

RNA Secondary Structures

Ivo L. Hofacker¹ and Peter F. Stadler^{2,*}

¹Institute for Theoretical Chemistry, University of Vienna,
Währingerstrasse 17, A-1090 Vienna, Austria

²Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig,
Härtelstrasse 16-18, D-04107 Leipzig, Germany.

Phone: ++49 341 97 16691, Fax: ++49 341 97 16709,
email: studla@bioinf.uni-leipzig.de

1. Secondary Structure Graphs

1.1. Introduction. The tendency of complementary strands of DNA to form double helices is well known since the work of Watson & Crick. Single stranded nucleic acid sequences will in general contain many complementary regions that have the potential to form double helices when the molecule folds back onto itself. The resulting pattern of double helical stretches interspersed with loops is what is called the *secondary structure* of an RNA or DNA. Secondary structure elements may in turn be arranged in space to form 3-dimensional tertiary structure, leading to additional noncovalent interactions, an example is shown in Fig. 1. Energetically, however, these tertiary interactions are weaker than secondary structure. As a consequence RNA folding can be regarded as a hierarchical process in which secondary structure forms before tertiary structure [130, 129]. Since formation of tertiary structure usually does not induce changes in secondary structure, the two processes can be described independently. Functional RNA molecules (tRNAs, ribosomal RNAs, etc., as opposed to pure coding sequences), usually have characteristic spatial structures — and therefore also characteristic secondary structures — that are prerequisites for their function. As a consequence, secondary structures are highly conserved in evolution for many classes of RNA molecules.

Both the experimental determination of full spatial RNA structures and computational predictions of RNA 3D-structures are very hard tasks, arguable even harder than the corresponding problems for proteins [62, 82]. Computational approaches to RNA tertiary structure thus have been successful only for selected cases, see next chapter. RNA secondary structures, on the other hand, not only have a definite physical meaning as folding intermediates and are useful tools in the interpretation RNA molecules, but they give rise to efficient efficient computational techniques. Secondary structure prediction and comparison, the focal topics of this chapter, have therefore become a routine tool in the analysis of RNA function.

RNA secondary structures consists of two distinct classes of residues: those that are incorporated in double helical regions (so called stems), and those that are not

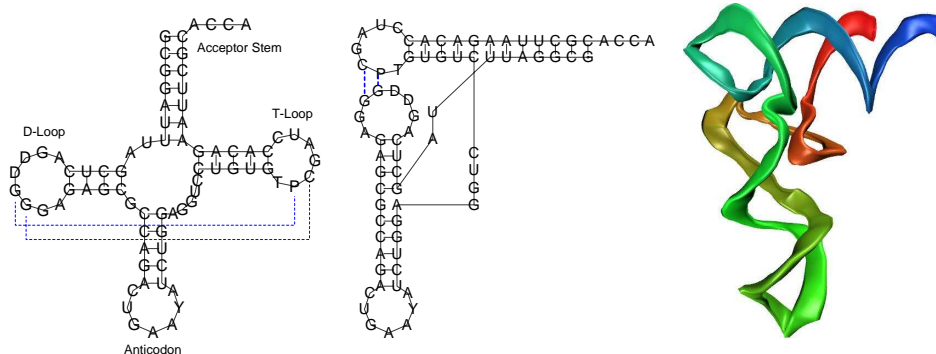


Figure 1. Secondary and tertiary structure of yeast phenylalanine tRNA. Left: The clover-leaf shaped secondary structure consisting of four helices. The dotted blue lines mark evolutionary conserved tertiary contacts. Middle: Co-axial stacking results in two extended helices that form the L-shaped tertiary structure. Right: Tertiary structure taken from PDB entry 4TRA. The color code (from red to blue) indicates the position along the chain.

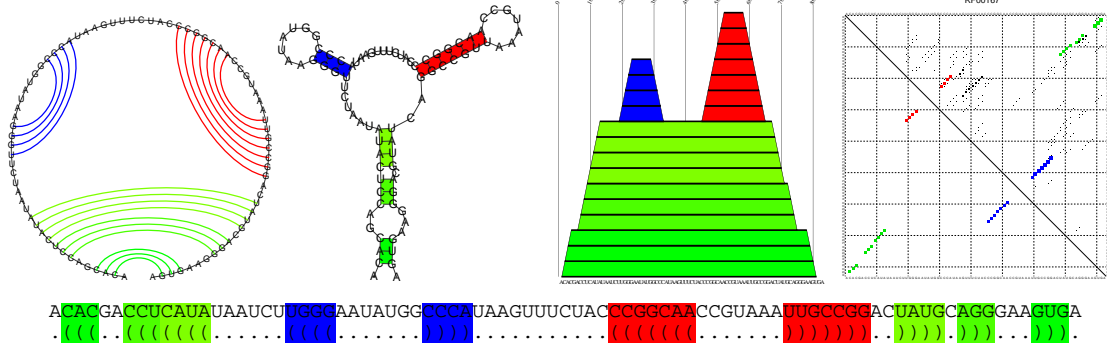


Figure 2. Representations of secondary structures. From left to right: Circle plot, conventional secondary structure graph, mountain plot, dot plot. Remove the backbone edges from the first two representations leaves the matching Ω . Below, the structure is shown in “bracket notation”, where each base pair corresponds to a pair of matching parentheses. The structure shown is the purine riboswitch (Rfam RF00167)

part of helices. For RNA, the double helical regions consist almost exclusively of Watson-Crick C-G and A-U pairs as well as the slightly weaker G-U wobble pairs. All other combinations of pairing nucleotides, called *non-canonical* pairs, are neglected in secondary structure prediction, although they do occur, especially in tertiary structure motifs.

1.2. Secondary Structure Graphs. A secondary structure is primarily a list of base pairs Ω . To ensure that the structure is feasible, a valid secondary structure should fulfill the following constraints:

- (1) A base cannot participate in more than one base pair, i.e., Ω is a matching on the set of sequence positions.
- (2) Bases that are paired with each other must be separated by at least 3 (unpaired) bases.
- (3) No two base pairs (i, j) and $(k, l) \in \Omega$ “cross” in the sense that $i < k < j < l$. Matchings that contain no crossing edges are known as loop matchings or circular matchings.

The first condition excludes tertiary structure motifs such as base triplets and G-quartets, the second takes into account that the RNA backbone cannot bend too sharply.

Base pairs that violate condition (3) are said to form a pseudoknot. While pseudoknots do occur in RNA structures, our definition (somewhat arbitrarily) classifies them as tertiary structure motifs. This is done in part because most dynamic programming algorithms cannot deal with pseudoknots. However, including pseudoknots entails other complications, since most hypothetical structures that violate condition 3 would also be sterically impossible. Furthermore, little is known about the energetics of pseudoknots, except for some data on H-type pseudoknots [41], the simplest and most common type of pseudoknot (see Fig. 3). Pseudoknots should therefore be regarded as a first step toward prediction of RNA tertiary structure.

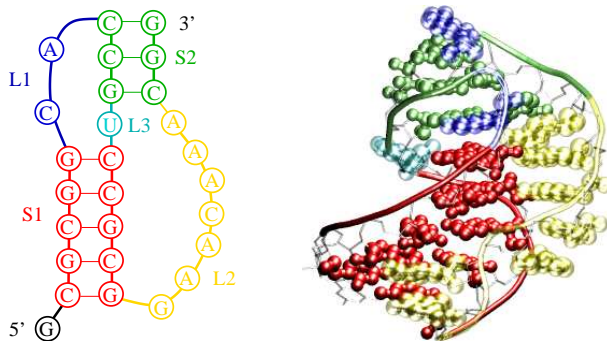


Figure 3. Example of an H-type pseudo-knot from beet western yellow virus. The crystal structure (right) shows that the two helices $S1$ and $S2$ are co-axially stacked to form a single 3-D helix.

Secondary structures can be represented by “secondary structure graphs”, first two panels in Fig. 2. In this representation one creates a graph whose nodes represent nucleotides. There are two kinds of edges, one kind representing the adjacency of nucleotides along the RNA sequence and the other kind representing base pairings. Condition (3) above assures that this graph is planar, and more precisely an *outer-planar graph*, in which all nodes can be arranged along a single face of the planar embedding made up by the edges forming the RNA sequence. We can therefore draw the secondary structure by placing the backbone on a circle and drawing a chord for every base pair such that no two chords intersect.

1.3. Mountain Plots and Dot Plots. A representation that works well for large structures and is well suited for comparing structures is the so-called mountain representation. In the mountain representation a single secondary structure is represented in a two-dimensional graph, in which the x -coordinate is the position k of a nucleotide in the sequence and the y -coordinate the number $m(k)$ of base pairs that enclose nucleotide k , third panel in Fig. 2.

Another possible representation is the dot plot, where each base pair (i, j) is represented by a dot or box in row i and column j of a rectangular grid, representing the contact matrix of the structure. Dot plots are well suited to represent structure *ensembles* by superimposing structural possibilities. In particular they are used to represent thermodynamic ensembles by plotting for each pair a box with area proportional to the equilibrium probability of the pair p_{ij} . Similarly, `mfold` uses colors to indicate the best possible energy for structures containing a particular pair, right-most panel in Fig. 2.

1.4. Trees and Forests. Secondary structures can also be stored compactly in strings consisting of dots and matching brackets: For any pair between positions i and j ($i < j$) we place an open bracket “(” at position i and a closed bracket “)” at j , while unpaired positions in the molecule are represented by a dot (“.”), see bottom of Fig. 2. Since base pairs may not cross, the representation is unambiguous.

An ordered forest F is a sequence of rooted ordered trees T_1, T_2, \dots, T_m such that within each tree T_i the left-to-right order of siblings (children of the same parent)

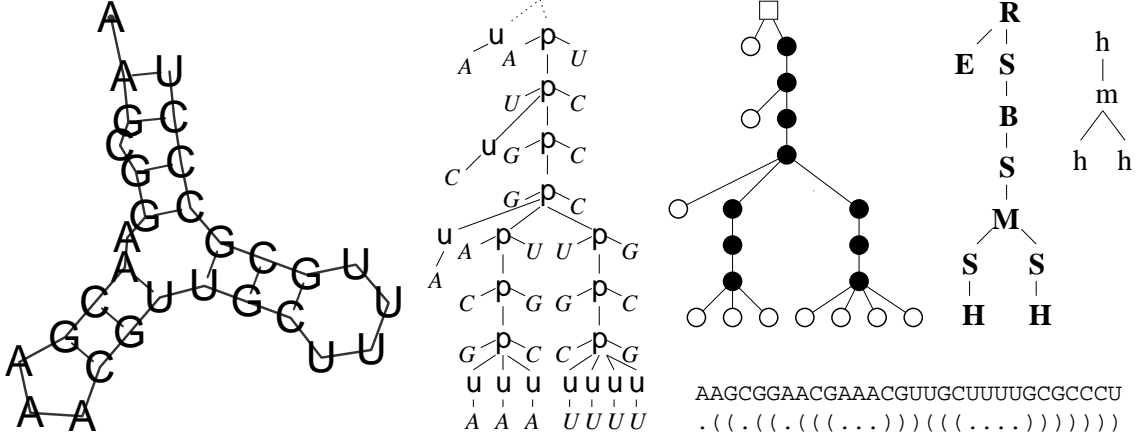


Figure 4. A variety of tree and forest representations of RNA secondary structures have been described in the literature. From left to right: conventional drawing, sequence annotated trees as used e.g. in *RNAforester* [51], “full tree” [32], Shapiro-style tree [119], and branching structure. For comparison, we also show the “bracket notation”

is given. In order to represent RNA secondary structures as ordered forests, we will need to associate a label from a suitable alphabet \mathcal{A} with each node.

This representation of secondary structures in terms of matched parentheses suggests to interpret the structure as a tree [119, 118]. In the *full tree* representation [32] each base pair corresponds to an interior node and each unpaired base is represented by a leaf, Fig. 4. A virtual root vertex is added mostly for graphical reasons.

Leaves may be labeled with the corresponding unpaired base, while interior nodes are labeled with the corresponding base pair. In an extended representation, two leaves, one labeled with the 5' and one labeled with the 3' nucleotide of the base pair, are attached as the left-most and right-most children to each interior vertex. In this representation the sequence of the molecule can be read off the leaves of the Bielefeld tree in pre-order .

Various coarse-grained representations have been considered. Homeomorphically irreducible trees represent entire helices as interior nodes while leaves correspond to runs of unpaired bases. Optionally, the length of such a structural element can be used as a weight. Shapiro-Zhang trees [119, 118] explicitly represent the different loop types (hairpin loop, interior loops, bulges, multiloops) as well as stacked regions with special labels. Fig. 4 summarizes a few examples.

1.5. Notes. Since RNA secondary structures are planar graphs, they can always be drawn on paper without intersections. Nevertheless, finding visually pleasing layout is difficult especially for large structures. Layout algorithms for RNA typically make use of the tree-like structure of secondary structure, see e.g. [18, 46, 98, 117]. The problem becomes more complicated when pseudoknots are allowed [48].

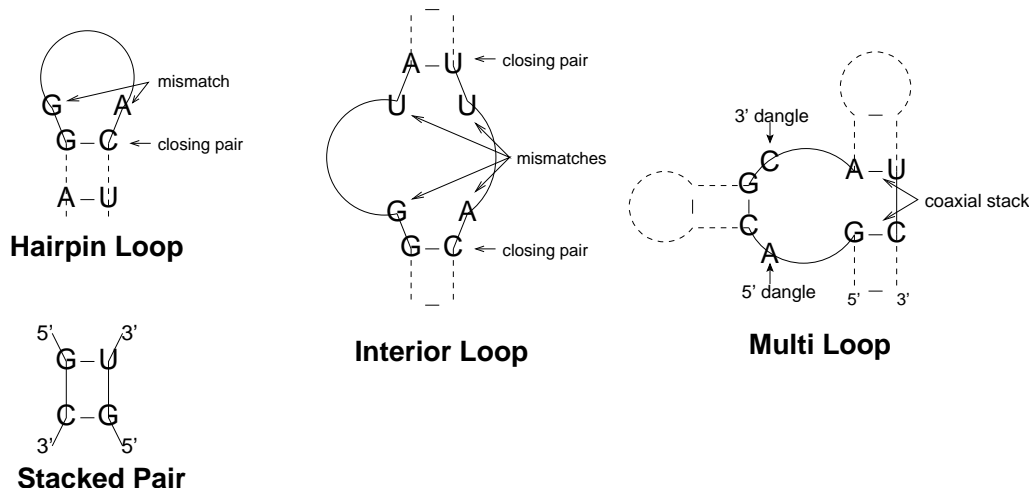


Figure 5. The major types of loops in RNA secondary structures

2. Loop-Based Energy Model

2.1. Loop decomposition. Secondary structures can be uniquely decomposed into *loops*, i.e., the faces of the planar drawing of the structure. More formally, we call a position k *immediately interior* of the pair (i, j) if $i < k < j$ and there exists no other base pair (p, q) such that $i < p < k < q < j$. A loop then consists of the closing pair (i, j) and all positions immediately interior of (i, j) . As a special case the exterior loop contains all positions not interior of any pair. The loops form a minimal cycle basis of the secondary structure graph, and this basis is unique for pseudoknot free structures [80].

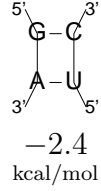
A loop is characterized by its length, i.e. the number of unpaired nucleotides in the loop, and its degree, given by the number of base pairs delimiting the loop (including the closing pair). Loops of degree 1 are called hairpin loops, interior loops have degree 2, and loops with degree > 2 are called multi-loops, see Fig. 5. Bulge loops are a special cases of interior loops in which there are unpaired bases only on one side, while stacked pairs correspond to an interior loop of size zero.

The loop decomposition forms the basis of the standard energy model for RNA secondary structures, where the total free energy of a structure is assumed to be a sum over the energies of its constituent loops. Because now the energy contribution of a base pair in a helix depends on the preceding and following pair, the model is often called the *nearest neighbor* model.

2.2. Energy Parameters. Note that a secondary structure corresponds to an *ensemble* of conformations of the molecule at atomic resolution, namely the set of all conformations compatible with a certain base pairing (hydrogen bonding) pattern. For example, no information is assumed about the spatial conformation of unpaired regions. The entropic contributions of these restricted conformations have to be taken into account, hence we are dealing with free energies which will be dependent on external parameters such as temperature and ionic conditions.

Table 1. Free energies for stacked pairs in kcal/mol.
Note that both base pairs have to be read in 5'-3' direction.

	CG	GC	GU	UG	AU	UA
CG	-2.4	-3.3	-2.1	-1.4	-2.1	-2.1
GC	-3.3	-3.4	-2.5	-1.5	-2.2	-2.4
GU	-2.1	-2.5	1.3	-0.5	-1.4	-1.3
UG	-1.4	-1.5	-0.5	0.3	-0.6	-1.0
AU	-2.1	-2.2	-1.4	-0.6	-1.1	-0.9
UA	-2.1	-2.4	-1.3	-1.0	-0.9	-1.3



-2.4
kcal/mol

Qualitatively, the major energy contributions are base stacking, hydrogen bonds, and loop entropies. While hydrogen bond and stacking energies *in vacuo* can be computed using quantum chemistry, the secondary structure model considers free energy differences between folded and unfolded states in an aqueous solution with rather high salt concentrations. As a consequence one has to rely on empirical energy parameters.

Energy parameters are typically derived by following the unfolding of RNA oligomers using A collection of energy parameters is maintained by the group of David Turner [145, 90, 91]. These standard parameters are measured in a buffer of 1M NaCl at 37C. As example we list the free energies for stacked pairs in table 1. Stacked pairs confer most of the stabilizing energy to a secondary structure, a single additional base pair can stabilize a structure by up to -3.4 kcal/mol. For comparison, the thermal energy¹ at room temperature is about $RT = 0.6$ kcal/mol, i.e., the stabilizing energy contribution of a single base pair is typically of the same order of magnitude as the thermal energy.

In general, loop energies depend on the loop type and its size. Except for small loops (which are tabulated exhaustively [91]), sequence dependence is conferred only through the base pairs closing the loop and the unpaired bases directly adjacent to the pair. Thus, the loop energy takes the form

$$E_{\text{loop}} = E_{\text{mismatch}} + E_{\text{size}} + E_{\text{special}} \quad (1)$$

where E_{mismatch} is the contribution from unpaired bases inside the closing pair and the base pairs immediately interior to the closing pair. The last term is used e.g. to assign bonus energies to unusually stable tetra loops, such as hairpin loops with the sequence motif GNRA. Polymer theory predicts that for large loops E_{size} should grow logarithmically. For multi-loops, however, energies that are linear in loop size and loop degree have to be used in order to allow efficient dynamic programming algorithms for structure prediction. While the model allows only Watson-Crick (AU, UA, CG and GC) and wobble pairs (GU, UG), non-standard base pairs in helices are treated as special types of interior loops in the most recent parameter sets.

The energy model above contains inaccuracies, on the one hand because it assumes that loop energies are strictly additive, on the other hand because energy parameters carry experimental errors (typically about 0.1 kcal/mol). Most seriously, the sequence

¹RNA energy parameters are still published in kcal/mol to facilitate comparison with previous parameter sets. 1kcal/mol \approx 4.2 kJ/mol in SI units.

dependence of loop energies has to be kept relatively simple, in order to deduce the parameters from a limited number of experiments.

2.3. Notes. Adjacent helices in multi-loops may stack co-axially to form a single extended helix. tRNA structures are prominent examples of this. In the four armed multiloop the acceptor stem co-axially stacks on the T stem, the anticodon stem stacks on the D-arm. This results in two extended helices which then form the L-shaped tertiary structure characteristic for tRNAs, see Fig. 1. Strictly speaking, co-axial stacking goes beyond the secondary structure model, since one has to know *which* helices in the loop will stack in order to include the energetic effect; the list of base pairs is no longer sufficient information to compute the energy. Co-axial stacking is also cumbersome to include in structure prediction algorithms. It has, however, been shown to improve prediction quality [135]. Useful energy parameters for structures with pseudoknots have so far only been collected for simple H-type pseudoknots [41].

3. The RNA Folding Problem

3.1. Counting structures and maximizing base pairs. In order to understand the basic ideas behind the dynamic programming algorithms for RNA folding, it is instructive to first consider the underlying combinatorial problem: *Given an RNA sequence x of length n , enumerate all secondary structures on x .* Let x_i denote the i -th position of x . We will simply write “ (i, j) pairs” to mean that the nucleotides x_i and x_j can form a Watson Crick or a wobble pair, i.e., $x_i x_j$ is one of GC, CG, AU, UA, GU, or UG. A subsequence (substring) will be denoted by $x[i..j]$. For notational convenience we interpret $x[j + 1..j]$ as the empty sequence and associate a single empty structure with it.

The basic idea is that a structure on n nucleotides can be formed in only two distinct ways from shorter structures: Either the first nucleotide is unpaired, in which case it is followed by an arbitrary structure on the shorter sequence $x[i + 1..j]$, or the first nucleotide is paired with some partner base, say k . In the latter case the rule that base pairs must not cross implies that we have independent secondary structures on the sub-intervals $x[i + 1..k - 1]$ and $x[k + 1..j]$. Graphically, we can write this decomposition of the set of structures like this:



It is now easy to compute the number N_{ij} of secondary structures on the subsequence $x[i..j]$ from positions i to j [140, 139]:

$$N_{ij} = N_{i+1,j} + \sum_{k, (i,k) \text{ pairs}} N_{i+1,k-1} N_{k+1,j} \quad (2)$$

with $N_{ii} = 1$. The independence of the structures on $x[i + 1..k - 1]$ and $x[k + 1..j]$ implies that we can simply multiply their numbers. This simple combinatorial

structure of secondary structures was realized by M. Waterman in the late 1970s [140, 139].

Historically, the first attempts at secondary structure prediction tried to maximize the number of base pairs in the structure. The solution to this problem by the Nussinov algorithm [101] is very similar to the combinatorial recursion above. Denote by E_{ij} the maximal number of base pairs in a secondary structure on $x[i..j]$. Using the decomposition of the structure set, we see that E_{ij} is the optimal choice among each of the alternatives. In this context, independence of two substructures in the paired cases implies that we have to optimize these substructures independently. If we like, we can associate each base pair with a weight (negative energy) β_{ik} which depends on x_i and x_j ; we arrive immediately at the recursion

$$E_{ij} = \max \left\{ E_{i+1,j}, \max_{k, (i,k) \text{ pairs}} \{E_{i+1,k-1} + E_{k+1,j} + \beta_{ik}\} \right\} \quad (3)$$

Replacing the weights by binding energies (which are negative for stabilizing interactions) we simply have to replace max by min in the above recursions. In practice, this simplified energy model does not lead to reasonable predictions in most cases. We use it here for didactic purposes and relegate as more detailed description of the complete RNA folding problem to subsection 3.3.

The energy contributions of individual base pairs are in the same order of magnitude as the thermal energy at room temperature. Thus RNA molecules exist in a distribution of structures rather than in a single ground-state structure. Thermodynamics dictates that, in equilibrium, the probability of a particular structure Ψ is proportional to its Boltzmann factor $\exp(-E(\Psi)/RT)$. Here $E(\Psi)$ is the energy of the sequence in conformation (secondary structure) Ψ , R is the molar gas constant (Boltzmann's constant in molar units), and T the absolute ambient temperature in Kelvin. This ensemble of structures is determined by its *partition function*

$$Z = \sum_{\Psi} \exp(-E(\Psi)/RT), \quad (4)$$

or, equivalently, by the free energy $\Delta G = -RT \ln Z$. The partition function Z can be computed in analogy to equ. (3). Using Z_{ij} as the partition function over all structures on sub-sequence $x[i..j]$ we obtain [93]

$$Z_{ij} = Z_{i+1,j} + \sum_{k, (i,k) \text{ pairs}} Z_{i+1,k-1} Z_{k+1,j} \exp(-\beta_{ik}/RT). \quad (5)$$

Note that we can transform the recursion for E_{ij} in equ.(3) into the equation for Z_{ij} simply by exchanging maximum operations with sums, sums with multiplications, and energies by their corresponding Boltzmann factors.

The partition function allows to compute the equilibrium probability of a structure Ψ as $p(\Psi) = \exp(-E(\Psi)/RT)/Z$. The formalism is also used to efficiently compute the equilibrium probability of a base pair $p_{ij} = \sum_{(i,j) \in \Psi} p(\Psi)$. To this end one needs to compute the partition function \hat{Z}_{ij} of structures *outside* the subsequence $x[i..j]$ using a recursion similar to the one above for Z . We can now compute the partition

Table 2. Comparison of backtracing recursions for different algorithms.

$\emptyset \rightarrow \mathfrak{S}$. while $\mathfrak{S} \neq \emptyset$ do $\pi \leftarrow \mathfrak{S}$; if π is complete then output π $[i, j] = I \in \Upsilon_\pi$. $\pi' = \pi \blacktriangleleft (i + 1)$ if $E(\pi') = E_{\text{opt}}$ then $\pi' \rightarrow \mathfrak{S}$; next ; for all $k \in [i, j]$ do $\pi' = \pi \blacktriangleleft (i, k)$ if $E(\pi') \leq E_{\text{opt}}$ then $\pi' \rightarrow \mathfrak{S}$; next ;	$\emptyset \rightarrow \mathfrak{S}$. while $\mathfrak{S} \neq \emptyset$ do $\pi \leftarrow \mathfrak{S}$; if π is complete then output π for all $[i, j] = I \in \Upsilon_\pi$ do $\pi' = \pi \blacktriangleleft (i + 1)$ if $E(\pi') \leq E_{\text{opt}} + \Delta E$ then $\pi' \rightarrow \mathfrak{S}$; for all $k \in [i, j]$ do $\pi' = \pi \blacktriangleleft (i, k)$ if $E(\pi') \leq E_{\text{opt}} + \Delta E$ then $\pi' \rightarrow \mathfrak{S}$;	$\emptyset \rightarrow \mathfrak{S}$. while $\mathfrak{S} \neq \emptyset$ do $\pi \leftarrow \mathfrak{S}$; if π is complete then output π for all $[i, j] = I \in \Upsilon_\pi$ do $\pi' = \pi \blacktriangleleft (i + 1)$ $\pi' \rightarrow \mathfrak{S}$ with probability $Z(\pi')/Z(\pi)$ for all $k \in [i, j]$ do $\pi' = \pi \blacktriangleleft (i, k)$ $\pi' \rightarrow \mathfrak{S}$ with prob. $Z(\pi')/Z(\pi)$
Algorithm B1. [101, 149] Backtracing a single structure	Algorithm B2. [144] Backtracing multiple structures	Algorithm B3. [21] Stochastic backtracing

function over all structures containing the pair (i, j) and thus its probability

$$p_{ij} = \widehat{Z}_{ij} Z_{i+1, j-1} \exp(-\beta_{ij}/RT)/Z. \quad (6)$$

Further variants of this scheme can be employed to compute e.g. the number of states with a given energy, to explicitly list all possible structures, or to determine structures that optimize other properties. In section 5.5 we will briefly mention how such variants can be constructed in a systematic way within the framework of *Algebraic Dynamic Programming* [37].

3.2. Backtracing. Recursion (3) computes only the optimal energy, not an optimal structure which realizes this energy. This is typical for most dynamic programming algorithms: one first compute the *value* of the optimum, then uses *backtracing* (sometimes called *backtracking*) to generate one (or more) structures in a step-wise fashion based on the information collected in the forward recursions. This section closely follows an exposition of the topic in [27]. The basic object is a *partial structure* π consisting of a collection Ω_π of base pairs and a collection Υ_π of sequence intervals in which the structure is not (yet) known. Positions that are known to be unpaired can easily be inferred from this information. The completely unknown structure on the sequence interval $[1, n]$ is therefore $\emptyset = (\emptyset, \{[1, n]\})$ while a structure is *complete* if it is of the form $\pi = (\Omega, \emptyset)$.

Suppose $I = [i, j] \in \Upsilon$ are positions for which the partial structure $\pi = (\Omega, \Upsilon)$ is still unknown. If we know that i is unpaired then $\pi' = (\Omega', \Upsilon')$ with $\Omega' = \Omega$ $\Upsilon' = \Upsilon \setminus \{I\} \cup \{[i + 1, j]\}$. If $(i, k), i < k \leq j$, is a base pair then $\Omega' = \Omega \cup \{(i, k)\}$ and $\Upsilon' = \Upsilon \setminus \{I\} \cup \{[i + 1, k - 1], [k + 1, j]\}$. Here we use the convention that empty intervals are ignored. Furthermore, base pairs can only be inserted within a single interval of the list Υ . We write $\pi' = \pi \blacktriangleleft (i)$ and $\pi' = \pi \blacktriangleleft (i, k)$ for these two cases.

The energy of a partial structure π is defined as

$$E(\pi) = \sum_{(k,l) \in \Omega} \beta_{kl} + \sum_{I \in \Upsilon} E_{\text{opt}}(I), \quad (7)$$

where $E_{\text{opt}}(I) = E_{ij}$ is the optimal energy for the substructure on the interval $I = [i, j]$

The standard backtracing for the minimal energy folding starts with the unknown structure. Instead of a recursive version we describe here a variant where incomplete structures are kept on a stack \mathfrak{S} . We write $\pi \leftarrow \mathfrak{S}$ to mean that π is popped from the stack and $\pi \rightarrow \mathfrak{S}$ to mean that π is pushed onto the stack.

If we want all optimal energy structures instead of a single representative we simply test all alternatives, i.e., we omit the **next** in the algorithm above. It is now almost trivial to modify the backtracing to produce all structures within an energy band $E_{\text{opt}} \leq E \leq E_{\text{max}}$ above the ground state.

Stochastic backtracing procedures for dynamic programming algorithm such as pairwise sequence alignment are well known [97]. Replacing Z_{ij} by N_{ij} in Algorithm B3 we recover recursions for producing a uniform ensemble of structures similar to the procedure for producing random structures without sequence constraint used in [126].

Note that the probabilities of $\pi \blacktriangleleft (i+1)$ and $\pi \blacktriangleleft (i, k)$ for all k add to 1 so that in each iteration we take exactly one step. Hence we simply fill one structure which we output as soon as it is complete.

3.3. Energy minimization in the loop-based energy model. Using the loop based energy model is essential in order to achieve reasonable prediction accuracies. As we shall see, the more complicated energy model results in somewhat more complicated recursions and requires additional tables. However, memory and CPU requirements are still $\mathcal{O}(n^3)$ and $\mathcal{O}(n^2)$. The main difference to the simple model discussed in the previous sections is that we now have to distinguish between different types of loops. Thus we have to further decompose the set of substructures enclosed by the base pair (i, k) according to the loop types: hairpin loop, interior loop, and multi(branched) loops, see Fig. 6. The hairpin and interior loop cases are simple since they reduce again to the same decomposition step.

The multiloop case is more complicated, however, since the multiloop energy depends explicitly on the number of substructures (“components”) that emanate from the loop. We therefore need to decompose the structures within the multiloop in such a way that we can at least implicitly keep track of the number of components. To this end we represent a substructure within a multiloop as a concatenation of two components: An arbitrary 5’ part that contains *at least* one component and a 3’ part that starts with a base pair and contains only a single component. These two types of multiloop substructures are now decomposed further into parts that we already know: unpaired intervals, structures enclosed by a base pair, and (shorter) multiloop substructures, see Fig. 6. It is not too hard to check that this decomposition really accounts for all possible structures and that each secondary structure has a unique decomposition.

Given the recursive decomposition of the structures, we can now rather easily derive the associated energy minimization algorithm. We will use the abbreviations $\mathcal{H}(i, j)$

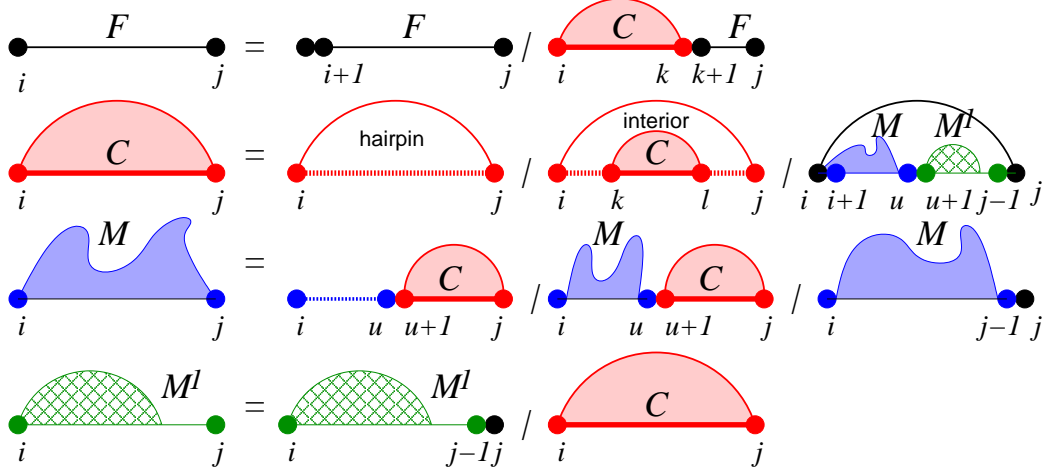


Figure 6. Decomposition of RNA secondary structure. Dotted lines indicate unpaired substructures, while full length denote arbitrary structures; base pairs are indicated as arcs. Multiloop contributions with an arbitrary number of components are shown as irregular “mountains”. See text for further details.

for the energy of a hairpin loop closed by the pair (i, j) , similarly $\mathcal{I}(i, j; k, l)$ shall denote the energy of an interior loop determined by the two base pairs (i, j) and (k, l) . We will also tabulate the following quantities:

F_{ij} free energy of the optimal substructure on the subsequence $x[i..j]$.

C_{ij} free energy of the optimal substructure on the subsequence $x[i..j]$ subject to the constraint that i and j form a base pair.

M_{ij} free energy of the optimal substructure on the subsequence $x[i..j]$ subject to the constraint that that $x[i..j]$ is part of a multiloop and has at least one component.

M_{ij}^1 free energy of the optimal substructure on the subsequence $x[i..j]$ subject to the constraint that that $x[i..j]$ is part of a multiloop and has exactly one component, which has the closing pair i, h for some h satisfying $i < h \leq j$.

The recursions for computing the minimum free energy of an RNA molecule in the loop based energy model were first formulated by Zuker & Stiegler [149]. They can be summarized as follows:

$$\begin{aligned}
 F_{ij} &= \min \left\{ F_{i+1,j}, \min_{i < k \leq j} (C_{ik} + F_{k+1,j}) \right\} \\
 C_{ij} &= \min \left\{ \mathcal{H}(i, j), \min_{i < k < l < j} (C_{kl} + \mathcal{I}(i, j; k, l)), \min_{i < u < j} (M_{i+1,u} + M_{u+1,j-1}^1 + a) \right\} \\
 M_{ij} &= \min \left\{ \min_{i < u < j} ((u - i + 1)c + C_{u+1,j} + b), \min_{i < u < j} (M_{i,u} + C_{u+1,j} + b), M_{i,j-1} + c \right\} \\
 M_{ij}^1 &= \min \{ M_{i,j-1}^1 + c, C_{ij} + b \},
 \end{aligned} \tag{8}$$

where we assume linear multiloop energies of the form $E_{\text{ML}} = a + b \cdot \text{degree} + c \cdot \text{size}$. In contrast to most implementations the version shown here decomposes structures

in such a way that each substructure occurs exactly once. While this is not strictly necessary for energy minimization, it allows us to use essentially the same recursions for all variants of the problem, including the computation of the partition function or the backtracing of all or a sample of suboptimal structures.

3.4. RNA Hybridization. Intermolecular base pairing between two RNA molecules can be treated in the same way as intramolecular interactions. The most straightforward approach is to concatenate the two molecules. One can then apply the folding algorithms for single molecules. There is only a single necessary modification to the folding algorithms: the energy contribution of the loop that contains the cut point is different. Implementations of this approach are `RNAcofold` [58] and `pairfold` [5].

From a physics point of view, however, additional effects need to be taken into account: the interaction of two distinct molecules is concentration dependent. Furthermore, there is an additional (entropic) contribution for the initiation of an intermolecular interaction. The extension of the folding algorithms of course compute both inter- and intra-molecular contributions. It is therefore necessary to correct for the initialization energy E^i :

$$\begin{aligned} Z_{AA} &= (Z_{AA}^\circ - Z_A^2) \exp(-E^i/RT), \\ Z_{BB} &= (Z_{BB}^\circ - Z_B^2) \exp(-E^i/RT), \quad \text{and} \\ Z_{AB} &= (Z_{AB}^\circ - Z_A Z_B) \exp(-E^i/RT). \end{aligned} \tag{9}$$

where Z° is the partition function as calculated from the folding algorithm for the concatenated sequences, Z_A and Z_B are the partition functions of the isolated molecules A and B . Standard statistical thermodynamics can then be used to compute the concentration dependencies of the complex formation, see e.g. [20] for a discussion in the context of RNA hybridization.

Various simplified approaches have been discussed in recent years. In particular, the most common approximation is to neglect the secondary structures of the two interacting molecules. This amounts to a model in which the concatenated structure can only have base pairs and interior loops and the cut point is located in the single hairpin loop. It does, however, result in a much faster algorithm with time complexity $\mathcal{O}(n \cdot m)$ instead of $\mathcal{O}((n + m)^3)$ for two sequences of length n and m . Algorithms for this case have been described in [20] and [105]; the `Vienna RNA Package` also provides an implementation. The `RNAhybrid` program was in particular used to detect microRNA/target interactions.

3.5. Pseudoknotted Structures. Many functionally important RNA structures contain pseudoknots, including rRNAs [12], RNase P RNAs [10, 49], and tmRNA [151]. Recently, algorithms have been described that are able to deal with certain classes of pseudoknotted structures. As we shall see below, these are however plagued by considerable computational costs. In addition, a common problem of all these approaches is the still very limited information about the energetics of pseudoknots [41, 66].

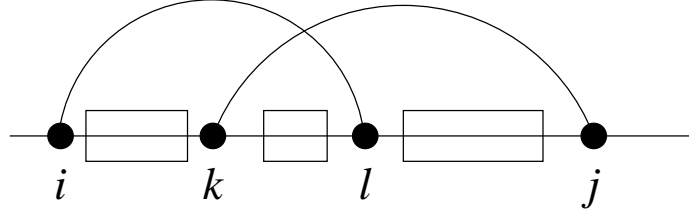


Figure 7. Additional requirements for computing pseudoknotted structures. In order to evaluate the contribution for the pseudoknot on $[i, j]$ we need to iterate over all combinations of cutpoints $k < l$.

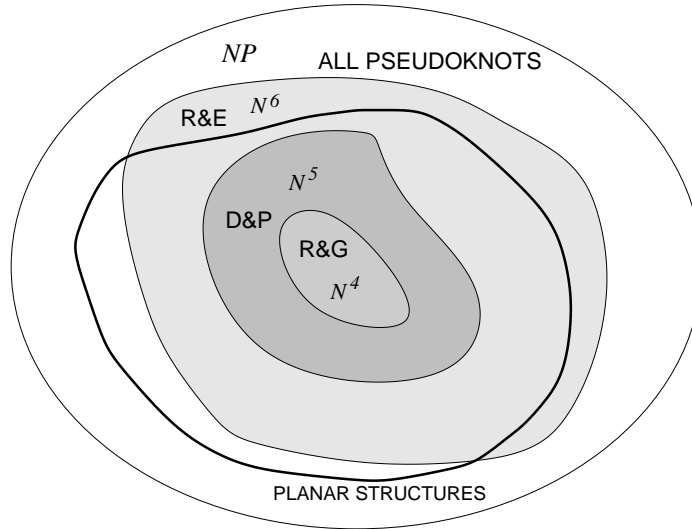


Figure 8. The most prominent classes of pseudoknotted structure are those investigated by Reeder and Giegerich R&G [104], Dirks and Pierce D&P [22], and Rivas and Eddy R&E [110].

In the general case of unrestricted pseudoknots the problem is NP-complete when a loop-based energy model is used [87, 3]. Arbitrarily complex pseudoknots, however, are also biologically unrealistic. While every secondary structure has a plausible 3D realization² this is not true for more general structures: it may well be impossible to embed a given arbitrary set of base pairs in three dimensional space such that chemically reasonable distances are maintained. By construction, such constraints cannot be incorporated into our graph-based model of RNA structure. One remedy is to restrict oneself to certain (simple) classes of pseudoknots.

Fig. 7 shows the algorithmic problem with pseudoknots. In principle one could include (a certain type of) pseudoknots as additional structural elements into the dynamic programming recursion. The further decomposition of the structure, however, requires at least two coupled cut points, here k and l , even if we assume that the structures of crossing arc sets are only single stems. This increases the CPU requirements to at least $\mathcal{O}(n^4)$. More realistic models, such as the H -type model, require additional memory as well.

²This follows directly from the tree structure of secondary-structures.

Algorithms for a number of different classes of pseudoknots have been published in recent years, see e.g. [110, 3, 87, 104, 22]. Fig. 8 summarizes the relationships between the algorithmic complexities of predicting secondary structures from some of these structure classes [16].

3.6. Notes. The basic counting recursion can be readily modified to enumerate other quantities of interest such as the structures with particular properties and distributions of structural elements, see e.g. [55]. The combinatorics of RNA secondary structures and related mathematical objects such as ordered trees, Motzkin paths, and non-crossing partitions is still an active area of research, see e.g. [13, 19, 81] and the references therein.

The recursions for the loop-based energy model as displayed above, in fact, give rise to $\mathcal{O}(n^4)$ CPU requirements due to the interior loop contribution. However, very long interior loops are extremely unlikely (and unstable), so that the length of interior can be bounded by a constant, e.g. $M = 30$. The interior loop contribution thus remains quadratic. Under certain plausible assumptions on the interior loop energies, a cubic time algorithm can be designed [86] that takes interior loops of all sizes into account.

A restriction of the folding algorithm to local structure is described in [52]. Here the maximum span $|j - i + 1|$ of a base pair (i, j) is bounded by a constant L . The resulting “scanning” algorithms are linear in time and space and hence can be used to screen entire genomes for locally stable structures.

Circular RNA molecules are rare, but their secondary structures are of considerable interest because structural features are important e.g. in viroids [123, 108]. A straightforward way of dealing with circular RNA molecules is to compute C_{ij} and M_{ij} also for the subsequences of the form $x[j..n]x[1..i]$ [150]. The disadvantage of this approach is, however, that it doubles the memory requirements. An alternative is described in [60].

A secondary structure Ω is *saturated* if none of its stems can be elongated, i.e., if any single base pair that is inserted into Ω does not stabilize the structure by stacking to any other base pairs. The recursions Fig. 6 can be modified to produce only saturated structure [26]. Similarly, we may call Ω *locally base pair optimal* if Ω cannot be expanded by any additional base pair. In [15] an dynamic programming algorithm is described that computes such locally optimal structures in quartic time with cubic memory requirements.

Prediction of pseudoknotted structures based on maximum matching can be done using algorithms for Maximum Weighted Matching [125]. While this approach requires only $\mathcal{O}(n^3)$ time, it cannot take the loop based energy model into account. “Iterated loop matching”, i.e., the repeated (greedy) application of the Nussinov algorithm, is another approximate way of computing pseudoknotted structures [113]. Finally, heuristics such as genetic algorithms can be used to compute pseudoknotted structures [78].

4. Conserved Structures, Consensus Structures, and RNA Gene Finding

4.1. The Phylogenetic Method. Most functional RNA molecules have characteristic secondary structures that are highly conserved in evolution. Well known examples include rRNAs, tRNAs, RNase P and MRP RNAs, the RNA component of signal recognition particles, tmRNA, group I and group II introns, and small nuclear RNAs. It is therefore of considerable practical interest to efficiently compute the consensus structure of a collection of such RNA molecules.

Given a sufficiently large database of aligned RNA sequences, one can directly infer a consensus secondary structure from the data. The basic idea is that substitutions in the sequence will respect the common structural constraints: Therefore, substitutions in helical regions have to be correlated, since in general only 6 (GC, CG, AU, UA, UG, and GU) out of the 16 combinations of two bases can be incorporated in the helix. Two columns in the alignment thus will co-vary if they form a base pair.

For concreteness, assume that we are given a multiple sequence alignment \mathbb{A} of N sequences. By \mathbb{A}_i we denote the i -th column of the alignment, while a_i^α is the entry in the α -th row of the i -th column. The length of \mathbb{A} , i.e., the number of columns, is n . Furthermore, let $f_i(\mathbf{X})$ be the frequency of base \mathbf{X} at aligned position i and let $f_{ij}(\mathbf{XY})$ be the frequency of finding simultaneously \mathbf{X} at position i and \mathbf{Y} at j .

The most common way of quantifying sequence covariation for the purpose of RNA secondary determination is the *mutual information* score [14, 45, 44]

$$MI_{ij} = \sum_{\mathbf{X}, \mathbf{Y}} f_{ij}(\mathbf{XY}) \log \frac{f_{ij}(\mathbf{XY})}{f_i(\mathbf{X})f_j(\mathbf{Y})} \quad (10)$$

Usually, the mutual information score makes no use of RNA base-pairing rules. For large datasets this is desirable, since it allows to identify non-canonical base pairs and tertiary interaction. For the small datasets considered in the following subsections, however, neglecting base pairing rules does more harm (by increasing noise) than good. In particular, mutual information does not account at all for consistent non-compensatory mutations, i.e., if we have, say, only GC and GU pairs at positions i and j then $MI_{ij} = 0$. Thus sites with two different types of base pairs are treated just like a pair of conserved positions.

A straightforward measure of covariation takes the form

$$C_{ij} = \sum_{\mathbf{XY}, \mathbf{X'Y'}} f_{ij}(\mathbf{XY}) \mathbf{D}_{\mathbf{XY}, \mathbf{X'Y'}} f_{ij}(\mathbf{X'Y'}). \quad (11)$$

where a suitable choice for the 16×16 matrix \mathbf{D} has entries $\mathbf{D}_{\mathbf{XY}, \mathbf{X'Y'}} = d_H(\mathbf{XY}, \mathbf{X'Y'})$ if both $\mathbf{XY} \in \mathcal{B}$ and $\mathbf{X'Y'} \in \mathcal{B}$ and $\mathbf{D}_{\mathbf{XY}, \mathbf{X'Y'}} = 0$ otherwise. Here $\mathcal{B} = \text{GC, CG, AU, UA, GU, UG}$ and $d_H(\mathbf{XY}, \mathbf{X'Y'})$ is the Hamming distance of \mathbf{XY} and $\mathbf{X'Y'}$. The idea here is that consistent mutations such as $\text{GC} \rightarrow \text{GU}$ should count less (here half) of a compensatory mutation such as $\text{GC} \rightarrow \text{AU}$. Note that equ.(11) is a scalar product, $C_{ij} = \langle f_{ij} \mathbf{D} f_{ij} \rangle$, and hence can be evaluated efficiently. If desired, \mathbf{D} could be replaced by a different kernel that e.g. could incorporate measured substitution rates [35].

The purely phylogenetic approach suffers from two limitations: (1) It requires a very large set of sequences in order to obtain a reliable estimate of covariance or mutual information for each pair of sequences. With the exception of rRNAs and tRNAs, such large datasets are usually not (yet) available. (2) It is sensitive to alignment errors, and hence not applicable to very diverse sets of sequences. A possibly remedy is provided by approaches towards solving the folding and alignment problems simultaneously or iteratively. These are discussed in the following section.

4.2. Conserved Structures. The amount of data that is required for inferring structures can be reduced dramatically by taking thermodynamics of folding into account. Indeed, [47] suggested to resolve ambiguities in the phylogenetic analysis based on thermodynamic considerations.

The converse approach, however, namely to use the information which base pairs are thermodynamically plausible, appears to be more efficient. Most of the alignment based methods therefore start from thermodynamics-based folding and use the analysis of sequence covariations or mutual information for post-processing, see e.g., [77, 84, 85, 54, 59, 71]. We describe here the `alidot` algorithm [54, 59].

For each of the aligned sequences secondary structures are computed separately. The resulting lists of base pairs from either minimum free energy calculations or from a partition function calculation are then superimposed by using the multiple sequence alignment to determine which pairs in different sequences are equivalent. For each pair we now have both thermodynamic and sequence covariation information, which is used to hierarchically rank order base pairs depending on their support across the entire data set. A greedy procedure then extracts contiguous stems from the rank ordered list and combines them to a partial secondary structure which contains only those sequence/structure elements that are significantly conserved throughout the aligned input sequences.

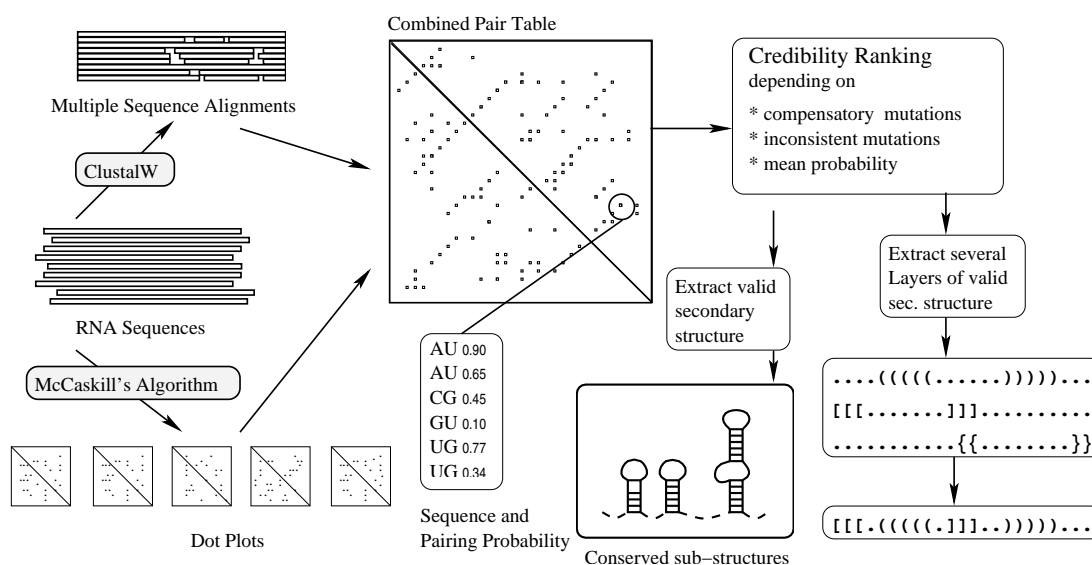


Figure 9. Flow chart of the `alidot` algorithm.

An alternative to the ranking/greedy approach of `alidot` is to compute a score or weight w_{ij} for each possible base pair. The program `ConStruct` [85] uses a simple scoring function that exclusively combines the base pairing probabilities of the individual sequences. Covariance or mutual information score as well as contributions that consider the potential to extend the pair to a longer helix [141] could easily be included. A secondary structure can then be computed using the Nussinov algorithm with weights w_{ij} for the base pairs. The downside of this approach is that it returns a global secondary structure rather than a collection of well-supported local features.

Comparative approaches are based on the fact that RNA secondary structure is quite fragile against randomly placed point mutations. Our earlier computational studies suggest, that even with 85% sequence identity we should expect no significant structural similarity [33, 116]. While this result may seem surprising, there has been convincing experimental evidence, see e.g. [115]. Methods such as `alidot` thus can discriminate very well between conserved and non-conserved RNAs. Both `alidot` and `ConStruct` require interactive work and are therefore best suited for small genomes as found in RNA viruses [142, 131, 56].

4.3. Consensus Structures. Sometimes it is known *a priori* that the aligned sequences should fold into a common secondary structure. This is the case e.g. for rRNAs, tRNAs, and many other small non-coding RNA molecules. In this case it makes sense to ask, what is the most stable structure that can be formed simultaneously by all (or almost all) input sequences. This problem is solved in a rather straightforward way by `RNAalifold` [53]. It treats the entire alignment like a single sequence and solves the secondary structure problem for this “generalized sequence”. To this end, of course, an extension of the standard energy model to alignments is required. `RNAalifold` simply averages the energy contribution over all sequences. In the simple case of base pair dependent energies this means

$$\beta_{ij}^{\mathbb{A}} = \frac{1}{N} \sum_{\alpha} \beta_{x_i^{\alpha}, x_j^{\alpha}} \quad (12)$$

For the realistic energy model, energies for the different loop types are averaged individually.

Both the mutual information score and the covariance score assign a bonus to compensatory mutation. Neither score deals with inconsistent sequences, i.e., with sequences that cannot form a base pair between positions i, j . The simplest ansatz for this purpose is to simply count the number of sequences q_{ij} that cannot form a canonical base pair between columns i and j . Here, combinations of a nucleotide and a gap are counted as inconsistent while gap-gap combinations (i.e. deletions of an entire base pair) are ignored.

In a multiple alignment of a larger number of sequences we have to expect occasional sequencing errors and of course there will be alignment errors. Thus we cannot simply mark a pair of positions as non-pairing if a single sequence is inconsistent. Furthermore, there is the possibility of a non-standard base pair [44]. Thus we define a threshold value for the combined score $B_{ij} = C_{ij} - \phi_1 q_{ij}$ and declare a pair of positions i, j as non-pairing if B_{ij} is too small.

Figure 10 shows the consensus structure of the *mir-105* microRNA family as an example. Such consensus structures are needed for the derivation of pattern descriptions that can be used to search for structurally similar RNAs in genomic DNA, as briefly described in the following section.

4.4. RNA Gene Finding. It is, of course, possible to identify genomic sequences that are homologous to known RNA genes, using either BLASTN or, as in the case of tRNAs, more specialized methods. For most functional noncoding RNA (ncRNA) molecules the secondary structure is much more conserved than their sequence. This can be used to identify putative noncoding RNA sequences using programs such as RNAmot [34], tRNAscan [83], or HyPa [40]. Nevertheless, all these approaches are restricted to searching for new members of the few well-established families such as tRNAs, snoRNAs, microRNAs, and certain spliceosomal RNAs.

A different approach is taken in the program QRNA [111]. This method for comparative analysis of two aligned homologous sequences can detect novel structural RNA genes by deciding whether the substitution pattern fits better with (a) synonymous substitutions, which are expected in protein-coding regions, (b) the compensatory

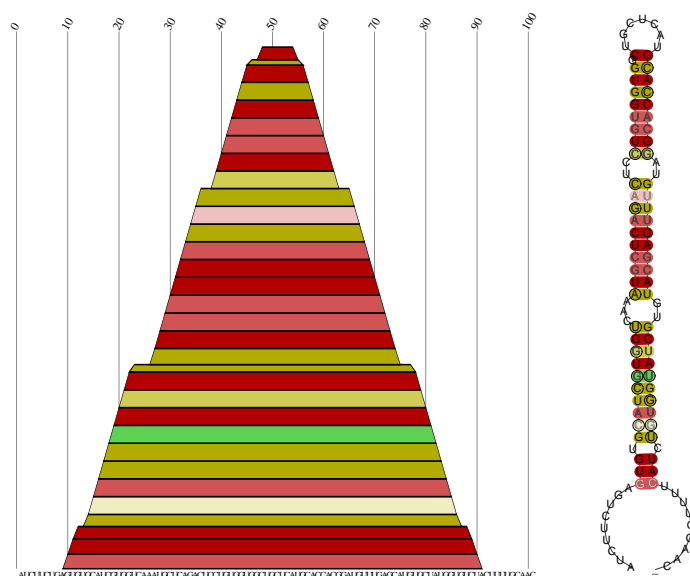


Figure 10. Consensus secondary structure of the 11 sequences from mammalian microRNA *mir-105*. Sequence are taken from the *microRNA Registry* (version 6.0) and from *blast* searches in vertebrate genomes.

L.h.s.: *Mountain plot*: A base pair (i, j) is represented by a slab ranging from i to j . The 5' and 3' sides of stems thus appear as up-hill and down-hill slopes, respectively, while plateaus indicate unpaired regions. Colors indicate sequence variation by encoding the number of different types of base pairs (GC, CG, AU, UA, GU, UG) that occur in the two paired columns of the alignment. Pairs with conserved sequence are shown in red; ocher, green, cyan, blue, violet indicate two to six types of base pairs. Pairs with one or two inconsistent mutations are shown in (two degrees of) pale colors.

R.h.s.: In the *conventional secondary structure graph* paired positions are color coded as in the mountain plot. Consistent mutations are indicated by circles around the varying position, compensatory mutations thus are marked by circles around both pairing partners.

mutations consistent with some base paired secondary structure, or (c) uncorrelated mutations.

The **alidot** approach has never been used for large scale gene finding since it has turned out to be non-trivial to assign statistical significance values to its results. Most recently, however, a conceptually related technique has been developed that is efficient and sensitive enough to allow genome-wide screens for RNAs.

The program **RNAz** [136] combines a comparative approach (scoring conservation of secondary structure) with the observation [76, 138, 8] that ncRNAs are thermodynamically more stable than expected by chance. This excess stability is conveniently measured in terms of the z -score

$$z = \frac{E - \overline{E}}{\sigma} \quad (13)$$

where \overline{E} and σ are mean and standard deviation of the distribution of shuffled sequences. Instead of dealing with individual sequences, **RNAz** uses multiple sequence alignments of potential RNAs from different species as input. The computation of the z by direct sampling is extremely time-consuming. In **RNAz** it is therefore replaced by a support vector machine that has been trained to solve the regression problems of estimating \overline{E} and σ from properties of the input sequences.

Structural conservation is also quantified in thermodynamical terms: The structure conservation index S is defined as the ratio of the average energy of the consensus structure (as computed by **RNAalifold**) and the average of the unconstrained folding energies of the individual sequences. An alignment of identical sequences thus has $S = 1$. On the other hand, completely unrelated sequences will not be able to form a consensus structure since there are always some sequences that contradict any particular pairing, thus $S = 0$. Sequences that form a well-conserved consensus sequence in the presence of sequence covariations, finally, will have the same energy contributions in the consensus and in the individual folds. In addition, however, the consensus energy contains the bonus contributions for sequence covariations, so that we obtain $S > 1$.

RNAz uses a support vector machine (SVM) [17] to determine from the z -score and the structure conservation index whether a given multiple sequence alignment is a structurally conserved RNA. Surveys of animal genomes [137, 96] reveal a very large number of previously unknown candidates for both independent ncRNAs and structured cis-acting elements in mRNAs.

4.5. Notes. Including covariation information is also a good way to improve the accuracy of structure predictions including pseudoknots. One approach is to forgo a loop-based energy model and use base pair scores instead, in which case the resulting Maximum Weighted Matching problem can be solved efficiently [125]. Good accuracies can be achieved by using a combination of covariance and thermodynamic criteria for scoring potential base pairs [141]. The **ILM** program of Ruan *et al.* [113], uses the Nussinov algorithm iteratively in order to build pseudoknotted structures.

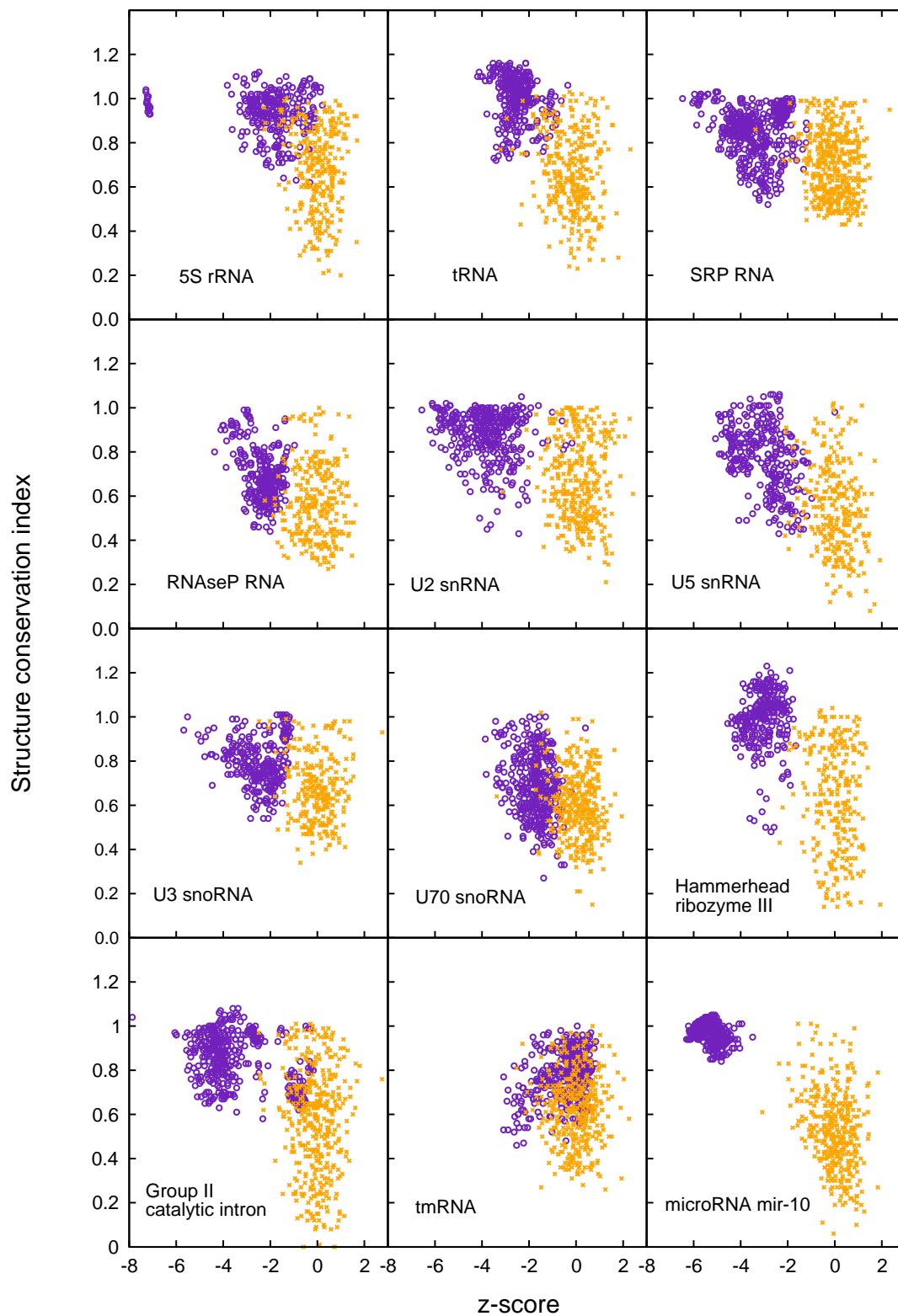


Figure 11. Scatter plot of structure conservation index S (x-axis) and energy z -score (y-axis) for different families of structured non-coding RNAs. In each panel, the properties of the true sequences (dark) are compared with controls obtained by shuffling the sequence. Data are taken from [136].

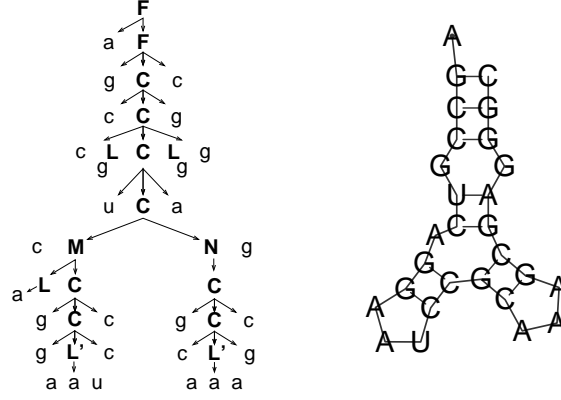


Figure 12. Parse tree and secondary structure drawing for a small example structure, using the grammar from equ.(15). Productions of the form $L \rightarrow \emptyset$ are left out for simplicity.

5. Grammars for RNA Structures

5.1. Context Free Grammars and RNA Secondary Structures. The recursions for RNA folding in Fig. 6 suggest a close connection with certain grammars. More precisely, we may interpret Fig. 6 as the production rules of an “RNA language”. The tree representations in Fig. 4, on the other hand are suggestive of a connection between RNA structures and parse trees of a grammar that generates RNA sequence. As we shall see in this section, these connections can be made precise and open the door to the application of learning techniques in RNA bioinformatics.

Recall that a formal language \mathcal{L} is a set of strings over a given alphabet \mathcal{A} . A *grammar* G for the language \mathcal{L} consists of

- a set T of *terminals* which are the letters of the alphabet \mathcal{A} possibly augmented by the null-character ϵ .
- a set N of *non-terminals* which represent the syntactic categories of \mathcal{L}
- a set P of *production* or *derivation rules* which are used to derive the strings in \mathcal{L} . Each production consists of a non-terminal “head” that is produced and a string of zero or more non-terminal and terminals (the “body” of the production)
- a single non-terminal $S \in N$ that is designated as the start symbol.

The “dot-parenthesis” grammar for RNA, in the simplest case, can be written as $G_0 = (T, N, P, s)$ with $T = \{ (,), ., \emptyset \}$, $N = \{ S \}$, $s = S$, and

$$P = \{ S \rightarrow S., S \rightarrow (S)S, S \rightarrow \emptyset \}, \quad (14)$$

where \emptyset denotes the empty string. The grammar above is *context-free* since all productions are of the form $V \rightarrow w$, where V is a non-terminal and w is a string consisting of terminals and/or non-terminals. The grammar generates strings of dots and balanced parenthesis, the parse trees of this grammar correspond to the secondary structures. More elaborate grammars can be designed that explicitly encode different types of loops or other substructures. In particular, the decompositions of the structure sets in section 3.3 can be recast in terms of a grammar:

$$\begin{aligned}
F &\rightarrow uF|CF|\emptyset \\
C &\rightarrow pL'\bar{p} \mid pLCL\bar{p} \mid pMN\bar{p} \\
M &\rightarrow LC \mid MC \mid Mu \\
N &\rightarrow Nu \mid C \\
L' &\rightarrow uuuL \\
L &\rightarrow uL \mid \emptyset
\end{aligned} \tag{15}$$

This grammar generates RNA sequences, while again the parse trees correspond to secondary structures. The terminal u denotes an unpaired base, while p and \bar{p} is a shorthand for one of the six pairing combinations of bases. The start symbol F represents any structure, L stands for an unpaired sequence within a loop, M and N The production for L' enforces the minimum length of a hairpin loop.

Chomsky normal forms have only productions of the form $V \rightarrow XY$ and $V \rightarrow a$ with $V, X, Y \in N$ and $a \in T$. One can show that every context-free grammar can be converted to normal form, i.e., there is a CFG in normal form that produces the same language \mathcal{L} .

Given a context-free grammar $G = (T, N, P, s)$ we obtain a *stochastic context-free grammar* (SCFG) by assigning probabilities $\mathbb{P}(\alpha)$ to all productions $\alpha \in P$ such that $\sum_{\alpha \in P} \mathbb{P}(\alpha) = 1$ is satisfied.

The probabilities associated with the individual productions take on the role of the energy parameters in the previous sections. While the energy parameters must be measured directly, the values of $\mathbb{P}(\alpha)$ can be inferred from *training sets* of known sequence/structure pairs in a generic machine learning setting. Thus they can, at least in principle, readily combine different sources of information that can be expressed probabilistically, such as an evolutionary model (derived from a comparative analysis of RNA sequences) and a biophysically motivated model of structure plausibility. In the following three subsection we briefly outline the basic techniques: finding the most likely parse-tree, computing the probability of a given word, and the estimation of production probabilities from a given dataset. None of these algorithms is RNA specific; they rather apply to any SCFG in Chomsky normal form.

5.2. CYK Algorithm. The analog of the minimum free energy folding problem in the SCFG setting can be phrased in the following way: Given a string $x \in \mathcal{L}$, find the most likely parse tree for x in a grammar G .

Under the assumption that G is in Chomsky normal form, there is an efficient (polynomial-time) solution to this question the Cocke-Younger-Kasami (CYK) algorithm [146]:

Let $w(i, j, V)$ denote the likelihood of the most likely parse tree on the substring $x[i..j]$ rooted at the nonterminal V . Clearly, we have $w(i, i, V) = \log \mathbb{P}(V \rightarrow x_i)$ for all i and V . For all larger substrings, $j > i$, we try all productions of the form $V \rightarrow XY$ and select the one that maximizes the likelihood. This immediately leads to the recursion

$$w(i, j, V) = \max_X \max_Y \max_{i \leq k < j} (\log \mathbb{P}(V \rightarrow XY) + w(i, k, X) + w(k + 1, j, Y)) \tag{16}$$

with the initialization $w(i, i, V) = \log \mathbb{P}(V \rightarrow x_i)$. The same type of backtracing approach as in the Nussinov algorithm can be used to explicitly recover the parse tree, which corresponds to the secondary structure of the RNA molecule.

5.3. Inside and Outside Algorithms. Instead of retrieving the most likely parse tree one may instead be interested in the probabilities of generating substrings in a particular way. In particular, let $p(i, j, V)$ be the probability that the “inside” substring $x[i..j]$ is generated by the non-terminal V . Furthermore, let $q(i, j, V)$ be the probability that the “outside” substrings $x[1..i-1] \cup x[j+1..n]$ are generated from the start symbol S under the condition that (the parse sub-tree of) the subsequence $x[i..j]$ is rooted at V . Conceptually, these quantities correspond to the partition functions inside and outside of a subsequence $x[i..j]$. It is straightforward to derive the corresponding *inside recursion*:

$$p(i, j, V) = \sum_X \sum_Y \sum_{k=i}^j \mathbb{P}(V \rightarrow XY) p(i, k, X) p(k+1, j, Y) \quad (17)$$

which is initialized with $p(i, i, V) = \mathbb{P}(V \rightarrow x_i)$. The *outside recursion* consists of two parts, depending on whether the root V of the interior parse tree is the right or the left non-terminal in the previous production. This yields:

$$\begin{aligned} q(i, j, V) = & \sum_X \sum_Y \sum_{k < i} \mathbb{P}(Y \rightarrow XV) p(k, i-1, X) q(k, j, Y) + \\ & \sum_X \sum_Y \sum_{k > j} \mathbb{P}(Y \rightarrow VX) q(i, k, Y) p(j+1, k, X) \end{aligned} \quad (18)$$

with the initial conditions $q(1, n, S) = 1$ and $q(1, n, X) = 0$ for all $X \in N \setminus \{S\}$. The probability to produce the sequence x is

$$\mathbb{P}(x) = p(1, n, S) = \sum_X q(i, i, X) \mathbb{P}(X \rightarrow x_i) \quad (19)$$

5.4. Parameter Estimation. One problem with SCFG approaches is that the production probabilities have to be estimated from data. To this end, we compute the expected number $c(V)$ that V is used to parse x and the expected numbers $c(\alpha)$ that production α is used in the derivation of x . It is straightforward to derive

$$\begin{aligned} c(V) &= \frac{1}{\mathbb{P}(x)} \sum_{i,j=1}^n p(i, j, V) q(i, j, V) \\ c(V \rightarrow a) &= \frac{1}{\mathbb{P}(x)} \sum_{i:x_i=a} q(i, i, V) \mathbb{P}(V \rightarrow a) \\ c(V \rightarrow XY) &= \frac{1}{\mathbb{P}(x)} \sum_{i,j=1}^n \sum_{k=i}^j q(i, j, V) p(i, k, X) p(k+1, j, Y) \mathbb{P}(V \rightarrow XY) \end{aligned} \quad (20)$$

Updated estimates for the production probabilities can thus be obtained as $\mathbb{P}'(\alpha) = c(\alpha)/c(V)$ for all $\alpha \in P$. The procedure is then repeated until $\sum_{\alpha} |\mathbb{P}'(\alpha) - \mathbb{P}(\alpha)| < \varepsilon$, where ε is a user-defined accuracy.

5.5. Algebraic Dynamic Programming. Algebraic Dynamic Programming (ADP) [37] was introduced to facilitate and systematize the development of dynamic programming algorithms. Conceptually, a DP algorithm consists of three components: a search space of candidate solutions (in our case RNA secondary structures), a scoring scheme (free energies, partition functions, etc.), and an objective function (minimize [energy], sum up [Boltzmann factors]). The idea behind ADP is to separate these three aspects. For a comprehensive discussion of ADP in the context of bioinformatics we refer to [37]. We can give here only a very brief, qualitative sketch of the topic.

The search space is defined by a *yield grammar*, i.e., a tree grammar that generates a string language by mapping its terminal symbols at the leaves of the tree into sequences of symbols. A tree grammar is similar to a context-free grammar, with terminal and non-terminal symbols, and productions where the right hand sides are trees (formulas) from some underlying term algebra. Intuitively, first the search space is “constructed” by enumerating all candidate solutions. This is a parsing problem for which standard solutions, so-called tabulating yield parsers, exist. Scoring and choice are described in terms of an *evaluation algebra* which is independent of the details of the search space.

The main advantage is that complex variants of folding problems can be implemented very easily: It suffices to modify the grammar to restrict the dynamic programming recursions to all canonical secondary structures, i.e., those that have no isolated base pairs. Conversely, the evaluation algebra can be changed easily. Once the energy model is implemented, one can change the choice function from minimizing energies to adding up Boltzmann factors or listing all structures within an energy range.

The restrictions of the search space can be quite dramatic: One can, for example restrict oneself to *saturated* secondary structures, which consist solely of maximally extended stacking regions, i.e. no adjacent single stranded nucleotides exist that could form a base pair and stack on top of a helix [26]. A particularly interesting application of the ADP framework is *RNAshapes* [36] which can be used to systematically generate (sub)optimal RNA structures belonging to distinct coarse-grained structural classes. For example, one can search for the most stable clover-leaf shaped secondary structure that can be formed by the input sequence.

5.6. Notes. Due to space restrictions we only gave a brief sketch of the SCFG approach to RNA secondary structures. A variety of implementations of SCFG-based algorithms are available for different purposes: *pfold* [75, 74] as an SCFG-approach to “folding an alignment” similar in spirit to the thermodynamics-based *RNAalifold*.

A general approach to computing suboptimal parse trees, similar in spirit to the back-tracing of RNA secondary structures with suboptimal energies, is described in [70]. A systematic comparison of several alternative grammar models for RNA secondary structures showed that the actual performance of SCFGs can depend considerably on the details of the grammar being used [23].

A practical problem for the application of SCFGs is that one needs a grammar that is both unambiguous and in Chomsky normal form. The decomposition of Fig. 6,

for example, does not satisfy this requirement, because the last case in the second line, for example, requires non-terminals for the closing base pair as well as for the two enclosed multiloop components. Without discussing the details here, this creates problems in particular with the multiloop decomposition.

Sean Eddy’s *Infernal* [25] creates a covariance model from local alignments and can be used to search a sequence database for sequences that are likely to be produced from this SCFG. *Rsearch* [73] aligns an RNA query to target sequences, using SCFG algorithms to score both secondary structure and primary sequence alignment simultaneously.

So-called pair SCFGs can be used to solve the combined folding and alignment problem in analogy to Sankoff’s algorithm described in the next section, see e.g. [64]. The *QRNA* program [112] uses a pair SCFG to compute the probability that the substitution pattern in a pairwise alignment is derived from RNA secondary structure conservation. It has been used successfully to predict ncRNA candidates in *E. coli* and *S. cerevisiae* [109, 94]. Most recently, Pedersen *et al.* [103, 102] devised an SCFG-based algorithm for detecting conserved secondary structure motifs specifically within coding sequences. An SCFG-like approach to pseudoknotted structures can be found in [11].

6. Comparison of Secondary Structures

Many classes of functional RNA molecules, including tRNAs, rRNAs, and many other “classical” ncRNAs, are characterized by highly conserved secondary structures but little detectable sequence similarity. Reliable multiple alignments can therefore be constructed only when the shared structural features are taken into account. Since multiple alignments are used as input for many subsequent methods of data analysis, structure based alignments are an indispensable necessity in RNA bioinformatics. This problem is far from being solved in a satisfactory way, both because the available approaches are computationally expensive, and because little is known about the evolution of RNA at the structural level, and hence on the appropriate edit cost parameters.

6.1. String-Based Alignments. The problem of comparing two structures Ψ_1 and Ψ_2 of the *same* RNA molecule is trivial: Since a secondary structure is simply a set of base pairs one may use for example the size of symmetric difference between the two sets $|\Psi_1 \Delta \Psi_2|$ as a distance measure that is obviously a metric. In other words, we simply count the number of base pairs that occur in one of the structures but not in both,

The question immediately becomes non-trivial, however, if we do not assume that the two structures have the same underlying sequence length, i.e., if we do not know *a priori* which sequence positions in the two molecules correspond to each other.

As we have seen, RNA secondary structures can be faithfully represented as strings over the alphabet $\{(,), .\}$. Clearly, we can use this string representation to compute a metric on secondary structures by means of standard sequence alignment methods, e.g. using the Needleman-Wunsch algorithm [99].

This approach can be generalized to a comparison of base pair probability matrices [7]. From the pairing probabilities of base i we construct a vector containing the probabilities of being paired upstream $p^<(i) = \sum_{j>i} P_{ij}$, downstream $p^>(i) = \sum_{j<i} P_{ji}$, or unpaired $p^\circ(i) = 1 - p^<(i) - p^>(i)$. The resulting profiles can be aligned by means of a standard string/profile alignment algorithm in $\mathcal{O}(n^2)$ time using

$$\rho = \sqrt{p_A^>p_B^>} + \sqrt{p_A^<p_B^<} + \sqrt{p_A^\circ p_B^\circ} \quad (21)$$

as the match score (or $1 - \rho$ as an edit cost). While this approach of “*string-like alignments*” is fast, it often produces misaligned pairs:

Sequence alignment	Structure alignment
CAGUCUCAGGUGGUUGGGCU-	CAGUCUCAGGUGGUUG-GGCU
.((((.((((....)))))))-	.((((.((((....)))--)))
UAG-CUGAGGUG-UCGUGCUA	-UAGC-UGAGGUGUCGUGCUA
(((-((((....-))))).)))	-(((((-((((....)))..)))

Figure 13. Sequence vs. structure alignment. Compared to the structural alignment (right), the sequence alignment (from `ClustalW`) mis-aligns 5 of the 7 base pairs.

6.2. Tree Editing. The string-based alignments above essentially use only the information whether a nucleotide is paired or unpaired, but neglect the connectivity information who pairs with whom. This limitation can be overcome by methods based on the tree representation of secondary structures. Of particular interest are tree editing and the related tree alignment, since they are still fast enough to be applicable to genome wide surveys. We present these approaches in detail here since there does not appear to be a good textbook exposition of this topic.

The three most natural operations (“moves”) that can be used to convert ordered trees (and, more generally, ordered forests) into each other are depicted in Fig. 14:

- (1) **Substitution** ($x \rightarrow y$) consists of replacing a single vertex label x by another vertex label y .
- (2) **Insertion** ($\emptyset \rightarrow z$) consists of adding a vertex z as a child of x , thereby making z the parent of a consecutive subsequence of children of x . A node z can also be inserted at the “top level”, thereby becoming the root of a tree.
- (3) **Deletion** ($z \rightarrow \emptyset$) consists of removing a vertex z , its children thereby become to children of the parent x of z . Removing the root of a tree produces a forest in which the children of z become roots of trees.

Naturally, we associate a *cost* with each edit operation, which we will denote by $\gamma(x \rightarrow y)$, $\gamma(\emptyset \rightarrow z)$, and $\gamma(z \rightarrow \emptyset)$ for substitutions, insertions, and deletions, respectively. We assume that γ is a metric on the extended alphabet $\mathcal{A} \cup \{\emptyset\}$. By using an appropriate alphabet of vertex labels, one can easily include sequence information in the cost function.

A sequence of moves that transforms a forest F_1 into a forest F_2 is known as an *edit script*. Its cost is the sum of the costs of edit operations in the script.

A *mapping* from F_1 to F_2 is a binary relation $M \in V(F_1) \times V(F_2)$ between the vertex sets of the two forests such that for pairs $(x, y), (x', y') \in M$ holds

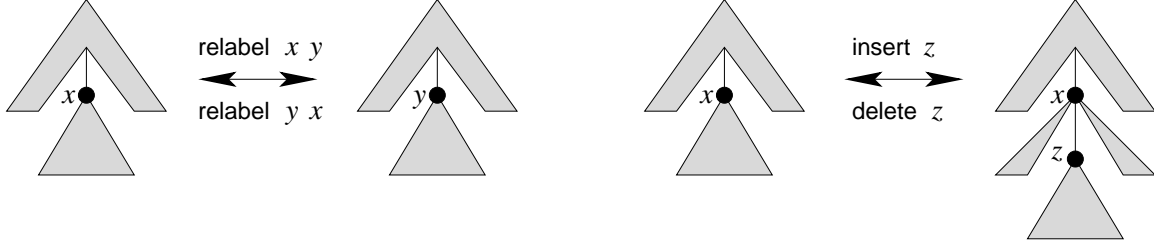


Figure 14. Elementary operations in tree editing

- (1) $x = x'$ if and only if $y = y'$. (one-to-one condition)
- (2) x is an ancestor of x' if and only if y is an ancestor of y' . (ancestor condition)
- (3) x is to the left of x' if and only if y is to the left of y' . (sibling condition)

By definition, for each $x \in F_1$ there is a unique “partner” in $y \in F_2$ such that $(x, y) \in M$ or there is no partner at all. In the latter case we write $x \in M'_1$. Analogously, we write $y \in M'_2$ if $y \in F_2$ does not have a partner in F_1 . With each mapping we can associate the cost

$$\gamma(M) = \sum_{(x,y) \in M} \gamma(x \rightarrow y) + \sum_{y \in M'_2} \gamma(\emptyset \rightarrow y) + \sum_{x \in M'_1} \gamma(x \rightarrow \emptyset) \quad (22)$$

Clearly, each edit operation gives rise to a corresponding mapping between the initial and the final tree. In the case of a substitution, all vertices have partners, in the case of insertion and deletion there is exactly one vertex without partner.

Mappings are relations, and hence they can be composed in a natural way: Consider three forests F_1 , F_2 , and F_3 and mappings M_1 from F_1 to F_2 and M_2 from F_2 to F_3 . Then

$$M_1 \circ M_2 = \{(x, z) \mid \exists y \in V(F_2) \text{ such that } (x, y) \in M_1 \text{ and } (y, z) \in M_2\} \quad (23)$$

is a mapping from F_1 to F_3 . It is easy to convince oneself that the cost function defined in (22) is subadditive under composition, $\gamma(M_1 \circ M_2) \leq \gamma(M_1) + \gamma(M_2)$. Using this result and the fact that every mapping can be obtained as a composition of edit operations one can show that the minimum cost mapping is equivalent to the minimum cost edit script [128].

For a given forest F we note by $F - x$ the forest obtained by deleting x and $F \setminus T(x)$ is the forest obtained from F by deleting with x all descendants of x . Note that $T(x) - x$ is the forest consisting of all trees whose roots are the children of x .

Now consider two forests F_1 and F_2 and let v_i be the root of the right-most tree in F_i , $i = 1, 2$ and an optimal mapping M . Apart from the trivial cases, in which one of the two forests is empty, we have to distinguish three cases: (i) v_2 has no partner in the optimal mapping. In this case v_2 is inserted and the optimal mapping consists of an optimal mapping from F_1 to $F_2 - v_2$ composed with the insertion of v_2 . (ii) v_1 has no partner. This corresponds to the deletion of v_1 . (iii) both v_1 and v_2 have partners. In this case $(v_1, v_2) \in M$.

To see this, one can argue as follows: Suppose $(v_1, h) \in M$, $h \neq v_2$ and $(k, v_2) \in M$. By the one-to-one condition, $k \neq v_1$. By the sibling condition, if v_1 is to the right of k then h must

be to the right of v_2 . If v_1 is a proper ancestor of k then h must be a proper ancestor of v_2 by the ancestor condition. Both cases are impossible, however, since both v_1 and v_2 are by construction rightmost roots.

For each of the three cases it is now straightforward to recursively compute the optimal cost of M . We arrive directly at the dynamic programming recursion

$$D(F_1, F_2) = \min \begin{cases} D(F_1 - v_1, F_2) + \gamma(v_1 \rightarrow \emptyset) \\ D(F_1, F_2 - v_2) + \gamma(\emptyset \rightarrow v_2) \\ D(T(v_1) - v_1, T(v_2) - v_2) + D(F_1 \setminus T(v_1), F_2 \setminus T(v_2)) + \gamma(v_1 \rightarrow v_2) \end{cases} \quad (24)$$

which allows us to compute the tree edit distance $D(F_1, F_2)$ from smaller sub-problems. The initialization is the distance between $D(\emptyset, \emptyset) = 0$ of two empty forests. In the cases where one of the two forests is empty, equ.(24) reduces to $D(\emptyset, F_2) = D(\emptyset, F_2 - v_2) + \gamma(\emptyset \rightarrow v_2)$, and $D(F_1, \emptyset) = D(F_1 - v_1, \emptyset) + \gamma(v_1 \rightarrow \emptyset)$.

One can show that the time complexity of this algorithm is bounded by $\mathcal{O}(|F_1|^2 |F_2|^2)$. Various more efficient implementations exist, see in particular [147, 72]. A detailed performance analysis of the algorithm by Zhang & Shasha [147] is given in [24].

A common feature of all tree representations discussed above is that each subtree $T(x)$ rooted at a vertex x corresponds to an interval I_x of the underlying RNA sequence. We can thus regard every pair (v_1, v_2) as a prescription to match up the intervals I_{v_1} with I_{v_2} between the two input sequences. In particular, if v_1 and v_2 are leaves in the forests F_1 and F_2 , then they correspond to individual bases. Interior nodes serve as delimiters of intervals in Giegerich's encoding, while they correspond to base pairs in the encoding used in the **Vienna RNA Package**. In either case, one can derive all (mis)matches directly from M . The sibling and ancestor properties of M guarantee that (mis)matches preserve the order in which they appear on the RNA sequence. All other nucleotides, i.e., those that correspond to vertices $v_1 \in M'_1$ and $v_2 \in M'_2$ are deleted or inserted, respectively, in the appropriate positions. Every mapping M therefore implies a (canonical) pairwise alignment $A(M)$ of the underlying sequences.

6.3. Tree Alignments. An alternative way of defining the difference of two forests are *tree alignments* [69]. Consider a forest G with vertex labels taken from $(\mathcal{A} \cup \{-\}) \times (\mathcal{A} \cup \{-\})$. Then we obtain restrictions $\pi_1(G)$ and $\pi_2(G)$ by considering only the first or the second coordinate of the labels, respectively, and by then deleting all nodes that are labeled with the gap character $-$, see Fig. 15. We say that G is an alignment of the two forests F_1 and F_2 if $F_1 = \pi_1(G)$ and $F_2 = \pi_2(G)$. Naturally, we score the alignment G by adding up the costs of the label pairs:

$$\gamma(G) = \sum_{(v_1, v_2) \in G} \gamma(v_1 \rightarrow v_2) \quad (25)$$

where a pair $(v_1, -)$ corresponds to the edit operation $(v_1 \rightarrow \emptyset)$. On the other hand, an alignment G defines a mapping M_G from F_1 to F_2 by setting $(v_1, v_2) \in M_G$ iff $(v_1, v_2) \in G$ and neither v_1 nor v_2 is a gap character. One easily verifies that the three defining properties of mapping are satisfied. Furthermore, it follows that $\gamma(M_G) = \gamma(G)$ as pair of the form $(v_1, -)$ and $(-, v_2)$ corresponds to deletion and insertion

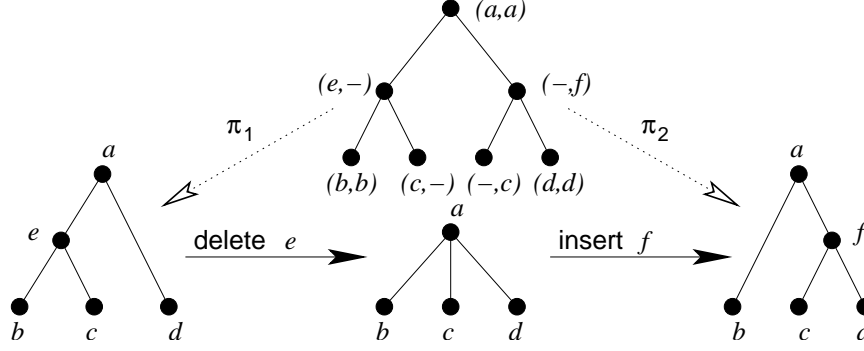


Figure 15. Alignment of two forests F_1 and F_2 and a mapping from F_1 to F_2 that cannot be derived from an alignment.

operations, respectively. Note that, in the special case of two a totally disconnected forests, the problem reduces to ordinary sequence alignment with additive gap costs.

However, as the example in Fig. 15 shows, not all mappings derive from alignments. It follows, therefore, that the minimum cost alignment is more costly than the minimum cost edit script, in general.

In order to compute the optimal alignment, let us first investigate the decomposition of an alignment at a particular (mis)match (v_1, v_2) or in/del $(-, v_2)$ or $(v_1, -)$. We will need a bit of notation, see Fig. 16. Let F be an ordered forest. By $i : F$ we denote the subforest consisting of the first i trees, while $F : j$ denotes the subforest starting with the $j+1$ -th tree. By F^\downarrow we denote forest consisting of the children-trees of the root $v = r_F$ of the first tree in F . $F^\rightarrow = F : 1$ is the forest of the right siblings trees of F .

Now consider an alignment A of two forests F_1 and F_2 . Let $a = r_A$ be the root of its first tree. We have either:

- (1) $a = (v_1, v_2)$. Then $v_1 = r_{F_1}$ and $v_2 = r_{F_2}$; A^\downarrow is an alignment of F_1^\downarrow and F_2^\downarrow ; A^\rightarrow is an alignment of F_1^\rightarrow and F_2^\rightarrow .
- (2) $a = (-, v_2)$. Then $v_2 = r_{F_2}$; for some k , A^\downarrow is an alignment of $k : F_1$ and F_2^\downarrow and A^\rightarrow is an alignment of $F_1 : k$ with F_2^\rightarrow .
- (3) $a = (v_1, -)$. Then $v_1 = r_{F_1}$; for some k , A^\downarrow is an alignment of F_1^\downarrow and $k : F_2$ and A^\rightarrow is an alignment of F_1^\rightarrow with $F_2 : k$.

See Fig. 16 for a graphical representation.

Let $S(F_1, F_2)$ be the optimal score of an alignment of the forests F_1 and F_2 . For easier comparison with the tree-editing algorithm in the previous section we formulate the problem here as a minimization problem. One can, however, just as well maximize appropriate similarity scores. The three cases discussed above and in Fig. 16 imply the following dynamic programming recursion:

$$S(F_1, F_2) = \min \begin{cases} S(F_1^\downarrow, F_2^\downarrow) + S(F_1^\rightarrow, F_2^\rightarrow) + \gamma(v_1 \rightarrow v_2) \\ \min_k S(k : F_1, F_2^\downarrow) + S(F_1 : k, F_2^\rightarrow) + \gamma(\emptyset \rightarrow r_{F_2}) \\ \min_k S(F_1^\downarrow, k : F_2) + S(F_1^\rightarrow, F_2 : k) + \gamma(r_{F_1} \rightarrow \emptyset) \end{cases} \quad (26)$$

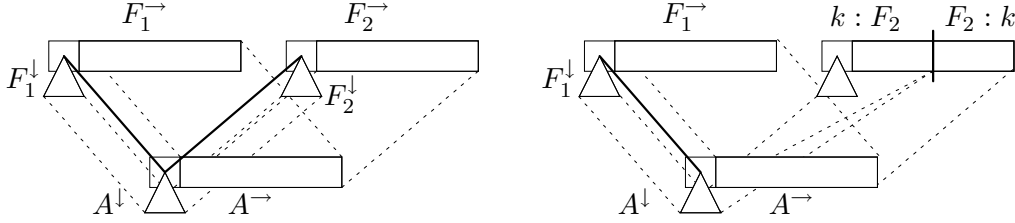


Figure 16. Decomposition of tree alignments. L.h.s.: In the match case the subtrees F_1^\downarrow and F_2^\downarrow are aligned to form A^\downarrow and correspondingly, the sibling subforests F_1^\rightarrow and F_2^\rightarrow must be aligned to yield A^\rightarrow . R.h.s.: in the deletion case the subforest $F_1^\downarrow - v_1$ must be aligned with a part $k : F_2$ of the second forest. F_1^\rightarrow then must be aligned with remainder of $F_2 : k$ of the top-level trees of F_2 . The insertion case is analogous to the deletion case, with the roles of F_1 and F_2 exchanged.

In the special cases where one of the forests is empty this reduces to $S(\emptyset, F_2) = S(\emptyset, F_2^\downarrow) + S(\emptyset, F_2^\rightarrow) + \gamma(\emptyset \rightarrow r_{F_2})$ for the insertion case, and $S(F_1, \emptyset) = S(F_1^\downarrow, \emptyset) + S(F_1^\rightarrow, \emptyset) + \gamma(r_{F_1} \rightarrow \emptyset)$ for the deletion case. The initial condition is again $S(\emptyset, \emptyset) = 0$.

In order to estimate the resource requirements for this algorithm, we observe that we have to consider only those sub-forests of F_1 and F_2 that consist of trees rooted at an uninterrupted interval of sibling nodes. These forests have been termed *closed sub-forests* in [51]. If d_i is the maximum of the number of trees and the numbers of children of the nodes in F_i , we see that there are at most $\mathcal{O}(d_i^2)$ closed subforests at each node and hence at most $\mathcal{O}(|F_1| |F_2| d_1^2 d_2^2)$ entries $S(F_1, F_2)$ need to be computed, each of which requires $\mathcal{O}(d_1 + d_2)$ operations, i.e., tree alignments can be computed in polynomial time. A compact, memory-efficient encoding of the subforests is described in detail in [51], where a careful analysis shows that pairwise tree alignments can be computed in $\mathcal{O}(|F_1| d_1 |F_2| d_2)$ space and $\mathcal{O}(|F_1| |F_2| d_1 d_2 (d_1 + d_2))$ time.

6.4. The Sankoff Algorithm and Its Variants. David Sankoff described an algorithm that simultaneously allows the solution of the structure prediction and the sequence alignment problem [114]. The basic idea is to search for a maximal secondary structure that is common to two RNA sequences. Given a score $\sigma_{ij,kl}$ for the alignment of the base pairs (i, j) and (k, l) from the two sequences (as well as gap penalties γ and scores α_{ik} for matches of unpaired positions) we compute the optimal alignment recursively from alignments of the subsequences $x[i..j]$ and $y[k..l]$. Let $S_{ij,kl}$ be the score of the optimal alignment of these fragments. We have

$$S_{ij,kl} = \max \left\{ S_{i+1,j;kl} + \gamma, S_{ij;k+1,l} + \gamma, S_{i+1,j;k+1,l} + \alpha_{ik}, \right. \\ \left. \max_{(p,q) \text{ paired}} \{ S_{i+1,p-1;k+1,q-1} + \sigma_{ij,pq} + S_{p+1,j;q+1,l} \} \right\} \quad (27)$$

Backtracing is just as easy as in the RNA folding case. Only now π is a partial alignment of two structures and we insert aligned positions instead of positions in individual structures. More precisely we have to insert individual columns or pairs of columns of the form

$$\pi \blacktriangleleft \begin{pmatrix} i. \\ - \end{pmatrix} \quad \pi \blacktriangleleft \begin{pmatrix} - \\ j. \end{pmatrix} \quad \pi \blacktriangleleft \begin{pmatrix} i. \\ j. \end{pmatrix} \quad \pi \blacktriangleleft \begin{pmatrix} i(& j) \\ p(& q) \end{pmatrix} \quad (28)$$

into a growing partial alignment π , just as we insert unpaired bases or base pairs in the backtracing of the folding algorithm in section 3.2.

This algorithm is computationally very expensive, however. It requires $\mathcal{O}(n^4)$ memory and $\mathcal{O}(n^6)$ CPU time. Currently available software packages such as `foldalign` [39, 65] and `dynalign` [92] therefore implement only restricted versions. The simple, maximum matching style version is used in `pmcomp` [57] as an approach to comparing base pairing probability matrices.

6.5. Multiple Alignments. Pairwise alignment methods, be they for sequences or structures, can be readily generalized to alignments of many objects. Usually, it is too costly to compute optimal multiple alignments exactly, and one therefore resorts to heuristics such as progressive multiple alignments. `pmmulti` [57], for example, produces multiple structural alignments in the context of the Sankoff algorithm by calling `pmcomp` for pairwise alignments. For tree alignments, the `RNAforester` programs can be used to compute both pairwise and progressive multiple alignments.

As we have seen above, the mappings produced by tree editing do not correspond to tree alignments in general. These methods can therefore not be used for comparing multiple structures. The edit scripts can, however, be interpreted in terms of a *sequence* alignment. One may therefore still use these methods as the starting point for multiple sequence alignments. This is the central idea of the `MARNA` program [121] which uses pairwise structural alignments as input to the multiple alignment program `T-Coffee` [100].

6.6. Notes. Various variants, specializations, and generalizations of the tree-editing approach have been described in recent years. Examples include efficient algorithms for similar trees [67] and with simplified edit cost models [120]. Tree-editing with restricted mappings M satisfying stronger requirements on structural conservation are described in [148]. Let $\text{lca}(a, b)$ denote the “last common ancestor” of a and b . For all $(x', x''), (y', y''), (z', z'') \in M$ holds: $\text{lca}(x', y')$ is a proper ancestor of z' if and only if $\text{lca}(x'', y'')$ is a proper ancestor of z'' . Other variants of tree edit distances are discussed e.g. in [132]. A tree-edit model for RNA that allows additional “node-fusion” and “edge-fusion” events is described in [4].

More general edit models with application to RNA structures are described in [68, 88], an alignment distance for pseudoknotted structures can be found in [9].

A very different approach to the pairwise comparison of RNA structures, with or without pseudoknots, converts the RNA alignment problem into an integer programming problem [79]. Recently, efficient algorithms based on Lagrangian relaxation have been developed [6], that have helped to make the performance of this approach comparable to other methods.

A partition function version of the Sankoff algorithm, which can be used to compute the probabilities of all possible (mis)matches in a structural alignment of two RNA base pairing probability matrices is described in [61]. RNA structure comparison can also be recast in the SCFG framework [63]. The corresponding pair-SCFG algorithms correspond to the Sankoff algorithm.

7. Kinetic Folding

7.1. Folding Energy Landscapes. The folding dynamics of a particular RNA molecule can also be studied successfully within the framework of secondary structures. The folding process is determined by the energy landscape (or Potential Energy Surface, PES, in the terminology of theoretical chemistry). Instead of considering all possible spatial conformations, it is meaningful to partition the conformation space into sets of conformations that belong to a given secondary structure. Instead of a smooth surface defined on a space of real-valued coordinate vectors we are therefore dealing with a landscape on a complex graph [107]. The vertices of this graph are the secondary structures that can be formed by the given RNA sequence, the edges are determined by a rule specifying which structures can be interconverted, and the height of the landscape at a structure x is its free energy $E(x)$. Typically, one considers a “move set” that allows the insertion and deletion of single base pairs. In addition, a shift-move, which changes (i, j) to (i, k) or (h, j) is sometimes included [28]. Further coarse-grainings of this landscape can be achieved e.g. by considering secondary structures as composed of stacks instead of individual base pairs.

7.2. Kinetic Folding Algorithms. Several groups have designed kinetic folding algorithms for RNA secondary structures, mostly in an attempt to obtain more accurate predictions or in order to include pseudoknots, see e.g. [89, 95, 2, 43, 127]. Only a few papers have attempted to reconstruct folding pathways [50, 42, 124]. These algorithms generally operate on a list of all possible helices and consequently use move sets that destroy or form entire helices in a single move. Such a move set can introduce large structural changes in a single move and furthermore, *ad hoc* assumptions have to be made about the rates of helix formation and disruption. A more local move set is, therefore, preferable if one hopes to observe realistic folding trajectories.

The process of kinetic folding itself can be modeled as *homogeneous Markov chain*. The probability p_x that a given RNA molecule will have the secondary structure x at time t is given by the master equation

$$\frac{dp_x}{dt} = \sum_{y \in X} r_{xy} p_y(t). \quad (29)$$

where r_{xy} is the rate constant for the transition from secondary structure y to secondary structure x in the deterministic description [38]. The transition state model dictates an expression of the form

$$r_{yx} = r_0 e^{-\frac{E_{yx}^\ddagger - E(x)}{RT}} \quad \text{for } x \neq y \quad \text{and} \quad r_{xx} = - \sum_{y \neq x} r_{yx} \quad (30)$$

where the transition state energies E_{yx}^\ddagger must be symmetric, $E_{yx}^\ddagger = E_{xy}^\ddagger$, and r_0 is a scaling constant. In the simplest case one can use

$$E_{yx}^\ddagger = \max\{E(x), E(y)\} \quad (31)$$

For short sequences or very restricted subsets of conformations equation (29) can be solved exactly or integrated numerically [127]. Solving the master equation for larger conformation spaces is out of the question. In such cases the dynamics can

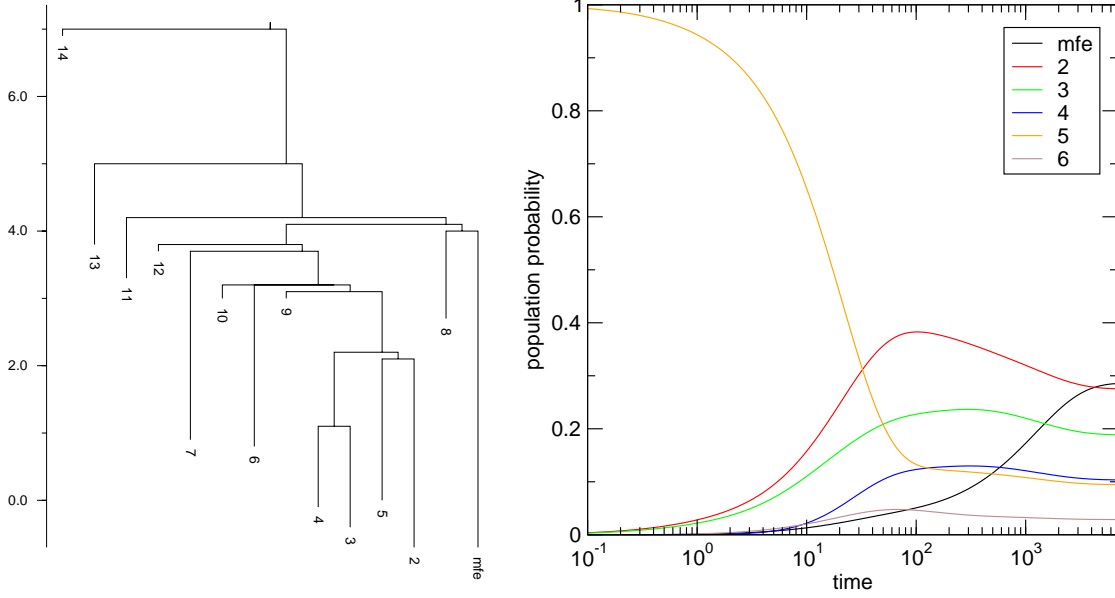


Figure 17. L.h.s.: Barrier tree of short artificial sequence UAUGCUGCGGCCUAGGC. The leaves of the tree are the local minima of the energy landscape. R.h.s.: folding kinetics from the open structure. Population density p_α for the basin containing the local minimum α is shown for the six largest basins as a function of time.

be obtained by simulating the Markov chain directly by a rejection-less Monte Carlo algorithm [31] and sampling a large number of trajectories.

7.3. Approximate Folding Trajectories and Barrier Trees. An alternative approach to the direct simulation of the master equation (29) starts with a more detailed analysis of folding energy landscape. Let us start with a few definitions:

A conformation x is a global minimum if $E(x) \leq E(y)$ for all $y \in X$ and a local minimum if $E(x) \leq E(y)$ for all neighbors y of x . The energy \hat{E} of the lowest saddle point separating two local minima x and y is

$$\hat{E}[x, y] = \min_{\mathbf{p} \in \mathbb{P}_{xy}} \max_{z \in \mathbf{p}} E(z) \quad (32)$$

where \mathbb{P}_{xy} is the set of all paths \mathbf{p} connecting x and y by a series of consecutive transformations taken from the move set. If the energy function is non-degenerate then there is a unique saddle point $s = s(x, y)$ connecting x and y characterized by $E(s) = \hat{E}[x, y]$. To each saddle point s there is a unique collection of conformations $B(s)$ that can be reached from s by a path along which the energy never exceeds $E(s)$. In other words, the conformations in $B(s)$ are mutually connected by paths that never go higher than $E(s)$. This property warrants to call $B(s)$ the *basin of attraction* below the saddle s .

Two situations can arise for any two saddle points s and s' with energies $E(s) < E(s')$. Either the basin of s is a “sub-basin” of $B(s')$ or the two basins are disjoint. This property arranges the local minima and the saddle points in a unique hierarchical structure which is conveniently represented as a tree, termed *barrier tree*.

An efficient flooding algorithm [29] can be used to identify local minima and saddle-points starting e.g. from the complete list of suboptimal secondary structures produced by the `RNAsubopt` program [144]. Consider a stack Σ which initially contains all secondary structure in the order of ascending energy. We pop the element z from the top of Σ and check which of its neighbors we have already seen before, i.e., which of its neighbors have a lower energy. There are three cases:

- (1) z has no neighbor with lower energy, then it is a local minimum, i.e., a new leaf of the barrier tree.
- (2) z has only lower energy neighbors that all belong the same basin, say $B(x)$. Then z itself also belongs to $B(x)$.
- (3) z has lower energy neighbors in two or more different basins. In this case z is the saddle point separating these basins, i.e., an interior vertex of the barrier tree. For the subsequent computation we now unify all basins connected by z into a new basin $B(z)$ and remove its sub-basins from the list of “active” basins.

At the end, we are left with the barrier tree of the landscape. As a by-product we also obtain the assignment of each secondary structure to its basin $B(x)$. Instead of searching through the list of all previously encountered structures it is more efficient to generate all neighbors of z and to check whether they have already been seen before by means of a hash-table lookup. The procedure thus runs in $\mathcal{O}(LD)$ time, where L is length of the list of structures and D is the maximal number of moves that can be applied to a secondary structure. The program `barriers` implements this algorithm [29].

A description of the energy landscape or the dynamics of an RNA molecule based on all secondary structures is feasible only for very small sequences. We therefore need to coarse-grain the representation of the energy landscape. Let $\Pi = \{\alpha, \beta, \dots\}$ be a partition of the state space. The classes of such a partition are *macro-states*. As a concrete example consider the partition of X defined by the gradient basins $\mathcal{B}(z)$ of the local energy minima. To each macrostate α we can assign the partition function

$$Z_\alpha = \sum_{x \in \alpha} e^{-E(x)/RT} \quad (33)$$

and the corresponding free energy

$$G(\alpha) = -RT \ln Z_\alpha \quad (34)$$

The transition rates between macrostates can be obtained at least approximately from the elementary rate constants using the assumption that the random process is equilibrated within each macro-state [143]. Then

$$r_{\beta\alpha} = \sum_{y \in \beta} \sum_{x \in \alpha} r_{yx} \frac{e^{-E(x)/RT}}{Z_\alpha} \quad \text{for } \alpha \neq \beta \quad (35)$$

We can use the transition state model to define the free energies of the transition state G^\ddagger by setting

$$r_{\beta\alpha} = r_0 e^{-\frac{G_{\beta\alpha}^\ddagger - G(\alpha)}{RT}} \quad (36)$$

A short computation then yields

$$G_{\beta\alpha}^{\neq} = -RT \ln \sum_{y \in \beta} \sum_{x \in \alpha} e^{-\frac{E_{yx}^{\neq}}{RT}} \quad (37)$$

as one would expect.

In practice one can compute $r_{\beta\alpha}$ “on the fly” while executing the **barriers** program if two conditions are satisfied: (a) For each x we can efficiently determine to which macro-state it belongs and (b) the double sum in equ. 35 needs to be evaluated only for pairs of neighboring conformations (x, y) . Condition (a) is easily satisfied for each of the gradient basins: in each step of the **barriers** algorithm all neighbors y of the newly added structures x that have a smaller energy have already been processed. Condition (b) is satisfied by construction of the microscopic transition rates r_{xy} , which vanish unless x is a neighbor of y . In the case of short sequence, both the microscopic model and the macro-state model can be solved exactly. In many cases, see e.g. Fig. 17, the macro-state model provides a very good approximation of the dynamics.

7.4. RNA Switches. Some RNA molecules exhibit two meta-stable conformations, whose equilibrium can be shifted easily by external events, such as binding of another molecule. This can be used to regulate gene expression, when the two mutually exclusive alternatives correspond to an active and in-active conformation of the transcript. The best known example of such behavior are the riboswitches [133] found in the 5'-UTRs of bacterial mRNAs, where the conformational change is triggered by binding of a small organic molecule.

Molecules that may be RNA switches can be recognized by inspection of the barrier tree, but this is feasible only for rather short sequences. The **paRNAss** program [134] instead uses a sample of suboptimal structures, and computes for every pair of structures “morphological” distance (e.g. tree edit distance) and a simple estimate of the energy barrier. The structures are then clustered according to these two measures, RNA switches are expected to exhibit two well separated clusters.

Interestingly, for any two secondary structures there exist sequences that are *compatible* with both structures, i.e. that can form both structures in principle [106]. If both structures are reasonably stable, it is not hard to design switching sequences with these two structures as stable conformations [30].

7.5. Notes. The analysis of landscapes becomes technically more complicated when structures, in particular adjacent structures, may have the same energy. In this case there is no unique definition of gradient basins and a variety of concepts, all related to saddle points, have to distinguished, see [29] for further details.

The notion of barrier trees can be generalized to multi-valued landscapes, which arise e.g. in the context of multi-objective optimization problems with conflicting constraints [122].

A computationally simpler alternative to the macro-state approach for transition rate is to assume an Arrhenius law $r_{\beta\alpha} \sim \exp(E^{\neq}/RT)$ and to approximate the transition

state energy E^\neq by the energy of the saddle point between the local minima α and β [143]

A generalization of the “intersection theorem” characterizes sets of more than two secondary structures that can be realized simultaneously by a common RNA sequence [30]. This observation can be used as a starting point for computational designs of switches with multiple states [1].

8. Concluding Remarks

Secondary structure drives the RNA folding process, arguably even more so than it is the case for proteins. This renders the prediction of RNA secondary structure highly relevant for the prediction of RNA structure and analysis, in general. As this chapter shows, the field of RNA structure prediction is comparatively well developed. As a matter of fact, it is one of the fields in bioinformatics that benefit most comprehensively from algorithmic methods derived from computer science. The comparatively technical makeup of this chapter is a mirror of this phenomenon. Notably, very different questions, which in the protein world require different mathematical models, can be described and analyzed in the RNA case at the level of secondary structures: the thermodynamics of folding as well as the thermodynamics of RNA-RNA interactions are accessible via the same parameters and the same algorithms that can also be used to compute consensus structures in an evolutionary context or to investigate the dynamics of the folding process itself.

With the increased importance of RNA in biology, in general — consider, for instance, the recent surge in work on RNAi (see also Chapter 46) and in the analysis of structural aspects of mRNA in the context of gene regulation — RNA secondary structure prediction is rapidly becoming an obligatory tool in the arsenal of bioinformatics analysis methods.

References

- [1] **Abfalter, I., C. Flamm and P. F. Stadler.** Design of Multi-Stable Nucleic Acid Sequences. 1-7. Mewes, H.-W. and Heun, V. and Frishman, D. and Kramer, S. Proceedings of the German Conference on Bioinformatics. GCB 2003 belleville Verlag Michael Farin 2003.
- [2] **Abrahams, J. P., M. van den Berg, E. van Batenburg and C. Pleij.** 1990. Prediction of RNA Secondary Structure, Including Pseudoknotting, by Computer Simulation. Nucl. Acids Res. **18**:3035-3044.
- [3] **Akutsu, T.** 2001. Dynamic Programming Algorithms for RNA Secondary structure prediction with pseudoknots. Discr. Appl. Math. **104**:45-62.
- [4] **Allali, J. and M.-F. Sagot.** 2005. A New Distance for High Level RNA Secondary Structure Comparison. IEEE/ACM Trans. Comp. Biol. Bioinf. **2**:3-14.
- [5] **Andronescu, M., Z. Zhang and A. Condon.** 2005. Secondary structure prediction of interacting RNA molecules.. J. Mol. Biol. **345**:987-1001.
- [6] **Bauer, M. and G. Klau.** Structural Alignment of Two RNA Sequences with Lagrangian Relaxation. 113-125. Fleischer, R. Proceedings of the 15th International Symposium, ISAAC 2004, Hong Kong Springer Verlag 2004.
- [7] **Bonhoeffer, S., J. S. McCaskill, P. F. Stadler and P. Schuster.** 1993. RNA Multi-Structure Landscapes. A Study Based on Temperature Dependent Partition Functions. Eur. Biophys. J. **22**:13-24.

- [8] **Bonnet, E., J. Wuyts, P. Rouzé and Y. Van de Peer.** 2004. Evidence that microRNA precursors, unlike other non-coding RNAs, have lower folding free energies than random sequences. *Bioinformatics* **20**:2911-2917.
- [9] **Brinkmeier, M.** 2005. Structural Alignments of Pseudo-knotted RNA-molecules in polynomial time. TU Ilmenau Technical Reports.
- [10] **Brown, J. W., J. M. Nolan, E. S. Haas, M. A. T. Rubio, F. Major and N. R. Pace.** 1996. Comparative analysis of ribonuclease P RNA using gene sequences from natural microbial populations reveals tertiary structural elements. *Proc. Natl. Acad. Sci. USA* **93**:3001-3006.
- [11] **Cai, L., R. L. Malmberg and Y. Wu.** 2003. Stochastic modeling of RNA pseudoknotted structures: a grammatical approach. *Bioinformatics* **19 Suppl 1**:i66-73.
- [12] **Cannone, J. J., S. Subramanian, M. N. Schnare, J. R. Collett, L. M. D'Souza, Y. Du, B. Feng, N. L. n, L. V. Madabusi, K. M. Mller, N. Pande, Z. Shang, N. Yu and R. R. Gutell.** 2002. The Comparative RNA Web (CRW) Site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics* **3**:2.
- [13] **Chen, W. Y. C., E. Y. P. Deng and R. R. X. Du.** 2005. Reduction of m -regular noncrossing partitions. *Eur. J. Comb.* **26**:237-243.
- [14] **Chiu, D. K. and T. Kolodziejczak.** 1991. Inferring consensus structure from nucleic acid sequences. *CABIOS* **7**:347-352.
- [15] **Clote, P.** 2005. An efficient algorithm to compute the landscape of locally optimal RNA secondary structures with respect to the Nussinov-Jacobson energy model. *J. Comp. Biol.* **12**:83-101.
- [16] **Condon, A., B. Davy, B. Rastegari, S. Zhao and F. Tarrant.** 2004. Classifying RNA pseudoknotted structures. *Theor. Computer Sci.* **320**:35-50.
- [17] **Cristianini, N. and J. Shawe-Taylor.** 2000. An Introduction to Support Vector Machines. Cambridge University Press Cambridge, UK
- [18] **De Rijk, P. and R. De Wachter.** 1997. RnaViz, a program for the visualisation of RNA secondary structure. *Nucleic Acids Res.* **25**:4679-4684.
- [19] **Deutsch, E. and L. W. Shapiro.** 2002. A bijection between ordered trees and 2-Motzkin paths and its many consequences. *Discrete Math.* **256**:655-670.
- [20] **Dimitrov, R. A. and M. Zuker.** 2004. Prediction of Hybridization and Melting for Double-Stranded Nucleic Acids. *Biophys. J.* **87**:215-226.
- [21] **Ding, Y., C. Chan and C. Lawrence.** 2004. Sfold web server for statistical folding and rational design of nucleic acids. *Nucleic Acids Research* **32**:W135-41.
- [22] **Dirks, R. and N. Pierce.** 2003. A partition function algorithm for nucleic Acid secondary structure including pseudoknots. *J. Comput. Chem.* **24**:1664-1677.
- [23] **Dowell, R. D. and S. R. Eddy.** 2004. Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics* **5**:7 (1-14).
- [24] **Dulucq, S. and L. Tichit.** 2003. RNA secondary structure comparison: exact analysis of the Zhang-Shasha tree-edit algorithm. *Theor. Comp. Sci.* **306**:471-484.
- [25] **Eddy, S.** 2002. A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure. *BMC Bioinformatics* **3**:18.
- [26] **Evers, D. J. and R. Giegerich.** Reducing the conformation space in RNA structure prediction. 118-124. Hofestädt, Ralf and Wingender, Edgar Proceedings of the German Conference on Bioinformatics GCB'01, Braunschweig Bioinformation Systems e.V. 2001.
- [27] **Flamm, C., I. L. Hofacker and . P. F. Stadler.** 2004. Computational Chemistry with RNA Secondary Structures. *Kemija u industriji* **53**:315-322.
- [28] **Flamm, C., I. L. Hofacker and P. F. Stadler.** 1999. RNA *in silico*: The Computational Biology of RNA Secondary Structures. *Adv. Complex Syst.* **2**:65-90.
- [29] **Flamm, C., I. L. Hofacker, P. F. Stadler and M. T. Wolfinger.** 2002. Barrier Trees of Degenerate Landscapes. *Z. Phys. Chem.* **216**:155-173.
- [30] **Flamm, C., I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler and M. Zehl.** 2001. Design of Multi-Stable RNA Molecules. *RNA* **7**:254-265.

- [31] **Flamm, C., W. Fontana, I. Hofacker and P. Schuster.** 2000. RNA folding kinetics at elementary step resolution. *RNA* **6**:325–338.
- [32] **Fontana, W., D. A. M. Konings, P. F. Stadler and P. Schuster.** 1993. Statistics of RNA Secondary Structures. *Biopolymers* **33**:1389-1404.
- [33] **Fontana, W., P. F. Stadler, E. G. Bornberg-Bauer, T. Griesmacher, I. L. Hofacker, M. Tacker, P. Tarazona, E. D. Weinberger and P. Schuster.** 1993. RNA Folding Landscapes and Combinatory Landscapes. *Phys. Rev. E* **47**:2083-2099.
- [34] **Gautheret, D., F. Major and R. Cedergren.** 1990. Pattern searching/alignment with RNA primary and secondary structures: an effective descriptor for tRNA. *Comput. Appl. Biosci.* **6**:325-331.
- [35] **Gibson, A., V. Gowri-Shankar, P. G. Higgs and M. Rattray.** 2005. A comprehensive analysis of mammalian mitochondrial genome base composition and improved phylogenetic methods.. *Mol. Biol. Evol.* **22**:251-64.
- [36] **Giegerich, R., B. Voss and M. Rehmsmeier.** 2004. Abstract shapes of RNA. *Nucleic Acids Res.* **32**:4843-4851.
- [37] **Giegerich, R.** 2000. A systematic approach to dynamic programming in bioinformatics. *Bioinformatics* **16**:665-677.
- [38] **Gillespie, D. T.** 1976. A General Method for Numerically Simulating the Stochastic Time Evolution of Coupled Chemical Reactions. *J. Comput. Phys.* **22**:403.
- [39] **Gorodkin, J., L. J. Heyer and G. D. Stormo.** 1997. Finding the most significant common sequence and structure motifs in a set of RNA sequences. *Nucl. Acids Res.* **25**:3724–3732.
- [40] **Gräf, S., D. Strothmann, S. Kurtz and G. Steger.** 2001. HyPaLib: a database of RNAs and RNA structural elements defined by hybrid patterns. *Nucl. Acids. Res.* **29**:196-198.
- [41] **Gulyaev, A. P., F. H. D. van Batenburg and C. W. A. Pleij.** 1999. An approximation of loop free energy values of RNA H-pseudoknots. *RNA* **5**:609-617.
- [42] **Gulyaev, A. P., van Batenburg and C. W. A. Pleij.** 1995. The computer simulation of RNA folding pathways using an genetic algorithm.. *J. Mol. Biol.* **250**:37-51.
- [43] **Gulyaev, A. P.** 1991. The computer simulation of RNA folding involving pseudoknot formation. *Nucl. Acids Res.* **19**:2489-2493.
- [44] **Gutell, R. R., A. Power, G. Z. Hertz, E. J. Putz and G. D. Stormo.** 1992. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. *Nucl. Acids Res.* **20**:5785-5795.
- [45] **Gutell, R. R. and C. R. Woese.** 1990. Higher order structural elements in ribosomal RNAs: Pseudo-knots and the use of noncanonical pairs. *Proc. Natl. Acad. Sci. USA* **87**:663-667.
- [46] **Han, K., D. Kim and H. J. Kim.** 1999. A vector-based method for drawing RNA secondary structure. *Bioinformatics* **15**:286-297.
- [47] **Han, K. and H.-J. Kim.** 1993. Prediction of common folding structures of homologous RNAs. *Nucl. Acids Res.* **21**:1251-1257.
- [48] **Han, K. and Y. Byun.** 2003. PSEUDOVIEWER2: Visualization of RNA pseudoknots of any type. *Nucleic Acids Res.* **31**:3432-3440.
- [49] **Harris, J. K., E. S. Haas, D. Williams and D. N. Frank.** 2001. New insight into RNase P RNA structure from comparative analysis of the archaeal RNA. *RNA* **7**:220-232.
- [50] **Higgs, P. G.** 1995. Thermodynamic Properties of Transfer RNA: A Computational Study. *J. Chem. Soc. Faraday Trans.* **91**:2531-2540.
- [51] **Höchsmann, M., T. Töller, R. Giegerich and S. Kurtz.** Local Similarity in RNA Secondary Structures. 159-168. Blauvelt, Pal Proc of the Computational Systems Bioinformatics Conference, Stanford, CA, August 2003 (CSB 2003) IEEE Press 2003.
- [52] **Hofacker, I. L., B. Priwitzer and P. F. Stadler.** 2004. Prediction of Locally Stable RNA Secondary Structures for Genome-Wide Surveys. *Bioinformatics* **20**:191-198.
- [53] **Hofacker, I. L., M. Fekete and P. F. Stadler.** 2002. Secondary Structure Prediction for Aligned RNA Sequences. *J. Mol. Biol.* **319**:1059-1066.
- [54] **Hofacker, I. L., M. Fekete, C. Flamm, M. A. Huynen, S. Rauscher, P. E. Stolorz and P. F. Stadler.** 1998. Automatic Detection of Conserved RNA Structure Elements in Complete RNA Virus Genomes. *Nucl. Acids Res.* **26**:3825–3836.

- [55] **Hofacker, I. L., P. Schuster and P. F. Stadler.** 1998. Combinatorics of RNA Secondary Structures. *Discr. Appl. Math.* **89**:177-207.
- [56] **Hofacker, I. L., R. Stocsits and P. F. Stadler.** 2004. Conserved RNA Secondary Structures in Viral Genomes: A Survey. *Bioinformatics* **20**:1495-1499.
- [57] **Hofacker, I. L., S. H. F. Bernhart and P. F. Stadler.** 2004. Alignment of RNA Base Pairing Probability Matrices. *Bioinformatics* **20**:2222-2227.
- [58] **Hofacker, I. L., W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker and P. Schuster.** 1994. Fast Folding and Comparison of RNA Secondary Structures. *Monatsh. Chem.* **125**:167-188.
- [59] **Hofacker, I. L. and P. F. Stadler.** 1999. Automatic Detection of Conserved Base Pairing Patterns in RNA Virus Genomes. *Comp. & Chem.* **23**:401-414.
- [60] **Hofacker, I. L. and P. F. Stadler.** 2006. Memory Efficient Folding Algorithms for Circular RNA Secondary Structures. *Bioinformatics* **22**.
- [61] **Hofacker, I. L. and S. P. F..** The Partition Function Variant of Sankoff's Algorithm. 728-735. Bubak, Marian and van Albada, Geert Dick and Sloot, Peter M. A. and Dongarra, Jack J. *Computational Science - ICCS 2004* Springer 2004.
- [62] **Holbrook, S. R.** 2005. RNA Structure: The Long and the Short of it. *Curr. Op. Struct. Biol.* **15**:302-308.
- [63] **Holmes, I. and G. M. Rubin.** 2002. Pairwise RNA structure comparison with stochastic context-free grammars. *Pac. Symp. Biocomput.* :163-174.
- [64] **Holmes, I.** 2005. Accelerated probabilistic inference of RNA structure evolution. *BMC Bioinformatics* **6**:73 [epub].
- [65] **Hull Havgaard, J., R. Lyngsø, G. Stormo and J. Gorodkin.** 2005. Pairwise local structural alignment of RNA sequences with sequence similarity less than 40%. *Bioinformatics* **21**:1815-1824.
- [66] **Isambert, H. and E. D. Siggia.** 2000. Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme. *Proc. Natl. Acad. Sci. USA* **97**:6515-6520.
- [67] **Jansson, J. and A. Lingas.** 2003. A fast algorithm for optimal alignment between similar ordered trees. *Fund. Inf.* **56**:105-120.
- [68] **Jiang, T., G. Lin, B. Ma and K. Zhang.** 2002. A general edit distance between beteen RNA structures. *J. Comp. Biol.* **9**:371-388.
- [69] **Jiang, T., J. Wang and K. Zhang.** 1995. Alignment of Trees — an alternative to tree edit. *Theor. Comp. Sci.* **143**:137-148.
- [70] **Jiménez, V. M. and A. Marzal.** Computation of the N best parse trees for weighted and stochastic context-free grammars. 183-192. Francesc J. Ferri and José Manuel Iñesta Quereda and Adnan Amin and Pavel Pudil *Advances in Pattern Recognition, Joint IAPR International Workshops SSPR 2000 and SPR 2000* Springer 2000.
- [71] **Juan, V. and C. Wilson.** 1999. RNA Secondary Structure Prediction Based on Free Energy and Phylogenetic Analysis. *J. Mol. Biol.* **289**:935-947.
- [72] **Klein, P.** Computing the Edit distance between unrooted ordered trees. 91-102. Gianfranco Bilardi and Giuseppe F. Italiano and Andrea Pietracaprina and Geppino Pucci *Algorithms - ESA '98, 6th Annual European Symposium, Venice, Italy, August 24-26, 1998, Proceedings* Springer 1998.
- [73] **Klein, R. J. and S. R. Eddy.** 2003. RSEARCH: Finding homologs of single structured RNA sequences. *BMC Bioinformatics* **4**:1471-2105.
- [74] **Knudsen, B. and J. Hein.** 2003. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucl. Acids Res.* **31**:3423-3428.
- [75] **Knudsen, B. and J. J. Hein.** 1999. Using stochastic context free grammars and molecular evolution to predict RNA secondary structure. *Bioinformatics* **15**:446-454.
- [76] **Le, S.-Y., J.-H. Chen, K. Currey and J. Maizel.** 1988. A Program for Predicting Significant RNA Secondary Structures. *CABIOS* **4**:153-159.
- [77] **Le, S. Y. and M. Zuker.** 1991. Predicting Common Foldings of Homologous RNAs. *J. Biomol. Struct. Dyn.* **8**:1027-1044.

- [78] **Lee, D. and K. Han.** 2002. Prediction of RNA Pseudoknots — Comparative Study of Genetic Algorithms. *Genome Informatics* **13**:414-415.
- [79] **Lenhof, H.-P., K. Reinert and M. Vingron.** 1998. A polyhedral approach to RNA sequence structure alignment. *J. Comput. Biol.* **5**:517-530.
- [80] **Leydold, J. and P. F. Stadler.** 1998. Minimal Cycle Basis of Outerplanar Graphs. *Elec. J. Comb.* **5**:209-222 [R16: 14 p.].
- [81] **Liao, B. and T. Wang.** 2002. An enumeration of RNA secondary structure. *Math. Appl.* **15**:109-112.
- [82] **Louise-May, S., P. Auffinger and E. Westhof.** 1996. Calculations of Nucleic Acid Conformations. *Curr. Op. Struct. Biol.* **6**:289-298.
- [83] **Lowe, T. M. and S. Eddy.** 1997. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucl. Acids Res.* **25**:955-964.
- [84] **Lück, R., G. Steger and D. Riesner.** 1996. Thermodynamic Prediction of Conserved Secondary Structure: Application to the RRE element of HIV, the tRNA-like element of CMV, and the mRNA of Prion Protein. *J. Mol. Biol.* **258**:813-826.
- [85] **Lück, R., S. Gräf and G. Steger.** 1999. ConStruct: A tool for thermodynamic controlled prediction of conserved secondary structure. *Nucl. Acids Res.* **27**:4208-4217.
- [86] **Lyngsø, R. B., M. Zuker and C. N. Pedersen.** 1999. Fast evaluation of internal loops in RNA secondary structure prediction. *Bioinformatics* **15**:440-445.
- [87] **Lyngsø, R. B. and C. N. S. Pedersen.** 2000. RNA Pseudoknot Prediction in Energy-Based Models. *J. Comp. Biol.* **7**:409-427.
- [88] **Ma, B., L. Wang and K. Zhang.** 2002. Computational similarity between RNA structures. *Theor. Comp. Sci.* **276**:111-132.
- [89] **Martinez, H. M.** 1984. An RNA folding rule. *Nucl. Acid Res.* **12**:323-335.
- [90] **Mathews, D. H., J. Sabina, M. Zuker and H. Turner.** 1999. Expanded Sequence Dependence of Thermodynamic Parameters Provides Robust Prediction of RNA Secondary Structure. *J. Mol. Biol.* **288**:911-940.
- [91] **Mathews, D. H., M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker and D. H. Turner.** 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. USA* **101**:7287-7292.
- [92] **Mathews, D. H. and D. H. Turner.** 2002. Dynalign: An Algorithm for Finding Secondary Structures Common to Two RNA Sequences. *J. Mol. Biol.* **317**:191-203.
- [93] **McCaskill, J.** 1990. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers* **29**:1105-1119.
- [94] **McCutcheon, J. P. and S. R. Eddy.** 2003. Computational identification of non-coding RNAs in *Saccharomyces cerevisiae* by comparative genomics. *Nucl. Acids Res.* **31**:4119-4128.
- [95] **Mironov, A. A., L. P. Dyakonova and A. E. Kister.** 1985. A Kinetic Approach to the Prediction of RNA Secondary Structures. *Journal of Biomolecular Structure and Dynamics* **2**:953.
- [96] **Missal, K., D. Rose and P. F. Stadler.** 2005. Non-coding RNAs in *Ciona intestinalis*. *Bioinformatics* **21** S2:i77-i78.
- [97] **Mückstein, U., I. L. Hofacker and P. F. Stadler.** 2002. Stochastic Pairwise Alignments. *Bioinformatics* **S153-S160**:18.
- [98] **Muller, G., C. Gaspin, A. Etienne and E. Westhof.** 1993. Automatic display of RNA secondary structures. *Comput. Appl. Biosci.* **9**:551-561.
- [99] **Needleman, S. B. and C. D. Wunsch.** 1970. A general method applicable to the search for similarities in the aminoacid sequences of two proteins. *J. Mol. Biol.* **48**:443-452.
- [100] **Notredame, C., D. Higgins and J. Heringa.** 2000. T-Coffee: A novel method for multiple sequence alignments. *J. Mol. Biol.* **302**:205-217.
- [101] **Nussinov, R., G. Piecznik, J. R. Griggs and D. J. Kleitman.** 1978. Algorithms for Loop Matching. *SIAM J. Appl. Math.* **35**:68-82.
- [102] **Pedersen, J. S., I. M. Meyer, R. Forsberg and J. Hein.** 2004. An evolutionary model for protein-coding regions with conserved RNA structure. *Mol. Biol. Evol.* **21**:1913-1922.

- [103] **Pedersen, J. S., I. M. Meyer, R. Forsberg, P. Simmonds and J. Hein.** 2004. A comparative method for finding and folding RNA secondary structures within protein-coding regions. *Nucl. Acids Res.* **32**:4925-4936.
- [104] **Reeder, J. and R. Giegerich.** 2004. Design, implementation and evaluation of a practical pseudoknot folding algorithm based on thermodynamics. *BMC Bioinformatics* **5**:104.
- [105] **Rehmsmeier, M., P. Steffen, M. Höchsmann and R. Giegerich.** 2004. Fast and effective prediction of microRNA/target duplexes. *RNA* **10**:1507-1517.
- [106] **Reidys, C., P. F. Stadler and P. Schuster.** 1997. Generic Properties of Combinatory Maps: Neutral Networks of RNA Secondary Structures. *Bull. Math. Biol.* **59**:339-397.
- [107] **Reidys, C. M. and P. F. Stadler.** 2002. Combinatorial Landscapes. *SIAM Review* **44**:3-54.
- [108] **Repsilber, D., S. Wiese, M. Rachen, A. W. Schroder, D. Riesner and G. Steger.** 1999. Formation of metastable RNA structures by sequential folding during transcription: time-resolved structural analysis of potato spindle tuber viroid (-)-stranded RNA by temperature-gradient gel electrophoresis. *RNA* **5**:574-584.
- [109] **Rivas, E., R. J. Klein, T. A. Jones and S. R. Eddy.** 2001. Computational identification of noncoding RNAs in *E. coli* by comparative genomics. *Curr. Biol.* **11**:1369-1373.
- [110] **Rivas, E. and S. R. Eddy.** 1999. A Dynamic Programming Algorithm for RNA Structure Prediction Including Pseudoknots. *J. Mol. Biol.* **285**:2053-2068.
- [111] **Rivas, E. and S. R. Eddy.** 2001. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* **2**:19 pages.
- [112] **Rivas, E. and S. R. Eddy.** 2001. Noncoding RNA gene detection using comparative sequence analysis. *BMC Bioinformatics* **2**:8.
- [113] **Ruan, J., G. D. Stormo and W. Zhang.** 2004. An Iterated loop matching approach to the prediction of RNA secondary structures with pseudoknots. *Bioinformatics* **20**:58-66.
- [114] **Sankoff, D.** 1985. Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. *SIAM J. Appl. Math.* **45**:810-825.
- [115] **Schultes, E. A. and D. P. Bartel.** 2000. One Sequence, Two Ribozymes: Implications for the emergence of New Ribozyme Folds. *Science* **289**:448-452.
- [116] **Schuster, P., W. Fontana, P. F. Stadler and I. L. Hofacker.** 1994. From Sequences to Shapes and Back: A case study in RNA secondary structures. *Proc. Roy. Soc. Lond. B* **255**:279-284.
- [117] **Shapiro, B. A., J. Maizel, L. E. Lipkin, K. Currey and C. Whitney.** 1984. Generating non-overlapping displays of nucleic acid secondary structure. *Nucleic Acids Res.* **12**:75-88.
- [118] **Shapiro, B. A. and K. Zhang.** 1990. Comparing Multiple RNA Secondary Structures Using Tree Comparisons. *CABIOS* **6**:309-318.
- [119] **Shapiro, B. A.** 1988. An algorithm for comparing multiple RNA secondary structures. *CABIOS* **4**:387-393.
- [120] **Shasha, D. and K. Zhang.** 1990. Fast algorithm for the unit cost editing distance between trees. *J. Algorithms* **11**:581-621.
- [121] **Siebert, S. and R. Backofen.** MARNA: A server for multiple alignment of RNAs. 135-140. Mewes, H.-W. and Heun, V. and Frishman, D. and Kramer, S. Proceedings of the German Conference on Bioinformatics. GCB 2003 belleville Verlag Michael Farin 2003.
- [122] **Stadler, P. F. and C. Flamm.** 2003. Barrier Trees on Poset-Valued Landscapes. *Genetic Prog. Evolv. Mach.* **7-20**:4.
- [123] **Steger, G., H. Hofmann, J. Fortsch, H. J. Gross, J. W. Randles, H. L. Sanger and D. Riesner.** 1984. Conformational transitions in viroids and virusoids: comparison of results from energy minimization algorithm and from experimental data. *J. Biomol. Struct. Dyn.* **2**:543-571.
- [124] **Suvernev, A. and P. Frantsuzov.** 1995. Statistical Description of Nucleic Acid Secondary Structure Folding. *J. Biomolec. Struct. Dyn.* **13**:135-144.
- [125] **Tabaska, J. E., R. B. Cary, H. N. Gabow and G. D. Stormo.** 1998. An RNA folding method capable of identifying pseudoknots and base triples. *Bioinformatics* **14**:691-699.

- [126] **Tacker, M., P. F. Stadler, E. G. Bornberg-Bauer, I. L. Hofacker and P. Schuster.** 1996. Algorithm Independent Properties of RNA Structure Prediction. *Eur. Biophys. J.* **25**:115-130.
- [127] **Tacker, M., W. Fontana, P. F. Stadler and P. Schuster.** 1994. Statistics of RNA Melting Kinetics. *Eur. Biophys. J.* **23**:29-38.
- [128] **Tai, K.** 1979. The Tree-to-Tree Correction Problem. *J. ACM* **26**:422-433.
- [129] **Thirumalai, D., N. Lee, S. A. Woodson and D. K. Klimov.** 2001. Early Events in RNA Folding. *Annu. Rev. Phys. Chem.* **52**:751-762.
- [130] **Thirumalai, D.** 1998. Native Secondary Structure Formation in RNA may be a Slave to Tertiary Folding. *Proc. Natl. Acad. Sci.* **95**:11506-11508.
- [131] **Thurner, C., C. Witwer, I. Hofacker and P. F. Stadler.** 2004. Conserved RNA Secondary Structures in Flaviviridae Genomes. *J. Gen. Virol.* **85**:1113-1124.
- [132] **Valiente, G.** An Efficient Bottom-Up Distance between Trees. 212–219. Werner, Bob Proc. 8th Int. Symposium on String Processing and Information Retrieval SPIRE’01 IEEE Computer Science Press 2001.
- [133] **Vitreschak, A. G., D. A. Rodionov, A. A. Mironov and M. S. Gelfand.** 2004. Riboswitches: the oldest mechanism for the regulation of gene expression?. *Trends Gen.* **20**:44-50.
- [134] **Voss, B., C. Meyer and R. Giegerich.** 2004. Evaluating the Predictability of Conformational Switching in RNA. *Bioinformatics* **20**:1573-1582.
- [135] **Walter, A., D. Turner, J. Kim, M. Lyttle, P. Müller, D. Mathews and M. Zuker.** 1994. Co-axial stacking of helices enhances binding of oligoribonucleotides and improves predictions of RNA folding. *Proc. Natl. Acad. Sci. USA* **91**:9218-9222.
- [136] **Washietl, S., I. L. Hofacker and P. F. Stadler.** 2005. Fast and reliable prediction of noncoding RNAs. *Proc. Natl. Acad. Sci. USA* **102**:2454-2459.
- [137] **Washietl, S., I. L. Hofacker, M. Lukasser, A. Hüttenhofer and P. F. Stadler.** 2005. Mapping of conserved RNA Secondary Structures predicts Thousands of functional Non-Coding RNAs in the Human Genome. *Nature Biotech.* **23**:1383-1390.
- [138] **Washietl, S. and I. L. Hofacker.** 2004. Consensus Folding of Aligned Sequences as a New Measure for the Detection of Functional RNAs by Comparative Genomics.. *J. Mol. Biol.* **342**:19-30.
- [139] **Waterman, M. S. and T. F. Smith.** 1978. RNA secondary structure: A complete mathematical analysis. *Mathematical Biosciences* **42**:257-266.
- [140] **Waterman, M. S.** 1978. Secondary structure of single - stranded nucleic acids. *Studies on foundations and combinatorics, Advances in mathematics supplementary studies*, Academic Press N.Y. **1**:167 - 212.
- [141] **Witwer, C., I. L. Hofacker and P. F. Stadler.** 2004. Prediction of Consensus RNA Secondary Structures Including Pseudoknots. *IEEE/ACM Trans. Comp. Biol. Bioinf.* **1**:65-77.
- [142] **Witwer, C., S. Rauscher, I. L. Hofacker and P. F. Stadler.** 2001. Conserved RNA Secondary Structures in Picornaviridae Genomes. *Nucl. Acids Res.* **29**:5079-5089.
- [143] **Wolfinger, M. T., W. A. Svrcek-Seiler, C. Flamm, I. L. Hofacker and P. F. Stadler.** 2004. Exact Folding Dynamics of RNA Secondary Structures. *J. Phys. A: Math. Gen.* **37**:4731-4741.
- [144] **Wuchty, S., W. Fontana, I. L. Hofacker and P. Schuster.** 1999. Complete Suboptimal Folding of RNA and the Stability of Secondary Structures. *Biopolymers* **49**:145-165.
- [145] **Xia, T., J. SanatLucia Jr., M. E. Burkard, R. Kierzek, S. J. Schroeder, X. Jiao, C. Cox and D. H. Turner.** 1998. Parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick pairs. *Biochemistry* **37**:14719–14735.
- [146] **Younger, D. H.** 1967. Recognition and parsing of context-free languages in time n^3 . *Information & Control* **10**:189-208.
- [147] **Zhang, K. and D. Shasha.** 1989. Simple fast algorithms for the editing distance between trees and related problems. *SIAM J. Computing* **18**:1245-1262.
- [148] **Zhang, K.** 1995. Algorithms for the constrained editing problem between ordered labeled trees and related problems. *Pattern Recogn.* **28**:463-474.

- [149] **Zuker, M. and P. Stiegler.** 1981. Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research* **9**:133-148.
- [150] **Zuker, M.** 1989. On finding all suboptimal foldings of an RNA molecule. *Science* **244**:48-52.
- [151] **Zwieb, C., I. Wower and J. Wower.** 1999. Comparative sequence analysis of tmRNA. *Nucl. Acids Res.* **27**:2063-2071.