Computational Molecular Biology*, pp. 201–209, 2000.

[9] H. Matsui, K. Sato, and Y. Sakakibara, "Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures," *Bioinformatics*, Vol. 21, No. 11, pp. 2611–2617, 2005.

[10] J. S. McCaskill, "The equilibrium partition function and base pair binding probabilities for RNA secondary structure," *Biopolymers*, Vol. 29, pp. 1105–1119, 1990.

[11] P. Mundra, M. Kumar, K. K. Kumar, V. K. Jayaraman, and B. D. Kulkarni, "Using pseudo amino acid composition to predict protein subnuclear localization: Approached with PSSM," *Pattern Reconition Letters*, Vol. 28, pp. 1610–1615, 2007.

[12] J. Reeder, *Algorithms for RNA Secondary Structure Analysis: Prediction of Pseudoknots and the Consensus Shapes Approach*. Phd thesis, der Technischen Fakult*ä*t, der Universit*ä*t Bielefeld, 12 2007.

[13] J. Reeder and R. Giegerich, "Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics," *BMC Bioinformatics*, Vol. 5, pp. 104–116, 2004.

[14] J. Reeder, P. Steffen, and R. Giegerich, "pknotsRG: RNA pseudoknot folding including near-optimal structures and sliding windows," *Nucleic Acids Research*, Vol. 35, pp. 1–5, 2007.

[15] E. Rivas and S. R. Eddy, "A dynamic programming algorithm for RNA structure prediction including pseudoknots," *Journal of Molecular Biology*, Vol. 285, pp. 2053–2068, 1999.

[16] F. Tahi, "A fast algorithm for RNA secondary structure prediction including pseudoknots," *Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering*, Bethesda, Maryland, USA, pp. 11–17, 2003.

[17] F. H. D. van Batenburg, A. P. Gultyaev, C. W. A. Pleij, J. Ng, and J. Oliehoek, "Pseudobase: a database with RNA pseudoknots," *Nucleic Acids Research*, Vol. 28, No. 1, pp. 201–204, 2000.

[18] X. Yu, J. Cao, Y. Cai, T. Shi, and Y. Li, "Predicting rRNA-, RNA-, and DNA-binding proteins from primary structure with support vector machines," *Journal of Theoretical Biology*, Vol. 240, pp. 175–184, 2006.

[19] M. Zuker, "Mfold web server for nucleic acid folding and hybridization prediction," *Nucleic Acids Research*, Vol. 31, No. 13, pp. 3406–3415, 2003.