Tree Decomposition Based Fast Search of RNA Structures Including Pseudoknots in Genomes

Yinglei Song
Department of Computer Science
University of Georgia
Athens, GA 30602, USA
song@cs.uga.edu

Chunmei Liu
Department of Computer Science
University of Georgia
Athens, GA 30602, USA
chunmei@cs.uga.edu

Russell Malmberg
Department of Plant Biology
University of Georgia
Athens, GA 30602, USA
russell@plantbio.uga.edu

Fangfang Pan
Department of Plant Biology
University of Georgia
Athens, GA 30602, USA
fpan@plantbio.uga.edu

Liming Cai
Department of Computer Science
University of Georgia
Athens, GA 30602, USA
cai@cs.uga.edu

Abstract

Searching genomes for RNA secondary structure with computational methods has become an important approach to the annotation of non-coding RNAs. However, due to the lack of efficient algorithms for accurate RNA structure-sequence alignment, computer programs capable of fast and effectively searching genomes for RNA secondary structures have not been available. In this paper, a novel RNA structure profiling model is introduced based on the notion of a conformational graph to specify the consensus structure of an RNA family. Tree decomposition yields a small tree width t for such conformation graphs (e.g., $t = 2$ for stem loops and only a slight increase for pseudoknots). Within this modelling framework, the optimal alignment of a sequence to the structure model corresponds to finding a maximum valued isomorphic subgraph and consequently can be accomplished through dynamic programming on the tree decomposition of the conformational graph in time $O(k^t N^2)$, where k is a small parameter, and N is the size of the profiled RNA structure. Experiments show that the application of the alignment algorithm to search in genomes yields the same search accuracy as methods based on a Covariance model with a significant reduction in computation time. In particular, very accurate searches of tmRNAs in bacteria genomes and of telomerase RNAs in yeast genomes can be accomplished in days, as opposed to months required by other methods.

The tree decomposition based searching tool is free upon request and can be downloaded at our site http://www.uga.edu/RNA-Informatics/software/index.php.

Keywords: RNA secondary structure profiling, Pseudoknot search, Tree decomposition, Covariance model

1. Introduction

Non-coding RNAs (ncRNAs) are biologically important and play fundamental roles in a variety of biological processes such as gene regulation, chromosome replication, and RNA modification [9, 22, 18]. Recently, with the large amount of available sequence data, homologous searching based on computational methods has become one of the important approaches to the identification of new ncRNAs [17, 25, 12]. The core part of such a search program is an algorithm that aligns a target sequence to an RNA profile. To optimally identify the structure of remote homologs, the profile needs to include conserved conformations caused by long distance nucleotide base pairs (stems) as well as sequence conservation.

Most existing RNA search programs [17, 14, 4, 19] are based on the Covariance model (CM) developed by Eddy and Durbin [6] which enables the profiling of base pairs as well as single nucleotides. While CM can achieve high accuracy on searching for pseudoknot-free structures, it cannot profile the crossing stems of a pseudoknot. In general, CM based search is computationally inefficient on structures with more than 300 nucleotides. For instance, the commonly used CYK structure-sequence alignment algorithm requires a computation time $O(N^4)$ for a profiled pseudoknot-free RNA containing N nucleotides [16]. To reduce the computation time needed for searching on long genomes or large sequence databases, a preprocessing step can be used to filter out portions of a genome which are unlikely to contain the desired pattern [2, 17, 27]. The filtration based methods can significantly reduce the search time but the amount of speedup may not be guaranteed. These techniques have yet to be applied to searches for structures containing pseudoknots.

To profile pseudoknot structures, a few models based on stochastic grammar systems have been proposed. Rivas and Eddy [23] introduced a formal grammar to describe the legal structures identified by their thermodynamics based pseudoknot prediction algorithm [24]. This grammar is based on a number of auxiliary symbols used to reorder the string generated by an otherwise context-free grammar (CFG). Our previous work [20] on the stochastic parallel communicating grammar system extended the conventional statistical CFG with a few additional regular grammar components. The crossing stems in a pseudoknot can be generated by a parallel and cooperative derivation of all grammar components in the system. In addition, Uemura *et al* [15] used tree adjoining grammars for pseudoknot modelling. However, for all these models the computation time and memory space costs needed to perform optimal structure-sequence alignment are $O(N^5)$ and $O(N^4)$ respectively. In practice, these models cannot be directly used for profiling and searching.

On the other hand, searching for pseudoknots may be significantly speeded up with heuristic approaches. For example, ERPIN, a search tool developed by Gautheret and Lambert [10], disassembles the secondary structure of an RNA family into separate stem loops. It scans the genome to search for possible hit locations for each stem loop structure and reports a hit when a combination of hit locations for different stem loops can conform with the overall structure. ERPIN does not allow gaps in the alignment and can therefore miss important remote homologs. Another approach, first proposed by Brown and Wilson [4] and further developed by us [19], models pseudoknots with the intersection of several SCFG or CM components. The optimal alignment score of a sequence is computed by combining the scores obtained from aligning the sequence to all components separately. This approach has the same drawback in computation time as CM based methods, therefore is not suitable for moderately large RNA structures.

In this paper, we introduce a novel RNA structure (including pseudoknot) profiling method that can lead to efficient techniques for structure-sequence alignment and thus very fast search programs. We profile an RNA structure with a conformational graph, in which each vertex represents a base region of a stem and each edge connects two base regions if they form a stem or they constitute the two ends of a loop. With this method, the optimal structure-sequence alignment corresponds to a generalized subgraph isomorphism (embedding) problem in which the guest graph is the conformational graph, usually of a naturally small tree width t. We develop a dynamic programming algorithm over the tree decomposition of the conformation graph, based on which, an optimal alignment can be found in time $O(k^t N^2)$ for a given integer parameter k. The value of k can be effectively determined by a statistical cut off and is also small in nature. Compared with the dynamic programming algorithm used in CM based search, our new algorithm is significantly faster.

We performed experiments on several ncRNA families to test the accuracy and efficiency of the searching algorithm. Our experiments showed that, using a significantly reduced amount of computation time, the searching algorithm based on this new model can achieve the same accuracy as the CM based searching does. Specifically, on average, the algorithm is about 24 and 50 times faster than CM based methods on searching for pseudoknot free sequences that contain around 90 and 150 nucleotides respectively. Our experiments also demonstrated an even more significant advantage of the algorithm over the CM based searching in computation time when the profiled RNAs contain pseudoknots. As a test of the model on real genomes, we used the algorithm to search for the tmRNA gene in two bacterial genomes, and the telomerase RNA gene on two yeast genomes. Both the tmRNA and the telomerase genes were very accurately detected on both genomes in days, a task that would have needed months of computation time if a CM based searching model has been used.

2. Methods and Models

We view the consensus secondary structure of an RNA family as a topological relation among basic structural units, each of which is a stem or a loop. Our new structure model consists of two components: a *conformational graph* that represents the relationship among all basic structural units, and a set of simple CMs and profile HMMs, each modelling a stem or a loop.

In the conformational graph H, each vertex defines either of the base pairing regions of some stem. The graph is a

mixed graph containing both directed and undirected edges. Each undirected edge connects two base pairing regions that form a stem. Two base regions are connected with a directed edge (from 5' to 3') if they are the two ends of a loop. Technically, we add two additional vertices s (called *source*) and t (called *sink*) to the graph. Figure 1(a) and (b) show the consensus structure of an RNA family and the corresponding conformational graph. A consensus structure is usually obtained from a multiple structural alignment of a family of RNAs whose structure information is known. Therefore, in addition to the conformational graph, statistical models such as CMs and profile HMMs can be constructed for all stems and loops involved in the structure.

In this framework, a *target sequence* is a segment in a (possibly long) genome sequence. We use the profile of each stem to scan the target sequence to identify all pairs of regions in the target sequence that have statistically significant scores of (structural) alignment with the stem profile. These pairs of regions are called *images* of the stem. We define *parameter* k to be the maximum number of images of a stem over all stems in the structure. k has two interesting properties. First, k is a function of a statistical cut-off value. For example, for any stem, the number of images scored above certain Z-score threshold is inversely proportional to the threshold value. Second, the value k is generally small in nature, especially when a more effective statistical cut-off is applied (see section 4).

Given the set of images of all profiled stems in the structure, an *image graph* can be constructed. Similar to the construction of a conformational graph, each vertex defines one of the two base pairing regions of some stem and each undirected edge connects two base pairing regions that form a stem, but now a directed edge connects every two base regions (5' to 3') so long as they do not overlap. Based on the construction, each vertex u in the conformational graph H can only be mapped to a specific set of k vertices in the image graph G, each of which is called an *image of the vertex* u. Figure 1(c) and (d) illustrate the mapping from stems to their images and the corresponding image graph constructed.

The optimal structure-sequence alignment between an RNA structure profile and a target sequence is equivalent to the following generalized subgraph isomorphism problem: given a conformational graph H and an image graph G, find an one-to-one mapping f from vertices in H to their images in a subgraph S of G such that

1. (u, v) is an edge in H if and only if $(f(u), f(v))$ is an edge in S,
2. for any set of vertices in G representing overlapping regions on the target sequence, at most one of them can be selected to the subgraph S, and
3. the total score achieves the maximum when calculated from the score sum of the simultaneous alignment of all regions selected by the mapping to the stems and loops in the profile.

The defined problem is an optimization problem, different from the classical subgraph isomorphism decision problem. Section 3 gives an optimal algorithm for this optimization problem.

Searching a genome for a desired structure is accomplished by scanning through it with a window of a length determined by the size of the profiled structure. For a target sequence within the window frame, the optimal structure-sequence alignment is performed. Locations of the window with statistically significant scores are considered hits.

3. Algorithms

In this section, we present the details of a tree decomposition based efficient parameterized algorithm for the optimal alignment of a target sequence to an RNA structure profile. For this purpose, we first review the concepts of tree decomposition and tree width.

Definition 3.1 ([26]) *Let $G = (V, E)$ be a graph, where V is the set of vertices in G, E denotes the set of edges in G. Pair (T, X) is a tree decomposition of graph G if it satisfies the following conditions:*

1. *$T = (I, F)$ defines a tree, the sets of vertices and edges in T are I and F respectively,*
2. *$X = \{X_i | i \in I, X_i \subseteq V\}$, 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 $u \in X_i$ and $v \in X_i$,*
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 tree width of the 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 decompositions of G.

Intuitively, in a tree decomposition of a graph, vertices are placed into a number of bags, each of which is represented by a node in the tree. A valid tree decomposition requires that every edge in the graph is "covered" by at least one tree node, and nodes that contains the same vertex must form a connected subtree of the tree. Figure 2 gives an example of a tree decomposition.

The notion of tree decomposition can be used to investigate the "tree-like" property of graphs. For graphs of small tree width t, many optimization problems can be solved via dynamic programming over tree decomposition in time

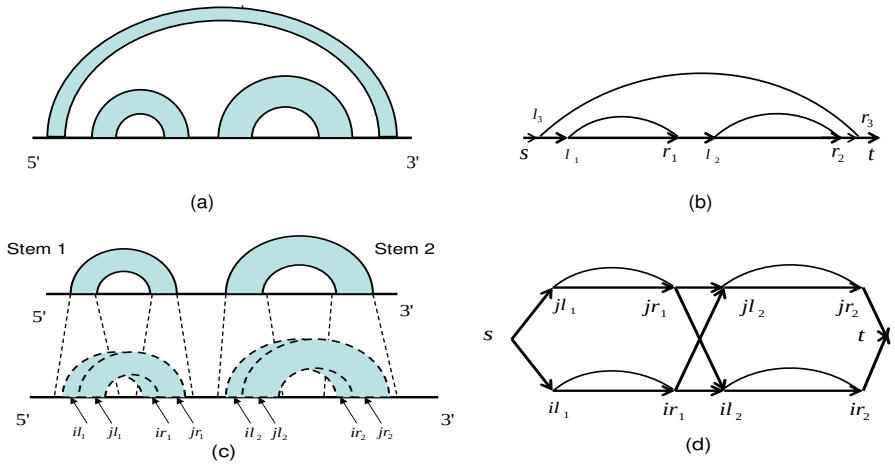

Figure 1. (a) An RNA structure that contains both nested and parallel stems. (b) The corresponding conformational graph. (c) A secondary structure (top), and the mapped regions and images for its stems on the target sequence (bottom). The dashed lines specify the possible mappings between stems and their images. (d) The image graph formed by the images of its stems on a target sequence. (il_1, ir_1) and (jl_1, jr_1) for stem 1, and (il_2, ir_2) and (jl_2, jr_2) for stem 2.

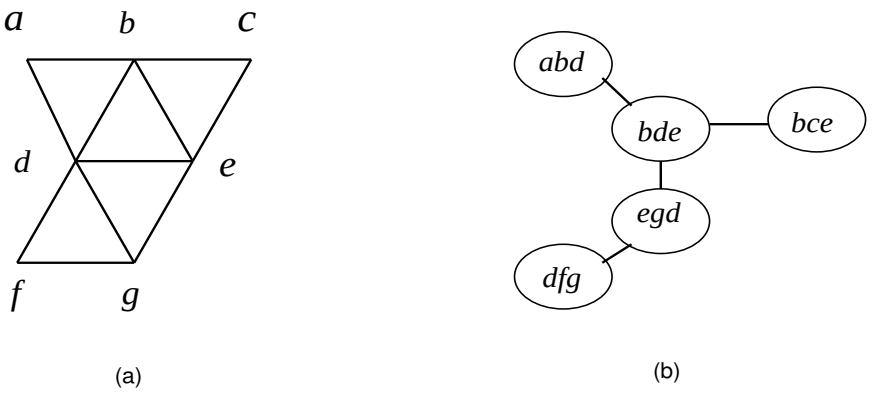

Figure 2. (a) An example of a graph. (b) The optimal tree decomposition for the graph in (a).

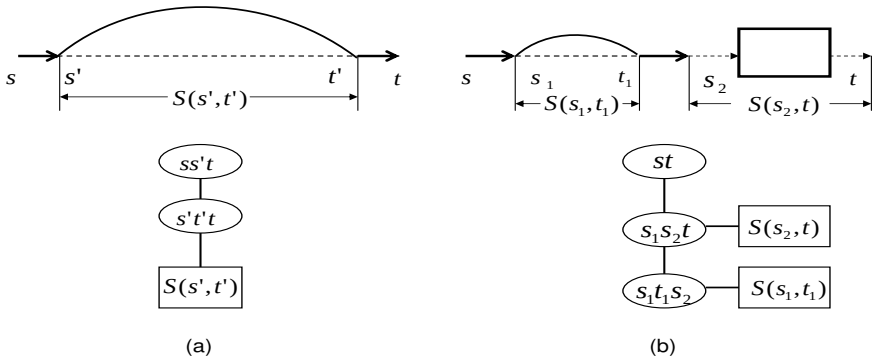

Figure 3. (a) The tree decomposition for secondary structure that contains an outer stem formed by s' and t'. (b) The tree decomposition for secondary structure that contains a leading structure unit with an outer stem (s_1, t_1).

$O(2^t n)$ for graphs of size n [1]. For the subgraph isomorphism problem, unfortunately, such efficient algorithms only exist for very small fixed guest graph H and host graph G with a small tree width t or being planar [21, 8], thus are not applicable to the RNA structure-sequence alignment investigated in this paper.

3.1 Tree Decomposition of Conformational Graphs

Although finding the optimal tree width and tree decomposition for a general graph is NP-hard [3], the conformational graph of a pseudoknot-free structure is simply an outer-planar graph which has tree width 2 [3]. We describe briefly in the following a linear time recursive algorithm to find an optimal tree decomposition for such conformational graphs.

A pseudoknot-free structure can only be either of the following two cases: a single outermost stem "containing" all other stems within, and parallel stems. The tree decomposition algorithm just needs to deal with these two situations recursively.

(a) If the graph has a single outmost stem (s', t'), the algorithm generates two connected tree nodes $\{s, s', t\}$ and $\{s', t', t\}$. Then it recursively produces a subtree (decomposition) for the part in between s' and t', and connects the root of the subtree to node $\{s', t', t\}$, as shown in Figure 3(a). It returns the node $\{s, s't\}$ as the root of the tree decomposition.

(b) If the graph consists of parallel stems, as shown in Figure 3(b), the algorithm generates a tree node $\{s_1, t_1, s_2\}$ for the first stem (s_1, t_1) and connect it to another tree node $\{s_1, s_2, t\}$. Recursively, it produces a subtree for the part in between s_1 and t_1 and connects the root of the subtree to node $\{s_1, t_1, s_2\}$. Similarly, it creates a subtree for the part in between s_2 and t and connects the root of the subtree to node $\{s_1, s_2, t\}$. It further connects the $\{s_1, s_2, t\}$ to the third node $\{s, t\}$ which is returned as the root of the tree decomposition.

Now we consider *pseudoknot structures*, which are secondary structures with as least two stems that structurally cross. Tree decomposition for the conformational graph of a pseudoknot structure can be obtained by extending a tree decomposition for the conformational graph of a pseudoknot-free structure, since a pseudoknot structure can be viewed as the combination of a maximal pseudoknot-free structure with some additional *crossing* stems. Adding an edge (u, v) that represents one of the crossing stems (Figure 4(a)) to the conformational graph of the pseudoknot-free structure, called the *primary conformational graph*, may only increase the tree width by 1. This can be achieved by first including the two disconnected vertices u and v into the conformational graph and finding a tree decomposition using the algorithm specified earlier in this section (Figure 4(b)). The tree decomposition is then extended by including v in every tree node on the path of the tree from the node containing u to the node containing v, thus accommodating the additional edge (u, v). (Figure 4(c)). Theoretically, if there are c crossing stems in a pseudoknot structure, the tree width of the corresponding conformational graph has tree width at most $2 + c$. Real RNA pseudoknots have a much smaller tree width. For example, Figure 6 shows the structure of tmRNA that contains 4 pseudoknots. It is a pseudoknot-free structure combined with crossing stems $(H, h), (N, n)$, (S, s), $(W, w), (X, x)$, (Z, z), and $(\#, 3)$. The tree width of the corresponding conformational graph is at most 4 since only crossing stems (W, w) and (X, x) are not independent of each other, increasing the tree width

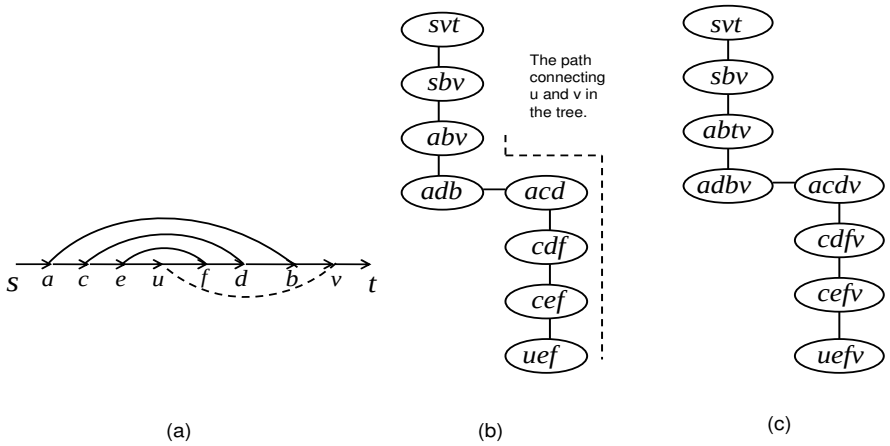

Figure 4. (a) The conformational graph of a pseudoknot structure. The dashed edge (u, v) represents one of the crossing stems. (b) A tree decomposition of the graph with vertices u and v disconnected. (c) A tree decomposition of the graph by adding v to tree nodes on the path connecting u and v to cover the additional edge (u, v).

by 2.

The above technique can be improved by considering adding one "stack" of nested crossing stems, instead of one crossing stem, to the primary conformational graph at a time. Theoretically, we can prove that the tree width can only increase at most by 4 for the addition of a "stack" of crossing stems independent of the number of stems in this "stack". Assume that there are d such "stacks", each with l_i nested stems such that $\sum_{i=1}^{d} l_i = c$, the total number of crossing stems. Then the tree width of the conformational graph of the pseudoknot is bounded by $2 + d \times \min_{1 \le i \le d}\{l_i, 4\}$. The technical details of the proof are omitted.

3.2 Tree Decomposition Based Optimal Alignment Algorithm

An alignment between a structure profile and a target sequence is essentially an isomorphism between the conformational graph H for the structure profile and some subgraph of the image graph G for the target sequence. To find such an isomorphism, we adopt the general dynamic programming technique [1] over the tree decomposition of H. However, because the general technique can only directly be applied to the subgraph isomorphism on small fixed graph H and graph G of a small tree width [21], we introduce some additional techniques to solve the problem in our setting. We present a summary and some details of the new optimal alignment algorithm in the following.

The dynamic programming over the tree decomposition to find an optimal alignment is based on the maintenance of a dynamic programming table for each node in the tree. An entry in a table includes a possible combination of images of vertices in the corresponding tree node and the validity and partial optimal alignment score associated with the combination. The table thus contains a column allocated for each vertex in the node and two additional columns V and S to maintain validities and partial optimal alignment scores respectively.

In a bottom up fashion, the algorithm first fills the entries in the tables for all leaf nodes. Specifically, for vertices in a leaf node, a combination of their images is valid if the corresponding mapping satisfies the first two conditions for isomorphism (see section 2) and the partial optimal alignment score for a valid combination is the sum of the alignment scores of loops and stems induced by images of vertices that are only contained in the node. For an internal node X_i in the tree, without loss of generality, we assume X_j and X_k are its children nodes. For a given combination e_i of images of vertices in X_i, the algorithm checks the first two conditions for isomorphism (see section 2) and sets e_i to be invalid if one of them is not satisfied. Otherwise, the algorithm queries the tables for X_j and X_k. e_i is set to be valid if and only if there exist valid entries e_j and e_k from the tables of X_j and X_k such that e_j and e_k have the same assignment of images as that of e_i for vertices in $X_i \cap X_j$ and $X_i \cap X_k$ respectively. The partial optimal alignment score for a valid entry e_i includes the alignment scores of stems and loops induced by images of vertices only in X_i and the maximum partial alignment scores over all valid entries e_j's and e_k's with the same assignment of images for vertices in $X_i \cap X_j$ and $X_i \cap X_k$ as that of e_i in tables for

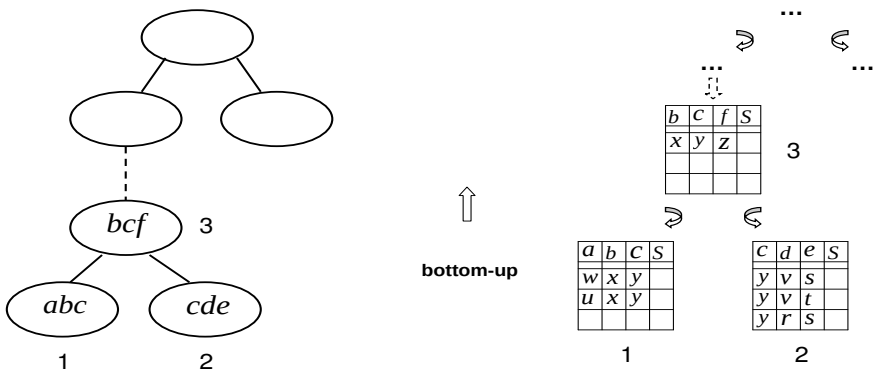

Figure 5. A sketch of the dynamic programming approach for optimal alignments. The algorithm maintains a dynamic programming table in each tree node. Starting with leaves of the tree, the algorithm follows a bottom-up fashion. In computing the table for a parent node, only combinations of the images of the vertices in the node are considered. In every such combination, only one locally best combination (computed in the children tables) is used for vertices that occur in the children nodes but not in the parent node.

X_j and X_k respectively. Figure 5 provides an example for the overall algorithm. In particular, nodes 1 and 2 are leaf nodes and their dynamic programming tables are computed by enumerating all possible combinations of images of vertices in them. For internal node 3, to determine the validity and partial optimal alignment scores of entry x, y, z, the algorithm needs to query the table in node 1 for all entries that assign image x to vertex b and y to c since $X_3 \cap X_1 = \{b, c\}$, and the table in node 2 for all entries that assign y to vertex c. We omit the column for validities in Figure 5 for simplicity, since we can mark a combination to be invalid by setting its partial optimal alignment score to be $-\infty$. The optimal overall alignment score can thus be obtained from the table in the root of the tree by selecting the entry with the maximum partial alignment score. A recursive process starting with this entry can be used to trace back the optimal alignment.

Some steps in the algorithm need to be elaborated. First, let a tree node contain t vertices $\{v_1, v_2, \ldots, v_t\}$ from conformational graph H. If the number of images of v_i in the image graph G is at most k, a dynamic programming table of the size $O(k^t)$ is sufficient to accommodate all possible combinations of the images since exactly one image is chosen for every vertex v_i for the isomorphism. Second, when computing the table, the algorithm uses information from the tables of its children nodes but only partial alignment scores associated with vertices $\{v_1, v_2, \ldots, v_t\}$ can be directly used. This way, the size of a dynamic programming table for each node is always bounded by $O(k^t)$. Third, it is easy to see that the algorithm satisfies the isomorphism condition (1) (given in section 2) for alignment. The satisfaction of the condition (2) by the algorithm can be proved inductively based on the transitivity of the partial order defined by the directed edges. We omit the proof from this paper.

The alignment score is the sum of the scores for aligning individual stems and loops in the structure profile. The alignment score for a stem is calculated between the stem profile and a chosen image in the target of the stem. Since any loop in the structure is between some two stems, the alignment score for a loop is calculated between its profile and the sequence segment in the target within the two chosen images for the two stems.

The running time of the dynamic programming process is $O(k^t n)$ over a tree decomposition of tree width t and tree size n without including the time for alignment. Let N be the size of the overall structure profile containing m stems of lengths $s_1, \ldots, s_m$ and r loops of lengths $l_1, \ldots, l_r$. Then $N = \sum_{i=1}^{m} 2s_i + \sum_{j=1}^{r} l_j$ and $n = O(m)$. Since for each stem and loop profile, its optimal alignment to a counterpart in the target sequence takes a quadratic time, the total time for the optimal alignment algorithm is $O(k^t N^2)$.

4 Tests and Evaluation Results

We performed experiments to test the accuracy and efficiency of the algorithm and compared the performance of the algorithm with that of the CM-based searching. The training data was obtained from Rfam database [13], for each family, we choose up to 60 sequences with their pairwise identities lower than 80% from the structural align-

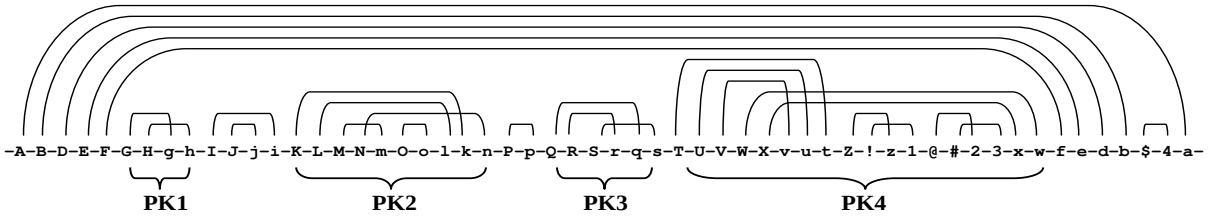

Figure 6. Diagram of stems in the secondary structure of a tmRNA. Upper case letters indicate base regions that pair with the corresponding lower case letters. The four pseudoknots constitute the central part of the tmRNA gene and are labeled as Pk1, Pk2, Pk3, Pk4 respectively.

ment of seed sequences. In practice, to obtain a reasonably small value for the parameter k, the upper bound on the number of images that a stem can map to, we constrain the images of a stem within certain region, called the *constrained image region* of the stem, in the target sequence. For this, we assume that, for homologous sequences, the distances from the pairing region of a given stem to the 3' end follow a Gaussian distribution. We compute the mean and standard deviation of distances from its two pairing regions to the 3' end of the sequence respectively, evaluated over all training sequences. For training data representing distant homologs of an RNA family, we can effectively divide data into groups so that a different but related profile can be built for each group and used for search. This ensures a small value for the parameter k in the models.

As a first profiling and searching experiment, we inserted several RNA sequences from the same family into a random background generated with the same base composition as the sequences in the family. We then used both our algorithm and a CM-based searching algorithm we previously developed [19] to search for the inserted sequences. We compared the sensitivity and specificity of both searching algorithms on several different RNA families. To test the performance of the algorithm on real genomes, we used the algorithm to search for non-coding RNA genes in real biological genomes.

4.1 Searching for Pseudoknot Free Sequences

We used both the tree-decomposition based and the CM based algorithm to search for about 30 pseudoknot free RNA structures inserted in a random background of 10^5 nucleotides generated with the same base composition. We determined the statistical distribution for the alignment scores with a random sequence of 3000 nucleotides, which is generated with the same base composition as that of the sequence to be searched, with a method similar to that used by RSEARCH [14]. An alignment score with a Z-score greater than 5.0 is reported as a hit in both searching programs. In our experiments, for each stem, the algorithm selects k images with the maximum alignment scores within the constrained image region of the stem. In order to evaluate the impact of the parameter k on the accuracy of the algorithm, we carried out the same searching experiments for each given k.

Table 1 shows that, on tested RNA families, the tree decomposition based algorithm achieves the same searching accuracy as that of the CM based algorithm when the parameter k is equal to or larger than 6. From Table 2, compared to the CM based searching, the tree decomposition based algorithm requires a significantly reduced amount of computation time when the parameter k is 6. On most of the tested families, the tree decomposition based searching is more than 20 times faster than the CM based searching.

4.2 Searching for Sequences with Pseudoknots

We also performed searching experiments on several RNA families that contain pseudoknot structures. For each family, we inserted about 30 structures that contain pseudoknot structures into a background randomly generated with the same base composition as that of the inserted sequences. The training data was also obtained from the Rfam database [13] where we selected up to 40 sequences with pair wise identity lower than 80% from the seed alignment for each family. We used both the tree decomposition based algorithm and the CM based algorithm to identify the inserted sequences. For both algorithms, the threshold of alignment scores for reporting a hit is determined by a Z-score value 5.0.

Tables 3 and 4 show the comparisons of both searching accuracy and computation time for both algorithms. It is evident that, on families with pseudoknots, the tree decomposition based algorithm achieves the same accuracy as that of the CM based algorithm when the parameter k reaches a value of 7. In particular, the computation time needed by the algorithm is about 66 and 38 times less than that of the CM based algorithm on Alpha_RBS and Tombus_3_IV, the

RNA	LE	CM based		Tree decomposition based							
				$k=5$		$k=6$		$k=7$		$k=8$	
		SE	SP	SE	SP	SE	SP	SE	SP	SE	SP
Entero_CRE	61	80.65	100	74.19	100	80.65	100	80.65	100	80.65	100
Entero_OriR	73	100	100	100	100	100	100	100	100	100	100
Let_7	84	100	100	95.8	100	95.8	100	100	100	100	100
Lin_4	72	100	100	100	88.9	100	94.11	100	94.11	100	94.11
Purine	103	93.10	100	93.10	96.43	93.10	96.43	93.10	96.43	93.10	96.43
SECIS	68	100	97.30	100	97.30	100	97.30	100	97.30	100	97.30
S_box	112	100	100	100	92.86	100	92.86	100	96.30	100	96.30
Tymo_tRNA-like	86	100	96.67	100	96.67	100	96.67	100	96.67	100	96.67

Table 1. A comparison of the searching accuracy of the tree decomposition based and CM based algorithms in terms of sensitivity and specificity. LE is the average length of sequences in the family, SE and SP are sensitivity and specificity in percentage respectively.

RNA	CM based	Tree decomposition based							
		$k=5$		$k=6$		$k=7$		$k=8$	
	RT	RT	SU	RT	SU	RT	SU	RT	SU
Entero_CRE	57.96	2.60	22.2×	2.85	20.3×	3.21	18.1×	3.38	17.2×
Entero_OriR	103.08	4.77	21.6×	4.91	21.0×	5.26	19.6×	5.42	19.0×
Let_7	157.11	13.94	11.3×	14.97	10.5×	16.38	9.6×	16.92	9.3×
Lin_4	132.51	2.45	54.1×	3.22	41.2×	4.25	31.2×	5.10	26.0×
Purine	179.29	6.61	27.1×	7.09	25.3×	8.49	21.1×	9.61	18.7×
SECIS	185.21	8.48	21.8×	9.14	20.3×	10.23	18.1×	10.89	17.0×
S_box	756.27	26.10	29.0×	29.76	25.4×	34.76	21.8×	41.01	18.4×
Tymo_tRNA-like	185.05	4.34	42.6×	5.01	37.0×	6.10	30.3×	7.07	26.2×

Table 2. The computation time for both searching algorithms on all pseudoknot free RNA families. RT is the computation time in minutes, SU is the amount of speed up compared to the CM based searching algorithm.

RNA	LE	CM based		Tree decompostion based							
				$k=5$		$k=6$		$k=7$		$k=8$	
		SE	SP	SE	SP	SE	SP	SE	SP	SE	SP
Alpha_RBS	110	100	96.00	91.67	88.00	95.80	92.00	100	96.00	100	96.00
Antizyme_FSE	55	100	100	92.86	100	96.43	100	100	100	100	100
HDV_ribozyme	95	100	100	100	97.37	100	97.37	100	97.37	100	97.37
IFN_gamma	170	100	100	100	100	100	100	100	100	100	100
Tombus_3_IV	95	100	100	92.31	100	100	100	100	100	100	100
corona_pk3	65	100	94.80	100	97.37	100	97.37	100	97.37	100	97.37

Table 3. The searching accuracy for both tree decomposion based and CM based algorithms on RNA sequences containing pseudoknots.

two families that contain more than 100 nucleotides. This demonstrates the promising advantage of the tree decomposition based algorithm over the CM based searching method in computation time when the structural pattern for which

RNA	CM based	Tree decomposition based							
		$k = 5$		$k = 6$		$k = 7$		$k = 8$	
	RT	RT	SU	RT	SU	RT	SU	RT	SU
Alpha_RBS	27.85	0.24	116.0×	0.31	90.1×	0.42	66.3×	0.55	50.6×
Antizyme_FSE	0.94	0.10	9.4×	0.13	7.2×	0.18	5.2×	0.23	4.1×
HDV_ribozyme	6.54	0.22	29.7×	0.34	19.2×	0.52	12.6×	0.79	8.3×
IFN_gamma	31.24	0.47	66.5×	0.72	43.4×	1.07	29.2×	1.52	20.6×
Tombus_3_IV	15.45	0.17	90.9×	0.27	57.2×	0.40	38.6×	0.57	27.1×
corona_pk3	2.89	0.12	24.1×	0.15	19.3×	0.20	14.5×	0.26	11.1×

Table 4. The computation time for both searching algorithms on all RNA families that contain pseudoknots. The amount of RT is in hours.

one is searching contains more than 100 nucleotides.

4.3 Search on Biological Genomes

To test the peformance of the algorithm on real genomes, we used the algorithm to search biological genomes for structural patterns that contain pseudoknots. For example, the secondary structure formed by nucleotides in the 3' untranslated region in the genomes of the corona virus family contains a pseudoknot structure. This pseudoknot was recently shown to play important roles in the replication of the viruses in the family [5]. We selected four genomes from the corona virus family and used the algorithm to search for this pseudoknot. For bacteria, the tmRNA is essential for the trans-translation process and is responsible for adding a new C-terminal peptide tag to the incomplete protein product of a broken mRNA [11]. The secondary structure of tmRNA contains four pseudoknots and Figure 6 provides a sketch of the stems that constitute the secondary structure of a tmRNA. The tree decomposition based algorithm was also used to search for tmRNA genes on the genomes of two bacteria organisms, *Haemophilus influenzae* and *Neisseria meningitidis*. Both of the genomes contain more than 10^6 nucleotides. Among the bacteria containing tmRNAs, these two are relatively distant from each other evolutionarily. To test the accuracy and efficiency of the algorithm on genomes with a significantly larger size, we used the algorithm to search for the telomerase RNA gene in the genomes of two yeast organisms, *Saccharomyces cerevisiae* and *Saccharomyces bayanus*, both of which contain more than 10^7 nucleotides. Telomerase RNA is responsible for the addition of some specific simple sequences onto the chromosome ends [7].

The parameter k used in the tree decomposition based algorithm for searching all genomes is 7. Table 5 provides the real locations of the searched patterns, the locations annotated by the tree decompostion based and CM based algorithms respectively. The table clearly shows that, compared with CM based searching, the tree decomposition based model and searching algorithm are able to achieve the same accuracy with a significantly reduced amount of computation time. Both our new program and the CM base program have 100% sensitivity and specificity for searches in genomes. Searching a genome of moderate size for a structural pattern as complex as tmRNA gene only needs days of computation time, instead of months.

5 Conclusions

In this paper, we introduce a novel graph theoretical model for profiling RNA structures including pseudoknots. This approach profiles the fundamental structural units that form the secondary structure of an RNA family separately, and the structural relations among the structural units are described with a conformational graph. Based on this generic framework, an image graph can be constructed by determining the possible locations of each stem on a target sequence. The target sequence can be efficiently aligned to the profiling model by computing the maximum valued subgraph isomorphic to the conformational graph in the image graph. Our experiments demonstrated that this approach is able to achieve the same searching accuracy as CM based methods while requiring only a small fraction of the computation time needed by them. Based on this profiling model and the optimal alignment algorithm, we are able to accurately determine the locations of ncRNAs with complex structural patterns in genomes of a moderate size in days.

The time complexity of the alignment is $O(k^t N^2)$, for an RNA family that contains N nucleotides and has a conformational graph with tree width t. Parameter k is an upper bound of the number of images of each stem on a target sequence and can be effectively determined with a statistical cut-off value based on the constrained mapped regions of a stem. Our experiments also showed that, on most tested RNA families, a value of 7 is sufficient for achieving the same accuracy as that of the CM based searching methods.

OR	ncRNA	Tree decomposition based			CM based			Real location		GL
		Left	Right	RT	Left	Right	RT	Left	Right	
BCV	3'PK	30798	30859	0.053	30798	30859	1.24	30798	30859	0.31
MHV	3'PK	31092	31153	0.053	31092	31153	1.27	31092	31153	0.31
PDV	3'PK	27802	27882	0.048	27802	27882	1.17	27802	27882	0.28
HCV	3'PK	27063	27125	0.047	27063	27125	1.12	27063	27125	0.27
HI	tmRNA	472209	472574	44.0	472210	472575	1700	472210	472575	18.3
NM	tmRNA	1241197	1241559	52.9	1241197	1241559	2044	1241197	1241559	22.0
SC	TLRNA	307688	308429	492.3	–	–	–	307691	308430	103.3
SB	TLRNA	7121529	7122284	550.2	–	–	–	7121532	7122282	114.8

Table 5. A comparison of the accuracy and efficiency for both algorithms on searching biological genomes. OR is the name of the organism; GL is the length of the genome in multiples of 10^5 nucleotides. BCV is Bovine corona virus; MHV is Murine hepatitus virus; PDV is Porcine diarrhea virus; HCV is Human corona virus; HI and NM represent *Haemophilus influenzae* and *Neisseria meningitidis* respectively. SC and SB represent *Saccharomyces cerevisiae* and *Saccharomyces bayanus* respectively. RT is the single CPU time needed to identify the ncRNA in hours. For tmRNA and telomerase RNA searches, RT is estimated from the time needed by a parallel search with 16 processors.

Acknowledgments

We would like to thank the anonymous reviewers for their constructive remarks on an earlier version of this paper.

References

[1] S. Arnborg and A. Proskurowski. Linear time algorithms for np-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23:11–24, 1989.

[2] V. Bafna and S. Zhang. Fastr: Fast database search tool for non-coding rna. *Proceedings of the 3rd IEEE Computational Systems Bioinformatics Conference*, pages 52–61, 2004.

[3] H. L. Bodlaender. Classes of graphs with bounded tree-width. *Tech. Rep. RUU-CS-86-22*, University of Utrecht, 1986.

[4] M. Brown and C. Wilson. Rna pseudoknot modeling using intersections of stochastic context free grammars with applications to database search. *Proceedings of Pacific Symposium on Biocomputing*, pages 109–125, 1995.

[5] S. J. G. B. H. T. F. Dombrowski and P. S. Masters. Characterization of the rna components of a putative molecular switch in the 3' untranslated region of the murine coronavirus genome. *Journal of Virology*, 78:669–682, 2004.

[6] S. Eddy and R. Durbin. Rna sequence analysis using covariance models. *Nucleic Acids Research*, 22:2079–2088, 1994.

[7] A. T. D. N. L. S. L. J. L. S. A. Elela and R. J. Wellinger. A phylogenetically based secondary structure for the yeast telomerase rna. *Current Biology*, 14:1148–1158, 2004.

[8] D. Eppstein. Subgraph isomorphism in planar graphs and related problems. *Journal of Graph Algorithms and Applications*, 3.3:1–27, 1999.

[9] D. N. Frank and N. R. Pace. Ribonuclease p: unity and diversity in a trna processing ribozyme. *Annu Rev Biochem*, 67:153–180, 1998.

[10] D. Gautheret and A. Lambert. Direct rna motif definition and identification from multiple sequence alignments using secondary structure profiles. *Journal of Molecular Biology*, 313:1003–1011, 2001.

[11] N. N. B. F. J. F. A. R. F. G. H. Himeno and A. Muto. Functional and structural analysis of a pseudoknot upstream of the tag-encoded sequence in e. coli tmrna. *Journal of Molecular Biology*, 286(3):733–744, 1999.

[12] E. R. R. J. K. T. A. Jones and S. R. Eddy. Computational identification of noncoding rnas in e. coli by comparative genomics. *Current Biology*, 11:1369–1373, 2001.

[13] S. G.-J. A. B. M. M. A. Khanna and S. R. Eddy. Rfam: an rna family database. *Nucleic Acids Research*, 31:439–441, 2003.

[14] R. J. Klein and S. R. Eddy. Rsearch: Finding homologs of single structured rna sequences. *BMC Bioinformatics*, 4:44, 2003.

[15] Y. U. A. H. Y. Kobayashi and T. Yokomori. Tree adjoining grammars for rna structure prediction. *Theoretical Computer Science*, 210:277–303, 1999.

[16] R. D. S. E. A. Krogh and G. Mitchison. *Biological sequence analysis: Probabilistic models of proteins and nucleic acids*. Cambridge Univ. Press, The Edinburgh Building, Cambridge CB2 2RU, UK, 1998.

[17] T. M. Lowe and S. R. Eddy. trnascan-se: A program for improved detection of transfer rna genes in genomic sequence. *Nucleic Acids Research*, 25:955–964, 1997.

[18] Z. Y. Q. Z. K. Luo and Q. Zhou. The 7sk small nuclear rna inhibits the cdk9/cyclin t1 kinase to control transcription. *Nature*, 414:317–322, 2001.

[19] C. L. Y. S. R. Malmberg and L. Cai. Profiling and searching for rna pseudoknot structures in genomes. *Proceedings*

of 2005 International Workshop in Bioinformatics Research and Applications, to appear, 2005.

[20] L. C. R. Malmberg and Y. Wu. Stochastic modeling of pseudoknot structures: A grammatical approach. *Bioinformatics*, 19:i66–i73, 2003.

[21] J. Matousek and R. Thomas. On the complexity of finding iso- and other morphisms for partial k-trees. *Discrete Mathematics*, 108:343–364, 1992.

[22] V. T. N. T. K. A. A. Michels and O. Bensaude. 7sk small nuclear rna binds to and inhibits the activity of cdk9/cyclin t complexes. *Nature*, 414:322–325, 2001.

[23] E. Rivas and S. Eddy. The language of rna: a formal grammar that includes pseudoknots. *Bioinformatics*, 16:334–340, 2000.

[24] E. Rivas and S. R. Eddy. A dynamic programming algorithm for rna structure prediction including pseudoknots. *Journal of Molecular Biology*, 285:2053–2068, 1999.

[25] E. Rivas and S. R. Eddy. Noncoding rna gene detection using comparative sequence analysis. *BMC Bioinformatics*, 2:8, 2001.

[26] N. Robertson and P. D. Seymour. Graph minors ii. algorithmic aspects of tree-width. *Journal of Algorithms*, 7:309–322, 1986.

[27] Z. Weinberg and W. L. Ruzzo. Faster genome annotation of non-coding rna families without loss of accuracy. *Proceedings of the Eighth Annual International Conference on Computational Molecular Biology*, pages 243–251, 2004.