J. Math. Biol. (2008) 56:129–144
DOI 10.1007/s00285-007-0107-5

Mathematical Biology

Variations on RNA folding and alignment: lessons from Benasque

Athanasius F. Bompfünewerer · Rolf Backofen · Stephan H. Bernhart · Jana Hertel · Ivo L. Hofacker · Peter F. Stadler · Sebastian Will

Received: 2 October 2006 / Revised: 24 April 2007 / Published online: 5 July 2007

Abstract Dynamic programming algorithms solve many standard problems of RNA bioinformatics in polynomial time. In this contribution we discuss a series of variations on these standard methods that implement refined biophysical models, such as a restriction of RNA folding to canonical structures, and an extension of structural alignments to an explicit scoring of stacking propensities. Furthermore, we demonstrate that a local structural alignment can be employed for ncRNA gene finding. In this context we discuss scanning variants for folding and alignment algorithms.

Keywords RNA folding · Secondary structure alignment · Dynamic programming

Mathematics Subject Classification (2000) 90C27 · 90C90 · 92C40

A. F. Bompfünewerer
Zentralfriedhof Wien, 3. Tor Simmeringer Haupstraße, 1110 Wien, Austria

A. F. Bompfünewerer · S. H. Bernhart · I. L. Hofacker · P. F. Stadler
Department of Theoretical Chemistry, University of Vienna, Währingerstraße 17, 1090 Wien, Austria

R. Backofen · S. Will
Bioinformatics Group, Department of Computer Science, University of Freiburg,
Georges-Köhler-Allee, Geb. 106, 79110 Freiburg, Germany

J. Hertel · P. F. Stadler (✉)
Bioinformatics Group, Department of Computer Science,
and Interdisciplinary Center for Bioinformatics, University of Leipzig,
Härtelstraße 16-18, 04107 Leipzig, Germany
e-mail: studla@bioinf.uni-leipzig.de

P. F. Stadler
Santa Fe Institute, 1399 Hyde Park Rd., Santa Fe, NM 87501, USA

1 Introduction

Starting with the discovery of microRNAs and the advent of genome-wide transcriptomics, non-protein-coding transcripts have moved from a fringe topic to a central field of research in molecular biology. Most well-described functional RNAs have distinctive structures that are conserved over evolutionary time scales. As efficient algorithms exist to predict RNA secondary structure, bioinformatics has played an important role in RNA research almost from the beginning [23].

We describe here a collection of novel improved variants of RNA folding and alignment that were conceived during the 2006 RNA Meeting in Benasque.[1] In particular, the accuracy of structure prediction can sometimes be improved by considering only *canonical structures*, i.e., those that contain no "isolated basepairs". This leads to the idea to emphasize stacking also in other contexts, such as structure comparison. The prerequisite for such approaches is the observation that stacking probabilities can be efficiently computed as a post-processing of dynamic programming table entries that are computed and stored already in the folding algorithms as implemented in the `Vienna RNA Package`.

2 The loop-based energy model

RNA secondary structures can be seen as outer-planar graphs whose vertices are the nucleotides and whose edges represent the covalent backbone of the molecule as well as the basepairs. This class of graphs has a unique outer-planar embedding whose bounded faces form the unique minimum cycle basis [13]. These faces, in the context of RNA usually called *"loops"*, have a direct biophysical interpretation as stabilizing stacked basepairs or entropically destabilizing elements. Thus they form the units of the standard additive energy model. Energy parameters that depend on sequence, length, and type of the loops have been carefully measured over the last two decades [16,17]. From the biophysical point of view one distinguishes hairpin loops, stacked base pairs, bulges, true interior loops, and multi(branched) loops. From an algorithmic point of view one can treat bulges, stacked pairs, and true interior loops as subtypes of interior loops. We shall see below, however, that in some cases stacked pairs require separate treatment.

We consider an RNA sequence x of length n. The nucleotide at sequence position k is x_k, and $x[k,l]$ denotes the sub-sequence $(x_k,\dots,x_l)$. Hairpin loops are uniquely determined by their closing pair (k,l). The energy of a hairpin loop is parametrized as

$$\mathcal{H}(k,l) = \mathcal{H}(x[k,l]) = \mathcal{H}(x_k, x_{k+1}, \ell, x_{l-1}, x_l) \tag{1}$$

where ℓ is the length of the loop (expressed as the number of its unpaired nucleotides). Each interior loop is determined by the two base pairs enclosing it. Its energy is

[1] RNA-2006, Benasque, Spain, 14–27 July 2006.

tabulated as

$$\mathcal{I}(k,l;\,p,q)=\mathcal{I}(x[k,p],x[q,l])=\mathcal{I}(x_k,x_{k+1};\,\ell_1;\,x_{p-1},x_p;\,x_q,x_{q+1};\,\ell_2;\,x_{l-1},x_l) \tag{2}$$

where $\ell_1 = p-k+1$ is the length of unpaired strand between k and p and $\ell_2 = l-q+1$ is the length of the unpaired strand between q and l. Symmetry of the energy model dictates $\mathcal{I}(k,l;\,p,q) = \mathcal{I}(q,p;\,l,k)$. If $\ell_1 = \ell_2 = 0$ we have a (stabilizing) stacked pair, if only one of ℓ_1 and ℓ_2 vanish we have a bulge. For multiloops, finally, we have an additive energy model of the form $\mathcal{M} = a+b\times B+c\times\ell$ where ℓ is the length of multiloop (again expressed as the number of unpaired nucleotides) and $B \geq 2$ is the number of branches, not counting the branch in which the closing pair of the loop resides.

Modern versions of the energy parameters also consider so-called *dangling ends*, i.e., extra energy contributions of the terminal base pairs in multiloops and exterior to any other basepair. Although they can be fairly easily included in the algorithms below, we suppress the dangling end contributions for clarity of presentation. They are, however, implemented in the programs discussed here.

3 The basic folding recursions

Energy-directed RNA folding is solved by dynamic programming algorithms that are based on decomposing the set of possible structures into sets of sub-structures that are defined on sub-sequences. This decomposition can be chosen such that each possible structure appears in exactly one of the subcases, see Fig. 1 (top) for a graphical representation. In the course of RNA folding algorithms for linear RNA molecules, the Vienna RNA Package [7,9] computes the following arrays for $i < j$, which correspond to the distinct types of substructures in Fig. 1 (top).

- 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 basepair.
- M_{ij} free energy of the optimal substructure on the subsequence $x[i,j]$ subject to the constraint that $x[i,j]$ is part of a multiloop and has at least one component, i.e., a sub-sequence that is enclosed by a basepair.
- M^1_{ij} free energy of the optimal substructure on the subsequence $x[i,j]$ subject to the constraint 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 "conventional" energy minimization algorithm for linear RNA molecules [26, 27] can be summarized in the following way. We give the recursions in the form in which they are implemented in the Vienna RNA Package [7,9]:

$$F_{ij} = \min\left\{F_{i+1,j},\ \min_{i<k\leq j} C_{ik} + F_{k+1,j}\right\}$$

$$C_{ij} = \min\left\{\mathcal{H}(i,j),\ \min_{i<k<l<j} C_{kl} + \mathcal{I}(i,j;k,l),\ \min_{i<u<j} M_{i+1,u} + M^1_{u+1,j-1} + a\right\}$$

$$M_{ij} = \min\left\{\min_{i<u<j}(u-i+1)c + C_{u+1,j}+b,\ \ \min_{i<u<j} M_{i,u} + C_{u+1,j}+b,\ \ M_{i,j-1}+c\right\}$$
$$M^1_{ij} = \min\left\{M^1_{i,j-1} + c,\ \ C_{ij} + b\right\} \tag{3}$$

The initialization is $F_{ii}=0$, $C_{ii}=M_{ii}=M^1_{ii}=+\infty$. Memory consumption is quadratic in sequence length.[2] In our current implementation, the total length of interior loops (second-term in the recursion for C) is limited such that $(j-l-1)+(k-i-1)\leq 30$. This restriction leads to a cubic run-time. Under certain conditions on the interior loop energies, which are realized for the Turner energy model, a cubic algorithm can also be obtained without a length restriction for interior loops [15].

The corresponding recursions for the partition functions (Z_{ij}, Z^B_{ij}, Z^M_{ij}, Z^{M1}_{ij}) are obtained by replacing minimum operations with sums and additions with multiplications [18]:

$$\begin{aligned}
Z_{ij} &= Z_{i+1,j} + \sum_{i<k\leq j} Z^B_{ik} Z_{k+1,j}\\
Z^B_{ij} &= e^{-\beta\mathcal{H}(i,j)} + \sum_{i<k<l<j} Z^B_{kl} e^{-\beta\mathcal{I}(i,j;k,l)} + \sum_{i<u<j} Z^M_{i+1,u} Z^{M1}_{u+1,j-1} e^{-\beta a}\\
Z^M_{ij} &= \sum_{i<u<j} e^{-\beta(u-i+1)c} Z^M_{u+1,j} + \sum_{i<u<j} Z^M_{i,u} Z^B_{u+1,j} e^{-\beta b} + Z^M_{i,j-1} e^{-\beta c}\\
Z^{M1}_{ij} &= Z^{M1}_{i,j-1} e^{-\beta c} + Z^B_{ij} e^{-\beta b}\\
Z_{ii} &= 1,\ Z^B_{ii} = Z^M_{ii} = Z^{M1}_{ii} = 0
\end{aligned} \tag{4}$$

As usual, $\beta = 1/RT$ is the inverse thermal energy. Uniqueness of the substructure decomposition is crucial in this case since the partition function is a weighted count over all structures.

4 Canonical secondary structures

A *canonical* secondary structure does not contain isolated ("lonely") basepairs, i.e., pairs that are not stacked on another pair. In Fig. 1 we compare the structure decomposition of the usual, unconstrained, secondary structures with the decomposition of constrained structures. Restrictions of the energy minimization algorithms to this subclass have already been included in the `Vienna RNA Package`. The requirement of a unique structural decomposition, which can be waived in the case energy minimization, however, requires some algorithmic changes in partition function approaches.

Let Z^*_{ij} denote the partition function over all structures that are enclosed by a basepair and that become canonical when enclosed by an additional exterior basepair $(i-1, j+1)$; this corresponds to the structures marked (C) in Fig. 1. The usual

[2] In most applications the matrices M^1 and F arrays can be replaced by linear arrays that store only the current and previous row or column.

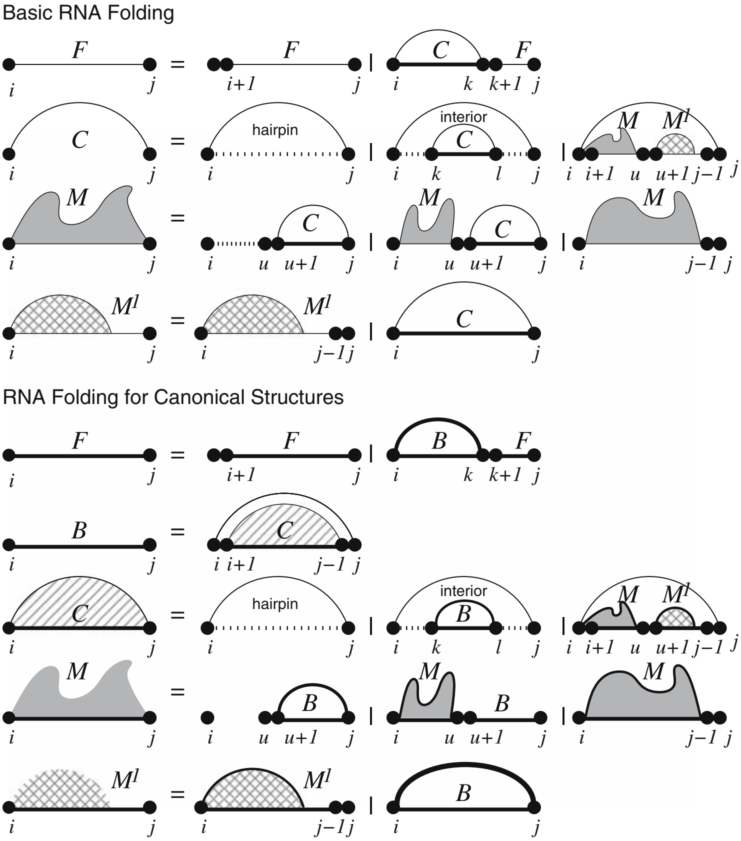

Fig. 1 Comparison of the structure decompositions used in RNA folding algorithms for unconstrained structures (*top*) and canonical structures (*below*). Thin arcs denote substructures that are closed by a possibly lonely basepair but otherwise canonical. *Thick arcs* mark canonical substructures enclosed by a stacked pair. We note that the decomposition stays almost the same provided one reinterprets the closed structures as canonical structures enclosed by a basepair (B). The only difference is the representation of B as an enclosed structure (C) that becomes canonical when enclosed by a single outer basepair

forward recursions (4) remain almost unchanged: the recursion for Z^{*}_{ij} is precisely as given for Z^{B}_{ij} in Eq. 4 above, while the Z^{B}_{ij}

$$Z^{B}_{ij} = Z^{*}_{i+1,j-1} e^{-\beta\mathcal{I}(i,j,i+1,j-1)} \tag{5}$$

is computed by adding the stacking contribution to the enclosed structure. The multiloop decomposition remains unchanged.

The outward recursions are bit more problematic, however. Denote by $\widehat{Z}^{B}_{ij}$ the partition function over all canonical structures outside the pair (i, j), and let $\widehat{Z}^{*}_{ij}$ be the partition function over all structures outside (i, j) that become canonical when enclosed by the additional basepair $(i-1, j+1)$. Again there is little change compared to the "usual" backwards recursion as implemented in the `Vienna RNA Package`.

The main difference is that the case of a stacking interaction is different here, since the sub-structures excluded by (i, j) need not be canonical. It is sufficient that $(i-1, j+1)$ is paired. Therefore, the stacking term depends on $\widehat{Z}^*$ instead of Z^B. We thus obtain the following recursions:

$$\begin{aligned}\widehat{Z}^*_{ij} &= \underbrace{Z_{1,i-1}Z_{j+1,n}}_{\text{exterior}} + \sum_{\substack{h<i<j<l\\(h,l)\neq(i-1,j+1)}} \underbrace{\widehat{Z}^B_{h,l}e^{-\beta\mathcal{I}(h,l;i,j)}}_{\text{interior loop}} + \underbrace{\widehat{Z}^*_{i-1,j+1}e^{-\beta\mathcal{I}(i-1,j+1;i,j)}}_{\text{stack}} + \sum_{h<i<j<l}\widehat{Z}^B_{h,l}\\ &\quad\times\left\{\underbrace{e^{-\beta(i-h-1)c}Z^M_{j+1,l-1}}_{\text{multiloop right}} + \underbrace{Z^M_{h+1,i-1}e^{-\beta(l-j-1)c}}_{\text{multiloop left}} + \underbrace{Z^M_{h+1,i-1}Z^M_{j+1,l-1}}_{\text{multiloop both}}\right\}\\ \widehat{Z}^B_{ij} &= \widehat{Z}^*_{i-1,j+1}e^{-\beta\mathcal{I}(i-1,j+1;i,j)}\end{aligned} \tag{6}$$

From these quantities we can immediately compute the probability of the stack $(i, j; i+1, j-1)$ as

$$P[i, j] = \frac{Z^B_{i,j}\widehat{Z}^B_{i+1,j-1}}{Ze^{-\beta\mathcal{I}(i,j;i+1,j+1)}} \tag{7}$$

Note that we have to divide by the stacking contribution since this factor is contained in both $\widehat{Z}^B_{i,j}$ and $Z^B_{i+1,j-1}$. The individual base pairing probabilities can be computed as follows:

$$P_{ij} = \frac{1}{Z}\left[Z^*_{ij}\widehat{Z}^B_{ij} + Z^B_{ij}\widehat{Z}^*_{ij} - Z^B_{ij}\widehat{Z}^B_{ij}\right] \tag{8}$$

The last term accounts for the fact that in the first two terms we have twice included the structures that are canonical both inside and outside the pair (i, j).

The partition function algorithm for canonical structures has been included in the latest version of the `Vienna RNA Package`. Figure 2 gives an example in which the restriction to canonical basepairs increases the accuracy of the structure prediction. Since fewer basepairs have to be considered in the canonical case, there is also a moderate performance gain.

5 Probabilities for stacks

The probability of a structural motif can be computed easily once the Z^B table has been filled. Of particular interest are the probabilities for stacks and helices, which we derive below. The conditional probability for the stacking of two pairs is

$$\text{Prob}\,[(i+1, j-1)|(i, j)] = \frac{Z^B_{i+1,j-1}}{Z^B_{i,j}}e^{-\beta\mathcal{I}(i,j;i+1,j-1)} \tag{9}$$

The probability for a stack of length at least ℓ inside the pair (i, j) can be determined by a similar expression:

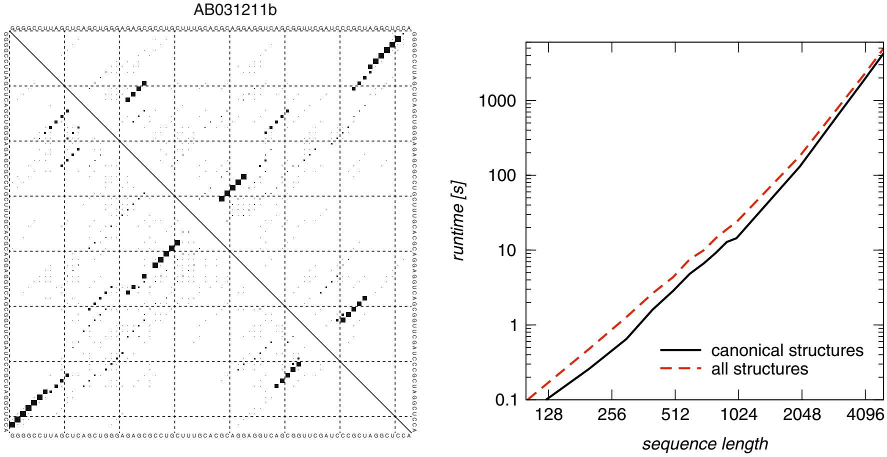

Fig. 2 Canonical versus general secondary structures. *Left-hand side panel* Comparison of base pairing probabilities of a tRNA (Genbank Accession number AB031211) with isolated pairs (*lower left*) and without (*upper right*) triangular matrix. In this example, the characteristic clover leaf shape is more apparent in the distribution of canonical structures. *Right-hand side panel* comparison of the performance of the folding and backtracing algorithm as implemented in the current version of the `Vienna RNA Package`. As expected, computing canonical structures is cheaper by a constant factor since fewer potential base pairs have to be considered

$$P_{i,j}^{(\ell)} = P_{ij} \frac{Z_{i+\ell-1,j-\ell+1}^{B}}{Z_{i,j}^{B}} \prod_{u=0}^{\ell-2} e^{-\beta \mathcal{I}(i+u,j-u;i+u+1,j-u-1)} \tag{10}$$

Equivalently, Eq. (11), expresses $P_{i,j}^{(\ell)}$ in terms of the conditional stacking probabilities

$$P_{i,j}^{(\ell)} = P_{ij} \prod_{u=0}^{\ell-2} \text{Prob}\,[(i+u+1, j-u-1)|(i+u, j-u)] \tag{11}$$

The probability that the stack extends exactly ℓ basepairs from (i, j) can be computed as

$$P_{ij}^{[\ell]} = P_{ij}^{(\ell)} - P_{ij}^{(\ell+1)} \tag{12}$$

We can also rather easily enforce that (i, j) is the first pair of the stack. We start from Bayes' equation:

$$\text{Prob}\,[(i+1, j-1)|(i, j)]\, P_{ij} = \text{Prob}\,[(i, j)|(i+1, j-1)]\, P_{i+1,j-1} \tag{13}$$

Thus, the probability that $(i-1, j+1)$ is not a pair *given* that (i, j) is paired is

$$\text{Prob}\,[\neg(i-1, j+1)|(i, j)] = 1 - \text{Prob}\,[(i, j)|(i-1, j+1)]\, \frac{P_{i-1,j+1}}{P_{i,j}} \tag{14}$$

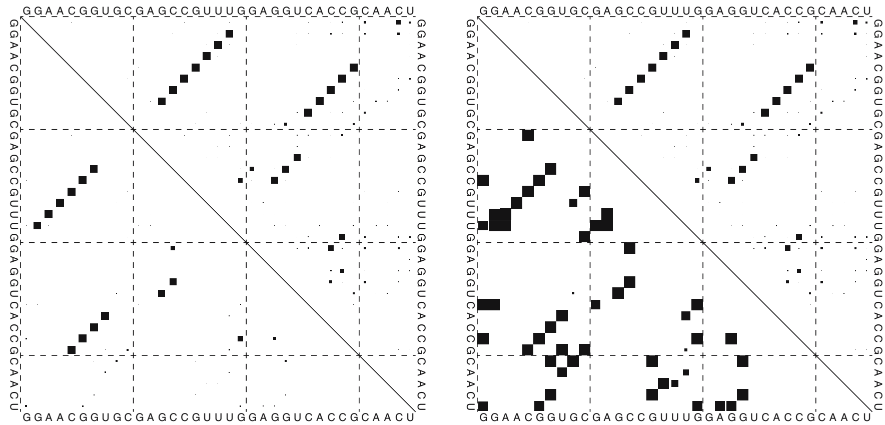

Fig. 3 Comparison of base pairing probabilities P_{ij} (*upper right of both dot plots*), stacking probabilities $P_{ij}^{(2)}$ (*lower left of left dot plot*), and conditional stacking probabilities Prob $[(i, j)|(i+1, j-1)]$ (*lower left of right dot plot*) for a short artificial sequence. Even for low probability helices, conditional stacking probabilities are often close to 1, which is due to the cooperativity of helix formation

The probability $\underline{\mathrm{P}}_{ij}^{(\ell)}$ of stack of length at least ℓ with the first pair (i, j) is thus

$$
\begin{aligned}
\underline{\mathrm{P}}_{ij}^{(\ell)} &= \mathrm{Prob}\,[\neg(i-1, j+1)|(i, j)]\, P_{ij}^{(\ell)} \\
&= \left[1 - \frac{P_{i-1,j+1}}{P_{i,j}} \frac{Z_{i,j}^{B}}{Z_{i-1,j+1}^{B}} e^{-\beta\mathcal{I}(i-1,j+1;i,j)}\right] P_{i,j} \frac{Z_{i+\ell-1,j-\ell+1}^{B}}{Z_{i,j}^{B}} \prod_{u=0}^{\ell-2} e^{-\beta\mathcal{I}(i+u,j-u;i+u+1,j-u-1)} \\
&= P_{ij}^{(\ell)} - P_{i-1,j+1} \frac{Z_{i+\ell-1,j-\ell+1}^{B}}{Z_{i-1,j+j}^{B}} e^{-\beta\mathcal{I}(i-1,j+1;i,j)} \prod_{u=0}^{\ell-2} e^{-\beta\mathcal{I}(i+u,j-u;i+u+1,j-u-1)} \qquad (15)
\end{aligned}
$$

As we would expect, we obtain

$$
\underline{\mathrm{P}}_{ij}^{(\ell)} = P_{ij}^{(\ell)} - P_{i-1,j+1}^{(\ell+1)} \qquad (16)
$$

Finally, the probability for a stack of length exactly ℓ with first pair (i, j) is

$$
\begin{aligned}
\underline{\mathrm{P}}_{ij}^{[\ell]} &= \underline{\mathrm{P}}_{ij}^{(\ell)} - \underline{\mathrm{P}}_{ij}^{(\ell+1)} \\
&= P_{ij}^{(\ell)} - P_{i-1,j+1}^{(\ell+1)} - P_{ij}^{(\ell+1)} + P_{i-1,j+1}^{(\ell+2)} \\
&= P_{ij}^{[\ell]} - P_{i-1,j+1}^{[\ell+1]} \qquad (17)
\end{aligned}
$$

Figure 3 compares Prob $[(i, j)|(i+1, j-1)]$ with P_{ij}. In general we see a similar pattern, indicating that stacked basepairs are highly correlated.

6 Structural alignment with stacking

In many applications it is convenient to avoid the full loop based energy model and to use a scoring scheme based on pre-computed pair probabilities instead. Essentially, such an approach treats pair probabilities as if they were independent. Including the effect of basepair stacking should improve this situation without introducing much additional complexity.

As an example we consider the structural alignment program LocARNA [24]. Similar to PMcomp [8], LocARNA implements a Nussinov-style version of the Sankoff algorithm [20] in which scores for basepair matches are pre-computed using thermodynamic folding algorithms. Lin et al. [14] define a hierarchy of problems for the general edit distance of two RNA sequences with structures. Each of the sequences is either annotated with no structure (plain), a fixed pseudoknot-free structure (nested), or a fixed arbitrary structure (crossing). They present algorithms for the case nested/plain and crossing/nested and show that the crossing/plain problem is MAX SNP-hard. Our algorithm is structurally similar to their $\mathcal{O}(n^2, m^2)$ time algorithm for the crossing/nested problem.

Let M_{ijkl} be the optimal score of an alignment of the subsequences $A[i..j]$ and $B[k..l]$. Furthermore, D_{ijkl} is the optimal score of an alignment of the subsequences $A[i..j]$ and $B[k..l]$ with matching basepairs (arcs) (i, j) and (k, l). With a gap penalty γ, and sequence dependent scores σ for unpaired (mis)match and α for arc matches one finds the following recursions (see also Fig. 4):

$$M_{i,j,k,l} = \max \begin{cases} M_{i,j-1,k,l} + \gamma \\ M_{i,j,k,l-1} + \gamma \\ M_{i,j-1,k,l-1} + \sigma(i, j) \\ \max\limits_{\substack{i<p<j \\ k<q<l}} \left(M_{i,p-1,k,q-1} + D_{p,j,q,l}\right) \end{cases} \tag{18}$$

$$D_{i,j,k,l} = M_{i+1,j-1,k+1,l-1} + \alpha(i, j, k, l)$$

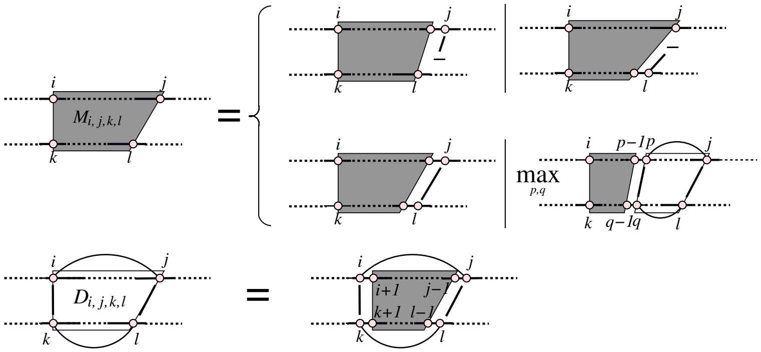

Fig. 4 Recursion scheme for sequence–structure alignment in illustration of Eq. 19. There are four distinct cases for obtaining the scores in the matrix M and a single derivation for the entries of D

In order to incorporate stacking we have to introduce an additional set of parameters α' that score the extension of a series of adjacent arcs. Furthermore, we have to distinguish the extension of an arc-enclosed alignment from the formation of an individual enclosing arc:

$$D_{i,j,k,l} = \max \begin{cases} \alpha(i,j,k,l) + \max \begin{cases} M_{i+1,j-2,k+1,l-1} + \gamma \\ M_{i+1,j-1,k+1,l-2} + \gamma \\ M_{i+1,j-2,k+1,l-2} + \sigma(i-1,j-1) \\ \max\limits_{\substack{i+1<p<l-1 \\ k+1<q<l}} \left(M_{i+1,p-1,j+1,q-1} + D_{p,j-1,q,l-1} \right) \end{cases} \\ \alpha'(i,j,k,l) + D_{i+1,j-1,k+1,l-1} \end{cases} \tag{19}$$

The global alignment score is $M_{1,|A|,1,|B|}$.

For local alignment with a score threshold T we have to include the *"un-aligned"* state with scores $M_{i,j,k,l} = 0$ as additional alternative in Eq. (18). Backtracking then starts at local maxima of the $M_{1,j,1,l}$ matrix that exceed the threshold T.

In our implementation, only the D array (for pairs of matched basepairs only) and the values $M_{1,j,1,l}$ are stored permanently. Since the number of basepairs in the input structures is linear in sequence length provided only pairs with a minimum pairing probability p_0 are considered, the algorithm requires $\mathcal{O}(|A| \times |B|)$ memory, see [24] for details.

A natural way of defining the scoring functions α and α' is to use appropriately scaled logarithms of base pairing and conditional base pairing probabilities, respectively. For molecule A these quantities are

$$\begin{aligned} \Psi^{A}_{ij} &= \max\{0, \log(P^{A}_{ij}/p_0)/\log(1/p_0)\} \\ \Psi^{A*}_{ij} &= \max\{0, \log(\mathrm{Prob}\,[(i,j)|(i+1,j-1)]\,/p_0)/\log(1/p_0)\}, \end{aligned} \tag{20}$$

analogous expressions Ψ^{B}_{ij} and Ψ^{B*}_{ij} hold for the second molecule. The user-defined parameter p_0 is the minimum pairing probability of basepair that considered to be part of the input structure. The most straight-foward cost model then assumes $\alpha(i,j,k,l) = \Psi^{A}_{ij} + \Psi^{B}_{kl}$ and $\alpha'(i,j,k,l) = \Psi^{A*}_{ij} + \Psi^{B*}_{kl}$, respectively.

Thus, only the base matches A_i—B_j that are *not* part of matching basepairs (arcs) are given a sequence dependent score $\sigma(i,j)$ [third entries in eqs. (18) and (19)], respectively. Both LocARNA and PMComp also use a sequence-specific component $\tau(i,j,k,l)$ for matching basepairs in the definition of $\alpha(i,j,k,l)$. For this purpose one can utilize e.g. the RIBOSUM basepair substitution scores [12], a 16 × 16 matrix derived from substitution probabilities between all possible pairs of nucleotides.

The question remains, how *arc breaking* should be treated, which occurs whenever an input basepair in one sequence is not matched against another basepair. If two known secondary structures instead of basepair probability matrices are used as input, it is biophysically meaningful to explicitly penalize arc breaking, see e.g. [1,11]). These score could be added as an extra row and column to the RIBOSUM model. However, the recursion scheme above or variants thereof can only be applied if the score for

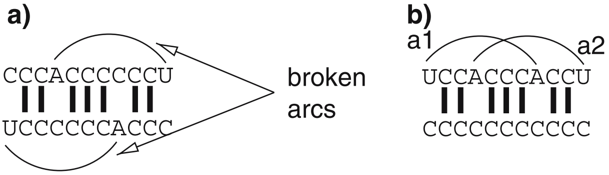

Fig. 5 **a** Crossing interaction produced by *broken arcs*. **b** Both arcs are broken here since they are not part of a basepair match

arc breaking has a special form, namely the left and right ends of the broken arcs are scored independently. The reason is that considering a broken arc as an entity would produce crossing interactions as in Fig. 5a for example. In contrast, tree-editing like algorithms (see [6] and the references therein) treat arcs as entities that can be added or deleted only as a whole. In such models it is natural to have scores that depend on the type of basepair; in general, these will not be the sum of contributions for the paired bases.

In the case of `LocARNA` and `PMcomp` where basepair probability matrices are given as an input, scoring of arc breaking does not make sense. Consider the case where we have two crossing arcs a_1 and a_2 in the basepair probability matrices having a similar probability as in Fig. 5b. In this case, we simply do not know which of the two arcs are part of the real structure of the first sequence. Furthermore, both arcs are exclusive. Hence, we could score at most one arc breaking, and we simply do not have any information which one should be scored.

7 Using structural alignment algorithms for gene finding

A natural application of local structural alignments is homology based RNA gene finding. To this end, a structure-annotated genomic DNA can be efficiently computed by `RNALfold` [10] or `RNAplfold` [2]. These programs compute local RNA structure by restricting the maximum span of basepairs (i, j) to $|j - i| < L$.

Utilizing `LocARNA` for sequence/structure homology search requires several modifications. We first extended the algorithm so that its backtracking routine not only returns the optimal local alignment but the k-best local alignments. This was achieved by an interval splitting strategy on top of the basic algorithm. For complexity reasons, we process overlapping windows (size 1,000,000) instead of complete chromosomes. For each window, the RNA search pattern is locally aligned to the window. Local folding allows us to restrict the maximal reach basepairs L to the size of the search pattern.

The following example is intended to demonstrate that local sequence/structure alignments can indeed be utilized for genome-wide homology searches at acceptable computational costs.

For our application to box H/ACA snoRNAs we furthermore modify the sequence scoring scheme so that in/dels in the conserved sequence boxes can be prohibited. As a typical non-coding RNA, snoRNA is largely characterized by structure and shows

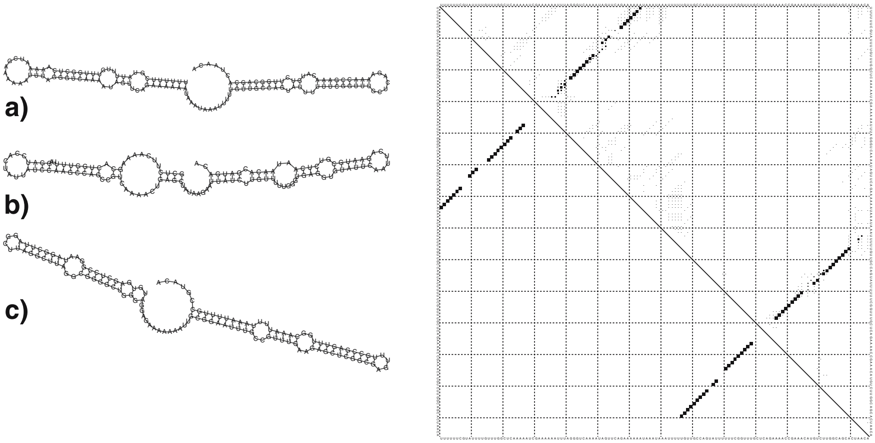

Fig. 6 *Left* secondary structure of the best three snoRNA candidates. *Right* Dot plot produced by `RNAfold` of first candidate (**a**). Candidates show specific snoRNA features like: two stems of approximately equal length, ACA-box at $3'$ end and H-box (consensus: ANANNA) between both stems

only small sequence conservation. For the example in Fig. 6 we derived the snoRNA pattern from the human U65 snoRNA. First, we compute a pair probability matrix using McCaskill's algorithm [18] (`RNAfold -p`). Due to the very low sequence conservation, we ignore most of the sequence information. We enforce, however, exact matches of the H-box (motif `ANANNA` between the two stems) and the ACA-box (motif `ACA` at the $3'$ end). We conduct our search on chromosome I of *Caenorhabditis elegans*, which has a size of approximately 16 megabases. The entire computation took approximately 8 h on a Pentium 4 3 GHz, 4 GB RAM running Linux, about half of the time was used by `locarna`, the other half was used to pre-compute the base pairing probabilities with `RNAplfold` ($L = 138$, the size of the query pattern).

We determined the 400, regarding to the locarna-score, top-ranking local alignments of the snoRNA pattern. Three good examples are shown in Fig. 6. We then folded all hits using `RNAfold` and selected a top-list of 78 snoRNA candidates whose minimum energy structure resembles the typical shape of box H/ACA snoRNAs. In this set we find the single box H/ACA snoRNA *cer-3* that is reported for Chr.I in the "Wormbook" [22]. In addition, we recover two of the nine novel ncRNAs on Chr.I that have been reported as likely box H/ACA snoRNAs in recent experimental screens [3,25].

The `LocARNA` algorithm [24] could be modified to a true scanning variant in which memory requirements are independent of the subject database (apart from storing the input itself). The idea is similar to the `RNALfold`-style "scanning" algorithms, another variant of which is described in Sect. 8 below. We observe that the length w of any local alignment with a score $T \geq 0$ for the query sequence is bounded since the maximal possible alignment score must exceed the prescribed threshold T. One easily derives the estimate

$$0 \leq T \leq M(\max \sigma + \max(\alpha, \alpha')/2) - |w - M|\gamma \tag{21}$$

where M is the length of query sequence and L is the maximal span of a basepair in subject sequence. It follows that

$$w \le M + [M(\max\sigma + \max(\alpha, \alpha')/2) - T]/\gamma \tag{22}$$

It is therefore sufficient to store a window of size w backward from the current position l in the subject database and to start backtracking from within this "active" window of size $M \times w$. It is easy to avoid producing groups of similar alignments by enforcing a minimum distance in the database between the start positions for consecutive calls to the backtracking routine.

8 A scanning version of RNAup

Regulatory RNAs often interact with a target RNA by forming inter-molecular helices. Duplex formation, e.g. between an siRNA and its mRNA target, is facilitated if the binding site is accessible, i.e. in an unpaired region. To predict possible binding sites it is therefore of interest to compute the probability $P^0[i, j]$ that a sequence interval $[i, j]$ is unpaired, as is done in the RNAup program [19]. The probability that small regions are unpaired is also frequently computed by sampling structures from the Boltzmann distribution using e.g. Sfold [4] or the stochastic backtracking option in RNAfold. This introduces sampling errors, especially when the accessibility is small. In contrast, RNAup computes $P^0[i, j]$ directly by dynamic programming. However, since RNAup folds the complete molecule, its CPU requirements still scale as $\mathcal{O}(n^3)$, making it unsuitable for very long sequences. Instead, one can use a windowing technique as an approximation. As in the case of RNAplfold it is of interest to replace explicit computations of individual sequence windows by a "scanning approach" that directly computes the average over all pertinent sequence windows of a fixed length L.

As shown in [19], the values of $P^0[i, j]$ can be computed from the equation

$$P^0[i, j] = \frac{Z_{1,i-1}Z_{j+1,n}}{Z_n} + \sum_{h<i,j<l} P_{h,l}\text{Prob}\,[[i, j]|(h, l)]\,, \tag{23}$$

where Prob $[[i, j]|(h, l)]$ is the probability that $[i, j]$ is an unpaired region within the loop with closing pair (h, l). Note that this probability is independent of the structures outside the pair (h, l).

As in RNAplfold [2] we define the average over all folding windows of the probability that (i, j) is paired:

$$\pi^L_{ij} = \frac{1}{L - (j - i) + 1} \sum_{u=j-L}^{i} P^{u,L}_{ij}. \tag{24}$$

where $P^{u,L}_{ij}$ is the probability that i, j is paired in a window of size L starting at u. Now we compute the average $\pi^0[i, j]$ of the $P^0[i, j]$ values over all windows of length L containing $[i, j]$, i.e.,

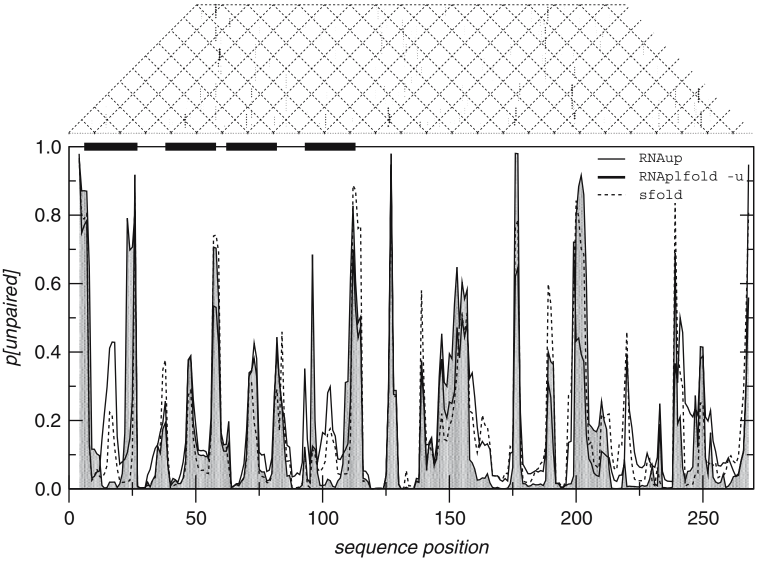

Fig. 7 Dot plot and probability of four bases being unpaired in an artificially designed 3′ UTR targeted by the *cxcr4* siRNA [5]. Accessibilities were computed using three methods: `RNAplfold -u 4 -L 100 -w 120` (*thick line*), `RNAup` (*shaded area*), and `Sfold` (*thin dotted line*, also using a maximum span of 100). Note that for $L = n$ `RNAup` and `RNAplfold` would give identical results. *Bars* above the accessibility plot denote the binding sites. Note that the highest probabilities of being unpaired are observed at the "seed-sites", i.e., at the 3′ end the binding sites

$$\begin{aligned}\pi^0[i,j] &= \frac{1}{L-(j-i)+1}\sum_{u=j-L}^{i} P^0[i,j]\\ &= \frac{1}{L-(j-i)+1}\sum_{u=j-L}^{i}\frac{Z^{u,L}_{1,i-1}Z^{u,L}_{j+1,n}}{Z^{u,L}_{1,n}}\\ &\quad+\sum_{h=j-L}^{i-1}\sum_{l=j+1}^{i+L}\frac{L-(h-l)+1}{L-(j-i)+1}\pi^L_{hl}\mathrm{Prob}\,[[i,j]|(h,l)] \qquad (25)\end{aligned}$$

Since

$$\mathrm{Prob}\,[[i,j]|(h,l)] = Z_{hl}[i,j]/Z^B_{ij} \qquad (26)$$

is independent of the folding window as long as $[h,l] \subseteq [u, u+L-1]$, and the computation of $Z_{hl}[i,j]$ requires only partition function entries in the interval $[h,l]$, we have here a way of combining `RNAup` and `RNAplfold`.

This algorithm can be used e.g. to obtain a quick estimate of the availability of putative binding sites for miRNA target prediction. It seems reasonable to assume that in order for the miRNA to initiate binding, at least a small part of the binding site must be unpaired and thus accessible for the initial contacts. Target site accessibility thus has been used to improve the design of efficient siRNAs [21] using `Sfold`. Figure 7 shows an application to an siRNA example from the literature. The scanning version of RNAplfold offers two advantages: (1) instead of using a sampling approach, the target site accessibility is computed exactly, and (2) RNAplfold can handle sequences of the the size of a chromosome, while the Sfold web server limits sequence length to 5,000.

Acknowledgments This work has been funded, in part, by the Austrian GEN-AU projects "bioinformatics integration network II" and "non-coding RNA", the German DFG Bioinformatics Initiative BIZ-6/1-2, and the Federal Ministry of Education and Research in the context of the Jena Center for Bioinformatics. We would like to thank to Elena Rivas and Eric Westhof for organizing, and the NIH and the ESF for funding the 2006 Benasque meeting. This work originated as a collaborative effort to which all its authors contributed equally. None of us wishes to claim *First Author* or *Senior Author* credit for this publication. In response to the increasingly prevalent abuse of automatically generated bibliometric measures, we asked our fictional friend Athanasius F. Bompfünewerer to assume the position of first author and to collect all those automatically generated "Brownie Points".

References

1. Backofen, R., Will, S.: Local sequence–structure motifs in RNA. J. Bioinform. Comput. Biol. **2**, 681–698 (2004)
2. Bernhart, S., Hofacker, I.L., Stadler, P.F.: Local RNA base pairing probabilities in large sequences. Bioinformatics **22**, 614–615 (2006)
3. Deng, W., Zhu, X., Skogerbø, G., Zhao, Y., Fu, Z., Wang, Y., He Housheng Cai, L., Sun, H., Liu, C., Li, B.L., Bai, B., Wang, J., Cui, Y., Jai, D., Wang, Y., Du, D., Chen, R.: Organisation of the *Caenorhabditis elegans* small noncoding transiptome: genomic features, biogenesis and expression. Genome Res. **16**, 30–36 (2006)
4. Ding, Y., Chan, C.Y., Lawrence, C.E.: Sfold web server for statistical folding and rational design of nucleic acids. Nucleic Acids Res. **32**(Web Server issue), W135–W141 (2004)
5. Doench, J.G., Sharp, P.A.: of mioRNA target selection in translational repression. Genes Dev. **18**, 504–511 (2004)
6. Dulucq, S., Tichit, L.: Secondary structure comparison: exact analysis of the Zhang-Shasha tree-edit algorithm. Theor. Comput. Sci. **306**, 471–484 (2003)
7. Hofacker, I.L.: Vienna RNA secondary structure server. Nucleic Acids Res. **31**, 3429–3431 (2003)
8. Hofacker, I.L., Bernhart, S.H.F., Stadler, P.F.: Alignment of RNA base pairing probability matrices. Bioinformatics **20**, 2222–2227 (2004)
9. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. Monatsh. Chem. **125**(2), 167–188 (1994)
10. Hofacker, I.L., Priwitzer, B., Stadler, P.F.: Prediction of locally stable RNA secondary structures for genome-wide surveys. Bioinformatics **20**, 191–198 (2004)
11. Jiang, T., Lin, G., Ma, B., Zhang, K.: A general edit distance between RNA structures. J. Comput. Biol. **9**, 371–88 (2002)
12. Klein, R.J., Eddy, S.R.: RSEARCH: finding homologs of single structured RNA sequences. BMC Bioinform **4**(1), 44 (2003)
13. Leydold, J., Stadler, P.F.: Minimal cycle basis of outerplanar graphs. Elec. J. Comb. **5**, 209–222 [R16: 14 p.] (1998)
14. Lin, G.H., Ma, B., Zhang, K.: Edit distance between two RNA structures. In: Proceedings of the 5th Annual International Conference on Computational Biology RECOMB01, pp. 211–220. ACM Press (2001)
15. Lyngsø, R.B., Zuker, M., Pedersen, C.N.: Fast evaluation of internal loops in RNA secondary structure prediction. Bioinformatics **15**, 440–445 (1999)
16. Mathews, D., Sabina, J., Zuker, M., Turner, H.: Expanded sequence dependence of thermodynamic parameters provides robust prediction of RNA secondary structure. J. Mol. Biol. **288**, 911–940 (1999)
17. Mathews, D.H., Disney, M.D., Childs, J.L., Schroeder, S.J., Zuker, M., Turner, D.H.: Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc. Natl. Acad. Sci. USA **101**, 7287–7292 (2004)

18. McCaskill, J.S.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers **29**, 1105–1119 (1990)
19. Mückstein, U., Tafer, H., Hackermüller, J., Bernhart, S., Stadler, P.F., Hofacker, I.L.: Thermodynamics of RNA-RNA binding. Bioinformatics **22**, 1177–1182 (2006)
20. Sankoff, D.: Simultaneous solution of the RNA folding, alignment, and proto-sequence problems. SIAM J. Appl. Math. **45**, 810–825 (1985)
21. Shao, Y., Wu, Y., Chan, C.Y., Mcdonough, K., Ding, Y.: Rational design and rapid seening of antisense oligonucleotides for prokaryotic gene modulation. Nucleic Acids Res. **34**, 5660–5669 (2006)
22. Stricklin, S.L., Griffiths-Jones, S., Eddy, S.R.: C. elegans noncoding RNA genes. WormBook doi:10.1895/wormbook.1.7.1. http://www.wormbook.org/chapters/www_noncodingRNA/noncodingRNA.html (2005)
23. Tinoco, I., Uhlenbeck, O.C., Levine, M.D.: Estimation of secondary structure in ribonucleic acids. Nature **230**, 362–367 (1971)
24. Will, S., Reiche, K., Hofacker, I.L., Stadler, P.F., Backofen, R.: Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. PLoS Comp. Biol. **3**, e65 (2007)
25. Zemann, A., op de Bekke, A., Kiefmann, M., Brosius, J., Schmitz, J.: Evolution of small nucleolar RNAs in nematodes. Nucleic Acids Res. **34**, 2676–2685 (2006)
26. Zuker, M., Sankoff, D.: RNA secondary structures and their prediction. Bull. Math. Biol. **46**, 591–621 (1984)
27. Zuker, M., Stiegler, P.: Optimal computer folding of larger RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**, 133–148 (1981)