BIOINFORMATICS

Vol. 00 no. 00
Pages 1–8

Topology and prediction of RNA pseudoknots

Christian M. Reidys[1,2]*, Fenix W.D. Huang[1], Jørgen E. Andersen[3], Robert C. Penner[3,4], Peter F. Stadler[5−9], and Markus E. Nebel[10]

[1]Center for Combinatorics, LPMC-TJKLC, Nankai University Tianjin 300071, P.R. China
[2]College of Life Science, Nankai University Tianjin 300071, P.R. China
[3]Center for Quantum Geometry of Moduli Spaces Aarhus University, DK-8000 Århus C, Denmark
[4]Math and Physics Departments, California Institute of Technology, Pasadena, California, USA
[5]Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig, Härtelstraße 16-18, D-04107 Leipzig, Germany.
[6]Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany
[7]RNomics Group, Fraunhofer IZI, Perlickstraße 1,D-04103 Leipzig, Germany
[8]Inst. f. Theoretical Chemistry, University of Vienna, Währingerstraße 17, A-1090 Vienna, Austria
[9]The Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, New Mexico, USA
[10]Department of Computer Science, University of Kaiserslautern, Germany

Received on *****; revised on *****; accepted on *****

Associate Editor: *****

ABSTRACT

Motivation: Several dynamic programming algorithms for predicting RNA structures with pseudoknots have been proposed that differ dramatically from one another in the classes of structures considered.

Results: Here we use the natural topological classification of RNA structures in terms of irreducible components that are embeddable in surfaces of fixed genus. We add to the conventional secondary structures four building blocks of genus one in order to construct certain structures of arbitrarily high genus. A corresponding unambiguous multiple context free grammar provides an efficient dynamic programming approach for energy minimization, partition function, and stochastic sampling. It admits a topology-dependent parametrization of pseudoknot penalties that increases the sensitivity and positive predictive value of predicted base pairs by 10-20% compared to earlier approaches. More general models based on building blocks of higher genus are also discussed.

Availability: The source code of `gfold` is freely available at `http://www.combinatorics.cn//cbpc/gfold.tar.gz`

Contact: duck@santafe.edu

Supplementary information Supplementary material containing a complete presentation of the algorithms, full proofs of theorems, and detailed performance data are available at *Bioinformatics online*.

1 INTRODUCTION

The global conformation of RNA molecules is to a large extent determined by topological constraints encoded at the level of secondary structure, i.e., by the mutual arrangements of the base paired helices (Bailor *et al.*, 2010). In this context, secondary structure is understood in a wider sense that includes pseudoknots. Although the vast majority of RNAs has simple, i.e., pseudoknot-free, secondary structure, `PseudoBase` (Taufer *et al.*, 2009) lists more than 250 records of pseudoknots determined by a variety of experimental and computational techniques including crystallography, NMR, mutational experiments, and comparative sequence analysis. In many cases, they are crucial for molecular function. Examples include the catalytic cores of several ribozymes (Doudna and Cech, 2002), programmed frameshifting (Namy *et al.*, 2006), and telomerase activity (Theimer *et al.*, 2005), reviewed in (Staple and Butcher, 2005; Giedroc and Cornish, 2009).

Secondary structures can been interpreted as matchings in a graph of permissible base pairs (Tabaska *et al.*, 1998). The energy of RNA folding is dominated by the stacking of adjacent base pairs, not by the hydrogen bonds of the individual base pairs (Mathews *et al.*, 1999). In contrast to maximum weighted matching, the general RNA folding problem with a stacking-based energy function is NP-complete (Akutsu, 2000; Lyngsø and Pedersen, 2000). The most commonly used RNA secondary structure prediction tools, including `mfold` (Zuker, 1989) and the `Vienna RNA Package` (Hofacker *et al.*, 1994), therefore exclude pseudoknots.

Polynomial-time dynamic programming algorithms can be devised, however, for certain restricted classes of pseudoknots. In contrast to the $O(n^2)$ space and $O(n^3)$ time solution for simple secondary structures (Waterman, 1978; Nussinov *et al.*, 1978; Zuker and Stiegler, 1981), however, most of these approaches are computationally much more demanding. The design of pseudoknot folding algorithms thus has been governed more by the need to limit computational cost and achieve a manageable complexity of the recursion than the conscious choice of a particularly natural search space of RNA structures. As a case in point, the class of structures underlying the algorithm by Rivas and Eddy (1999) was characterized only in a subsequent publication (Rivas and Eddy, 2000). The following references provide a certainly incomplete list of dynamic programming approaches to RNA structure prediction using different structure classes characterized in terms of recursion equations

*to whom correspondence should be addressed. Phone: *86-22-2350-6800; Fax: *86-22-2350-9272; `duck@santafe.edu`

and/or stochastic grammars: Rivas and Eddy (1999); Uemura Y. *et al.* (1999); Akutsu (2000); Lyngsø and Pedersen (2000); Cai *et al.* (2003); Dirks and Pierce (2003); Deogun *et al.* (2004); Reeder and Giegerich (2004); Li and Zhu (2005); Matsui *et al.* (2005); Kato *et al.* (2006); Chen *et al.* (2009). The inter-relationships of some of these classes of RNA structures have been clarified in part by Condon *et al.* (2004) and Rødland (2006). In addition to these exact algorithms, a plethora of heuristic approaches to pseudoknot prediction have been proposed in the literature; see e.g., (Metzler and Nebel, 2008; Chen, 2008) and the references therein.

At least three distinct classification schemes of RNA contact structures have been proposed: Haslinger and Stadler (1999) suggested using book-embeddings, Jin *et al.* (2008) focused on the maximal set of pairwise crossing base pairs, and Bon *et al.* (2008) based the classification on topological embeddings. While these classifications have in common that simple secondary structure forms the most primitive class of structures, they differ already in the construction of the first non-trivial class of pseudoknots. Despite their mathematical appeal, however, no efficient (polynomial-time) algorithms are available for predicting pseudoknotted structures even in the simplest case of 3-non-crossing RNA structures. A practically workable approach to 3-non-crossing structures requires the enumeration of an exponentially growing number of diagrams which are then "filled in" by means of dynamic programming (Huang *et al.*, 2009); a Monte-Carlo approach utilizing the topological approach with a very simple matching-like energy model was explored by (Vernizzi and Orland, 2005).

In this contribution, we show that the topological classification of RNA structures can be translated into efficient dynamic programming algorithms whose computational complexity is directly related to the maximal genus that is considered in certain irreducible building blocks of the structure.

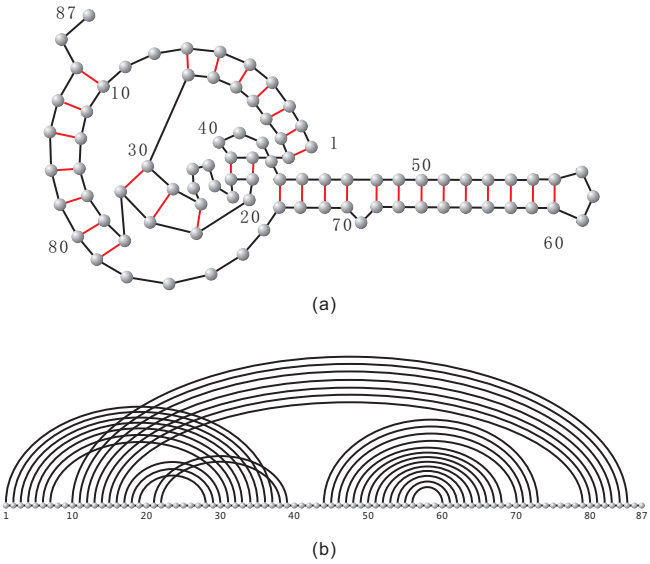


Fig. 1. RNA structures as planar graphs and diagram.

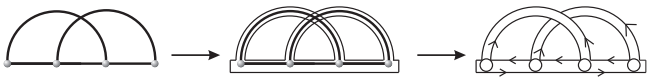


Fig. 2. Inflation of edges and vertices to ribbons and disks

2 RESULTS

2.1 Topology of RNA Structures

Diagram Representation. RNA molecules are linear biopolymers consisting of the four nucleotides **A**, **U**, **C**, and **G** characterized by a sequence endowed with a unique orientation ($5'$ to $3'$). Each nucleotide can interact (base pair) with at most one other nucleotide by means of specific hydrogen bonds. Only the Watson-Crick pairs **GC** and **AU** as well as the wobble **GU** are admissible. These base pairs determine the secondary structure. Note that we have neglected here base triples and other types of more complex interactions. Secondary structures can thus be represented as graphs or, more conveniently, labelled diagrams. In such a diagram, the backbone of the polymer is horizontally drawn as a chain. This chain consists of vertices and arcs respectively representing the nucleotides and covalent bonds. The base pairs are represented as arcs in the upper half-plane; see Fig. 1.

Thus, we shall identify a structure with a labelled graph over the vertex set $[N] = \{1, 2, \ldots, N\}$ represented by drawing the vertices $1, 2, \ldots, N$ on a horizontal line in the natural order and the arcs (i, j), where $i < j$, in the upper half-plane.

Fatgraph representation. In order to understand the topological properties of RNA molecules we need to pass from the picture of RNA as diagrams or contact-graphs to that of topological surfaces. Only the associated surface carries the important invariants leading to a meaningful filtration of RNA structures. Formally, we will view an RNA molecule as a topological surface having its diagram as deformation retract (Andersen *et al.*, 2010). The main idea is to "thicken" the edges into (untwisted) bands or ribbons and to expand each vertex to a disk as shown in Fig. 2. This inflation of edges leads to a fatgraph $\mathbb{D}$ (Loebl and Moffatt, 2008; Penner *et al.*, 2010).

A fatgraph, sometimes also called "ribbon graph" or "map", is a graph equipped with a cyclic ordering of the incident half-edges at each vertex. Thus, $\mathbb{D}$ refines its underlying graph D insofar as $\mathbb{D}$ encodes the ordering of the ribbons incident on its disks; in a further extension, the ribbons may also be allowed to twist giving rise to possibly non-orientable surfaces (Massey, 1967).

We interpret the boundary of each ribbon as oriented counterclockwise. A $\mathbb{D}$-cycle is constructed by following the boundaries of the ribbons from disk to disk thereby alternating between base pairs ribbon and backbone as shown in Fig. 3. The $\mathbb{D}$-cycles are therefore uniquely defined. Topological invariants such as the number of boundary components of the fatgraph $\mathbb{D}$ can thus be computed directly from the underlying diagram D. Furthermore, fatgraphs can be succinctly stored and conveniently manipulated on the computer as pairs of permutations (Penner *et al.*, 2010).

The fatgraph $\mathbb{D}$ gives rise to a unique surface $X_{\mathbb{D}}$, and each $\mathbb{D}$-cycle corresponds to a boundary component of $X_{\mathbb{D}}$, whose Euler characteristic and genus are given by

$$\chi(X_{\mathbb{D}}) = v - e + r \tag{2.1}$$

$$g(X_{\mathbb{D}}) = 1 - \frac{1}{2}\chi(X_{\mathbb{D}}), \tag{2.2}$$

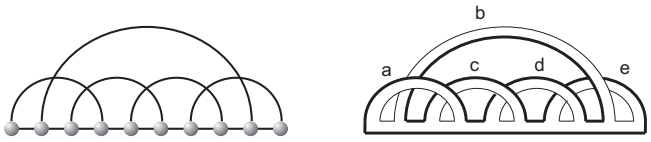


Fig. 3. Computing the number of boundary components. The diagram contains $5+9$ edges and 10 vertices. We follow the alternating paths described in the text and observe that there are exactly two boundary components (bold and thin). According to eq. (2.1), the genus of the diagram is given by $1-\frac{1}{2}(10-14+2)=2$.

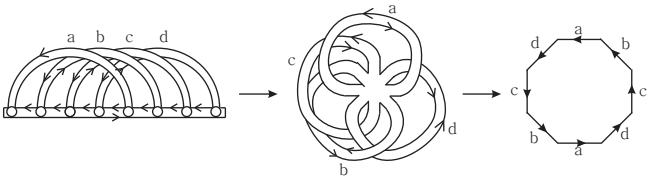


Fig. 4. Reduction to fatgraphs having a single vertex. Contracting the backbone of a diagram into a single vertex decreases the length of the boundary components and preserves the genus. The backbone of the polymer can be recovered by re-inflating the disk.

where v, e, r denotes the number of discs, ribbons and cycles in $\mathbb{D}$ (Massey, 1967). Furthermore, $\mathbb{D}$ has D as deformation retract.

We next make use of an additional feature of RNA structures, namely, that the backbone forms a unique oriented chain determined by the covalent bonds. Thus, the backbone can be collapsed to a single disk. The procedure can be undone by re-inflating the disk and rebuilding the backbone. This contracts the N vertices to a single one and removes the $N-1$ covalent bonds, see Fig. 4. It therefore preserves Euler characteristic and genus. Using the collapsed fatgraph, we see that the relation between the genus of the surface and the number of boundary components is determined by the number of arcs in the upper half-plane, namely,

$$2-2g-r=1-n, \tag{2.3}$$

where n is number of base pairs and r the number of boundary components. The latter can be computed easily and therefore controls the genus of the molecules.

From the collapsed fatgraph we can derive the *polygonal model* of the surface $X_{\mathbb{D}}$, that is, a polygon in which sides are identified in pairs; see Fig. 4.

2.2 γ-structures

The *shadow* of a diagram (RNA structure) is obtained by removing all non-crossing arcs, collapsing all isolated vertices and replacing all remaining stacks (i.e., adjacent parallel arcs) by single arcs; see Fig. 5. We show in this section that for any fixed g, there are only *finitely many* possible shadows S_g. This fact is of central algorithmic importance since an RNA structure of finite genus is thus determined by finitely many shadows.

We next come to the notion of γ-structures. An RNA structure is a γ-structure if its shadow can be decomposed iteratively *from bottom to top* by removing blocks of arcs corresponding to a shadow of genus γ and any stacked arcs that are induced by each

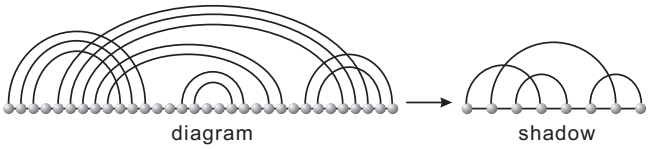


Fig. 5. Shadows: the shadow is obtained by removing all non-crossing arcs and isolated vertices and collapsing all resulting stacks into single arcs.

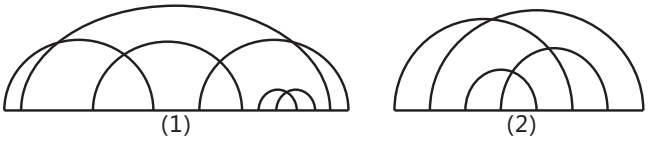


Fig. 6. γ-structures: we display the shadow of a 1-structure (left) having topological genus two and the shadow of the HDV-structure (right) (Ferré-D'Amaré *et al.*, 1998), a 2-structure having also genus two. Although both shadows have genus two, the HDV structure cannot be generated iteratively via successive removals of S_1-elements and stacked arcs and has order one. The structure displayed on the left has order two.

shadow-removal; see Fig. 6. Clearly, a γ-structure can have genus exceeding γ. Algorithmically speaking, a γ-structure is contained in the dynamic programming "hull" of the set S_γ of shadows of genus γ.

In order to quantify the relation between γ and the genus g of a γ-structure $\mathfrak{S}$, we consider the number $\omega(\mathfrak{S})$ of S_γ-substructures that have to be removed in order to decompose $\mathfrak{S}$. We call $\omega(\mathfrak{S})$ the *order* of $\mathfrak{S}$. Since stacked arcs that arise during the decomposition cannot contribute to the genus, we have

$$g(\mathfrak{S})=\omega(\mathfrak{S})\,\gamma. \tag{2.4}$$

The simplest class of structures are of course 0-structures, obtained as the dynamic programming hull over the base pair-free structure:

LEMMA 2.1. *An RNA structure is a* 0*-structure if and only if it is a simple secondary structure. In particular, a* 0*-structure always has genus* $g=0$.

PROOF. We first observe that a diagram of genus zero contains no crossing arcs. This follows from the fact that genus is a monotone non-decreasing function of the number of arcs (see eq. (2.3)) and that the genus of the matching (A) consisting of two mutually crossing arcs has only one boundary component and hence genus one; see Fig. 2. Second, we observe by induction on the number of arcs that each new non-crossing arc contributes a new boundary component and $2-2g-(r+1)=1-(n+1)$ shows that the genus remains zero. Structures consisting only of non-crossing arcs therefore have genus zero.

Next, we consider structures of arbitary genus. For their analysis, diagrams without isolated points, i.e., matchings, play a central role. Let $\mathscr{C}_g(n)$ be the set of matchings of genus g with n arcs, and let $\mathbf{c}_g(n):=|\mathscr{C}_g(n)|$ denote its cardinality. As shown by Andersen *et al.* (2010), the generating function $\mathbf{C}_g(z)=\sum_{n\geq 0}\mathbf{c}_g(n)\,z^n$ is

given by

$$\mathbf{C}_g(z) = P_g(z)\frac{\sqrt{1-4\,z}}{(1-4z)^{3g}}, \quad g \geq 1, \tag{2.5}$$

where $P_g(z)$ is an integral polynomial of degree $(3g-1)$ such that $P_g(1/4) \neq 0$. The number of genus zero matchings are well-known to be given by the Catalan numbers, and eq. (2.5) allows the derivation of explicit formulas for higher genera, for instance,

$$\mathbf{c}_1(n) = \frac{2^{n-2}(2n-1)!!}{3(n-2)!}, \ \mathbf{c}_2(n) = \frac{2^{n-4}(5n-2)(2n-1)!!}{90(n-4)!}.$$

Furthermore, the number $\mathbf{c}_g(2g)$ of matchings of genus g *having exactly* $2g$ *arcs*, i.e., matchings having exactly one boundary component, is the coefficient of z^{2g} in $P_g(z)$ and is given by

$$\mathbf{c}_g(2g) = \frac{(4g)!}{4^g(2g+1)!}.$$

Explicitly, we have $\mathbf{c}_1(2) = 1$, $\mathbf{c}_2(4) = 21$ and $\mathbf{c}_3(6) = 1485$ for example. These particular matchings will serve as "seeds" for our folding algorithm. More precisely, we shall use the following:

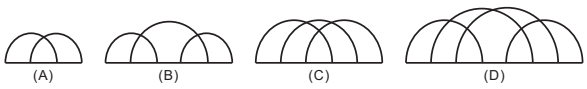
THEOREM 2.2. *For arbitrary genus g, the set S_g of shadows is finite. Every shadow in S_g contains at least $2g$ and at most $(6g-2)$ arcs.*

The special case $g = 1$, on which we focus in the algorithmic part of this contribution, is explicated in the Supplementary Material (SM).

PROOF. First note that if there is more than one boundary component, then there must be an arc with different boundary components on its two sides, and removing this arc decreases r by exactly one while preserving g since the number of arcs is given by $n = 2g + r - 1$. Furthermore, if there are ν_ℓ boundary components of length ℓ in the polygonal model, then $2n = \sum_\ell \ell\nu_\ell$ since each side of each arc is traversed once by the boundary. For a shadow, $\nu_1 = 0$ by definition, and $\nu_2 \leq 1$ as one sees directly. It therefore follows that $2n = \sum_\ell \ell\nu_\ell \geq 3(r-1)+2$, so $2n = 4g+2r-2 \geq 3r-1$, i.e., $4g - 1 \geq r$. Thus we have $n = 2g + (4g-1) - 1 = 6g - 2$, i.e. any shadow can contain at most $6g-2$ arcs. The lower bound $2g$ follows directly from $n = 2g + r - 1$ by observing $r = 1$.

Many S_g-shadows are in fact γ structures for some $\gamma < g$, that is, they can be constructed from elements of S_γ. One key result of this contribution is the following characterization of 1-structures:

THEOREM 2.3. *An RNA structure is a 1-structure if and only if its shadow can be decomposed by iteratively removing one of the four shadows*



In particular, 1-structures can have arbitrarily large topological genus.

PROOF. We only give a sketch here and refer to the SM for a full proof. First, we observe that taking the shadow preserves genus. Since (A) is the unique matching with two arcs of genus $g = 1$, it is contained in every matching of genus $g = 1$. An arc crossing into (A) preserves the genus and leads to either (B) or (C). While every arc added to (B) increases the genus, there is one possibility to preserve the genus when adding an arc to (C), namely, the addition leading to (D). It remains to observe that no further arc can be added to (D).

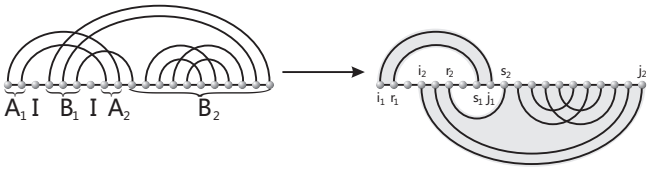


Fig. 7. Fragment-pairs in RNA structures: the rule $I \rightarrow IA_1IB_1IA_2IB_2S$ induces the fragment-pairs $[i_1, r_1]$, $[s_1, j_1]$ and $[i_2, r_2]$, $[s_2, j_2]$. Arcs connecting the two fragments of a pair are non-crossing, while arcs with both endpoints within the same fragment may be crossing such as those within $[s_2, j_2]$.

2.3 Minimum free energy folding of γ-structures

We have shown in the previous section that 0-structures are simple RNA secondary structures. Their minimum free energy (MFE) configuration can be obtained by dynamic programming recursions (Waterman, 1978; Zuker and Stiegler, 1981) derived from a decomposition into suitable substructures. This decomposition can be expressed in terms of a context-free grammar (Dowell and Eddy, 2004; Steffen and Giegerich, 2005). In the simplest case, which corresponds to evaluating base pairs only, we consider a single non-terminal symbol S representing an arbitrary diagram over a segment and three terminal symbols to represent isolated vertices (symbol :), openings (symbol () and closings (symbol)) of base pairs. We only need the three production-rules

$$S \rightarrow : S, \qquad S \rightarrow (S)S, \qquad S \rightarrow \varepsilon, \tag{2.6}$$

to generate the corresponding language $\mathcal{S}$.

Let us next consider RNA 1-structures. We shall use the that (1) any 1-structure can be inductively generated from genus one structures and (2) that every genus one structure has shadow (A), (B), (C), or (D), to specify a multiple context-free grammar (MCFG) (Seki *et al.*, 1991). In contrast to context-free grammars, the non-terminal symbols of MCFGs may consist of multiple components which must be expanded[1] in parallel. In this way, it becomes possible to couple separated parts of a derivation and thus to generate crossings. In the case of 1-structures, the language $\mathcal{S}$ is built upon sequences of intervals (*fragment-pairs*) $[i, r]$, $[s, j]$, where $(i, j), (r, s)$ are nested arcs. Arcs having endpoints in the different fragments are assumed to be non-crossing; see Fig. 7. For the MCFG, the fragments of a pair are associated with two different (coupled) components of a 2-dimensional non-terminal symbol.

Accordingly, we (re)introduce the following symbols:

[1] This coupling is only required for components that were generated by the same production step. Components, even if of the same kind, derived in different steps are independent of each other.

- non-terminal S, representing secondary structure elements (i.e., diagrams without crossing arcs) according to the rules given above,
- non-terminals I and T, representing an arbitrary 1-structure,
- non-terminals $\vec{X} = [X_1, X_2]$ with two components used to represent a fragment-pair with nested arcs, $X \in \{A, B, C, D\}$,
- terminals $(_X$, $)_X$ denoting the opening and closing of a base pair, resp., where X is one of A, B, C or D.

Different brackets as well as the different non-terminals of pattern $\vec{X}$ are used to distinguish nestings of the various kinds of shadows. Finally, we specify the production-rules of our unambiguous MCFG $\mathscr{R}_1$:

$$\begin{aligned}
I &\rightarrow S \mid T \\
S &\rightarrow (S)S \mid :S \mid \epsilon \\
T &\rightarrow I(T)S \\
T &\rightarrow IA_1IB_1IA_2IB_2S \\
T &\rightarrow IA_1IB_1IA_2IC_1IB_2IC_2S \\
T &\rightarrow IA_1IB_1IC_1IA_2IB_2IC_2S \\
T &\rightarrow IA_1IB_1IC_1IA_2ID_1IB_2IC_2ID_2S \\
\vec{X} &\rightarrow [(_X IX_1, X_2I)_X] \mid [(_X,)_X] ,
\end{aligned}$$

where $X \in \{A, B, C, D\}$ distinguishes the four types of pseudoknots.

THEOREM 2.4. *Any RNA* 1*-structure can be* ***uniquely*** *decomposed via $\mathscr{R}_1$, and any diagram generated via $\mathscr{R}_1$ is a* 1*-structure.*

PROOF. We only sketch the proof. All technical details can be found in the SM. The proof uses induction on the order $\omega(\mathfrak{S})$. We first demonstrate that any 1-structure $\mathfrak{S}$ can be uniquely decomposed. In the following, we consider unpaired vertices to be contained in $\mathcal{S}$. We start the decomposition from right to left. According to the above, we have to consider a paired vertex and distinguish the cases of the corresponding arc α to be (a) non-crossing or (b) crossing. In case (a), there exists a 1-structure that is nested in this arc. In case (b), we consider the set $C(\alpha)$ of arcs that are crossed by α. Consider the minimal arc α_* that crosses any $C(\alpha)$-arc, where minimal is with respect to the partial order $\prec$, where $(i, j) \prec (r, s)$ iff $r < i$ and $j < s$. It follows that $\alpha = (i, j)$ and $\alpha_* = (i_*, j_*)$ induce the fragment pair $[i, i_*]$ and $[j_*, j]$. Since $\mathfrak{S}$ is a 1-structure, this fragment pair belongs to a unique shadow of type (A), (B), (C), or (D), which in turn gives rise to at most four new fragment pairs. The theorem now follows since by construction the order of any substructure contained in such a fragment pair is reduced by one; see SM (Fig. 9).

If we make use of a naïve table-based parsing scheme, checking for each subword s of the input and for each rule f whether f can produce s, a rule like $f = I \rightarrow IA_1IB_2IC_1IA_2ID_1IB_2IC_2ID_2S$ introduces a complexity $O(n^{18})$: First, we must process $O(n^2)$ different subwords s induced by an input of size n. Second, each non-terminal but the first on the right-hand side of the production introduces an additional split point which specifies the part of s to be generated by the corresponding non-terminal. Since its location may freely be chosen within s, each split point gives rise to another loop variable, and hence contributes a factor $O(n)$ to the runtime.

Even if there are much more sophisticated parsing algorithms, it is useful to consider this simple scheme since it directly translates into a recursion for a dynamic programming algorithm typically used to compute structures of minimum free energy. Furthermore, it is possible to introduce intermediate steps in the derivation of our language by making use of additional non-terminals and production-rules such that the time complexity can be reduced to $O(n^6)$. For that purpose let the non-terminal I' represent 1-structures in which no structures with shadow (A), (B), (C) or (D) are nested and the last vertex is paired. We introduce the non-terminal symbols $\vec{U} = [U_1, U_2]$, $\vec{V} = [V_1, V_2]$ and $\vec{W} = [W_1, W_2]$ assumed to represent intermediate fragment-pairs and the production-rules

$$\begin{aligned}
\vec{U} &\rightarrow [IX_1, IX_2] \\
\vec{V} &\rightarrow [U_1U_1', U_2U_2'] \\
\vec{W} &\rightarrow [U_1, U_1'U_2U_2'] \mid [V_1, U_1V_2U_2]
\end{aligned}$$

where (U_1', U_2') is a marked copy of (U_1, U_2) used to identify the components which must later be expanded in a coupled way. Accordingly, we replace the derivations of T in $\mathscr{R}_1$ as follows:

$$\begin{aligned}
T &\rightarrow I(T)S \mid I'S \\
I' &\rightarrow V_1V_2 \mid U_1V_1U_2V_2 \mid U_1W_1U_2W_2
\end{aligned}$$

Note that syntactically, i.e., considered as dot-bracket representations, the 1-structures can be generated by a MCFG, parsable in time $O(n^5)$. However, in that case, corresponding brackets are not generated in a coupled way making the grammar inappropriate for algorithmic purposes.

As typical for dynamic programming and in analogy to our parsing scheme, we use 2-dimensional matrices to store the optimal structure over a fragment. The matrix is indexed by the sequence coordinates of the endpoints. It can be a simple secondary structure $\mathcal{S}$ or a substructure of higher genus. For the fragment-pairs, i.e., for the non-terminals of dimension two, 4-dimensional matrices indexed by the endpoints of both linked fragments are required to store the optimal structure over them. Suppose the pair of fragments is $[i, r]$ and $[s, j]$, and let $Gu(i, j; r, s)$ be the fragment-pair (associated with) $[U_1, U_2]$, $Gv(i, j; r, s)$ be the fragment-pair $[V_1, V_2]$, $Gw(i, j; r, s)$ be the fragment-pair $[W_1, W_2]$, and $G(i, j; r, s)$ be the fragment-pair $[X_1, X_2]$. The recursions for these matrices, summarized in graphical form in Fig. 8, are determined directly by the grammar.

We can conclude from the rewriting rules that the computation of the 2-dimensional matrices requires at most three loop variables, and there are $O(n^2)$ many of them. Accordingly, $O(n^5)$ operations are required to fill the associated 2-dimensional matrices. For the 4-dimensional matrices, two loop variables are needed for each of the corresponding rewriting rules (those with a left-hand side of dimension 2) for there are in each case two split points introduced by the right-hand sides of the corresponding productions. Since we need to compute $O(n^4)$ matrix entries, the total run time is in $O(n^6)$. Obviously, $O(n^4)$ space is required to store these tables. Accordingly, the algorithm can generate all 1-structures in $O(n^6)$ time and $O(n^4)$ space.

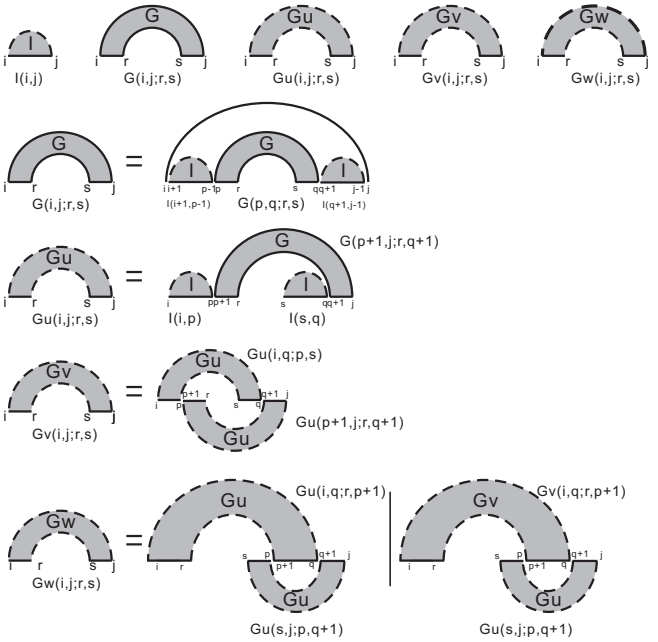


Fig. 8. The decomposition for 4-dimensional matrices G, Gu, Gv, and Gw.

2.4 Partition function and sampling

We have shown that the MCFG $\mathscr{R}_1$ uniquely generates all 1-structures, i.e., it is unambiguous. Consequently, $\mathscr{R}_1$ can be employed to count 1-structures over a given sequence x and to compute the corresponding partition function

$$Q = \sum_{s \in \mathfrak{S}_x} e^{-G(s)/RT},$$

where R is the universal gas constant, T is the temperature, $G(s)$ is energy of structure s over sequence x, and $\mathfrak{S}_x$ is the set of 1-structures in which all base pairs (i,j) satisfy the base pairing rules for RNA, i.e., $x_i x_j \in \{AU, UA, GC, CG, GU, UG\}$. Let $N_{i,j}$ denote the substructure represented by the nonterminal symbol N in $\mathscr{R}_1$ over the fragment $[i,j]$, and let $\vec{X}_{i,j;r,s}$ denote the fragment-pair $\vec{X} = [X_1, X_2]$, where $X_1 = [i,r]$ and $X_2 = [s,j]$ in the recursions for energy minimization. For each of these symbols, we introduce corresponding partial partition functions $Q_{N_{i,j}}$ and $Q_{\vec{X}_{i,j;r,s}}$. Since the MCFG is unambigous, the recursions for the partial partition functions are derived by replacing minima by sums and addition of energy contribution by multiplication of partial partition functions, see e.g., (Voß *et al.*, 2006). For instance, the recursion for the partition functions corresponding to the nonterminal symbol T reads

$$\begin{aligned} Q_{T_{i,j}} &= \sum_h Q_{I'_{i,h}} \times Q_{S_{h+1,j}} \\ &\quad + \sum_{h,\ell} Q_{I_{i,h-1}} \times Q_{T_{j+1,\ell-1}} \times Q_{S_{\ell+1,j}} \times e^{-E[h,\ell]/RT}, \end{aligned}$$

where $E[h,\ell]$ denotes the energy of the loop closed by the base pair (h,ℓ).

The probabilities $\mathbb{P}_{N_{i,j}}$ of partial structures of type N over the fragment $[i,j]$ and the probabilities $\mathbb{P}_{\vec{X}_{i,j;r,s}}$ of partial structures of type $\vec{X}$ over the fragment pair $[i,j],[r,s]$ are readily calculated from the partial partition functions. These "backward recursions" are analogous to those derived by McCaskill (1990) for crossing free structures: Let $\Lambda_{N_{i,j}}$ be the set of 1-structures containing $N_{i,j}$ and let $\Lambda_{\vec{X}_{i,j;r,s}}$ be the set of 1-structures containing the fragment-pair $\vec{X}_{i,j;r,s}$. It follows that we have

$$\mathbb{P}_{N_{i,j}} = \sum_{s \in \Lambda_{N_{i,j}}} \mathbb{P}_s, \qquad \mathbb{P}_{\vec{X}_{i,j;r,s}} = \sum_{s \in \Lambda_{\vec{X}_{i,j;r,s}}} \mathbb{P}_s.$$

Suppose $N_{i,j}$ or $\vec{X}_{i,j;r,s}$ are obtained by decomposing θ_s. The conditional probablities $\mathbb{P}_{N_{i,j}|\theta_s}$ and $\mathbb{P}_{\vec{X}_{i,j;r,s}|\theta_s}$ are then given by $Q_{\theta_s}(N_{i,j})/Q_{\theta_s}$ and $Q_{\theta_s}(\vec{X}_{i,j;r,s})/Q_{\theta_s}$ respectively. Here Q_{θ_s} represents the partition function of θ_s, and $Q_{\theta_s}(N_{i,j})$ and $Q_{\theta_s}(\vec{X}_{i,j;r,s})$ represent the partition functions for those θ_s-configurations that contain $N_{i,j}$ and $\vec{X}_{i,j;r,s}$ respectively. Taking the sum over all possible θ_s, we obtain

$$\mathbb{P}_{N_{i,j}} = \mathbb{P}_{\theta_s} \frac{Q_{\theta_s}(N_{i,j})}{Q_{\theta_s}}, \qquad \mathbb{P}_{\vec{X}_{i,j;r,s}} = \mathbb{P}_{\theta_s} \frac{Q_{\theta_s}(\vec{X}_{i,j;r,s})}{Q_{\theta_s}}.$$

From this backward recursion, one immediately derives a stochastic backtracing recursion from the probabilities of partial structures that generates a Boltzmann sample of 1-structures, see (Tacker *et al.*, 1996; Ding and Lawrence, 2003; Huang *et al.*, 2010) for analogous constructions.

The basic data structure for this sampling is a stack A which stores blocks of the form (i,j,N) (or $(i,j;r,s,\vec{X})$), presenting substructures of nonterminal symbols N over $[i,j]$ (or $\vec{X}$ over $[X_1,X_2]$ where $X_1 = [i,r]$ and $X_2 = [s,j]$). L is a set of base pairs storing those removed by the decomposition step in the grammar. We initialize with the block $(1,n,I)$ in A, and $L = \varnothing$. In each step, we pick up one element in A and decompose it via the grammar with probability Q^M/Q^N, where Q^N is the partition function of the block which is picked up from A, and Q^M is the partition function of the target block which is decomposed by the rewriting rule. The base pairs which are removed in the decomposition step are moved to L. For instance, according to the rewriting rule $T \to I(T)S$, the block (i,j,T) is decomposed into the three blocks: $(i,h-1,I)$, $(h+1,\ell-1,T)$, $(\ell+1,j,S)$ and one base pair (h,ℓ) which is to be removed. For fixed indices h, ℓ, where $i \le h < \ell \le j$, the probability of decomposing (i,j,T) reads

$$\mathbb{P}_{h,\ell} = \frac{Q_{I_{i,h-1}} \times Q_{T_{j+1,\ell-1}} \times Q_{S_{\ell+1,j}} \times e^{-E[h,\ell]/RT}}{Q_{T_{i,j}}}.$$

The sampling step is iterated until A is empty. The resulting 1-structure is the given by the list L of base pairs.

2.5 Software

Implementation. MFE folding, partition function including a computation of base pairing probabilities, and stochastic backtracing are implemented in `gfold`. The program is written in C.

Energy Model. Although the presentation above uses a simplified grammar that does not explicitly distinguish the usual loop types, `gfold` implements the Mathews-Turner energy model without dangles (Mathews *et al.*, 1999, 2004) for secondary structure elements. For pseudoknots, we use here an extended version of the

Dirks-Pierce (DP) model (Dirks and Pierce, 2003) that allows different penalties β_X for the four topologically distinct pseudoknot types $X = A, B, C, D$. We have observed that the values of β_X have a substantial influence on the accuracy of the predicted structures. In both NUPACK and pknotsRE, a common pseudoknot penalty β_1 is assigned whenever two gap matrices cross. Since the number of such crossings depends on the type of the pseudoknot, this algorithmic design would imply $\beta_A = \beta_1$, $\beta_B = \beta_C = 2\beta_1$, and $\beta_D = 3\beta_1$. In gfold, these parameters are independent and can be adjusted to improve the performance. Since most experimentally known pseudoknots are of types (A) and (B), we focused in particular on the ratio of β_A and β_B and found that both sensitivity and positive predictive value reach a maximum for $\beta_B = 1.3\beta_A$. The pseudoknot penalty of type (A) coincides with that of the DP model, i.e., $\beta_A = \beta_1 = 9.6$ [kcal/mol]. The other penalties are set to $\beta_B = 12.6$, $\beta_C = 14.6$, and $\beta_D = 17.6$; see SM for details. An alternative set of pseudoknot parameters described by Andronescu *et al.* (2010) can easily be incorporated but would require a re-adjustment of these four topological penalties.

Performance. The current implementation of gfold is applicable to sequences with a length up to $n \approx 150$ nucleotides on a PC with 1.2Gb memory, including the calculation of the partition function.

We have observed that gfold provides a substantial increase in both sensitivity (ratio of correctly predicted base pairs to the total number of base pairs in the reference structure) and a positive predictive value (PPV, ratio of correctly predicted base pairs to the total number of base pairs in the predicted structure) compared to the alternative dynamic programming approaches pknotsRE (Rivas and Eddy, 1999), NUPACK (Dirks and Pierce, 2003), and pknotsRG-mfe (Reeder and Giegerich, 2004), and that gfold provides a substantial increase in accuracy, cf. Fig. 9. In an evaluation on the entire Pseudobase (van Batenburg *et al.*, 2001), gfold achieves a sensitivity of 0.762 and PPV of 0.761. As detailed in SM (Tab.S-3), the performance varies substantially between different classes of sequences however. Interestingly, the more complex pseudoknots of type B are predicted with even higher accuracy (sensitivity 0.889, PPV 0.899) than the simpler, much more frequent type A.

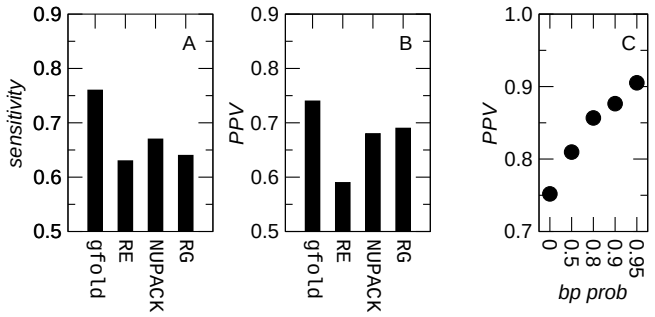


Fig. 9. Performance of gfold. Comparison of the average sensitivity (A) and PPV (B) of different prediction algorithms on a sample of 32 structures from Pseudobase. All details of this sample are given in the SM (Tab.S-2). (C) The PPV increases signficantly if only base pairs with larger pairing probabilities as predicted by the partition function version of gfold are included in the predicted structure.

The PPV of gfold predictions can be increased by filtering the base pairs of the MFE structure by their probability p of formation, which is computed by the partition function version of gfold. Accepting only base pairs with a predicted base pairing probability $p > 0.95$ increases the PPV from 0.76 to more than 0.9, see Fig. 9C. As for false positives, we folded 100 tRNA sequences from Sprinzl's tRNA database (Jühling *et al.*, 2009) and can report that gfold identifies 94% as pseudoknot-free. In comparison, NUPACK correctly identifies 86% and pknotsRG-mfe 89% of this sample set.

3 DISCUSSION

Combinatorial models of pseudoknotted RNA structures are limited in two ways: On the one hand, exact algorithmic folding can be constructed only for certain types of structures. On the other hand, the larger the structure sets are, the more base pairing patterns are contained in them that cannot be realized in nature due to steric constraints. Algorithm design so far has been mostly driven by the desire to reduce computational complexity. The idea behind gfold, in contrast, is to define a more suitable class of structures that can be generated by nesting and concatenating a small number of elementary building blocks. This recursive structure is captured by a fairly simple unambiguous multiple context-free grammar that translates in a canonical way to dynamic programming algorithms for computing the minimum energy structure and the partition function in $O(n^6)$ time and $O(n^4)$ space. In addition to MFE folding, we have implemented the computation of base pairing probabilities and a stochastic backtracing recursion, thus providing the major functionalities of RNA secondary structure prediction software for a very natural class of pseudoknotted structures.

The 1-structures considered here strike a balance between the generality necessary to cover almost all known pseudoknotted structures, and the restriction to topologically elementary structures that have a good chance to actually correspond to a feasible spatial structure. From a mathematical point of view, the characterization of structures in terms of irreducible components with given topological genus appears particularly natural and promises to reflect closely the ease with which a structure can be embedded in three dimensions. In addition, the grammar underlying gfold naturally distinguishes different types of pseudoknots and admits different energy parameters for them. We observe that this additional freedom of the parametrization leads to a substantial increase of sensitivity of type (B) pseudoknots, ($0.63 \rightarrow 0.889$) and PPV ($0.73 \rightarrow 0.899$) compared to the usage of a common penalty for each crossing of gap matrices. In terms of prediction accuracy, gfold thus compares favorably also with the leading alternative dynamic programming approaches to pseudoknotted structures.

Acknowledgements. This work was supported by the 973 Project of the Ministry of Science and Technology, the PCSIRT Project of the Ministry of Education, and the National Science Foundation of China to CMR and his lab, as well as the *Deutsche Forschungsgemeinschaft*, projects STA 850/2-1 & STA 850/7-1, the European Union FP-7 project QUANTOMICS (no. 222664) to PFS and his lab. JEA and RCP are supported by QGM, the Centre for Quantum Geometry of Moduli Spaces, funded by the Danish National Research Foundation

REFERENCES

Akutsu, T. (2000). Dynamic programming algorithms for RNA secondary structure prediction with pseudoknots. *Discr. Appl. Math.*, **104**, 45–62.

Andersen, J. E., Penner, R. C., Reidys, C. M., and Waterman, M. S. (2010). Enumeration of linear chord diagrams. *Comm. Pure and Appl. Math.* submitted.

Andronescu, M. S., Pop, C., and Condon, A. E. (2010). Improved free energy parameters for RNA pseudoknotted secondary structure prediction. *RNA*, **16**, 26–42.

Bailor, M. H., Sun, X., and Al-Hashimi, H. M. (2010). Topology links RNA secondary structure with global conformation, dynamics, and adaptation. *Science*, **327**, 202–206.

Bon, M., Vernizzi, G., Orland, H., and Zee, A. (2008). Topological classification of RNA structures. *J. Mol. Biol.*, **379**, 900–911.

Cai, L., Malmberg, R. L., and Wu, Y. (2003). Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. *Bioinformatics*, **19 S1**, i66–i73.

Chen, H.-L., Condon, A., and Jabbari, H. (2009). An $O(n^5)$ algorithm for MFE prediction of kissing hairpins and 4-chains in nucleic acids. *J. Comp. Biol.*, **16**, 803–815.

Chen, S. J. (2008). RNA folding: conformational statistics, folding kinetics, and ion electrostatics. *Annu Rev Biophys*, **37**, 197–214.

Condon, A., Davy, B., Rastegari, B., Zhao, S., and Tarrant, F. (2004). Classifying RNA pseudoknotted structures. *Theor. Comp. Sci.*, **320**, 35–50.

Deogun, J. S., Donis, R., Komina, O., and Ma, F. (2004). RNA secondary structure prediction with simple pseudoknots. In *Proceedings of the second conference on Asia-Pacific bioinformatics (APBC 2004)*, pages 239–246. Australian Computer Society.

Ding, Y. and Lawrence, C. E. (2003). A statistical sampling algorithm for rna secondary structure prediction. *Nucleic Acids Res.*, **31**, 7280–7301.

Dirks, R. M. and Pierce, N. A. (2003). A partition function algorithm for nucleic acid secondary structure including pseudoknots. *J. Comput. Chem.*, **24**, 1664–1677.

Doudna, J. A. and Cech, T. R. (2002). The chemical repertoire of natural ribozymes. *Nature*, **418**, 222–228.

Dowell, R. D. and Eddy, S. R. (2004). Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, **5**, 71.

Ferré-D'Amaré, A. R., Zhou, K., and Doudna, J. A. (1998). Crystal structure of a hepatitis delta virus ribozyme. *Nature*, **395**, 567–574.

Giedroc, D. P. and Cornish, P. V. (2009). Frameshifting RNA pseudoknots: structure and mechanism. *Virus Res.*, **139**, 193–208.

Haslinger, C. and Stadler, P. F. (1999). RNA structures with pseudo-knots: Graph-theoretical and combinatorial properties. *Bull. Math. Biol.*, **61**, 437–467.

Hofacker, I. L., Fontana, W., Stadler, P. F., Bonhoeffer, L. S., Tacker, M., and Schuster, P. (1994). Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.

Huang, F. W., Peng, W. W. J., and Reidys, C. M. (2009). Folding 3-noncrossing RNA pseudoknot structures. *J. Comp. Biol.*, **16**, 1549–1575.

Huang, F. W. D., Qin, J., Reidys, C. M., and Stadler, P. F. (2010). Target prediction and a statistical sampling algorithm for RNA-RNA interaction. *Bioinformatics*, **26**, 175–181.

Jin, E. Y., Qin, J., and Reidys, C. M. (2008). Combinatorics of RNA structures with pseudoknots. *Bull. Math. Biol.*, **70**, 45–67.

Jühling, F., Mörl, M., Hartmann, R. K., Sprinzl, M., Stadler, P. F., and Pütz, J. (2009). `tRNAdb` 2009: compilation of tRNA sequences and tRNA genes. *Nucleic Acids Res.*, **37**, D159–D162.

Kato, Y., Seki, H., and Kasami, T. (2006). RNA pseudoknotted structure prediction using stochastic multiple context-free grammar. *IPSJ Digital Courier*, **2**, 655–664.

Li, H. and Zhu, D. (2005). A new pseudoknots folding algorithm for RNA structure prediction. In L. Wang, editor, *COCOON 2005*, volume 3595, pages 94–103, Berlin. Springer.

Loebl, M. and Moffatt, I. (2008). The chromatic polynomial of fatgraphs and its categorification. *Adv. Math.*, **217**, 1558–1587.

Lyngsø, R. B. and Pedersen, C. N. (2000). RNA pseudoknot prediction in energy-based models. *J. Comp. Biol.*, **7**, 409–427.

Massey, W. S. (1967). *Algebraic Topology: An Introduction*. Springer-Veriag, New York.

Mathews, D., Sabina, J., Zuker, M., and Turner, D. (1999). Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.*, **288**, 911–940.

Mathews, D., Disney, M., Childs, J., Schroeder, S., Zuker, M., and Turner, D. (2004). Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc Natl Acad Sci*, **101**, 7287–7292.

Matsui, H., Sato, K., and Sakakibara, Y. (2005). Pair stochastic tree adjoining grammars for aligning and predicting pseudoknot RNA structures. *Bioinformatics*, **21**, 2611–2617.

McCaskill, J. S. (1990). The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, **29**, 1105–1119.

Metzler, D. and Nebel, M. E. (2008). Predicting RNA secondary structures with pseudoknots by MCMC sampling. *J. Math. Biol.*, **56**, 161–181.

Namy, O., Moran, S. J., Stuart, D. I., Gilbert, R. J. C., and Brierley, I. (2006). A mechanical explanation of RNA pseudoknot function in programmed ribosomal frameshifting. *Nature*, **441**, 244–247.

Nussinov, R., Piecznik, G., Griggs, J. R., and Kleitman, D. J. (1978). Algorithms for loop matching. *SIAM J. Appl. Math.*, **35**(1), 68–82.

Penner, R. C., Knudsen, M., Wiuf, C., and Andersen, J. E. (2010). Fatgraph models of proteins. *Comm. Pure Appl. Math.*, **63**, 1249–1297.

Reeder, J. and Giegerich, R. (2004). Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics*, **5**, 104.

Rivas, E. and Eddy, S. R. (1999). A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, **285**, 2053–2068.

Rivas, E. and Eddy, S. R. (2000). The language of RNA: A formal grammar that includes pseudoknots. *Bioinformatics*, **16**, 334–340.

Rødland, E. A. (2006). Pseudoknots in RNA secondary structures: Representation, enumeration, and prevalence. *J. Comp. Biol.*, **13**, 1197–1213.

Seki, H., Matsumura, T., Fujii, M., and Kasami, T. (1991). On multiple context free grammars. *Theor. Comp. Sci.*, **88**, 191–229.

Staple, D. W. and Butcher, S. E. (2005). Pseudoknots: RNA structures with diverse functions. *PLoS Biol.*, **3**, e213.

Steffen, P. and Giegerich, R. (2005). Versatile and declarative dynamic programming using pair algebras. *BMC Bioinformatics*, **6**, 224.

Tabaska, J. E., Cary, R. B., Gabow, H. N., and Stormo, G. D. (1998). An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics*, **14**, 691–699.

Tacker, M., Stadler, P. F., Bornberg-Bauer, E. G., Hofacker, I. L., and Schuster, P. (1996). Algorithm independent properties of RNA structure prediction. *Eur. Biophy. J.*, **25**, 115–130.

Taufer, M., Licon, A., Araiza, R., Mireles, D., van Batenburg, F. H. D., Gultyaev, A., and Leung, M.-Y. (2009). `PseudoBase++`: an extension of `PseudoBase` for easy searching, formatting and visualization of pseudoknots. *Nucleic Acids Res.*, **37**, D127–D135.

Theimer, C. A., Blois, C. A., and Feigon, J. (2005). Structure of the human telomerase RNA pseudoknot reveals conserved tertiary interactions essential for function. *Mol. Cell*, **17**, 671–682.

Uemura Y., Hasegawa, A., Kobayashi, S., and Yokomori, T. (1999). Tree adjoining grammars for RNA structure prediction. *Theor. Comp. Sci.*, **210**, 277–303.

van Batenburg, F. H. D., Gultyaev, A. P., and Pleij, C. W. A. (2001). PseudoBase: structural information on RNA pseudoknots. *Nucleic Acids Res.*, **29**, 194–195.

Vernizzi, G. and Orland, H. (2005). Large-N random matrices for RNA folding. *Acta Phys. Polon.*, **36**, 2821–2827.

Voß, B., Giegerich, R., and Rehmsmeier, M. (2006). Complete probabilistic analysis of RNA shapes. *BMC Biology*, **4**, 5.

Waterman, M. S. (1978). Secondary structure of single-stranded nucleic acids. *Adv. Math. (Suppl. Studies)*, **1**, 167–212.

Zuker, M. (1989). On finding all suboptimal foldings of an RNA molecule. *Science*, **244**, 48–52.

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