Efficient Algorithms for Co-folding of Multiple RNAs

Ronny Lorenz[1], Christoph Flamm[1], Ivo L. Hofacker[1,2], and Peter F. Stadler[1,3,4,5,6](✉)

[1] Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, 1090 Wien, Austria
{ronny,xtof,ivo,studla}@tbi.univie.ac.at

[2] Bioinformatics and Computational Biology, Faculty of Computer Science, University of Vienna, Währingerstraße 29, 1090 Wien, Austria

[3] Bioinformatics Group, Department of Computer Science, Interdisciplinary Center for Bioinformatics, and Competence Center for Scalable Data Services and Solutions Dresden/Leipzig, Universität Leipzig, Härtelstraße 16-18, 04107 Leipzig, Germany
studla@bioinf.uni-leipzig.de

[4] Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, 04103 Leipzig, Germany

[5] Facultad de Ciencias, Universidad National de Colombia, Sede Bogotá, Colombia

[6] Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA

Abstract. The simplest class of structures formed by $N \geq 2$ interacting RNAs consists of all crossing-free base pairs formed over linear arrangements of the constituent RNA sequences. For each permutation of the N strands the structure prediction problem is algorithmically very similar – but not identical – to folding of a single, contiguous RNA. The differences arise from two sources: First, "nicks", i.e., the transitions from one to the next piece of RNA, need to be treated with special care. Second, the connectedness of the structures needs to guaranteed. For the forward recursions, i.e., the computation of folding energies or partition functions, these modifications are rather straightforward and retain the cubic time complexity of the well-known folding algorithms. This is not the case for a straightforward implementation of the corresponding outside recursion, which becomes quartic. Cubic running times, however, can be restored by introducing linear-size auxiliary arrays. Asymptotically, the extra effort over the corresponding algorithms for a single RNA sequence of the same length is negligible in both time and space. An implementation within the framework of the `ViennaRNA` package conforms to the theoretical performance bounds and provides access to several algorithmic variants, include the handling of user-defined hard and soft constraints.

An earlier version of this contribution appeared in the Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies – Volume 3: Bioinformatics [26]. This work was supported in part by the German Federal Ministry of Education and Research (BMBF, project no. 031A538A, de.NBI-RBC, to PFS and project no. 031L0164C, RNAProNet, to PFS), and the Austrian science fund FWF (project no. I 2874 "Prediction of RNA-RNA interactions", project no. F 80 "RNAdeco" to ILH).

X. Ye et al. (Eds.): BIOSTEC 2020, CCIS 1400, pp. 193–214, 2021.
https://doi.org/10.1007/978-3-030-72379-8_10

Keywords: RNA folding · Interacting RNAs · Partition function · Outside recursion · Cubic-time algorithm

1 Introduction

RNA-RNA interactions play an important role in both eukaryotic [17] and procaryotic [14] gene regulation. In eukaryotes, RNA interference involves the binding of small RNAs from diverse sources to longer RNAs, usually leading to degradation. Post-transcriptional gene silencing by microRNAs is just one of the many variations on this theme. Both small interfering RNAs (siRNAs) and long non-coding RNAs (lncRNAs) are also involved in the regulation of splicing and the biogenesis of other RNAs, including microRNAs. RNA sponges, usually lncRNAs, sequester specific miRNAs to revert their silencing effects. The binding of lncRNAs such as *TINCR* to an mRNA can also contribute to the control of translation. In procaryotes, a large number of diverse and often lineage specific small RNAs (sRNAs) act as regulators of translation by binding to their target mRNAs inducing structural changes. In all these cases the RNAs interact by forming intermolecular base pairs. Such hetero-duplexes also form between spliceosomal RNAs during the assembly of the spliceosome and are crucial for the correct splicing. The maturation of ribosomal RNAs (rRNAs) and spliceosomal RNAs (snRNAs) requires chemical modifications, most of which are introduced by snoRNPs, which rely on the specific binding of small nucleolar RNAs (snoRNAs) with their rRNA or snRNA target.

An abundance of RNA-RNA interactions was recently reported by transcriptome-wide experiments [16]. This was not entirely unexpected as much earlier computations studies already found statistical evidence for extensive RNA-RNA interaction networks [38]. It is likely, therefore, that complexes composed of more than two RNAs may play important roles similar to the well-established protein complexes. In addition, higher order complexes have already be considered extensively in synthetic biology [8,21]. The prediction and analysis of multi-component RNA complexes thus has become an important task in computational biology, in particular in the context of strand displacement systems [3].

Many aspects of RNA structures, including their thermodynamic properties are well represented by their secondary structures, i.e., discrete base pairs. These already capture the dominating stabilizing and destabilizing contribution: the stacking of base pairs with in helical stem regions and the conformational entropy loss of unpaired regions relative to unconstrained RNA chains. These energetic contributions are compiled in the “loop-based” standard energy model [39]. Most computational studies of RNA structure exclude pseudoknots [31]. That is, secondary structures are not allowed to contain two base (i, j) and (k, l) such that $i < k < j < l$. This condition makes it possible to obtain efficient dynamic programming algorithms. Both the ground state structures [44] and the partition function of the equilibrium ensemble of secondary structures [28] can be computed in cubic time and quadratic space.

The formation of base pairs in a complex of two or more RNA molecules follows the same physical principles as the folding of a single, contiguous RNA chain. The same energy model (with a few simple extensions briefly discussed below) therefore applies to RNA-RNA interactions. However, complexes of multiple RNA strands fall into a class of structures that includes pseudoknot-like structures and thus is difficult to handle computationally. The pairwise case is captured well by the RIP model of [1]. Assuming that so-called tangle-structures do not occur, the RIP model is still amenable to dynamic programming solutions, although at the cost of $\mathcal{O}(n^6)$ time and $\mathcal{O}(n^4)$ space, for both ground-state structures and equilibrium base-pairing probabilities [9,20]. An extension to the multi-strand case was introduced in [29].

The full RIP model is computationally too demanding for most applications, hence approximations and simplifications are usually employed. Examples include a greedy, helix-based approach that allows essentially unrestricted matchings [6] and formalization as a constrained maximum weight clique problem [23]. An alternative is to assume a single, dominating interacting region, which is often – but not always – a plausible approximation, in particular if one of the partners is small as in the case miRNAs. In this scenario the energy of the interaction can decomposed into unfolding energies for the interaction sites on each partner and the hybridization energy of the exposed interaction regions [4,7,30]. A similar approach can be taken when interactions need to conform to specific patterns, is in the case of H/ACA snoRNAs binding to their targets [37].

In this contribution we consider a simplified model that excludes all pseudoknot-like structures. Conceputally, this amounts to computing a conventional, pseudoknot-free secondary on the concatenation of the interacting RNA strands, although with a suitably modified energy model (see below). Although some important types of interactions, in particular kissing-hairpins [13], cannot be modeled in this way, it is still a useful approximation in many situations. For $N = 2$ strands, this model has been analyzed in detail in [2,5,10]. For $N > 2$, the ground-state folding problem still remains essentially unchanged. The only necessary adaptation is a modification of the energy model to assign different energy contributions to substructures ("loops") that contain one or more *nicks*, as we shall the call the breakpoints between strands. Kinetic simulations of multistrand cofolding have been studied in [32]. For $N = 2$, the order of the strands does not matter. In fact, it is easy to see that every crossing-free set of base pairs on AB translates to a crossing-free set of pairs on the alternative order BA. This is no longer true for $N > 2$, however. We now have to consider the different permutations of the RNA strands. For connected structures, two permutations of the RNA strands either form the same set of crossing-free secondary structures (if one is a cyclic permutations of the other), or their sets of crossing-free secondary structures are disjoint [12]. As a consequence, it is necessary to compute the structures for all permutations (with a fixed first strand to exclude the equivalent cyclic permutations). Since the ensembles of (connected) structures are disjoint, one can perform these computations independently. An implementation for the general case is available in `NUPACK` [43].

The binding energies between strands in heteropolymeric structures are intrinsically concentration dependent because the number of particles changes when polymeric structures are formed [10]. Partition function computations therefore need to handle complexes separately that are composed of different combinations of strands. `RNAcofold` [5] initially ignores this issue and first computes a partition function Z_{AB} that includes both connected structures (in which the strands A and B are linked by at least one base pair) and conformation is which the monomers A and B form independent structures. The correct partition function is then obtained as $Z_{AB} - Z_A Z_B$. This approach seems to become tedious for higher-order interactions. `NUPACK` [12] instead considers only connected structures. It turns out that this leads only to a small modification of McCaskill's algorithm. While this avoids the complications arising from disconnected structures, it leads to more complicated outside recursions for computations of base pairing probabilities even though this step still follows the idea of McCaskill's outside recursions [28].

The key issue is that the computation of the probability of the base pair (k, l) needs to consider the case that (k, l) resides in a loop $\mathcal{L}$ with closing pair (i, j) that harbors *exactly* one nick. If the loop $\mathcal{L}$ were to contain two or more nicks, the structure would be disconnected, and hence excluded. Controlling the number nicks in the loop is conceptually simple. In practice, however, it is not trivial to handle without additional effort because all partition function variables computed in the inside recursions, outlined in Sect. 2, only cover connected substructures, and hence the cases with a nick in the exterior loop need to be handled separately. In Sect. 3 we show how this can be achieved efficiently. Section 5 briefly summarizes details and features of the implementation of `RNAmultifold` in the `ViennaRNA` package. Benchmarking data are provided in Sect. 6. Since RNA complex formation is inherently concentration dependent, Sect. 7 briefly describes how this issue is handled in `RNAmultifold`. Section 8 showcases the interactions between spliceosomal RNAs. Finally in Sect. 9 we address some questions and extensions that have been left open for future research and briefly discuss the limits of the approach taken in this contribution.

2 Inside Recursion

Our goal is to compute the partition function of an ensemble of connected, crossing-free secondary structures of $N \geq 1$ RNA strands with a total length n. We assume that the strands are given in a particular order π. Nucleotide positions are order consecutively from 1 to n is this order of strands. For fixed π, a structure is crossing-free if, given a base pair (i, j), another base pair with $i < k < j$ is allowed only if $i < l < j$. The set of crossing free structures remain the same under circular permutations and are disjoint for any other permutation of the strands [12]. The probability $p_{k,l}$ that (k, l) forms a base pair is therefore a weighted sum of of the base pairing probabilities $p_{k,l}[\pi]$ of all permutations π that fix the first strand. The contribution of each permutation π is proportional to its partition function $Q[\pi]$ [12], i.e., we have

$$p_{k,l} = \sum_{\pi} w(\pi) p_{k,l}[\pi] \quad \text{with} \quad w(\pi) = Q[\pi]/Q\,, \tag{1}$$

where $Q := \sum_{\pi} Q[\pi]$ is the total partition function of the complex. From an algorithmic point of view it therefore suffices to solve the folding problem for a fixed permutation π. We may therefore assume that the strands are indexed consecutively as $s = 1, \ldots, N$.

Complexes that contain the same RNA strand more than once imply symmetries that complicate the problem and need to considered at different levels [12]. Copies of the same RNA sequence are not distinguishable. In the general case, therefore, we have to interpret π not a permutation of the integers 1, 2, ..., N, but as the permutations of the letters in a word (with the first letter fixed), where letters correspond to strands accounting for the composition of the complex. We write $\Pi(\kappa)$ for the set of distinguishable non-cyclic permutations of the strands. For instance, we $\Pi('AAB') = \{AAB\}$ and $\Pi('ABAB') = \{AABB, ABAB\}$.

A related issue arises from secondary structures with r-fold rotational symmetry. Again, these are indistinguishable if they are formed over sequences with the same rotational symmetry. In the dynamic programming algorithm they cannot be separated from the non-symmetric structures. Algorithmically, therefore, they are over-counted by a factor of r, corresponding to an energy contribution of $-RT \ln r$. The same symmetry also reduces the distinguishable conformations by a factor of r, thus incurring an entropic penalty of $+RT \ln r$, exactly compensating the algorithmic overcounting [12]. As a consequence, the issue of rotational symmetry can be ignored in partition function calculations. We note that this not true for energy minimization. Since rotationally symmetric structures are destablized by the small – but not negligible – free energy contribution $RT \ln r$ that cannot be accounted for in the dynamic programming algorithm, the prediction of a symmetric ground state may be incorrect and the correct groundstate is the most stable non-symmetric structure, see [19] for details. Symmetries of the secondary structures also map nicks onto each other, r must be a divisor of N and in particular no symmetries are possible for $N = 1$, where the end of molecule is the only nick.

The standard energy model for RNA secondary structures [39] distinguishes three types of "loops": *Hairpin loops* contain no further interior base pairs. *Interior loops*, which contain stacked base pairs as the special case without intervening unpaired bases, contain exactly one interior base pair. *Multi-branch loops* (multi-loops for short) contain two more consecutive base pairs in their interior. Energy contributions for hairpin and interior loops are tabulated as function of closing (and interior) base pair and the sequence(s) of the unpaired stretches. In contrast, a linear approximation is used for multiloops to keep the number of parameters manageable and to ensure that the dynamic programming recursions can evaluated in cubic time and quadratic space.

McCaskill's original approach [28] to computing partition functions and the generalization to multi-strand problems considers *all* structures. Designed for single, contiguous sequences, of course all these structures are trivially connected. It is thus legitimate to re-interpret the variables appearing in McCaskill's algorithms as partition functions over *connected* structures only. As noted in [12] this implies that for $N > 1$ care has to be taken to enforce connectedness.

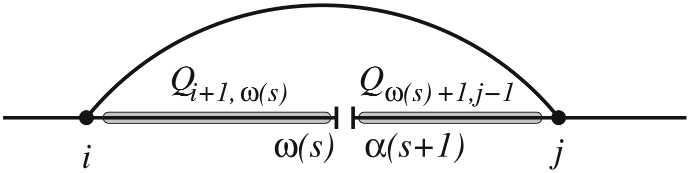

Fig. 1. Nicked loop case in the inside recursion. The RNA sequences are shown as horizontal line, base pairs as arcs. Here, the base pair (i, j) connects two connected components separated by a single nick between $\omega(s)$ and $\alpha(s+1) = \omega(s)+1$. Nicked loops are exterior. The connected secondary structures on the intervals $[i+1, \omega(s)]$ and $[\alpha(s+1), j-1]$ therefore contribute independently. As limiting cases, the nick may be adjacent to i or j, in which case one of the two intervals $[i+1, \omega(s)] = [i+1, i]$ or $[\alpha(s+1), j-1] = [j, j-1]$ is empty. By definition it then contributes as factor of 1 to the partition function. Figure from [26].

The notation in the contribution follows previous presentations of the `ViennaRNA` package [18,25,27]. We write Q_{ij} for the partition function over all crossing-free connected structures on the interval $[i, j]$. The partition function over all crossing-free connected structures on the interval $[i, j]$ that are enclosed by the base-pair (i, j) are denoted by Q^B_{ij}. The additive approximation of multiloop energies implies that the partition function of a multiloop can be decomposed into multiplicative contributions, one for the its closing base pairs (i, j), a term $Q^M_{i+1,u}$ describing the left part of loop containing at least one stem, and a term $Q^1_{u+1,j-1}$ covering the rightmost component containing exactly one stem whose outer-most base pair starts a position $u+1$. For a detailed description we refer to [28].

In order to handle connectedness we first note that if a structure on $[i, j]$ to which the closing pair (i, j) is added is already connected, then the recursions are the same as in McCaskill's original algorithm. The difference for $N > 1$ thus comes from the situations in which (i, j) connects two distinct components. Since the variables Q_{ij}, Q^B_{ij}, $Q^M_{i+1,u}$, and $Q^1_{u+1,j-1}$ all refer to connected structures only, the latter case has to be included as additional alternative in the decomposition of Q^B_{ij} [12]. It pertains to "loops" enclosed by (i, j) in which exactly one nick is "exposed", i.e., not covered by another base pair. From an energetic point of view, such a loop is *external*, i.e., it does not incur the usual destabilizing entropic contributions. The situation is outlined in Fig. 1.

We will need a bit of notation. Denote by $\omega(s)$ the 3'-most nucleotide position of strand s. The contribution of "nicked loops" is then given by

$$Q^N_{ij} = \sum_{s: i \le \omega(s) \le j} \mathrm{e}^{-\varepsilon_{ij}/RT} Q_{i+1,\omega(s)} Q_{\omega(s)+1,j-1} \tag{2}$$

with the additional constraint that either both i and $i+1$ as well as $j-1$ and j must be on the same strand, or the nick is adjacent to the base (i, j), in which case either $i = \omega(s)$ and $j-1$ and j are on a common strand, or $j-1 = \omega(s)$ and

i and $i+1$ are on a common strand. The energy contribution ε_{ij} of the nicked loop comprises only the dangling end terms, see [39] for details.

3 Outside Recursion

In order to compute the base pairing probability $p_{k,l}[\pi]$ we need to evaluate the ensemble of secondary structures that contain the base pair (k,l). All such structures are combinations of a secondary structure on $[k,l]$ and a partial secondary structure outside on $[1,k] \uplus [l,n]$. The non-crossing condition ensures that the inside and outside structures can be combined freely, with additive energies and thus multiplicative partition functions [28]. In fact, the "outside ensembles" can be constructed as complements of "inside ensembles" in a systematic manner [35]. A secondary structure containing (k,l) is connected if and only if both the substructures inside and outside of (k,l) are connected, where connectedness of the outside partial structure means that it is connected once the pair (k,l) is added. Denote by $\widehat{Q}_{k,l}[\pi]$ the partition function over all connected partial secondary structures outside of the base pair (k,l). The partition function over all connected structures that contain the pair (k,l) is then simply $\widehat{Q}_{k,l}[\pi]Q^B_{k,l}[\pi]$ and we obtain the base pairing probabilities for a given permutation of the strands as

$$p_{k,l}[\pi] = \widehat{Q}_{k,l}[\pi]Q^B_{k,l}[\pi]/Q[\pi] \tag{3}$$

where $Q[\pi] = Q_{1,n}[\pi]$ is the partition function over all connected secondary structures. The base pairing probabilities for a N-ary complex of interaction RNAs [12] therefore can be computed as

$$p_{k,l} = \sum_{\pi} w(\pi) p_{k,l}[\pi] = \frac{1}{Q} \sum_{\pi} \widehat{Q}_{k,l}[\pi]Q^B_{k,l}[\pi] \,. \tag{4}$$

The decomposition in Eq. (4) shows that we can compute the $p_{k,l}[\pi]$ independently for each permutation π. We therefore drop the reference to π in the following.

The ensemble of outside structures described by $\widehat{Q}_{k,l}$ consists of three mutually exclusive subsets of structures [28]: (1) structures in which (k,l) is not enclosed by any other base pair with partition function $\bar{Q}_{k,l}$ and (2) structures in which (k,l) is enclosed by another base pairs (i,j). The latter can be subdivided further depending on whether the loop enclosed by (i,j) contains (2a) no nick or (2b) exactly one nick. The corresponding partition functions are denoted by $\breve{Q}_{k,l}$ and $\ddot{Q}_{k,l}$, respectively. Recall that two or more nicks in a loop imply that the secondary structure is not connected. The recursions for $\bar{Q}_{k,l}$ and $\breve{Q}_{k,l}$ are identical to the ones developed in [28]. Since these recursions have been discussed repeatedly in the literature, we do not repeat the details here. It is worth noting, however, that a naïve implementation of the recursions for $\bar{Q}_{k,l}$ and $\breve{Q}_{k,l}$ requires $\mathcal{O}(n^4)$ time. It is not difficult, however, to reduce the time complexity to cubic with the help of auxiliary arrays of size $\mathcal{O}(n)$ [25,28].

The focus of this contribution is the additional multi-strand case, i.e., the partition function $\ddot{Q}_{k,l}$. In order to avoid boundary cases we allow also terms of $Q_{i,i-1} = 1$ denoting denoting empty intervals [25]. Note, however $Q^B_{i,j} = 0$ unless $i < j$, and the terms also vanish if $|j - i| < 3$ unless there is a nick between i and j since a hairpin loop contains a minimum of three unpaired bases. Thus the minimum span of a base pair within a single RNA strand is $|j - i| = 4$. There is no distance constraint across nicks, however. We write $\alpha(s)$ and $\omega(s)$ to denote its 5'-most and 3'-most nucleotide position for strands s. Recall that strands are numbered consecutively w.r.t. the given order π. Thus $\alpha(s+1) = \omega(s) + 1$. Furthermore, we write $\sigma(i) = s$ if and only if $\alpha(s) \leq i \leq \omega(s)$, i.e., if position i occurs in strand s. Finally, we will need the *same-strand indicator function* defined by $\xi_i = 1$ if $\sigma(i) = \sigma(i + 1)$ and $\xi_i = 0$ otherwise, as well as its complement $\bar{\xi}_i := 1 - \xi_i$.

To compute $\ddot{Q}_{k,l}$ we have to consider the relative position of focal base pair (k, l), the enclosing base pair (i, j) and the nick. There are two mutually exclusive cases: (1) the nick is located 3' (right) of (k, l), i.e., between l and j and (2) the nick is located 5' (left) of (k, l), i.e., between i and k. In either case the secondary structure enclosed by $[i, j]$ is divided into two independent parts by the nick, i.e., their partition functions can be computed separately, and we obtain

$$\ddot{Q}_{k,l} = \ddot{Q}^{3'}_{k,l} + \ddot{Q}^{5'}_{k,l} \qquad \text{with} \tag{5}$$

$$\ddot{Q}^{3'}_{k,l} = \sum_{\substack{1 \leq i < k \\ l < j \leq n}} \widehat{Q}_{i,j} Q_{i+1,k-1} \sum_{s | l < \alpha(s) \leq j} Q_{l+1,\alpha(s)-1} Q_{\alpha(s),j-1} \tag{6}$$

$$\ddot{Q}^{5'}_{k,l} = \sum_{\substack{1 \leq i < k \\ l < j \leq n}} \widehat{Q}_{i,j} Q_{l+1,j-1} \sum_{s | i \leq \omega(s) < k} Q_{i+1,\omega(s)} Q_{\omega(s)+1,k-1} \tag{7}$$

The evaluation of a single entry $\ddot{Q}_{k,l}$ according to Eqs. (6) or (7) requires $\mathcal{O}(n^2 N)$ operations for N strands with a total length n. The overall running time of $\mathcal{O}(n^4 N)$ by far exceeds the cubic time complexity of all other parts of the partition function algorithm. The additional factor nN is a serious practical burden. In the following section we show that time complexity can be reduced by rearranging these recursions in such a way that the recomputation of certain intermediate results can be avoided.

4 Computing $\ddot{Q}_{k,l}$ in Cubic Time

The key observation is that fixing the position l and computing the values of $\ddot{Q}_{k,l}$ consecutively for all k, we can pre-compute and store contributions that depend only on l and are required for all k. Again we have to consider nicks to the left and to right of (k, l) separately. Fixing the second index k in $\ddot{Q}^{5'}_{k,l}$, Eq. (7), only affects the number of choices for i and s. Moreover, for each strand s the choices of i are also fixed because i of the fixed upper bound $i \leq \omega(s)$. This suggests to pre-compute parts of the outside contribution for every s with

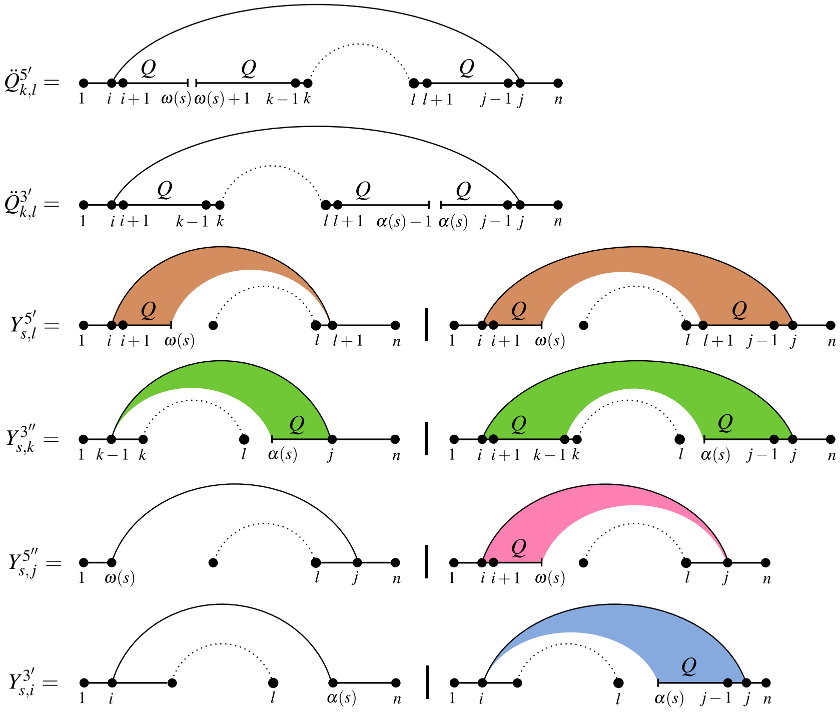

Fig. 2. Auxiliary arrays for computing base pair probabilities for the nicked-loop case. On top the two arrays $\ddot{Q}^{5'}_{k,l}$ and $\ddot{Q}^{3'}_{k,l}$ are sketched, showing the focal base pair (k, l), the enclosing pair (i, j), the position of the nick, and the partition function terms contributing to the loop. The two auxiliary arrays $Y^{5'}_{s,l}$ and $Y^{3''}_{s,k}$ (3rd and 4th line) collect contributions that are independent of the choice of i and j, thus reducing the effort to a sum over the strands s. Parts of these contributions are still re-computed repeatedly when iterating over the l and k. $Y^{5''}_{s,j}$ and $Y^{3'}_{s,i}$ (5th and 6th line) store these parts for reuse. Figure adapted from [26].

$\omega(s) < l$ and all possible choices of i and j. More precisely, we define, for each l, the auxiliary array

$$Y^{5'}_{s} = \xi_l \sum_{j>l} \xi_{j-1} Q_{l+1,j-1} \left(\widehat{Q}_{\omega(s),j} + \sum_{i<\omega(s)} \xi_i \cdot \widehat{Q}_{i,j} \cdot Q_{i+1,\omega(s)} \right). \tag{8}$$

A graphical representation of the contributions captured by $Y^{5'}_{s}$ is provided in Fig. 2. The auxiliary array (which can be overwritten as the outer loop progresses from value of l to the next, has size $\mathcal{O}(N)$ and each entry is computed in $\mathcal{O}(n^2)$ according to Eq. (8), resulting in a total effort of $\mathcal{O}(n^2 N)$. Equation (7) can now be rewritten as

$$\ddot{Q}^{5'}_{k,l} = \bar{\xi}_{k-1} Y^{5'}_{\sigma(k-1)} + \xi_{k-1} \sum_{s|\omega(s)<k} Q_{\omega(s)+1,k-1} Y^{5'}_{s} \,. \tag{9}$$

Each of the $\mathcal{O}(n^2)$ entries now requires $\mathcal{O}(nN)$ operations. Although we have achieved a reduction of the effort by a factor of n, the effort still exceeds the out goal of cubic time complexity.

A further improvement can be obtained by observing that parts of the sums required to compute $Y^{5'}_s$ for a given l can be re-used when $Y^{5'}_s$ is computed for $l-1$ because consecutive entries differ only by a single extra value of j. To make use of this observation we need to replace $Y^{5'}_s$ by $Y^{5'}_{s,l}$, i.e., an array of size $\mathcal{O}(nN)$ that retains the $Y^{5'}_s$ as l changes, together with an additional auxiliary array of the same size:

$$Y^{5'}_{s,l} = \xi_l \left(Y^{5''}_{s,l+1} + \sum_{j>l+1} Q_{l+1,j-1} \cdot Y^{5''}_{s,j} \right) \tag{10}$$

$$Y^{5''}_{s,j} = \xi_{j-1} \left(\widehat{Q}_{\omega(s),j} + \sum_{i<\omega(s)} \xi_i \widehat{Q}_{i,j} \cdot Q_{i+1,\omega(s)} \right) . \tag{11}$$

Since $Y^{5''}_{s,j}$ is independent of l and k we can now re-use the stored contributions for every pair (k,l). Proper care has to be taken to properly interleave the computations of $Y^{5''}_{s,j}$ with the part of the computation that loops over variable l because $\widehat{Q}_{i,j}$ only become available for $l<j$. This does not affect the effort required to pre-fill the array $Y^{5''}_{s,j}$, which is still $\mathcal{O}(n^2N)$. Hence, the time complexity for the evaluation of one entry of $\ddot{Q}^{5'}_{k,l}$ reduces to $\mathcal{O}(n)$. The overall time complexity to compute (9) thus becomes $\mathcal{O}(n^2N)$ time and $\mathcal{O}(nN)$ space.

Let us now turn the second case, a nick located $3'$ of base pair (k,l). Conceptually, we can use the same re-arrangement and pre-computation as for 5' nicks; the details differ, however. We start by observing that fixing the value of the index k affects the possible choices of i only. The contributions to the left of the nick, however, do not contain a re-usable factor independent of k because the (i) recursion the involves the full contribution of $Q_{i+1,k-1}$ and (ii) the strand-changes we need to consider only depend on the current value of l. Instead, there are contributions on the right hand side that can be pre-computed. Define the auxiliary array

$$Y^{3'}_{s,i} = \xi_i \left(\widehat{Q}_{i,\alpha(s)} + \sum_{j>\alpha(s)} \xi_{j-1} \widehat{Q}_{i,j} Q_{\alpha(s),j-1} \right) \tag{12}$$

of size $\mathcal{O}(nN)$. We observe that $Y^{3'}_{s,i}$ is independent of both k and l and thus they can be pre-computed and then re-used for any pair (k,l). Substituting Eq. (12) into Eq. (6) yields

$$\ddot{Q}^{3'}_{k,l} = \xi_{k-1} \sum_{i<k} \xi_i Q_{i+1,k-1} \left(\bar{\xi}_l Y^{3'}_{\sigma(l+1),i} + \xi_l \sum_{s|\alpha(s)>l} Q_{l+1,\alpha(s)-1} Y^{3'}_{s,i} \right) . \tag{13}$$

The $\mathcal{O}(n^2)$ values of $\ddot{Q}^{3'}_{k,l}$ therefore can be evaluated in total time $\mathcal{O}(n^3 N)$ the expense of storing the nN auxiliary values $Y^{3'}_{s,i}$. This does not meet our goal of cubic time complexity, however. A further reduction can be achieved by observing that the order of summation in Eq. (13) can be changed to make the inner sum independent of l. This suggests to introduce the auxiliary array

$$Y^{3''}_{s,k} = \xi_{k-1} \sum_{i<k} \xi_i Q_{i+1,k-1} Y^{3'}_{s,i} \tag{14}$$

of size $\mathcal{O}(nN)$. Figure 2 gives a graphical representation of the class of structures contributing to $Y^{3''}_{s,k}$. The array can be computed from all positions k all strands s in $\mathcal{O}(n^2 N)$ time. Substituting the auxiliary terms Eq. (14) into Eq. (6) yields a recursion similar to Eq. (9):

$$\ddot{Q}^{3'}_{k,l} = \bar{\xi}_l Y^{3''}_{\sigma(l+1),k} + \xi_l \sum_{s|\alpha(s)>l+1} Q_{l+1,\alpha(s)-1} Y^{3''}_{s,k}. \tag{15}$$

Assuming that the $\mathcal{O}(nN)$ values of $Y^{3''}_{s,k}$ are stored, it can be evaluated in $\mathcal{O}(n^2 N)$ total time. As for the 5' nicks, proper interleaving into the recursion is necessary because $Y^{3'}_{s,i}$ depends on $\widehat{Q}_{i,j}$. To this end, we fill $Y^{3'}_{\sigma(l+1),i}$ for all i if $\xi_l = 1$ and subsequently re-compute $Y^{3''}_{s,k}$.

In order to achieve cubic running time we have introduced four auxiliary arrays of size $\mathcal{O}(nN)$, $Y^{5'}_{s,j}$, $Y^{5''}_{s,j}$, $Y^{3'}_{s,i}$, and $Y^{3''}_{s,i}$, each of which can be filled in total time $\mathcal{O}(n^2 N)$. The matrix $\ddot{Q}^{3'}_{k,l}$ of course does not need to be stored. Instead, the value of $\ddot{Q}^{3'}_{k,l}$ can immediately be added to the other contributions of $\widehat{Q}_{k,l}$ and only the latter, or base pairing probabilities $p_{k,l}$, need to be committed to memory. The extra effort for the outside recursion thus matches the extra effort for the inside recursion of the multi-strand folding problem. The number of strands will be much smaller than the total sequence length, $N \ll n$, in any reasonable application scenario. The additional space and time resources required for the multi-strand version of McCaskill's partition function algorithms therefore are asymptotically negligible compared to the single-strand case.

5 Implementation

`RNAmultifold` is part of the `ViennaRNA` package [18,25], release `2.5.0a2`. It provides access to both minimum energy and partition function calculations for arbitrary numbers of strands N. The user can choose to either evaluate a single permutation of the given strands, all permutations corresponding to a given connected complex, or all connected complexes with up to N constituents. Figure 3 shows the base pairing probabilities of a toy example.

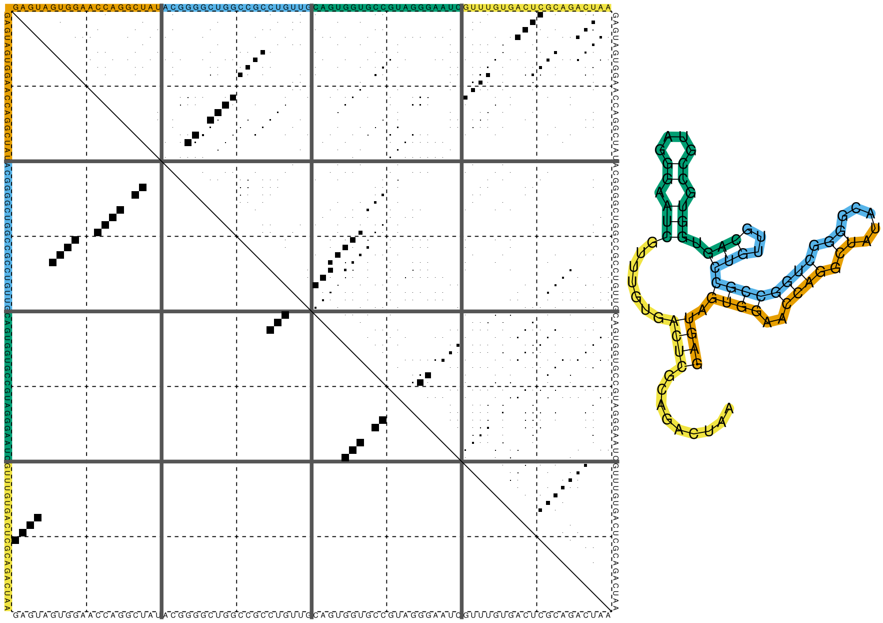

Fig. 3. Toy example with $N = 4$ strands (orange, cyan green and yellow) each of length 20, i.e., total length $n = 80$. The dot plot representation (left) shows the base pairing matrix for a fixed permutation π of the four strands in its upper right half. The lower left half shows the minimum free energy (MFE) structure for the same permutation. The area of each "dot" is proportional to $p_{ij}[\pi]$. Thick lines separate the four strand. The corresponding MFE structure is shown to the right.

A well-known practical issue for the implementations of partition function algorithms are overflow and underflow errors arising from the fact that partition functions consist of exponential terms that quickly grow beyond the range of floating point number as the system size n increases. The `ViennaRNA` package addresses this problem by working with rescaled terms of the form $q_{ij} := Q_{ij}/\zeta^{j-i+1}$. The scaling constant ζ is an estimate for the position-wise multiplicatice contribution to Q, i.e., $\sqrt[n]{Q} = \exp(-g/RT)$, where $g = G/n$ is an estimate for the free energy of folding per nucleotide position [18]. This approach is sufficient to keep q_{ij} and the corresponding restricted partition functions sufficiently close to 1 to avoid overflows for sequence length at least up to 10^4, which appears sufficient for practical applications. A very good estimate is to use the scaled ground state energy $g = E^*/n$. The value of E^* can be computed without numerical problem since the minimum energy computation is implemented using integer arithmetic [18].

The ViennaRNA package provides a flexible framework to handle constraints. It distinguishes *soft constraints*, which are implemented as additional pseudo-energy contributions associated with an unpaired base, a base pairs, or an loop, and *hard constraints* corresponding to forbidden or enforced base pairs [27]. A useful observation in this context is that hard constraints that enforce base pairs between strands can lead to forbidden permutations for $N > 2$: the observation that connected structures are crossing-free in only a single non-cyclic permutation also pertains to constraints. Three or more strands that are connected by hard constraints thus have feasible non-crossing structure only in a single permutation. All other permutations are excluded already during the preprocessing of the hard constraints. We note in passing that RNAmultifold also handles intra-strand G-quadruplexes in the same way as in a single RNA molecule [24]. Both RNA and DNA parameters can be used as in other components of the ViennaRNA package.

6 Benchmarking

We designed a benchmark data set aiming to minimize sequence-specific variations between instances with different numbers of strands. To this end we generated 10 random sequences for each length n and subdivided these into a different number N of separate strands. From the theoretical considerations in the Sect. 4 we expect that both memory consumption and running time should becomes independent of N for large values of the total sequence length n. Empirically, we found that the number of strands has a significant influence only for very short sequences with an average length of individual strands smaller than about 20 nt.

Figure 4 shows that RNAmultifold consistently outperforms NUPACK 3.2.2 [43]. For large sequences, the inside recursion of RNAmultifold is about 35× and the outside recursion is about 50–65× faster. The memory requirements of RNAmultifold are about 7× lower.

Both RNAfold and RNAcofold are contained in the ViennaRNA package and use identical energy parameters. The results of RNAmultifold and RNAfold ($N = 1$) as well as RNAcofold ($N = 2$) coincide within the expected numerical inaccuracies. These programs do not show significant differences in memory consumption. RNAfold is 10–15% faster than RNAmultifold. The outside recursion of RNAmultifold, however, is about two times faster than the corresponding part of RNAcofold. We do not show these small differences separately in Fig. 4.

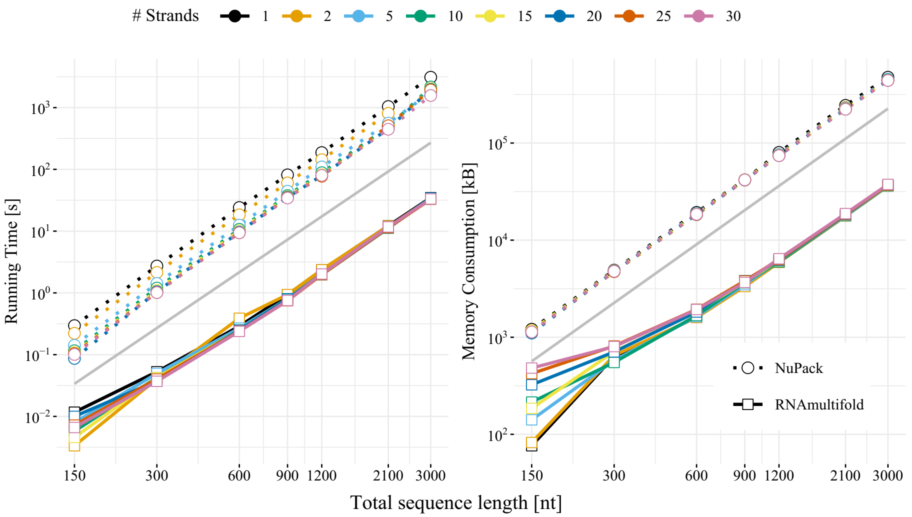

Fig. 4. Comparison of the performance measures for `NUPACK` (version 3.2.2) and `RNAmultifold` for different values of the total sequence length n and number N of strands. For each data point, 10 random instances were averaged. The theoretical asymptotic complexities $\mathcal{O}(n^3)$ for running time and $\mathcal{O}(n^2)$ for memory consumption are shown as thin gray lines. Figure adapted from [26].

7 Concentration Dependence

The formation of an RNA duplex is associated with an additional entropic contribution ε_0 for the initiation of helix formation. In the standard energy model, this term is already subsumed in the loop energies [33,39] and therefore does not appear for $N = 1$. In the case of RNA-RNA interactions, however, an initiation term must be associated with each nicked loop. Since a connected structure with N strands always has exactly $N - 1$ nicks, all connected structures in a complex with given composition are receive a contribution of $(N - 1)\varepsilon_0$, which cancels in Eq. (1) and thus can be ignored in the context of a fixed interaction complex. They do, however, play a role when complexes with a different number of constituents are compared. The partition function of the ensemble of connected structures of a complex κ composed of N (not necessarily distinct) RNA strands including the initiation correction is

$$Z_\kappa = \mathrm{e}^{-(N-1)\varepsilon_0/RT} \sum_{\pi \in \Pi(\kappa)} Q[\pi] \tag{16}$$

The stability of RNA-RNA complex is inherently concentration dependent. The easiest way to see this is to note that the association (and its reverse, the dissociation) of a complex

$$A_1 A_2 ... A_k + B_1 B_2 ... B_l \leftrightharpoons A_1 A_2 ... A_k B_1 B_2 ... B_l \tag{17}$$

changes the number of particles. The equilibrium constant for this reversible reaction is $K = Z_{A_1A_2\ldots A_kB_1B_2\ldots B_l}/Z_{A_1A_2\ldots A_k}Z_{B_1B_2\ldots B_l}$, see e.g. [5,10,12]. According to the law of mass action we can express the equilibrium constant for formation of κ from its constituent strands A_1, A_2,..., A_N as

$$K_\kappa = \frac{Z_\kappa}{Z_{A_1}Z_{A_2}\ldots Z_{A_N}} = \frac{[\kappa]}{[A_1]\,[A_2]\cdots[A_N]}, \tag{18}$$

where [...], as usual in the chemical literature, denotes the concentration of a complex or individual strand.

We introduce the membership matrix $\mathbf{A}$ whose entries $\mathbf{A}_{\alpha,\kappa}$ count the number of strands of type α in complex κ. Assume that our systems contains the total concentration c_α of strand α. The concentration $[\alpha]$ of a strand α that is not contained in a complex is thus

$$[\alpha] = c_\alpha - \sum_\kappa \mathbf{A}_{\alpha,\kappa}[\kappa] \tag{19}$$

Since the system (17) of reversible reactions in particular can be endowed with mass action kinetics, there is a unique equilibrium point [34]. Alternatively, this can be proved starting from the partition function of the grand-canonical ensemble [12]. In the same contribution it is shown that the equilibrium concentrations can be computed by maximizing a function h [12, equ. (3.7)], which in our notation reads

$$h(\vec{\lambda}) = \sum_\alpha(\lambda_\alpha c_\alpha - Z_\alpha \mathrm{e}^{\lambda_\alpha}) - \sum_\kappa Z_\kappa \exp\left(\sum_{\alpha'} \lambda_{\alpha'}\mathbf{A}_{\alpha',\kappa}\right) \tag{20}$$

Since the partition function for large molecules are in an "inconvenient" numerical range, we use the transformation $L_\alpha := \lambda_\alpha + \ln Z_\alpha$ to express the objective function in terms of the equilibrium constants and maximize:

$$h(\vec{L}) = \sum_\alpha(c_\alpha L_\alpha - \mathrm{e}^{L_\alpha}) - \sum_\kappa K_\kappa \exp\left(\sum_{\alpha'} L_{\alpha'}\mathbf{A}_{\alpha',\kappa}\right), \tag{21}$$

where we have omitted the constant term $-\sum_\alpha c_\alpha \ln Z_\alpha$ since it does not affect the maximum. The equilibrium concentrations can then be obtained from [12, equ. (3.12)], which we can rewrite as

$$[\alpha] = \mathrm{e}^{L_\alpha} \qquad\qquad [\kappa] = K_\kappa \prod_\alpha [\alpha]^{\mathbf{A}_{\alpha,\kappa}} \tag{22}$$

Note that the second equation recovers the law of mass action, Eq. (18). It is not difficult to obtain explicit expressions for the gradient and the Hessian of h (see Appendix). As suggested in [12], we use the Trust Region Method implemented as `find_min_trust_region()` in `dlib` [22]. Our implementation of $h(\vec{L})$ and its partial derivatives makes extensive use of the "log-sum-exp trick" to avoid overflow and underflow problems.

Writing $c = \sum_\alpha c_\alpha$ for the total concentration of RNA strands we can also compute the concentration-dependent probability of observing a base-pair between position i in strand α and position j in strand β by summing the $[\kappa]p_{ij}/c$ over all complexes κ (and strand α in case $\alpha = \beta$). If α and β appear more than once in a given complex, the base pairing probabilities need to be averaged over different combinations of interacting copies of α and β within each given complex.

8 Spliceosomale RNAs: A Showcase Applications

The spliceosome is highly dynamic, complex machinery comprising a multitude of proteins as well as the five spliceosomal snRNAs (U1, U2, U4, U5, U6). During the splicing reaction, its composition and internal structure, which also involves direct base pairing interactions between the snRNAs, is drastically rearranged [41]. Neglecting the mRNA target, the effect of RNA protein binding, and any chemical modifications of the snRNAs, we predict the formation of (parts of) the precatalytic spliceosome complex B, in particular its predecessor, the U4/U6.U5 tri-snRNP. To that end, we consecutively increased the concentrations of the individual snRNAs from an initial $0.05\mu M$ to $10\mu M$ in the order U6, U4, U5, and U2. Figure 5 shows the equilibrium concentrations of the snRNA complexes. We observe the formation of U4/U6 as soon as their constituents are available in sufficient concentrations. Upon adding U5, the U4/U6 complex becomes less favorable, instead the triplex U4/U6.U5 dominates the ensemble. Increasing the concentration of U2 afterwards, however, does not seem to affect the equilibrium concentration of U4/U6.U5 nor do we observe any appreciable increase in the concentration of the U4/U6.U5 + U2 tetraplex. Instead, U2 tends to form homo-tetramers. This discrepancy of the prediction with respect to the accepted model of splieceosomal complex formation might be attributed to our simplified model that omits the effect of chemical modifications of the snRNAs, and the impact of protein binding.

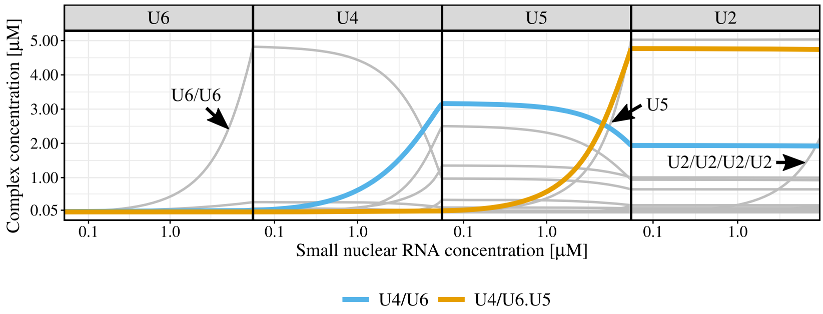

Fig. 5. Concentration dependence of the complexes formed by the human U6, U4, U5, and U2 spliceosomal snRNAs. The concentration of each snRNAs is increased from $0.05\mu M$ to $10\mu M$ in each sub-panel and then fixed at $10\mu M$ for the rest of the simulation. We observe the formation of the U4/U6 dimer complex and the U4/U6.U5 triplex. Increasing the concentration of U2 does not yield any noticable amounts of a U4/U6.U5 + U2 tetraplex. Instead, U2 tends to form homomultimers, possibly due to the lack of protein binding and chemical modifications of the snRNAs in our simplified model.

The base pairing probabilities $p_{k,l}$ can be used to obtain further derived quantities such as expected number N_{AB} of base pairs connecting any two strands A and B in a complex [12]. Since RNAmultifold provides access to the full framework for handling constraints in the ViennaRNA package [27], we easily can use hard constraints to exclude base pairs between certain strands. This provides a convenient thermodynamic estimate for the importance of a binary interaction in the complex. Denoting by Q the unconstrained partition function writing $Q_{A|B}$ for the partition function with the constraint that no base pairs can be formed between A and B. The contribution of the A-B interaction to the complex stability can then be measures by the partial opening energy

$$\Delta G_{A|B} = RT \ln Q - RT \ln Q_{A|B} \geq 0. \tag{23}$$

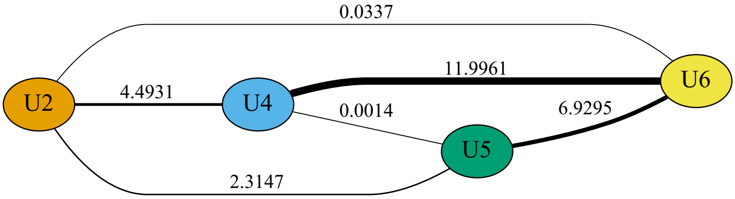

Fig. 6. Importance of binary interactions in the U2 + U4/U6.U5 snRNA complex expressed as $\Delta G_{A|B}$ in kcal/mol. The most stabilizing interactions are U4/U6 followed by U5/U6. Interactions of U2 with any snRNA other than U4 do not play an important role in the overall stability of the full tetraplex.

As an example, we again use the four snRNAs U2, U4, U5, and U6 and compute $\Delta G_{A|B}$ for each pair of interaction in the quaternary complex, see Fig. 6. The largest stabilizing contributions of any complex formed by the four snRNAs can be attributed to U4/U6 and U5/U6 interactions. While still noticable, the interaction between U2 and U4 only contributes a small amount to the overall energy of the complex. In particular, the interactions of U2 with any other snRNA appears energetically negligible.

9 Concluding Remarks and Future Challenges

RNAmultifold extends the ViennaRNA to handling the multi-strand RNA folding problem. For a fixed permutation π of the strands it computes the partition function (inside recursion) and the base pairing probabilities (outside recursion) in $\mathcal{O}(n^3)$ time and $\mathcal{O}(n^2)$. Our implementation has negligible overhead compared to RNAfold and RNAcofold. The performance compares favorably with NUPACK, at present the only competing software, saving nearly an order of magnitude in memory and about a factor of 50 in running time.

The ViennaRNA package provides access to multi-strand folding at three different levels of abstraction. First, computations can be conducted for fixed π as described at length in Sects. 2–6. The interface at the low level is useful in particular when large complexes are considered for which the set of permutations Π is too large to enumerate exhaustively. For smaller problems, functions are available that autonomously handle a complex with a given composition, returning e.g. aggregated base pairing probabilities, Eq. (1). At the top level, a mixture of strands and a list of allowed complexes can be defined to compute concentration-dependent observables.

Nevertheless, some issues remain open for future research. Some functionalities of the ViennaRNA are not yet available for multi-strand folding. Some of these features are straightforward extension of the partition function algorithms, and will become available with the next major release. This concerns in particular stochastic backtracking to sample individual structures with Boltzmann probabilities [11,36] and extensions of the RNA folding grammar necessary to handle multiple ligand binding sites [15] again making use of the constraints framework described in [27]. Since the symmetry effects compensate for partition functions, no symmetry corrections apply in the sampling process. The enumeration of suboptimal structures [42] is an extension of MFE folding algorithm [42]. Here we will have to take special care to properly treat the energy penalties associated with structures with symmetries that appear in particular in homo-dimers and -multimers [19].

A closer inspection of the folding recursions for different permutations π and π' reveals that parts of the arrays that need to be computed the forward recursions are identical. This suggests to avoid the recomputation to reduce the computational efforts. For larger numbers of strands and/or complexed composed of many strands it will be necessary to develop approximations that make it possible to decide without detailed computations which complexes and which permutations of strands within a complex need to be considered and which ones can be neglected.

RNAmultifold handles only pseudo-knot-free structures and thus excludes certain modes of RNA-RNA interactions such as kissing hairpins that are relevant both in biological and technological systems. While a large class of strand-displacement systems are pseudo-knot free, many of the sensor and signal amplification systems reviewed in [40] go beyond this paradigm. A simple extension of the approach taken here to pseudoknotted structures does not seem possible, however. Since there is no analog of the partitioning of connected structures into disjoint classes depending on the permutations of the strands, the entire "concatenation-like" paradigm becomes untenable. A possible alternative might be to used RNAup/intaRNA-like methods [4,7,30] to compute individual, localized interactions between entire complexes and to construct a network of exchange reactions between complexes. Such an approach, however, is very different from considering the full ensemble of all structures.

Availability

`RNAmultifold` can be downloaded as part of `ViennaRNA Package` 2.5.0a2 from www.tbi.univie.ac.at/RNA.

Appendix

Gradient and Hessian of *h*

Efficient optimization of h, Eq. (21), required the gradient and the Hessian of h, which we give here for convenience:

$$\begin{aligned}\frac{\partial h}{\partial L_\alpha} &= c_\alpha - \mathrm{e}^{L_\alpha} - \sum_\kappa \mathbf{A}_{\alpha,\kappa} K_\kappa \exp\left(\sum_{\alpha'} L_{\alpha'} \mathbf{A}_{\alpha',\kappa}\right) \\ \frac{\partial^2 h}{\partial L_\alpha \partial L_\beta} &= -\delta_{\alpha\beta}\mathrm{e}^{L_\alpha} - \sum_\kappa \mathbf{A}_{\alpha,\kappa}\mathbf{A}_{\beta,\kappa} K_\kappa \exp\left(\sum_{\alpha'} L_{\alpha'} \mathbf{A}_{\alpha',\kappa}\right)\end{aligned} \tag{24}$$

We note that the Hessian is negative definite since the sum can be written as $-\mathbf{M}\mathbf{M}^+$ with $\mathbf{M}_{\alpha,\kappa} = \mathbf{A}_{\alpha,\kappa}\sqrt{K_\kappa}\exp\left(\frac{1}{2}\sum_{\alpha'} L_{\alpha'}\mathbf{A}_{\alpha',\kappa}\right)$.

References

1. Alkan, C., Karakoç, E., Nadeau, J.H., Sahinalp, S.C., Zhang, K.Z.: RNA-RNA interaction prediction and antisense RNA target search. J. Comput. Biol. **13**, 267–282 (2006). https://doi.org/10.1089/cmb.2006.13.267
2. Andronescu, M., Zhang, Z.C., Condon, A.: Secondary structure prediction of interacting RNA molecules. J. Mol. Biol. **345**, 987–1001 (2005). https://doi.org/10.1016/j.jmb.2004.10.082
3. Badelt, S., Grun, C., Sarma, K.V., Wolfe, B., Shin, S.W., Winfree, E.: A domain-level DNA strand displacement reaction enumerator allowing arbitrary non-pseudoknotted secondary structures. J. R. Soc. Interface **17**, 20190866 (2020). https://doi.org/10.1098/rsif.2019.0866
4. Bernhart, S.H., Mückstein, U., Hofacker, I.L.: RNA accessibility in cubic time. Algorithms Mol. Biol. **6**, 3 (2011). https://doi.org/10.1186/1748-7188-6-3
5. Bernhart, S.H., Tafer, H., Mückstein, U., Flamm, C., Stadler, P.F., Hofacker, I.L.: Partition function and base pairing probabilities of RNA heterodimers. Algorithms Mol. Biol. **1**, 3 (2006). https://doi.org/10.1186/1748-7188-1-3
6. Bindewald, E., Afonin, K., Jaeger, L., Shapiro, B.A.: Multistrand RNA secondary structure prediction and nanostructure design including pseudoknots. ACS Nano **5**, 9542–9551 (2011). https://doi.org/10.1021/nn202666w
7. Busch, A., Richter, A., Backofen, R.: IntaRNA: efficient prediction of bacterial sRNA targets incorporating target site accessibility and seed regions. Bioinformatics **24**, 2849–2856 (2008). https://doi.org/10.1093/bioinformatics/btn544

8. Chappell, J., Watters, K.E., Takahashi, M.K., Lucks, J.B.: A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future. Curr. Opin. Chem. Biol. **28**, 47–56 (2015). https://doi.org/10.1016/j.cbpa.2015.05.018
9. Chitsaz, H., Salari, R., Sahinalp, S.C., Backofen, R.: A partition function algorithm for interacting nucleic acid strands. Bioinformatics **25**, i365–i373 (2009). https://doi.org/10.1093/bioinformatics/btp212
10. Dimitrov, R.A., Zuker, M.: Prediction of hybridization and melting for double-stranded nucleic acids. Biophys. J. **87**, 215–226 (2004). https://doi.org/10.1529/biophysj.103.020743
11. Ding, Y., Chan, C.Y., Lawrence, C.E.: Sfold web server for statistical folding and rational design of nucleic acids. Nucleic Acids Res. **32**, W135–W141 (2004). https://doi.org/10.1093/nar/gkh449
12. Dirks, R.M., Bois, J.S., Schaeffer, J.M., Winfree, E., Pierce, N.A.: Thermodynamic analysis of interacting nucleic acid strands. SIAM Rev. **49**, 65–88 (2007). https://doi.org/10.1137/060651100
13. Durand, G., et al.: A combinatorial approach to the repertoire of RNA kissing motifs; towards multiplex detection by switching hairpin aptamers. Nucleic Acids Res. **44**, 4450–4459 (2016). https://doi.org/10.1093/nar/gkw206
14. Dutta, T., Srivastava, S.: Small RNA-mediated regulation in bacteria: a growing palette of diverse mechanisms. Gene **656**, 60–72 (2018). https://doi.org/10.1016/j.gene.2018.02.068
15. Forties, R.A., Bundschuh, R.: Modeling the interplay of single stranded binding proteins and nucleic acid secondary structure. Bioinformatics **26**, 61–67 (2010). https://doi.org/10.1093/bioinformatics/btp627
16. Gong, J., Ju, Y., Shao, D., Zhang, Q.C.: Advances and challenges towards the study of RNA-RNA interactions in a transcriptome-wide scale. Quant. Biol. **6**(3), 239–252 (2018). https://doi.org/10.1007/s40484-018-0146-5
17. Guil, S., Esteller, M.: RNA-RNA interactions in gene regulation: the coding and noncoding players. Trends Biochem. Sci. **40**, 248–256 (2015). https://doi.org/10.1016/j.tibs.2015.03.001
18. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M., Schuster, P.: Fast folding and comparison of RNA secondary structures. Monatshefte für Chemie **125**, 167–188 (1994). https://doi.org/10.1007/BF00818163
19. Hofacker, I.L., Reidys, C.M., Stadler, P.F.: Symmetric circular matchings and RNA folding. Discret. Math. **312**, 100–112 (2012). https://doi.org/10.1016/j.disc.2011.06.004
20. Huang, F.W.D., Qin, J., Reidys, C.M., Stadler, P.F.: Partition function and base pairing probabilities for RNA-RNA interaction prediction. Bioinformatics **25**, 2646–2654 (2009). https://doi.org/10.1093/bioinformatics/btp481
21. Isaacs, F.J., Dwyer, D.J., Collins, J.J.: RNA synthetic biology. Nat. Biotechnol. **24**, 545–554 (2006). https://doi.org/10.1038/nbt1208
22. King, D.E.: Dlib-ml: a machine learning toolkit. J. Mach. Learn. Res. **10**, 1755–1758 (2009). /http://dlib.net/
23. Legendre, A., Angel, E., Tahi, F.: RCPred: RNA complex prediction as a constrained maximum weight clique problem. BMC Bioinform. **20**, 128 (2019). https://doi.org/10.1186/s12859-019-2648-1
24. Lorenz, R., et al.: 2D meets 4G: G-quadruplexes in RNA secondary structure prediction. IEEE Trans. Comput. Biol. Bioinf. **10**, 832–844 (2013). https://doi.org/10.1109/TCBB.2013.7
25. Lorenz, R., et al.: ViennaRNA package 2.0. Algorithms Mol. Biol. **6**, 26 (2011). https://doi.org/10.1186/1748-7188-6-26

26. Lorenz, R., Flamm, C., Hofacker, I.L., Stadler, P.F.: Efficient computation of base-pairing probabilities in multi-strand RNA folding. In: de Maria, E., Fred, A., Gamboa, H. (eds.) Proceedings of the 13th International Joint Conference on Biomedical Engineering Systems and Technologies – Volume 3: Bioinformatics. pp. 23–31. Scitepress, Setúbal (2020)
27. Lorenz, R., Hofacker, I.L., Stadler, P.F.: RNA folding with hard and soft constraints. Algorithms Mol. Biol. **11**, 8 (2016). https://doi.org/10.1186/s13015-016-0070-z
28. McCaskill, J.S.: The equilibrium partition function and base pair binding probabilities for RNA secondary structure. Biopolymers **29**, 1105–1119 (1990). https://doi.org/10.1002/bip.360290621
29. Mneimneh, S., Ahmed, S.A.: Multiple RNA interaction: beyond two. IEEE Trans. Nanobiosci. **14**, 210–219 (2015). https://doi.org/10.1109/TNB.2015.2402591
30. Mückstein, U., et al.: Translational control by RNA-RNA interaction: improved computation of RNA-RNA binding thermodynamics. In: Elloumi, M., Küng, J., Linial, M., Murphy, R.F., Schneider, K., Toma, C. (eds.) BIRD 2008. CCIS, vol. 13, pp. 114–127. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-70600-7_9
31. Reidys, C.M.: Combinatorial Computational Biology of RNA. Springer, Heidelberg (2011). https://doi.org/10.1007/978-0-387-76731-4
32. Schaeffer, J.M., Thachuk, C., Winfree, E.: Stochastic simulation of the kinetics of multiple interacting nucleic acid strands. In: Phillips, A., Yin, P. (eds.) DNA 2015. LNCS, vol. 9211, pp. 194–211. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21999-8_13
33. Serra, M.J., Turner, D.H.: Predicting thermodynamic properties of RNA. Methods Enzymol. **259**, 242–261 (1995). https://doi.org/10.1016/0076-6879(95)59047-1
34. Shear, D.B.: Stability and uniqueness of the equilibrium point in chemical reaction systems. J. Chem. Phys. **48**, 4144–4147 (1968). https://doi.org/10.1063/1.1669753
35. Höner zu Siederdissen, C., Prohaska, S.J., Stadler, P.F.: Algebraic dynamic programming over general data structures. BMC Bioinform. **16**(19), S2 (2015). https://doi.org/10.1186/1471-2105-16-S19-S2
36. Tacker, M., Stadler, P.F., Bornberg-Bauer, E.G., Hofacker, I.L., Schuster, P.: Algorithm independent properties of RNA structure prediction. Eur. Biophys. J. **25**, 115–130 (1996). https://doi.org/10.1007/s002490050023
37. Tafer, H., Kehr, S., Hertel, J., Stadler, P.F.: RNAsnoop: efficient target prediction for box H/ACA snoRNAs. Bioinformatics **26**, 610–616 (2010). https://doi.org/10.1093/bioinformatics/btp680
38. Backofen, R., et al.: RNAs everywhere: genome-wide annotation of structured RNAs. J. Exp. Zool. B: Mol. Dev. Evol. **308**(B), 1–25 (2007). https://doi.org/10.1002/jez.b.21130. The Athanasius F. Bompfünewerer RNA Consortium
39. Turner, D.H., Mathews, D.H.: NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. Nucleic Acids Res. **38**, D280–D282 (2010). https://doi.org/10.1093/nar/gkp892
40. Wang, F., Lu, C.H., Willner, I.: From cascaded catalytic nucleic acids to enzyme-DNA nanostructures: controlling reactivity, sensing, logic operations, and assembly of complex structures. Chem. Rev. **114**, 2881–2941 (2014). https://doi.org/10.1021/cr400354z
41. Will, C.L., Lührmann, R.: Spliceosome structure and function. Cold Spring Harb. Perspect. Biol. **3**, a003707 (2011). https://doi.org/10.1101/cshperspect.a003707

42. Wuchty, S., Fontana, W., Hofacker, I.L., Schuster, P.: Complete suboptimal folding of RNA and the stability of secondary structures. Biopolymers **49**, 145–165 (1999). https://doi.org/10.1002/(SICI)1097-0282(199902)49:2⟨145::AID-BIP4⟩3.0.CO;2-G
43. Zadeh, J.N., et al.: NUPACK: analysis and design of nucleic acid systems. J. Comput. Chem. **32**, 170–173 (2011). https://doi.org/10.1002/jcc.21596
44. Zuker, M., Stiegler, P.: Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. Nucleic Acids Res. **9**, 133–148 (1981). https://doi.org/10.1093/nar/9.1.133