Efficient Parameterized Algorithm for Biopolymer Structure-Sequence Alignment

Yinglei Song[1], Chunmei Liu[1], Xiuzhen Huang[2], Russell L. Malmberg[3], Ying Xu[4], and Liming Cai[1,⋆]

[1] Dept. of Computer Science, Univ. of Georgia, Athens GA 30602, USA
[2] Dept. of Computer Science, Arkansas State Univ., State University, AR 72467, USA
[3] Dept. of Plant Biology, Univ. of Georgia, Athens GA 30602, USA
[4] Dept. of Biochemistry and Molecular Biology, Univ. of Georgia, Athens, GA 30602, USA

Abstract. Computational alignment of a biopolymer sequence (e.g., an RNA or a protein) to a structure is an effective approach to predict and search for the structure of new sequences. To identify the structure of remote homologs, the structure-sequence alignment has to consider not only sequence similarity but also spatially conserved conformations caused by residue interactions, and consequently is computationally intractable. It is difficult to cope with the inefficiency without compromising alignment accuracy, especially for structure search in genomes or large databases.

This paper introduces a novel method and a parameterized algorithm for structure-sequence alignment. Both the structure and the sequence are represented as graphs, where in general the graph for a biopolymer structure has a naturally small tree width. The algorithm constructs an optimal alignment by finding in the sequence graph the maximum valued subgraph isomorphic to the structure graph. It has the computational time complexity $O(k^t N^2)$ for the structure of N residues and its tree decomposition of width t. The parameter k, small in nature, is determined by a statistical cutoff for the correspondence between the structure and the sequence. The paper demonstrates a successful application of the algorithm to developing a fast program for RNA structural homology search.

1 Introduction

Structure-sequence alignment plays the central role in a number of important computational biology methods. For instance, protein threading, an effective method to predict protein tertiary structure, is based on the alignment between the target sequence and structure templates in a template database [3, 5, 37, 19, 36]. Structure-sequence alignment is also essential to RNA structural homology search, a viable approach to annotating (and identifying new) non-coding RNAs [10, 12, 29, 22]. Structure-sequence alignment also finds applications in other bioinformatics tasks where structure plays an instrumental role, such as in the identification of the structure of intermolecular interactions [25, 27], and in the discovery of the structure of biological pathways through comparative genomics [8].

⋆ Corresponding author: cai@cs.uga.edu

The structure-sequence alignment is to find an optimal way to "fit" the residues of a target sequence in the spatial positions of a structure template. To be able to identify the structure of remote homologs, the alignment has to consider not only sequence similarity but also spatially conserved conformations caused by sophisticated interactions between residues, and consequently is computationally intractable. For example, it is both NP-hard for protein threading with amino acid interactions [18] and for thermodynamic determination of RNA secondary structure including pseudoknots [23].

The alignment problem has often been formulated as integer programming that characterizes residue spatial interactions with (a large number of) linear inequality constraints [36, 20]. Commercial software packages for linear programming are usually used to approximate the integer programming and to reduce the computation time. More sophisticated techniques, such as branch-and-cut, can be used to dynamically include only needed linear constraints [20, 28]. Moreover, a divide-and-conquer method based on the notion of "open-links" has also been devised to address the residue-residue interaction issue [37]. For RNA structure-sequence alignment, dynamic programming has been extended to include crossing patterns of RNA nucleotide interactions [32, 7]. The above algorithmic techniques cope with the alignment intractability, however, most of them still require computation time polynomial of a high-degree.

In this paper, we introduce an efficient structure-sequence alignment algorithm. Both structure and sequence are represented as mixed graphs (with directed and undirected edges); the optimal alignment corresponds to finding in the sequence graph the maximum valued subgraph isomorphic to the structure graph. In addition, we introduce an integer parameter k to constrain the correspondence between the graphs. A dynamic programming algorithm is developed over a tree decomposition of the structure graph. For each value of k, the optimal alignment can be found in time $O(k^t N^2)$ for each structure template containing N residues given a tree decomposition of tree width t.

Our algorithm is a parameterized algorithm [11], in which the naturally small parameter k determined by a statistical cutoff reflects the accuracy of the alignment. The new algorithm with the time complexity $O(k^t N^2)$ is more efficient than previous algorithms, for example, of the time complexity $O(N^k)$ [37]. This is also because the tree width t of the graph for a biopolymer structure is small in nature. For example, the tree width is 2 for the graph of any pseudoknot-free RNA and the width can only increase slightly for all known pseudoknot structures (see Figure 5). Our experiments also show that among 3890 protein tertiary structure templates compiled using PISCES [33], only 0.8% of them have tree width $t > 10$ and 92% have $t < 6$, when using a 7.5 $\mathring{A}$ C_β-C_β distance cutoff for defining pair-wise interactions (Figure 2(a)).

The alignment algorithm has been applied to the development of a fast RNA structure homology search program [31]. With a significantly reduced amount of computation time, the new search method achieves the same accuracy as searches based on the widely used Covariance model (CM) [13]. The new algorithm yields about 24 to 50 times of speed up for the search of pseudoknot-free RNAs with 90 to 150 nucleotides; it gains even more significant advantage for larger RNAs or structures including pseudoknots. In addition, for all the conducted tests, including the searches of medium to large RNAs in bacteria and yeast genomes, parameter $k \leq 7$ has been sufficient for the accurate identification.

2 Problem formulation

We formulate structure-sequence alignment as a *generalized* subgraph isomorphism problem. Graphs used here are *mixed* graphs contaning both undirected and directed edges. Let $V(G)$, $E(G)$, and $A(G)$ denote the vertex set, the undirected edge set, and the directed edge (arc) set of graph G, respectively.

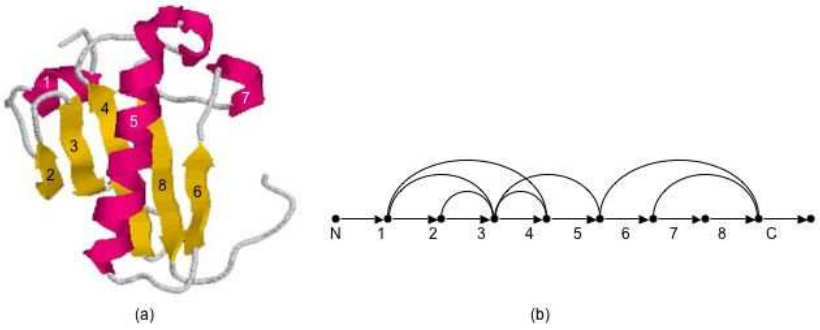

Fig. 1. (a) Folded ChainB of Protein Kinase C interacting protein with 8 cores (the PDB-file corresponding to PDB-ID 1AV5); (b) its corresponding structure graph.

Definition 1: A *structural unit* in a biopolymer sequence is a stretch of contiguous residues (nucleotides or amino acids). A non-structural stretch, between two consecutive structural units, is called a *loop*.

A structure of the sequence is characterized by interactions among structural units. For example, structural units in a tertiary protein are α helices and β strends, called *cores*. Figure 1(a) shows a protein structure with 8 structural units. In the RNA secondary structure, a structural unit is a stretch of nucleotides, one half of a stem formed by a stack of base pairings.

Given a biopolymer sequence, a *structure graph* H can be defined such that each vertex in $V(H)$ represents a structural unit, each edge in $E(H)$ represents the interaction between two structural units, and each arc in $A(H)$ represents the loop ended by two structural units. Figure 1(b) shows the structure graph for the folded protein in 1(a). Figure 5 shows the graph for bacterial tmRNAs.

The alignment between a structure template and a target sequence is to place residues of the sequence in the spatial positions of the template. Instead of placing individual residues to the spatial positions, the method we introduce in this paper allows us to put a stretch of residues as a whole in the position of some structural unit of the template. The sequence to be aligned to the structure is preprocessed so that all *candidates* in the sequence are identified for every structural unit in the template.

By representing each candidate as a vertex, the target sequence can also be represented as a mixed graph G, called a *sequence graph*. Each edge in $E(G)$ connects a pair of candidates that may possibly interact but do not overlap in sequence positions, and an arc in $A(G)$ connects two candidates that do not overlap.

Based on the graph representations, the structure-sequence alignment problem can be formulated as the problem of finding in the sequence graph G a subgraph isomorphic to the structure graph H such that the objective function based on the alignment score achieves the optimum. For this, we first introduce a mechanism to parameterize (and to scrutinize) the mapping between H and G.

Definition 2: A *map scheme* M between graphs H and G is a function: $V(H) \rightarrow 2^{V(G)}$ that maps every vertex in H to a subset of vertices in G. The maximum size of such a subset, $k = \max_{v \in V(H)}\{|M(v)|\}$, is called the *map width* of the map scheme.

A map scheme can be obtained in the preprocessing step that finds all candidates of every structural unit. The qualification of these candidates can usually be quantified by a statistical cutoff of the degree to which a candidate is aligned to a structural unit. One may simply choose the top k candidates for each structural unit. More sophisticated map schemes are possible (see section 4), in which ideally, the parameter k reflects the accuracy of alignment results. We define the following parameterized problem:

GENERALIZED SUBGRAPH ISOMORPHISM:
INPUT: mixed graphs H and G, and map scheme M of width k;
OUTPUT: a subgraph G' of G and an isomorphic mapping $f : V(H) \rightarrow V(G')$, constrained by $f(x) \in M(x)$ for any x, such that the objective function

$$\sum_{u \in V(H)} S_1(u, f(u)) + \sum_{(u,v) \in E(H)} S_2((u,v),(f(u),f(v))) + \sum_{\langle u,v\rangle \in A(H)} S_3(\langle u,v\rangle, \langle f(u), f(v)\rangle) \quad (1)$$

achieves the optimum (i.e., maximum or minimum).

Functions S_1, S_2, and S_3 are application dependent, scoring respectively three different alignments between the structure template and the target sequence: the alignment between a structural unit u and its candidate $f(u)$, the alignment between the interaction of two structural units (u, v) and the interaction of the corresponding candidates $(f(u), f(v))$, and the alignment between a loop (connecting two neighboring structural units u and v) and its correspondence loop in the sequence.

This problem generalizes the well-known NP-hard subgraph isomorphism decision problem. Efficient algorithms for subgraph isomorphism may be obtained on constrained instances. However, algorithms of this kind only exist for the cases where H is small, fixed, and G is planar or of a small tree width [1, 14, 24]. None of these conditions can be satisfied by the application in structure-sequence alignment, where the structure can be large and the sequence graph is often arbitrary.

We conclude this section by noting that the parameterization introduced on the map width does not trivialize the problem under investigation. In fact, one can transform NP-hard problem 3-SAT to (a decision version of) this problem when k is fixed to be 3, leading to the following theorem (the proof details are omitted).

Theorem 1: The problem GENERALIZED SUBGRAPH ISOMORPHISM remains NP-hard on map schemes of map width $k = 3$.

3 Parameterized alignment algorithm

Definition 3: [30] Pair (T, X) is a *tree decomposition* of a mixed graph H if

1. T is a tree,
2. $X = \{X_i | i \in V(T), X_i \subseteq V(H)\}$, and $\bigcup_{X_i \in X} X_i = V(H)$,
3. $\forall u, v, (u, v) \in E(H)$ or $\langle u, v \rangle \in A(H)$, $\exists i \in V(T)$ such that $u, v \in X_i$, and
4. $\forall i, j, k \in V(T)$, if k is on the path from i to j in tree T, then $X_i \cap X_j \subseteq X_k$.

The *tree width* of (T, X) is defined as $\max_{i \in V(T)}\{|X_i|\} - 1$. The *tree width of the graph* is the minimum tree width over all possible tree decompositions of the graph.

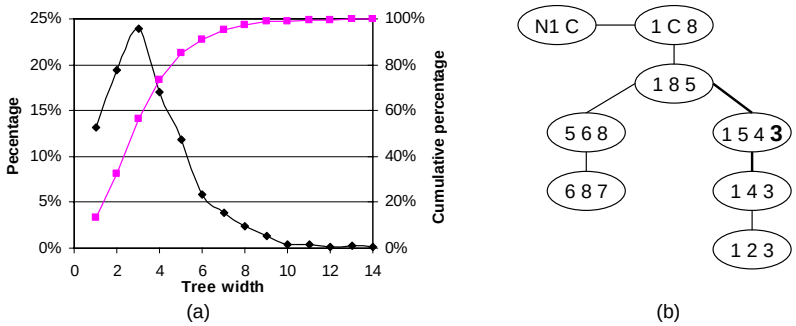

Fig. 2. (a) Tree width distribution of the graphs for 3,890 protein structure templates compiled using PISCES [33, 34]. (b) A tree decomposition for the structure graph in Figure 1(b).

Biopolymer structure graphs in general have small tree width. For instance, the tree width of the structure graphs for pseudoknot-free RNAs is 2, and it can only increase slightly for all known pseudoknots. Figure 2(a) gives a statistics on the tree width of about 3,890 protein structure templates compiled using PISCES [33, 34]. Figure 2(b) shows a tree decomposition for the protein structure graph in Figure 1(b).

3.1 Parameterized algorithm for subgraph isomorphism

We now describe a tree decomposition based parameterized algorithm for the problem GENERALIZED SUBGRAPH ISOMORPHISM formulated in section 2. Our algorithm assumes a given tree decomposition (T, X) of width t for structure graph H. Our algorithm follows the basic idea of the tree decomposition based techniques in [1, 2].

To simplify our discussion, we assume that T for the tree decomposition is a binary tree. The following notations will also be useful. Let $U \subseteq V(H)$ and $Y \subseteq V(G)$ such that $|U| = |Y|$. Then a mapping $f : U \rightarrow Y$ is a *valid mapping for* U if f is a subgraph isomorphism between the graph induced by U and the graph induced by Y. If $W \subseteq U$, then $f|_W$ is f *projected onto* W, therefore a valid mapping for W. A *partial isomorphism for* H *with respect to* X_i is a valid mapping f for $U = X_i \cup \bigcup_{k \in D(i)} X_k$, where $D(i)$ is the set of i's descendent nodes in the tree.

In a bottom up fashion, the algorithm establishes one table for each tree node. Let $X_i = \{u_0, u_1, \ldots, u_t\}$. Table m_i for tree node i consists of $|X_i| + 3$ columns, one for every vertex in X_i. Rows are all possible mappings for X_i restricted by the map scheme M; each row is of the form $\langle x_0, x_1, \ldots, x_t \rangle$ representing the mapping f, $f(u_l) = x_l$, $l = 0, 1, \ldots, t$. There are three additional columns in the table: V, S, Opt (see Figure 3). $V(f) =$ '$\surd$' if and only if mapping f is valid for X_i. $S(f)$ is the optimal score over all the partial isomorphism e for H with respect to X_i such that $f = e|_{X_i}$. $Opt(f)$ indicates whether $S(f)$ is the optimal over all valid mapping f' for X_i, where $f'|_{X_i \cap X_p} = f|_{X_i \cap X_p}$ for p, the parent node of i.

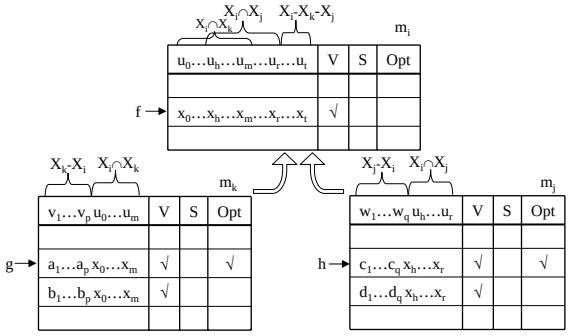

Fig. 3. Computing dynamic programming tables over a tree decomposition in which tree node i has two children k and j.

If i is a leaf node, the score $S(f)$ is simply the value computed based on formula (1) (given in section 2) for vertices in X_i only. If i is an internal node with children nodes k and j, $S(f)$ is the sum of the following three value :

1. The value computed for f with formula (1) for vertices in X_i only,
2. The maximum S value over all valid mappings g in table m_k such that $g|_{X_i \cap X_k} = f|_{X_i \cap X_k}$, and
3. The maximum S value over all valid mappings h in table m_j such that $h|_{X_i \cap X_j} = f|_{X_i \cap X_j}$.

Figure 3 illustrates the computation for row f in table m_i of the internal node i that has two children nodes k and j. The formal algorithm, GENSUBGISOMO, is outlined as a recursive process in Figure 4. The optimal score computed in the table for the root of the tree T is the best isomorphism score. A recursive routine can be used to trace back the corresponding optimal isomorphism. Details are omitted here.

We need to prove that the (bottom up) dynamic programming always produces correct partial isomorphisms. Since the algorithm automatically validates the isomorphism for locally involved vertices, it suffices to ensure that for every $u \in X_i$, the mapping from u to x for some $x \in M(u)$ does not conflict with an earlier mapping from v to x, for some vertex $v \in X_k$, where k is a descendent of i. Interestingly enough, for

ALGORITHMGENSUBGISOMO (T, X_i, M, i, m_i)
If i has left child k, GENSUBGISOMO(T, X_k, M, k, m_k);
If i has right child j, GENSUBGISOMO(T, X_j, M, j, m_j);
For every every mapping f for X_i, constrained by M
  If i has left child k in T
    Find in m_k a valid mapping g, such that $g|_{X_i \cap X_k} = f|_{X_i \cap X_k}$ of $Opt(g)$ being '$\surd$';
  If i has right child j in T
    Find in m_j a valid mapping h, such that $h|_{X_i \cap X_j} = f|_{X_i \cap X_j}$ of $Opt(h)$ being '$\surd$';
  Compute score $score(f)$ with formula (1) for X_i only;
  Let $S(f) = score(f) + S(g) + S(h)$;
  If i has parent p in T, and $S(f)$ maximizes over all f' with $f'|_{X_i \cap X_p} = f|_{X_i \cap X_p}$
    Let $Opt(f) =$ '$\surd$';
Return (m_i);

Fig. 4. An outline for the tree decomposition based recursive algorithm GENSUBGISOMO that solves the problem GENERALIZED SUBGRAPH ISOMORPHISM. The algorithm assumes the input of a tree decomposition (T, X) and a map scheme M; it returns table m_i for every node i in T.

mixed graphs H constructed from biopolymer structures, the non-conflict property is also automatically guaranteed. The following is a brief justification for this claim.

Note that the directed edges in graph H form the total order relation $(V(H), \preceq)$ defined as follows: $v \preceq u$ if (i) either $\langle u, v\rangle \in A(H)$, or (ii) $\exists w$, $v \preceq w$ and $\langle u, w\rangle \in A(H)$. This relation needs to be satisfied by any (partial) isomorphism. Assume vertices $u \in X_i$, $v \in X_k$, and k is one of i's descendants in the tree. Assume $v \preceq u$ (the case of $u \preceq v$ is similar). Then in general there exists j on the path from i to k, such that $\exists w \in X_j$, $v \preceq w$ and $w \preceq u$. An induction on the distance of the chain from u to v can assert that the mapping conflict cannot occur between u and v so long as $v \preceq u$.

Theorem 2. GENSUBGISOMO correctly solves the GENERALIZED SUBGRAPH ISOMORPHISM problem for every given tree decomposition and every given map scheme.

Corollary 3. Parameterized algorithm GENSUBGISOMO computes the optimal structure-sequence alignment for every given map scheme of width k.

3.2 Tree decomposition and total alignment time

For graphs with tree width t, theoretical algorithms [4] can find an optimal tree decomposition in time $O(c^t n)$ for some (possibly large) constant c. We introduce a simple greedy algorithm for tree decomposition that practically runs fast on structure graphs.

Given a structure graph H, undirected edges are selected such that removals of these edges from the graph result in an outerplanar graph. The removals of these edges are done by first removing an edge (but not the endpoints) that *crosses* with the maximum number of other edges, and then repeating the same process until the resulting graph contains no crossing edges. Note that two edges (u, v) and (u', v') in H *cross* each other if either $v' \preceq v \preceq u' \preceq u$ or $v \preceq v' \preceq u \preceq u'$ (see section 3.1 for the definition of the partial order $(V(H), \preceq)$.

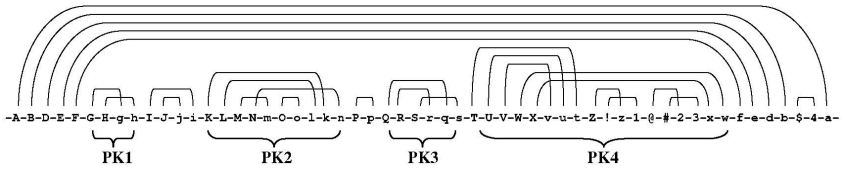

Fig. 5. Diagram of the pairing regions on the tmRNA gene. Upper case letters indicate base sequences that pair with the corresponding lower case letters. The four pseudoknots constitute the central part of the tmRNA gene and are called Pk1, Pk2, Pk3, Pk4 respectively.

A simple recursive algorithm can find a tree decomposition of tree width 2 for the remaining outerplanar graph. Then for each removed edge (u, v), in the tree we place v in every node on the (shortest) path from a node containing v to a node containing u. The tree decompositon shown in Figure 2(b) is obtained by first removing crossing edge $(3, 5)$. Then a tree decomposition for the remaining outerplanar graph is built, which is extended to the tree decomposition for the original graph by placing vertex 3 (in the bold font) in node $\{1, 5, 4\}$ on the path from node $\{1, 4, 3\}$ to node $\{1, 8, 5\}$. This strategy produces a tree decomposition of size at most $2 + c$ if there are c crossing edges removed. In reality, the obtained tree decomposition has much smaller tree width. For example, for the structure graph constructed from the bacterial $tmRNA$ structure (Figure 5), our strategy shall yield a tree decomposition of tree width 4 instead of 9. This algorithm is of linear time $O(|E(H) + |A(H)| + |V(H)|)$.

The running time for algorithm GENSUBGISOMO is $O(k^t t^2 n)$, for map width k, tree width t, number of vertices n in H. For each row in the table, the compliance with subgraph isomorphism needs to be validated and a score computed according to formula (1) (by looking up pre-computed values of functions S_1, S_2, S_3). The former step needs $O(t^2)$ and the latter $O(t^2 + 2t \log_2 k)$ (note that the rows of a table can be ordered to facilitate binary search by the computation for its parent node).

It takes $O(knN)$ time to preprocess the target sequence of length N to construct the sequence graph. Simultaneously, this step pre-computes the values of functions S_1, S_2. The values of function S_3 can then be pre-computed, using time $O(k \sum_{i=1}^{l} l_i^2) = O(knN)$, where l_i is the length of ith loop and l is the number of loops in the structure. Summing up the times needed by the preprocessing, tree decomposition, and ALGORITHM GENSUBGISOMO gives us a loose upper bound $O(k^t nN)$, or $O(k^t N^2)$, for the total time for the structure-sequence alignment.

4 Applications in fast RNA structural homology search

To evaluate the performance of our method and algorithm for structure-sequence alignment, we have applied them to the development of a fast program that can search for RNA structural homologs. We have also conducted extensive tests on finding medium to large RNA secondary structures (including pseudoknots) in both random sequences and biological genomes (bacteria and yeasts) [31]. We summarize our test results in the following.

4.1 Data preparations

The tests on RNA structure searches that we conducted can be grouped into three categories:

1. On 8 RNA pseudoknot-free structures, of medium size (61 - 112 nucleotides), inserted in random sequences of length 10^5,
2. On 6 RNA pseudoknot structures, of medium size (55 - 170 nucleotides), inserted in random sequences of length 10^5, and
3. On 3 RNA pseudoknot structures, of medium to large size (61 - 755), in a variety of genomes of lengths range from 2.7×10^4 to 1.1×10^7.

Each homologous RNA family is modelled with a structure graph. Each undirected edge in the graph represents a stem that is profiled with a simplified Covariance Model (CM) [13]. Each arc in the graph represents a loop (5' to 3') that is profiled with a profile Hidden Markov Model (HMM). In the first two categories of searches, for each family we downloaded from the Rfam database [16] 30 RNA sequences with their mutual identities below 80%. We used them to train the CMs and profile HMMs in the model.

For each family we downloaded from Rfam another 30 sequences with their mutual identities below 80% and use them for search. They were inserted in a random background of 10^5 nucleotides generated with the same base compositions. Using a method similar to the one used in RSEARCH [17], we computed the statistical distribution for the alignment scores with a random sequence of 3,000 nucleotides generated with the same base composition as the sequences to be searched. An alignment score with a Z-score exceeding 5.0 was reported as a hit. Both random sequences and genomes were scanned through with a window of a size correlated with the structure model size. The segment of the sequence falling within the window was aligned to the model with the structure-sequence alignment algorithm presented in the earlier sections.

For the tests of the third category, we searched for three RNA pseudoknot structures: the pseudoknot structure in the 3' UTR in the corona virus family [15], the bacterial tmRNA structure (see Figure 5) that contains 4 pseudoknots [26], and yeast telomerase RNA consisting of up to 755 nucleotides [9]. The structures for these RNAs were trained with 14, 85, and 5 available sequences respectively. The searched genomes for the 3' UTR pseudoknot were Bovine corona virus, Murine hepatitus virus, Porcine diarrhea virus, and Human corona virus, with the average length 3×10^4. The two searched bacteria genomes for the tmRNA were *Haemophilus influenzae* and *Neisseria meningitidis*, with the average length 2×10^6. Yeast genomes,*Saccharomyces cerevisiae* and *Saccharomyces bayanus* of the average length 11×10^6, were used to search for the telomerase RNA.

To obtain a reasonably small value for the parameter k, the map scheme between the structure and the sequence was designed with the constraint that candidates of a given stem were restricted in certain region in the target sequence. For this, we assumed that for homologous sequences, the distances from each pairing region of the given stem to the 3' end follow a Gaussian distribution, whose mean and standard deviation were computed based on the training sequences. For training sequences representing distant homologs of an RNA family, we could effectively divide data into groups so that a different but related structure model was built for each group and used for searches. This method ensures a small value for the parameter k in search models.

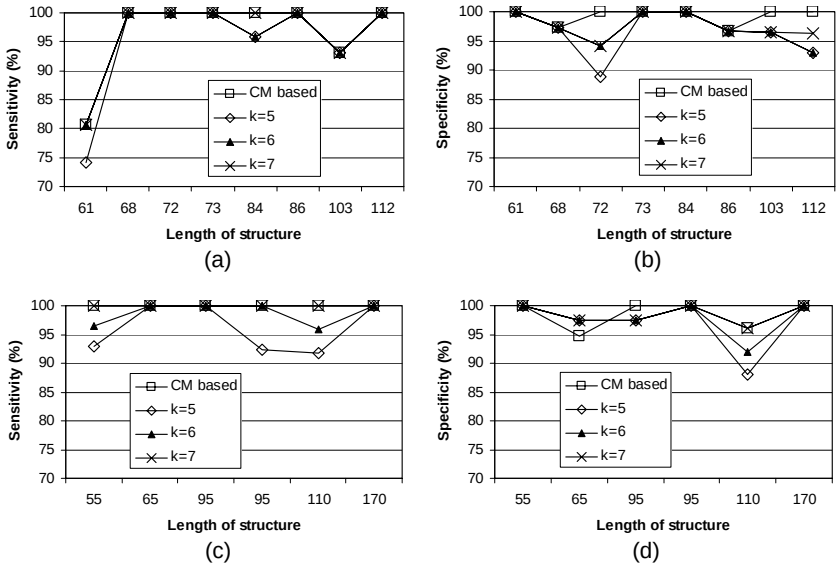

Fig. 6. Performance comparison between the tree decomposition based method and the CM based method on search for RNA structures, (a) and (b) for pseudoknot-free structures, (c) and (d) for pseudoknots.

4.2 Performance evaluations

We conducted the tests on the tree decomposition based search program and on a Covariance Model (CM) based search system[5] and compared the performances of the two. The tests results showed that, on all three categories, parameter $k = 7$ was sufficient for our new search program to achieve the same accuracy as the CM based search system does. But the computation time used by the new method was significantly reduced.

Figure 6(a) and (b) respectively show the sensitivity comparison and specificity comparison between the two search methods on pseudoknot-free RNA structures. These structures were from eight RNA families: Entero_CRE, SECIS, Lin_4, Entero_OriR, Let_7, Tymo_tRNA-like, Purine, and S_box, in the increasing order of their length. The tree decomposition based algorithm performed quite well for $k = 6$ and larger values.

Figure 6(c) and (d) respectively show the sensitivity comparison and specificity comparison between the two search methods on RNA pseudoknot structures. These were from six RNA families: Antizyme_FSE, corona_pk3, HDV_ribozyme, Tombus_3_IV, Alpha_RBS, and IFN_gamma, in the increasing order of their lengths. As for pseudoknot-free structures, the tree decomposition based searches for pseudoknots achieved the same performance as the CM based method for parameter values $k \leq 7$.

[5] We developed this CM based system [21] in the same spirit of Brown and Wilson's work [6] that profiles pseudoknots with intersection of CMs. CM was first introduced by Eddy and Durbin [13] and has proved very accurate in profiling for search of pseudoknot-free RNA structures.

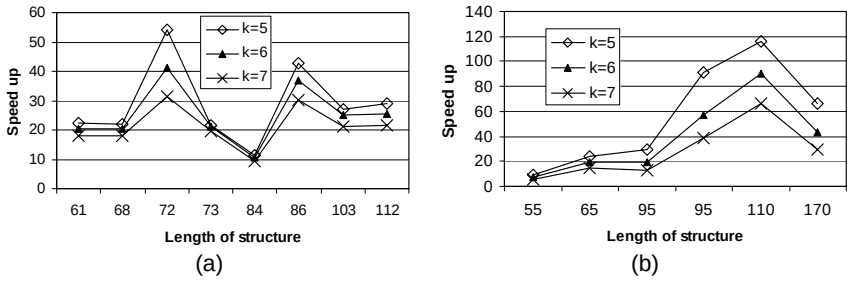

Fig. 7. The speed up of the tree decomposition based method over the CM based method: (a) on pseudoknot-free structures, and (b) on pseudoknot structures.

Figure 7 shows the speed up by the new method over the CM based method, for (a) pseudoknot-free and (b) pseudoknot structures. It is evident that for $k = 7$ the new method was about 20 to 30 times faster than the other method on pseudoknot-free structures. On the pseudoknot structures, typically on Alpha_RBS and Tombus_3_IV containing more than 100 nucleotides, the new method was 66 and 38 times faster, suggesting its advantage in the search of larger and more complex structures.

	ncRNA	Real location		Tree decomposition based			CM based			Genome length
		Left	Right	Left offset	Right offset	Time	Left off	Right off	Time	
3'PK	BCV	30798	30859	0	0	0.053	0	0	1.24	3.1×10^4
	MHV	31792	31153	0	0	0.053	0	0	1.27	3.1×10^4
	PDV	27802	27882	0	0	0.048	0	0	1.17	2.8×10^4
	HCV	27063	27125	0	0	0.047	0	0	1.12	2.7×10^4
tmRNA	HI	472209	472574	-1	-1	44.0	0	0	1700	1.83×10^5
	NM	1241197	1241559	0	0	52.9	0	0	2044	2.2×10^5
TLRNA	SC	307688	308429	-3	-1	492.3	-	-	-	1.03×10^7
	SB	7121529	7122284	-3	2	550.2	-	-	-	1.15×10^7

Fig. 8. Performance comparison between the tree decomposition based method and the CM based method on RNA structure searches on genomes. Offset is between the annotated and the real positions. Time unit is hour.

Figure 8 compares the search results obtained by the two methods on three types of RNA pseudoknots in virus, bacteria, and yeast genomes. Parameter $k = 7$ is used for the parameterized algorithm. Both methods achieve 100% sensitivity and specificity. It clearly shows that the new method had a speed-up of about 30 to 40 times over the other method for searches in virus and bacteria genomes. With the new method, searching genomes of a moderate size for structures as complex as tmRNA gene (see Figure 5) only took days, instead of months. Searching a larger genome such as yeast for larger structure like telomerase RNAs was also successful, a task not accomplishable by the CM based system within a reasonable amount of time.

5 Conclusions

We introduced a novel method and an efficient parameterized algorithm for the structure-sequence alignment problem by exploiting the small tree width of biopolymer structure graphs. The algorithm was applied to the development of a fast search program that is capable of accurately identifying complex RNA secondary structure including pseudo-knots in genomes [31]. Our method provides a new perspective on structure-sequence alignment that is important in a number of bioinformatics research areas where structure plays an instrumental role. In particular, we expect this method to yield very efficient algorithms for protein threading [35].

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. S. Arnborg, J. Lagergren, and D. Seese, "Easy problems for tree-decomposable graphs", *Journal of Algorithms* 12: 308-340, 1991.
3. J. Bowie, R. Luthy, and D. Eisenberg, "A method to identify protein sequences that fold into a known three-dimensional structure", *Science* 253: 164-170, 1991.
4. H. L. Bodlaender, "A linear time algorithm for finding tree-decompositions of small treewidth", *SIAM Journal on Computing* 25: 1305-1317, 1996.
5. S.H. Bryant and S.F. Altschul, "Statistics of sequence-structure threading", *Curr. Opinion Struct. Biol.* 5: 236-244, 1995.
6. M. Brown and C. Wilson, "RNA pseudoknot modeling using intersections of stochastic context free grammars with applications to database search", *Pacific Symposium on Biocomputing*, 109-125, 1995.
7. L. Cai, R. Malmberg, and Y. Wu, "Stochastic Modeling of Pseudoknot Structures: A Grammatical Approach", *Bioinformatics*, 19, $i66 - i73$, 2003.
8. T. Dandekar, S. Schuster S, B. Snel, M. Huynen, and P. Bork, "Pathway alignment: application to the comparative analysis of glycolytic enzymes", *Biochemical Journal.* 1: 115-24, 1999.
9. A. T. Dandjinou, N. Lévesque, S. Larose, J. Lucier, S. A. Elela, and R. J. Wellinger, "A phylogenetically based secondary structure for the yeast telomerase RNA.", *Current Biology*, 14: 1148-1158, 2004.
10. J.A. Doudna, "Structural genomics of RNA", *Nature Structural Biology* 7(11) supp. 954-956, 2000.
11. R. Downey and M. Fellows, *Parameterized Complexity*, Springer, 1999.
12. S.R. Eddy, "Computational genomics of non-coding RNA genes", *Cell* 109:137-140, 2002.
13. S. Eddy and R. Durbin, "RNA sequence analysis using covariance models", *Nucleic Acids Research*, 22: 2079-2088, 1994.
14. D. Eppstein, "Subgraph isomorphism in planar graphs and related problems", *Journal of Graph Algorithms and Applications*, 3.3: 1-27, 1999.
15. S. J. Geobel, B. Hsue, 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.
16. S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S. R. Eddy, "Rfam: an RNA family database", *Nucleic Acids Research*, 31: 439-441, 2003.
17. R. J. Klein and S. R. Eddy, "RSEARCH: Finding Homologs of Single Structured RNA Sequences.", *BMC Bioinformatics*, 4:44, 2003.

18. R.H. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is NP-complete", *Protein Engineering* 7: 1069-1068, 1994.
19. R.H. Lathrop, R.G. Rogers Jr, J. Bienkowska, B.K.M. Bryant, L. J. Buturovic, C. Gaitatzes, R.Nambudripad, J.V. White, and T.F. Smith, "Analysis and algorithms for protein sequence-structure alignment", in *Computational Methods in Molecular Biology*, Salzberg, Searls, and Kasif ed., Elsevier, 1998.
20. H-P. Lenhof, K. Reinert, and M. Vingron. "A polyhedral approach to RNA sequence structure alignment", *Journal of Computational Biology* 5(3): 517-530, 1998.
21. C. Liu, Y. Song, R. Malmberg, and L. Cai, "Profiling and searching for RNA pseudoknot structures in genomes", *Lecture Notes in Computer Science* 3515, 968-975, 2005.
22. 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.
23. S.B. Lyngso and C.N. Pedersen, "RNA pseudoknot prediction in energy-based models", *Journal of Computational Biololgy* 7(3):409-427, 2000.
24. J. Matousek and R. Thomas, "On the complexity of finding iso- and other morphisms for partial k-trees", *Discrete Mathematics*, 108: 343-364, 1992.
25. E.M. Marcotte, P. Matteo, HL. Ng, D.W. Rice, T.O. Yeates, and D. Eisenberg, "Detecting protein function and protein-protein interactions from genome sequences", *Science* 285: 751-753.
26. N. Nameki, B. Felden, J. F. Atkins, R. F. Gesteland, 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.
27. D.D. Pervouchine, "IRIS: Intermolecular RNA Interaction Search", *Genome Informatics* 15(2): 92-101, 2004.
28. K. Reinert, H-P. Lenhof, P. Mutzel , K. Mehlhorn , and J.D. Kececioglu, "A branch-and-cut algorithm for multiple sequence alignment", *Proceedings of the first annual international conference on Computational molecular biology*, 241-250, 1997.
29. E. Rivas and S. R. Eddy, "Noncoding RNA gene detection using comparative sequence analysis", *BMC Bioinformatics*, 2:8, 2001.
30. N. Robertson and P. D. Seymour, "Graph Minors II. Algorithmic aspects of tree-width", *Journal of Algorithms* 7: 309-322, 1986.
31. Y. Song. C. Liu, R. Malmberg, F. Pan, and L. Cai, "Tree decomposition based fast search of RNA secondary structures in genomes", *Proceedings of 2005 IEEE Computational Systems Biology Conference*, in press.
32. Y. Uemura, A. Hasegawa, Y. Kobayashi, and T. Yokomori, "Tree adjoining grammars for RNA structure prediction", *Theoretical Computer Science*, 210: 277-303, 1999.
33. G. Wang and R.L. Dunbrack, Jr. "PISCES: a protein sequence culling server", *Bioinformatics* 19: 1589-1591, 2003.
34. D. Xu, M.A. Unseren, Y. Xu, and E.C. Uberbacher, "Sequence-structure specificity of a knowledge based energy function at the secondary structure level", *Bioinformatics* 16:257-268, 2000.
35. J. Xu, "Rapid side-chain packing via tree decomposition", In *Proceedings of 2005 International Conference on Research in Computational Biology*, to appear.
36. J. Xu, M. Li, D. Kim, and Y. Xu. "RAPTOR: optimal protein threading by linear programming", *Journal of Bioinformatics and Computational Biology*, 1(1):95-113, 2003.
37. Y. Xu, D. Xu, and E.C. Uberbacher, "An efficient computational method for globally optimal threading", *Journal of Computational Biology* 5(3):597-614.