ACCURATE RNA 3D MODELING WITH BACKBONE K-TREE MODEL

by

XINGRAN XUE

(Under the Direction of Liming Cai)

ABSTRACT

Given the importance of non-coding Ribonucleic acids (RNAs) to cellular regulatory functions, it would be highly desirable to have accurate computational prediction of RNA 3D structure, a task which remains challenging. Even for a short RNA sequence, the space of tertiary conformations is immense; existing methods to identify native-like conformations mostly resort to random sampling of conformations to achieve computational feasibility. However native conformations may not be examined and prediction accuracy may be compromised due to sampling. State-of-the-art methods have yet to deliver satisfactory predictions for RNAs of length beyond 50 nucleotides.

This dissertation presents a novel 3D modeling method for RNA 3D prediction from predicted nucleotide interactions. The research is based on a novel graph model, called a *backbone k-tree*, to tightly constrain the nucleotides conformation considered for RNA 3D structures with the objective function defined over cliques of the *k*-tree. It is shown that the model can efficiently predict the optimal and suboptimal structures in atomic detail from the query interactions along with the k-tree. The results indicate that in most cases the new model can predict the 3D structure with a high accuracy. It thus provides a useful tool for the accurate prediction of RNA 3D structure.

INDEX WORDS: K-tree, RNA 3D structure prediction, RNA 3D modeling.

ACCURATE RNA 3D MODELING WITH BACKBONE K-TREE MODEL

by

XINGRAN XUE

B.S. Nanjing University of Post and Telecommunication, China, 2008

A Dissertation Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

ATHENS, GEORGIA

2015

ACCURATE RNA 3D MODELING WITH BACKBONE K-TREE MODEL

by

XINGRAN XUE

Major Professor:	Liming Cai
Committee:	Russell L. Malmberg
	Tianming Liu

Electronic Version Approved:

Suzanne Barbour
Dean of the Graduate School
The University of Georgia
December 2015

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

In the past decade, there have been many revelations of the importance of non-coding RNAs to cellular regulatory functions and thus a growing interest in the computational prediction of RNA 3D structures [1,2,3]. Backbone *k*-Tree Modeling is a suite of RNA-modeling tools. Here we provide the details of the methods and functions of RNA 3D structure prediction utility and related tools, with the aims of clarifying the scientific basis for the structure prediction, demonstrating capabilities, and reviewing limitations.

1.1 Motivation

This dissertation was motived by the task to predict the 3D structure of RNA given the primary structure as input. It is now widely recognized that RNA is a fundamental biological macromolecule with many biological functions at all stages of cellular life [4,5,6,7]. Besides the well-accepted functional properties of messenger RNA, transfer RNA and ribosomal RNA, many new non-coding RNAs are now known to perform catalytic regulatory roles that are essential to an organism's survival and evolution. [8,9,7,10]

The structural features of RNAs are of major importance to their biological functions because sequence alone does not provide sufficient functional information. In the past

two decades, significant progress has been made in RNA 3D structure determination through the use of X-ray crystallography and nuclear magnetic resonance (NMR) [11,12]. 3D structures have been classified and stored in databases for many RNAs [13]. However, experimentally determining the structure of a protein or RNA is a complicated and laborious task which is time-consuming and may not yield the structures with high throughput processes. Besides, the number of the ncRNAs with 3D structures determined is just a very small percentage of the estimated total number of ncRNAs [14,15,16].

One of the goals in RNA computational biology is to provide insights into how and RNA sequence leads to the specific fold of the RNA, many efforts had been made in this area [1,2]. Nevertheless, RNA 3D structure prediction from a single RNA sequence is a significant challenge. One major unresolved issue is the immense space of 3D conformations even for a short RNA sequence [17]. Existing methods usually employ random sampling algorithms for computation feasibility, which assemble sampled tertiary motifs into native-like structures [17,18,19,20,21,22,23,24]. To reduce the chance to miss native structures, the assembly algorithms have mostly been guided with constraining structural models. For example, MC-Fold/MC-Sym [22] assumes the 3D structure consists of 4-nt cyclic tertiary motifs constructible from the predicted secondary structure. Rosetta [19,20] *de novo* assembles 3D structure from a database of 3-nt 3D fragments. Other methods follow samplings that preserve the secondary structure [25,26,27] or intervention from human experts [28,29]. However, these constraining models do not necessarily ensure that native conformations are examined. In particular, the state-of-the-art methods have yet to deliver the desired prediction accuracy for RNA sequences of lengths beyond 50 [1].

1.2 This Dissertation

In this dissertation, we introduce a 3D modeling method for RNA 3D structure prediction. The work is based on a novel method to predict accurate RNA tertiary structure through predicting nucleotide interactions from sequences as an intermediate step [30]. Our method is guided by a novel graph model called a backbone *k*-tree, for small integer *k*, to globally constrain the nucleotide conformations considered for RNA 3D structures. In such a *k*-tree graph, nucleotides are organized into groups of size $k + 1$, such that nucleotide relations are permitted only for nucleotides belonging to the same group and groups are connected to each other with a tree topology. This model was inspired by our recent discovery of the small *treewidth* [31,32] of the nucleotide interaction relation graphs for more than 3,500 RNA chains extracted from 1,984 RNAs whose structure has been resolved [30].

The backbone *k*-tree based 3D modeling method is specifically designed to produce a 3D model from a predicted backbone *k*-tree and an associated set of nucleotide interactions. In this method, we assign one geometric motif to the interaction pattern associated with every $(k + 1)$-clique in the predicted k-tree. The set of consistent geometric motifs assigned to all cliques will form the predicted 3D structure. The geometric motifs were extracted from RNA Structure Atlas [33] in much the same way as the interaction patterns were extracted. Since there may be many motifs to consider for an interaction pattern, we defined an objective function to minimize the aggregated root-mean-square-deviations between every pair of "neighboring" $(k + 1)$-cliques in the predicted *k*-tree topology. This method is computationally efficient; because, given a k-tree, optimization

computation can be done in linear time for the fixed k value [34,35]. The algorithm has been implemented into a program called BkTree3D.

To ensure that the computed optimal structure can actually yield the native 3D structure, our method defines the scoring function over geometry candidates of detailed interaction patterns within every group of k+1 nucleotides. We consider nucleotide interactions from the established geometric nomenclatures and nucleotide interaction families [36,37], including base-base, base-phosphate, and base-ribose as well as base-stacking interactions [38,39].

To evaluate our 3D modeling method for RNA structure prediction, we tested BkTree3D on a benchmark set of 43 high resolution RNAs, which had been used to survey a number of state-of-the-art tertiary structure prediction methods [1]. For every one of the 43 RNAs, the 3D modeling program produced an optimally predicted 3D structure whose RMSD was calculated against the resolved structure. The deformation index (DI) [40], a measure that accounts for both root-mean-square deviation (RMSD) values and MCC [1], the quotient between them, was also calculated. The result data suggest a great potential of our method for RNAs beyond short lengths. The average RMSD is 4.6 Å. In particular, 14 out of the 18 RNAs with lengths exceeding 50 achieved RMSD values below 10 Å; To compare with other 3D structure prediction methods, we present the performance values on the 4 representative RNAs chosen in [1] which typically contain two hairpins and two junctions. Since both MC [22] and Rosetta [19] allow prediction of multiple optimal or suboptimal folds, we chose to use the averaged values of their solutions. For every one of the 4 RNAs, the RMSD achieved by BkTree3D is significantly smaller than the averaged value achieved by MC and Rosetta.

The dissertation is organized as follows. In the second chapter we will give introduction of RNA tertiary structure, review the current knowledge on RNA interactions and describe the prediction algorithms. In chapter 3 we will define some graph theory concepts which will be used in the remaining chapters. We will introduce the nucleotide interaction relations (NIRs) and backbone k-tree model. We will formally define the nucleotide interaction prediction problem and give the solution. In chapter 4 we will introduce the backbone 3-tree 3D modeling method, algorithm, performance and limitations. In chapter 5 we will discuss how to cope with the limits of backbone 3-tree 3D prediction. We will summarize our work in the conclusion section.

CHAPTER 2

BACKGROUND

A biological molecule (RNA, or protein) is a sequence of linearly chained bio-residues (nucleic acids or amino acids). Chemical and physical attraction between neighboring and distant residues that allows the formation of chemical substances that can stabilize its structure. While the underlying biological rules are yet to be fully understood, efforts has been made through computational modeling and analysis in past decades.

The information of an RNA sequence can be categorized at three different levels: primary structure, secondary structure and tertiary structure as shown in Fig. 2.1. The primary structure is the linear sequence of the bases (A, C, G, U). The secondary structure is the folding of the RNA molecule around itself, and reveals certain sub-structures such as loops that have some important features. The main component of the secondary structure is the pairing of bases, called base pairs. Adjacent base pairs can form a "stack", like a stack of plates and it is called helix. The base pairings are usually, but not always, Watson-Crick pairs between C-G and A-U.

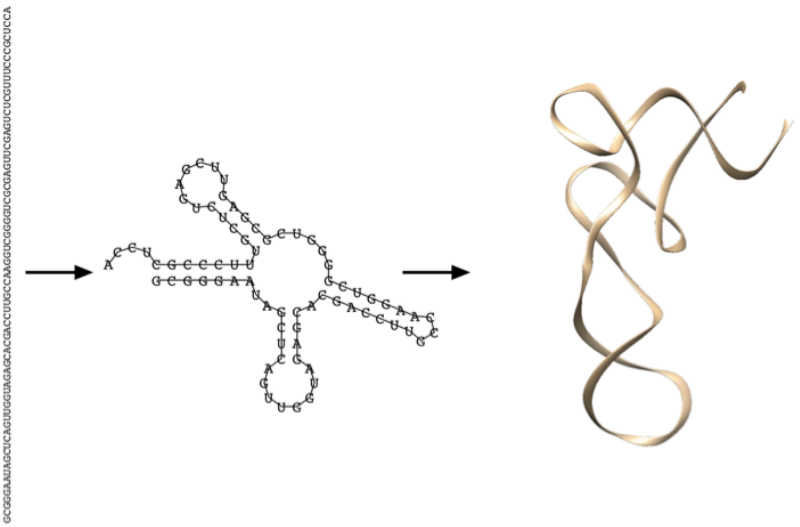


Fig. 2.1 RNA structural levels (from left to right) primary, secondary and 3D structure. Complementary regions in the primary sequence can form base pairs which define a pattern of loops and helices that constitute the secondary structure. Secondary structure elements interact with each other in space and form the tertiary structure.

Understanding RNA structure and function relies on our ability to identify RNA's major structural components. RNA molecules have been studied extensively at the secondary structure level [41,42,43], where building blocks include helical stems and single-stranded regions such as hairpins, internal loops, and junctions. Stems are formed by Watson-Crick base pairs GC and AU, along with the GU wobble base pair. A hairpin loop is a single-stranded region that folds back on to itself via regions of complementary base pairs. The single-stranded region between two stems is known as an internal loop, while a junction can be defined as the point of connection between three or more helical stems. Single-stranded regions can form pseudoknots when base pairs intercalate. Compact structures of RNAs can be formed by these basic secondary structure elements through tertiary interactions [44,45,46,47].

2.1 RNA Hierarchical Structure

A better understanding of RNA structures can be gained through understanding the interactions at each hierarchical structure level.

At the level of the primary sequence, the modular units are individual nucleotides, linked from 5′ to 3′ by phosphodiester bonds. See details in Fig. 2.2. When RNA molecules fold, the nucleotides interact with each other in characteristic ways. The most studied interactions are bases involved, base to base, base to sugar, and base to phosphate interactions. Sugar to sugar, sugar to phosphate, phosphate to phosphate and metal involved interactions also occur and contribute to stabilizing the complex RNA structures, but are much harder to detect from only sequence information [48,49].

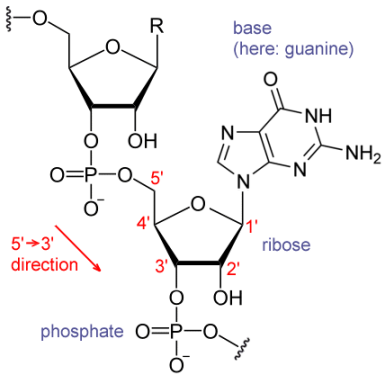


Fig. 2.2. Chemical structure of RNA. Each nucleotide consists a ribose sugar, with carbons numbered from 1' through 5'. A base is attached to 1' position. A phosphate group is attached to the 3' position of one ribose and 5' position of the next. Figure adopted from Wikipedia.

In secondary structure, single-stranded RNA molecules fold back on themselves through complementary Watson–Crick base pairs in an anti-parallel fashion. These complementary Watson–Crick base pairs stack on each other form Watson–Crick helices, the fundamental units of secondary structure. The helices are generally short (no longer than 10–15 WC pairs) because they are interrupted or terminated at their ends by hairpin, internal or junction loops. Also, the 3D structure of helices is generally considered as a

rigid object. The presence of a hydroxyl group at the 2' position of the ribose sugar causes the helix to adopt A-form geometry. Study shows about 60% of the nucleotides in a RNA sequence involve in Watson–Crick base pairs [50,51].

The 3D structures of a relatively small number of RNA molecules have been determined to the atomic resolution by X-ray crystallography or NMR each year. Although small in size compared to the protein 3D database, the RNA 3D structure database has expanded rapidly in the last few years. This database shows that most "loops" in 2D representations in fact form specific *3D motifs*, characterized by non-Watson–Crick base-pairing, base-stacking and base-phosphate interactions between nucleotides within the loops. Besides, most of the non-base paired bases participate in base-stacking, base–phosphate, or RNA–protein interactions. Thus, the loops comprise a significant fraction of a structured RNA molecule and most of their nucleotides have non-Watson-Crick interactions with other nucleotides.

2.2 Challenge in How RNA Folds

Modeling the 3D shape of RNA is complicated by the fact that they usually loosely compacted and have "homopolymer" nature of nucleic acids. The greater structural diversity of RNAs poses a challenge to computation. Moreover, the sensitivity of RNA

structures to ions, solvent, metabolites, and other biomolecules has made RNA structure determination at atomic resolution more difficult than for proteins.

The first roadblock to modeling RNA is that the conformational space is immense [49]. If RNA structures are compact then their volumes are expected to scale as $VR_G^3a^3N$, where R_G is the radius of gyration, a is a GG effective monomer length [52]. More generally, Flory showed that $R_G aN^v$, where the Flory exponent v = 1/3 for maximally compact structures. Computation of the sizes of RNA structures using the PDB coordinates reveals that R_G, follows the Flory scaling law, namely, $R_G = a_N N^{1/3}$ Å [51]. The pre-factor, a_N = 5.5 Å, corresponds approximately to the average distance between the phosphate groups (≈5.8 Å) along the ribose-phosphate backbone. In contrast to proteins, for example, for a given *N,* the approximate volume of RNA is larger than that of proteins whose R_G scales as R_G = 3.1 $N^{1/3}$ Å [50,51]. In other words, RNA molecules are more loosely packed than proteins. The difference is due to the larger size of the nucleotides compared to amino acids and the nature of interactions that stabilize the folded structures of RNAs and proteins.

Further complicating the issue is that the building blocks of RNA are the four nucleotides each with a base, ribose, and phosphate groups. The bases (two purines and two pyrimidines), that are chemically similar, interact with each other either through hydrogen bonding or base stacking. The secondary structural elements (helices, loops, bulges) are independently stable which gives the impression that the three dimensional assembly is built much the same way as complicated architecture using prefabricated

building blocks. However, the difficulty arises because of the chemical similarity of the nucleotides.

Additionally, the bases, their ability to form hydrogen bonds are all hydrophobic. The hydrophobic backbone along with lack of diversity in the bases makes RNA closer to a "homopolymer" than polypeptide chains [53]. The "homopolymer" nature of nucleic acids results in RNA structures being able to adopt alternate structures i.e., the stability differences between the folded and the other misfolded structures is not large. As a result, the energy landscape of RNA, even at the secondary structural level, contains many metastable conformations.

Last but not least, the folding mechanisms can be greatly altered by the occurrence of counter-ions because of the polyelectrolyte nature of RNA. In particular, the important role of valence, shape and size of the counter-ions in modulating the secondary structures and possibly altering them during the course of 3D structure formation, are difficult to predict [54].

Amid these complexities, it is fortuitous that the conformational preferences of the interactions between the nucleotides are not as diverse as would initially be expected.

2.3 RNA Motif and Nucleotide Interactions

2.3.1 RNA Motif

"Modular RNA 3D motifs are autonomous sets of interacting nucleotides that form a defined 3D structure" [55]. This definition of RNA 3D motif emphasizes the important role of physical interactions between nucleotides but the nucleotides similarities from the sequences. While the Watson–Crick helix is the most important RNA 3D motif, in this work we will focus on motifs that comprise non-Watson–Crick interactions. RNA 3D motifs serve the guiding and stabilizing roles in RNA folding. The RNA-RNA interactions that define 3D motifs include base-pairing, base-stacking, and base-backbone interactions. In terms of base pairing patterns, base pair interactions can be classified into twelve geometric families in terms of pairs of interacting edges, which can be Watson-Crick, Hoogsteen, Sugar, and glycoside bond orientation cis and trans, as classified by Leontis and Westhof [2,38,36].

Different sequences that can form the same 3D structure and carry out similar biological functions. 3D motifs are usually recurrent; homologous RNA molecules normally contain the same motifs at corresponding positions in their structures consistent with evolutionary conservation. They can also occur in unrelated RNA molecules as a result of convergent evolution. Studies have revealed that same recurrent motif sharing a number of core-nucleotides can be superposed in the 3D space [38]; each core-nucleotide from the equivalent position possesses the same interaction relationship to the others. For example, the stem helix is the most common 3D motif; two helices of the same length and stacking interactions but differing significantly in sequence can be superimposed nucleotide by nucleotide in a geometrically similar way [53,46].

2.3.2 Nucleotide Interactions

To take advantage of the resolved RNA structural data and RNA 3D motifs, significant efforts have been made to classifying the RNA interactions. One major breakthrough was made by Leontis and Westhof groups in early 2000s [36]. Thus we adopted such nucleotide interaction classifications and geometric nomenclatures from their established study.

RNA bases, purines and pyrimidines, present three edges for hydrogen-bonding interactions with each other, the Watson–Crick, the Hoogsteen, and the Sugar Edges, as illustrated for adenosine (A) in the left panel of Fig. 2.3 [36]. The RNA Sugar Edge includes the 2′-hydroxyl, a functional group that distinguishes RNA from DNA and plays an important role in RNA tertiary interactions and RNA chemistry. Bases can pair using any of the six combinations of the three edges, for example, the Watson–Crick Edge of one base with the Watson–Crick, Hoogsteen, or Sugar Edge of a second base. In addition, for each combination of edges, the bases can approach each other in two orientations, which are called *cis* and *trans,* by analogy to the geometric isomerism at carbon–carbon double bonds. As shown in the right panel of Fig. 2.3, in *cis* basepairs, the glycosidic bonds joining the bases to their respective sugar moieties are found on the same side of the axis shown in grey. This axis is defined by the hydrogen bonds joining the base edges.

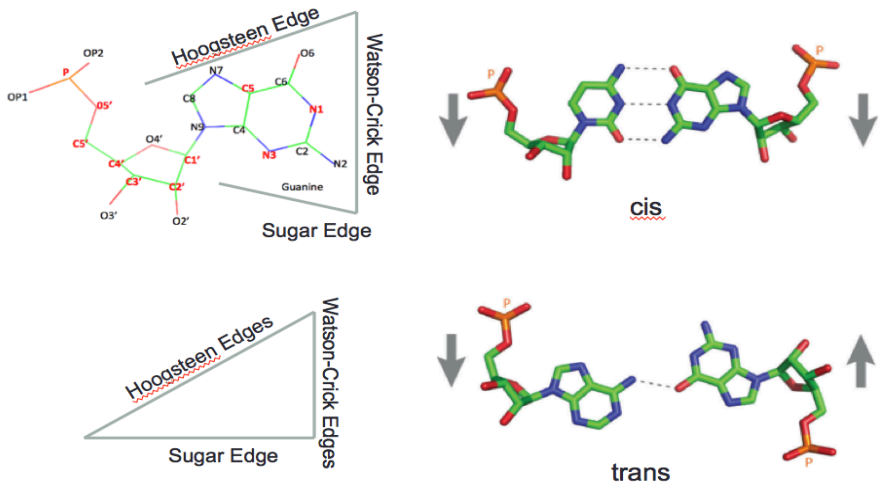


Fig. 2. 3 Base edges and base-pair geometric isomerism. (Upper left) The structure of adenosine showing the three base edges (Watson–Crick, Hoogsteen and Sugar-edge) available for hydrogen–bonding interactions. (Lower left) Representation of RNA base as a triangle. (*Right*) *cis* and *trans* base-pairing geometries, illustrated for two bases interacting with Watson–Crick edges. Figure adopted from [36].

2.4 Review of Other Work

Developments in RNA structure prediction studies have been previously extensively reviewed. Compared to protein folding, the RNA folding problem is at an early stage: current 3D RNA folding algorithms require manual manipulation or are generally limited to simple structures in terms of size and topology. However, many groups have now been solving this problem by a variety of techniques as organized in programs like NAST [21], MC [22], FARFAR [19], and others. These methods differ in the input data, prediction accuracy, and nucleotide representation.

A recent review examined the performance of 3D structure prediction algorithms, namely iFoldRNA, FARNA, NAST, and MC-SYM, for an RNA dataset of 43 structures of various lengths and motifs [1]. They found that most predictions have large root-mean square deviations from the crystallization structures (e.g., RMSD>6Å). While the prediction accuracy improves with added knowledge from the 2D structure and 3D contacts, the lack of appropriate functions that derive compact RNA structures and the failure to detect long-range contacts remain issues to be addressed. Below we describe the most recent approaches.

The nucleic acid simulation tool NAST [21] developed by Jonikas et al. is a molecular dynamics simulation tool consisting of a knowledge-based statistical potential function applied to coarse grained model with resolution of one bead per nucleotide residue. NAST requires secondary structure information and accepts tertiary contacts to direct the prediction. NAST's greatest strength is modeling large RNA molecules (e.g., 160 nt), which is a limitation to most of other programs. Overall, when only secondary structure information is considered, accurate prediction is limited to RNA structures with simple topologies such as hairpins with less than 34nt (8Å average RMSD). However, when input information from tertiary contacts is additionally provided, NAST can improve the prediction accuracy.

iFoldRNA [18] is a web-based program developed by the Dokholyan's group. It predicts RNA structures using a coarse grained model of three beads per nucleotide through molecular dynamics (MD) sampling. iFoldRNA does not require secondary structure

information and can rapidly predict structures for small RNAs (<50nt). However, as the RNA size increases, difficulties arise regarding long-range tertiary contacts (23Å average RMSD).

Das et al. introduced a 2010 update to Rosetta's Fragment Assembly of RNAs (FARNA) [19]. FARNA is an energy-based program that predicts RNA 3D structures from a sequence using one bead of a coarse-grained model. The folding simulation is guided by a knowledge-based energy function that takes into account both backbone conformations and side-chain interaction preferences observed in solved structures. Standard sampling difficulties reflect both algorithmic and force-field limitations. FARFAR is only applicable to small RNA (6–20nt), and has variable accuracy, from less than 1Å RMSD to more than 10Å (for 4-way junctions).

MC-Fold/MC-Sym [22] developed by Major's group also using the fragment assembly method for modeling RNA 3D structure. MC-Fold predicts RNA secondary structure first, using a free energy minimization function and the fragment database built from resolved RNAs. Then MC-Sym builds full-atom models of RNA structures using the 3D version of the nucleotide cyclic motif fragments. Similar as other sampling approaches, MC-Sym usually fails on longer sequence with complicated topologies.

Fresllsen et al. presented another coarse grained model called BARNACLE [56], which use a probabilistic model that represents RNA conformational flexibility using circular analogues under Gaussian distributions. The continuity of the local conformational space allows for less metastable samplings. Using secondary structure information,

BARNACLE generates reasonable RNA-like structures (<10Å RMSD) for small RNA molecules (<50nt). However, most RNA structures with rich topologies such as junctions and long-range contacts are longer than 50nt and cannot be predicted due to an increase in complexity of the probabilistic model.

An alternative approach to automated programs is ASSEMBLE [24], a manual-input program by Jossinet et al. that, like RNA2D3D [29], uses secondary or tertiary structure information from homologous RNAs to build a approximation RNA 3D model. Using an intuitive graphical interface, ASSEMBLE allows the manual insertion of base pairs and 3D motifs, as well as torsion angle modifications, rotations, and translations of modular elements. ASSEMBLE permits the input of electron density maps to improve the RNA model. While these user-input tools are practical, they rely on manual application of expert knowledge.

In general, for RNA sequences of medium to large sizes (50-100nt), even the best prediction methods lead to large RMSD values (20Å on average). Alternatives to RMSD measurements for RNAs have thus been suggested for RNA structural comparisons. Parisien et al. proposed assessing the global fold of a RNA at the nucleotide resolution by measuring a more "relative" RMSD value [40]. For example, for a de novo prediction of 100nt RNA, the RMSD should be within 20Å to be considered as an acceptable prediction. Such alternative comparison approaches should help distinguish successful models with RNA-like structures from less successful predictions.

CHAPTER 3

MODEL

In this chapter, we will introduce a novel method to predict nucleotide interactions from known or predicted canonical base pairs as an intermediate step towards accurate prediction of RNA 3D structure. Our method is guided by a novel graph model called *backbone k-tree* to predict nucleotide interactions [57,30]. The performance will be discussed in next chapter along with our 3D modeling method.

3.1 Graph Theory

In this section, some basic notations of graph theory are discussed. Graphs are mathematical structures, which are used to model pairwise relations between objects. They have proven to be an effective modeling tool in many disciplines, where vertices in the graph model are objects in the problem of interest, and the edges between the vertices are used to model the corresponding interactions among the objects.

DEFINITION 1. A graph G is defined as tuple (V, E), where V is the set of vertices and E is the set of edges. An edge $e = \{u, v\}$ is an unordered pair of distinct vertices.

Given an edge $e = \{u, v\}$ we say that u and v are the *endpoints* of e. A vertex is *incident* to an edge if it is one of the endpoints of that edge; so for any given two distinct vertices in a graph, we say that they are adjacent if they are both incident to a common edge. The

vertices adjacent to a vertex u are called its *neighbors*. A *path* in G is a non-empty graph $P = (V',E')$, $V' \subseteq V$ and $E' \subseteq E$, where $V' = \{v_1, v_2, ..., v_k\}$, $E' = \{v_1v_2, v_2v_3, \ldots, v_{k-1}v_k\}$ and the v_is are all distinct. A *cycle* $C = (V', E')$ is defined similarly with the exception that $v_1 = v_k$. Graph G is called *connected* if there is path between every pair of vertices in G. A *tree* T is a connected graph with no cycles. In a *rooted tree* one of the vertices is distinguished as the root. In a rooted tree, for any two neighboring vertices the one closer to the root is called the *parent* and the other is called a *child*.

Given a graph G, a vertex $v \in V$, and an edge $e \in E$, then $G - v$ denotes the graph ($V \setminus \{v\}$, $E \setminus \{e \in E : v \text{ is incident to } e\}$) and $G - e$ denotes the graph (V,$E \setminus \{e\}$). These two operations are called *deleting a vertex* and *deleting an edge*, respectively. Any graph that can be obtained via these two operations is a subgraph of G. If all the vertices of the graph $G = (V, E)$ are pairwise adjacent, then G is called *complete* and is denoted by K^n where n is the number of vertices. e.g., K^2 is an edge and K^3 is a triangle. A *clique* is a complete subgraph. A subgraph $G' = (V', E')$ of $G = (V, E)$ is called *induced* if E' contains all of the edges in E which have both endpoints in V'.

It is often the case that a problem which is intractable for general graphs becomes easy to solve when restricted to graphs with simple structures, such as trees. The reason can be intuited in the following way: Let T be rooted tree and T_u denote the sub-tree induced by u and its descendants. Then if $v_1, v_2, \ldots, v_k$ are the children of u, the solution of T_u is obtained from solutions on $T_{v_1}, T_{v_2}, ..., T_{v_k}$ considering how they interact at u. Many NP-hard graph problems can be solved efficiently when the underlying graph is restricted

to be a tree or tree-like structure [58]. The notion of *Tree-decomposition* was introduced to represent graphs that are of tree-like structures [59].

DEFINITION 2. A tree-decomposition of a graph $G = (V, E)$ is a pair

$$(\{X_i | i \in I\}, T = (I, F))$$

Where $\{X_i | i \in I\}$ is a family of subsets of V, one for each node of T, and T is a tree such that

1. $\bigcup_{i \in I} X_i = V$;

2. For each edges $\{u, w\} \in E$, there is an $i \in I$ with $u \in X_i$ and $w \in X_i$;

3. For all $i, j, k \in I$: if j is on the path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *treewidth* of a tree-decomposition $(\{X_i | i \in I\}, T = (I, F))$ is $max_{i \in I} |X_i| - 1$. The *treewidth of graph G* is the minimum treewidth over all possible tree-decompositions of G. Computing the treewidth and finding a corresponding tree-decomposition of a graph is an NP-hard problem. Research in algorithmic graph theory has revealed that many hard problems are easy when restricted to graphs of small treewidth [60,61,62].

3.2 Nucleotide Interaction Relation(NIR)

In this work, we consider all known types of nucleotide interactions of atomic-resolution. In particular, with the base triangle model consisting of Watson-Crick (W), Hoogsteen (H), and sugar (S) edges, base-base interactions have been fully characterized into rich 12 geometric types and 18 interaction families, according to involved edges, *cis* or *trans*, and *parallel* or *anti-parallel*, observed in crystal structures. For example, the cWW family contains, in addition to the canonical (i.e., *cis* Watson- Crick) base pairs, many non-canonical base-base interactions through W edges. More recently, classifications of nucleotide interactions have been extended to base-backbone interactions. There are 10 families identified for base-phosphate interactions based on the position of the interacting hydrogen atom in the base. Similarly, 9 additional families have been identified for base-ribose interactions. A few base stacking interactions have also been classified. Table 3.1 summarizes these classes of nucleotide interactions, which also includes the backbone interaction between two neighboring nucleotides [36,38].

We use notation $< X, Y, t >$ for a type t interaction between nucleotides X nucleotide Y (from 5' to 3'), where $X, Y \in \{A, C, G, U\}$. Let $I_{XY} = \{< X, Y, t >: t$ is an interaction type$\}$ and $I = \bigcup_{X,Y \in \{A,C,G,U\}} I_{XY}$.

Table 3.1 Categories, types and families of RNA nucleotide interactions, summarized from Leontis and Westhofs' works. It also includes the phosphodiester interaction between two neighboring nucleotides.

Categories	Types (Interaction Families)	Number
Base pairs	cWW, tWW, cWH, tWH, cHW, tHW, cWS, tWS, cSW, tSW, cHH, tHH, cHS, tHS, cSH, tSH, cSS, tSS	18
Base-phosphates	0BPh, 1BPh, 2BPh, 3BPh, 4BPh, 5BPh, 6BPh, 7BPh 8BPh, 9BPh	
Base-riboses	0BR, 1BR, 2BR, 3BR, 4BR, 5BR, 6BR, 7BR, 9BR	9
Bases stackings	s35, s53, s33, s55	4
Backbone-backbone	phosphodiester	1

We now introduce a graph model to describe *nucleotide interaction relationships* (NIRs) within the tertiary structure. Let the query RNA sequence be $S = S_1S_2, \ldots, S_n$, where $S_i \in$ {A,C,G,U}, for $1 \le i \le n$. Then the NIRs for the sequence can be modeled with a pair $< G; A >$, where $G = (V, E)$, V = { $S_i^{(i)}$: $1 \le i \le n$}, and A is an *association* such that for every pair $i < j$, $A(i,j) \subseteq Is_is_j$ is the set of interactions between nucleotides S_i and S_j in the native structure and $A(i,j) \neq \emptyset$ implies $(i,j) \in E$. We call G the NIR graph of the sequence. Note that because of phosphodiester bonds between neighboring nucleotides, the NIR graph G always contains the Hamiltonian path $(i, i + 1)$, i=1,2,...,n-1; these edges are named *backbone edges*.

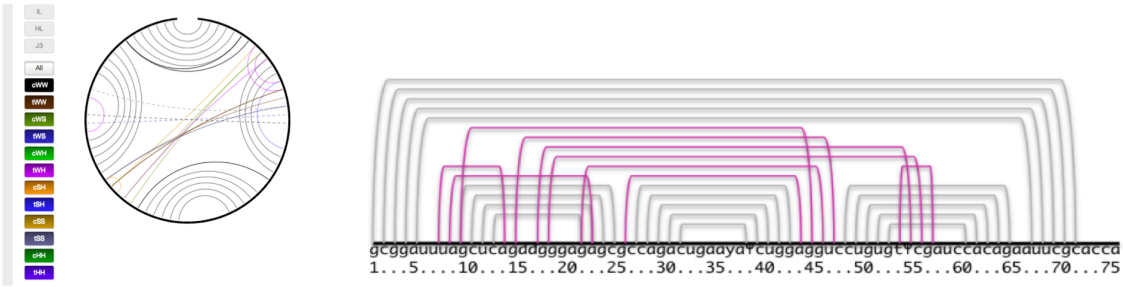


Fig. 3.1 left is a circle representation of tRNA(2DU3) *nucleotide interactions relation* graph (obtained from FR3D), each edge implies 1 or more interactions between two nucleotides. Figure right is a different NIR graph representation on 2DU3 with same interaction edges. Edges in gray color are canonical Watson-Crick base pairs and the purple edges indicate non-canonical interactions between nucleotide vertices.

We call the above graph the NIR graph of the sequence (Fig. 3.1). Note that in every NIR graph of n vertices, it contains all edges $(S_i^{(i)}, S_i^{(i+1)})$, for $1 \leq i \leq n-1$. These edges are called backbone edges.

In our recent investigation, we constructed NIR graphs for all RNAs whose tertiary structures were known from RNA Structure Atlas. We discovered that an overwhelming majority of these RNAs are of small treewidths (Fig. 3.2). If a graph has treewidth bounded by k, any clique obtained by deleting vertices and edges and contracting edges of the graph can contain at most $k + 1$ vertices. Thus the distribution of treewidths suggest that NIRs in the RNA tertiary structures are in general not arbitrarily complex.

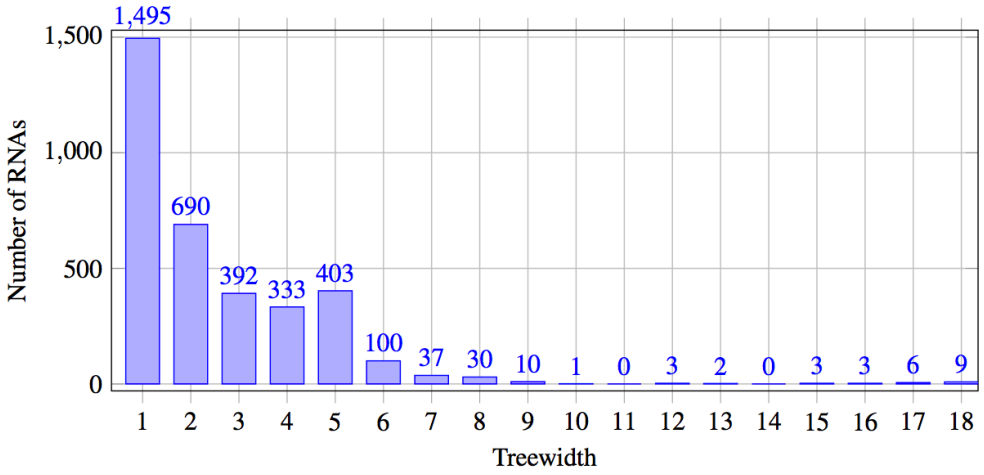


Fig. 3.2 Treewidth distribution of NIR graphs of more than 3,500 chains deriving from 1,984 resolved RNA tertiary structures in the RNA Structure Atlas. The RNAs with treewidth larger than 18 are omitted due to their very small number. These treewidths are actually upper bounds computed by a program; it is likely that the exact treewidths of the NIR graphs may actually be smaller.

3.3 Backbone *k*-Tree Model and Methods

3.3.1 Backbone *k*-Tree Model

DEFINITION 1. (Patil, 1986) Let $k \geq 1$. A k-tree is a graph that can be generated via the following rules:

1. A k-tree of k+1 vertices is a clique of k+1 vertices;

2. A k-tree of n vertices, for $n > k$+1, is a graph consisting of a k-tree G of n−1 vertices and a vertex v, which does not occur in G, such that v forms a $(k + 1)$-clique with some k-clique already in G.

Figure 3.3. shows a 3-tree with seven vertices in (a) and illustrates it in (b) with a tree-topology that connects the four 4-cliques in the graph.

Fig. 3.3 (a) 3-tree of 7 vertices by Definition 1, with the order of forming four 4-cliques: from clique {1, 2, 3, 6} (black edges), vertex 5 and blue edges added, then vertex 7 and red edges added, and finally vertex 4 and green edges added. (b) Illustration of the graph of (a) with a tree-topology connecting the four 4-cliques. (c) A backbone 3-tree for sequence AUUGGCA, of the same topology as shown in (a); backbone edges are in bold.

For any $k \geq 1$, a graph is of treewidth $\leq k$ if and only if it is a subgraph of a k-tree [63]. Therefore, NIR graphs for an overwhelming majority of known RNA tertiary structures are constrained in topology by k-trees, for small values of k. Because technically, every graph of treewidth bounded by k can be augmented with additional edges into a k-tree, we adopt such k-trees as the model for NIRs of the RNA tertiary structure. Given an NIR model <G; A>, where NIR graph G is of a treewidth $\leq$ k, one can augment G to a k-tree with additional edges, and for each newly added edge (i, j), let $A(i, j) = \emptyset$. Since such k-trees contain all backbone edges, they are called *backbone* k-*trees*.

DEFINITION 2. Let $k \geq 1$ be an integer. The backbone k-tree for an RNA sequence is an augmented NIR graph of the sequence, which is a k-tree.

Fig. 3.3.(c) shows a backbone 3-tree for sequence AUUGGCA. Note that backbone k-trees differ from general k-trees in that a backbone k-tree has to the designated Hamiltonian path (consisting of all the backbone edges).

With the backbone k-tree model, in order to predict the set A of nucleotide interactions from the query sequence, we propose to identify a backbone k-tree $G = (V, E)$ such that $(S_i^{(i)}, S_j^{(j)}) \in E$ if $\exists t \, \langle S_i^{(i)}, S_j^{(j)}, t \rangle \in A$

To ensure the identified G actually corresponds to the set of interactions that constitute the native structure of the query sequence, we need to quantify nucleotide interactions for combinatorial optimization of such a backbone k-tree G, as explained in the subsequent sections.

DEFINITION 3. Let q be a $(k + 1)$-clique in a backbone k-tree of query sequence S. An *interaction pattern* (ip) for clique q is a set $A(q)$ of nucleotide interactions, for some association A, such that $A(q) = \bigcup_{i,j \in q} A(i, j)$.

Given an ip $A(q)$ for clique q, we define the *induced subgraph by* $A(q)$, denoted with $H_{q,A(q)} = (q, E_{q,A(q)})$, is such that $(i, j) \in E_{q,A(q)}$ if $A(i, j) \neq \emptyset$.

DEFINITION 4. Let q be a *(k + 1)*-clique in the in a backbone *k-tree* of query sequence S. The *confidence* of a given ip $A(q)$ for clique q is defined as

$$f(q, A(q), S) = \sum_{(i,j) \in E_{q,A(q)}, \langle S_i, S_j, t \rangle \in A(i,j)} c_{q,H_{q,A(q)}}^{(i,j),t} \qquad (1)$$

where $c_{q,H_{q,A(q)}}^{(i,j),t}$ is the *confidence* of interaction $\langle S_i, S_j, t \rangle$ given q and subgraph $H_{q,A(q)}$ induced by ip $A(q)$.

For every clique q, with $\mathcal{P}(q)$, we denote the finite set of all ips for q. In the practical application, we may only include those ips in $\mathcal{P}(q)$, which have "high" confidences (e.g., above certain threshold).

DEFINITION 5. Let k be any fixed integer ≥ 2. The *nucleotide interaction prediction* problem NIP(k) is, given an input query sequence S, to identify a backbone k-tree $\langle G^*; A^* \rangle$, such that

$$(G^*; A^*) = \arg\max_{\langle G;A \rangle} \{ \sum_{q \ in \ G, A(q) \in \mathcal{P}(q),} f(q, A(q), S) \} \qquad (2)$$

3.3.2 ANNs

We constructed ANNs that compute confidences of nucleotide interactions, one ANN [64] for every specific nucleotide interaction $\langle S_i, S_j, t\rangle$ contained in a given specific interaction pattern $A(q)$ of a given *(k+1)*-clique q. We use $\mathcal{N}_{q,H_{q,A(q)}}^{(i,j),t}$ to denote such an ANN and $c_{q,H_{q,A(q)}}^{(i,j),t}$ for the confidence score that the ANN computes. Each ANN $\mathcal{N}_{q,H_{q,A(q)}}^{(i,j),t}$ consists of an input layer, two hidden layers (with 8 and 16 nodes, respectively), and an output layer. The output layer is a single unit producing a confidence value for interaction$\langle S_i^{(i)}, S_j^{(j)}, t\rangle$. The input layer consists of input units representing the selected global and local features. The features included the sequence length and the distance between the involved nucleotides as well as neighboring nucleotide types. In addition, we included the information of assumed canonical base pairs within the query sequence.

3.3.3 NIPDB and NIPCCTable

NIPDB is a database of all possible interaction patterns (ips) for every (k + 1)-clique, which was established by searching through the RNA Structure Atlas. To build the database, we first extracted a set $\mathcal{P}(q)$ of nucleotide interaction patterns for every (k + 1)-clique, which were found in the known 3D structures of RNAs with length ≤ 100 nucleotides. Then an unique identifier was assigned to each such clique by taking into

account both the nucleotides and their backbone distances. See Fig. 3.4 for example. Details of NIPDB construction will be addressed in next chapter.

NIPCTable is a matrix for compatibility between every pair of ips for two cliques that share all but one nucleotide vertex. To compute for the optimization problem formulated with (2), for every two (k + 1)-cliques q_1 and q_2 that are adjacent in a given k-tree, $A(q_1)$ and $A(q_1)$ are required to be compatible in the sense that the two interaction sets among the k common nucleotides of q_1 and q_2 are identical. The compatibility of all pairs of ips in NIPDB forms a binary matrix. For the efficiency, the compatibility can be precomputed before the prediction program is executed. The nucleotides in an ip are ordered from $5'$ to $3'$. Given two ips $A(q)$ and $A(p)$ each with k+1 nucleotides, we enumerate all (k+1)*k ways of mapping k nucleotides of $A(q)$ to k nucleotides of $A(p)$. For each of the mappings, if the selected two sets of k nucleotides are not identical, two ips are not compatible; otherwise, we further verify the interactions among the two sets of k nucleotides and the compatibility holds when they are identical.

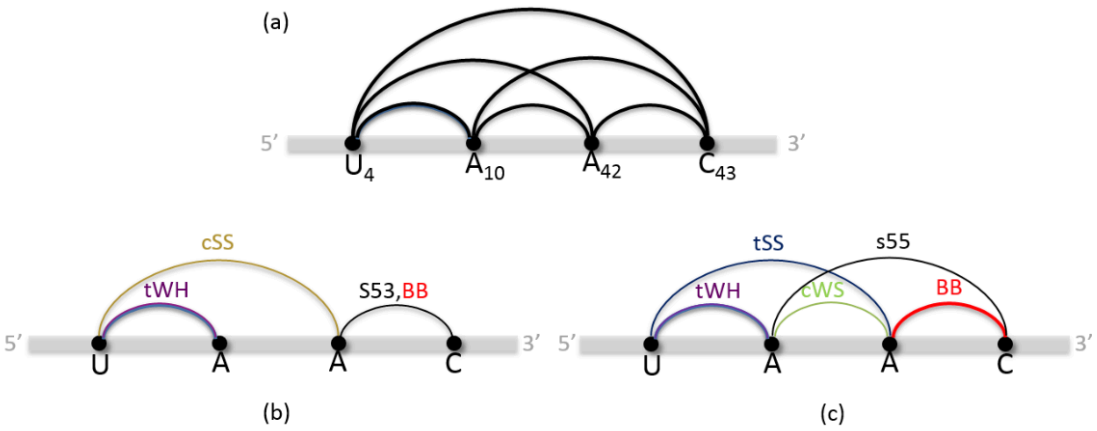


Fig. 3.4 (a) An example of 4-clique q with nucleotides U_4,A_{10},A_{42},C_{43}, where 4,10,42,43 are the indices of the nucleotides in the sequence. The identifier of q is UAAC001, where UAAC are the ordered nucleotides, 0 (resp. 1) encodes the number of nucleotides between two nucleotides along the backbone is larger than (resp. equal to) 0. Two possible interaction patterns (ips) for the clique of identifier UAAC001 in (a), with interactions labeled between the nucleotides (see Table 1), can be found by searching the identifier in the interaction patterns database NIPDB (see Section 2.3 in the main text). NIPDB has been established by extracting ips from the known RNAs of length < 100 nucleotides, e.g., one of the sources of the ip in (b) is chain D of the tRNA (PDB ID: 2DU3) with PDB numbers {908, 914, 946, 947} for the 4 nucleotides. All extracted ips are grouped according to their identifiers in NIPDB. (b) and (c) also show the subgraphs induced by two ips, respectively, where each interaction induces one edge of the subgraph and if multiple interactions induce the same edge, only one edge is shown.

3.3.4 Method and Algorithm

Our method consists of three major components to solve the NIP(k) problem: Data repositories include NIPDB and NIPCTable; A set of ANNs each computes the confidence for a specific interaction between two given nucleotides on the query

sequence; A dynamic programming algorithm that computes the solution to equation (2). Given the query sequence S and the known or predicted canonical basepairs on S, our method first employs ANNs to compute $c_{q,H_{q,A(q)}}^{(i,j),t}$. Then, it runs the dynamic programming algorithm to solve equation (2). During the dynamic programming calculation, the chosen ips need to be compatible across the cliques. This is ensured by the assertion $Q(A(q), A(p))$,which checks (1) $A(q)$ and $A(p)$ have the same set of interactions on the edges shared by cliques q and p by looking up table NIPCCTable; and (2) any pattern of interactions between a single nucleotide and multiple others has to exist in the structure database.

Our algorithm solves the NIP(k) problem by producing a pair $< G^*, A^* >$ satisfying equation (2) for the query sequence. The algorithm maximizes the confidence score of a backbone k-tree spanning over the query sequence nucleotides by a dynamic programming process. To derive recurrences for the dynamic programming, we followed the basic process of creating k-trees given in Definition 1. The inclusion of backbone edges in the k-trees disallows introducing edges in arbitrary order and thus makes the search space much smaller.

By *interval* $[i..j]$, for $i \leq j$, we mean the set of consecutive integers between i and j, inclusive. Two intervals $[i..j]$ and $[h..l]$ are *non-overlapping* if either $j \leq h$ or $l \leq i$. Let the query sequence be S of length n and q be a $(k + 1)$-clique formed by $k + 1$ vertices drawn from {1,2,...,n}. Let C be a set of non-overlapping intervals and $A(q) \in \mathcal{P}(q)$ be an ip for clique q. We define function $M(q, C, A(q), S)$ to be the maximum confidence of

a *k*-tree constructed beginning from clique q, which includes backbone edge (i, i+1) for every pair of integers i and $i + 1$ both contained in some interval in C. Then we obtain the following recurrence:

$$M(q, C, A(q), S)$$

$$= \max_{x \in q, y \notin q, y \in [i..j] \in C, p = q|_y^x} \{ \max_{A(q) \in \mathcal{P}(q), R(C_1, C_2), Q(A(p), A(q))} \{ M(p, C_1, A(p), S)$$

$$+ M(q, C_2, A(q), S) + f(q, A(q), S) \}\} \quad (3)$$

where $q|_y^x = q \cup \{y\}\{x\}$, $Q(A(p), A(q))$ asserts that the chosen ip $A(p)$ be compatible with the ip $A(q)$, and $R(C_1, C_2)$ represents the choices of two sets of intervals, C_1 and C_2.

Recurrence (3) offers a bottom-up process to compute $M(q, C, A(q), S)$. Intuitively, the idea is to create a new clique p from q by introducing a new nucleotide vertex y. There may be one or more sub-*k*-trees, some stemming from p while others from q (but not including vertex y). Since these sub-*k*-trees will never join together again, interval sets are used to ensure backbone edges will be properly created. In particular, the set of backbone edges in the *k*-tree corresponding to the value of function $M(q, C, A(q), S)$ contains only those edges between consecutive indexes specified in the intervals in C. Initially, C may include intervals allowing all backbone edges. To complete the recurrence (3), we need the following base case:

$$M(q, C, A(q), S) = 0 \text{ if } C = \emptyset$$

which will be first computed in a bottom-up dynamic programming calculation.

CHAPTER 4

BACKBONE *k*-TREE BASED 3D MODELING

In this chapter we will discuss the backbone *k*-tree based 3D modeling (BkTree3D) method, the derived algorithm and performance. Some geometric definitions will be introduced as well to facilitate discussion of the BkTree3D method. BkTree3D is a three-dimensional modeling method that outputs the predicted RNA 3D structure at high-resolution. Here we first summarize the underlying techniques of the method.

The method BkTree3D is to be pipelined with BkTree, the nucleotide interaction prediction method. BkTree3D takes as input the predicted nucleotide interactions along with the predicted backbone k-tree that has organized the nucleotides into (k + 1)-cliques. BkTree3D assigns one geometric motif candidate to the set of predicted nucleotide interactions within every clique. The candidates were first extracted from RNA Structure Atlas [33] with the set of interactions considered. Then the geometry of each candidate was collected from PDB [65]. Since there are usually more than one motif candidate for every clique, motifs are selected to achieve the highest consistency across all cliques in the k-tree. The consistency between two motifs selected for two respective "neighboring" cliques is measured with the root-mean-square-deviation (RMSD) on the two motif geometries involving the k common nucleotides shared by the two cliques. The predicted 3D structure consists of a collection of motifs selected for all the cliques, which has the lowest sum of the RMSDs across all pairs of "neighboring" cliques in a *k*-tree. The input

k-tree enables a dynamic programming algorithm to compute the RMSD sum very efficiently in both time and space, in particular, in time $O(C^2N)$, where C is the number motif candidates considered for each clique, and N is the number of cliques in the k-tree, linearly proportional to the length n of the query RNA sequence. In addition, the method assumes an option to use Amber [66] for structure refinement.

4.1 Geometry Candidate and Database

Recall that in chapter 3 an interaction pattern for a clique q is defined as a set $A(q)$ of nucleotide interactions, for some association A, such that $A(q) = \cup_{i,j \in q} A(i,j)$, where $A(i,j)$ associates edge (i,j) with a set of simultaneous interactions between nucleotides i and j. Technically, interaction patterns are used for each associations. An interaction pattern contains a *key* and a set of interactions assigned to it. The purpose of defining such identifier is to break down the number of candidates that one key may have. See Fig. 4.1 for details.

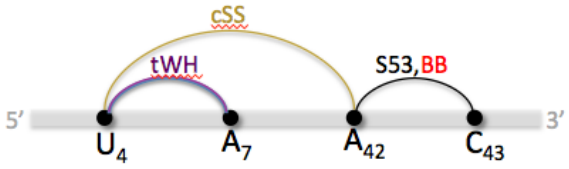


Fig. 4.1 An example of 4-clique q with nucleotides U4, A7, A42, C43, where 4, 7, 42, 43 are the indices of the nucleotides in the sequence. The A(q) in this case contains key

UAAC 301 1101 and interaction {1-2 tWH, 1-3 cSS, 3-4 S53}. UAAC are of the ordered nucleotides' number, 0 (reps. 1, 2, 3) in single underline encodes the number of nucleotides between two neighboring nucleotides along the backbone is larger than 3 (resp. equal to 1, 2, 3). Numbers 0 and 1 in wave underline encodes if a nucleotide has a canonical cWW interaction with nucleotide outside of the clique q, 1 for paired, 0 otherwise.

We use notation g_{X} to denote a *geometry candidate* for X, where X is an interaction association of a set of cliques. That is,

$$g_{\mathrm{X}} = \bigcup_{A(q)\in \mathrm{X}} g_{A(q)}$$

where, for every clique q, the geometry candidate $g_{A(q)}$ is a collection of geometric coordinates for every atom in the nucleotides in q. $g_{A(q)}$ satisfies the $A(q)$ key and interaction constrains, obtained from a native structures. We use $g_{A(q)}^{\{x\}}$ to denote the geometry candidate for nucleotide *x* in q. $g_{A(q)}^{s}$ is the geometry candidate for a subset *s* of q, such that

$$g_{A(q)}^{s} = \bigcup_{x\in s} g_{A(q)}^{\{x\}}$$

Furthermore, with $\mathcal{G}(\mathrm{X})$, we denote a finite set of geometry candidates for X.

We have constructed a geometry database (GDB) contains all possible geometry candidates from the training data. To build the database, we first enumerated a set of nucleotide interaction patterns for every (*k*+1)-clique, which were found in the known 3D structures of RNAs in the PDB databank. Then for every pattern, we extracted a set of geometry candidates from these resolved PDB structures. 747 PDB entries with

redundancy were processed to construct the GDB. As a result, GDB consists 354,831 interaction patterns and over 30 million geometry candidates in total. For each interaction pattern, we only stored index references from the coordinate file in pdb format to lighten the resource request from the database. For example, reference [2QUS_A 20 21 30 31] includes PDB id, Chain id and k+1 residue indexes. Then the detailed coordinates' information can be retrieved from PDB file in further 3D modeling calculation. Furthermore, we divided the database into sub categories including *cww*, *dense*, *others* and *recover* to facilitate the query process. See Fig. 4.2 for details of GDB.

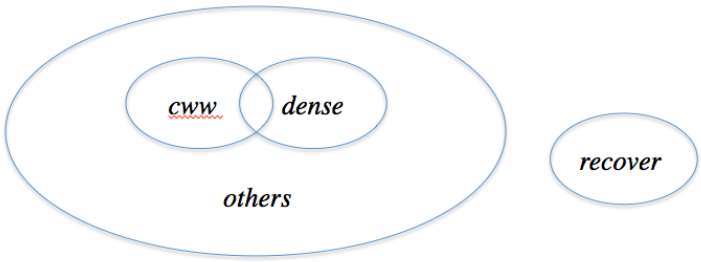


Fig. 4.2 An illustration of geometry databases relationships. cww database only consists of patterns that contain at least one canonical cWW interaction. We defined dense pattern to be such that for every S_i, S_j in q, there exists an interaction type t that <S_i, S_j, t> in A(q). The dense database consists of only dense interaction patterns. The rest of interaction patterns are categorized into the others database. We also constructed the recover database by extracting interaction patterns contain only 4 contiguous nucleotides and neighboring "s35" interaction(s) from the whole database.

Considering the number of geometry candidates for every interaction pattern is usually large and the geometrical redundancy between these candidates, we introduced an

additional clustering step to group the geometry candidates. In the clustering, we applied an all-to-all geometry alignment to all candidates for a given $A(q)$. The 0.5Å RMSD cutoff value was adopted in the alignment, which means the RMSD value is less than 0.5Å for every two geometry candidates in each group after clustering. Then we selected one geometry candidate to represent each group. The detailed explanation of such alignment calculation will be given in next section. Because this clustering step is a time and resource consuming work, thus we only applied it to the *cww* and *dense* databases, which contain interaction patterns that are most commonly used by the B*k*Tree3D modeling program. It allows us to reduce half the size of the original *cww* database and even more in the *dense* database.

4.2 Geometric Alignment

Let q be a (k+1)-clique, we defined *aggregated interaction* patterns of q as formed by merging interaction patterns from two or more cliques that are connected in the k-tree. A geometry candidate for an aggregated interaction pattern is formed by alignment of geometry candidates, which is to achieve optimal geometrical alignment measured by the RMSD value, from each merged interaction pattern. Before we discuss how the geometry candidates were selected, we need to introduce the RMSD measurement and some geometric manipulation concepts to derive our desired alignment function.

4.2.1 RMSD Measurement

The root-mean-square deviation (RMSD) is the measure of the average distance between the atoms (usually the backbone atoms) of two superimposed proteins or RNAs. In the study of globular RNA conformations, one customarily measures the similarity in three-dimensional structure by the RMSD of all atomic coordinates after optimal rigid body superposition. RMSD of C4' atomic coordinates of RNA sugar or RMSD of atoms in RNA backbone are also widely used in many studies. Typically RMSD is used to make a quantitative comparison between the structure of a partially folded protein and the structure of the native state. For example, the CASP protein structure prediction competition uses RMSD as one of its assessments of how well a submitted structure matches the native state.

In particular, RMSD is measured with:

$$RMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \delta_i^2}$$

where δ is the distance between N pairs of equivalent atoms.

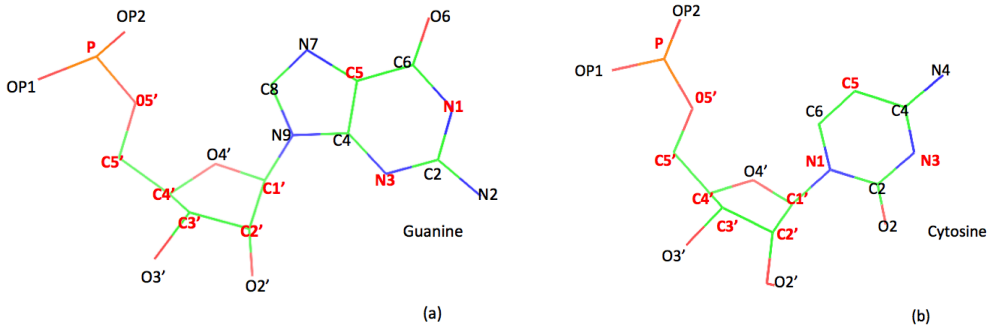
Normally a rigid superposition that minimizes the RMSD is performed, and this minimum is returned. Given two sets of n points of v and w, the RMSD is defined as follows:

$$RMSD(v,w) = \sqrt{\frac{1}{N}\sum_{i=1}^{N} \|v_i - w_i\|^2}$$

$$= \sqrt{\frac{1}{n}\sum_{i=1}^{n} \left((v_{ix} - w_{ix})^2 + \left(v_{iy} - w_{iy}\right)^2 + (v_{iz} - w_{iz})^2\right)} \qquad (1)$$

An RMSD value is expressed in length units. We adopt the most common used unit in structural biology, Ångström(Å), which is equal to 10^{-10}m.

In this work, we considered a set of atomic coordinates including P, O5', C5', C4', C3', C2', C1', C5, N3 and N1 to calculate the RMSD (see Fig 4.3.). They cover critical atoms in backbone, sugar and base in terms of conformation, and can save computation time without loosing accuracy for our optimization purpose.



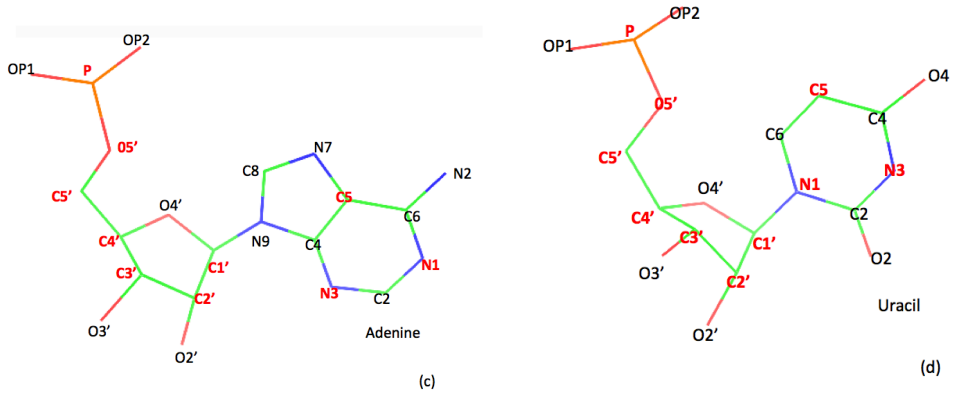


Fig. 4.3 Structural atoms of nucleic acid constituents. They contain common phosphate group and sugar, and differ in bases of G, C, A and U. Atoms in red color are selected to calculate the RMSD value.

4.2.2 Alignment Function

The geometrical alignment between two (k+1)-clique p and q is accomplished by a superimposition of common nucleotides' positions in geometric candidate $g_{A(p)}$ from $\mathcal{G}(A(p))$ onto the corresponding nucleotides' positions in geometric candidate $g_{A(q)}$ from $\mathcal{G}(A(q))$, achieving the minimum RMSD. In general, the structure comparison (and alignment) is an NP-hard problem. The alignment function used in this work is imported from Bio.PDB package, particularly, from the *Superimposer* object. It calculates the rotation and translation matrix that superposes two equal sized sets of atoms on top of each other in such a way that their RMSD is minimized. *Superimposer* follows the algorithm in book [67] that is based on singular value decomposition (SVD) [67].

Here we briefly introduce the method used by *Superimposer*. First, it calculates the centers of two sets of 3D coordinates P' and Q' on centroids $c1$ and $c2$. To place P' and Q' into a same coordinate system, it calculates two coordinate sets P and Q,

$$P = P' - c1$$

$$Q = Q' - c2$$

Every coordinate in P and Q is essentially a vector from centroid to its original coordinate in P' and Q'.

Then, it calculates a correlation matrix M,

$$M = P^T Q$$

where

$$M_{ij} = \sum_{k=1}^{N} P_{ik}\, Q_{kj},$$

The optimal rotation, *Rot*, can be calculated using the following SVD routine,

$$M = UDV^T$$

where U and V^T are unitary matrices, and D is an diagonal matrix with non-negative real numbers on the diagonal.

Next, it decides whether it is necessary to correct the rotation matrix to ensure a right-handed coordinate system

$$d = sign\big(det\,(VU^T)\big)$$

Finally, the optimal rotation matrix *Rot* can be calculated as

$$Rot = V\begin{pmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & d\end{pmatrix}U^T$$

And the corresponding transformation matrix *Tran*,

$$Tran = c2 - c1Rot$$

To get the transformed coordinates,

$$Q_{tran^p} = QRot + Tran \tag{2}$$

where Q_{tran^p} denotes Q transformed to P. In this way, Q will be rotated and translated onto P. And the alignment score of RMSD, is calculated based on $RMSD(Q_{tran^p}, P)$ from equation(2).

Therefore, we define the geometry alignment function γ between geometry candidates as

$$\gamma\left(g_{A(p)},\ g_{A(q)}\right) = RMSD(g^S_{A(q)_{tran^{g_{A(p)}}}},\ g^S_{A(p)}) \tag{3}$$

where $s = p \cap q$.

The alignment score *rmsd* between two neighboring clique p and q in the k-tree, under the association A, is defined as,

$$rmsd(A(p), A(q)) = \min_{g_{A(p)} \in \mathcal{G}(A(p)),\ g_{A(q)} \in \mathcal{G}(A(q))} \{\gamma(g_{A(p)}, g_{A(q)})\} \quad (4)$$

We use notation $c(q)$ to define a set of cliques formed by children cliques of q, and $c(q)_i$ to be i-th child of q, given some order of the children. The *rmsd* between q and $c(q)$, under the association A, is defined as,

$$rmsd(A(q), A(c(q)))$$

$$= \min_{g_{A(q)} \in \mathcal{G}(A(q)), g_{A(c(q)_i)} \in \mathcal{G}(A(c(q)_i))} \sum_{i=1,\ g_{A(c(q)_i)} \in \mathcal{G}(A(c(q)_i))}^{n} \gamma(g_{A(q)}, g_{A(c(q)_i)}) \quad (5)$$

where n is the size of $c(q)$.

We define the aggregated geometry candidate $g_{A(p)} \circ g_{A(q)}$ or $g_{A(p \cup q)}$ to be such that $g_{A(q)}$ is aligned onto $g_{A(p)}$ based on equation (3), and $g_{A(q)}^{\{y\}|y \in p \setminus q}$ is recomputed based on its rotation and transformation matrix (see equation (2)),

$$g_{A(p)} \circ g_{A(q)} = g_{A(p)} \cup g_{A(q)}^{\{y\}|y \in p \setminus q}{}_{tran^{g_{A(p)}}} \quad (6)$$

Furthermore, we define a set of aggregated geometry candidates as $\mathcal{G}(g_{A(p)} \circ g_{A(q)})$ or $\mathcal{G}(A(p \cup q))$.

4.3 Optimization problem

DEFINITION 4.1 Let q be a (k+1)-clique and $c(q)$ be the set of children (k+1)-cliques of q in the k-tree. The alignment score of a given geometry candidate $g_{A(q)}$ for clique q and the geometric candidate for $g_{A(c(q))}$ is defined as

$$f\left(q, A(q), g_{A(q)}, c(q), g_{A(c(q))}\right) = \sum_{i=1}^{n} \gamma(g_{A(q)}, g_{A(c(q)_i)}) \qquad (a)$$

where n is the size of $c(q)$.

We formally define the RNA 3D prediction problem as follow:

DEFINITION 4.2 Let k ≥ 2 be any fixed integer. The *RNA 3D structure prediction problem* is, given backbone k-tree, called T, a set of nucleotide interactions $A(T)$, and the geometry candidate $\mathcal{G}(A(T))$ to identify a structure $\mathcal{S}$ such that

$$\mathcal{S} = \underset{g_{A(q)}, g_{A(c(q)_i)} \in \mathcal{G}(A(T))}{arg\ min} \left\{ \sum_{q \in T} f\left(q, A(q), g_{A(q)}, c(q), g_{A(c(q))}\right) \right\} \qquad (b)$$

We define an objective function to optimize as follows. Let q be a $(k+1)$-clique from which a sub backbone k-tree T_q rooted at q, has been predicted. We define objective function $M\left(q, A(q), \mathcal{G}\left(A(q)\right)\right)$ to be the minimum sum of all root-mean-square-deviations between every pair of children $(k+1)$-cliques in T_q . This has the following recurrence to conform to the general dynamic programing scheme,

$$M\left(q, A(q), \mathcal{G}\left(A(q)\right)\right)$$

$$= \min_{g_{A(q)} \in \mathcal{G}\left(A(q)\right)} \sum_{i=1,\ c(q)_i \in c(q),\ g_{A(c(q)_i)} \in \mathcal{G}\left(A(c(q)_i)\right)}^{n} \left\{\gamma\left(g_{A(q)}, g_{A(c(q)_i)}\right) + M\left(c(q)_i, A(c(q)_i), \mathcal{G}\left(A(c(q)_i)\right)\right)\right\} \qquad (c)$$

where n is the size of $c(q)$. The base case for the recurrence is $M\left(q, A(q), \mathcal{G}\left(A(q)\right)\right) = 0$ if $\mathrm{c(q)} = \emptyset$ which will be first calculated in the dynamic programming.

Recurrence (c) provides a bottom-up process to compute $\mathrm{M}\left(\mathrm{q}, \mathrm{A(q)}, \mathcal{G}\left(\mathrm{A(q)}\right)\right)$. The idea is to choose one geometry candidate from each clique that can be best superposed together. The alignment score of the produced 3D structure is computed as the sum of alignment score of geometry candidates chosen for all involved (k+1)-cliques. Geometric compatible between neighboring cliques is evaluated by alignment function γ. Thus, geometry candidates don't possess any alignment score alone until the candidates from the children cliques align onto it. Such alignment score is accumulated from the bottom to the root of T, and the desired structure can be retrieved following the trace-back process of dynamic programming.

4.4 Algorithms

4.4.1 General Algorithm for RNA 3D Structure Prediction

Our algorithm for RNA 3D structure prediction solves the problem by outputting a 3D structure that satisfies equation (b) in section 4.3 for the input topology k-tree T, together with the corresponding interaction association $A(T)$. In this section, we will address more detailed steps of the optimization computation including: parsing tree and geometry candidate assignment; merging geometries through dynamic programming; tracing back geometries and assembling the predicted structure.

At the first step, it assigns a finite set of geometry candidates to each $A(q)$ from T and A by querying the GDBs, in the order of *dense*, *cww* and *other*. Besides, we established a mechanism to relax restrictions of $A(q)$ if a query $A(q)$ was not covered in the GDBs by modifying the *key* and interactions of $A(q)$. A clique q will be removed from the k-tree T if the query again returns null after the relaxation.

Then the algorithm minimizes the alignment score of a 3D structure over the T and A by a dynamic programming process. Recurrence (c) leads to a bottom-up process to compute $M\left(q, A(q), \mathcal{G}\left(A(q)\right)\right)$ with the base case $M = 0$ for a leaf clique q, which will be the first calculated. In particular, the alignment function γ computes the RMSD between every neighboring (k+1)-cliques on the k common nucleotide vertexes that they share. For every clique q in k-tree T, a dynamic programming table is constructed for the sub k-tree rooted at q. The table records all functional values, including mainly the geometry candidate indexes in GDB, alignment scores and trace-back information, each computed from optimal functional values for the sub k-trees rooted at the children cliques of q.

Once the optimization is done, the alignment score calculated for the root clique covers the value of the optimal solution. The optimal values of geometry candidates can be recovered, from root down, by tracing back the calculations already performed from the dynamic programming tables. To fold the final structure, it aligns the geometry candidates chosen for each clique from root down based on equation (6). For every pair of "neighboring" cliques, geometry coordinates from the parent clique are kept, and the coordinates for introduced nucleotide vertex are included to the overall structure after the alignment of these two cliques is done.

4.4.2 Structure Minimization

Considering that the *backbone* coordinates of final predicted RNA structure may loose certain accuracy during the assembling, we refined the all–atom structure by using sander energy minimization from AMBER suite as a final step.

Sander is an Amber module that carries out energy minimization, molecular dynamics, and NMR refinements. The acronym stands for Simulated Annealing with NMR-Derived Energy Restraints, but this module has also been used for a variety of simulations that have nothing to do with NMR refinement.

Sander provides direct support for several force fields for nucleic acids, and for several water models and other organic solvents. The basic force field adopted here has the

following form, which is about the simplest functional form that preserves the essential nature of molecules in condensed phases:

$$
\begin{aligned}
V(\mathbf{r}) \quad = \quad & \sum_{bonds} K_b(b-b_0)^2 + \sum_{angles} K_\theta(\theta-\theta_o)^2 \\
& + \sum_{dihedrals} (V_n/2)(1+\cos[n\phi-\delta] \\
& + \sum_{nonb\,ij} (A_{ij}/r_{ij}^{12}) - (B_{ij}/r_{ij}^{6}) + (q_i q_j/r_{ij})
\end{aligned}
$$

In this work, we refined our predicted all-atom RNA structure by using sander energy minimization pipelined the tleap generated topology file and restart file that required by *sander*. The force field files were adopted from http://fch.upol.cz/en/rna_chi_ol/. One thousand steps of minimization were run without restraints. The sample minimization input file is described in below:

Constant Volume Minimization
 # Control section
&cntrl
ntwe = 500, ntwx = 500, ntpr = 500,
nsnb = 25, dielc = 80, cut = 12.0,
ntb = 0,
maxcyc = 1000, ntmin = 1, ncyc = 1000, dx0 = 0.01, drms = 0.0001,
ntp = 0,
ibelly = 0, ntr = 0,
imin = 1,
/

Here we take 2QUS.pdb as an example for the *tleap* script we used to generate restart and topology files. More details will be provided in the *readme* file of the B*k*Tree3D package.

```
#tleap script for 2qus_ predict_structure.pdb
source /path/to/forcefield_file/leaprc.ff14SBsc
loadamberprep /path/to/prep_file/ff99bsc0chiOL/all_nuc94_ol_bsc0.in
loadamberparams  /path/to/params_file/ff99bsc0chiOL/frcmod.ol.dat
mol=loadpdb 2QUS_predict_structure.pdb
saveamberparm mol 2QUS_rst.prmtop 2QUS_rst.prmcrd
savepdb mol 2QUS_rst.pdb
quit
```

4.5 Pre-processing and other issues

4.5.1 Pre-processing

The accuracy of our proposed BkTree3D method for solving the RNA 3D structure prediction problem highly relies on the inputted interaction association *A*. Based on table 4.1, the BkTree3D can produce accurate 3D structures in general by assigning the real nucleotides interactions to the *k*-tree *T*. However, the BkTree predicted interactions might hardly reach such accuracy and a relatively high false positive rate is also a stumbling block during the 3D structure construction. Furthermore, stacking interactions with different facing conformations were trained as a single stacking interaction in ANN machine learning step. As a result, BkTree provided stacking facing conformation that might not always lead to the correct prediction. Therefore, we have introduced a preprocessing step that establishes a serial of knowledge-based rules to modify the predicted interactions before using them in the 3D model step.

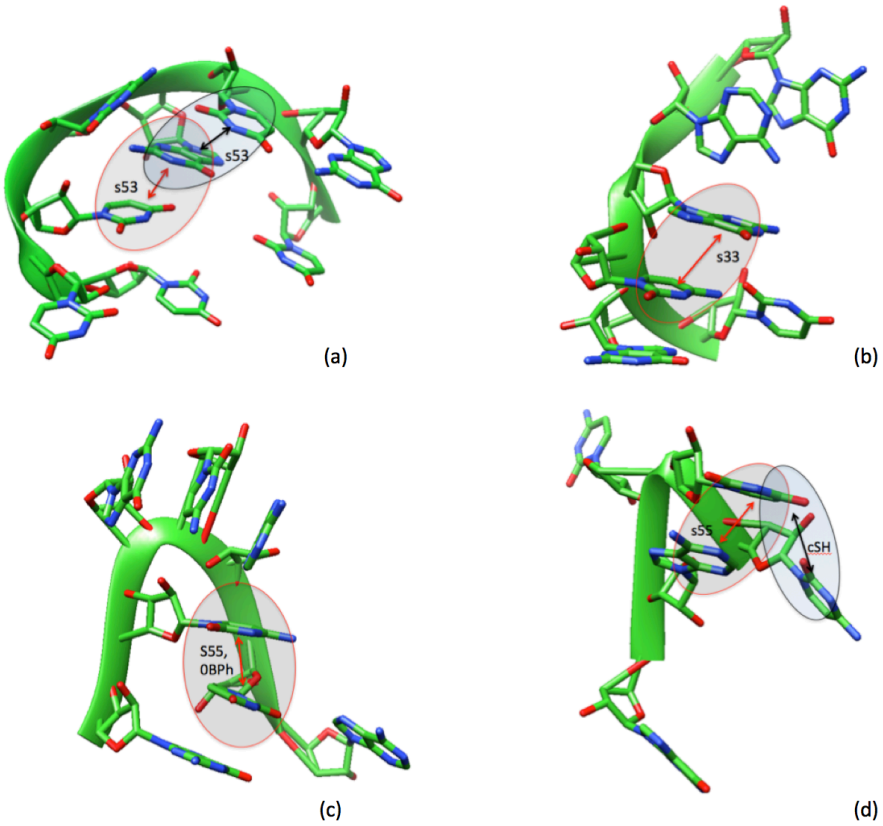


Fig. 4.4 Examples of unnatural shapes of RNA backbones. Stacking interactions other than s35 between neighboring nucleotides will force the backbone twist in a way that its torsion angles are far off their average values.

In Fig. 4.4, we give examples of uncommon backbone interactions that cause unnatural shapes in the modeled backbone 3D structures. The pre-processing step solves the problem by removing or replacing certain interactions and results in a more nature backbone spin. More specifically, rules including removing contiguous s53; removing s55 or s33 within 3 intervals; removing contiguous interactions in *cis* or *trans* conformation and replacing contiguous s33 with s35. The proposed rules help to improve the result of 3D prediction generally (see result in section 4.7).

4.5.2 Missing nucleotides recovery

After taking pre-processing into account, there is a possibility that certain nucleotide vertices might be missed from the predicted structure if the preprocessed $A(q)$ along with its modified interaction patterns were all not found in GDB in the step of parsing T and A. See Fig. 4.5 for an example. To solve such a problem, we basically created an $A(q)$ that covers the missing nucleotide and its neighboring ones and obtained the geometry candidates by querying the *recover* GDB. Then similar to geometry alignment and assembly, the $g_{A(q)}$ that achieves the minimal RMSD value with the predicted structure will be chosen to recover the missing nucleotide based on equation (5) and (6).

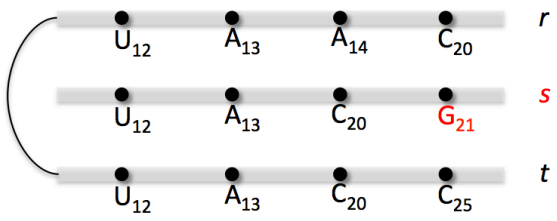


Fig. 4.5 Example of missing nucleotide during parsing k-tree. Assuming r, s and t are 4-cliques in a tree T; r and s, s and t are neighboring cliques; A(s) and modified A(s) are not found in GDBs. We will ignore the clique s by connecting clique r and t and reassign the parent-child relationship during the k-tree parsing. Hence nucleotide vertex G_{21} in color red will be missed in the final predicted structure. Note that, not every ignored clique could cause a missing vertice. It is still a valid k-tree if we change clique t to $\{G_{11}, U_{12}, A_{13}$ and $G_{21}\}$, but no vertex will be missed in this case.

4.5.3 Improved version

We have also implemented an improved version that can return suboptimal 3D structures. More specifically, we modified the equation (4) and recurrence (c) to $(4)^*$ and $(c)^*$ to return multiple geometry candidates.

$$rmsd(p,q) = \min^{m}_{g_{A(p)} \in \mathcal{G}(A(p)),\, g_{A(q)} \in \mathcal{G}(A(q))} \{\gamma(g_{A(p)}, g_{A(q)})\} \qquad (4)^*$$

$$M\left(q, A(q), \mathcal{G}(A(q))\right)$$

$$= \min^{m}_{g_{A(q)} \in \mathcal{G}(A(q))} \sum_{i=1,\ c(q)_i \in c(q),\ g_{A(c(q)_i)} \in \mathcal{G}(A(c(q)_i))}^{n} \Big\{\gamma(g_{A(q)}, g_{A(c(q)_i)})$$

$$+ M\left(c(q)_i, A(c(q)_i), \mathcal{G}(A(c(q)_i))\right)\Big\} \qquad (c)^*$$

where m is the number of structures.

The dynamic programming table is modified accordingly to store these additional values. One thing to note is that we made copies of the last geometry candidate returned from $M\left(q, A(q), \mathcal{G}(A(q))\right)$ until $\mathcal{G}(A(q))$ fulfills the size of m, if the size of $\mathcal{G}(A(q))$ is less than m for a clique q. Following properties enables this improved version: first, our 3D modeling method is computationally efficient in both time and space; second, the preserved monotone property within the alignment score calculation along the k-tree topology. The results will be presented in next section.

4.6 Performance Evaluation

4.6.1 Test Data

We tested our BkTree and BkTree3D modeling method on a list of 43 RNAs that had been used as a benchmark set [1]. This RNA data set consists of 43 high resolution (3.4 Å or better) structures varies in sizes and motifs. The lengths are from 16 to 128 nucleotides. The structural topologies range from hairpins to riboswitches, pseudoknots and junctions.

4.6.2 Overall Performance

For each of the all 43 RNAs, the general 3D modeling program produced an optimally predicted 3D structure whose RMSD was calculated against the resolved structure. To emphasis the significant of our model, all the RMSD results in our evaluation were calculated without AMBER minimization. The *deformation index* (DI), a measure that accounts for both RMSD and MCC, the quotient between them, was also calculated (see Table 4.1). Fig. 4.7 plots the RMSD values for all the 43 RNAs. These data suggest a great potential of our method for RNAs beyond short lengths. In particular, 12 out of the 18 RNAs with lengths exceeding 50 nucleotides achieved RMSD values below 10 Å; Fig. 4.6 shows some of such examples. On the other hand, the sharp increase of RMSD values for some of longer RNAs in Fig. 4.6 by no means indicates the failure of the backbone *k*-tree model or the proposed 3D modeling method on these RNAs. Most of them appear to require a backbone 4-tree model, beyond the capability of the 3-tree based BkTree3D program (see more in chapter 5).

To verify the rationality of our 3D modeling program, we performed two tests. First, Table 4.2 presents the RMSD values of 42 RNA structures (see note in Table 4.1) predicted from BkTree3D by taking the real interaction association and BkTree predicted tree topology $< \mathrm{A}^{\mathrm{real}}, \mathrm{T} >$. The result shows the majority has RMSD values around 2~4 Å; 35 out of 42 RNAs achieved RMSD values below 5 Å. From Fig. 4.8, which plots these RMSDs, we also noticed the unbalanced increase of two longer RNAs. Similar as general 3D modeling program, tree-width of 3 is one of the limitations. Besides, topology T is not constructed based on $\mathrm{A}^{\mathrm{real}}$, so additional interactions might be missed from $< \mathrm{A}^{\mathrm{real}}, T >$ as well. In the second test, for each of all 42 RNAs, we removed all geometry candidates came from RNA we are predicting from the GDBs. When we calculating RNA 2QUS for example, no geometry candidate with source PDB ID of 2QUS will be considered to construct the final structure. Table 4.3 lists the RMSD values for such test. From Fig. 4.9, which plots RMSD values from Table 4.3, we can see that no noticeable RMSD increase compared to Fig. 4.10 has been occurred.

Table 4.4 presents the RMSD values of 42 BkTree3D predicted RNA structures with interaction "preprocessing" applied. From Fig. 4.10, we can see the improvement after "preprocessing". The average RMSD value dropped from 6.9 Å to 4.8 Å. Table 4.5 contains the results of "improved" version. For each of all 42 RNAs, "improved" BkTree3D produced 5 structures including the optimal and sub-optimal ones. The best and average RMSD values are also presented.

Table 4.1. RMSDs and DIs of the predicted 3D structures against the native structures for the 43 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with 3D modeling from nucleotide interactions predicted by BkTree. Note that 2FK6 had been updated in PDB databank after the survey paper was published. We will remove 2FK6 from the test set in our further evaluations to maintain the consistency.

PDB ID	Len.	STY	MCC	EdgeDiff	RMSD	DI	Topology
2F8K	15	0.85	0.85	12/12	2.911	3.397	Hairpin
361D	19	0.70	0.63	12/12	2.525	3.976	Hairpin
2AB4	20	1	0.95	10/10	2.753	2.887	Hairpin
2ANN	23	0.75	0.70	12/12	4.351	6.153	Hairpin
2QUX	25	0.90	0.80	15/15	3.162	3.924	Hairpin
1RLG	25	0.95	0.77	15/15	2.630	3.374	Hairpin, internal loop
387D	26	0.86	0.77	10/10	6.069	7.837	Hairpin
1MSY	27	0.97	0.95	23/23	1.304	1.373	Hairpin
1L2X	28	0.88	0.88	23/22	3.534	4.006	Pseudoknot
2AP5	28	0.82	0.74	23/21	6.231	8.390	Pseudoknot
1JID	29	0.93	0.82	20/20	2.346	2.849	Hairpin, internal loop
3SNP	29	0.93	0.89	15/15	1.198	1.341	Hairpin, internal loop
430D	29	0.94	0.85	21/21	4.335	5.054	Hairpin, internal loop
1OOA	29	0.93	0.82	22/22	4.165	5.053	Hairpin, internal loop
2OZB	33	0.93	0.86	26/26	3.497	4.047	Hairpin, internal loop
1MJI	34	0.84	0.84	29/28	1.758	2.090	Hairpin, internal loop
1ET4	35	0.67	0.75	28/25	4.565	6.049	Pseudoknot
2HW8	36	0.93	0.86	30/30	6.673	7.710	Hairpin, internal loop
1I6U	37	0.91	0.90	28/28	3.095	3.419	Hairpin, internal loop
1ZHO	38	0.95	0.89	29/29	3.264	3.663	Hairpin, internal loop
1F1T	38	0.81	0.71	31/31	4.258	5.927	Hairpin, internal loop
1XJR	47	0.88	0.84	37/37	2.494	3.036	Hairpin, internal loop
1S03	47	0.83	0.82	35/35	3.485	4.146	Hairpin, internal loop
2PXB	49	0.98	0.96	36/36	1.140	1.184	Hairpin, internal loop
1U63	49	0.94	0.78	38/38	2.087	2.665	Hairpin, internal loop
2FK6	53	0.77	0.73	43/33	11.01	14.907	Pseudoknot, 3-way junction
3E5C	54	0.84	0.78	40/39	2.042	2.593	3-way (riboswitch)
1MZP	55	0.64	0.68	44/38	11.23	16.337	Hairpin, internal
1DK1	57	1	0.94	46/45	1.029	1.091	3-way
1MMS	58	0.74	0.78	57/45	2.032	2.600	3-way
3EGZ	65	0.70	0.68	39/35	6.433	9.393	3-way (riboswitch)
2QUS	69	0.75	0.75	48/42	9.864	13.017	Pseudoknot, 3-way junction
1KXK	70	0.96	0.94	54/54	3.230	3.422	Hairpin, internal loop
2DU3	71	0.78	0.74	50/43	3.751	5.047	4-way junction (tRNA)
2OIU	71	0.90	0.86	51/51	5.629	6.475	3-way junction (riboswitch)
1SJ4	73	0.78	0.79	53/38	13.99	17.539	Pseudoknot, 4-way junction
1P5O	77	0.97	0.87	51/51	6.826	7.832	Hairpin, internal loop
3D2G	77	0.80	0.84	70/60	4.552	5.396	3-way junction (riboswitch)
2HOJ	79	0.87	0.85	64/58	4.024	4.694	3-way junction (riboswitch)

2GDI	80	0.84	0.81	63/56	5.176	6.314	3-way junction (riboswitch)
2GIS	95	0.87	0.848	79/67	14.87	17.527	Pseudoknot,4-way
1LNG	97	0.85	0.82	76/70	10.86	13.159	3-way junction (SRP)
1MFQ	128	0.81	0.78	101/92	24.58	31.140	3-way junction (SRP)

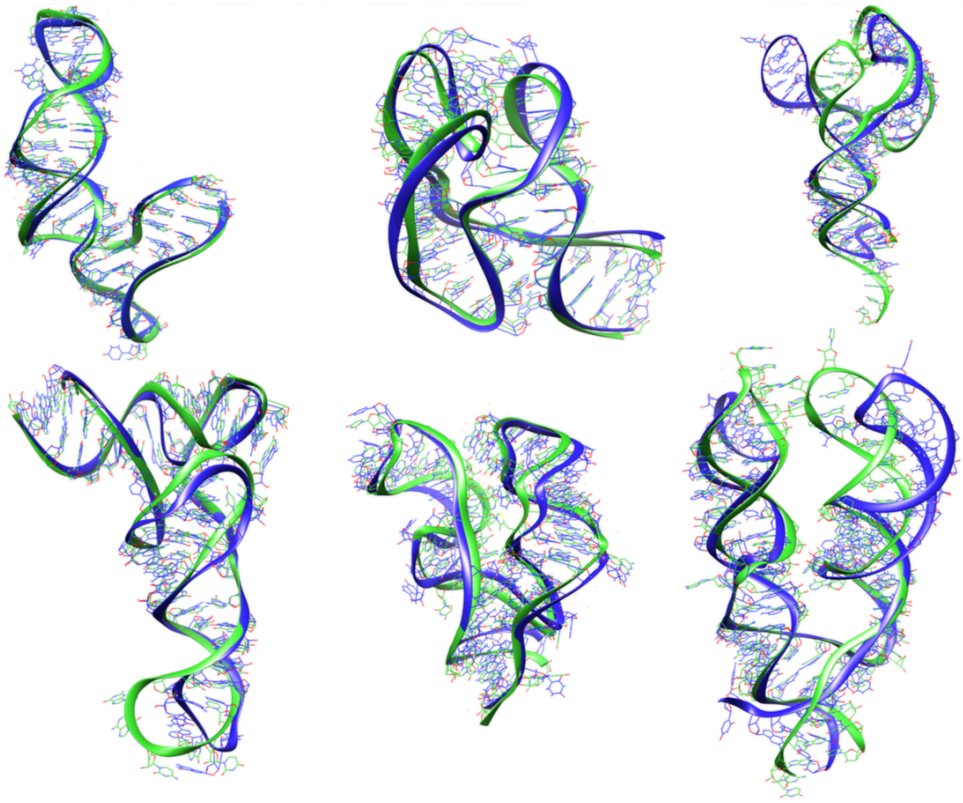


Fig. 4.6 Examples of predicted 3D models (blue) superimposed with respective native structures (green). The superimpositions in the frist row from left to right: 1DK1 (57 nucleotides, 3-way junction, 1.029˚A); 1MMS (58 nucleotides, 3-way junction, 2.032˚A); 2QUS (69 nucleotides, pseudoknot, 9.864 ˚A). The superimpositions in the second row from left to right: 2DU3 (71 nucleotides, 4-way junction, tRNA, 3.751 ˚A); 3D2G (77 nucleotides, 3-way junction, riboswitch, 4.552˚A), 1LNG (97 nucleotides, 3-way junction, SRP, 10.862˚A).

Table 4. 2. RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with BkTree3D modeling from resolved nucleotide interactions.

PDB ID	Len.	RMSD	PDB	Len.	RMSD	PDB	Len.	RMSD
2F8K	15	2.02	2OZB	33	1.365	1MMS	58	4.768
361D	19	3.99	1MJI	34	1.105	3EGZ	65	4.893
2AB4	20	3.01	1ET4	35	3.816	2QUS	69	3.001
2ANN	23	3.04	2HW8	36	1.574	1KXK	70	2.666
2QUX	25	1.05	1I6U	37	1.569	2DU3	71	0.558
1RLG	25	4.61	1ZHO	38	2.07	2OIU	71	1.21
387D	26	0.06	1F1T	38	2.501	1SJ4	73	7.333
1MSY	27	1.00	1XJR	47	1.355	1P5O	77	11.47
1L2X	28	1.45	1S03	47	1.512	3D2G	77	2.461
2AP5	28	2.85	2PXB	49	0.884	2HOJ	79	8.283
1JID	29	1.77	1U63	49	1.421	2GDI	80	2.227
1OOA	29	0.159	3E5C	54	7.598	2GIS	95	5.713
3SNP	29	2.132	1MZP	55	4.862	1LNG	97	19.91
430D	29	2.26	1DK1	57	4.944	1MFQ	128	20.79

Table 4. 3. RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with 3D modeling from "preprocessed" nucleotide interactions predicted by BkTree. Each of all RNAs was not included in the geometrical training data when being calculated.

PDB ID	Len.	RMSD	PDB	Len.	RMSD	PDB	Len.	RMSD
2F8K	15	2.694	2OZB	33	2.347	1MMS	58	2.973
361D	19	5.825	1MJI	34	4.173	3EGZ	65	10.33
2AB4	20	2.571	1ET4	35	5.214	2QUS	69	11.13
2ANN	23	2.649	2HW8	36	4.542	1KXK	70	3.261
2QUX	25	2.156	1I6U	37	3.051	2DU3	71	2.993
1RLG	25	4.021	1ZHO	38	4.328	2OIU	71	5.9
387D	26	6.002	1F1T	38	8.174	1SJ4	73	11.18
1MSY	27	2.337	1XJR	47	2.88	1P5O	77	5.166
1L2X	28	3.750	1S03	47	2.907	3D2G	77	7.162
2AP5	28	2.711	2PXB	49	2.506	2HOJ	79	3.583
1JID	29	4.180	1U63	49	5.293	2GDI	80	3.22
1OOA	29	2.262	3E5C	54	1.716	2GIS	95	14.34
3SNP	29	2.263	1MZP	55	5.833	1LNG	97	10
430D	29	4.282	1DK1	57	1.686	1MFQ	128	21.33

Table 4. 4. RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with 3D modeling from "preprocessed" nucleotide interactions predicted by BkTree.

PDB ID	Len.	RMSD	PDB	Len.	RMSD	PDB	Len.	RMSD
2F8K	15	2.674	2OZB	33	2.143	1MMS	58	2.864
361D	19	5.809	1MJI	34	4.173	3EGZ	65	10.33
2AB4	20	2.787	1ET4	35	5.214	2QUS	69	11.01
2ANN	23	1.642	2HW8	36	4.255	1KXK	70	3.261
2QUX	25	2.156	1I6U	37	3.051	2DU3	71	2.993
1RLG	25	2.169	1ZHO	38	4.094	2OIU	71	5.9
387D	26	6.002	1F1T	38	8.015	1SJ4	73	11.18
1MSY	27	2.49	1XJR	47	2.88	1P5O	77	5.166
1L2X	28	2.433	1S03	47	2.707	3D2G	77	7.162
2AP5	28	1.997	2PXB	49	2.506	2HOJ	79	2.564
1JID	29	3.249	1U63	49	4.991	2GDI	80	3.22
1OOA	29	2.262	3E5C	54	1.716	2GIS	95	14.13
3SNP	29	1.602	1MZP	55	5.648	1LNG	97	7.959
430D	29	4.165	1DK1	57	1.686	1MFQ	128	19.3

Table 4. 5. RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with *improved* BkTree3D modeling from nucleotide interactions predicted by BkTree with *preprocessing*. Each of RMSD1~5 represents one predicted structure. The average RMSD values are also calculated.

PDB ID	Len.	RMSD1	RMSD2	RMSD3	RMSD4	RMSD5	BEST	AVERAGE
2F8K	15	2.674	2.644	2.644	3.544	3.544	2.644	3.004
361D	19	5.809	4.950	4.975	4.939	5.005	4.939	5.139
2AB4	20	2.787	2.472	2.466	2.543	2.431	2.431	2.497
2ANN	23	1.642	2.474	2.526	2.367	2.224	2.224	2.448
2QUX	25	2.156	3.938	3.984	3.874	4.205	2.156	3.632
1RLG	25	2.169	4.245	3.854	3.996	4.514	3.854	4.126
387D	26	6.002	6.442	6.762	8.708	7.986	6.002	7.180
1MSY	27	2.49	2.133	2.064	2.100	2.087	2.064	2.144
1L2X	28	2.433	3.956	4.817	4.842	4.614	3.750	4.396
2AP5	28	1.997	2.711	2.711	2.711	2.711	2.711	2.711
1JID	29	3.249	2.029	1.828	2.044	2.039	1.828	2.024
1OOA	29	2.262	4.244	4.258	4.256	4.055	4.055	4.219
3SNP	29	1.602	3.277	3.098	5.184	3.285	2.263	3.422
430D	29	4.165	1.789	2.739	2.739	2.750	1.602	2.324
2OZB	33	2.143	3.889	4.013	3.174	3.987	3.037	3.620

1MJI	34	4.173	4.196	4.243	3.939	4.060	3.939	4.193
1ET4	35	5.214	5.764	5.138	5.264	5.207	5.138	5.362
2HW8	36	4.255	3.742	3.667	3.836	3.635	3.579	3.692
1I6U	37	3.051	3.432	3.414	3.486	3.273	3.273	3.448
1ZHO	38	4.094	3.912	3.622	4.039	3.934	3.622	3.912
1F1T	38	8.015	4.925	5.903	5.944	4.429	4.429	5.257
1XJR	47	2.88	2.670	2.562	2.542	2.401	2.401	2.545
1S03	47	2.707	3.113	3.035	2.626	3.197	2.267	2.848
2PXB	49	2.506	3.186	3.554	5.375	2.900	2.571	3.517
1U63	49	4.991	4.979	5.181	6.136	6.059	4.979	5.468
3E5C	54	1.716	2.041	3.253	2.021	3.024	1.785	2.425
1MZP	55	5.648	7.685	7.976	8.633	9.966	5.731	7.998
1DK1	57	1.686	1.994	1.982	2.228	3.092	1.685	2.196
1MMS	58	2.864	3.701	4.439	4.240	4.025	3.483	3.977
3EGZ	65	10.33	11.95	12.46	11.32	11.20	10.33	11.45
2QUS	69	11.01	9.857	15.39	17.66	21.89	15.39	14.83
1KXK	70	3.261	5.089	6.140	5.032	6.574	4.114	5.390
2DU3	71	2.993	3.303	2.703	3.187	3.494	2.703	3.136
2OIU	71	5.9	6.832	10.24	6.127	8.771	10.24	7.582
1SJ4	73	11.18	11.44	10.08	10.23	11.58	10.08	11.02
1P5O	77	5.166	5.477	6.657	5.978	5.694	5.466	5.854
3D2G	77	7.162	7.939	7.523	10.32	9.456	10.32	8.588
2HOJ	79	2.564	2.750	3.709	3.162	3.461	2.566	3.130
2GDI	80	3.22	3.149	4.142	3.189	4.025	3.112	3.523
2GIS	95	14.13	15.32	17.45	15.80	16.51	14.13	15.84
1LNG	97	7.959	7.804	8.330	7.735	8.801	7.735	8.177
1MFQ	128	18.3	24.42	24.92	23.37	24.19	22.37	23.85

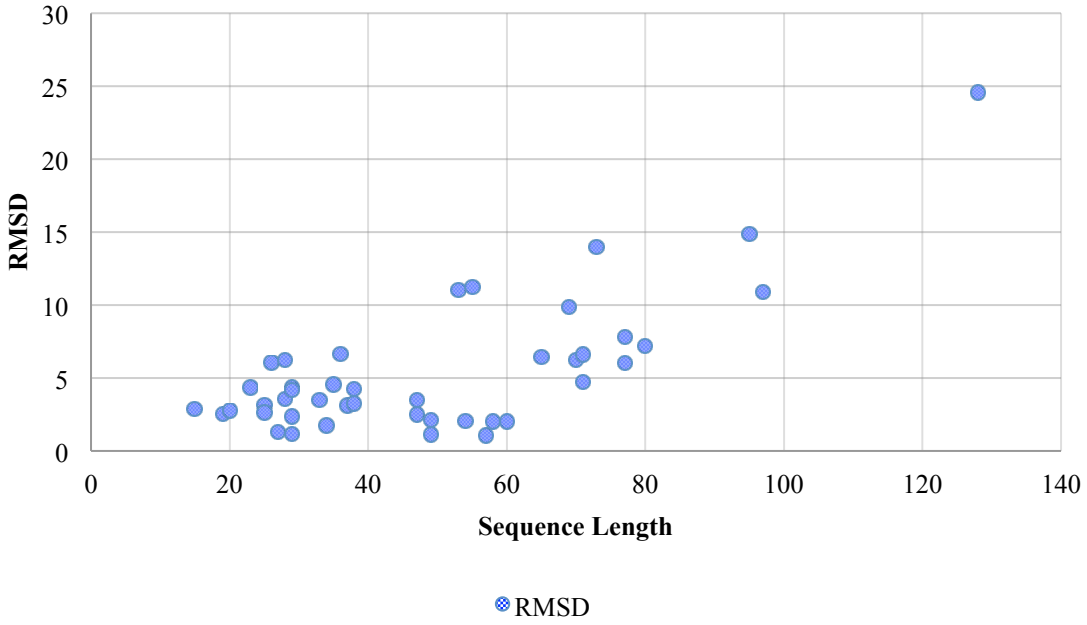


Fig. 4.7 RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with 3D modeling from nucleotide interactions predicted by BkTree.

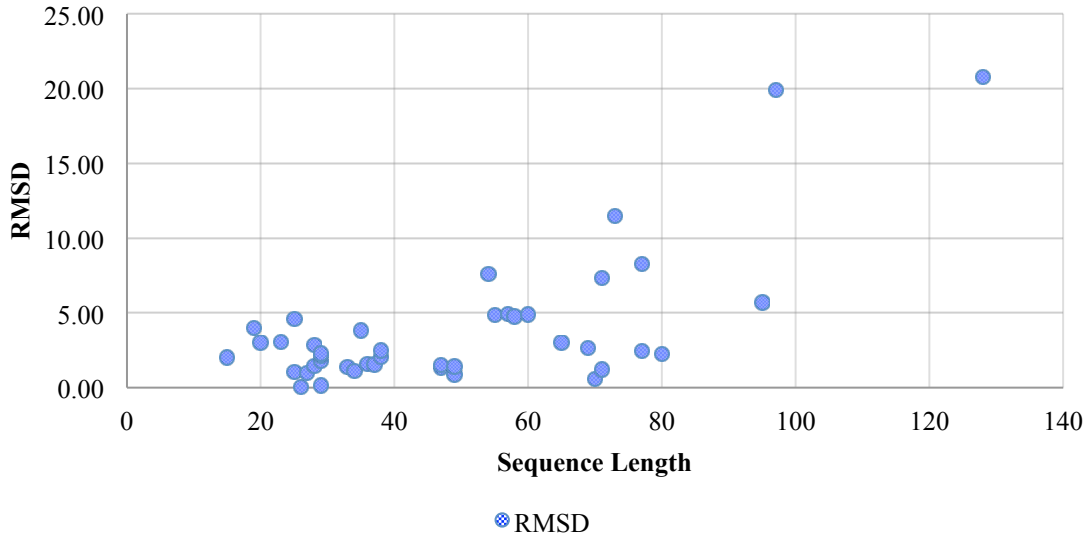


Fig. 4.8 RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The 3D structure prediction was with BkTree3D modeling from resolved nucleotide interactions.

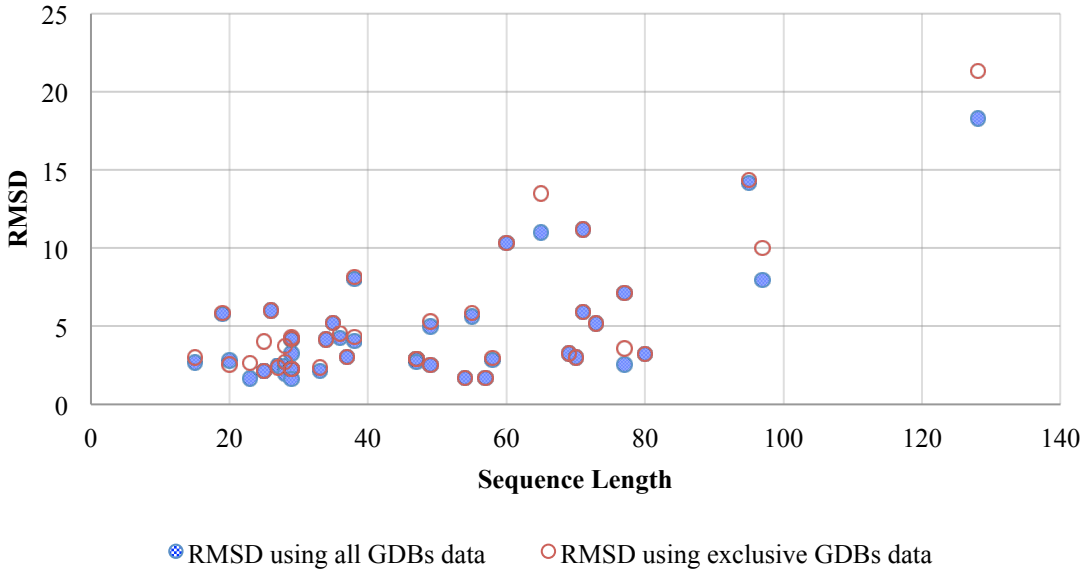


Fig. 4.9 Comparison of RMSDs generated by BkTree3D using all GDBs data and exclusive GDBs data, sorted by the RNA sequence lengths. Exclusive GDBs is constructed by removing geometric candidates sourced from testing set (42 RNAs). The plot was generated by merging the RMSD results from Table 4.3 and 4.4.

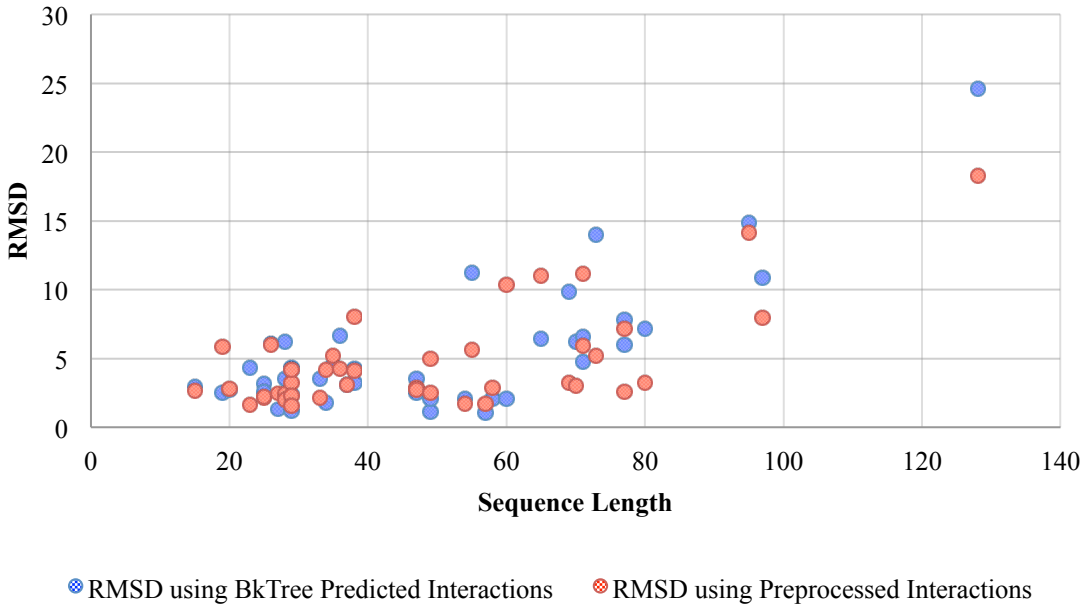


Fig. 4.10 Comparison of RMSDs generated by BkTree3D using BkTree Predicted interactions and preprocessed interactions, sorted by the RNA sequence lengths. The plot was derived by merging the RMSD results from Table 4.1 and 4.4.

4.6.3 Performance Comparison with Other Methods

To compare with other 3D structure prediction methods, Table 4.6 presents the performance values on the 4 representative RNAs chosen in [1] which typically contain two hairpins and two junctions. Since both MC and Rosetta allow prediction of multiple optimal or suboptimal folds, we chose to use the best and the averaged values of their solutions. For every one of the 4 RNAs, the RMSD achieved by BkTree3D is significantly smaller than both the best and the averaged values achieved by MC and Rosetta.

The more recently developed 3D structure prediction program RNA-MoIP (Reinharz et al., 2013) shows some remarkable improvements of the geometries with an integer programming method to fit tertiary motifs into the predicted secondary structure. In Table 4.7, BkTree3D is compared with RNA-MoIP [27] on all the 9 RNAs tested by RNA-MoIP, all of lengths exceeding 50 nucleotides. This is not an ideal comparison as we have input to BkTree program the known canonical Watson-Crick base pairs instead of the predicted secondary structure. It shows that for 5 out of the 9 RNAs, our method achieved an RMSD value even smaller than the minimum RMSD achieved by RNA-MoIP on each of the 5 RNAs. On the 6th RNA 2HOJ, our RMSD is 4.024 ˚A, close to the minimum RMSD 3.19 ˚A but much smaller than the average 7.19 ˚A achieved by RNA-MoIP. On two slightly longer RNA sequences 1LNG and 1MFQ, our predictions yielded RMSDs larger than what RNA-MoIP produced. We will further discuss the performance issue in next chapter.

Table 4. 6. List of performance values predicted using MC, Rosetta and BkTree on 4 representative RNAs chosen by (Laing and Schlick, 2010). The results generated by MC and Rosetta are obtained from the survey paper (Laing, 2014; Laing and Schlick, 2010). We display the results of each RNA in two categories, where the average and the best performances of up to five folds are shown in the upper and lower category respectively. The highest values among the results of three methods are displayed in bold.

		MC				Rosetta				BkTree			
	PDB	1KXK	1XJR	2OIU	2QUS	1KXK	1XJR	2OIU	2QUS	1KXK	1XJR	2OIU	2QUS
	NT	70	47	71	69	70	47	71	69	70	47	71	69
Avg	STY	81	76	76	78	74	71	63	58	97	91	92	80
	PPV	89	87	92	86	85	83	87	86	94	84	86	80
	MCC	0.849	0.813	0.836	**0.819**	0.793	0.767	0.740	0.706	**0.958**	**0.878**	**0.892**	0.8
	RMSD	9.49	8.74	16.85	18.41	17.23	11.63	18.10	15.73	**5.46**	**2.71**	**5.02**	**9.07**
	DI	11.16	10.74	20.14	22.44	21.69	15.21	24.72	22.80	**5.69**	**3.08**	**5.62**	**11.33**
Best	STY	91	90	93	87	88	87	91	90	97	91	92	80
	PPV	84	82	80	79	76	78	75	73	94	84	86	80
	MCC	0.874	0.859	0.862	**0.829**	0.817	0.794	0.826	0.810	**0.958**	**0.878**	**0.892**	0.8
	RMSD	8.71	5.72	14.02	15.92	12.65	8.88	15.98	12.07	**5.46**	**2.71**	**5.02**	**9.07**
	DI	10.03	7.08	16.78	19.19	15.53	10.97	21.58	15.03	**5.69**	**3.08**	**5.62**	**11.33**

Table 4. 7. RMSD comparison between BkTree and RNA-MoIP. Boldface indicates the smaller RMSD value of each RNA. "-" indicates the RNA that RNA-MoIP failed on.

	Length	RNA-MoIP			BkTree
		Min	Avg	SD	
3E5C	53	-	-	-	**2.042**
1DK1	57	2.95	4.76	0.99	**1.029**
1MMS	58	5.66	7.65	0.86	**2.032**
2DU3	71	2.23	**2.91**	0.44	3.751
3D2G	77	5.34	7.35	1.34	**4.552**
2HOJ	79	3.19	7.19	2.31	**4.024**
2GDI	80	-	-	-	**5.176**
1LNG	97	2.73	**6.30**	1.91	10.862
1MFQ	128	9.07	**14.34**	5.01	24.588

CHAPTER 5

3D MODELING WITH INCOMPLETE INFORMATION

The evaluation results have highlighted the performance of our method as a feasible one that can produce accurate 3D structure prediction of RNA sequences beyond short lengths. However, the tests also have revealed certain limitations of our current *k*-tree model, especially, the choices of parameter *k* for the *k*-tree may significantly affect the prediction results on 3D modeling performance. Moreover, the number of allowed geometry candidates is a factor as well, particularly due to the existence of "dangling vertices" in interaction patterns. Therefore, this chapter will discuss how to cope with the limits of current backbone 3-tree 3D modeling method.

5.1 Dangling Vertices And Carry On Method

Assumes that p and c are two neighboring cliques and $A(p)$ and $A(c)$ are interaction patterns for p and c respectively. We define the geometry alignment between $g_{A(p)}$ and $\mathcal{G}(A(c))$ to be:

$$rmsd\left(g_{A(p)}, \mathcal{G}(A(c))\right) = \min_{g_{A(c)} \in \mathcal{G}(A(c))} \{\gamma(g_{A(p)}, g_{A(c)})\} \qquad (7)$$

where γ is the geometry alignment function adopted from equation (3) in chapter 4.

DEFINITION 5.1 Let q be a (k+1)-clique in a backbone k-tree of $< A,\ T >$, where A is the interaction association and T is the k-tree topology. An interaction pattern $A(q)$ is called a *dangling* interaction pattern if there is a vertex i in q such that for every j in q, for every interaction type $t, < i, j, t > \notin A(q)$. And vertex i is called a *dangling* vertex of clique q.

In other words, dangling vertex is an independent vertex that has no interaction to other ones for the given interaction pattern $A(q)$.

Let q and c be two neighboring (k+1)-cliques from a k-tree T, q is the parent of c, we define vertex i is "carriable" between c and q, if i is in both c and q and is a *dangling* vertex in q but not in c; and "*non-carriable*" if i is dangling in both q and c, or i doesn't exist in c. For example, in Fig. 5.1(a), vertex C_{20} is *carriable* between cliques q and c because it is dangling in clique q but has interaction with G_{21} in clique c. Fig. 5.1(b) and (c) give examples of non-*carriable* dangling vertices between neighboring cliques.

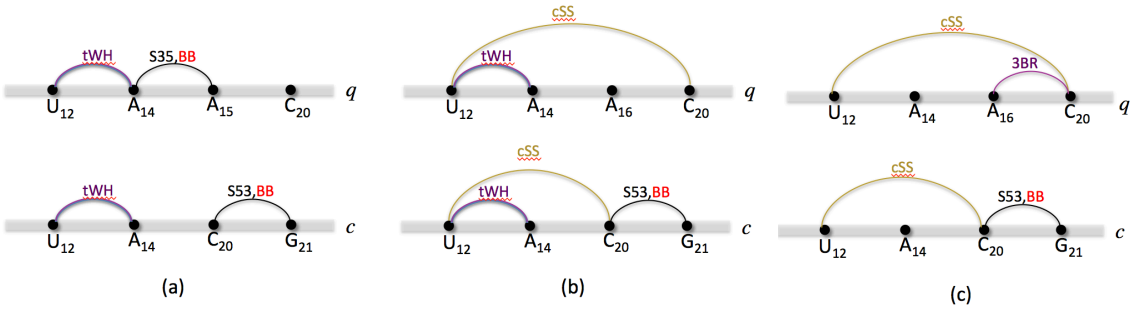


Fig. 5.1 Examples of dangling vertices and patterns: (a) a carriable dangling vertex C_{20} between clique c and q. (b) a non-carriable dangling vertex A_{16} between clique c and q. (c) a non-carriable dangling vertex A_{14} between clique c and q.

Generally, given two nucleotides and a type of interaction between them, the conformation space of these nucleotides are usually very limited. On the other hand, the conformations could be relatively arbitrary without the interaction constraint. The dangling vertex in a clique is the later case. There are nearly 80% interaction patterns from our database which have at least one dangling vertex. Although we have established the *key* to help filter out the geometry patterns, dangling vertices may still jeopardize the accuracy of predicted structures because of their geometrical uncertainties.

To solve the issue caused by *dangling* vertices, first we define the *geometry update* method as following, assuming p and c are two neighboring (k+1)-cliques from a k-tree T, c is a child clique of p. If nucleotide vertex x is a carriable dangling vertex between c and p, the geometry alignment between $g_{A(p)}$ and $g_{A(c)}$ would be calculated by function $\gamma\left(g_{A(p)}^{p\backslash\{x\}}, g_{A(c)}^{c\backslash\{x\}}\right)$. The updated geometry candidate for $g_{A(p)}$ would be calculated based on $g_{A(p)}^{p\backslash\{x\}} \circ g_{A(c)}$ defined by equation (6) in chapter 4. Then for every $g_{A(p)}$ in $\mathcal{G}(A(p))$, we apply the *geometry update* between $g_{A(p)}$ and $g_{A(c)}$, where $g_{A(c)}$ is calculated by the alignment function $\gamma\left(g_{A(p)}^{p\backslash\{x\}}, \mathcal{G}_{A(c)}^{c\backslash\{x\}}\right)$, to construct an updated $g_{A(p)}$.

The rest of the algorithm is similar to the BkTree3D method except that, the geometry alignment between cliques p and $c(p)_i$ would be calculated before p and $c(p)_j$ in the

$rmsd(p, c(p))$ (equation (5) in chapter 4) calculation, if x is a carriable dangling vertex between $c(p)_i$ and p; x is not carriable between $c(p)_j$ and p; and $i \neq j$.

The evaluation on carry-on method will be addressed in later section.

5.2 5-Clique Approximation With 4-Cliques

Recall that, the evaluation tests from section 4.5 have revealed that some RNAs are of more complex structures and their nucleotide interaction relationships are actually beyond the capability of the backbone 3-tree model. In particular, column EdgeDiff of Table 5.1 shows that all the seven RNAs with higher RMSD values (also see Table 4.1) have more than a few edges in their NIR graphs which cannot be included by even the best backbone 3-tree model. For example, the 3-tree model can miss 6 and 9 edges in the NIR graphs of RNAs 1LNG and 1MFQ, respectively. These edges correspond to some important nucleotide interactions including those between the hairpins of two helices in these two signal recognition particle RNAs. Failure to predict these crossing interactions has resulted in the considerably high RMSD values in their BkTree3D predicted structures. Figure 5.2(a) shows the modeled 3D structure of 1LNG deviates from its native structure due to those missed interactions across the two hairpins. More specifically, failure to capture the crossing interactions for pseudoknots structure results in a more loosely compacted 3D structure compares to the resolved one. From Fig. 5.2(b), we can see that the kissing-hairpin arms of our predicted 3D structure of 2QUS are distant instead of folding toward each other.

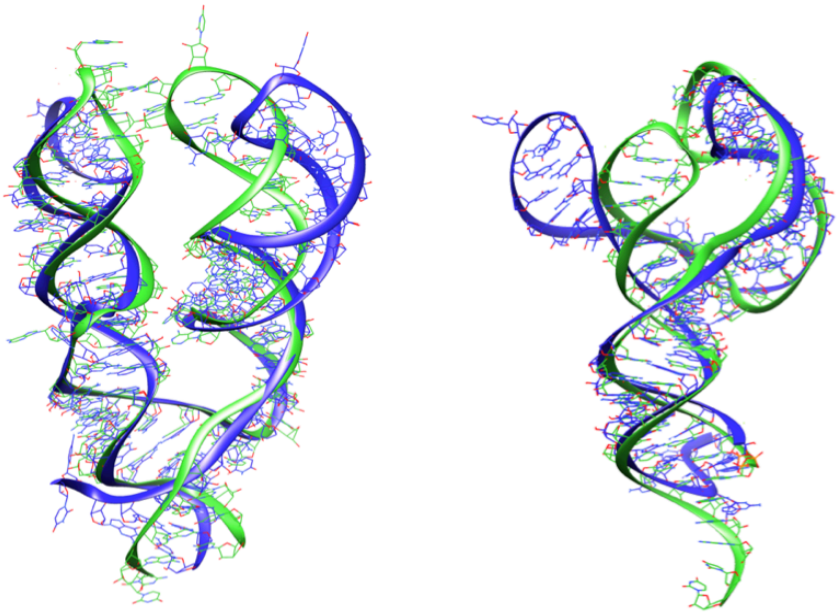


Fig. 5.2 Examples of predicted 3D models (blue) superimposed with respective native structures (green). The superimpositions (a): 1LNG (97 nucleotides, 3-way junction, SRP, 10.862˚A); (b) 2QUS (69 nucleotides, pseudoknot, 9.864˚A).

We propose a method of approximating the geometries for 5-clique to improve the performance limited by the 3-tree model. Let q be a 5-clique, we use $sub(q)$ to denote a set of 4-cliques which are subsets of q. To construct an approximated geometry for a 5-clique q, we first rank all 4-cliques from $sub(q)$ based on their numbers and types of interactions, then select the top three cliques r, s, and t to perform the approximation calculation.

DEFINITION 5.2 Let p be a 4-clique, the *ranking value* of a given interaction pattern $A(p)$ for the a 4-clique p is defined as

$$RV(p) = \left|\bigcup_{i,j \in p} A(i,j)\right|^2 + \left|\bigcup_{i,j \in p, i<j, i+1 \neq j} A(i,j)\right|$$

where the symbol "| |" denotes "the size of".

In this way, the 4-cliques are first ranked by their numbers of interactions; then sorted by the number of interactions between non-neighboring nucleotides since the neighboring ones have the backbone constraints already.

DEFINITION 5.3 Let q be a 5-clique and r, s, t in sub(q) be 4-cliques that have the best RV(r, s, t) values. Then, given geometry candidates set $\mathcal{G}(A(r)), \mathcal{G}(A(s)), \mathcal{G}(A(t))$, the set of approximated geometry candidates for $A(q)$ is defined as

$$\mathcal{G}(A(q))^* = \bigcup_{g_1 \in \mathcal{G}(A(r)),\, g_2 \in \mathcal{G}(A(s)),\, g_3 \in \mathcal{G}(A(t))} (g_1 \circ g_2) \circ g_3 \qquad (8)$$

where the operation $\circ$ is defined in equation (6) in chapter 4.

To implement the $\mathcal{G}(A(q))^*$, we first calculate and sort all $rmsd(g_{A(r)}, g_{A(s)})$s for every $g_{A(r)}$ in $\mathcal{G}(A(r))$ and $g_{A(s)}$ in $\mathcal{G}(A(s))$, and construct a $\mathcal{G}(A(r \cup s))$ from those have top c RMSD values based on equation (6) in chapter 4, where c is a parameter as the number of geometry candidates built for $\mathcal{G}(A(r \cup s))$. Then, following the same method, we can calculate all $rmsd(g_{A(r \cup s)}, g_{A(t)})$s and construct a $\mathcal{G}(A(r \cup s \cup t))$, which is $\mathcal{G}(A(q))^*$, contains c geometry candidates. Note that, we applied $rmsd()$ calculation twice instead of calculating $rmsd(g_{A(r)}, g_{A(s)}, g_{A(t)})$ directly to achieve the efficiency.

Let S be a RNA sequence, <A, T> be a backbone k-tree for S, S_i and S_j are two nucleotide vertices in S, and t be an interaction type. Assuming <S_i, S_j, t> is not presented in the interaction association A, we define the *path* between S_i and S_j to be the shortest path of cliques that can connect q_{S_i} (a clique containing S_i) and q_{S_j} (a clique containing S_j) over the k-tree T.

The 5-clique approximation method is general for adding the <S_i, S_j, t> to the k-tree T with any type of interactions. However, without further information, the method is limited to the canonical cWW interactions that are missed in the earlier interaction prediction stage. Therefore, to apply the geometry approximation method to a given backbone k-tree $< A,\ T >$, we first identify the missing <S_i, S_j , cWW> by comparing differences between the predicted interaction association and inputted canonical *cww* one. A *path* connecting q_{S_i} and q_{S_j} can be calculated based on the missing <S_i, S_j, cWW>. Then we add the vertex S_i to every clique on the *path* if S_i doesn't present in the clique already and construct the $\mathcal{G}(A(q))^*$ based on equation (8). The rest of the algorithm is similar to the general BkTree3D method.

Note that, to avoid the creation of 6-cliques, we can only add more than one interactions to the k-tree <A, T> if their *path*s are not overlapping in the tree topology.

5.3 Performance Evaluations

To test the carry-on method, we adopted the same data set used in Chapter 4. For each of the all 42 RNAs, the carry-on 3D modeling program produced an optimally predicted 3D

structure whose RMSD was calculated against the resolved structure. All the RMSD results in our evaluation were calculated without AMBER minimization. Table 5.1 presents the RMSD values of 42 RNA structures predicted by the carry-on 3D modeling method. The averaged increased RMSDs for 12 RNAs is 0.85Å and the decreased (improved) RMSDs for 11 RNAs is 1.9Å. Fig 5.3 plots the improvement made by the carry-on method. Fig 5.4 shows some of such examples.

To verify the 5-clique approximation method, we have only tested it on 2QUS (69 nucleotides, pseudoknot). See Fig. 5.5 for details. This method is not as practical as others because of the approximation step requires a more larger candidates pool and can be very time consuming.

Table 5. 1. RMSDs of the predicted 3D structures against the native structures for the 42 RNAs, sorted by the RNA sequence lengths. The number of dangling patterns (Dangling) and carrible dangling patterns (Carrible) were listed. Column 5 shows the edge difference for each of the RNAs. RMSD1 shows previous RMSD values without applying the carry-on method. RMSD2 shows RMSD values calculated by the carry-on method. The improved values were colored in green, otherwise in red. The averaged increased RMSDs is 0.85 Å for 12 entries (red) and the decreased (improved) RMSDs is 1.9 Å for 11 entries (green).

PDB ID	Len.	Dangling	Carrible	EdgeDiff	RMSD1	RMSD2	Topology
2F8K	15	0	0	12/12	2.674	2.674	Hairpin
361D	19	2	1	12/12	5.809	4.721	Hairpin
2AB4	20	0	0	10/10	2.787	2.787	Hairpin
2ANN	23	0	0	12/12	1.842	1.842	Hairpin
2QUX	25	0	0	15/15	2.156	2.156	Hairpin
1RLG	25	1	0	15/15	2.169	2.383	Hairpin, internal loop
387D	26	2	2	10/10	6.002	4.806	Hairpin
1MSY	27	1	1	23/23	2.49	2.194	Hairpin

1L2X	28	4	4	23/22	2.433	2.967	Pseudoknot
2AP5	28	5	5	23/21	1.997	2.57	Pseudoknot
1JID	29	1	1	20/20	3.249	2.049	Hairpin, internal loop
3SNP	29	0	0	15/15	1.802	1.802	Hairpin, internal loop
430D	29	0	0	21/21	4.165	4.165	Hairpin, internal loop
1OOA	29	1	0	22/22	2.262	2.262	Hairpin, internal loop
2OZB	33	0	0	26/26	2.343	2.343	Hairpin, internal loop
1MJI	34	3	2	29/28	4.173	4.349	Hairpin, internal loop
1ET4	35	5	3	28/25	5.214	4.474	Pseudoknot
2HW8	36	1	1	30/30	4.255	4.208	Hairpin, internal loop
1I6U	37	0	0	28/28	3.651	3.651	Hairpin, internal loop
1ZHO	38	1	1	29/29	4.094	5.07	Hairpin, internal loop
1F1T	38	1	1	31/31	8.015	2.82	Hairpin, internal loop
1XJR	47	2	1	37/37	2.88	4.70	Hairpin, internal loop
1S03	47	1	0	35/35	2.907	2.907	Hairpin, internal loop
2PXB	49	2	2	36/36	2.506	2.592	Hairpin, internal loop
1U63	49	1	1	38/38	4.991	4.562	Hairpin, internal loop
3E5C	54	0	0	40/39	1.716	1.716	3-way (riboswitch)
1MZP	55	11	7	44/38	6.648	8.077	Hairpin, internal
1DK1	57	0	0	46/45	1.686	1.686	3-way
1MMS	58	1	1	57/45	2.864	2.864	3-way
3EGZ	65	4	4	39/35	10.33	7.14	3-way (riboswitch)
2QUS	69	3	3	48/42	11.01	7.01	Pseudoknot,3-way
1KXK	70	0	0	54/54	3.261	3.261	Hairpin, internal loop
2DU3	71	4	3	50/43	2.993	4.491	4-way junction (tRNA)
2OIU	71	5	3	51/51	6.3	7.98	3-way junction
1SJ4	73	12	6	53/38	12.18	11.78	Pseudoknot,4-way
1P5O	77	2	0	51/51	5.766	5.766	Hairpin, internal loop
3D2G	77	2	0	70/60	7.162	7.162	3-way junction
2HOJ	79	3	3	64/58	2.564	3.257	3-way junction
2GDI	80	2	0	63/56	3.52	3.52	3-way junction
2GIS	95	10	7	79/67	14.13	14.80	Pseudoknot,4-way
1LNG	97	5	4	76/70	8.924	4.427	3-way junction (SRP)
1MFQ	128	2	0	101/92	19.3	19.3	3-way junction (SRP)

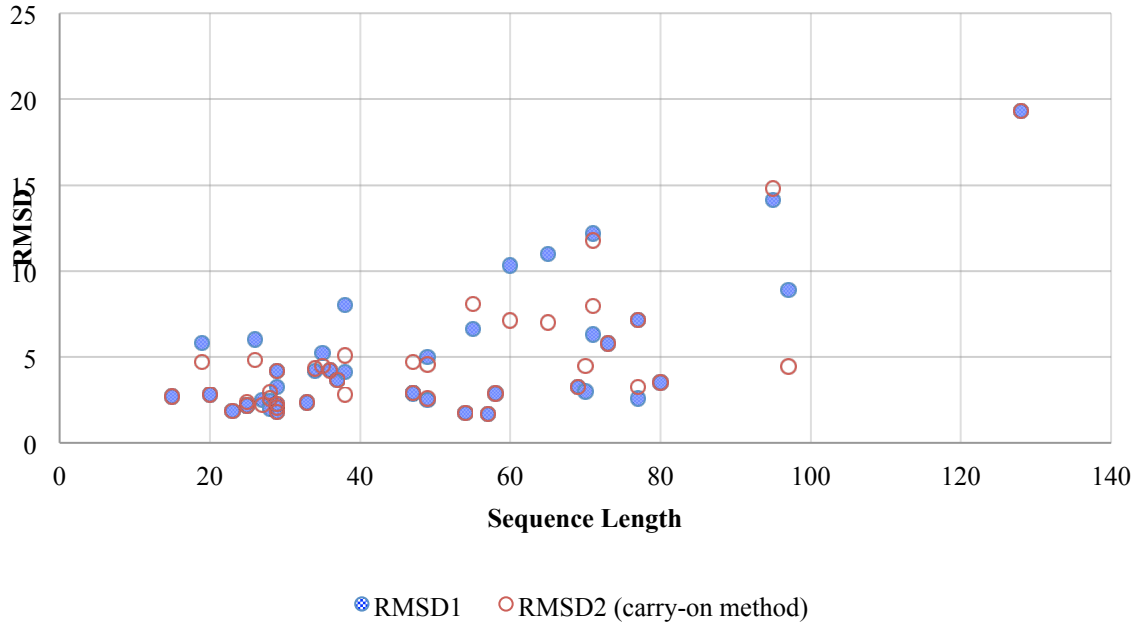


Fig. 5.3 The improvement of RMSDs generated by BkTree3D using the carry-on method, sorted by the RNA sequence lengths.

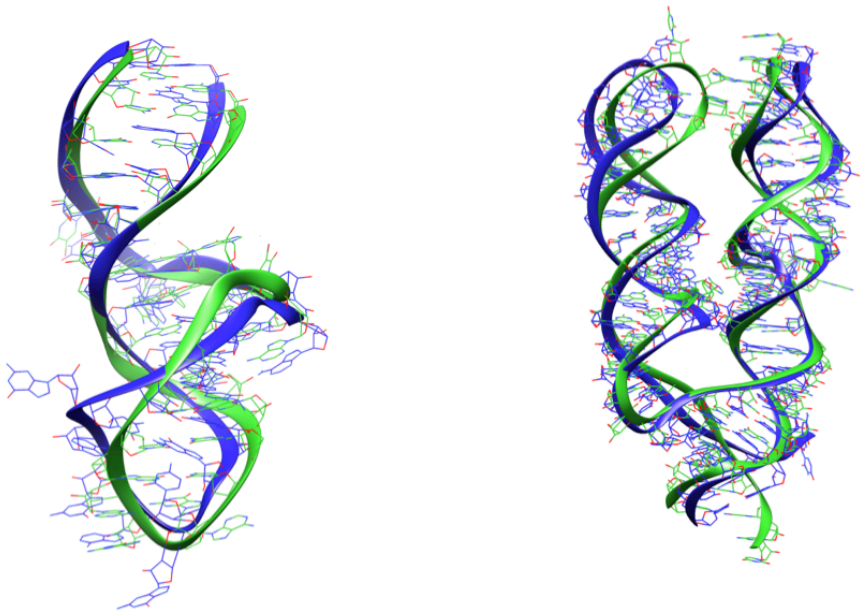


Fig. 5.4 Examples of predicted 3D models (blue) with the carry-on method superimposed with respective native structures (green). The superimposition on the left is 1F1T (38 nucleotides, hairpin, internal loop, 2.82Å); on right is 1LNG (97 nucleotides, 3-way junction, SRP, 4.43 Å). Their RMSDs are improved from 8.015Å to 2.82Å and 8.92Å to 4.43Å respectively.

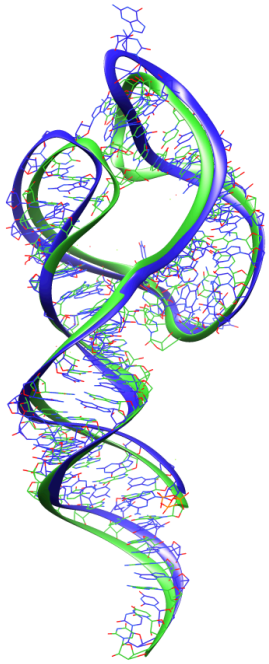


Fig. 5.5 Example of predicted 3D model (blue) with the 5-clique approximation method superimposed with respective native structures (green). The superimposition is 2QUS (69 nucleotides, pseudoknot, 4.94 Å). The RMSD is improved from 11.0Å to 4.94Å.

5.4 Other Proposed Methods

5.4.1 Dangling Geometry Database

We have foreseen that the performance of our 3D modeling method for the sequences of lengths over 100 may not be optimistic for several reasons besides the choice of the parameter *k* for the *k*-tree. First, the training data selected for building the geometry database are limited to the RNA sequences of length less than 100. Secondly, most of the dangling patterns are non-carriable in the predicted 3-trees. By the first reason, the conformation space of our geometry model might be limited for the dangling patterns, in particular, their dangling vertices. Furthermore, there is no constraint in current BkTree program to avoid the non-carriable dangling patterns. This means the 3D model will need to assign arbitrary geometries for dangling vertices from a limited geometry candidate pool. Therefore, if the correct geometry of dangling vertex is not covered in GDBs, our model probably would fail to deliver an accurate 3D modeling result.

To better handle the dangling geometry pattern issue, first we would like to know if there are geometrical redundancies for non-dangling vertices in a given set of geometry candidates $\mathcal{G}(\mathrm{A}(q))$ with dangling vertex *x* in clique q. Once we gain such knowledge, a new GDB named "*dangling database*" can be constructed to store geometry candidates for dangling interaction patterns.

Given a dangling interaction pattern $\mathrm{A}(q)$ with *x* be a dangling vertex in q, we will perform an all-to-all geometry alignment for every geometry candidate $g_{A(q)}^{q\setminus\{x\}}$ in $\mathcal{G}(\mathrm{A}(q))$

with a 0.5Å RSMD cut-off value to group them into clusters. The new set of geometry candidate $\mathcal{G}(\mathrm{A}(q))$ we are storing into *dangling* database would be one $g_{A(q)}^{q\backslash\{x\}}$ for each clustered group. Since we only store clustered geometries for non-dangling vertices, our preliminary test shows that the size of new constructed $\mathcal{G}(A(q))$ will be reduced dramatically. For example, given a dangling interaction pattern $\mathrm{A}(q)$ that contains key AGUC0010001 and interactions {1-3 cww, 1-4 s35, 3-4 s35} where the second vertex G is dangling, the size of $\mathcal{G}(A\ (q))$ can be reduced from 280 to 21 after the clustering of non-dangling vertices.

Furthermore, based on a BkTree predicted backbone 3-tree $< A,\ T >$, we may also construct a new 3-tree from its NIR graph that contains a minimal number of non-carriable dangling vertices by using the general 3-tree graph program which takes a fixed graph as input and output an optimal 3-tree that contains maximum number of edges from the inputted graph. The confidence score assignment function will be adjusted accordingly to fit into our goal. To implement this, we will assign a significant negative score to every clique that contains newly introduced dangling vertex during the 3-tree calculation. In the meantime, cliques with *dense* interaction patterns will receive more rewards on their scores. Such construction would be time efficient since the input graph is fixed. Then we will recursively apply the *geometry update* between neighboring cliques once it is needed and calculate more concrete geometry positions for dangling vertices during the dynamic programming instead of attempting to include more arbitrary geometry candidates.

5.4.2 Interaction Manipulation

To solve the problem of missing long-term interactions for pseudoknot structures, we have also investigated another possibility that involves interaction insertion and deletion. See Fig. 5.6 for an example. In Fig. 5.6(b), our 3D program can still predict the accurate structure from its known NIRs graph with cWW interaction edges missing from the kissing-hairpin. It suggests that an accurate backbone interaction association may recover the shape of missed long-distance interactions. Therefore, by inserting one s55 interaction to each side of kissing-hairpin helixes (Fig. 5.6 (c)), their backbones can be forced to twist toward each other instead of positioning with an wide-open "Y" shape.

We have yet to make the above process fully automatic and some human aid may still be needed to insert the needed interactions. But it offers an alternative way to study and predict RNA structures. We plan to gain more knowledge about the backbone motifs in pseudoknot structures and establish a sufficient backbone interaction set to recover the shape from the missed long-distance interactions through a more comprehensive analysis.

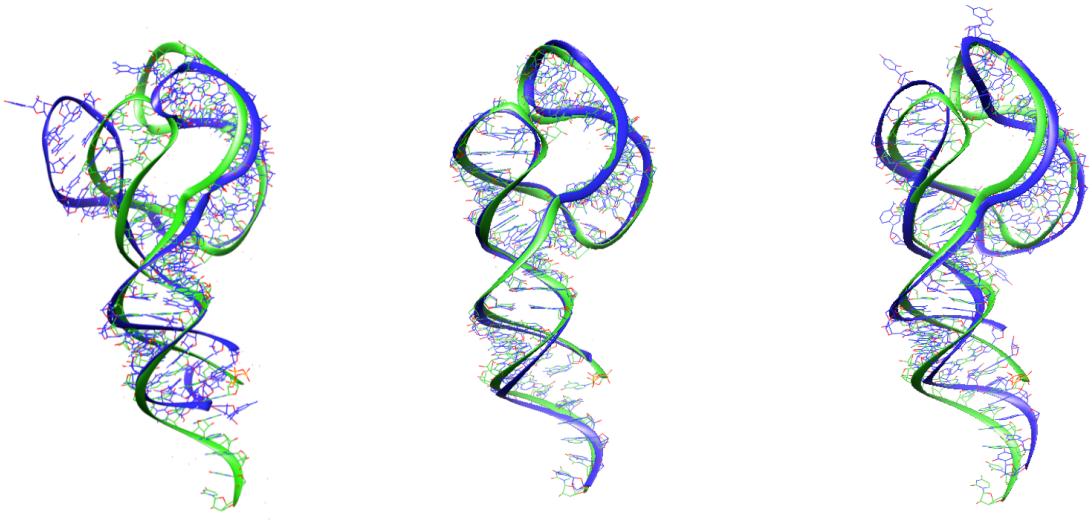


Fig. 5.6 Examples of predicted 2QUS (69 nucleotides, pseudoknot) 3D models (blue) superimposed with respective native structures (green). The predicted model in superimposition (a) is using BkTree predicted interactions (9.864Å); (b) is using resolved interactions (1.529 Å); (c) is using modified predicted interactions (5.58 Å).

CONCLUSION

We introduced a non-conventional framework to predict RNA 3D structures through predicting nucleotide interactions from sequences as an intermediate step. The underlying backbone k-tree model drastically reduces the space of plausible nucleotide interaction relations thus the geometries, permitting not only efficient but also effective prediction of RNA 3D models. The evaluation results have highlighted the performance of our method as a feasible one that can produce accurate 3D structure prediction of RNA sequences beyond short lengths.

To predict nucleotide interactions, our method is guided by the backbone k-tree, for small integer k, to globally constrain the nucleotide interaction relationships (NIRs) that constitute the 3D structure. Then the 3D modeling method introduced in Chapter 4 is specifically designed to produce a 3D model from a predicted backbone k-tree and an associated set of nucleotide interactions. The recursive rules for backbone k-tree generation have allowed us to associate scoring function with k-trees in a natural way.

The evaluation tests have revealed that our method is robust in the sense that only choices of k for the k-tree model may affect the prediction results, especially on 3D modeling performance. Table 4.1 shown that the BkTree3D modeling achieved RMSD values < 6.7Å for all but seven in the set of 43 RNAs. Then we improved the average RMSD value from 6.9Å to 4.9Å with an additional interactions preprocessing step. In the rationality

test, all RNAs achieved the same low RMSDs (see Table 4.2 and 4.3 for details). Furthermore, we implemented an improved version that can return suboptimal 3D structures ensured by the preserved monotone property within the alignment score calculation along the k-tree topology.

We have also investigated several methods that have potentials to cope with the limitations of our current k-tree model, especially, the choices of parameter k for the k-tree. Without a proper dangling database (yet to implement), the carry-on method can already improve the 3D model results generally (see Table 5.1 for details). Although the proposed 5-clique approximation method is not as practical as others, it may take advantage from the dangling database as well since the new added nucleotide tends to be dangling to the other ones. We are also optimistic that the interaction manipulation can lead us to a more accurate prediction by taking advantage of the recent growth of knowledge in the rich, high-resolution structured data.

WORKS CITED

1. Laing, C., Schlick, T.: Computational approaches to tertiary modeling of RNA. Journal of Physics(22:283101) (2010)
2. Leontis, N. B., Westhof, E.: RNA 3D Structure Analysis and Prediction. Springer (2012)
3. Doudna, J., Cech, T.: The chemical repertoire of natural ribozymes. nature 418, 222-228 (2002)
4. Baeyens, K. J., Jancarik, J., Holbrook, S. R.: Use of low-molecular-weight polyethylene glycol in the crystallization of RNA oligomers. Bio- logical Crystallography 50, 764-767 (1994)
5. Barta, A., Steiner, G., Brosius, J., Noller, H. F., Kuechler, E.: Identification of a site on 23s ribosomal-RNA located at the peptidyl transferase center. Proceedings of the National Academy of Sciences of the USA 81, 3607-3611 (1984)
6. Batey, R. T.: Advances in methods for native expression and purification of RNA for structural studies. Current Opinion in Structural Biology 26, 1-8 (2014)
7. Hannon, G. J.: RNA interference. Nature 42, 244-251 (2002)
8. Gillet, R., Felden, B.: Emerging views on tmRNA-mediated protein tagging and ribosome rescue. Mol. Microbiol. 42, 879-885 (2001)
9. Ruvkun, G.: Glimpses of a tiny RNA world. science 294, 797-799 (2001)

10. He, L.: A microRNA polycistron as a potential human oncogene. Nature 435, 828-833 (2005)

11. Felden, B.: RNA structure: experimental analysis. Curr. Opin. Microbiol 10, 286-291 (2007)

12. Latham, M. P.: NMR methods for studying the structure and dynamics of RNA. Chembiochem 6, 1492-1505 (2005)

13. Marti-Renom, M. A., Capriotti, E.: Computational RNA Structure Prediction. Bioinform 3, 32-45 (2008)

14. Abraham, M., Dror, O., Nussinov, R., Wolfson, H.: Analysis and classification of RNA tertiary structures.. RNA 14, 274-289 (2008)

15. Batey, R. T., Rambo, R. P., Doudna, J. A.: Tertiary motifs in RNA structure and folding. Angew. Chem. Int. Ed. 38, 2326-2343 (1999)

16. Westhof, E., Auffinger, P.: RNA Tertiary Structure. (2000)

17. Ding, Y.: Ab initio RNA folding by discrete molecular dynamics: from structure prediction to folding mechanisms. RNA 14, 1164-1173 (2008)

18. Sharma, S., Ding, F., Dokholyan , N. V.: iFoldRNA: three-dimensional RNA structure prediction and folding. Bioinformatics 24, 1951-1952 (2008)

19. Das, R., Baker, D.: Automated de novo prediction of native-like RNA tertiary structures. Proc. Natl Acad. Sci. 104, 14664–14669 (2007)

20. Das, R.: Structural inference of native and partially folded RNA by high-throughput contact mapping. Proc. Natl Acad. Sci. USA 105, 4144-4149 (2008)

21. Jonikas, M. A.: Coarse-grained modeling of large RNA molecules with knowledge-

based potentials and structural filters., 189-199 (2009)

22. Parisien, M., Major, F.: The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. Nature 452, 51-55 (2008)
23. Martinez, H. M., Maizel, J. V., Shapiro, B. A.: RNA2D3D: a program for generating, viewing, and comparing 3-dimensional models of RNA. J. Biomol. Struct. Dyn. 25, 669-683
24. Jossinet, F. In: assemble. Available at: http://www.bioinformatics.org/assemble/
25. Bida, J. P., Maher, L. J.: Improved prediction of RNA 3D structure with insights into native state dynamics. RNA 18, 385-393 (2012)
26. Popenda, M., Szachniuk, M., Antczak, M., Purzycka, K. J., Lukasiak, P., Bartol, N., Blazewicz, J., Adamiak, R. W.: Automated 3D structure composition for large RNAs.. Nucleic Acids Research
27. Reinharz, V., Major, F., Waldispu¨hl, J.: Towards 3D structure prediction of large RNA molecules: an integer programming framework to insert local tertiary motifs in RNA secondary structure. Bioinformatics 28, i207-i214 (2013)
28. Jossinet, F., Ludwig, T. E., Westhof, E.: Assemble: An interactive graphical tool to analyze and build RNA architectures at the 2D and tertiary levels.. Bioinformatics 26 , 2057-2059 (2010)
29. Martinez, H. M., Maizel, J.V., Shapiro, B. A.: RNA2Dtertiary: A program for generating, viewing, and comparing 3-dimensional models of RNA.. Journal of Biomolecular Structure Dynamics 25, 669-683 (2008)
30. Ding, L., Xue, X., LaMarca, S., Mohebbi, M., Samad, A., Malmberg, R., Cai, L.: Ab

initio prediction of RNA nucleotide interactions with backbone k-tree model. Proceedings of 1st Workshop on Computational Methods for Structural RNAs (CMSR'14)

31. Bodlaender, H. L., Koster, A. M. C. A.: Treewidth computations I. Upper bounds. Information and Computation 208(3), 259-275 (2010)
32. Van Leeuwen, J.: Graph algorithms Handbook of Theoretical Computer Science, A: Algorithms and Complexity theory., North Halland (1990)
33. Sarver, M., Zirbel, C. L., Stombaugh, J., Mokdad, A., Leontis, N. B.: FR3D: Finding Local and Composite Recurrent Structural Motifs in RNA 3D Structures.. Journal of Mathematical Biology 56, 215-252 (2008)
34. Bodlaender, H. L.: Dynamic programming on graphs with bounded treewidth. Proc. 15th International Colloquium on Automata, Languages and Programming, Lecture Notes in Computer Science 317, 105-118 (1988)
35. Arnborg, S., Proskurowski, A.: Linear time algorithms for NP-hard problems restricted to partial k-trees. Discrete Applied Mathematics 23, 11-24 (1989)
36. Leontis, N. B., Westhof, E.: Geometric nomenclature and classification of RNA base pairs. RNA, 499-512 (2001)
37. Leontis, N. B., Stombaugh, J., Westhof, E.: The non-Watson–Crick base pairs and their associated isostericity matrices. Nucleic Acids Res 30, 3497-3531 (2002)
38. Leontis, N. B., Lescoute, A., Westhof, E.: The building blocks and motifs of RNA architecture. Curr. Opin. Struct. Biol 16, 279-287 (2006)
39. Zirbel, C. L.: Classification and energetics of the base–phosphate interactions in

RNA. Nucleic Acids Res. 37, 4898–4918 (2009)

40. Parsien, M., Cruz, J. A., Westhof, E., Major, F.: New metrics for comparing and assessing discrepancies between RNA 3D structures and models. RNA 15, 1875-1885 (2009)

41. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res 9, 133–148 (1981)

42. Hofacker, I. L., Stadler, P. F.: Memory efficient folding algorithms for circular RNA secondary structures. Bioinformatics 22, 1172-1176 (2006)

43. Do, C. b., Woods, D. A., Batzoglou, S.: RNA secondary structure prediction from sequence alignments using a network of k-nearest neighbor classifiers RNA., 342-352 (2006)

44. Thirumalai, D., Woodson, S.: Kinetics of folding of proteins and RNA. Acc Chem Res 29, 433-439 (1996)

45. Zhuang, X., Bartley, L., Babcock, H., Russell, R., Ha, T., Herschlag, D., Chu, S.: A single-mole- cule study of RNA catalysis and folding. science 288, 2048–2051 (2000)

46. Thirumalai, D., Lee, N., Woodson, S., Klimov, D.: Early events in RNA folding. Annu Rev Phys Chem 52, 751–762 (2001)

47. Rangan, P., Masquida, B., Westhof, E., Woodson, S.: Assembly of core helices and rapid terti- ary folding of a small bacterial group I ribozyme. Proc Natl Acad Sci USA 100, 1574–1579 (2003)

48. Sykes, M., Levitt, M.: Describing RNA structure by libraries of clustered nucleotide dou- blets.. J Mol Biol 351, 26–38 (2005)

49. Richardson, J., Schneider, B., Murray, L., Kapral, G., Immormino, R., Headd, J., Richardson, D., Ham, D., Hershkovits, E., Williams, L., Keating, K., Pyle, A., Westbrook, J., Berman, H.: RNA backbone: consensus all-angle conformers and modular string nomenclature (an RNA ontology consortium contribution). RNA 14, 465–481 (2008)

50. Dima, R. I., Thirumalai, D.: Probing the instabilities in the dynamics of helical fragments from mouse PrPc. Proc. Natl. Acad. Sci. 101(43), 15335-15340 (2004)

51. Hyeon, C., Dima, R., Thirumalai, D.: Size, shape, and flexibility of RNA structures. J. Chem. Phys. 125 (2006)

52. Thirumalai, D., Hyeon, C.: Theory of RNA Folding: From Hairpins to Ribozymes. In : Non-Protein Coding RNAs 17. 27-48

53. Thirumalai, D., Hyeon, C.: RNA and protein folding: common themes and variations. Biochemistry 44(13), 4957–4970 (2005)

54. Chauhan, S., Woodson, S.: Tertiary interactions determine the accuracy of RNA folding. J Am Chem Soc 130, 1296–1303 (2008)

55. Nasalean, L., Stombaugh, J., Zirbel, C., Leontis, N. B.: RNA 3D Structural Motifs: Definition, Identification, Annotation, and Database Searching. In : Non-Protein Coding RNAs 20. 1-26

56. Fresllsen , J., Moltke, I., Thiim, M., Mardia, K. V., Hamelryck, T.: A Probabilistic Model of RNA Conformational Space. (2009)

57. Ding, L., Samad, A., Xue, X., Huang, X., Malmberg, R., Cai, L.: Stochastic k-tree grammar and its application in biomolecular structure modeling.. Lecture Notes in

Computer Science 8370, 308-322

58. Bodlaender, H. L.: A Tourist Guide through Treewidth. Acta Cybernetica (1993)
59. Bodlaender, H. L.: A Partial K-Arboretum of Graphs With Bounded Treewidth. Theoretical Computer Science 209, 1-45 (1998)
60. Arnborg, S., Corneil, D., Proskurowski, A.: Complexity of finding embeddings in a k-tree. SIAM Journal on Matrix Analysis and Applications , 277-284 (1987)
61. Robertson, N., Seymour, P. D.: Graph minors III: Planar tree-width. Journal of Combinatorial Theory 36, 49-64
62. Robertson, N., Seymour, P. D.: Graph minors II. Algorithmic aspects of tree-width. Journal of Algorithms 7, 309-322 (1986)
63. Bodlaender, H. L.: Treewidth: Characterizations, Applications, and Computations.. Proceedings of Workshop in Graph Theory , 1-14 (2006)
64. Mitchell, T.: Machine Learning., New york, USA (1997)
65. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N., Bourne, P. E.: The Protein Data Bank. Nucleic Acids Research 28, 235-242 (2000)
66. Salomon-Ferrer, R., Case, D. A., Walker, R. C.: An overview of the Amber biomolecular simulation package. WIREs Computational Molecular Science. (2012)
67. Golub, G., Van Loan: MATRIX COMPUTATIONS. (1989)