BIOINFORMATICS

Vol. 27 ISMB 2011, pages i129–i136
doi:10.1093/bioinformatics/btr220

A folding algorithm for extended RNA secondary structures

Christian Höner zu Siederdissen[1,*], Stephan H. Bernhart[1], Peter F. Stadler[1,2,3,4,5,6] and Ivo L. Hofacker[1,5]

[1]Institute for Theoretical Chemistry, University of Vienna, A-1090 Vienna, Austria, [2]Bioinformatics Group, Department of Computer Science, and Interdisciplinary Center for Bioinformatics, University of Leipzig, D-04107 Leipzig, [3]Max Planck Institute for Mathematics in the Sciences, [4]RNomics Group, Fraunhofer IZI, D-04103 Leipzig, Germany, [5]Center for Non-Coding RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, DK-1870 Frederiksberg, Denmark and [6]The Santa Fe Institute, Santa Fe, 87501 NM, USA

ABSTRACT

Motivation: RNA secondary structure contains many non-canonical base pairs of different pair families. Successful prediction of these structural features leads to improved secondary structures with applications in tertiary structure prediction and simultaneous folding and alignment.

Results: We present a theoretical model capturing both RNA pair families and extended secondary structure motifs with shared nucleotides using 2-diagrams. We accompany this model with a number of programs for parameter optimization and structure prediction.

Availability: All sources (optimization routines, RNA folding, RNA evaluation, extended secondary structure visualization) are published under the GPLv3 and available at www.tbi.univie.ac.at/software/rnawolf/.

Contact: choener@tbi.univie.ac.at

1 INTRODUCTION

The classical RNA secondary structure model considers only the Watson–Crick AU and GC base pairs as well as the GU wobble pair. A detailed analysis of RNA 3D structures, however, reveals that there are 12 basic families of interactions between the bases, all of which appear in nature (Leontis and Westhof, 2001; Leontis *et al.*, 2002). Moreover, virtually all known RNA tertiary structures contain the so-called non-Watson–Crick base pairs. This has led to the development of an extended presentation of RNA contact structures with edges labeled by their pairing type (an example can be seen in Fig. 1). This extended description of base pairing is commonly termed after its inventors the Leontis–Westhof (LW) representation.

The LW representation has proved to be a particularly useful means of analyzing 3D structures of RNA as determined by X-ray crystallography and NMR spectroscopy (Leontis and Lescoute, 2006). In particular, it has led to the discovery of recurrent structural motifs, such as kink-turns and C-loops, that act as distinctive building blocks of 3D structures. The sequence variation in these structural motifs follows combinatorial rules that can be understood by the necessity to maintain the overall geometry when base pairs are exchanged. These isostericity rules are discussed in detail by Lescoute *et al.* (2005); Stombaugh *et al.* (2009). As a new level of RNA structure description, the ability to predict non-standard base pairs can be expected to improve the performance of RNA structure prediction. Furthermore, information about evolutionary conservation of the isostericity classes of these non-standard base pairs will improve consensus structure-prediction and structure-dependent RNA gene finding.

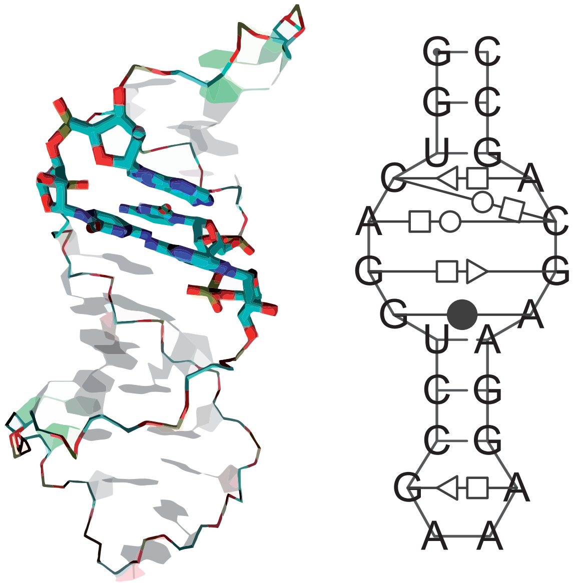

Fig. 1. Example of a structure containing base triplets. The inner part (bases 14–37) of the PDB structure 1dul is shown in a 3D representation and as a 2D structure plot displaying the non-standard base pairs in LW representation. The four bases highlighted in the 3D structure form the two base triplets that can be seen in the upper part of the interior loop in the 2D structure.

Since many additional interactions beyond the standard base pairs are represented in the LW formalism, what was considered to be a loop in classical secondary structures can now appear as complex structures of non-standard base pairs. These non-standard base pairs effectively divide the long ‘classical’ loops into much shorter ones. Parisien and Major (2008) proposed a model that contains loops with no more than four unpaired bases. For unbranched structures, the model is scored using a statistical potential estimated from the available 3D structures by counting the relative frequencies of base pairs, short unbranched loops of particular shapes in dependence of their sequences and combinations of loops with a common base pair. An accompanying folding procedure, `MC-Fold` (Parisien and Major, 2008), which exhaustively enumerates stem-loop components, is available and has been used very successfully as a first step toward the *de novo* prediction of RNA 3D structures

[*]To whom correspondence should be addressed.

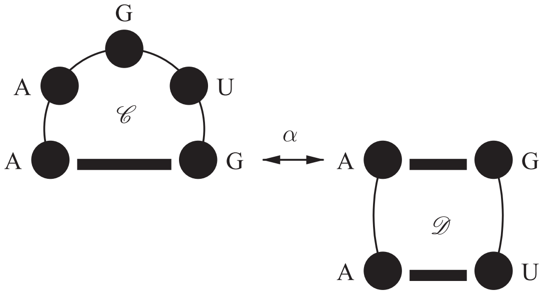

Fig. 2. MC-Fold and MC-Fold-DP both consider small loops, like the hairpin AAGUG ($\mathscr{C}$) and the 2×2 stack AAGU ($\mathscr{D}$) (read clockwise, starting bottom left). Each loop is scored by a function $E_c(\mathscr{C}|\text{AAGUG})$. The stack ($\mathscr{D}$) follows analoguously. The interaction term between two loops is calculated as indicated by the arrow (α), where the two loops are overlayed at the common AG pair. The contribution of the interaction is $E_{\text{junction+hinge}}(\mathscr{C},\mathscr{D};\theta;\text{A},\text{G})$ with θ the unknown pair family.

using MC-Sym (Parisien and Major, 2008), which takes as input the proposed secondary structure from MC-Fold.

2 MC-FOLD REVISITED

2.1 Algorithm

Like ordinary secondary structure prediction tools, MC-Fold (Parisien and Major, 2008) is based on a decomposition of the RNA structure into 'loops'. In contrast to the standard energy model, however, it considers the full set of base pair types available in the LW representation. Each base pair, therefore, corresponds to a triple $(i,j;\theta)$ where θ is one of the 12 types of pairs. In this model, ordinary secondary structures are the subset of pairs with Watson-Crick—Watson-Crick type ($\theta=$'WW') and the two nucleotides form one of the six canonical combinations $\{\mathsf{AU},\mathsf{UA},\mathsf{CG},\mathsf{GC},\mathsf{GU},\mathsf{UG}\}$. This extension of the structure model also calls for a more sophisticated energy model. While the standard model assumes the contributions of the loops to be strictly additive, MC-Fold also considers interactions between adjacent unbranched loops (hairpins, stacked pairs, bulges and general interior loops). This means that the total energy of a structure is not only dependent on the loop types present, but also on the arrangement of these loops. Dispensing with details of the parametrization, the scoring function of MC-Fold for a structure $\mathfrak{S}$ on sequence x can be written as follows (see Fig. 2):

$$\begin{aligned}E(\mathfrak{S}|x)=&\sum_{\mathscr{C}}E_c(\mathscr{C}|x[\mathscr{C}])\\&+\sum_{\substack{\mathscr{C}',\mathscr{C}''\\(k,l)=\mathscr{C}'\cap\mathscr{C}''}}E_{j+h}(\mathscr{C}',\mathscr{C}'';\theta;x[k],x[l])\end{aligned}\tag{2.1}$$

where $\mathscr{C},\mathscr{C}',\mathscr{C}''$ are different loops of $\mathfrak{S}$. The additive term E_c tabulates the (sequence-dependent) contributions of the loops. The interaction term E_{j+h} accounts for the 'junction' and 'hinge' terms in stem–loop regions. These interaction terms depend on the type of the adjacent loops as well as on the type θ and sequence $(x[k],x[l])$ of the base pair that connects them. For multiloops, only the additive term is considered.

Let us ignore multiloops for the moment. A basepair $(i,j;\theta)$ then encloses a loop of type $\mathscr{L}$ which is either a hairpin or encloses a loop $\mathscr{K}$. It is connected to $\mathscr{K}$ by a base pair $(k,l;\psi)$ with $i<k<l<j$. Let $B_{ij}(\theta;\mathscr{L})$ be the minimal energy of a structure on $x[i..j]$ enclosed by a base pair $(i,j;\theta)$ with an outermost loop of type $\mathscr{L}$. Note that, in our notation, the loop type $\mathscr{L}$ also specifies its length and hence implicitly determines the coordinates of the inner base pair of an interior loop: $(k,l)=(i+\ell_1(\mathscr{L}),j-\ell_2(\mathscr{L}))$. For simplicity, we write $(k(\mathscr{L}),l(\mathscr{L}))$. If $\mathscr{L}$ is a hairpin, then $B_{ij}(\theta;\mathscr{L})=\mathscr{H}[i,j;\theta;\text{hairpin}]$, a tabulated energy parameter. Otherwise, we have the recursion

$$B_{ij}(\theta;\mathscr{L})=\min_{\psi,\mathscr{K}}\Big(\mathcal{I}[i,j;\theta;\mathscr{L};\psi,\mathscr{K}]+B_{k(\mathscr{L}),l(\mathscr{L})}(\psi;\mathscr{K})\Big)\tag{2.2}$$

This can be expanded to a full 'next-nearest-neighbor' model by enforcing an explicit dependence on the type of the inner base pair:

$$\begin{aligned}B_{ij}(\theta;\mathscr{L};\psi)=\min_{\psi,\mathscr{K},\phi}\Big(&\mathcal{I}[i,j;\theta;\mathscr{L};\psi;\mathscr{K};\phi]\\&+B_{k(\mathscr{L}),l(\mathscr{L})}(\psi;\mathscr{K};\phi)\Big)\end{aligned}\tag{2.3}$$

The effort to evaluate this recursion equation for a fixed base pair (i,j) is L^3T^3, where L is the number of loop types and T is the number of base pair types. While this prefactor is inconveniently large, we nevertheless obtain an $\mathcal{O}(n^2)$ [or $\mathcal{O}(n^3)$ with multibranched loops] folding algorithm instead of the exponential runtime of MC-Fold.

The problem with this general form of energy parametrization is the unmanageable number of parameters that need to be measured, estimated or learned from a rather limited set of experiments and known RNA structures.

2.2 Parametrization and implementation

Since the folding problem for the MC-Fold model can be solved in polynomial time, the associated parameter estimation problem becomes amenable to advanced parameter optimization techniques (Andronescu *et al.*, 2007; Do *et al.*, 2008). At present, however, we have opted to extend the original MC-Fold parameters only by simple sparse data corrections that can be applied on top of the original MC-Fold database. This has the advantage of allowing a direct comparison between the original version of MC-Fold and our dynamic programming version MC-Fold-DP. In contrast to the original version, MC-Fold-DP can cope with large data sets and long sequences (3 s for 250 nt, about 24 s for 500 nt with MC-Fold-DP, compared to 660 s for 100 nt with MC-Fold).[1]

In terms of algorithmic design, we have made several changes. The grammar underlying MC-Fold-DP follows the ideas of Wuchty *et al.* (1999). This makes the generation of all suboptimal structures in an energy band above the ground state possible. The decomposition of interior loops into small loops implies that MC-Fold-DP runs in $\mathcal{O}(n^3)$ time without the need for the usual explicit truncation of long interior loops. The recursion that fills stem loops [*Nucleotide Cyclic Motifs* (NCMs) in the nomenclature of Parisien and Major (2008)] is now reduced to a function $\text{NCM}(i,j,\text{type}_{i,j},k,l,\text{type}_{k,l})$. For the matrix, entry $(i,j,\text{type}_{i,j})$ is minimized over all $(k,l,\text{type}_{k,l})$ with (k,l) determined by the newly inserted motif $\text{type}_{i,j}$. Hairpins are even simpler: they follow $\text{NCM}(i,j,\text{type}_{i,j},\ldots)$ but there is no inner part $(k,l,\text{type}_{k,l})$.

[1] Note that the implementation of MC-Fold-DP has *not* been aggressively optimized apart from using the polynomial-time algorithm.

The total number of motif types is small (15 in the original set, of which not all are actually used). Both the time and space complexities are, therefore, small enough to handle RNAs with a length of several hundred nucleotides, i.e. in the range that is typically of interest. In fact, the time complexity is similar to ordinary secondary structure prediction where interior loop size is bounded by a constant. Since the grammar is unambiguous, it is also straightforward to compute partition functions and base pairing probabilities, although this feature is not available in the current implementation.

3 BEYOND 1-DIAGRAMS

3.1 Base triplets

An important restriction of secondary structures is that each nucleotide interacts with at most one partner. In combinatorial terms, secondary structures are 1-diagrams. A closer analysis of the available 3D structures, however, reveals that many nucleotides form specific base pairs with two other nucleotides, forming *'base triplets'* or, more generally, *'multi-pairings'*. Cross-free b-diagrams with maximal number b of interaction partners for each nucleotide can be treated combinatorically in complete analogy with (pseudoknot-free) secondary structures by conceptually splitting each node into as many vertices as there are incident base pairs (arcs). As in the case of secondary structures, we say that (i,j) and (k,l) cross if $i<k<j<l$ or $k<i<l<j$. A b-diagram is non-crossing if no two arcs cross. Base pairs can then be well-ordered also in this extended setting: two distinct arcs $(i,j)\neq(k,l)$ are either *nested* $(i\leq k<l\leq j)$ or *juxtaposed* $(i<j\leq k<l)$. This observation is used in `RNAMotifScan` (Zhong *et al.*, 2010) to devise a dynamic programming algorithm for sequence structure alignments along the lines of `RNAscf` (Bafna *et al.*, 2006) or `locarna` (Will *et al.*, 2007), which in turn are restricted variants of the Sankoff algorithm (Sankoff, 1985).

Here, we consider only structures with at most two base pairs involving the same nucleotide, i.e. 2-diagrams. In this case, there is a convenient string representation generalizing the Vienna (dot-parentheses) notation for secondary structures by introducing three additional symbols `<`, `>`, `X` for positions in which two arcs meet: `((` = `<`, `))` = `>` and `)(` = `X`. For general b, the number of necessary symbols grows quadratically, $s_b=(b+1)(b+2)/2$, since each must encode b_1 opening and b_2 closing pairs with $b_1,b_2\geq 0$ and $b_1+b_2\leq b$. These symbols provide a direct representation of the arc nodes '$\triangleleft, \triangleright, \times$' of Figure 3 and are an optional output of the folding program described below to visualize 2-diagrams in the secondary structure.

3.2 A grammar with base triplets

In order to design a dynamic programming folding algorithm for cross-free 2-structures we need a decomposition, i.e. a grammar for 2-structures. For practical applications, it is desirable to have not only a minimization algorithm, but also a partition function version. To this end, an unambiguous grammar is required (Dowell and Eddy, 2004; Reeder *et al.*, 2005). A simple version, treating base pairs as the elementary entities is shown in Figure 3. It translates into an extension of either a Nussinov-style algorithm for maximizing the number of base pairs or a recursion for counting the number of non-crossing 2-diagrams. Let F_{ij} denote the minimum energy of a structure on the sequence interval $x[i..j]$. We have

$$F_{ij}=\min\begin{cases} F_{i+1,j} \\ C_{ij} \\ \epsilon^a_{i,j}+U_{i,j-1} \\ \epsilon^b_{i,j}+V_{i+1,j} \\ \epsilon^c_{i,j}+W_{i,j} \\ \min\limits_{i<k<j}\begin{cases} C_{i,k}+F_{k+1,j} \\ \epsilon^a_{i,k}+U_{i,k-1}+F_{k+1,j} \\ \epsilon^b_{i,k}+V_{i+1,k}+F_{k+1,j} \\ \epsilon^c_{i,k}+W_{i,k}+F_{k+1,j} \\ \epsilon^a_{i,k}+F_{i+1,k-1}+U_{k,j} \\ \epsilon^c_{i,k}+U_{i,k-1}+U_{k,j} \end{cases} \end{cases} \tag{3.1}$$

and analogous recursion for U_{ij},V_{ij} and W_{ij}, denoting the minimum energies over all structures whose left, right or both ends, are involved in a triplet. The symbol C_{ij} refers to structures enclosed by a non-triplet base pair. In the simplest case, $C_{ij}=\epsilon_{ij}+F_{i+1,j-1}$ (lower right corner of Fig. 3). The terminal symbols are the unpaired base •, the ordinary base pairs and the three types of base pairs involved in triplets, contributing $\epsilon_i=0$ sequence-dependent energy increment ϵ_{ij} and sequence-dependent energy increments ϵ^a_{ij}, ϵ^b_{ij} and ϵ^c_{ij}, respectively. The recursion is initialized with $F_{ii}=0$.

Only certain combinations of types of base pairs can occur in triplets. Thus, in a refined model we need to replace U_{ij}, V_{ij} and W_{ij} by $U_{ij}[\nu]$, $U_{ij}[\mu]$ and $W_{ij}[\xi]$ explicitly referring to the base pair type(s) of the triplet. Furthermore, the energy parameters also become type dependent $\epsilon^a_{ij}\rightarrow\epsilon^a_{ij}[\rho]$ or even $\epsilon^a_{ij}[\rho,\nu]$ where ρ is the type of the pair itself and ν is type of the second pair of the triplet. The first variant is chosen for Nussinov-like algorithms, where each individual base pair is evaluated, splitting triplets, and the second variant is more fitting for Turner-like nearest neighbor models. In that case, recursion on W changes to $W_{ij}[\nu,\mu]$ to reflect the pairing choice being made.

3.3 Full loop-based model

The grammar of Figure 3 can be extended to incorporate the standard loop-based Turner energy model (Turner and Mathews, 2010) (which distinguishes hairpin loops, stacks of two base pairs, bulges, interior loops and multibranched loops). The modification of the grammar is tedious but rather straightforward, as seen in Figure 4. Instead of treating the base pairs themselves as terminal symbols (as in Fig. 3), this role is taken over by entire loops. Note that as in the case of ordinary secondary structures, each loop in a given structure is uniquely determined by its closing pair. The energy contributions now depend, in a more complex way, on the characteristics of the loop, hence we also need additional non-terminals to describe e.g. the components of multiloops.

We use a decomposition that is similar to that of `MC-Fold` and in addition encompasses 2-diagrams. A $p\times q$-loop, $p\leq q$, consists of p nucleotides on one strand and q nucleotides on the other one. In particular, 2×2-loops correspond to stacked base pairs, $1\times q$-loops, $q>1$ are triplets and 2×3-loops are stacks with a bulged-out nucleotide. In addition to hairpin loops and these $p\times q$-loops, we consider generic bulges with and without a shared nucleotide, interior loops of larger sizes and multibranched loops,

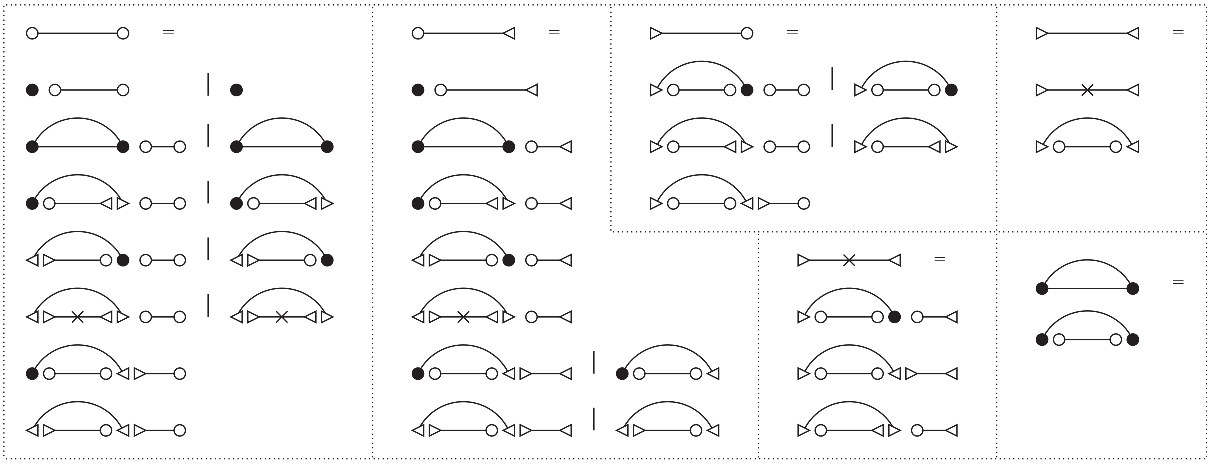

Fig. 3. A simple unambiguous grammar for non-crossing 2-diagrams (The symbols used here denote (non-)terminals in a context-free grammar and are not to be confused with the LW notation used in other figures). Connected parts of diagrams correspond to terminal [individual bullet with no arc (closed circle) = unpaired nucleotide; arc with circular end points (closed circle, open circle) = base pair; arc with triangular endpoints (left-faced triangle, right faced triangle, cross) = part of base triple] or non-terminal (horizontal lines and semicircle) symbols of the grammar. It is important to realize that left-faced and right-faced triangles refer to the same nucleotide when they are adjacent. In terms of a recursion, the index for both left-faced and right-faced triangles is therefore the same. One triangle 'points' to the outer arc and one to the inner arc incident to the same nucleotide.

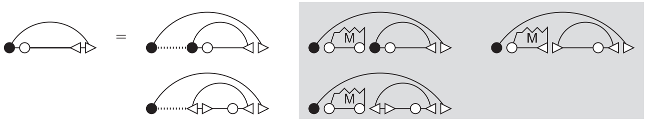

Fig. 4. Decomposition of one non-terminal in the full loop-based model with triples. The l.h.s. of the production rule denotes a structure enclosed by a base pair where the base at the 3′ end is part of a triple. The second base pair of this triple ends within the structure. The structural element is either bulge like (first column) or multiloop like. In the first case, we have to distinguish whether the enclosed structure has a normal pair or a triple at its 5′ side. In the multiloop case, we use the linear decomposition into components familiar from the Turner model with a non-terminal denoting a partial multiloop containing at least one base pair. Here, we need to distinguish whether the 5′ end of the rightmost component and 3′end of the left components are triples or not. As the multiloop part is not implemented in our current version, it is grayed out.

again possibly with shared nucleotides. Figure 4 gives an example for the full loop-based decomposition of one particular non-terminal. In our current implementation, we use several simplifications in particular for multiloops that involve triplets. Some information on the complete grammar used in our implementation can be found in the Appendix A, other information is available on the RNAwolf homepage.

4 IMPLEMENTATION

4.1 Folding software

The implementation available on the RNAwolf homepage is written in the high-level functional programming language Haskell. While this leads to an increase in running times (by a constant factor), the high-level notation and a library of special functions lead to very concise programs, and enable, e.g. the use of multiple cores.

Currently, the following algorithms are implemented: (i) an optimizer which takes a set of melting experiments and the PDB database as input and produces a parameter file optimized as described below. (ii) A folding program which expects a sequence of nucleotides as input and produces an extended secondary structure prediction which includes nucleotide pairs of non-canonical types. Furthermore, it can contain motifs with base triplets. (iii) An evaluation program which expects both, a sequence and a secondary structure. The input is then evaluated to return the score of said structure and, if requested, tries to fill the given (canonical) structure with additional pairs. This allows to turn a classical secondary structure into an extended secondary structure by filling large loops with non-canonical pairs.

At the moment, base triplets have been restricted slightly in that shared nucleotides are only possible in stem structures, not within a multibranched loop motif. Allowing shared nucleotides between two helices of a multiloop would slow down multiloops by a significant factor. Nevertheless, we will lift this restriction for the full nearest neighbor model we plan to implement. In the full model, we will be able to use data gathered from our current model to reduce the combinatorial complexity of the algorithm within multibranched loops.

4.2 Parameter estimation

In contrast to the Turner model, which considers only canonical base pairs [i.e. Watson–Crick and GU (wobble) pairs], we include all types of base pairs. Thus, we also have to derive parameters for all possible base pair families in our motifs of choice. To this end, we need to find sufficient evidence for each parameter and we need an efficient numerical algorithm for optimizing the parameters.

(i) Even if a large body of sequence/structure pairs is available to train the parameters, it is still highly unlikely that each parameter is witnessed. A simple calculation for canonical stacked pairs already produces $4^4 \times 12^2 = 36\,864$ (ignoring symmetries) parameters to be trained. While symmetries reduce the number of distinct parameters, canonical stacks still require $\sim 10\,000$ independent parameters. In total, the number of parameters easily reaches 10^5, which means that only a very small set of parameters will actually be observed in experimentally verified structures.

(ii) The second problem is of numerical nature in that it gets hard to estimate a solution in $\mathbb{R}^{100000}$ even under ideal circumstances. In addition, the computational effort for the computation of the solution vector is rather high. There are two different types of approaches to this problem, described in some detail by Andronescu *et al.* (2007). In max-margin formulations, parameters are optimized such as to drive them away from wrongly determined structures and toward correctly determined ones. Alternatively, the conditional likelihood of known structures is maximized. Andronescu *et al.* (2010) described an extension of the algorithm that can deal with unobserved configurations by employing a hierarchical statistical model.

We have selected yet another way of dealing with the immense number of features. Instead of optimizing the full set of parameters directly, we first optimize the parameters for a restricted model closely following the simple unambiguous grammar given in Figure 3. In short a loop of type m (e.g. stacked pair, bulge, etc.) enclosed by two pairs p_1, p_2 is assigned an energy $\epsilon(m)+\epsilon^m(p_1)+\epsilon^m(p_2)$, where $\epsilon(m)$ depends on the type and size of the loop but is independent of sequence, and the pair energies depend on the identity of the nucleotides as well as the LW type (e.g. GC,cWW).

We call our model *enhanced Nussinov* as it distinguishes between loops of different types (say bulges of different lengths are assigned different scores) but assumes that pair energies are independent as in the Nussinov model.

This approach has several advantages. First, the resulting algorithm is an accessible 'toy-model' that can be employed to test different hypotheses. Second, the estimated parameters provide a useful set of priors for the full model. This is important since, in contrast to the work of Andronescu *et al.* (2007), we cannot derive a complete set of priors from known data. Finally, the computational requirements are significantly lower. Training a full Boltzmann model for conditional likelihood maximization might easily have taken months of CPU time (Andronescu *et al.*, 2010).

Here, we utilize both melting experiments and PDB data for parameter estimation. Melting experiments yield a small set of sequences, structures and corresponding free energies. The structural data, unfortunately, provides almost exclusively canonical Turner features and no information regarding the base pair family, although it can be assumed that all pairs are of the Watson–Crick (cWW) style. The PDB data, on the other hand, contain not only non-canonical base pairs, but also provide information on the base pair family. In addition, PDB entries typically refer to structures that are much larger than those used in melting experiments.

Together, both sets provide data required for the estimation of an extended set of parameters. In order to keep computation times short, we employ the original no-max-margin constraint-generation approach used by Andronescu *et al.* (2007). While not providing the most accurate parameters in the original paper, the relatively short runtimes of $\sim$1 CPU day are convenient for experimental purposes. In addition, since we are training an enhanced Nussinov-style model, we can assume that the prediction accuracy is limited by the structure of the model. More advanced, and hence computationally more expensive, training methods are therefore unlikely to lead to substantial improvements of the prediction accuracy.

4.3 Optimization

Our task is to estimate the energy contributions x_j for a given collection of features j. In this context, a feature corresponds to a terminal symbol in our grammar with a fixed underlying sequence, such as as GC/GC stacked pair or a 1×3-loop with sequence (G—AUC) where GA is a Hoogsteen pair and GC is a Watson–Crick pair. We are given the following types of data: (i) a matrix A whose entries $A_{i,j}$ encode how often feature j occurs in sequence/structure pair i, and (ii) a vector y containing measured melting temperatures y_i for experiment i.

Constraints are now generated as follows. For each entry k of the PDB, we extract the (extended) secondary structure features. This means that neither pseudoknots nor intermolecular interactions (which require more complicated grammars) are considered. The entry f_j^T of the row vector f^T counts how often feature j is observed in the structure. Using the current parameter values x (see below), the sequence of PDB entry k is folded and the corresponding feature vector g^T is constructed. If the predicted fold has a lower free energy than the known structure, a new constraint $(f-g)^Tx\le 0$ is introduced. Note that f^Tx and g^Tx are, by construction, the free energies of the known and the predicted structure evaluated with the current parameters x. Since the true structure is expected to be the thermodynamic ground state, its free energy must be smaller than that of any other structure. The constraint matrix D contains all currently active constraints where $D_{k,\cdot}$ is the k-th active constraint (in this notation $D_{k,\cdot}$ selects the k-th row, while $D_{\cdot,l}$ would select the l-th column).

Following Andronescu *et al.* (2007), we use a slack variable d_k for each constraint so that $D_{k,\cdot}x\le d_k$. This guarantees that the problem remains feasible as otherwise conflicting constraints could reduce the feasible set for x to the empty set. The slack variables d_k are bounded from below by $0\le d_k$ because $(f-g)^Tx\ge 0$, with equality for cooptimal structures.

Norm minimization problems can drive individual variables x_i to extreme values. We, therefore, constrain the energy contribution of individual features to $|x_i|<5$ kcal/mol. A subset S of features that act as penalties are constrained to positive values, $x_j>0$ for $j\in S$. The set S is defined along the following principles: unpaired loop regions destabilize the structure relative to a random coil and hence should be penalized. Hairpins, bulges and interior loops fall into this category. In addition, 1×2 and 2×3 stems, which are otherwise modeled as 2×2 stems, are penalized. Hence, for e.g. the 2×3 loop `CAUGG` with `A` unpaired, we have $\epsilon(CG)+\epsilon(UG)+\epsilon(2\times 3)$ where $\epsilon(2\times 3)$ is the penalty term.

Parameter estimation is thus reduced to the constrained norm optimization problem

$$\left\| \begin{pmatrix} A & 0 \\ D & -I \end{pmatrix} \begin{pmatrix} x \\ d \end{pmatrix} - \begin{pmatrix} y \\ 0 \end{pmatrix} \right\|_2 \qquad (4.1)$$

with the linear constraints

$$-5<x_j<5, \qquad 0<x_l, \quad l\in S, \qquad 0<d_k. \qquad (4.2)$$

Since this optimization problem is convex it can be solved efficiently.

The parameter vector x is optimized iteratively. Initially, D is empty and no slack variables d are used. After the first step, all PDB sequences have been folded and those for which the predicted structure is different from the known structure are included as a row in D as described above. The slack variables are initialized as $d_k=D_{k,\cdot}x+\gamma$ for each constraint k, where $\gamma\in\mathbb{R}_+$ is a small constant. Iterations of the optimization procedure continue until no more constraints have to be added.

The computational effort required, both to estimate the parameters and to fold a single sequence, is higher than what is required for the Turner model. The additional computational effort required by the folding algorithm is mainly a result of the inclusion of the pair family information. In the case of 2-loops (stacks, bulges, interior loops), we incur an additional factor of 12 since each possible pair family has to be considered. More problematic are multibranched loops in the case of shared nucleotides as now there are up to 12×11 possibilities to connect a shared nucleotide with its pairing partners.

4.4 Comparison with turner parameters

A comparison with the parameter sets by Turner (Turner and Mathews, 2010) shows that individual contributions are similar enough to make the the 'enhanced Nussinov' model a useful prior in the parameter optimization for the full model. Consider, for example, canonical 2×2 stacks, where one pair is of type **GC, cWW** type and the other pair is of type **XY, cWW**, with **XY** $\in${**GC,CG,AU,UA,GU,UG**} and **cWW** stands for *cis/Watson–Crick/Watson–Crick*, the canonical pair type. In the Turner-2004 model, energy contributions range from -1.5 to -3.4 kcal/mol, while the base-pair contribution for the **GC, cWW** pair is -1.36 kcal/mol in the optimized 'enhanced Nussinov' model. Depending on the second pair, we observe discrepancies of $\approx$0.5 when comparing the sum of individual pair energies to the total stacking energy. This level of agreement is expected and suggests that it makes sense in later iterations of parameter estimation to constrain features to tighter intervals than the current setting of using the open interval of $]-5,5[$.

5 RESULTS AND DISCUSSION

MC-Fold-DP: `MC-Fold-DP` and the original `MC-Fold` by (Parisien and Major, 2008) show comparable performance on a

Table 1. Prediction accuracy of `MC-Fold`, `MC-Fold-DP` and `RNAfold 1.8.4` on a set of 347 sequences from the RNAstrand database

Algorithm	Count	MCC	*F*	S	PPV
`MC-Fold`, ≤50 nt	298	0.74	0.74	0.80	0.70
`MC-Fold`, ≤100 nt	37	0.54	0.54	0.66	0.46
`MC-Fold`, >100 nt	12	0.49	0.49	0.53	0.46
`MC-Fold-DP`, ≤50 nt	298	0.71	0.71	0.77	0.68
`MC-Fold-DP`, ≤100 nt	37	0.53	0.53	0.64	0.45
`MC-Fold-DP`, >100 nt	12	0.38	0.37	0.51	0.29
`RNAfold`, ≤50 nt	298	0.76	0.76	0.73	0.81
`RNAfold`, ≤100 nt	37	0.73	0.73	0.73	0.73
`RNAfold`, >100 nt	12	0.63	0.63	0.66	0.60

All sequences are <200 nt long. The longest sequence took just under an hour of computation time using `MC-Fold`. `MC-Fold-DP` can compute the predicted structure in ∼1 s (loading the `MC-Fold` motif database requires an additional 1–2 s). Prediction quality has been measured on canonical base pairs only for comparison purposes. Note the small number of sequences >100 nt. (MCC, Matthews correlation coefficient; F, F-Measure; S, Sensitivity; PPV, Positive Predictive Value).

set of 347 sequences selected from the RNAstrand (Andronescu *et al.*, 2008) database. There are several differences between the two algorithms. First, the runtime, where `MC-Fold-DP` is about ×200−×1000 faster for biologically relevant sequences (i.e. <1000 nt). Table 1 shows a small comparison of the prediction accuracy given different measures. Second, we allow for sparse data correction, which can be disabled by the user. And third, the algorithm accepts non-canonical input (e.g. 'N' characters) and can be configured to calculate approximate scores for motifs containing such characters.

Differences in predictions are the result of internals of the orignial algorithm that have remained unknown to us since they are not described in full detail in Parisien and Major (2008).

It should be noted that our reformulation makes `MC-Fold-DP` amenable for the parameter optimization approaches pioneered by Andronescu *et al.* (2010) for which a polynomial-time prediction algorithm is crucial. The non-ambiguous grammar allows even the advanced, Boltzmann Likelihood-based, approaches to be employed. This presents an opportunity for future research.

RNAwolf: we compared our *enhanced Nussinov* algorithm to three state-of-the-art thermodynamic folding algorithms [RNAfold (Hofacker *et al.*, 1994), UNAFold (Markham and Zuker, 2008) and RNAstructure (Reuter and Mathews, 2010)] to assess the prediction quality of our model. We folded a subset of 550 randomly chosen structures from RNAstrand (Andronescu *et al.*, 2008) and compared the *F*-measure of our results with those of the other programs. The results in Figure 5A show that, not unexpectedly, the 'enhanced Nussinov' algorithm cannot compete with state-of-the-art tools due to its simplified energy model.

Interestingly, once we focused on data gathered from the PDB database (Fig. 5B), the results showed a remarkable improvement. This could suggest that the PDB structures used for training do not sufficiently cover the RNA structure space and that additional RNAs (for which only secondary structure information is available) should be included in the training.

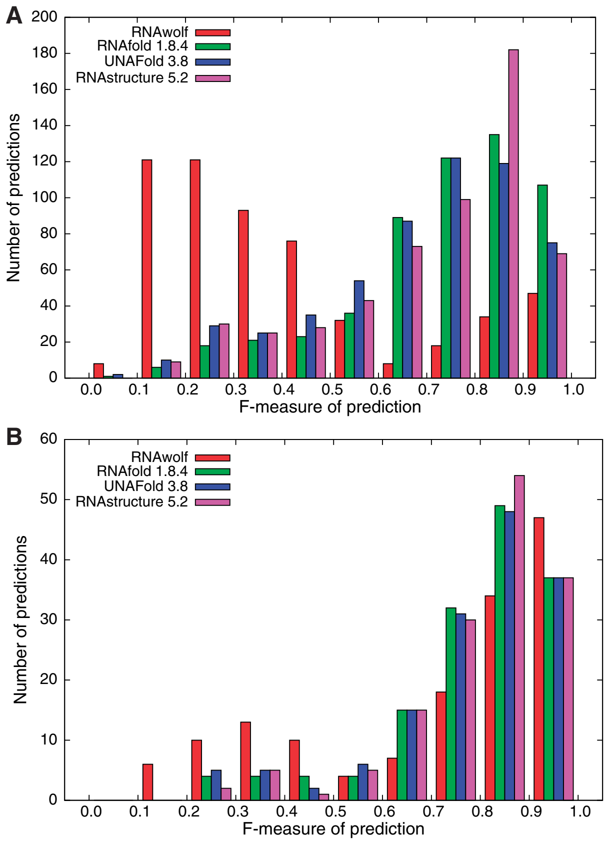

Fig. 5. (**A**) Histogram of *F*-measures for different folding algorithms, given 550 random RNAstrand entries. (**B**) *F*-measures, given 155 PDB entries from the RNAstrand, which are a subset of the 550 random RNAstrand entries.

Because of the large number of base pair types, the 'enhanced Nussinov-algorithm', has to perform more work than classical secondary structure prediction programs when filling the dynamic programming matrices. This is reflected by rather high runtimes (25 s for 100 nt, 110 s for 200 nt). However, the asymptotic time complexity is still in $\mathcal{O}(n^3)$.

A constrained folding variant of the 'enhanced-Nussinov' algorithm can be used, for example, to predict non-canonical base pairs in large interior loops of structures. As an example, Figure 6, shows that `RNAwolf` is able to correctly predict the non-canonical base pairs in a situation where the canonical base pairs are already given, i.e. where the input consists of both the sequence and a dot-bracket string representing canonical Watson–Crick base pairs. Only the zig-zag motif (upper part of the interior loop) was not predicted, presumably due to the large penalty of +3.89 for each of the two 1×2 stacks.

Further results and a semi-automatic system for secondary structure prediction comparison (`SSPcompare`) are available on the `RNAwolf` homepage. Table 1 has been created using said program.

6 CONCLUSION

Large experimentally verified RNA structures contain a sizable number of non-canonical base pairs (Stombaugh *et al.*, 2009).

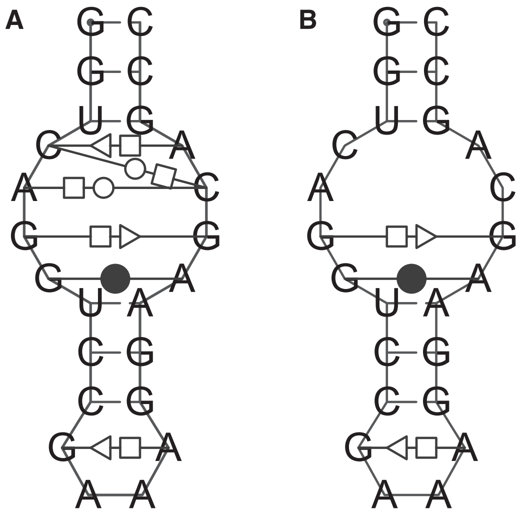
A
B
G C G C U G C A A C G G G A U A C G C G G A A A
G C G C U G C A A C G G G A U A C G C G G A A A

Fig. 6. Prediction of non-canonical base pairs with RNAwolf. (**A**) Known structure of PDB entry 1dul. (**B**) Constrained prediction (canonical base pairs were given) of 1dul. Only the central part of the structure is shown. The outer part of the stem contains only canonical base pairs and is not shown.

However, only a few RNA folding programs predict non-canonical pairs (Do *et al.*, 2006; Parisien and Major, 2008). With the exception of MC-Fold, the pair families are not explicitly taken into account. Here, we have shown that the prediction of non-canonical pairs together with the corresponding pair families and their possible interactions in base triples is feasible by efficient dynamic programming approaches. Although direct thermodynamic measurements are not available to cover all aspects of such an extended and refined model of RNA structures, meaningful parameter sets can nevertheless be constructed. To this end, the information of the thermodynamic measurements is combined with a feature analysis of 3D structures using one of several approaches to large-scale parameter optimization. The extended combinatorial model, which in essence covers the LW representations of RNA structures, allows a much more detailed modeling of the intrinsic structures in particular of hairpins, interior loops and bulges.

We emphasize that our contribution does not yet provide a full-fledged loop-based LW-style energy model. In essence, we still lack an implementation for the full model of multiloops. As the example of Figure 6 suggests, interactions of adjacent loops as in the MC-Fold model may also be required to obtain satisfactory prediction accuracies for practical applications. Due to the computational cost, it will also make sense to investigate the trade-off between further refinements of the model and speed-ups resulting from additive approximations. Another facet that naturally should be taken into account is coaxial stacking, in particular in the context of multiloops (Tyagi and Mathews, 2007). We have demonstrated here that the goal of an accurate, practically applicable folding algorithm for LW structures is meaningful and reachable: the work of Parisien and Major (2008) shows that major improvements of prediction accuracy can be obtained by employing LW-based folding algorithms. Although RNAwolf does not yet reach the desired levels of accuracy, it allows us to explore the missing components of the energy model in a systematic manner, and it demonstrates that this can be achieved without leaving the realm of fast, efficient and exact dynamic programming approaches. The next step, therefore, is a toolkit for optimizing parameters in the full loop-based model.

An interesting possibility for further extensions of the model is the explicit incorporation of recurring RNA structural motifs with non-canonical pairs, such as Kink-Turns (Klein *et al.*, 2001), into the grammar and the energy model. This may be particularly useful in those cases where motifs are not crossing-free and hence would require a pseudoknot version of the folding algorithm. While the inclusion of various types of pseudoknots is conceptually not more difficult than for ordinary secondary structures, the parametrization of such models will be even more plagued by the lack of training data in the LW framework.

The folding algorithm introduced here, furthermore, sets the stage for a complete suite of bioinformatics tools for LW structures. Simple extension can cover the cofolding of two or more RNAs along the lines of (Bernhart *et al.*, 2006; Dimitrov and Zuker, 2004; Dirks *et al.*, 2007). Consensus structures can be predicted from given sequence alignments using the same recursions. As in RNAalifold (Bernhart *et al.*, 2008), it suffices to redefine the energy parameters for alignment columns instead of individual nucleotides. Instead of RIBOSUM-like scores as measures of conservation (Klein and Eddy, 2003), one naturally would employ the isostericity rules for the individual base pair types (Leontis *et al.*, 2002; Lescoute *et al.*, 2005). Inverse folding algorithms (Andronescu *et al.*, 2004; Busch and Backofen, 2006; Hofacker *et al.*, 1994) design RNA sequences that fold into prescribed structures by iteratively modifying and folding sequences to optimize their fit to substructures of the target. This strategy can immediately be generalized to LW structures; in fact, in essence it suffices to replace secondary structure folding by LW style folding. Combining the algorithmic ideas of this contribution with the Sankoff-style alignment approach of Zhong *et al.* (2010) and the progressive multiple alignment scheme of mlocarna (Will *et al.*, 2007) directly leads to an LW variant of structural alignment algorithms.

ACKNOWLEDGEMENTS

The authors thank the participants of the *Refined presentation of RNA structures* workshop for lively discussions, Marc Parisien for kindly answering questions about MC-Fold, the curators of the FR3D database, and Ronny Lorenz for providing comparative data for other folding programs.

Funding: Austrian GEN-AU projects ‘bioinformatics integration network III’ and ‘regulatory ncRNAs’ in part.

Conflict of Interest: none declared.

REFERENCES

Andronescu,M. *et al.* (2004) A new algorithm for RNA secondary structure design. *J. Mol. Biol.*, **336**, 607–624.

Andronescu,M. *et al.* (2007) Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics*, **23**, i19–i28.

Andronescu,M. *et al.* (2008) RNA STRAND: The RNA secondary structure and statistical analysis database. *BMC Bioinformatics*, **9**, 340.

Andronescu,M. *et al.* (2010) Computational approaches for RNA energy parameter estimation. *RNA*, **16**, 2304–2318.

Bafna,V. *et al.* (2006) Consensus folding of unaligned RNA sequences revisited. *J. Comput. Biol.*, **13**, 283–295.

Bernhart,S.H. *et al.* (2006) Partition function and base pairing probabilities of RNA heterodimers. *Algorithms Mol. Biol.*, **1**, 3.

Bernhart,S.H. *et al.* (2008) RNAalifold: improved consensus structure prediction for RNA alignments. *BMC Bioinformatics*, **9**, 474.

Busch,A. and Backofen,R. (2006) INFO-RNA — a fast approach to inverse RNA folding. *Bioinformatics*, **22**, 1823–1831.

Dimitrov,R.A. and Zuker,M. (2004) Prediction of hybridization and melting for double-stranded nucleic acids. *Biophys. J.*, **87**, 215–226.

Dirks,R.M. *et al.* (2007) Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev.*, **49**, 65–88.

Dowell,R.D. and Eddy,S.R. (2004) Evaluation of several lightweight stochastic context-free grammars for RNA secondary structure prediction. *BMC Bioinformatics*, **5**, 7.

Do,C.B. *et al.* (2006) CONTRAfold: RNA secondary structure prediction without physics-based models. *Bioinformatics*, **22**, e90–e98.

Do,C.B. *et al.* (2008) Efficient multiple hyperparameter learning for log-linear models. In Platt,J.C. *et al.* (eds), *Advances in Neural Information Processing Systems 20. Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, 2007*, MIT Press, pp. 3–6.

Hofacker,I.L. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Mh. Chemie/Chem. Mon.*, **125**, 167–188.

Klein,D.J. *et al.* (2001) The kink-turn: a new RNA secondary structure motif. *EMBO J.*, **20**, 4214–4221.

Klein,R.J. and Eddy,S.R. (2003) RSEARCH: finding homologs of single structured RNA sequences. *BMC Bioinformatics*, **4**, 44.

Leontis,N.B. and Westhof,E. (2001) Geometric nomenclature and classification of RNA base pairs. *RNA*, **7**, 499–512.

Leontis,N.B. *et al.* (2002) The non-Watson-Crick base pairs and their associated isostericity matrices. *Nucleic Acids Res.*, **30**, 3497–3531.

Lescoute,A. *et al.* (2005) Recurrent structural RNA motifs, isostericity matrices and sequence alignments. *Nucleic Acids Res.*, **33**, 2395–2409.

Leontis,N.B. and Lescoute,A.W.E. (2006) The building blocks and motifs of RNA architecture. *Curr. Opin. Struct. Biol.*, **16**, 279–287.

Markham,N.R. and Zuker,M. (2008) UNAFold: software for nucleic acid folding and hybridization. *Methods Mol. Biol.*, **453**, 3–31.

Parisien,M. and Major,F. (2008) The MC-Fold and MC-Sym pipeline infers RNA structure from sequence data. *Nature*, **452**, 51–55.

Reeder,J. *et al.* (2005) Effective ambiguity checking in biosequence analysis. *BMC Bioinformatics*, **6**, 153.

Reuter,J.S. and Mathews,D.H. (2010) RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**, 129–129.

Sankoff,D. (1985) Simultaneous solution of the RNA folding, alignment, and protosequence problems. *SIAM J. Appl. Math.*, **45**, 810–825.

Stombaugh,J. *et al.* (2009) Frequency and isostericity of RNA base pairs. *Nucleic Acids Res.*, **37**, 2294–2312.

Turner,D.H. and Mathews,D.H. (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, **38**, D280–D282.

Tyagi,R. and Mathews,D.H. (2007) Predicting helical coaxial stacking in RNA multibranch loops. *RNA*, **13**, 939–951.

Will,S. *et al.* (2007) Inferring non-coding RNA families and classes by means of genome-scale structure-based clustering. *PLoS Comput. Biol.*, **3**, e65.

Wuchty,S. *et al.* (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers*, **49**, 145–165.

Zhong,C. *et al.* (2010) `RNAMotifScan`: automatic identification of RNA structural motifs using secondary structural alignment. *Nucleic Acids Res.*, **38**, e176.

APPENDIX A

A PAIRFAMILY-AWARE GRAMMAR

Here, we discuss in some more detail how the base pair types affect the grammar and, hence, the folding algorithm. We start from Figure 3 and the corresponding recursion in Equation (3.1). Each base pair is now colored by its LW family. In particular, therefore, base pairs have type-dependent energy contributions $\epsilon_{ij}[\vartheta]$ for pairs not involved in base triples and energy contributions depending on the type of the pair and on the type of the incident pairs: $\epsilon^{a}_{ij}[\vartheta,\psi]$ if the $5'$ nucleotide i is a triplet, $\epsilon^{a}_{ij}[\vartheta,\phi]$ if the $3'$ nucleotide i is a triplet and $\epsilon^{c}_{ij}[\vartheta,\psi,\phi]$ if both delimiting nucleotides are triplets. Similarly, therefore, non-terminals delimited by triples must be colored by the base pair type(s) to allow the evaluation of the energy of the enclosing base pair. In the simplest case, as implemented in `RNAwolf`, we may assume that $\epsilon^{b}_{ij}[\theta,\psi]$ only depends on the pair type θ for permitted combinations of pair types and is $+\infty$ otherwise. With explicit representation of the pair family types, Equation (3.1) becomes

$$F_{ij}=\min\begin{cases} F_{i+1,j}, C_{ij}, U'_{ij}, V'_{ij}, W'_{ij} \\ \min\limits_{i<k<j}\begin{cases} C_{ik}+F_{k+1,j} \\ V'_{i+1,k}+F_{k+1,j} \\ U'_{i,k-1}+F_{k+1,j} \\ W'_{ij}+F_{i+1,k} \\ F_{i+1,k-1}+\min\limits_{\theta,\psi}\left\{U_{kj}[\psi]+\epsilon^{a}_{ik}[\theta,\psi]\right\} \\ \min\limits_{\theta,\psi,\phi}\left\{U_{i,k-1}[\psi]+U_{k,j,\phi}+\epsilon^{c}_{ik}[\theta,\psi,\phi]\right\} \end{cases} \end{cases}$$

Here, we use the abbreviations

$$U'_{ij}=\min_{\theta,\psi}\left(U_{i,j-1}[\theta]+\epsilon^{a}_{ij}[\theta,\psi]\right)$$

$$V'_{ij}=\min_{\theta,\psi}\left(V_{i+1,j}[\theta]+\epsilon^{b}_{ij}[\theta,\psi]\right)$$

$$W'_{ij}=\min_{\theta}\left(W_{i,j}[\psi,\phi]+\epsilon^{c}_{ij}[\theta,\psi,\phi]\right)$$

which are obtained by carrying out the optimization over the combinations of base pairing types at all triples.

The non-terminal C, designating a structure enclosed by an ordinary base pair remains unchanged since the minimization $C_{ij}=F_{i+1,j-1}+\min_{\theta}\epsilon_{ij}[\vartheta]$ can be carried out in the simplified energy model. The triplet terms, however, are now conditioned on the pair family at all nodes represented as triangles in Figure 3. For instance, for a structure delimited by triplet vertices at both ends which are not connected by a pair, we obtain a recursion of the form

$$W^{*}_{ij}[\theta,\psi]=\min_{i<k<j}\begin{cases} F_{i+1,k-1}+\epsilon^{a}_{ik}[\theta]+U_{k+1,j}[\psi] \\ \min_{\phi}F_{i+1,k-1}+\epsilon^{b}_{ik}[\theta,\phi]+W_{k+1,j}[\phi,\psi] \\ \min_{\phi}V_{i+1,k}[\phi]+\epsilon^{b}_{ik}[\theta,\phi]+V_{k+1,j}[\psi] \end{cases}$$

and $W_{ij}[\theta,\psi]=W^{*}_{ij}[\theta,\psi]$ if $\theta\neq\psi$ and $W_{ij}[\theta,\theta]=\min\{W^{*}_{ij}[\theta,\theta],F_{i+1,j-1}+\epsilon_{ij}[\theta]$.

Similar recursions are obtained for the full loop-based model. For instance, for the two interloop terms in Figure 4 we have to compute

$$V^{*}_{ij}[\theta,\psi]=\min_{k,l,\psi}\begin{cases} \mathfrak{I}[i,j,\theta|k,l\psi]+V_{kl}[\psi] \\ \mathfrak{I}'[i,j,\theta|k,l\psi]+\min_{\phi}W_{kl}[\psi,\phi,\theta] \end{cases}$$

where the matrices V^{*}, V and W now refer to the non-terminal symbols in Figure 4 and $\mathfrak{I}[\dots]$ and $\mathfrak{I}'[\dots]$ denote the tabulated energy contributions for the two different types of interior loops with $3'$-triplet. For more detail, we refer to the Supplementary Material which we will make available together with the full loop-based model on the RNAwolf homepage.