*Bioinformatics*, 35(21), 2019, 4298–4306
doi: 10.1093/bioinformatics/btz222
Advance Access Publication Date: 28 March 2019
Original Paper

Structural bioinformatics

# Determining parameters for non-linear models of multi-loop free energy change

**Max Ward[1,*], Hongying Sun[2,3], Amitava Datta[1], Michael Wise[1,4] and David H. Mathews[5]**

[1]Computer Science & Software Engineering, The University of Western Australia, Crawley, WA, Australia, [2]Department of Biochemistry & Biophysics, University of Rochester, Rochester, NY, USA, [3]Center for RNA Biology, University of Rochester, Rochester, NY, USA, [4]The Marshall Centre for Infectious Diseases Research and Training, The University of Western Australia, Crawley, WA, Australia and [5]Department of Biostatistics & Computational Biology, University of Rochester, Rochester, NY, USA

*To whom correspondence should be addressed.

Associate Editor: Alfonso Valencia

Received on September 24, 2018; revised on February 10, 2019; editorial decision on March 9, 2019; accepted on March 27, 2019

## Abstract

**Motivation:** Predicting the secondary structure of RNA is a fundamental task in bioinformatics. Algorithms that predict secondary structure given only the primary sequence, and a model to evaluate the quality of a structure, are an integral part of this. These algorithms have been updated as our model of RNA thermodynamics changed and expanded. An exception to this has been the treatment of multi-loops. Although more advanced models of multi-loop free energy change have been suggested, a simple, linear model has been used since the 1980s. However, recently, new dynamic programing algorithms for secondary structure prediction that could incorporate these models were presented. Unfortunately, these models appear to have lower accuracy for secondary structure prediction.

**Results:** We apply linear regression and a new parameter optimization algorithm to find better parameters for the existing linear model and advanced non-linear multi-loop models. These include the Jacobson-Stockmayer and Aalberts & Nandagopal models. We find that the current linear model parameters may be near optimal for the linear model, and that no advanced model performs better than the existing linear model parameters even after parameter optimization.

**Availability and implementation:** Source code and data is available at https://github.com/maxhwardg/advanced_multiloops.

**Contact:** max.ward-graham@research.uwa.edu.au

**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

## 1 Introduction

Ribonucleic acid (RNA) is a fundamental molecule in biology. It has dual purpose, serving as both a carrier of genetic information and as a functional molecule. This duality underlies the RNA world hypothesis, which posits that life started with RNA as the primary building block (Bernhardt, 2012; Gilbert, 1986). In fact, it has been suggested that eukaryote development is facilitated by an RNA machine consisting of a cascade of functional RNAs (Amaral *et al.*, 2008). Non-coding RNAs (ncRNAs) are transcribed to serve a biological purpose outside of being translated into protein. For example, ncRNAs can act as catalysts (Doudna and Cech, 2002), regulate gene expression (He and Hannon, 2004; Meister and Tuschl, 2004), and can recognize sequences (Kiss-László *et al.*, 1996).

Because RNA secondary structure informs the complete RNA structure (Tinoco and Bustamante, 1999), and because structure informs function, there has been significant interest in determining RNA secondary structure. Experimental methods, such as nuclear magnetic resonance, cryo-electron microscopy, and X-ray crystallography, require significant investment and effort (Neidle, 2010). Comparative methods, which exploit conservation of structure through sequence covariation, require the sequences from many species, covering a specific range of nucleotide substitutions (Pace *et al.*, 1999; Rivas *et al.*, 2017). The third type of method finds an RNA secondary structure given just the primary sequence. We call these *de novo* methods. Because these do not require auxiliary information, and because they generally under other methods, they have been an important area of research (Lorenz *et al.*, 2011; Mathews, 2006; Reuter and Mathews, 2010; Zuker, 2003).

We refer to *de novo* RNA secondary structure prediction as 'RNA folding'. The dominant model for RNA folding algorithms is a thermodynamic model, which uses a *nearest neighbor model* to estimate folding stability. The nearest neighbor model was determined by Turner and coworkers (Andronescu *et al.*, 2014; Jaeger *et al.*, 1989; Mathews *et al.*, 1999, 2004; Xia *et al.*, 1998), and is based on groundwork laid by Tinoco *et al.* (1971, 1973) and Salser (1978). Zuker and Stiegler (1981) developed a dynamic programing algorithm that was able to use the nearest neighbor model to fold RNAs. Their recursions proved robust, and have been modified as the model was revised and expanded. Some examples of this include the introduction of coaxial stacking (Mathews *et al.*, 2004; Rivas and Eddy, 1999), the extension to include some types of pseudoknots (Rivas and Eddy, 1999), and the inclusion of SHAPE information (Deigan *et al.*, 2009). An exception from this, however, is the treatment of multi-loops.

Multi-loops (see Fig. 1) are loops closed by three or more base pairs. Multi-loop free energy change in RNA folding algorithms was first described in its current form by Zuker and Sankoff (1984). They used a linear model on the number of unpaired nucleotides ($u$) and branches ($b$) in a multi-loop. As such, we refer to this model as the *linear model*. The parameters for the current state of the linear model comes from work by Mathews and Turner (2002), and is found in modern RNA folding packages (Lorenz *et al.*, 2011; Reuter and Mathews, 2010). It has the following form (in *kcal/mol*):

$$\Delta G^{\circ}_{37} = 9.3 - 0.6b + 0u \tag{1}$$

Other models have been suggested. Notably, a model based on Jacobson-Stockmayer polymer theory (Jacobson and Stockmayer, 1950) was suggested by Salser (1978), but it was not included in RNA folding dynamic programing algorithms until recently (Ward *et al.*, 2017). Despite this, it is used in RNA folding packages (Lorenz *et al.*, 2011; Reuter and Mathews, 2010; Zuker, 2003) for energy calculation. The most recent parameters for this model were published by Mathews *et al.* (1999). We refer to this model as the *logarithmic model* due to its logarithmic dependence on unpaired nucleotides. The logarithmic model is described as follows (in *kcal/mol*):

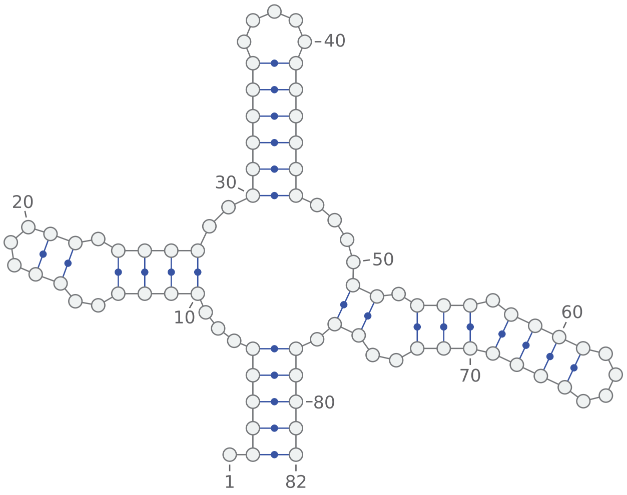

**Fig. 1.** An example of a multi-loop. The central loop with four exiting branches is a multi-loop

$$\Delta G^{\circ}_{37} = \begin{cases} 10.1 - 0.3b - 0.3u & \text{if u} \leq 6 \\ 10.1 - 0.3b - 0.3 \times 6 + 1.1 \times \ln(u/6) & \text{otherwise} \end{cases} \tag{2}$$

A third model was proposed by Aalberts and Nandagopal (2010). We refer to this as the *AN model* after the authors' initials. The AN model defines multi-loop free energy change as a function of the number of length-*a* and -*b* segments in a multi-loop. These segment lengths represent the distance between consecutive nucleotides in the RNA backbone, and across multi-loop helical branches respectively. Call the number of length-*a* segments $N$, and the number of length-*b* segments $M$. The model is defined (in *kcal/mol*) to be:

$$\Delta G^{\circ}_{37} = \frac{59}{36} kT \ln\left(N^{\frac{6}{5}}a^2 + M^{\frac{6}{5}}b^2\right) + C \tag{3}$$

In this equation, the values of $a$ and $b$ are defined in angstroms as $a = 6.2$ and $b = 15$. Also, $k$, $T$, and $C$ refer to the Boltzmann constant, the absolute temperature, and a scaling factor, respectively. By default, the temperature typically is set to 310.15 K. Also, $C$ typically should be set to zero as suggested by Aalberts and Nandagopal (2010).

The first efficient RNA folding algorithms using the logarithmic and AN models were published recently (Ward *et al.*, 2017). These algorithms represent new recursions that use dynamic programing to minimize free energy under the nearest neighbor thermodynamic model. Substantial changes to the existing recursions were needed to incorporate the new, non-linear models. Unexpectedly, the linear model was superior for predicting the secondary structure. This may be because the linear model has been actively used for decades, and its parameters underwent several refinements. Early approaches used parameters optimized for prediction accuracy (Jaeger *et al.*, 1989), as did later approaches (Andronescu *et al.*, 2007; Mathews *et al.*, 1999, 2004). The parameters in Equation (1) come from linear regression (Mathews and Turner, 2002) using a set of experimental data, as do the majority of parameters in the Turner model. In contrast, the parameters for the logarithmic model are partially theoretical, and partially optimized for prediction accuracy (Mathews *et al.*, 1999). The AN model parameters are purely theoretical (Aalberts and Nandagopal, 2010).

To definitively compare the linear, logarithmic, and AN models, optimized parameters for the logarithmic and AN models were needed. We apply linear regression with an updated dataset to find new parameters for all models. In addition, we devise a new algorithm to optimize the parameters of all the models for prediction on known structures. Our algorithm is well suited to optimizing small parameter sets, such as multi-loop models. We also use these parameters to make predictions on a large set of RNAs with known structures. The results are compared with determine which model is most accurate for structure prediction.

RNA folding methods that do not require the experimentally based nearest neighbor model exist. These include statistical methods. Rivas reviewed the current state of RNA folding algorithms and contrasted minimum free energy (MFE) folding algorithms to statistical methods (Rivas, 2013). Statistical methods include algorithms such as CONTRAfold (Do *et al.*, 2006) and TORNADO (Rivas *et al.*, 2012). In a sense, they generalize the MFE algorithms, as they free themselves from a thermodynamic model, and can learn from known structures via statistical training. Rivas found that, while the statistical methods are able to explore different models of RNA folding; only modest improvements over the classical MFE folding algorithms can be achieved at present.

Statistical RNA folding methods have convenient algorithms for training parameters for new models. We opted, however, to not to use a statistical method for two reasons. First, we are focusing on thermodynamic models, and the logarithmic and AN models nearest neighbor models are thermodynamic in origin. It is informative to analyze thermodynamic parameters for these models as they can be put into the context of existing parameters. Second, the existing implementations of folding algorithms for the logarithmic and AN models are for MFE structure prediction (Ward *et al.*, 2017). Re-implementing them for a statistical method would require substantial work. Similarly, implementing a partition function variant of the two algorithms, to estimate structure formation probabilities, would require substantial work. The probabilities would be required for some types of training (Andronescu *et al.*, 2010) and for a maximum expected accuracy folding algorithm (Lu *et al.*, 2009). Free energy minimization provides the most probable structure under the thermodynamic model, and there is much to learn from predictions of the most probable structure.

## 2 Materials and methods

### 2.1 Nearest neighbor model derivation through linear regression for multi-loops

Folding stabilities for RNA multi-branch loops were collected (Diamond *et al.*, 2001; Hill and Schroeder, 2017; Liu *et al.*, 2011; Mathews and Turner, 2002). Multiple linear regression (Cohen *et al.*, 2013), which fits model parameters to best match the data, was performed on this dataset using R (Ihaka and Gentleman, 1996). To select the set of parameters that are significant to the stability of multi-loops, we use a stepwise algorithm to prevent overfitting, and the criterion we choose is the Akaike Information Criterion (AIC) (Akaike, 1998; Chambers *et al.*, 1990):

$$\mathrm{AIC} = -2\ln(L) + 2k \tag{4}$$

The larger the likelihood value ($L$) is, the smaller the AIC. Likewise, the fewer the number of parameters in the model ($k$), the smaller the AIC. The best model is the model with the smallest AIC value. When applying the stepwise algorithm, the direction could be forward, backward, or both (Efroymson, 1960). The method starts with a full model. At each step, it works by evaluating AIC values for dropping each candidate parameter, and for adding each candidate parameter between the current model and the full model. A parameter is then either removed or added if a better AIC can be achieved. The steps continue until the lowest AIC value is reached.

The likelihood value ($L$) of a given model is the probability of a proposed model being true given the dataset (Jeffreys, 1998):

$$L(\theta_1, \theta_2, \theta_3, \ldots, \theta_k, \sigma^2 | \Delta G^\circ_{37}) = \prod_{i=1}^{n} p(\Delta G^\circ_{37}(i)) \tag{5}$$

Here, $\theta$ is a parameter in the proposed model; $\sigma$ is standard error; $p(\Delta G^\circ_{37}(i))$ is the probability of predicting observation $i$:

$$p(\Delta G^\circ_{37} | \theta_1, \theta_2, \theta_3, \ldots, \theta_k, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(\Delta G^\circ_{37} - \Delta G'^\circ_{37})}{2\sigma^2}} \tag{6}$$

In this equation, $\Delta G^\circ_{37}$ is the experimental value for observation $i$; $\Delta G'^\circ_{37}$ is the predicted value using the proposed model and this value can be calculated using:

$$\Delta G'^\circ_{37} = \sum_{i=1}^{|\theta|} \theta_i f_i \tag{7}$$

where $\theta$ is a parameter value vector and $f$ is parameter frequency vector.

### 2.2 An algorithm for optimizing multi-loop model parameters

We applied algorithmic optimization of model parameter performance for RNA folding to the linear, logarithmic, and AN models. This has precedent (Jaeger *et al.*, 1989; Mathews *et al.*, 1999, 2004). A notable advance was the iterative constraint generation (ICG) technique of Andronescu *et al.* (2007). We call these algorithms *parameter optimization* algorithms.

To begin, some fundamental definitions should be provided. A parameter optimization algorithm aims to optimize the values assigned to parameters in a model so that folding accuracy is good on known structures in a training set. This is different to a technique like linear regression, which fits parameters to experimental values.

To use such an algorithm, we need a set $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$ containing known RNA sequence–structure pairs on which to optimize performance. $x_i$ denotes a sequence and $y_i$ denotes the corresponding structure. A nearest neighbor model can be said to score an RNA sequence–structure pair's folding free energy change as the sum of the free energy changes of features—such as a specific type of hairpin loop, or a branch in a multi-loop (Mathews *et al.*, 1999, 2004; Turner and Mathews, 2009). A model therefore informs how to score each of these features. As such, each feature corresponds uniquely to a parameter in a model. Thus, a model is described by a parameter vector $\theta$ with a dimension for each feature. Each element in $\theta$ is the free energy change corresponding to a particular feature. An RNA sequence–structure pair $p$ is described by a feature vector $f(p)$ with $|\theta|$ dimensions where $f(p)_i$ is the frequency of feature $i$ in $p$. The free energy of $p$ is thus provided by Equation (7).

#### 2.2.1 The ICG algorithm

Our solution to parameter optimization for multi-loop models modifies the constraint generation approach of Andronescu *et al.* (2007, 2010). We refer to their algorithm as the *ICG* algorithm.

The ICG algorithm starts with an arbitrary parameter vector $\theta$, and attempts to iteratively improve it (Andronescu *et al.*, 2007). As with most parameter optimization algorithms, the algorithm must start with a set $S$ of known sequence–structure pairs. Every primary sequence in $S$ is then folded subject to $\theta$. The resulting predictions can be represented as a set $S'$ of sequence–structure pairs. Each $p' \in S'$ *corresponds* to a true sequence–structure pair $p \in S$, which has the same primary sequence, but may have a different structure. Note that we refer to the corresponding true sequence–structure pair of a folded pair $p'$ as $p$.

Let us define the total free energy change of a sequence–structure pair $x$ using a parameter vector $\theta$ to be $\Delta G^\circ(x, \theta)$. Now, if we have two sequence–structure pairs, a true structure $p$, and a corresponding pair $p'$ such that $p' \neq p$, then we would expect $\Delta G^\circ(p, \theta) < \Delta G^\circ(p', \theta)$ if $\theta$ were an optimal parameter vector. This is because a true structure should be the MFE structure for a given primary sequence at equilibrium. It follows that $\sum_{i=1}^{|\theta|} \theta_i f(p)_i < \sum_{i=1}^{|\theta|} \theta_i f(p')_i$. This gives us a constraint for each $p' \in S'$ such that $p \neq p'$. If we can satisfy all these constraints by modifying the parameter vector $\theta$, then we have found parameter values that predict the true sequence-structure as the MFE sequence–structure for $S' \cup S$, which is tantamount to optimizing prediction performance.

In the ICG algorithm, Andronescu *et al.* (2007) apply this idea iteratively to build $S'$, a corresponding set of constraints, and an improved $\theta$. Constrained optimization algorithms can be used to solve these constraints in each iteration generating a new $\theta$ for the next. When $\theta$ converges, it is expected to be an improved parameter value assignment. Note that convergence happens when $\theta$ does not change after an iteration. Important extensions to what we describe include the incorporation of experimental thermodynamic information, and robustness to the lack of a complete solution to generated constraints. It is not always possible to solve all the constraints and enforce that all true structures have MFE, so it is important to slacken this requirement in practice.

### 2.2.2 Our multi-loop parameter optimization algorithm

The ICG algorithm optimizes all parameters simultaneously. Although this is a powerful, holistic approach, we optimize only the multi-loop parameters while keeping the other model parameters fixed to existing values. The reason is that we wanted to compare the multi-loop models fairly. Additionally, we believe it is informative to see how the parameters fit with the existing thermodynamic parameters. The multi-loop models are fit to a limited dataset and small changes in parameter values are known to have large effects on the predicted structures relative to other parameters (Zuber *et al.*, 2017). Optimizing this restricted space has two advantages.

First, the ICG algorithm attempts to find a parameter set that minimizes the free energy of a true structure relative to other possible structures. This is an indicator of the prediction accuracy of a parameter set, but it is sometimes not a good measure. For example, it is possible for the free energy of the true structure to be close to the MFE, and the MFE structure to be dissimilar to the true structure. We preferred to optimize the accuracy directly by maximizing F-score, and found that this yielded more accurate parameter sets, as expected. Comparable approaches have been used before that maximize the probability of the true structures (Andronescu *et al.*, 2010). These require more computation time (6–8 CPU months) than the ICG algorithm (1–3 CPU days) (Andronescu *et al.*, 2010), and the implementation of a partition function version of the new algorithms for the logarithmic and AN models, which do not exist yet. The second advantage of our approach is computation time. For the ICG algorithm, it was reported that solving the systems of constraints required around 80% of the total computation time (Andronescu *et al.*, 2010). Solving constraints is not the bottleneck for our algorithm.

We refer to our algorithm as the *Iterative Brute Force* (IBF) algorithm. It is iterative in the same way that ICG is, but instead of using constraint generation, a form of brute force is used. It modifies the basic template of the ICG algorithm for parameter optimization to only multi-loop parameters, a relatively small set. Andronescu *et al.* (2007) reported that the Turner 1999 nearest neighbor model (Mathews *et al.*, 1999) had 363 free parameters to be optimized. In contrast, the logarithmic model has only five free parameters (Equation 2). In addition, we limit these five parameters to thermodynamically plausible ranges, and choose values rounded to a tenth of a *kcal/mol*, resulting in a total number of combinations that is in the hundreds of millions. Although this is large, it is not astronomical, and it is feasible to try every combination.

This allows us to formulate a parameter optimization algorithm that does not require the generation and solving of constraints. In abstract terms, the IBF algorithm works as follows. We start with an arbitrary parameter vector $\theta$ containing only multi-loop parameters, and a set of known sequence–structure pairs $S$ on which to optimize performance.

Similar to ICG, the IBF algorithm iteratively improves $\theta$. In each iteration, each primary sequence in $S$ is folded subject to $\theta$ to predict a MFE structure. The resulting sequence-structure is added to $S'$, which is the set of folded sequence-structures from all iterations. Next, we update $\theta$. This is done by examining every thermodynamically plausible configuration for $\theta$, and picking the best. We only examine parameter vectors that have variation in multi-loop parameters. In addition, we define the best parameter vector to be the vector whose MFE prediction yields the highest average F-score (see Section 2.3.3 for a definition) in $S \cup S'$. We judge F-score in $S \cup S'$ by evaluating $\Delta G^{\circ}(p,\theta) \forall p \in S \cup S'$, and picking the sequence–structure pair with the minimum $\Delta G^{\circ}$ for each unique primary sequence as the prediction. The average F-score is then the average of the F-scores for each primary sequence.

In our pseudocode for Algorithm 1, three subroutines are used: $fold(x,\theta)$, $\Delta G^{\circ}(p,\theta)$, and $fscore(p,p')$. The $fold(x,\theta)$ function applies a folding algorithm to the sequence in $x$ subject to a parameter vector $\theta$, returning a folded structure. As explained, $\Delta G^{\circ}(p,\theta)$ computes the free energy change of a sequence–structure pair subject to an energy model parameter vector $\theta$, returning the free energy change. The $fscore(p,p')$ function computes the F-score of $p'$ given that $p$ contains the true structure. The *fold* function could implement one of the algorithms of Ward *et al.* (2017). Similarly, the implementation of $\Delta G^{\circ}$ could decompose a structure into its features and score them in $O(n)$ time for a sequence of $n$ nucleotides (Sloma and Mathews, 2016). This can be improved on, however.

**Algorithm 1.** A description of the basic IBF algorithm

```
$\theta \leftarrow$ an arbitrary initial parameter vector
$S \leftarrow$ a set of sequence – structure pairs to optimize on
$S' \leftarrow \{\}$
loop
  $S' \leftarrow S' \cup \{fold(x,\theta) : (x,y) \in S\}$
  $next\theta \leftarrow \theta$
  $bestscore \leftarrow -\infty$
  for all $\theta' \in$ candidate parameter vectors do
    $score \leftarrow 0$
    for all $p \in S$ do
      $predictedp \leftarrow p$
      $predictede \leftarrow \Delta G^{\circ}(p,\theta')$
      for all $p' \in S' | p'$ corresponds to $p$ do
        $e \leftarrow \Delta G^{\circ}(p',\theta')$
        if $e \leq predictede$ then
          $predictede \leftarrow e$
          $predictedp \leftarrow p'$
        end if
      end for
      $score \leftarrow score + fscore(p, predictedp)$
    end for
    $score \leftarrow \frac{score}{|S|}$
    if $score > bestscore$ then
      $bestscore \leftarrow score$
      $next\theta \leftarrow \theta'$
    end if
  end for
  if $next\theta = \theta$ then return $\theta$
  end if
  $\theta \leftarrow next\theta$
end loop
```

### 2.2.3 Optimizing the IBF algorithm

There is an important optimization to the $\Delta G^{\circ}$ function we can achieve. We only vary the multi-loop parameters, and this allows us to optimize $\Delta G^{\circ}(p,\theta)$ to $O(m)$ where $m$ is the number of multi-loops in a structure. In practice, we found $m$ to be small. The maximum number of multi-loops for any structure in our training set is 9, and the average is 1.2.

This speed-up is achieved largely through pre-computation. Features not related to the multi-loop parameters never change free-energy score. Thus, when we first add a sequence–structure pair $p$ to either $S$ or $S'$, we compute the free energy change of $p$ with no free-energy change contribution, called the *multi-loop-free* $\Delta G^{\circ}$. We also keep account of all multi-loop features for $p$. Luckily, for the models we consider, this is a constant amount of information for each multi-loop. For example, the logarithmic model requires only that we store the number of branches and unpaired nucleotides for each multi-loop. Now, when computing $\Delta G^{\circ}(p,\theta)$, we compute the free-energy change for the saved multi-loop features of $p$ subject to $\theta$, and then sum this with the multi-loop free $\Delta G^{\circ}$. This allows computation of $\Delta G(p,\theta)$ in $O(1)$ per multi-loop.

Another major optimization comes from parallelization. Note that the loop over all candidate parameter vectors is finding the maximum score parameter vector, which can be done efficiently in parallel (Cook *et al.*, 1986). Likewise, folding all the elements of $S$, which is the first operation done in each iteration of the outer-loop, can be done in parallel. In short, each iteration of our algorithm is embarrassingly parallel.

### 2.2.4 Seeding of structures

We found that our parameter optimization algorithm was sensitive to the initial arbitrary choice of $\theta$. To make the training more consistent, we seed $S'$ with random structures by randomly selecting five parameter vectors $(\theta_1, \theta_2, \ldots, \theta_5)$ from the space of all thermodynamically plausible parameter vectors and filling $S'$ with the folding results subject to these vectors. Five was empirically found to be sufficient. This appears to smooth out the optimization space so that the best parameter set is usually found.

### 2.2.5 Separating RNA families

There are families of related RNA such as tRNAs or 5S ribosomal RNAs. RNAs within a family have similar structures because they are orthologs. This can be a problem during training. Consider a training set containing 100 tRNAs, and 10 5S rRNAs. If we naively run a parameter optimization algorithm, it will over-fit to tRNA-like structures, since there are more of these in the training set. To deal with this effect, we modify our IBF algorithm to take the final score of a parameter vector to be the average of family averages.

## 2.3 Experiments

### 2.3.1 Folding algorithms and source code

The algorithms for folding RNAs under the multi-loop models are described by Ward *et al.* (2017). Implementations of these algorithms, the source code for the IBF algorithm, and all our datasets are available at https://github.com/maxhwardg/advanced_multiloops.

### 2.3.2 Parameter training evaluation

We used the IBF algorithm for parameter optimization on the linear, logarithmic, the AN models. The algorithm was implemented using the C++11 standard using RNA structure 5.8 (Reuter and Mathews, 2010) for free energy change functions. We divided the dataset in some subsets for training and validation.

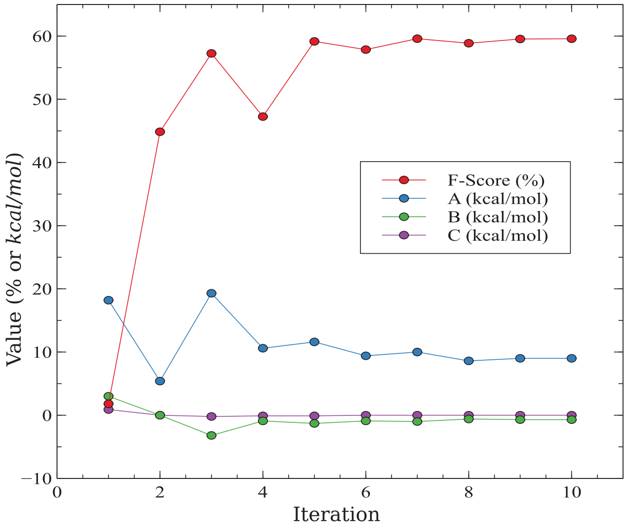

**Fig. 2.** The IBF algorithm converges to good parameters. This plots the progress of the IBF algorithm training to the linear model parameters using the complete dataset for scoring. The x-axis is the number of iterations. The red line indicates the F-score as a percentage. The values for the best parameters (comprising of *A*, *B* and *C* from Supplementary Table S1) are also shown. The blue line shows *A* in *kcal/mol*. The green line shows *B* in *kcal/mol*. The purple line shows *C* in *kcal/mol*

Each model was trained using the IBF algorithm until convergence. We used a thermodynamically reasonable range of parameters for each model, summarized in Supplementary Tables S1–S3. We include in the Results the final trained parameters and their performance on the validation dataset.

Figure 2 shows the progress of the IBF algorithm on an example case. Converge of parameters can be observed.

### 2.3.3 F-score

We use F-score as the measure of the quality of RNA secondary structure prediction. F-score is the harmonic mean of sensitivity (also called recall) and positive predictive value (PPV; also called precision). Sensitivity, $s$, is the fraction of known pairs correctly predicted and PPV, $p$, is the fraction of predicted pair in the known structure.git F-score is $\frac{2sp}{s+p}$. This is a widely used measure of performance (Lorenz *et al.*, 2011; Rivas, 2013). For a predicted pair to be considered correct, we required an exact match in position of the 5' and 3' nucleotides in the pair. Statistical significance was determined using a paired $t$-test as implemented by the SciPy library.

### 2.3.4 Dataset

We required a training set of known sequence–structure pairs. For this purpose, we started with the 'ArchiveII' database available at http://rna.urmc.rochester.edu/pub/archiveII.tar.gz (Sloma and Mathews, 2016). We modified this database by replacing the set of tRNAs with those from RNAstralign (https://rna.urmc.rochester.edu/pub/RNAStralign.tar.gz) (Tan *et al.*, 2017), which include helices in the variable loop. Duplicate primary sequences and sequence–structure pairs with missing structural information, were removed from the resulting dataset. The final dataset is available at https://github.com/maxhwardg/advanced_multiloops, and comprises 9821 RNAs. It includes tRNAs (Jühling *et al.*, 2009), Signal Recognition Particle (SRP) RNAs (Gorodkin *et al.*, 2001), telomerase RNAs (Griffiths-Jones *et al.*, 2005), 5S rRNAs (Szymanski *et al.*, 2000), 16 s rRNAs, 23 s rRNAs (Andronescu *et al.*, 2008; Cannone *et al.*, 2002), tmRNAs (Zwieb *et al.*, 2003), Group I (Andronescu *et al.*, 2008; Cannone *et al.*, 2002) and II Introns (Michel *et al.*, 1989) and RNase P RNAs (Brown, 1999).

### 2.3.5 Training and validation data

For training, we used two datasets. We call these the *large* training set, and the *small* training set. The large training set comprises tRNAs, SRP RNAs, Telomerase RNAs, 5S rRNAs, 16S rRNAs and 23S rRNAs. The large training set contains 2100 RNA sequence–structure pairs. The small training set contained tRNAs, SRP RNAs, and 5S rRNAs whose length was no >300 nt. The small training set contains 1949 sequence–structure pairs. We have both a large and small dataset for training because the AN model folding algorithm is slower [with complexity of $O(n^5)$ for n nucleotides (Ward *et al.*, 2017)], and we had to avoid longer RNA sequences when training it. The large training set was used to train the linear and logarithmic models, whose folding algorithm time complexities are $O(n^3)$ and $O(n^4)$, respectively (Ward *et al.*, 2017).

For validation, we use a similar division of our data. We had a *complete set* that includes all RNAs in our dataset. RNase P RNA, Groups I and II introns, and tmRNAs are in the complete validation set but not in the large training set. We also had a *small set* for validation that includes only RNAs whose length was no >300 nt. 16S and 23S rRNAs, RNase P RNA, Group I Introns and tmRNAs are in the small validation set but not in the small training set.

The complete set was use for validation of the linear and logarithmic models. The small set was used for validation of the AN model. To ensure we can determine the validity of our results, we left out families from training so they can be used for comparison. We opted to do this rather than training on all families and excluding RNAs from each family as a validation set. RNAs from the same family have similar structures, and we aim to avoid the issues this might cause with validation (Lu *et al.*, 2009; Rivas *et al.*, 2011).

## 3 Results

### 3.1 Linear regression

Linear regression was used on a set of 89 available optical melting measurements (Diamond *et al.*, 2001; Hill and Schroeder, 2017; Liu *et al.*, 2011; Mathews and Turner, 2002) to fit multi-branch parameters for the linear model (Zuker and Sankoff, 1984), the logarithmic model (Jacobson and Stockmayer, 1950) and the AN model (Aalberts and Nandagopal, 2010). In total 74 of the measurements were used to derive 2004 nearest neighbor model (Mathews *et al.*, 2004) and 14 are more recent. For the AN model (Equation 3), *a* and *b* were not fit because they are distances measured in crystal structures. The offset (C) value accommodates how close the loop ends need to be for the loop to close. It is the mean of the residuals, which were calculated as the difference between the experimental value and the predicted value using the AN model.

### 3.2 Found parameters and folding accuracy

The parameters we found using the IBF algorithm and linear regression are summarized in Table 1. These parameters were used to fold RNAs to test accuracy. The accuracy results are summarized in Tables 2 and 3 and Figure 3. Since the existing linear model parameters (Equation 1) are the benchmark that other models should improve upon, these tables compare models and parameters to them. We shall also make this comparison when explaining the results.

The parameters for linear regression to the linear and logarithmic models were tested using the complete dataset. These can be found in Table 2. The fit parameters for the linear model appear to be worse than the existing parameters with statistical significance ($P < 0.05$) for three families (23S and 5S rRNAs, and tRNAs), but significantly better for two (RNase P RNAs and tmRNAs). The linear regression parameters for the logarithmic model, however, were not significantly better for any family, and significantly worse for four families (16S and 23S rRNAs, RNase P RNAs and tmRNAs). The existing literature parameters for the logarithmic model (Equation 2) were also tested. These were significantly better for two families (5S rRNAs and SRP RNAs), but significantly worse for three (RNase P RNAs, tRNAs and tmRNAs). Our new fit parameters for the linear model, the fit parameters for the logarithmic model, and the existing parameters for the logarithmic model appear worse than the existing parameters for the linear model.

**Table 1.** The parameter sets for multi-loop models including the linear model (Equation 1), the logarithmic model (Equation 2) and the Aalberts & Nandagopal model (Equation 3)

| | Existing | Regression | IBF |
|---|---|---|---|
| Linear model | | | |
| Initiation | 9.3 | 12.3 ± 1.1 | 9.4 |
| Branch | −0.6 | −0.9 ± 0.3 | −0.9 |
| Unpaired | 0.0 | −0.1 ± 0.1 | 0.0 |
| Logarithmic model | | | |
| Initiation | 10.1 | 12.3 ± 1.1 | 13.0 |
| Branch cost | −0.3 | −0.9 ± 0.3 | −0.6 |
| Unpaired cost | −0.3 | −0.2 ± 0.1 | −0.7 |
| Logarithm coefficient | 1.1 | 3.5 ± 2.7 | 1.8 |
| Pivot to logarithmic | 6 *nt* | 8 *nt* | 7 *nt* |
| AN model | | | |
| *a*-length | 6.2 Å | 6.2 Å | 2.9 Å |
| *b*-length | 15 Å | 15 Å | 15 Å |
| Offset (*C*) | 0 | 2.28 ± 1.38 | 1.6 |

*Note*: Comprises existing parameters from the literature (Aalberts and Nandagopal, 2010; Mathews and Turner, 2002; Mathews *et al.*, 1999), he parameters we found using linear regression, and parameters we found using IBF. All values are in *kcal/mol* unless otherwise specified. The *a*- and *b*-length parameters were not fit using linear regression

The IBF parameters for the linear and logarithmic models were trained using tRNAs, SRP RNAs, telomerase RNAs, 5S rRNAs, 16S rRNAs and 23S rRNAs. Therefore we focus on RNase P RNAs, Groups I and II Introns and tmRNAs in comparing parameters found using IBF. The linear model parameters found using IBF were significantly better ($P < 0.05$) for RNase P RNAs. Interestingly, some families used for training were a little worse with significance (5S rRNAs, telomerase and SRP RNAs); this appears to be balanced by a large significant improvement to tRNAs. The IBF parameters for the logarithmic model were not significantly better for any family they were not trained on, and they are worse with statistical significance for RNase P RNAs and tmRNAs. For families they were trained on, again there is a large significant improvement in the performance of tRNAs. Overall, it appears that the IBF parameters for the linear model are comparable to the existing linear model, and the IBF parameters for the logarithmic model are worse.

A smaller dataset was used for the AN model. The folding results using this dataset can be found in Table 3. The existing parameters for the AN model (Equation 3) were significantly ($P < 0.05$) worse for RNase P RNAs, SRP RNAs and tRNAs, and were not better for any family. The parameters found using linear regression were significantly worse for RNase P RNAs, Group I Introns and tRNAs, but were significantly better for SRP RNAs. The AN model

**Table 2.** Average F-score of optimized model parameters on the complete validation dataset

| | (16S rRNAs) | (23S rRNAs) | (5S rRNAs) | RNase P RNAs | Grp. I Introns | Grp. II Introns | (SRP RNAs) | (tRNAs) | (Telomerase RNAs) | tmRNAs | Average |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Existing linear | 0.518 | 0.669 | 0.599 | 0.529 | 0.483 | 0.255 | 0.594 | 0.710 | 0.465 | 0.405 | 0.523 |
| IBF linear | 0.521 | 0.675 | *0.587* | **0.542** | 0.477 | 0.267 | *0.571* | **0.742** | *0.451* | 0.409 | 0.524 |
| LR linear | 0.502 | *0.633* | *0.573* | **0.539** | 0.474 | 0.300 | 0.593 | *0.690* | 0.459 | **0.436** | 0.520 |
| Existing logarithmic | 0.515 | 0.667 | **0.618** | *0.502* | 0.479 | 0.248 | **0.602** | *0.688* | 0.463 | *0.387* | 0.517 |
| IBF logarithmic | 0.514 | 0.680 | 0.604 | *0.513* | 0.468 | 0.256 | 0.591 | **0.746** | 0.471 | *0.392* | 0.524 |
| LR logarithmic | *0.481* | *0.592* | 0.595 | *0.483* | 0.465 | 0.249 | 0.595 | 0.713 | 0.460 | *0.368* | 0.500 |
| Number of RNAs | 88 | 30 | 1283 | 454 | 98 | 11 | 928 | 6430 | 37 | 462 | 9821 |

*Note*: Larger F-scores are better predictions. Statistically significant advantages over the existing linear model are in bold, while significant losses are in *italic*. Results using parameters determined using linear regression (LR), IBF, and the existing parameters are included. RNA families used for training are denoted by parentheses. Averages of family averages are included in the last column. The *t*-test *P*-values can be found in Supplementary Table S4.

**Table 3.** Average F-score of trained model parameters on the small validation dataset

| | 16S rRNAs | 23S rRNAs | (5S rRNAs) | RNase P RNAs | Grp. I Introns | (SRP RNAs) | (tRNAs) | tmRNAs | Average |
|---|---|---|---|---|---|---|---|---|---|
| Existing linear | 0.630 | 0.834 | 0.599 | 0.513 | 0.505 | 0.597 | 0.710 | 0.470 | 0.607 |
| Existing Aalberts & Nandagopal | 0.611 | 0.722 | 0.599 | *0.480* | 0.492 | *0.580* | *0.694* | 0.449 | 0.578 |
| IBF Aalberts & Nandagopal | 0.606 | 0.724 | *0.592* | *0.491* | 0.482 | 0.596 | *0.705* | 0.467 | 0.583 |
| LR Aalberts & Nandagopal | 0.623 | 0.667 | 0.602 | *0.464* | *0.463* | **0.608** | *0.626* | 0.444 | 0.562 |
| Number of RNAs | 29 | 5 | 1283 | 111 | 21 | 767 | 6430 | 10 | 8656 |

*Note*: Larger F-scores are better predictions. Statistically significant advantages over the existing linear model are in bold, while significant losses are in *italic*. Results using parameters determined using linear regression (LR), IBF, and the existing parameters are included. RNA families used for training are denoted by parentheses. Averages of family averages are included in the last column. The *t*-test *P*-values can be found in Supplementary Table S5.

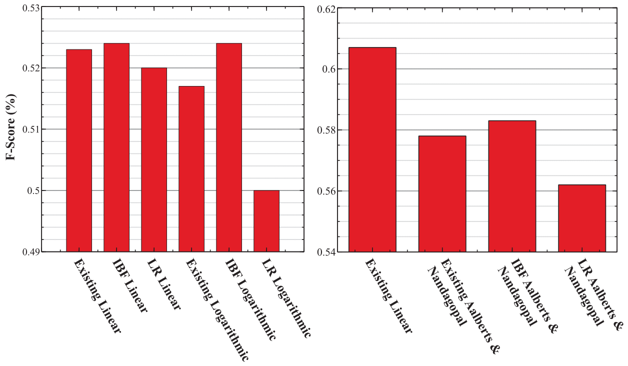

**Fig. 3.** The average F-score percentages of parameter sets for the linear, logarithmic and Aalberts & Nandagopal models. The left panel depicts averages using the complete validation dataset. The right panel depicts averages using the small validation set. The average is an average of family averages. More data can be found in Tables 2 and 3

parameters were trained using IBF on SRPs, tRNAs and 5S rRNAs. They were not better for any family, and were actually significantly worse for some families they were trained on (tRNAs and 5S rRNAs). Overall, it appears as though no parameter set for the AN model is better than the linear model using existing parameters.

## 4 Discussion

We found that none of the parameters we tried were clearly better than the existing parameters for the linear model. In fact, many of the parameters we found are noticeably worse, the AN model parameter we found in particular. We believe that this constitutes evidence that the logarithmic and AN models are not better than the linear model. This implies that they should not be used to calculate free energy change. This additionally means that the models based in polymer theory do not outperform a heuristic. One possible reason for this is that polymer models are designed to account for the entropy of loop closure, but multi-branch loops demonstrate substantial enthalpies for loop closure (Lu *et al.*, 2006) that complicate the estimation of loop free energy change.

Our results echo our previous findings. In 2017, we tested these models using their existing parameters, and also applied rudimentary parameter optimization to the same models (Ward *et al.*, 2017). Though we have used much more comprehensive methods for finding parameters, our results are similar. This reinforces the conclusion that the existing linear model is the best available for secondary structure prediction.

Interestingly, the parameters we found using IBF for the linear model were similar to the existing parameters. In addition, they are almost identical to the parameters found previously by optimization (Mathews *et al.*, 2004; Mathews and Turner, 2002). We believe that this may be evidence that the current linear model parameters are nearly optimal, at least for the current full nearest neighbor parameter model.

The linear model parameters found using linear regression did not outperform existing linear model parameters. There are several possible explanations. One possible reason is that there are not enough optical melting experiments to derive the nearest neighbor parameters and the linear model is over-fitted. The available dataset of multi-branch loop optical melting experiments contains only three- and four-way branching multi-loops. Additionally, most loops are closed with the same helices. This dataset does not represent the large space of possible multi-loop structures. Furthermore, a multi-loop that was designed to closely mimic the sequence of the 5S rRNA multi-loop was substantially more stable than other studied sequences (Diamond *et al.*, 2001), suggesting that the features that stabilize native multi-branch loops are specific and not well modeled by the simple, sequence-independent functions we currently use. Another important reason is possible non-nearest neighbor phenomena. The linear regression model does not provide any information for non-nearest neighbor phenomena.

## 5 Conclusion

We hypothesize that there do not exist parameters for the linear, logarithmic or AN models that are notably superior to the existing linear model parameters when used with the current parameters assigned to the rest of the nearest neighbor model. As can be seen by our results, the prediction performance of folding algorithms is influenced greatly by the scoring of multi-loops. Thus, we propose that a significant increase in folding algorithm accuracy may come from a new model of multi-loop free energy change. We put this forward as the next step in improving our current thermodynamic model of RNA secondary structure. One possible route to improving the multi-loop model is to build from the model of Mathews and Turner (2002), which penalizes the average asymmetry of branches in a multi-loop. This might be a better model of RNA free energy change, and should be investigated as we have investigated the logarithmic and AN models by implementing and testing folding algorithms.

## Funding

This work was supported by National Institutes of Health [R01GM076485 to H.S. and D.H.M.]

*Conflict of Interest*: none declared.

## References

Aalberts,D.P. and Nandagopal,N. (2010) A two-length-scale polymer theory for RNA loop free energies and helix stacking. *RNA*, **16**, 1350–1355.

Akaike,H. (1998) *Information Theory and an Extension of the Maximum Likelihood Principle*. Springer, New York.

Amaral,P.P. *et al*. (2008) The eukaryotic genome as an RNA machine. *Science*, **319**, 1787–1789.

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.

Andronescu,M. *et al*. (2014) The determination of RNA folding nearest neighbor parameters. *Methods Mol. Biol*., **1097**, 45–70.

Bernhardt,H.S. (2012) The RNA world hypothesis: the worst theory of the early evolution of life (except for all the others). *Biol. Direct*, **7**, 23.

Brown,J.W. (1999) The ribonuclease p database. *Nucleic Acids Res*., **27**, 314.

Cannone,J.J. *et al*. (2002) The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, **3**, 2.

Chambers,J. *et al*. (1990) *Statistical Models in S*. Physica-Verlag.

Cohen,J. *et al*. (2013) *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge.

Cook,S. *et al*. (1986) Upper and lower time bounds for parallel random access machines without simultaneous writes. *SIAM J. Comput*., **15**, 87–97.

Deigan,K.E. *et al*. (2009) Accurate SHAPE-directed RNA structure determination. *Proc. Natl. Acad. Sci. USA*, **106**, 97–102.

Diamond,J.M. *et al*. (2001) Thermodynamics of three-way multibranch loops in RNA. *Biochemistry*, **40**, 6971–6981.

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

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

Efroymson,M.A. (1960) Multiple regression analysis. In: Rolston, and Wilf,H.S. (eds) *Mathematical Methods for Digital Computers*, New York, John Wiley.

Gilbert,W. (1986) Origin of life: the RNA world. *Nature*, **319**, 618.

Gorodkin,J. *et al*. (2001) SRPDB (signal recognition particle database). *Nucleic Acids Res*., **29**, 169–170.

Griffiths-Jones,S. *et al*. (2005) Rfam: annotating non-coding RNAs in complete genomes. *Nucleic Acids Res*., **33**, D121–D124.

He,L. and Hannon,G.J. (2004) MicroRNAs: small RNAs with a big role in gene regulation. *Nat. Rev. Genet*., **5**, 522.

Hill,A.C. and Schroeder,S.J. (2017) Thermodynamic stabilities of three-way junction nanomotifs in prohead RNA. *RNA*, **23**, 521–529.

Ihaka,R. and Gentleman,R. (1996) R: a language for data analysis and graphics. *J. Comput. Graph. Stat*., **5**, 299–314.

Jacobson,H. and Stockmayer,W.H. (1950) Intramolecular reaction in polycondensations. i. the theory of linear systems. *J. Chem. Phys*., **18**, 1600–1606.

Jaeger,J.A. *et al*. (1989) Improved predictions of secondary structures for RNA. *Proc. Natl. Acad. Sci. USA*, **86**, 7706–7710.

Jeffreys,H. (1998). *Theory of Probability*. Oxford University Press.

Jühling,F. *et al*. (2009) tRNAdb 2009: compilation of tRNA sequences and tRNA genes. *Nucleic Acids Res*., **37**, D159–D162.

Kiss-László,Z. *et al*. (1996) Site-specific ribose methylation of preribosomal RNA: a novel function for small nucleolar RNAs. *Cell*, **85**, 1077–1088.

Liu,B. *et al*. (2011) Fluorescence competition and optical melting measurements of RNA three-way multibranch loops provide a revised model for thermodynamic parameters. *Biochemistry*, **50**, 640–653.

Lorenz,R. *et al*. (2011) ViennaRNA package 2.0. *Algorithms Mol. Biol*., **6**, 26.

Lu,Z.J. *et al*. (2006) A set of nearest neighbor parameters for predicting the enthalpy change of RNA secondary structure formation. *Nucleic Acids Res*., **34**, 4912–4924.

Lu,Z.J. *et al*. (2009) Improved RNA secondary structure prediction by maximizing expected pair accuracy. *RNA*, **15**, 1805–1813.

Mathews,D.H. (2006) Revolutions in RNA secondary structure prediction. *J. Mol. Biol*., **359**, 526–532.

Mathews,D.H. and Turner,D.H. (2002) Experimentally derived nearest-neighbor parameters for the stability of RNA three-and four-way multibranch loops. *Biochemistry*, **41**, 869–880.

Mathews,D.H. *et al*. (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol*., **288**, 911–940.

Mathews,D.H. *et al*. (2004) Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proc. Natl. Acad. Sci. USA*, **101**, 7287–7292.

Meister,G. and Tuschl,T. (2004) Mechanisms of gene silencing by double-stranded RNA. *Nature*, **431**, 343–349.

Michel,F. *et al*. (1989) Comparative and functional anatomy of group ii catalytic introns me>Gunter<. *Gene*, **82**, 5–30.

Neidle,S. (2010) *Principles of Nucleic Acid Structure*. Academic Press.

Pace,N.R. *et al*. (1999) *The RNA World*, 2nd edn. Cold Spring Harbor Laboratory Press.

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

Rivas,E. (2013) The four ingredients of single-sequence RNA secondary structure prediction. a unifying perspective. *RNA Biol*., **10**, 1185–1196.

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

Rivas,E. *et al*. (2011) A range of complex probabilistic models for RNA secondary structure prediction that include the nearest-neighbor model and more. *RNA*, **18**, 193–212.

Rivas,E. *et al*. (2012) A range of complex probabilistic models for RNA secondary structure prediction that includes the nearest-neighbor model and more. *RNA*, **18**, 193–212.

Rivas,E. *et al*. (2017) A statistical test for conserved RNA structure shows lack of evidence for structure in lncRNAs. *Nat. Methods*, **14**, 45.

Salser,W. (1978) Globin mRNA sequences: analysis of base pairing and evolutionary implications. In: *Cold Spring Harbor Symposia on Quantitative Biology*, Vol. 42. Cold Spring Harbor Laboratory Press, pp. 985–1002.

Sloma,M.F. and Mathews,D.H. (2016) Exact calculation of loop formation probability identifies folding motifs in RNA secondary structures. *RNA*, **22**, 1808–1818.

Szymanski,M. *et al.* (2000) 5S ribosomal RNA database Y2K. *Nucleic Acids Res.*, **28**, 166–167.

Tan,Z. *et al.* (2017) TurboFold II: rNA structural alignment and secondary structure prediction informed by multiple homologs. *Nucleic Acids Res.*, **45**, 11570–11581.

Tinoco,I. and Bustamante,C. (1999) How RNA folds. *J. Mol. Biol.*, **293**, 271–281.

Tinoco,I. *et al.* (1971) Estimation of secondary structure in ribonucleic acids. *Nature*, **230**, 362–367.

Tinoco,I. *et al.* (1973) Improved estimation of secondary structure in ribonucleic acids. *Nature*, **246**, 40–41.

Turner,D.H. and Mathews,D.H. (2009) Nndb: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, 208–282.

Ward,M. *et al.* (2017) Advanced multi-loop algorithms for RNA secondary structure prediction reveal that the simplest model is best. *Nucleic Acids Res.*, **45**, 8541–8550.

Xia,T. *et al.* (1998) Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry*, **37**, 14719–14735.

Zuber,J. *et al.* (2017) A sensitivity analysis of RNA folding nearest neighbor parameters identifies a subset of free energy parameters with the greatest impact on RNA secondary structure prediction. *Nucleic Acids Res.*, **45**, 6168–6176.

Zuker,M. (2003) Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.*, **31**, 3406–3415.

Zuker,M. and Sankoff,D. (1984) RNA secondary structures and their prediction. *Bull. Math. Biol.*, **46**, 591–621.

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

Zwieb,C. *et al.* (2003) tmRDB (tmRNA database). *Nucleic Acids Res.*, **31**, 446–447.