J. Math. Biol. (2008) 56:145–159
DOI 10.1007/s00285-007-0124-4

Mathematical Biology

Rapid ab initio prediction of RNA pseudoknots via graph tree decomposition

Jizhen Zhao · Russell L. Malmberg · Liming Cai

Received: 1 January 2007 / Published online: 29 September 2007

Abstract The prediction of RNA secondary structure including pseudoknots remains a challenge due to the intractable computation of the sequence conformation from nucleotide interactions under free energy models. Optimal algorithms often assume a restricted class for the predicted RNA structures and yet still require a high-degree polynomial time complexity, which is too expensive to use. Heuristic methods may yield time-efficient algorithms but they do not guarantee optimality of the predicted structure. This paper introduces a new and efficient algorithm for the prediction of RNA structure with pseudoknots for which the structure is not restricted. Novel prediction techniques are developed based on graph tree decomposition. In particular, based on a simplified energy model, stem overlapping relationships are defined with a graph, in which a specialized maximum independent set corresponds to the desired optimal structure. Such a graph is tree decomposable; dynamic programming over a tree decomposition of the graph leads to an efficient optimal algorithm. The final structure predictions are then based on re-ranking a list of suboptimal structures under a more comprehensive free energy model. The new algorithm is evaluated on a large number of RNA sequence sets taken from diverse resources. It demonstrates overall

The preliminary version of this paper appeared in the proceedings of the 6th Workshop on Algorithms for Bioinformatics (WABI 2006).

J. Zhao (✉) · L. Cai (✉)
Department of Computer Science, University of Georgia, Athens, GA 30602, USA
e-mail: jizhen@cs.uga.edu

L. Cai
e-mail: cai@cs.uga.edu

R. L. Malmberg
Department of Plant Biology, University of Georgia, Athens, GA 30602, USA
e-mail: russell@plantbio.uga.edu

sensitivity and specificity that outperforms or is comparable with those of previous optimal and heuristic algorithms yet it requires significantly less time than the compared optimal algorithms.

Keywords RNA secondary structure prediction · Pseudoknot · Thermodynamic energy · Tree decomposition · Tree width · Maximum independent set

1 Introduction

The secondary structure of an RNA molecule is formed due to short or long distance pairings between nucleotides in the sequence. Base pair regions either single, nested or parallel are called *stem-loops*; base pair regions crossing each other are called *pseudoknots* [29]. Pseudoknots are important structures in RNA molecules and often play important functional roles such as catalysis, RNA splicing, transcription regulation [2, 14, 24]. Knowing the secondary structures of RNA molecules is critical for determining their three dimensional structures and understanding their functions. Automated prediction of RNA secondary structure is thus in demand since it is expensive and time consuming to experimentally determine the structure.

It is computationally challenging to predict RNA secondary structure including pseudoknots. In particular, the problem of predicting RNA pseudoknots with the minimum free energy is provably NP-hard for the nearest neighbor model [16]. Practical approaches to cope with this computational challenge are either to restrict the class of pseudoknots under consideration or to employ heuristics in the algorithms. Optimal algorithms for restricted pseudoknot classes are usually thermodynamics-based extended from Zuker's algorithm for the prediction of pseudoknot-free structures [32]. In such algorithms, the predicted optimal structure of a single RNA sequence is the one with the global minimum free energy based on a set of thermodynamical parameters. For example, the recently developed PKNOTS [20] can handle the widest classes of pseudoknots. However, its time complexity $O(n^6)$ makes it infeasible to fold RNA sequences of a moderate length. The computational efficiency may be improved at the cost of further restricting the structure of pseudoknots [3, 16, 30], but still with the time complexity $O(n^5)$ or $O(n^4)$. Most such algorithms produce only the optimal solution, while suboptimal ones that may reveal the true structure are often ignored. However, the partition function approach, based on the calculation of equilibrium base-pairing probabilities, captures the contributions of all suboptimal structures [8].

On the other hand, computationally efficient heuristic methods have also been explored to allow unrestricted pseudoknot structures. Iterated loop matching (ILM) [22] is one such method. It finds the most stable stem, adds it to the candidate secondary structure and then masks off the bases forming the stem and iterates on the left sequence segments until no other stable stem can be found. One structure is reported at the end. Another algorithm, HotKnots [19], does the prediction in a slightly different way. It keeps multiple candidate structures rather than only one and builds each of them in a similar but more elaborate way. These methods can usually be fast, yet they often do not provide an optimality guarantee for the predicted structure or a quality measure on the predicted structure with respect to the optimal structure. Other heuristic methods based on genetic algorithms usually do not address the optimality issue either [1,7].

In this paper, we introduce a novel approach for the optimal prediction of RNA pseudoknots for which the structure is not restricted. Our method is based on a simplified free energy model without accounting for loop energies [18,22]. In this method, stable stems are selected from an RNA sequence as vertices of a graph; vertices are connected with edges if corresponding stems conflict (i.e., overlap) in their positions in the sequence. The optimal structure of an RNA sequence corresponds to a collection of non-conflicting stable stems, which can be found by seeking the maximum weighted independent set (WIS) from the graph. We observe that stable stems can be so selected that the resulting graph is of a moderately small tree width t. Based on a tree decomposition of the graph, a dynamic programming algorithm for WIS of the worst-case time complexity $O(2^t n)$ is obtained, where n is the number of vertices in the graph, at most quadratic in the length of the RNA sequence. This is an efficient prediction algorithm parameterized on the tree width t, which is usually small. In addition, the algorithm re-ranks a list of suboptimal structures based on the more comprehensive energy functions used in PKNOTS [20].

We implemented our algorithm TdFOLD and evaluated its performance on various RNA sequence sets from different sources. The test results showed high efficiency and high accuracy for our algorithm. TdFOLD was tested against PKNOTS, ILM and HotKnots on a set of 50 tRNA's, a set of 50 small RNA sequences containing pseudoknots with length ranging from 23 to 113, and a set of 11 large RNA's with length range from 210 to 412. The results showed that overall, in terms of the sensitivity and specificity of the prediction, TdFOLD outperforms the optimal algorithm PKNOTS and the heuristic algorithms ILM and HotKnots. In time efficiency, it outperforms PKNOTS and HotKnots, and is comparable with ILM.

Graph theoretic methods have previously been explored for RNA structure prediction [29]. Our method is different from the previous ones in two respects. Our graphs constructed from the RNA sequence contain vertices describing stems instead of nucleotides; making stem to be the smallest structural unit can greatly simplify the complexity of the problem. More importantly, our graph algorithm takes advantage of the tree decomposition technique on the formulated graphs. In fact, it has been demonstrated that the RNA secondary structure can be profiled with a conformational graph of small tree width [26]. The underlying graph constructed for the ab initio structure prediction is essentially an augmentation of the conformational graph in which additional vertices and edges are added only for the overlapping stems, thus inheriting the tree decomposability which makes the algorithm efficient.

We note that our algorithm, like other ab initio ones, is suitable for predicting the structure of single RNA sequence. When related structurally homologous sequences are available, the accuracy of RNA structure prediction can usually be improved through the use of comparative analysis. This uses the information of the covarying residues in a set of multiple sequences or additional phylogenetic relationship of these sequences and may produce the most reliable prediction for the consensus structure [10,13,15,23,28]. Because such methods inevitably involve multiple sources of data or computational tools, they usually rely on human intervention. Nevertheless, a fully automated comparative analysis process was proposed [9,10] for RNA consensus structure prediction. Due to its computational complexity, the process has only been implemented for pseudoknot-free RNAs [10]. With algorithm TdFOLD, and a tree decomposition based structure-sequence alignment algorithm we developed earlier [26], the efficient implementation of this fully automated process for pseudoknots becomes possible. The paper concludes with a presentation of how TdFOLD could be used in automatid comparative RNA analysis.

2 Methods and algorithm

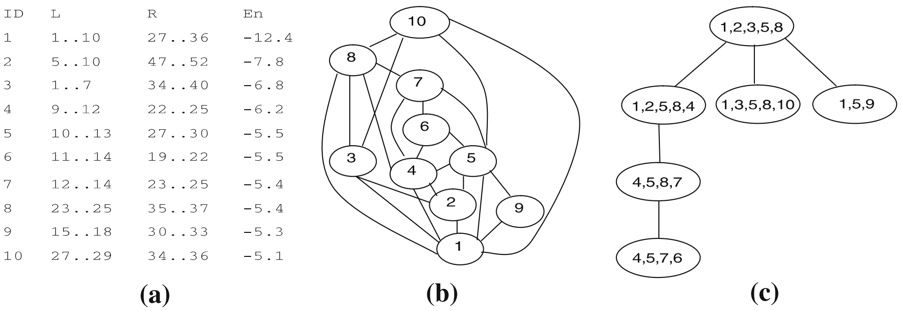
Given an RNA sequence, our algorithm first builds a pool of stable stems. A secondary structure is indeed a set of compatible stems (i.e., the stems do not share one or more bases between each other). Based on the observation that a set of compatible stems with minimum or near minimum total stem energies tend to be in the true structure, our algorithm first finds a number of secondary structures with minimum or near minimum total stem energies by a tree decomposition based procedure for a graph formed by the stable stems. These secondary structures are then reordered by counting the stem stabilizing and loop destabilizing energies together, and reported as the predicted structures.

2.1 Problem formulation

A (canonical) base pair is either a Watson–Crick pair (A-U or C-G) or wobble pair G-U. A *stem* is a set of stacked nucleotide base pairs on an RNA sequence s. In general a stem S can be associated with four positions (i^l, j^l, i^r, j^r), where $i^l < j^l < i^r < j^r$, on the sequence s such that (a) $(s[i^l], s[j^r])$ and $(s[j^l], s[i^r])$ are two canonical base pairs; and (b) for any two base pairs $(s[x], s[y])$, $(s[z], s[w])$ in the stem S, either $i^l \leq x < z \leq j^l$ and $i^r \leq w < y \leq j^r$, or $i^l \leq z < x \leq j^l$ and $i^r \leq y < w \leq j^r$. Region $s[i^l..sj^l]$ is the *left region* of the stem and $s[i^r..j^r]$ is the *right region* of the stem. Stem S is *stable* if the formation of its base pairs allows the thermodynamic energy $\Delta(S)$ of the stem to be below a predefined threshold parameter $E < 0$. Figure 1a shows all the stable stems in Ec_Pk4 with $E = -5$ kcal/mol, the fourth pseudoknot in *Escherichia coli* tmRNA [31], and their corresponding free energy values.

A *stem graph* $G_s = (V, E)$ can be defined for the RNA sequence s, where each vertex in V uniquely represents a stable stem on s, and E contains an edge between two vertices if and only if the corresponding two stems (a, b, c, d) and (x, y, z, w) conflict in their positions, i.e., one or both of the regions $s[a \ldots b]$ and $s[c \ldots d]$ overlap with at least one of the regions $s[x \ldots y]$ and $s[z \ldots w]$. Figure 1b shows the stem graph for Ec_Pk4 constructed according to the stable stems given in Fig. 1a. The stem graph is a weighted graph, with a weight on every vertex. Usually, the weight of a vertex can simply be absolute value of the thermodynamic energy $\Delta(S)$ of the stem S corresponding to the vertex. The weight may also be adjusted by scaling it (non-)linearly according to the length of the corresponding stem or the distance between the left and right regions of the stem.

The problem of predicting the optimal structure of the RNA then corresponds to finding a collection of non-conflicting stems from its stem graph which achieves the



ID	L	R	En
1	1..10	27..36	-12.4
2	5..10	47..52	-7.8
3	1..7	34..40	-6.8
4	9..12	22..25	-6.2
5	10..13	27..30	-5.5
6	11..14	19..22	-5.5
7	12..14	23..25	-5.4
8	23..25	35..37	-5.4
9	15..18	30..33	-5.3
10	27..29	34..36	-5.1

Fig. 1 **a** Ten stable stems in Ec_Pk_4, the fourth pseudoknot in *E. coli* tmRNA molecule, including their *left* and *right* regions, and thermodynamic energies; **b** stem graph for Ec_Pk_4; and **c** a tree decomposition of the stem graph with tree width 4

maximum total weight. This is exactly the same as the graph theoretic problem: finding the maximum weighted independent set (WIS) in the stem graph.

Note the optimality of the secondary structures depends on the total stem energies only (this was previously adopted by both the primitive method [18] and the more elaborate one [22] too). In order to rectify the possible bias caused by not counting the loop energies, we output optimal as well as a number of sub-optimal structures. These structures are then re-ordered according to the whole energies including stem stabilizing and loop destabilizing energies, and reported as the predicted structures.

2.2 Identifying stable stems

For our purpose, stable stems are defined according to a set of parameters (in addition to the energy threshold parameter E). In particular, a stem contains at least P base pairs; the loop length in between the left and right region of the stem is at least L; free energy using only Turner's base stacking energy parameters is at most E. Bulges within a stem are allowed, for which the stem essentially becomes a set of substems separated by the bulges. In addition, parameter T limits the minimum substem length, and parameter B limits the maximum bulge length. The thermodynamic energy $\Delta(S)$ of stem S is calculated by taking into account both the stacking energies and the destabilizing energies caused by bulges. The default values for parameters P, L, E, T, and B are set to 3, 3, -5, 3, 0 according to our previous experiments, but they may be adjusted according to different class of RNA's. We set B to 0 since it works well for a lot of RNAs. Besides 0, 1 and 2 are good choices for B for some other RNAs. We leave the choice to the users. A procedure similar to the one used in [13] is employed to identify all the stable stems. These stable stems are called the initial stable stem pool.

If two stems only share a few bases, we may resolve the conflict by considering some maximal substems of these two stems. For example, one stem A formed by $s[a \ldots a{+}9]$ and $s[b{-}9 \ldots b]$ conflicts with another stem B formed by $s[b{-}1 \ldots b{+}8]$ and $s[c \ldots c + 9]$, we could add two more stems formed by $s[a + 2 \ldots a + 9]$ and $s[b - 9 \ldots b - 2]$, $s[b + 11 \ldots b + 8]$ and $s[c \ldots c + 7]$, which are shortened from A and B respectively. This is controlled by a switch parameter S. If it is on, an extended stable stem pool will be built: all the stems in the initial stem pool will be imported into it; for each pair of conflict stems, get the maximal sub-stems that can resolve the conflict and meet the requirements defined by the parameters P, L, E, T, B and add them to the extended pool. If it is off, we just use the initial pool as all of the stable stems. According to our experiments, the use of the extended pool can improve the accuracy for some but not all of the RNAs. We also noticed that the longer a sequence is, the more the spurious stems will be produced. We incorporated some biological knowledge to reduce the spurious stems for long sequences, e.g. parameter adjusting according to the properties of some RNAs, not allowing G–U pairs at the ending of a stem.

2.3 Tree decomposition based algorithm

Definition [21] A *tree decomposition* of graph $G = (V, E)$ is a pair (T, X) if it satisfies:

1. $T = (I, F)$ is a tree with node set I and edge set F,
2. $X =\{X_i : i \in I, X_i \subseteq V\}$, $\bigcup_i X_i = V$ and $\forall u \in V, \exists i \in I$ such that $u \in X_i$,
3. $\forall (u, v) \in E, \exists i \in I$ such that $u, v \in X_i$,
4. $\forall i, j, k \in I$, if k is on the path that connects i and j in tree T, then $X_i \cap X_j \subseteq X_k$

The *width* of a tree decomposition (T, X) is defined as $\max_{i \in I} |X_i| - 1$. The *tree width* of the graph G is the minimum tree width over all possible tree decomposition of G. If T is restricted to be a path, we refer to (T, X) as a *path decomposition* and the best width over all of the path decompositions as the *path width* of G.

The tree decomposition is rooted in the deep graph minor theorems by Robertson and Seymour [21]. It provides a topological view on a graph and the tree width measures how much the graph is "tree-like". Figure 1c shows a tree decomposition for the stem graph given in Fig. 1b.

Many computationally intractable graph problems can be easily solved on graphs of small tree width. In particular, a large number of such graph problems, while intractable on general graphs, can be solved in linear time, given a tree decomposition of tree width $\leq t$, for a fixed t. Maximum weighted independent set (WIS) is one such problem [5]; it has the time complexity $O(2^t n)$. The factor 2^t is due to the dynamic programming enumeration, in each node of the tree decomposition, of all partial independent sets formed by the t vertices.

2.3.1 Algorithm details

Now we describe the tree decomposition based dynamic programming algorithm that finds the maximum weighted independent set from the stem graph $G = (V, E)$. It assumes a binary tree decomposition (T, X), where $X = \cup X_{i=1}^m$, for the stem graph, where $m = O(|V|)$, $|X_i| = t$, for $i = 1, \ldots, m$. We only discuss the process for achieving the optimal solution. The technical details for getting suboptimal solutions are similar.

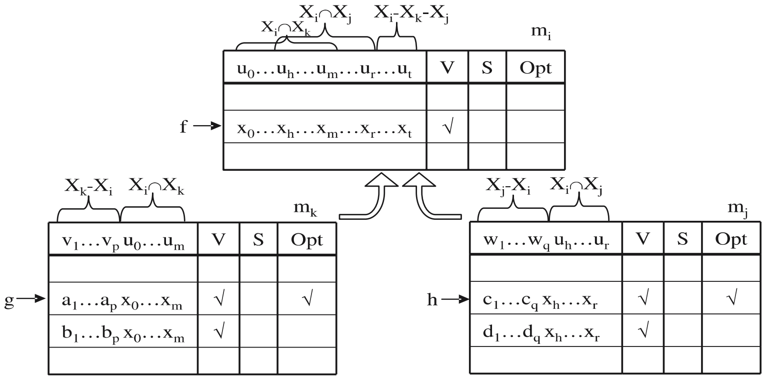


Fig. 2 Dynamic programming table construction over tree decomposition. Table m_i is computed also based on the computed tables m_k and m_j. Row $f = (x_0, \ldots, x_h, \ldots, x_m, \ldots, x_r, \ldots, x_t)$ in table m_i is computed from row g in table m_k and row h of table m_j, Row g is the optimal for columns $X_k - X_i$ given the value $(x_0, \ldots, x_m)$ for columns $X_k \cap X_i$. Similarly, row h is the optimal for columns $X_j - X_i$ given the value $(x_h, \ldots, x_r)$ for columns $X_j \cap X_i$

The algorithm constructs one dynamic programming table m_i for every tree node $X_i = \{v_1, \ldots, v_t\}$. Table m_i records all possible partial independent sets in the subgraph induced by the set of all the vertices in the subtree rooted at i of the tree decomposition. There are t columns in the table m_i, one for each vertex in the corresponding tree node X_i. Rows are the combinations of these vertices; a vertex is selected if and only if the corresponding column takes value 1. There are additionally three columns V, S, Opt in the table. Column V records whether each row represents a valid independent set, column S is the weight of the valid independent set represented by each row. Column Opt, more sophisticated, is explained in the following.

These tables are constructed in a bottom-up fashion, from leaves to the apex of the tree decomposition (see Fig. 2). Every table contains rows, with each being some combination of the vertices in the corresponding node. Column V is set to be 1 if the row represents a valid independent set. Column Opt is set 1 if and only if:

- The row represents a valid independent set.
- S in this row is optimal among all the rows with different choices in the columns corresponding to the vertices in $X_i - X_p$, given the chosen values same as this row in the columns corresponding to the vertices in $X_i \cap X_p$, where node p is the parent of node i.

Column S is set differently based on whether the current node is a leaf or an internal node in the tree decomposition. For a leaf node, S is 0 if the row is not a valid independent set; otherwise S is the corresponding weight of the set. For an internal node i that has two children j and k whose tables m_j, m_k have been computed, for each row in table m_i, column S is computed as $S = w_1 + w_2 + w_3 - w_4$, where

- w_1 is the weight of the row in table m_j with the same combination in the columns corresponding to the vertices in $X_j \cap X_i$ that has column $Opt = 1$;
- w_2 is the weight of the row in table m_k with the same combination in the columns corresponding to the vertices in $X_k \cap X_i$ that has column $Opt = 1$;

- w_3 is the weight of the independent set formed by the choices in columns corresponding to the vertices in $X_i - X_j - X_k$; and
- w_4 is the weight of the independent set formed by the same combination in the columns corresponding to the vertices in $X_i \cap X_j \cap X_k$.

In implementation, the computation of table T_i of node i does not enumerate all combinations of vertices in X_i. Instead, in general, a greedy algorithm is used to partition set X_i into a collection of cliques. Consider the sequence as a straight line and the left (right) region of a stem as an interval. Let all the left regions of the stable stems included in the tree node form an interval graph. Choose an interval (left region) with the right end at the leftmost position among all of the intervals, record all the intervals that overlap with this interval as a clique and remove them, recursively call on the interval graph left until it is empty. A linear time in t is enough for this procedure. Once the collection of cliques is obtained for X_i, combinations of vertices are only considered by taking at most one vertex from every clique.

A similar technique can also be used to further improve the efficiency of the algorithm for some long sequences: we build the stem graph based on the left regions of the stems, then we use a similar procedure to the above one to build a path decomposition of the graph. Each node of the path decomposition is indeed a clique consisting of overlapping half stems. We only need to choose zero or one half stem from each node. During the dynamic programming, each combination is associated with C (a user defined number, default 4000) suboptimal partial solutions rather than the optimal one only, and the conflicts caused by the right regions are resolved during the dynamic programming.

2.3.2 Tree decomposition of stem graph

Finding the optimal tree decomposition is NP-hard [4], we use a simple, fast heuristic algorithm to produce a tree decomposition for the given stem graph. This algorithm is based on a heuristic method for greedy fill-in [12]. This method will produce a tree decomposition with small tree width but not necessary the optimal one, i.e., the tree width of the tree decomposition might be greater than the tree width of the stem graph. Note this will not affect the optimal property of the tree decomposition based method described above since along any tree decomposition the optimal solution could be found.

2.3.3 Reordering suboptimal structures

The list of candidate structures, including the optimal and the suboptimal ones, are reordered based on a more sophisticated energy model. In particular, we recalculate the free energy for each of the candidate structures obtained from the previous step using a procedure implemented in [19] according to the energy model in [17,25] together with the one in [8], which take the stem stabilizing energies, loop destabilizing, and pseudoknot energies into account.

Table 1 Test set one: sequence IDs of 50 tRNAs, taken from [27]

GA0001	GA1262	GA2492	GA3755	GA4966	GC2866	GD1723	GD5199	GE2095	GE4739
GF1407	GF4687	GG0841	GG2136	GG3917	GH0128	GH4536	GI1748	GI4502	GK1078
GK4537	GM0313	GM2284	GM4471	GM5945	GN2837	GP1341	GP3879	GP5312	GQ2684
GR0044	GR0793	GR1516	GR2309	GR3541	GR4508	GR4705	GR4740	GR5278	GT0109
GT1418	GT4178	GT5273	GV0579	GV1734	GV4391	GV5554	GW1796	GW5332	GY4135

Table 2 Test set two: sequence IDs of 50 small pseudoknotted RNAs (with their reference citations)

Sequence type	Sequence IDs
mRNA	Bt-PrP, Ec_alpha, Ec_S15, Hs-PrP, T4_gene32 [31]
tmRNA	Lp_PK1, Ec_PK1, Ec_PK4 [31]
Ribozymes	satRPV, Tt-LSU-P3P7, Bp_PK2 [31]
Viral tRNA like	OYMV, APLV, CGMMV, SBWMV1, BSMVbeta, CGMMV_PKbulge, ORSV-S1, AMV3 [31]
Viral 3′ UTR	BSBV3, TMV-L_UPD-PK3, STMV_UPD1-PK3, BVQ3_UPD-PKb, BSBV1_ ,UPD-PKc, PSLVbeta_UPD-PK1, PSLVbeta_UPD-PK3, SBWMV1_UPD-PKb [31]
Viral ribosomal	minimal IBV, MMTV, MMTV-vpk, pKA-A, BWYV, SRV-1, T2_gene32[11];
RNA shifting signals	EIAV, PLRV-S [31]
Ribozymes	HDV-It_ag [31]
Telomerase RNA	T.the_telo [31]
Aptamers	NGF-L6 [31]
rRNA	Sc_18S-PKE21-7 [31]
Antizyme ribosomal	
Frame shifting site	Rr_ODCanti [31]
Viral RNA	PSIV_IRES [31]; TYMV, TMV.L, TMV.R [19]
HIV-1-RT ligand RNA	HIVRT32, HIVRT322, HIVRT33 [19]
Hepatitis virus ribozyme	HDV, HDV_anti [19]

3 Evaluation results

3.1 Data sets

In order to test the effectiveness and the efficiency of our algorithm, we evaluated it on small and large, pseudoknotted and pseudoknot-free RNAs. We used three sets of RNA sequences. The first set consists of 50 tRNAs, shown in Table 1, with lengths ranging from 71 to 79 (with the average 75). The second set consists of 50 small RNA sequences or sequence segments with pseudoknot structures (see Table 2) of lengths ranging from 23 to 113 (with the average 53). The third set consists of 11 large pseudoknot or pseudoknot free RNA sequences (see Table 3) of lengths ranging from 210 to 412 (with the average 344).

Table 3 Test set three: large RNAs containing pseudoknots (with their reference citations)

Sequence type	Sequence IDs
RNaseP RNA	A. ferrooxidans, A. laidlawii (pseudoknot free), A. tum, B. anthracis, B. halodurans, CPB147, D. desulfuricans, EM14b-9, E. thermomarinus, T. roseum [6]
Telomerase RNA	telo.human [7]

3.2 Experiment details

We compared the performance of our algorithm TdFOLD and that of algorithms PKNOTS [20], ILM [22], and HotKnots [19]. We chose the optimal algorithm PKNOTS since it can deal with the most comprehensive type of pseudoknot, while some newer algorithms can only deal with very restricted kinds of pseudoknots. We ran all these algorithms on the tRNAs and the set of small pseudoknot RNAs, and ran all but PKNOTS on the set of large RNAs. We evaluated both accuracy and efficiency of these algorithms. The accuracy is measured in both sensitivity and specificity. Let RP be the number of base pairs in the real structure, TP (true positive) be the number of correctly predicted base pairs and FP (false positive) be the number of predicted base pairs that do not exist as real structures. We define SE (sensitivity) as TP/RP, and SP (specificity, also called as positive predictive value) as $TP/(TP + FP)$. The perfect prediction should yield 1 for both sensitivity and specificity values.

For tRNA, the pseudoknot option for PKNOTS was turned off since it affects the predictions little for tRNA based on some previous tests. For TdFOLD, parameters were set to default values and the number of output solutions was set to 40 for tRNAs and small pseudoknotted RNAs. For HotKnots and TdFOLD, we only collect the top prediction result from the program output. We also determine the number of best predictions for each program on all data sets. Here, we say that a program has the *best prediction* for the secondary structure of an RNA if the sensitivity or specificity of the prediction is not worse than any of the predictions from other programs.

The experiments were run on a PC with 2.8 GHz Intel(R) Pentium 4 processor and 1-GB RAM, running RedHat Enterprise Linux version 4 AS. Running times were measured using the "time" function. The testing results are summarized in Tables 4, 5, and 6. Due to space limitations, we omit the data for each tRNA and each small RNA with pseudoknots.

Table 4 Summary of testing results on 50 tRNAs

	TdFOLD			HotKnots			ILM			PKNOTS		
	SE	SP	*T*	SE	SP	*T*	SE	SP	*T*	SE	SP	*T*
Min	0.33	0.29	0.26	0.33	0.25	0.57	0.33	0.25	0.01	0	0	0.11
Max	1.00	1.00	1.37	1.00	1.00	8.32	1.00	1.00	0.15	1.00	1.00	0.24
Average	0.81	0.75	0.54	0.72	0.66	3.33	0.75	0.61	0.03	0.78	0.73	0.41

SE sensitivity, *SP* specificity, *T* time (seconds)

Table 5 Summary of testing results on small pseudoknotted RNAs

	TdFOLD			HotKnots			ILM			PKNOTS		
	SE	SP	*T*	SE	SP	*T*	SE	SP	*T*	SE	SP	*T*
Min	0	0	0.04	0	0	0.05	0	0.25	0.001	0	0	0.27
Max	1.00	1.00	0.57	1.00	1.00	57.0	1.00	1.00	0.05	1.00	1.00	>1hr
Average	0.76	0.79	0.36	0.69	0.72	5.84	0.73	0.69	0.03	0.78	0.73	1066

SE sensitivity, *SP* specificity, *T* time in seconds (if not otherwise noted)

Table 6 Summary of testing results on large RNAs (pseudoknots)

		TdFOLD			HotKnots			ILM		
	L	SE	SP	*T*	SE	SP	*T*	SE	SP	*T*
A. ferrooxidans	344	0.42	0.38	14.5	0.43	0.38	169	0.38	0.36	0.71
A. laidlawii	316	0.45	0.35	3.13	0.68	0.52	655	0.57	0.45	1.15
A. tum.RNaseP	400	0.58	0.7	0.46	0.61	0.63	1432	0.77	0.82	1.24
B. anthracis	408	0.43	0.57	0.47	0.35	0.32	222	0.43	0.42	1.49
B. halodurans	412	0.56	0.6	0.48	0.53	0.44	13483	0.56	0.53	0.99
CPB147	298	0.18	0.17	7.62	0.59	0.52	170	0.30	0.25	0.85
D. desulfuricans	360	0.62	0.61	5.59	0.58	0.52	159	0.48	0.45	0.63
EM14b-9	355	0.72	0.73	4.58	0.66	0.51	29710	0.67	0.61	0.86
E. thermomarinus	331	0.46	0.41	1.86	0.65	0.56	148	0.54	0.47	0.83
telo.human	210	0.86	0.69	3.17	0.28	0.18	157	0.70	0.55	0.81
T. roseum	350	0.62	0.63	1.83	0.24	0.19	2714	0.51	0.45	1.10
Average	344	0.54	0.53	3.97	0.54	0.49	4456	0.51	0.44	0.97

L length, *SE* sensitivity, *SP* specificity, *T* time in seconds

3.3 Prediction accuracy

Tables 4, 5 and 6 summarize the testing results for different programs on the three RNA data sets. In Tables 4 and 5, we also record the minimum (maximum) value among the predictions for all of the sequences as min (max) since we did not show the data for each sequence. For example, among the 50 predictions for tRNAs from TdFOLD, in term of sensitivity, the worst one (min) is 0.33 and the best (max) one is 1.

Table 4 shows that TdFOLD has sensitivity 0.81 and specificity 0.75 on average for the tRNA prediction, which are slightly better than PKNOTS and significantly better than ILM and HotKnots. Table 5 shows that, for the small pseudoknotted RNAs, TdFOLD has average sensitivity 0.76, which is less than PKNOTS but greater than ILM and HotKnots. On the other hand, TdFOLD has average specificity 0.79, which outperforms all the others. TdFOLD is slightly better in overall accuracy than PKNOTS, which reports the optimal structure according its sophisticated energy model. Table 6 shows the prediction accuracy on the large RNA's. TdFOLD maintains the same sensitivity (0.54) as HotKnots, which is slightly better than ILM. TdFOLD has the highest specificity.

As we mentioned, a program has the best prediction for an RNA sequence if the sensitivity or specificity of the prediction is not worse than any of the predictions by other programs. For example, for tRNA, in sensitivity, TdFOLD, PKNOTS, HotKnots, ILM have 30, 29, 15, and 20 best predictions, respectively. In specificity, they have 23, 27, 19, and 4, respectively. For small pseudoknotted RNAs, they have 29, 31, 27, 24 for sensitivity and 29, 26, 22, 20 for specificity, respectively. For large RNAs, TdFOLD, HotKnots and ILM have 6, 3, 4 for sensitivity and 7, 1, 4 for specificity. Thus, TdFOLD performs better than, or as well as, the other programs tested.

We also noticed that different initial stem pools could affect the prediction results. The predictions based on the initial pools according to the parameter values mentioned in this paper may not necessarily be the best ones. It could be straightforward to improve the accuracy of our algorithm: generate multiple initial stem pools for an RNA sequence according to different parameters values; run our current version of the algorithm to produce multiple sets of predictions; pick up a number of the best ones from the multiple sets of the predictions according to the full energy model. Given the efficiency of our algorithm, such an extension is reasonable. Since it could be prejudicial to choose any one particular set of values of the parameters, this extension could also rectify the bias caused by choosing only one parameter value set at some degree.

3.4 Time efficiency

Efficiency comparisons are also given in Tables 4, 5 and 6 on each data set, respectively. For tRNA's, the average running time of 0.54 seconds for TdFOLD is slower than the average 0.03 of ILM and the average 0.41 of PKNOTS but faster than the average 3.33 of HotKnots. This is not a surprise because we turned the pseudoknot option off for PKNOTS. For small pseudoknotted RNA's, TdFOLD is slower than ILM (0.36 vs. 0.03 seconds), while much faster than HotKnots and PKNOTS (5.84 and 1,066 s). For large RNA sequences, it is comparable (slightly slower) than ILM (3.97 vs. 0.97 s) while much faster than HotKnots (4,456 s) on average. In general, the speed of TdFOLD is comparable to ILM and much faster than PKNOTS and HotKnots.

3.5 Suboptimal structures

According to Tables 4, 5 and 6, all of the programs could predict some sequences (different for each program) totally wrong (zero sensitivity and/or specificity). This reveals that the available thermodynamic parameters for RNA secondary structures may not be optimal for all RNA classes. Thus it is hard to guarantee that the structure with the minimum free energy is the true structure. This makes the output of a list of low energy suboptimal structures a valuable feature of a structure prediction algorithm. Among the previous algorithms, only HotKnots can output suboptimal structures but with a substantial sacrifice in efficiency. Some other existing dynamic programming based algorithms can list suboptimal structures but only work for restricted classes of pseudoknots. In contrast, our algorithm can output suboptimal predictions of RNAs with any class of pseudoknots without using much more time than reporting the optimal structure.

The prediction results for 23 tRNAs and 19 short pseudoknotted RNAs are improved by considering the top five structures, rather than only the top one among the 40 output predictions for each sequence. By "improved" we mean that there is at least one suboptimal prediction with both the sensitivity and specificity better than (or the same as) those of the optimal prediction. If there is more than one prediction improved over the top one, we choose the best among all the improved. Some statistics are shown below.

For the 50 tested tRNAs, the average sensitivity and specificity are improved to 0.91 and 0.85, respectively, from 0.81 to 0.75. The perfect predictions also increase to 15 from 11. For the short pseudoknotted RNAs, although the number of perfect predictions remains unchanged, the sensitivity and specificity increase to 0.81 and 0.85 from 0.76 and 0.79, respectively. An extreme example is CGMMV (the $3'$ end of cucumber green mottle mosaic virus RNA with tRNA-like pseudoknotted structure): the top one prediction was totally wrong, while the second prediction has sensitivity 0.57 and specificity 0.4. For the large RNA sequences, some of the suboptimal predictions also improved over the optimal ones. We did not collect the statistics due to the small data set.

4 Application in automated comparative RNA analysis

We now discuss an application of our algorithm to automated comparative RNA analysis. We first note that our algorithm, like other ab initio ones, is suitable for predicting the structure of single RNA sequence. When related structurally homologous sequences are available, the accuracy of RNA structure prediction can usually be improved through the use of comparative analysis. This usually uses the information of the covarying residues in a set of multiple sequences or additional phylogenetic relationship of these sequences and thus may produce the most reliable prediction for the consensus structure [10,13,15,23,28]. Because such methods inevitably involve multiple sources of data or computational tools, they often rely on human intervention.

Nevertheless, a fully automated comparative analysis process exists [9,10] for RNA consensus structure prediction that iterates between the following two steps to refine the prediction: (a) build an optimal (or nearly optimal) structure model given the current multiple alignment; and (b) build a multiple alignment given the current structure model. The algorithm for step (b) is structure-sequence alignment that can align every sequence in the set to the structure model. In the implementation for pseudoknot-free RNAs, covariance models were used for the structure model and the corresponding alignment algorithm is CYK-based [10]. Once every sequence in the set is aligned to the structure model, a multiple structural alignment (and thus a structure model) is actually generated. Therefore step (a) is only difficult in the initial step to produce a structure alignment without the structure model. In the work for pseudoknot-free RNAs [10], the initial step is to do multiple sequence alignment and to compute the mutual information content between every pair of aligned columns. A folding algorithm is then used to predict the consensus structure, yielding the initial model for the process [9]. For RNA pseudoknots, both algorithms for steps (a) and (b) can be computationally intensive; the implementation remains a computational challenge.

The tree decomposable model and tree decomposition based techniques make it possible to implement efficiently the automated comparative analysis process. Based on an earlier work of ours, pseudoknots can be profiled with the conformational graph model [26] of small tree width; the efficient optimal structure-sequence alignment developed is ideal for step (b). In addition, the algorithm introduced in this paper can be employed to construct an initial structure model for multiple RNAs. We discuss some technical details in the following.

Given a set of multiple RNA sequences, a multiple sequence alignment can be obtained. As it was done for pseudoknot-free RNAs, the mutual information content $M_{i,j}$ can be computed for every pair of aligned columns i, j. Which is defined as the relative entropy

$$M_{i,j} = \sum_{x_i, y_j \in \{A,C,G,U\}} f(x_i, y_j) \log \frac{f(x_i, y_i)}{f(x_i) f(y_j)}$$

where $f(x_i, y_j)$ is the frequency for nucleotides x_i, y_j to occur in pair in these two columns i, j, and $f(x_i)$ and $f(y_j)$ are for independent occurrences. The multiple alignment can be regarded as a "generic sequence" consisting of columns as "nucleotides". The pairwise interactions between columns result in a conformation structure of the "generic sequence", yielding a consensus structure for the multiple sequences. Therefore, we can use our structure prediction algorithm TdFOLD to predict the structure of the "generic sequence" using the mutual information content $M_{i,j}$ as "pairing energy" between columns i and j.

5 Conclusion

In this paper, we presented a tree decomposition based fast RNA folding algorithm, which is efficient, accurate, not limited to any specific class of pseudoknots, and can report a list of suboptimal structures. Combined with an efficient structure-sequence alignment algorithm we developed earlier [26], it also can be used to implement an automated comparative RNA structure analysis process that can infer the pseudoknot consensus structure from a set of unaligned, large RNA sequences.

References

1. Abrahams, J., van den Berg, M., van Batenburg, E., Pleij, C.: Prediction of RNA secondary structure, including pseudoknotting, by computer simulation. Nucleic Acids Res. **18**, 3035–3044 (1990)
2. Adams, P.L., Stahley, M.R., Kosek, A.B., Wang, J., Strobel, S.A.: Crystal structure of a self-splicing group i intron with both exons. Nature **430**, 45–50 (2004)
3. Akutsu, T.: Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. Discrete Appl. Math. **104**, 45–62 (2000)
4. Bodlaender, H.L.: Classes of graphs with bounded tree-width. Tech. Rep. RUU-CS-86-22, Dept. of Computer Science, Utrecht University, the Netherlands (1986)
5. Bodlaender, H.L.: Dynamic programming algorithms on graphs with bounded tree-width. In: Proceedings of the 15th International Colloquium on Automata, Languages and Programming, pp. 105–119. Springer Verlag, Lecture Notes in Computer Science, vol. 317, (1987)
6. Brown, J.: The ribonuclease *p* database. Nucleic Acids Res. **27**, 314 (1999)

7. Chen, J.-H., Le, S.-Y., Maize, J.V.: Prediction of common secondary structures of RNAs: a genetic algorithm approach. Nucleic Acids Res. **28**(4), 991–999 (2000)
8. Dirks, R., Pierce, N.: A partition function algorithm for nucleic acid secondary structure including pseudoknots. J. Comput. Chem. **24**, 1664–1677 (2003)
9. Durbin, R., Eddy, S.R., Krogh, A., Mitchison, G.J.: Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids. Cambridge University Press, Cambridge (1998)
10. Eddy, S.R., Durbin, R.: RNA sequence analysis using covariance models. Nucleic Acids Res. **22**, 2079–2088 (1994)
11. Giedroc, D., Theimer, C., Nixon, P.: Structure, stability and function of RNA pseudoknots involved in stimulating ribosomal frame shifting. J. Mol. Biol. **298**, 167–185 (2000)
12. Hicks, I.V., Koster, A.M.C.A., Kolotoglu, E.: Branch and tree decomposition techniques for discrete optimization. In: Tutorials in Operations Research: INFORMS, New Orleans 2005 (2005)
13. Ji, Y., Xu, X., Stormo, G.D.: A graph theoretical approach for predicting common RNA secondary structure motifs including pseudoknots in unaligned sequences. Bioinformatics **20**(10), 1591–1602 (2004)
14. Ke, A., Zhou, K., Ding, F., Cate, J.H., Doudna, J.A.: A conformational switch controls hepatitis delta virus ribozyme catalysis. Nature **429**, 201–205 (2004)
15. Knudsen, B., Hein, J.: Pfold: RNA secondary structure prediction using stochastic context-free grammars. Nucleic Acids Res. **31**(13), 3423–3428 (2003)
16. Lyngso, R.B., Pedersen, C.N.S.: RNA pseudoknot prediction in energy-based models. J. Comput. Biol. **7**(3–4), 409–427 (2000)
17. Mathews, D.H., Sabina, J., Zuker, M., Pederson, C.N.S.: Expanded sequence dependence of the thermodynamic parameters improves prediction of RNA secondary structure. J. Mol. Biol. **288**, 911–940 (1999)
18. Nussinov, R., Pieczenik, G., Griggs, J., Kleitman, D.: Algorithms for loop matchings. SIAM J. Appl. Math. **35**, 68–82 (1978)
19. Ren, J., Rastegart, B., Condon, A., Hoos, H.H.: HotKnots: heuristic prediction of RNA secondary structures including pseudoknots. RNA **11**, 1194–1504 (2005)
20. Rivas, E., Eddy, S.R.: A dynamic programming algorithm for RNA structure prediction including pseudoknots. J. Mol. Biol. **285**, 2053–2068 (1999)
21. Robertson, N., Seymour, P.D.: Graph minors ii. Algorithmic aspects of tree width. J. Algorithms **7**, 309–322 (1986)
22. Ruan, J., Stormo, G.D., Zhang, W.: An iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. Bioinformatics **20**(1), 58–66 (2004)
23. Sakakibara, Y., Brown, M., Hughey, R., Mian, I.S., Sjölander, K., Underwood, R.C., Haussler, D.: Stochastic context-free grammars for tRNA modeling. Nucleic Acids Res. **22**, 5112–5120 (1994)
24. Schimmel, P.: RNA pseudoknots that interact with components of the translation apparatus. Cell **58**(1), 9–12 (1989)
25. Serra, M.J., Turner, D.H., Freier, S.M.: Predicting thermodynamic properties of RNA. Meth. Enzymol. **259**, 243–261 (1995)
26. Song, Y., Liu, C., Malmberg, R.L., Pan, F., Cai, L.: Tree decomposition based fast search of RNA structures including pseudoknots in genomes. In: Proceedings of 2005 Computational System Bioinformatics Conference, pp. 223–234. IEEE Computer Society (2005)
27. Sprinzl, M., Horn, C., Brown, M., Ioudovitch, A., Steinberg, S.: Compilation of tRNA sequences and sequences of tRNA genes. Nucleic Acids Res. **26**, 148–153 (1998)
28. Steffen, P., Voss, B., Rehmsmeier, M., Reeder, J., Giegerich, R.: Rnashapes: an integrated RNA analysis package based on abstract shapes. Bioinformatics **22**(4), 500–503 (2006)
29. Tabaska, J., Cary, R., Gabow, H., Stormo, G.: An RNA folding method capable of identifying pseudoknots and base triples. Bioinformatics **14**(8), 691–699 (1998)
30. Uemura, Y., Hasegawa, A., Kobayashi, S., Yokomori, T.: Tree adjoining grammars for RNA structure prediction. Theor. Comput. Sci. **210**, 277–303 (1999)
31. van Batenburg, F., Gultyaev, A., Pleij, C., Ng, J., Oliehoek, J.: Pseudobase: a database with RNA pseudoknots. Nucleic Acids Res. **28**, 201–204 (2000)
32. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**(1), 133–148 (1981)