Predicting RNA Secondary Structures: One-grammar-fits-all Solution

Menglu Li[1], Micheal Cheng[2], Yongtao Ye[1], Wk Hon[2], Hf Ting[1], Tw Lam[1], Cy Tang[2,3], Thomas Wong[4], and Sm Yiu[1(✉)]

[1] Department of Computer Science, The University of Hong Kong, Hong Kong, China
[2] Department of Computer Science, National Tsinghua University, Hsinchu City, Taiwan
[3] Department of Computer Science, Providence University, Taichung City, Taiwan
[4] CSIRO Ecosystem Sciences, Canberra, Australia
smyiu@cs.hku.hk

Abstract. RNA secondary structures are known to be important in many biological processes. Many available programs have been developed for RNA secondary structure prediction. Based on our knowledge, however, there still exist secondary structures of known RNA sequences which cannot be covered by these algorithms. In this paper, we provide an efficient algorithm that can handle all RNA secondary structures found in Rfam database. We designed a new stochastic context-free grammar named Rectangle Tree Grammar (RTG) which significantly expands the classes of structures that can be modelled. Our algorithm runs in $O(n^6)$ time and the accuracy is reasonably high, with average PPV and sensitivity over **75**%. In addition, the structures that RTG predicts are very similar to the real ones.

1 Introduction

Secondary structures of RNA molecules play important roles in their functionalities [1, 2]. Many methods have been proposed to predict RNA secondary structures. Although the majority of RNAs have simple secondary structures, pseudoknots (base pairs crossing each other) are found in almost all classes of RNAs. Pseudoknots are known to be involved in biological functions such as stimulating ribosomal frameshifting [3, 4]. The existence of pseudoknots make the secondary structure prediction an NP-hard problem, in general [5, 6]. Existing algorithms attempt to solve the problem by considering a restricted set of pseudoknots [7–18]. Not all existing pseudoknots can be modelled. In terms of prediction accuracy, CentroidAlifold[9] generalized a centroid estimator that maximizes the expected accuracy of structure prediction. Tabei, Yasuo and Kiryu[16] proposed a fast multiple sequence alignment method named MXScarna in which the optimal structure that maximized a heuristic scoring function was found during the group alignments of stem component sequences. RNAaliFold[17] precomputed alignments using a combination of free-energy and a covariation measures, whilst TurboFold[18] utilized an iterative probabilistic method to predict secondary structures for multiple RNA sequences.

R. Harrison et al. (Eds.): ISBRA 2015, LNBI 9096, pp. 211–222, 2015.
DOI: 10.1007/978-3-319-19048-8_18

Despite of so many algorithms to predict RNA secondary structures, there exist secondary structures of known RNAs in Rfam [19] that cannot be covered by existing efficient algorithms[1]. Figure 1 shows such an example.

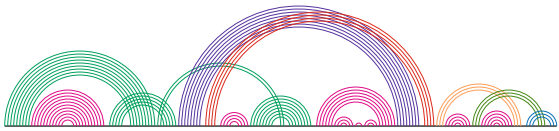


Fig. 1. A structure in Rfam which cannot be handled by existing efficient algorithms

In this paper, we proposed a grammar-based machine learning method to predict secondary structures for all RNA sequences in Rfam. Enlightened by [20], we designed a new stochastic context-free grammar called Rectangular Tree Grammar (RTG), which can model all possible secondary structures of known RNA sequences in the Rfam database. Each structure can be generated by a unique operation path, that is, the only sequence of operations that yields this sequence. A set of paths is obtained using some real RNA sequences with known structures. Rule transition probabilities and base emission probabilities are calculated based on this set. In order to determine the unknown secondary structure of a RNA sequence, dynamic programming is adopted to generate the most probable structure. This procedure takes $O(n^6)$ time, where n is the length of input RNA sequence.

The proposed approach was evaluated using several sets of sequences with one containing pseudoknot-free structures and the others with different types of pseudoknots. We compared the performance of RTG with popular prediction algorithms including gfold[7], CentroidAlifold[9], pknotsRG[21], NUPACK[22], MXScarna[16], RNAaliFold[17] and TurboFold[18]. The experimental results have shown that our approach outperforms others substantially with high PPV and sensitivity, especially on highly-pseudoknotted sequences.

2 Method

2.1 RNA Secondary Structure Definitions

Let $S = s_1 s_2 \dots s_n$ be an RNA sequence of length n. $M_{x,y}$ is the set of base pairs in the range $[x, y]$, $M_{x,y} = \{(i,j) | x \leq i < j \leq y, (s_i, s_j)$ is a base pair$\}$.

Banding: The secondary structure of $s_x \dots s_y$ is a *banding* if it satisfies the following conditions:
(i) for any $i, j, k, l \in [x, y], i \neq k, j \neq l$, if $(i,j) \in M_{x,y}$ and $(k,l) \in M_{x,y}$, then $i < k < l < j$ or $k < i < j < l$.
(ii) $(x, y) \in M_{x,y}$.

[1] We only consider algorithms which run in $O(n^6)$ time.

Gapped Banding: The secondary structure of $s_x \dots s_y \cup s_p \dots s_q$ is a *gapped banding* if it satisfies the following conditions:
(i) By cutting out the gap $s_{y+1} \dots s_{p-1}$, the secondary structure over this new sequence $s_x \dots s_y s_p \dots s_q$ is a banding.
(ii) $\forall (i,j) \in M_{[x,y]\cup[p,q]}, (i,j)$ is across the gap.

Regular Structure: A structure is a regular structure if no base pair crossing exists, that is, the secondary structure of $s_x \dots s_y$ is a *regular structure* if $\nexists i,j,k,l \in [x,y]$ such that $(i,j) \in M_{x,y}, (k,l) \in M_{x,y}$, and $i < k < j < l$.

Standard Pseudoknot of Degree k: A structure is a *standard pseudoknot* of degree k ($k \geq 3$) if it is either a *simple standard pseudoknot* of degree k or a *gapped standard pseudoknot* of degree k.

For any $1 \leq w \leq k-1$, let $H_w = \{(i,j) \in M_{x,y} | x_{w-1} \leq i < x_w \leq j < x_{w+1}\}$. We allow $j = x_k$ for H_{k-1} to resolve the boundary case.

The secondary structure of $s_x \dots s_y$ is a *simple standard pseudoknot* of degree k ($k \geq 3$) if there exists a set of $x_1, x_2, \dots, x_{k-1}$ that satisfies the following conditions (Figure 2):
(i) $x = x_0 < x_1 < x_2 < \dots < x_{k-1} < x_k = y$.
(ii) $\forall w \in [1, k-1]$, H_w is a gapped banding.
(iii) $\forall (i,j) \in M_{x,y}$, $\exists w$ such that $(i,j) \in H_w$.
(iv) $\forall w \in [1, k-1]$, if $(i,j) \in H_w, (k,l) \in H_{w+1}$, then $i < k < j < l$.
(v) there does not exist two base pairs $(i,j) \in H_w, (k,l) \in H_v, v - w \geq 2$, such that $i < k < j < l$.

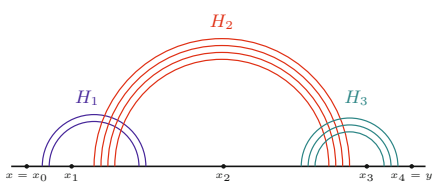


Fig. 2. A simple standard pseudoknot of degree 4

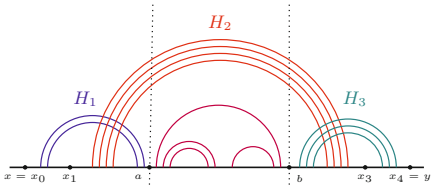


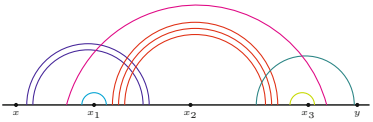
Fig. 3. A gapped standard pseudoknot of degree 4, where $s_a \dots s_b$ forms a regular structure

The secondary structure of $s_x \dots s_y$ is a *gapped standard pseudoknot* of degree k ($k \geq 3$) if there exists a, b such that $s_{a+1} \dots s_{b-1}$ is a structure defined above and $s_x \dots s_a \cup s_b \dots s_y$ satisfies the following conditions (Figure 3):
(i) By cutting out the gap $s_{a+1} \dots s_{b-1}$, the secondary structure over this new sequence $s_x \dots s_a s_b \dots s_y$ is a standard pseudoknot.
(ii) if $(i,j) \in M_{[x,a]\cup[b,y]}$ is across the gap, then $\exists w$ such that $(i,j) \in H_w$. Moreover, $\forall (k,l) \in H_w$ is across the gap.

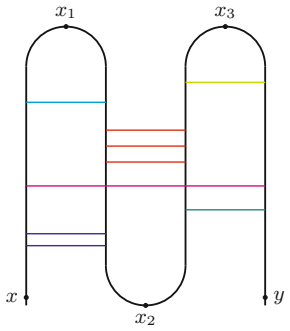
Based on our analysis to Rfam database, we focus on all standard pseudoknots of degree k ($k \geq 3$) in this paper.

Three Banding Structure: The secondary structure of $s_x \dots s_y$ is a *three banding structure* if we can find x_1, x_2, x_3 such that all the following conditions are satisfied.
(i) $x \leq x_1 \leq x_2 \leq x_3 \leq y$.
(ii) $\forall (i,j) \in M_{[x,y]}$, it must belong to one of the sets $L_{12}, L_{23}, L_{34}, L_{14}$ as defined below.
(iii) for any two pairs $(i,j) \in L_{ab}$ and $(k,l) \in L_{ab}$, then $i < k < l < j$ or $k < i < j < l$. where $L_{ab} = L_{12}, L_{23}, L_{34}$ or L_{14}.

Let $L_{12} = \{(i,j) | x \leq i \leq x_1 \leq j \leq x_2\}$, $L_{23} = \{(i,j) | x_1 \leq i \leq x_2 \leq j \leq x_3\}$, $L_{34} = \{(i,j) | x_2 \leq i \leq x_3 \leq j \leq y\}$, $L_{14} = \{(i,j) | x \leq i \leq x_1, x_3 \leq j \leq y\}$.



(a) A three banding structure.



(b) Twist this three banding structure so that all base pairs are parallel.

Fig. 4. A three banding structure and its twisted view

Figure 4 illustrates a three banding structure $s_x \dots s_y$ (Figure 4(a)) and how it is twisted so that all its base pairs are grouped into four parallel sets (Figure 4(b)), i.e., L_{12} (blue and cyan pairs), L_{23} (red pairs), L_{34} (green and lime pairs) and L_{14} (magenta pairs).

k-Crossing Structure: $s_x s_{x+1} \dots s_y$ is a *k-crossing structure* ($k \geq 3$) if it is either a *simple k-crossing structure* or a *gapped k-crossing structure*. Intuitively, in a k-crossing structure, there exist k gapped bandings where any two of them crosses each other.

For any $(1 \leq w \leq k)$, let $H_w = \{(i,j) \in M_{x,y} | x_{w-1} \leq i < x_w, x_{w-1+k} \leq j < x_{w+k}\}$. We allow $j = x_{2k}$ for H_k to resolve the boundary case. Let $C_w (1 \leq w \leq 2k) = \{(i,j) \in M_{x,y} | x_{w-1} \leq i < j < x_w\}$. $j = x_j$ is allowed for C_{2k}. A *crossing set* is defined as $CH_w = H_w \cup C_w \cup C_{w+k} (1 \leq w \leq k)$.

The secondary structure of $s_x \dots s_y$ is a *simple k-crossing structure* ($k \geq 3$) if there exist $x_0, x_1, \dots, x_{2k}$ that satisfy the following conditions:
(i) $x = x_0 < x_1 < \dots < x_{2k-1} < x_{2k} = y$.
(ii) $\forall\ (i,j) \in M_{x,y}$, $\exists w$ such that $(i,j) \in CH_w$.
(iii) $\forall w \in [1,k], CH_w$ is a regular structure, a standard pseudoknot or a three banding structure.

The secondary structure of $s_x \dots s_y$ is a *gapped k-crossing structure* if and only if there exists a, b such that $s_{a+1} \dots s_{b-1}$ is a defined structure and $s_x \dots s_a \cup s_b \dots s_y$ satisfies the following conditions:
(i) By cutting out the gap $s_{a+1} \dots s_{b-1}$, the secondary structure over this new sequence $s_x \dots s_a s_b \dots s_y$ is a k-crossing structure.
(ii) $\forall w \in [1,k], \forall (i,j) \in H_w, (i,j)$ is across the gap $s_{a+1} \dots s_{b-1}$.
(iii) $\forall (i,j) \in M_{[x,a] \cup [b,y]}$ is across the gap, $\exists w \in [1,k]$ such that $(i,j) \in H_w$ and $\nexists w \in [1,k]$ such that $(i,j) \in C_w$.

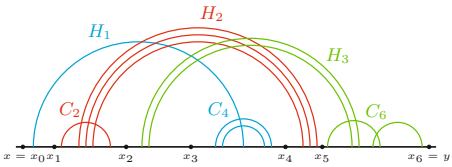


Fig. 5. A 3-crossing structure

Figure 5 depicts a 3-crossing structure. Each color denotes a crossing set, i.e., CH_1 (cyan), CH_2 (red) and CH_3 (green).

Recall the example in Figure 1. The difficulty of this structure lies on two mixed substructures called 3-crossing and standard pseudoknots for which none of the existing algorithms can model (the green basepairs form a standard pseudoknot; the green, blue, and the red basepairs form a 3-crossing structure). As a matter of fact, among all classes of Rfam structures we defined below, only gfold[7] can generate some extremely simple 3-crossing structures with $CH_w = H_w$. None of the aforementioned algorithms can generate k-crossing structures ($k \geq 4$).

2.2 Rectangle Tree and Complete Tree

We have observed that the classic grammar-based algorithm, Simple Linear Tree Adjoining Grammar[23], is incapable of predicting some highly-pseudoknotted structures (e.g k-crossing structures). To predict these structures, we introduce a new grammar called *Rectangle Tree Grammar*(RTG).

Let V be a finite set of alphabets and Σ be a set of terminal alphabets where $\Sigma \subset V$. Let γ be a *tree* over V such that
(i) each internal node must be labeled with a nonterminal symbol.
(ii) each leaf node can be labeled with a nonterminal or terminal symbol.
(iii) each internal node can have any number of children.
(iv) each edge can be labeled red or black.

$Y(\gamma)$ (ie. *yield* of tree) is defined as breadth-first search output of γ where all the nonterminal symbols are ignored.

A tree is *rectangle* if it satisfies all the conditions below:
(i) all the internal nodes should be labeled with nonterminal symbols.
(ii) there is only one leaf labeled with nonterminal symbol. This node is called N_4. The path from the root to N_4 is called the *backbone*.

(iii) there is only one *red edge* that defines the *insertion point* of this tree which is along the backbone.
(iv) considering the red edge $N_2 - N_3$, N_3 is the only child of N_2.
(v) the path from root to N_2 is the longest path in upper tree, the path from N_3 to N_4 is the longest path in bottom tree.

According to the definition above, a rectangle tree can be divided into two parts by splitting through red edge, the yield of upper tree is γ_U, the yield of bottom tree rooted at N_3 is γ_B. $Y(\gamma) = \gamma_U\gamma_B$, the position between γ_U and γ_B is called an *insertion point* (where other structures can be inserted in). Figure 6 is an example of a rectangle tree.

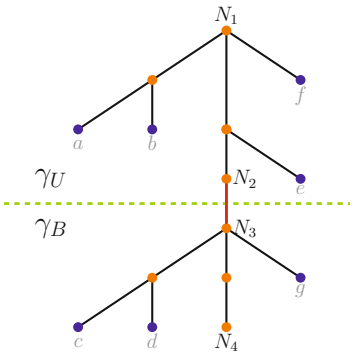


Fig. 6. A rectangle tree. Orange nodes and blue are labeled with nonterminal and terminal symbols, respectively. Its yield is $fabe, gcd$. The comma represents its insertion point.

A tree is *complete* if it satisfies all the following conditions:
(i) only one leaf (labeled as N_4) is labeled with nonterminal symbol.
(ii) there is no red edge, i.e., no more base pairs will be added.
(iii) the path from root to N_4 is the longest path in the tree.
By labeling the red edge in Figure 6, the rectangle tree becomes a complete tree. The yield is $fabegcd$. To predict the secondary structure of an RNA sequence, we compute the most probable rectangle tree whose yield is exactly the given sequence.

2.3 Grammar States

A rectangle tree or a complete tree has a unique state. As shown in Table 1, a *state* corresponds to the secondary structure represented by this tree. The first seven states are for rectangle trees. The remaining three are for complete trees.

2.4 Tree Operations

There are multiple ways to add bases into a tree. A *tree operation* defines how a single base, a base pair or another tree are allowed to be added. In this section,

Table 1. Grammar states and the corresponding secondary structures of RNA sequence $s_i \dots s_k \cup s_l \dots s_j$ or $s_i \dots s_j$

State	Structure Description
B	banding or gapped banding
$B3$	the structure is three banding, insertion point is the insertion point of the second banding.
BL	the structure is standard pseudoknot of degree k($k \geq 3$), insertion point is the insert point of the rightmost banding.
BR	the structure is standard pseudoknot of degree k($k \geq 3$), insertion point is the insert point of the leftmost banding.
BLR	the structure is standard pseudoknot of degree k($k \geq 4$), insertion point can be insertion point of any banding except the leftmost and rightmost one.
G	2-crossing structure, after another Cr operation, it will transit to state H.
H	k-crossing structure($k \geq 3$).
CPP	both s_i and s_j are paired bases.
CPS	s_i is a paired base, s_j is a single base.
CSP	s_i is a single base, s_j is a paired base.
CSS	both s_i and s_j are single bases.

we introduce tree operations from state to state so that it is clear why each operation is needed. For simplicity, we use S_1 to denote γ_U (the yield of upper tree) and S_2 to denote γ_B (the yield of bottom tree).

Gapped Banding (State B). A gapped banding is divided by insertion point into two parts: S_1 and S_2. Basically, base pairs and single bases of a banding is allowed to be added from outmost inwards. For rectangle trees, single bases can only be added at the end of S_1 (at N_2) or at the beginning of S_2 (at N_3). To obtain a gapped banding, the following tree operations are designed:

- $L23$: add a base pair X into the tree, where the head and tail of X are added to the end of S_1 and the beginning of S_2, respectively.
- $Ls2$: add a single base to the end of S_1.
- $Ls3$: add a single base to the beginning of S_2.

Three Banding (State $B3$). A basic idea to generate a three banding structure is to add gapped bandings in the twisted structure in a top-down manner. Besides $L23$, $Ls2$ and $Ls3$, there are three more legal operations to add a gapped banding (or a base pair) X:

- $L12$: add the head of X to the beginning of S_1; tail of X to the end of S_1.
- $L34$: add the head of X to the beginning of S_2; tail of X to the end of S_2.
- $L14$: add the head of X to the beginning of S_1; tail of X to the end of S_2.

Standard Pseudoknot of Degree k (State BL). To generate standard pseudoknot of degree $k(k \geq 3)$, we designed operation LR (Figure 7). Operation LR inserts the upper tree of α above N_1 of β and its bottom tree above N_2. At the same time, the insertion point is updated to insertion point of β. As a result, base pairs across upper tree and bottom tree in α and β would cross. Moreover,

the update of insertion point prevents base pairs in α from crossing base pairs in subsequent trees adjoined with LR later. After $(k-2)$ LR operations, standard pseudoknot of degree k is generated.

k-Crossing (State G and H). In k-crossing structures, without considering embedded substructures, all base pairs can be grouped into different crossing sets $CH_1 \ldots CH_k$, where CH_w $(\forall w \in [1,k])$ is a regular structure (state B), a standard pseudoknot (state BL, BR or BLR) or a three banding structure (state $B3$). Standard pseudoknots of state BL and BR have their insertion point within the leftmost and rightmost banding, respectively. Otherwise, if the insertion point comes from neither the leftmost nor the rightmost banding, this standard pseudoknot is in state BLR.

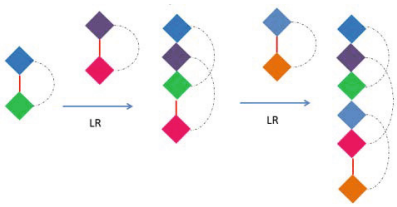


Fig. 7. After 2 LR operations over 3 gapped bandings, a standard pseudoknot of degree 4 is generated

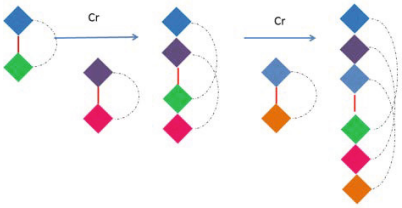


Fig. 8. A 3-crossing can be generated by 2 Cr operations

Operation LL is designed for state BR and BLR. An LL operation on rectangle tree ϵ with ζ inserts the upper tree of ζ above N_3 of ϵ and its bottom tree under N_4. Then by operating LR on $\alpha_1 LR \ldots LR\alpha_i$ with $\epsilon_1 LL \ldots LL\epsilon_j$, standard pseudoknot with insertion point in its $(i+1)$th banding is generated.

After generation of all the crossing sets, we designed the operation Cr to link them up. As is shown in Figure 8, operation Cr on rectangle tree α with β inserts upper tree of β under N_2 of α and bottom tree under N_4. So base pairs between upper tree and bottom tree in α and β would cross. After $(k-1)$ Cr operations, k-crossing can be generated.

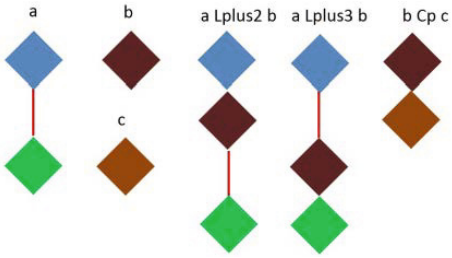


Fig. 9. $Lplus2$, $Lplus3$ and Cp for embedding and concatenation. a is a rectangle tree, b and c are complete trees

Embedding and Concatenation (State CPP). By applying operation Lm to label the red edge of a rectangle tree to black, a complete tree is generated. Embedding operations $Lplus2$ and $Lplus3$ insert a complete tree to N_2 and N_3 of a complete tree, respectively. The concatenation operation Cp can concatenate two complete trees. The above three operations are also explained in Figure 9. Note that if a rectangle tree α embeds (using $Lplus2$ or $Lplus3$) some complete trees, it becomes a new rectangle tree α'. And the state of α' remains the same as that of α.

Single Bases at Both Ends (State CPS, CSP and CSS). As required by RTG, gapped bandings are always generated at first. Afterwards, applying proper tree operations as defined above, these gapped bandings compose a more complicated structure. When no base pairs are to be inserted, Lm alters this rectangle tree (representing this complicated structure) to a complete tree of state CPP. Note that the first base s_i and the last base s_j must have been boundary bases of gapped bandings. When there are single bases at either end of an RNA sequence, operation $Ls1$ and $Ls4$ are used to add single bases to the beginning of S_1 and the end of S_2, respectively.

2.5 Grammar

A *RTG grammar rule* clarifies whether a specific operation is applicable to rectangle trees (in CPP, CPS, CSP or CSS state) or complete trees (in any other state). All the rules are tabulated in Table 2. In the table, α is a single base. (α,β) is a base pair. ($b1, b2$) and ($b3, b4$) are rectangle trees, where comma denotes their insertion points. (c), ($c1$), and ($c2$) represent complete trees.

After applying RTG grammar rules, the state of the predicted structure transits into another. All valid transitions defined by the grammar rules will be given in the full paper. For the dynamic programming algorithm for structure prediction and the parameter training, we follow the standard techniques (details will be given in the full paper).

Table 2. RTG rules

Operation	Input	output
Ls2_α	(b1,b2)*(α)	(s1α,s2)
Ls3_α	(b1,b2)*(α)	(b1,αb2)
L12	(b1,b2)*(b3,b4)	(b3b1b4,b2)
L23_α_β	(b1,b2)*(α,β)	(b1α,βb2)
L34	(b1,b2)*(b3,b4)	(b1,b3b2b4)
L14	(b1,b2)*(b3,b4)	(b3b1,b2b4)
LL	(b1,b2)*(b3,b4)	(b3,b1b4b2)
LR	(b1,b2)*(b3,b4)	(b1b3b2,b4)
Lplus2	(b1,b2)*(c)	(b1c,b2)
Lplus3	(b1,b2)*(c)	(b1,cb2)
Lm	(b1,b2)	(b1b2)
Cr	(b1,b2)*(b3,b4)	(b1b3,b2b4)
Cp	(c1)*(c2)	(c1c2)
Ls1	(c1)*(α)	(αc1)
Ls4	(c1)*(α)	(c1α)

3 Experiments

A total of 564 RNA sequences from 44 families were extracted from Rfam database for our experiments. All these families were classified into three sets D1, D2 and D3. D1 consists of regular structures (15 families). D2 contains standard pseudoknots of degree ≥ 3 (27 families). D3[2]comprises a set of 3-crossing structures (2 families). We carried out a 10-fold cross-validation on D1, D2 and D3 datasets separately. More specifically, take D1 as an example. In each round of validation, a total of 334 sequences in D1 were randomly partitioned into ten equal-size subsets. Out of these ten subsets, one subset was retained to test the model, while the other nine subsets were used to train this model. To eliminate variability, 10 rounds were performed using different partitions. The performance evaluated below is based on the average among 10 rounds. We compared the performance of our RTG method with seven popular softwares. For softwares that take multiple sequences as inputs, like TurboFold, CentroidAlifold and RNAalifold, we provided them with each family of sequences as an input. The performance was measured using positive predictive value (PPV) and sensitivity defined below. PPV $= \frac{\alpha}{\gamma}$ and sensitivity $= \frac{\alpha}{\beta}$, where α is the number of correctly reported base pairs, β is the total number of reported base pairs, and γ is the total number of base pairs in the Rfam.

Table 3. PPV and sensitivity of RTG and seven other softwares on $D1$, $D2$ and $D3$

Dataset	Software	PPV(%)	Sensitivity(%)	Software	PPV(%)	Sensitivity(%)
D1	pknotsRG[21]	62.21	28.97	gfold[7] [3]	54.5	24.6
D2		71.72	65.92		67.35	53.28
D3		19.78	10.13		11.00	6.70
D1	NUPACK[22]	51.41	24.36	CentroidAlifold[9]	93.53	36.73
D2		74.24	62.63		50.24	43.71
D3		37.52	18.88		24.89	12.38
D1	MXScarna[16]	75.54	38.76	RNAalifold[17]	77.69	45.60
D2		48.01	52.50		43.98	51.77
D3		13.73	7.30		23.40	24.88
D1	TurboFold[18]	75.46	34.01	**RTG**	**80.22**	**62.81**
D2		55.09	42.07		**80.56**	**75.09**
D3		20.80	10.74		**71.95**	**71.36**

Table 3 summarized the comparison of secondary structure prediction for RTG and seven other state-of-the-art programs. Our RTG program often outperforms other programs in terms of PPV and sensitivity. The experiment has revealed that 3-crossing dataset is hard to predict for other programs, which is consistent with our analysis of previous algorithms. However, the prediction of RTG program is accurate to a certain extent.

[2] There are only two families in Rfam with this complicated structures and one of the families (RF02032) is too long that our server does not have enough memory to handle it, we only extracted the 3-crossing structure (without considering embedded substructure) to run.

Apart from RTG, NUPACK[22] and RNAalifold[17] performed best in estimating the secondary structure for 3-crossing dataset. The performance regarding this dataset is further illustrated in Figure 10, which presents the predicted structure of NUPACK, RNAalifold and RTG over AE005174-2 as well as the trusted annotation in Rfam. The underlined parentheses($< - >, A - a$ and $B - b$) denotes the correctly predicted base pairs.

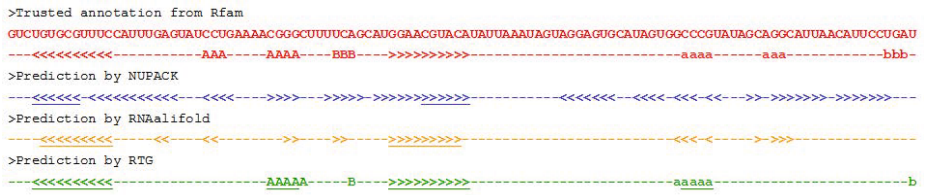


Fig. 10. An detailed comparison for predicting the structure of AE005174-2(RF00140) in Rfam

Evidently, RTG behaved the best with PPV = 87.5% and sensitivity = 70.0%. The PPV and sensitivity of RNAalifold were 56.2% and 45.0%, respectively. NUPACK reached even lower PPV and sensitivity. In addition to its high accuracy evaluated using PPV and sensitivity, RTG predicted a structure much more similar to the ground truth. RTG thought the secondary structure of AE005174-2 is a 3-crossing. Furthermore, it almost pointed out all the bandings correctly. Even for pairs denoted by $B - b$, the pairing position was very close. However, NUPACK and RNAalifold predicted it as regular structures, which was way far from its real structure.

References

1. Ten Dam, E., Pleij, K., Draper, D.: Structural and functional aspects of rna pseudoknots. Biochemistry 31(47), 11665–11676 (1992)
2. Lee, K., Varma, S., SantaLucia Jr., J., Cunningham, P.R.: In vivo determination of rna structure-function relationships: analysis of the 790 loop in ribosomal rna. Journal of Molecular Biology 269(5), 732–743 (1997)
3. Brierley, I., Digard, P., Inglis, S.C.: Characterization of an efficient coronavirus ribosomal frameshifting signal: requirement for an rna pseudoknot. Cell 57(4), 537–547 (1989)
4. Giedroc, D.P., Theimer, C.A., Nixon, P.: Structure, stability and function of rna pseudoknots involved in stimulating ribosomal frameshifting. J. Mol. Biol. 298, 167–185 (2000)
5. Lyngsø, R.B., Pedersen, C.N.S.: RNA pseudoknot prediction in energy-based models. Journal of Computational Biology 7(3-4), 409–427 (2004)

[3] There are some families gfold cannot run, so its PPV and sensitivity does not include these families.

6. Lyngsø, R.B.: Complexity of pseudoknot prediction in simple models. In: Díaz, J., Karhumäki, J., Lepistö, A., Sannella, D. (eds.) ICALP 2004. LNCS, vol. 3142, pp. 919–931. Springer, Heidelberg (2004)
7. Reidys, C.M., Huang, W.D., Andersen, F., Penner, J.E., Stadler, R.C., Nebel, P.F., Topology, M.E.: prediction of rna pseudoknots. Bioinformatics 27(8), 1076–1085 (2011)
8. Ren, J., Rastegari, B., Condon, A., Hoos., H.H.: Hotknots: Heuristic prediction of rna secondary structures including pseudoknots. RNA 11, 1494–1504 (2005)
9. Hamada, M., Kiryu, H., Sato, K., Mituyama, T., Asai., K.: Predictions of RNA secondary structure using generalized centroid estimators. Bioinformatics 25(4), 465–473 (2009)
10. Zakov, S., Goldberg, Y., Elhadad, M., Ziv-Ukelson., M.: Rich parameterization improves rna structure prediction. Journal of Computational Biology 18(11), 1525–1542 (2011)
11. Bindewald, E., Shapiro., T.K.B.: Cylofold: secondary structure prediction including pseudoknots. Nucleic Acids Research suppl.(W), 368–387 (2010)
12. Akutsu., T.: Dynamic programming algorithms for rna secondary structure prediction with pseudoknots. Discrete Applied Mathematics 104, 45–62 (2000)
13. Chen, H., Condon, A., Jabbari, H.: An o(n(5)) algorithm for mfe prediction of kissing hairpins and 4-chains in nucleic acids. Discrete Applied Mathematics 16(6), 803–815 (2009)
14. Dirks, R., Pierce, N.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. J. Comput. Chem. 24(13), 1664–1677 (2003)
15. Rivas, E., Eddy, S.R.: A dynamic programming algorithm for rna structure prediction including pseudoknots. J. Mol. Biol. 285, 2053–2068 (1999)
16. Tabei, Y., Kiryu, H., Kin, T., Asai, K.: A fast structural multiple alignment method for long rna sequences. BMC Bioinformatics 9(1), 33 (2008)
17. Bernhart, S.H., Hofacker, I.L., Will, S., Gruber, A.R., Stadler, P.F.: Rnaalifold: improved consensus structure prediction for rna alignments. BMC Bioinformatics 9(1), 474 (2008)
18. Harmanci, A.O., Sharma, G., Mathews, D.H.: Turbofold: iterative probabilistic estimation of secondary structures for multiple rna sequences. BMC Bioinformatics 12(1), 108 (2011)
19. Griffiths-Jones, S., Moxon, S., Marshall, M., Khanna, A., Eddy, S.R., Bateman, A.: Rfam: annotating non-coding rnas in complete genomes. Nucleic Acids Research 33(suppl. 1), D121–D124 (2005)
20. Cai, L., Malmberg, R.L., Wu, Y.: Stochastic modeling of rna pseudoknotted structures: a grammatical approach. Bioinformatics 19(suppl. 1), i66–i73 (2003)
21. Reeder, J., Steffen, P., Giegerich, R.: pknotsrg: Rna pseudoknot folding including near-optimal structures and sliding windows. Nucl. Acids Res. 35, 320–324 (2007)
22. Zadeh, J.N., Steenberg, C.D., Bois, J.S., Wolfe, B.R., Pierce, M.B., Khan, A.R., Dirks, R.M., Pierce, N.A.: Nupack: Analysis and design of nucleic acid systems. Journal of Computational Chemistry 32, 170–173 (2011)
23. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree adjoining grammars for rna structure prediction. Theoretical Computer Science 210(2), 277–303 (1999)