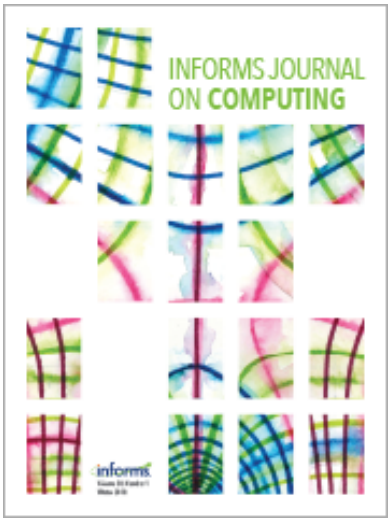


INFORMS Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

ExpertRNA: A New Framework for RNA Secondary Structure Prediction

Menghan Liu, Erik Poppleton, Giulia Pedrielli, Petr Šulc, Dimitri P. Bertsekas

To cite this article:
Menghan Liu, Erik Poppleton, Giulia Pedrielli, Petr Šulc, Dimitri P. Bertsekas (2022) ExpertRNA: A New Framework for RNA Secondary Structure Prediction. INFORMS Journal on Computing 34(5):2464-2484. https://doi.org/10.1287/ijoc.2022.1188

http://pubsonline.informs.org/journal/ijoc

INFORMS JOURNAL ON COMPUTING
Vol. 34, No. 5, September–October 2022, pp. 2464–2484
ISSN 1091-9856 (print), ISSN 1526-5528 (online)

ExpertRNA: A New Framework for RNA Secondary Structure Prediction

Menghan Liu,[a] Erik Poppleton,[b] Giulia Pedrielli,[a,*] Petr Šulc,[b] Dimitri P. Bertsekas[a,c]

[a] School of Computing Informatics and Decision Systems Engineering, Arizona State University, Tempe, Arizona 85281; [b] School of Molecular Sciences and Center for Molecular Design and Biomimetics, Arizona State University, Tempe, Arizona 85281; [c] Massachusetts Institute of Technology, Electrical Engineering, Cambridge, Massachusetts 02139
*Corresponding author

Contact: mliu126@asu.edu (ML); epopplet@asu.edu (EP); gpedriel@asu.edu, https://orcid.org/0000-0001-6726-9790 (GP); psulc@asu.edu (PS); dbertsek@asu.edu (DPB)

Received: September 12, 2020
Revised: March 9, 2021; September 4, 2021; December 22, 2021; January 27, 2022
Accepted: January 29, 2022
Published Online in Articles in Advance: April 19, 2022

https://doi.org/10.1287/ijoc.2022.1188

Abstract. Ribonucleic acid (RNA) is a fundamental biological molecule that is essential to all living organisms, performing a versatile array of cellular tasks. The function of many RNA molecules is strongly related to the structure it adopts. As a result, great effort is being dedicated to the design of efficient algorithms that solve the "folding problem"—given a sequence of nucleotides, return a probable list of base pairs, referred to as the secondary structure prediction. Early algorithms largely rely on finding the structure with minimum free energy. However, the predictions rely on effective simplified free energy models that may not correctly identify the correct structure as the one with the lowest free energy. In light of this, new, data-driven approaches that not only consider free energy, but also use machine learning techniques to learn motifs are also investigated and recently been shown to outperform free energy–based algorithms on several experimental data sets. In this work, we introduce the new ExpertRNA algorithm that provides a modular framework that can easily incorporate an arbitrary number of rewards (free energy or nonparametric/data driven) and secondary structure prediction algorithms. We argue that this capability of ExpertRNA has the potential to balance out different strengths and weaknesses of state-of-the-art folding tools. We test ExpertRNA on several RNA sequence-structure data sets, and we compare the performance of ExpertRNA against a state-of-the-art folding algorithm. We find that ExpertRNA produces, on average, more accurate predictions of nonpseudoknotted secondary structures than the structure prediction algorithm used, thus validating the promise of the approach.

Summary of Contribution: ExpertRNA is a new algorithm inspired by a biological problem. It is applied to solve the problem of secondary structure prediction for RNA molecules given an input sequence. The computational contribution is given by the design of a multibranch, multiexpert rollout algorithm that enables the use of several state-of-the-art approaches as base heuristics and allowing several experts to evaluate partial candidate solutions generated, thus avoiding assuming the reward being optimized by an RNA molecule when folding. Our implementation allows for the effective use of parallel computational resources as well as to control the size of the rollout tree as the algorithm progresses. The problem of RNA secondary structure prediction is of primary importance within the biology field because the molecule structure is strongly related to its functionality. Whereas the contribution of the paper is in the algorithm, the importance of the application makes ExpertRNA a showcase of the relevance of computationally efficient algorithms in supporting scientific discovery.

History: Accepted by Paul Brooks, Area Editor for Applications in Biology, Medicine, & Healthcare.
Funding: The research presented in this paper was partially supported by the National Science Foundation [Grant 2007861] to principal investigator Dr. Pedrielli.

Keywords: computational science • biology • computational methods • dynamic programming • applications • deterministic • industries • pharmaceutical

1. Introduction

Ribonucleic acid (RNA) is a fundamental biological macromolecule that is essential to all living organisms, performing a versatile array of cellular tasks, including information transfer, enzymatic function, sensing, regulation, and structural function (Elliott and Ladomery 2017). RNA has recently emerged as a promising drug target with new therapeutic approaches aiming to develop drugs that target RNA rather than proteins. Moreover, designed RNA molecules are used in the rapidly growing fields of synthetic biology and RNA nanotechnology with applications to diagnostics,

immunotherapy, drug delivery, and realization of logical operations inside cells (Guo 2010, Hochrein et al. 2013, Geary et al. 2014, Green et al. 2014, Han et al. 2017, Qi et al. 2020). In this paper, we propose a new framework, ExpertRNA, for the automatic folding of nonpseudoknotted secondary structures for RNA molecular compounds. ExpertRNA builds upon the fortified rollout algorithm and generalizes the architecture to allow for the consideration of multiple experts that can evaluate, at each iteration, the solutions generated by the base heuristic.

Each RNA molecule is made up of a sequence of individual units, nucleotides (bases), which are of four common types: A, U, G, and C. Individual RNA sequences range in length from tens (tRNAs, siRNAs) to tens of thousands (viral genomes, long noncoding RNAs), and many contain further chemical modifications of the individual bases (Carell et al. 2012). Whereas identity is defined by sequence, the function of an RNA molecule is determined by its structure, that is, the way nucleotides interact in space. Biochemists often break down RNA structure into four categories: Primary structure refers to the sequence. Secondary structure makes up the majority of the bonds in the structure and includes the "canonical base pairs" by which A pairs with U and G pairs with C and the "wobble base pair" by which G pairs with U. This provides a 2-D representation of the structure of the molecule and is the most commonly used level. Tertiary structure defines 3-D contacts via weaker, nonconical interactions. Finally, quaternary structure includes intermolecular interactions with other RNA molecules. Given the impact of structure on RNA functionality, the accurate computational prediction of the secondary and tertiary structure of RNA is an ongoing area of great interest in the computational biology community (Cruz et al. 2012, Calonaci et al. 2020, Wayment-Steele et al. 2020).

1.1. Secondary Structure Prediction

Most tools for secondary structure prediction (Zuker and Stiegler 1981, Hofacker 2003, Reuter and Mathews 2010, Zadeh et al. 2011) attempt to identify the structure that minimizes the free energy (FE) associated with the RNA molecule upon pairing a subset of the nucleotides, that is, the energy released by folding a completely unfolded RNA sequence. The underlying assumption is that the structure with the lowest free energy is also the most likely structure the RNA will adopt. Equivalently, this family of approaches relies on the basic idea that the lower the FE, the more stable the RNA structure. A *first* challenge for this family of approaches is that it is not possible to exactly calculate the free energy because of the (i) incomplete understanding of the RNA molecular interactions and (ii) the impractical computational cost of detailed kinetic simulation tools. As a result, several approximate models are proposed in the literature (Xia et al. 1998; Mathews et al. 1999; Andronescu et al. 2007, 2010) to estimate the free energy associated with a given secondary structure. Most of the computational savings are a result of ignoring tertiary interactions. A *second* and possibly deeper challenge is that this model assumes that an "optimal" structure is one that pairs nucleotides in a way that minimizes the free energy (MFE). However, RNA is known to fold cotranscriptionally (Yu et al. 2021), and equivalently, RNA molecules might adopt a kinetically preferred structure different from the global free energy minima.

In light of these challenges, alternative approaches to structure prediction are proposed. Stochastic kinetic folding algorithms (Isambert and Siggia 2000, Sun et al. 2018) approximate the folding kinetics of RNA molecules as they are transcribed. Data-driven approaches also started to become popular, and they use machine learning to evaluate structures rather than FE or kinetic models. These include ContraFold, DMfold, and structure prediction with neural networks (Do et al. 2006, Wang et al. 2019, Calonaci et al. 2020). Furthermore, in the attempt to achieve advantages of model- or data-driven approaches, methods are proposed that attempt to aggregate multiple information sources to get more accurate secondary structure prediction. Within the data-driven category, some examples of information sources are the experimentally determined SHAPE data (Low and Weeks 2010, Lucks et al. 2011) and evolutionary covariation information (Calonaci et al. 2020). On the model-driven side, statistical ensemble approaches are used to boost the solutions obtained by the different FE-driven folding algorithms. To the knowledge of the authors, ensemble methods allow mixing solutions from different algorithms only upon completion; that is, they do not enable interaction among the algorithms whereas they are running (Aghaeepour and Hoos 2013). A recent survey on a set of secondary structure–prediction tools reports mixed results with data-driven approaches generally outperforming the ones based on nearest neighbor free energy models and with model-based ensemble approaches showing competitive results (Wayment-Steele et al. 2020).

1.2. Tertiary Structure Prediction

Concerning the tertiary structure prediction problem, fewer approaches can be found in the literature (Westhof and Auffinger 2006, Tan and Chen 2010, Seetin and Mathews 2011, Miao et al. 2017, Watkins et al. 2020). In fact, the prediction of tertiary structures is particularly challenging, and most prediction methods only work for short RNA sequences (tens of nucleotides). Data-driven approaches also attract attention for tertiary structure prediction. However, their accuracy

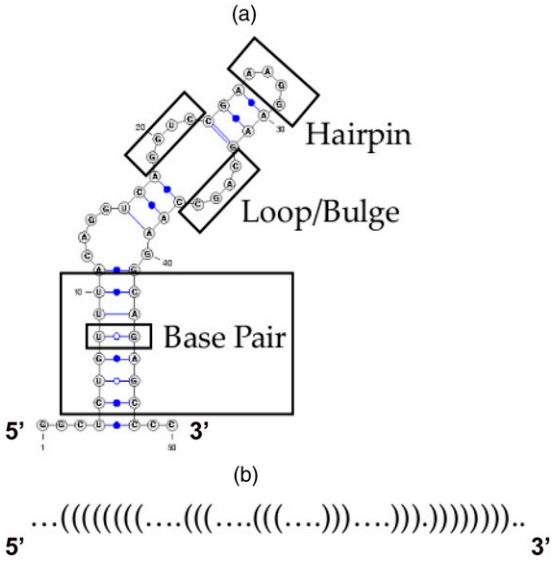
remains limited because of the small number of 3-D RNA structure data sets available for model training and verification.

Stemming from the observation that several folding algorithms are proposed in the literature for secondary and, even if fewer, for tertiary structure prediction without any approach dominating the other, we propose the idea to build a framework that can exploit several folding tools and criteria to evaluate the quality of a folded sequence *during* the algorithm execution. The aim of our approach is to achieve a better RNA structure prediction quality. The result of our work is ExpertRNA, which combines multiple secondary structure prediction algorithms (equivalently referred to as folding algorithms or folders), whose predictions are sequentially evaluated by a set of *experts* that score the quality of the predicted structures based upon their own scoring criteria. ExpertRNA is based on rollout (Bertsekas 2020), an approximate dynamic programming technique, which given an algorithm called the base policy (or base heuristic), generates an improved algorithm, called the rollout policy, as evaluated by one or more experts. Rollout also allows the use of multiple base policies, and it provably improves (relative to the expert score) on the performance of each of them.

We argue that ExpertRNA has the potential to exploit the strengths of the folding tool being used by the algorithm itself as a base heuristic by improving on the solution by using several experts. In this manuscript, we focus on pseudoknot-free secondary structure prediction, and we use RNAfold (a free energy–based folder; Lorenz et al. 2011) as the base heuristic (Section 2), and as the expert, we use two different implementations of the state-of-the-art tool ENTRNA (a classifier that evaluates the quality of a sequence-structure pair; Su et al. 2019) to judge sequence-structure pairs generated by a folding algorithm (Section 2).

We test ExpertRNA against two popular RNA sequence-structure data sets: Rfam (Burge et al. 2013) and the Mathews data set (Ward et al. 2017). We compare the performance of ExpertRNA against RNAfold when used in isolation. We find that, in the sequences taken from Rfam, ExpertRNA produces, on average, more accurate predictions of secondary structure, and performs, on average, the same on the Mathews data set.

Figure 1. (Color online) Example of an RNA Secondary Structure Representation with Highlighted Structural Motifs



Notes. (a) Two-dimensional planar structure representation. (b) Dot–bracket notation. This notation only represents the base pairing and does not provide the type of nucleotide.

2. Relevant Literature

With respect to our work, we distinguish two main branches of research within the rich literature in RNA secondary structure prediction: (i) *model-driven* and (ii) *data-driven* approaches. As mentioned in Section 1, most model-driven approaches use the FE as reference reward function to be minimized. Algorithms differ in the model used to approximate the FE and the mechanisms to explore the space of possible structures in the attempt to find the MFE. Data-driven approaches use machine learning techniques to evaluate the quality of the structure as a replacement or possibly in addition to FE. In this case, FE is interpreted as *a* feature instead of *the* reward.

2.1. Model-Driven Folding

The most common approach for evaluating RNA structures is its free energy, which can be thought of as the energy released by folding a completely unfolded RNA sequence. It can also be interpreted as the amount of energy that must be added in order to unfold a folded molecule. As a result, a sequence is most likely found in a minimum free energy structure. No closed form is available for the exact free energy calculation, and algorithms differ in the way they approximate such computation and the way they search in the space of secondary structures attempting to find the one(s) with associated minimum (approximated) free energy (Lorenz et al. 2011). An example of minimum free energy approximation relies on the nearest neighbor model (NNM) (Xia et al. 1998). The NNM relies on the assumption that each base pair contributes to the overall free energy independently from one another, and a base pair is only influenced by the immediately adjacent base pairs. As a result of these assumptions, the total free energy is the sum of all the base pair free energies along with additional terms that account for the free energy contribution of motifs, such as hairpin loops, internal loops, and

bulges and junctions (Mathews et al. 1999) (Figure 1). Using the free energy minimization criteria, any predicted optimal secondary structure for an RNA molecule depends on the model of folding and the specific folding energies used to calculate that structure. A consequence of this is that different optimal structures may be obtained if the folding energies are changed even slightly.

An example of algorithm relying on FE is RNAfold, a de facto standard tool for RNA secondary structure prediction. The RNAfold software (Lorenz et al. 2011) makes use of the NNM to sequentially assign nucleotides to the partially folded sequence. Importantly, RNAfold allows *constrained folding*, that is, users can input information concerning pairings between nucleotides that they wish to fix and/or just constraining nucleotides to be paired without defining the nucleotide with which to pair it. This second type of constraint is not generally implementable by folding packages and led us to choose RNAfold as the base heuristic for this first version of ExpertRNA. Further examples of model-based RNA-folding approaches are stochastic kinetic folding algorithms. Rather than using dynamic programming, base pairings are randomly sampled, biasing toward bonds with high energy. Energy is estimated using kinetic models, and for computational efficiency, the kinetic model is called only to evaluate the partially folded structure (Isambert and Siggia 2000, Zadeh et al. 2011, Sun et al. 2018). Another approach is AveRNA, an ensemble-based prediction method for RNA secondary structure (Aghaeepour and Hoos 2013). AveRNA combines a set of existing MFE-based secondary structure prediction procedures into an ensemble-based method aiming to achieve higher prediction accuracy. The underlying algorithms use different models for free energy. These models, first presented in Andronescu et al. (2007, 2010), use the constraint-generation method to sequentially produce constraints that enforce known structures to have energies lower than other structures for the same molecule. Such a method results in several FE approximation variants.

It is important to, again, highlight that one of the main drawbacks of model-driven methods is that energy models are approximations of the not fully known complex physical interactions and folding conditions that determine the folded structure of the molecule in natural or laboratory settings. Because of uncertainties in the folding model and the folding energies, the "true" folding may not be the optimal folding determined by the algorithm. In fact, several and suboptimal structures may exist within a few percent of the minimum energy.

2.2. Data-Driven Folding

Approaches utilizing statistical learning have recently received increasing attention. Different folding algorithms may use different statistical methods and different features to generate base pairing decisions. The CONTRAfold algorithm falls into this category of approaches (Do et al. 2006). CONTRAfold applies stochastic context-free grammars for representing RNA structures as random objects that are modified within a fully automated statistical learning procedure that sequentially biases the sampling distribution responsible for the making and breaking of base pairs within the sequentially formed RNA structure. In particular, CONTRAfold sequentially forms and breaks base pairs, updating the probability associated with the several bonding alternatives, which is formulated using conditional log-linear models (CLLMs). The hyperparameters of the CLLMs are estimated using several features of RNA sequence-structure pairs. The model determines the likelihood of a base pairing (or a set of pairings) to be selected by the algorithm (Do et al. 2006). The algorithm CycleFold (Sloma and Mathews 2017) differs from ContraFold because it uses sets of RNA bases as building blocks for the search instead of the individual nucleotides. Each of these sets has a detailed energetic characterization, thus improving the accuracy of the energy prediction patterns. The building blocks are subsequently folded in a way similar to other data-driven approaches (Dieckmann et al. 1996). LearnToFold (Chowdhury et al. 2019) uses an approximate folding tool (LinearFold) as an inference engine for search and a structured support vector machine to bias the sampling toward more favorable pairings. SPOT-RNA (Singh et al. 2019) uses deep contextual learning for base pair prediction, and it is one of the few to include tertiary interactions even if approximated. Deep and transfer learning (Hanson et al. 2020) are trained in order to calculate the likelihood of each nucleotide to be paired with any other nucleotide in a sequence. ENTRNA (Su et al. 2019) is a supervised learning algorithm that uses a support vector machine to produce an evaluation of the quality of a RNA secondary structure. In particular, ENTRNA receives as input a sequence-structure pair, and it returns a measure of the likelihood that such a pair folds, and it is stable. Such likelihood measure is defined by the authors as *foldability*. The higher the foldability, the more likely the sequence-structure pair is to exist and be stable. Alternatively, the more likely that sequence is to fold into that structure. ENTRNA, as any data-driven approach, needs to be trained by examples of existing sequence-structure pairs, each associated with a set of features (sequence-segment entropy (SSE), free energy, relative number of nucleotides of type G and C, relative number of paired nucleotides). In particular, the SSE metric is proposed in (Su et al. 2019) to give a measure of diversity of the nucleotide segments within the RNA sequence. However, to be trained, ENTRNA *also* requires failed examples (i.e., sequence-structure pairs that cannot fold in a stable

manner). This is a challenge because of the unavailability of "failed" experiments, which the authors tackle by using the positive-unlabeled learning method to fill in the failed examples (Li et al. 2009). Nevertheless, this remains a source of inaccuracy for the approach. Similar to ENTRNA, RNAStructure (Reuter and Mathews 2010) uses both data-driven features and free energy models to evaluate sequence-structure pairs. RNAstructure first contained a method to predict the lowest free energy structure; it was subsequently expanded to include bimolecular folding, hybridization thermodynamics, and stochastic sampling of structures. It provides methods for constraining structures based on empirical characteristics of similar sequences. Finally, recent extensions calculate partition functions for secondary structures common to two sequences and can perform stochastic sampling of common structures.

2.3. Contributions

The contribution of this paper is both algorithmic and applied: (i) We extend the rollout approach to be used with multiple heuristics and multiple experts whereas controlling the number of branches active at each iteration of the algorithm, thus allowing control of the computational expense. We allow the different experts to interact, which leads to better solutions than allowing experts to only judge the branches dedicated to them (Bertsekas 2020). (ii) We exploit this algorithmic capability to maintain multiple parallel rollout branches allowing the several experts and, potentially, folding algorithms to maintain multiple solutions. To our knowledge, unlike all the previously published algorithms, ExpertRNA is the only approach allowing the different folding algorithms to interact whereas the structure is being folded. We believe that this capability is the key to overcome the different challenges from the several secondary structure prediction approaches (model- and data-driven).

3. Proposed Approach

The main idea underlying ExpertRNA is to formulate secondary structure prediction (folding) as a sequential assignment problem. Based on the description in Section 2, we assume a sequence of nucleotides is given as input, together with a set of physical constraints that define possible nucleotide-to-nucleotide interactions (base pairing) that can be translated into feasible actions. It is important to formalize the way we model RNA sequences and the related structures before introducing actions and constraints.

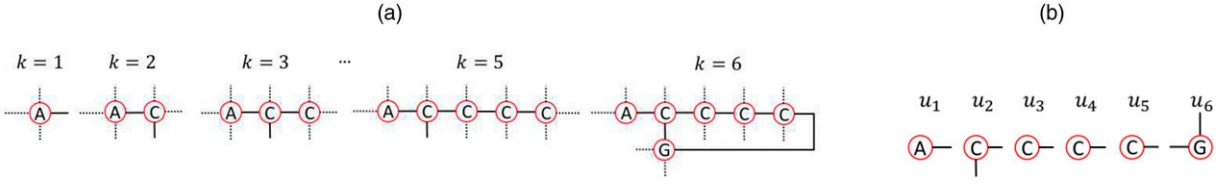
Specifically, our work makes use of the dot–bracket notation to represent RNA structures (Antczak et al. 2018). According to this formalism, each nucleotide is represented by a letter, symbolizing the type of base (A, C, G, U) and a "dot" (.) if the nucleotide is unpaired, that is, it does not establish a bond with any other nucleotide in the sequence, or a "bracket" otherwise. If a nucleotide pairs with another, it can either be the origin of the pairing, in which case we represent it using an open bracket "(", or the sink, in which case we use a close bracket ")" (Figure 1). Figure 1 shows an example of an RNA sequence-structure (Figure 1(a)) and the corresponding dot bracket notation (Figure 1(b)). In the following, we define the main components of ExpertRNA.

3.1. State and Action Modeling in ExpertRNA

As previously mentioned, ExpertRNA sequentially attaches bases to a partially formed structure deciding whether to pair a base (open or closed bracket) or not. Given the same sequence, a different dot–bracket assignment results in a different structure.

More specifically, when we select the next element from the input sequence $x \in \{A, U, C, G\}$, we connect it to the rest of the structure by means of three alternative actions (represented as the top, bottom, and right arcs from each node in Figure 2). The right arc, used in isolation, corresponds to a dot (.) in the dot–bracket notation and establishes a sequential link between bases (i.e., no base pair), whereas the up/down (open/closed) arcs establish a base pair type of connection, thus corresponding to a bracket. The number of links that can be activated at iteration k is a function of the partial structure formed by the k nucleotides already assigned as well as the remaining $N - k$ nucleotides in the sequence that have not been assigned yet. Figure 2 reports an illustrative example of the sequential folding of the RNA molecule. We start selecting a nucleotide (A in the example in Figure 2(a)), and we

Figure 2. (Color online) Example of Assignment



Notes. (a) Example of sequential assignment. At each iteration k, the state is represented by the assigned nucleotides and the corresponding bonds (i.e., sequential or base pair type of connection). (b) Corresponding set of controls.

activate the right link, corresponding to a ".". At the second iteration, we select C, the next nucleotide in the sequence, and we activate the downward link (open bracket "(") and the right link, thus electing C as part of a base pair. Note that, at this point, the sink of the base pair is not established, but we have only decided that C will be part of a base pair as origin. This means that, further in the path generation, we need to elect one of the candidates to be paired with C. The third to fifth actions are identical, and they result in three C-type bases with the right link activated, corresponding to three dots (.). At this point, we select G as the next base with the upward link active (closed bracket ")"), thus electing G as the base pair, and we connect it to the only feasible origin C.

3.1.1. Actions. Generalizing the example in Figure 2, the possible actions that we can implement at each iteration are to sequence, open base pair, and close base pair with the associated dot–bracket notation ".", "(", and ")", respectively. ExpertRNA proceeds to form a structure starting from the unpaired 5′ extreme (the first nucleotide in the sequence, Figure 1) and completed once all the elements are assigned; that is, the 3′ extreme of the sequence has been reached (the "last" nucleotide in the sequence, Figure 1). Also, we assume that bases assigned and links activated at iteration h cannot be changed at any iteration $k > h$. Let us consider a sequence of actions $\{u_1, \ldots, u_k\}$ corresponding to the assignment of the first k nucleotides out of an N-nucleotide-long sequence with $k < N$. Equivalently, $\{u_1, \ldots, u_k\}$ is a partial assignment. Our objective is to sequentially perform sequence-structure assignments using a set of well-defined elementary actions. A structure is complete when $k = N$.

3.1.2. Feasibility Determination. As previously mentioned, at each iteration of the algorithm, we can either execute a (i) "null" action (sequencing) or (ii) base pairing the nucleotide. When evaluating the feasibility of an action, the following items need to be considered:

- Nucleotides of type A can only form a base pair with U
- Nucleotides of type G can only form a base pair with C or U
- Nucleotides of type U only form a base pair with A or G
- Nucleotides of type C only form a base pair with G
- To be paired, i and j need to satisfy $|i - j| \geq 4$, where i and j are the location indexes of the two nucleotides

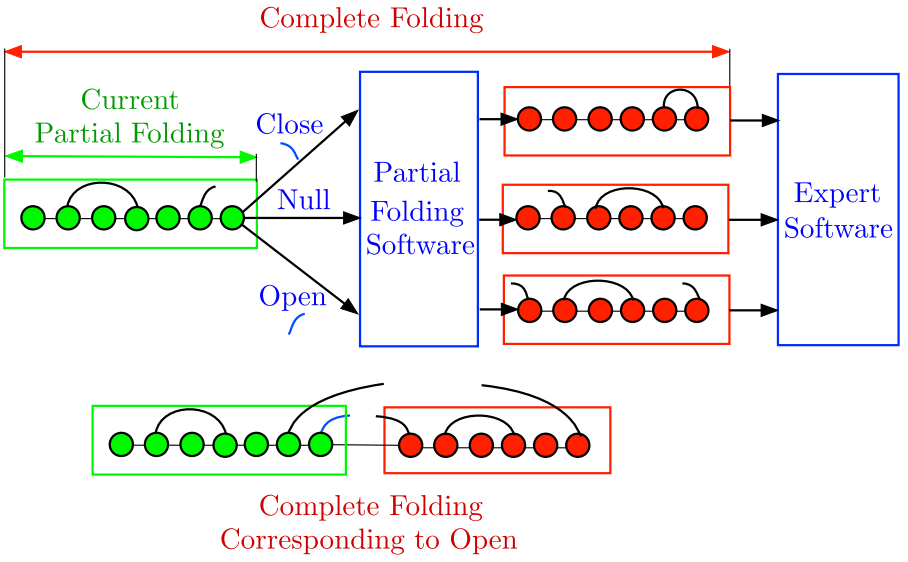
Hence, at each iteration, the set of feasible actions depends on the partial sequence $\tilde{u}_1, \ldots, \tilde{u}_k$ generated by the procedure up to iteration k as well as the nucleotides $k+1, \ldots, N$ that remain to be sequenced.

3.2. ExpertRNA Algorithm

The proposed ExpertRNA tackles the problem of RNA secondary structure prediction as a sequential assignment of (known) nucleotides.

Figure 3 shows the structure of our ExpertRNA approach with its two main algorithmic components: (i) the partial folding and the (ii) expert software. The algorithm sequentially adds elements to the incomplete structure ("current partial folding" in Figure 3), which we initialize to be the empty set. The first

Figure 3. (Color online) Overview of the Proposed ExpertRNA



nucleotide is chosen as the first element of the input sequence provided by the user. At each step, the subsequent nucleotide is selected, and we can choose whether to simply *sequence* it to the last assigned nucleotide ("null" action in Figure 3) or pair it with any nucleotide in the existing structure ("close" in Figure 3) or pair it with an element still to be assigned ("open" in Figure 3). The definition of these actions is motivated by the physical laws that govern molecular bonding (as previously specified in feasibility determination).

3.2.1. Rollout Method. The rollout method was first introduced for the solution of discrete optimization problems in Bertsekas et al. (1997) and Bertsekas and Tsitsiklis (1996). Rollout aims to find a solution of the general problem of minimizing a function $F(u)$ over all $u = (u_1, \ldots, u_N)$, satisfying certain constraints. The components $u_1, \ldots, u_N$ can take a finite number of values, so this is a difficult discrete optimization. Rollout aims to find a suboptimal solution that improves over the solution produced by a given algorithm called the base heuristic (in our case, the base heuristic is RNAfold, but any other folding software can be used). In particular, in our case, the solution is suboptimal with respect to the true reward (i.e., the reward optimized by the actual, unknown structure into which the RNA folds). Such reward is unknown and, therefore, replaced by the expert reward. This is accomplished by a sequence of minimizations of cost functions that are defined by $F(u)$ and by partial solutions produced by the base heuristic.

Our assumption is that, given any partial sequence $(\tilde{u}_1, \ldots, \tilde{u}_{k-1})$, and any feasible value u_k, the base heuristic produces a feasible complete solution of the form

$$(\tilde{u}_1, \ldots, \tilde{u}_{k-1}, u_k, \hat{u}_{k+1}, \ldots, \hat{u}_N), \tag{1}$$

by constructing the complementary sequence $(\hat{u}_{k+1}, \ldots, \hat{u}_N)$ based on the knowledge of $(\tilde{u}_1, \ldots, \tilde{u}_{k-1}, u_k)$. We also assume that the base heuristic can be applied to generate a feasible solution

$$(\hat{u}_1, \ldots, \hat{u}_N), \tag{2}$$

starting from the "empty" partial sequence. Initially, the rollout algorithm uses the base heuristic to generate, for each feasible value u_1, the complementary sequence $(\hat{u}_2, \ldots, \hat{u}_N)$, and computes the initial solution component $\tilde{u}_1$ as

$$\tilde{u}_1 \in \arg\min_{u_1 \in U_1} F(u_1, \hat{u}_2, \ldots, \hat{u}_N).$$

Where U_1 is the set of feasible solutions at the first iteration. Then, sequentially for every iteration $k \geq 2$, given the partial solution $(\tilde{u}_1, \ldots, \tilde{u}_{k-1})$, the rollout algorithm considers all feasible values of u_k and applies the base heuristic to generate the complete solution

$$(\tilde{u}_1, \ldots, \tilde{u}_{k-1}, u_k, \hat{u}_{k+1}, \ldots, \hat{u}_N),$$

cf. Equation (1). It then computes the value of u_k that minimizes the cost function over all these complete solutions:

$$\tilde{u}_k \in \arg\min_{u_k} F(\tilde{u}_1, \ldots, \tilde{u}_{k-1}, u_k, \hat{u}_{k+1}, \ldots, \hat{u}_N),$$

and fixes u_k at the computed value $\tilde{u}_k$. It then repeats with $(\tilde{u}_1, \ldots, \tilde{u}_{k-1})$ replaced by $(\tilde{u}_1, \ldots, \tilde{u}_k)$. After N steps, the rollout algorithm produces the complete sequence

$$\tilde{\boldsymbol{u}} = (\tilde{u}_1, \ldots, \tilde{u}_N),$$

which is called the *rollout solution*. The fundamental result underlying the rollout algorithm is that, under certain assumptions, we have *cost improvement*, that is,

$$F(\tilde{u}_1, \ldots, \tilde{u}_N) \leq F(\hat{u}_1, \ldots, \hat{u}_N), \tag{3}$$

where $(\hat{u}_1, \ldots, \hat{u}_N)$ is the solution produced by the base heuristic starting with the empty partial solution (cf. Equation (3)). Even when the assumptions needed for cost improvement are not satisfied, a simple modification, the so-called *fortified rollout algorithm*, produces a modified sequence that satisfies the cost improvement property in Equation (3).

In addition to the fortified, several other versions of the rollout algorithm are proposed in the literature; we refer to the reinforcement learning textbook by Bertsekas (2019) and the monograph by Bertsekas (2020) for a detailed account, which includes discussions of rollout algorithms that incorporate constraints. Another version that is relevant to this work is *rollout with an expert*, which applies to problems for which we do not know the cost function F of the problem, but instead, we have access to an expert that can rank any two feasible solutions $u^1 = (u_1^1, \ldots, u_N^1)$ and $u^2 = (u_1^2, \ldots, u_N^2)$ by comparing their values $F(u^1)$ and $F(u^2)$ (see Bertsekas 2019, section 2.4.3, and Bertsekas 2020, section 2.3.6). Still another version that is relevant to this work is rollout with multiple heuristics, which allows one to use multiple base policies simultaneously, and also a variant that maintains multiple partial solutions simultaneously and selectively enlarges some of these partial solutions.

3.2.2. ExpertRNA Detailed Description. In this paper, we use several algorithmic variants involving fortified rollout with multiple heuristics and multiple partial solutions. The expert that can rank two solutions is provided by the software package ENTRNA. The base heuristics are provided by the software package RNAfold (Section 2). It is important to note, however, that any expert software and base heuristic software are allowed within our algorithmic framework subject to relatively weak restrictions (see Bertsekas 2019, 2020).

In the current formulation of the rollout algorithm, we require the folder to be able to start from a partially formed structure and, furthermore, impose that during the folding, a selected base has to end up base paired. Currently, only RNAfold allows for such a specific formulation of constraints. In the future version of our algorithm, we will modify the action definition so that it can work with folders that require specifying which base pairs are to be formed.

3.2.2.1. ExpertRNA Main Procedure.

In the following, we provide the detailed iterations of ExpertRNA as a special implementation of a multibranch fortified rollout algorithm.

Step 0. Initialization: Provide the sequence of nucleotides $\mathbf{s} = (s_k), k = 1, 2, \ldots, N$ to be folded; initialize the current partial folding to be empty $\boldsymbol{u}_0 = ()$. Provide a folding software that, given a partial folding, returns a complete structure. Provide an expert software with associated score function $\mathcal{S}^e : S \to [0, 1]$, where S is the set of all feasible complete structures. Initialize the fortified solution $\mathbf{u}_e^* = ()$. Set $k \leftarrow 1$.

Step 1. Move generation: Given the partial solution $(u_1, \ldots, u_k)$, generate all feasible next assignments $u_{k+1}^1, \ldots, u_{k+1}^m$, $m \leq 3$ that satisfy the feasibility conditions (Section 3).

For each feasible move $u_{k+1}^g \in U_k$, $g = 1, \ldots, m$, where U_k is the set of feasible actions at iteration k for the kth nucleotide in the sequence:

Step 2. Structure completion: Use the heuristic H_k to generate the complete solution for the gth feasible action:

$$\mathbf{u}_{k+1}^g = H_k(u_1, \ldots, u_{k-1}, u_k^g) = (u_1, \ldots, u_k, u_{k+1}^g, \ldots, u_N^g),$$

and RNAfold (Section 2; Lorenz et al. 2011) in its constrained version is used such that, considering the current state (i.e., the sequence of performed assignments up to the kth iteration) and the next assignment u_{k+1}^g completes the structure based on minimum free energy (Zuker et al. 1999).

Step 3. Structure evaluation: Score the candidate structures $\{\mathbf{u}_{k+1}^g\}, g = 1, \ldots m$ using the expert scoring system $\mathcal{S}^e(\mathbf{u}_{k+1}^g)$ and save the top solution $\mathbf{u}_{k+1}^{(1)}$, where $\mathbf{u}_{k+1}^{(h)} = \left(u_1, \ldots, u_{k+1}^{(h)}, \ldots, u_N^{(h)}\right)$, and the superscript $(\cdot)$ is to be interpreted as order index.

Step 4. Update the fortified solution: Update the fortified solution for the expert e: if $\mathcal{S}^e\left(\mathbf{u}_{k+1}^{(1)}\right) > \mathcal{S}^e(\mathbf{u}_e^*)$, $\mathbf{u}_e^* \leftarrow \mathbf{u}_{k+1}^{(1)}$, where $\mathbf{u}_{k+1}^{(1)} = \left(u_1, \ldots, u_{k+1}^{(1)}, \ldots, u_N^{(1)}\right)$.
$k \leftarrow k + 1$.

Step 5. Stopping condition: If $k == N$, STOP. Else go to step 1.

3.2.2.2. ExpertRNA Extensions.

A first extension to this basic algorithm we implemented is to maintain a set of B active branches. More specifically, the top-B branches are maintained at each point in time in the algorithm execution. Given that we have a maximum of $B_k \cdot 3$ feasible actions at each iteration k, where B_k is the number of branches being maintained at iteration k, in general $B_k \leq B$. In particular, the set U_k^b of feasible actions from a single branch b satisfies $|U_k^b| \leq 3$. A second extension we implemented, with respect to the basic algorithm with multiple branches, was to allow for multiple experts. ExpertRNA is such that B is set to be a multiple of the number of experts. This choice is justified by the fact that no prior information is available that can help us choosing for which expert we should maintain more branches. Finally, in case multiple experts are adopted, a number of fortified solutions bounded by the number of experts is maintained at each iteration. The fortified solution(s) is updated at iteration k if a structure with higher reward is identified for any of the experts. Then, at each iteration, a solution generated by a feasible action is compared with the fortified solution. If no action achieves a better score, the fortified action is chosen instead.

3.2.2.3. ExpertRNA Move Generation.

A focal aspect of ExpertRNA is the generation of the set of feasible actions at iteration k, U_k (step 1, move generation in the general ExpertRNA algorithm). In the following, we detail the procedure for the generation of the feasible actions.

Step 0. Initialization: Iteration index k, partially folded structure $u_1, \ldots, u_k$, and the unfolded nucleotide sequence $\mathbf{s}_{k+1:N} = (s_l)_{l=k+1}^N, s_l \in \{A, U, C, G\}$. Initialize the set of feasible actions to the empty set $U_{k+1} = \emptyset$.

Step 1. Action generation: A feasible action represents the pair of nucleotides and the modality used to attach the nucleotide to the rest of the sequence. There are three possible actions: (i) open base pair, which we refer to as u_{k+1}^o; (i) close base pair, which we refer to as u_{k+1}^c; (ii) and the null action (i.e., simple sequencing), which we refer to as u_{k+1}^n.

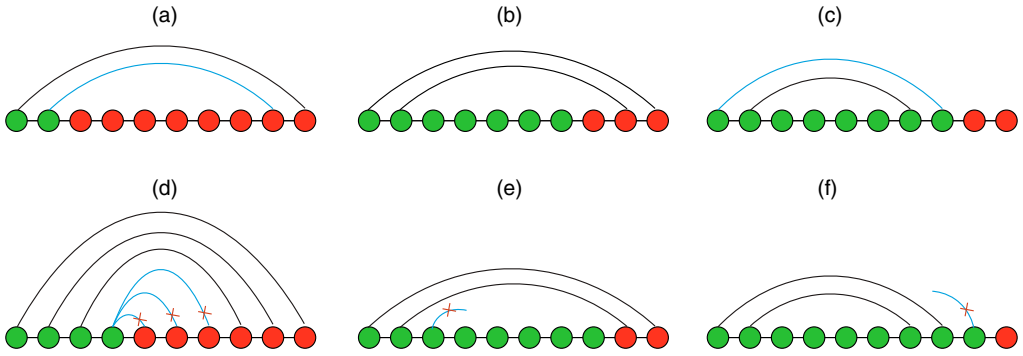
Step 2. Feasibility certification: We verify whether the actions are feasible.

Open base pairing (Figure 4, (a) and (d)). *Condition* 1: Given the incomplete structure $\mathbf{u}_k$, derive $\overline{h}^o$, that is, the number of open base pairs that have not been closed. If the size of the remaining sequence satisfies $N - k > \overline{h}^o + 4 + 1$, then condition 1 is satisfied, and we generate the set of candidate paired bases for the $(k + 1)$ st base.

Condition 2: If, among the nucleotides to pair, there exist at least one nucleotide that can physically pair with u_{k+1}^o, then condition 2 is satisfied.

Condition 3: If all the bases with open brackets $u_h^o, h = 1, \ldots, k$ can close if the $(k + 1)$ st base is not a closed bracket, then condition 3 is satisfied.

Figure 4. (Color online) Examples of Feasible and Infeasible Conditions for Action as u^o_{k+1}, u^c_{k+1}, and u^n_{k+1}



Notes. The light grey circles are the bases within the folded partial chain (the last light grey circle is the position we are currently assigning), and the dark grey part is the unfolded part. The black arcs are implemented pairings, whereas the blue arcs are being tested for feasibility. (a) The action u^o_{k+1} is feasible. (b) Simple sequencing u^n_{k+1} is feasible. (c) Closing a base pair u^c_{k+1} at the position is feasible. (d) Opening a base pair u^o_{k+1} at the position is infeasible because open base pairing condition 1 is violated. (e) Sequencing u^n_k at the position is infeasible because null pairing condition 1 is violated. (f) Closing the base pair u^c_k at the position is infeasible because close base pairing condition 2 is violated.

If all conditions are satisfied $U_{k+1} \leftarrow U_{k+1} \cup u^o_{k+1}$.

Close base pairing (Figure 4, (c) and (f)). Condition 1: Given the incomplete structure $\mathbf{u}_{k+1}$, derive $\overline{h}^o$, that is, the number of open base pairs that have not been closed if $\overline{h}^o$. If $\overline{h}^o > 0$, condition 1 is satisfied.

Condition 2: If there is an open base $u^o_h, h \le k-4$ and the nucleotide is compliant with the current base, condition 2 is satisfied.

If all conditions are satisfied, $U_{k+1} \leftarrow U_{k+1} \cup u^c_{k+1}$.

Null pairing (Figure 4, (b) and (e)). *Condition* 1: Given the incomplete structure $\mathbf{u}_k$, derive $\overline{h}^o$, that is, the number of open base pairs that have not been closed if $\overline{h}^o < N-k-1$; then, condition 1 satisfied.

Condition 2: Calculate the number of nucleotides of the whole chain minus the position of the last closed nucleotide, and denote this number as $\mathbf{r_k}$. If $\mathbf{r_k} + 4 \ge \overline{h}^o$, condition 2 is satisfied.

Condition 3: If the $(k+1)$ th nucleotide is unpaired, check whether the whole chain can be completed complying with feasibility constraints, which means all remaining incomplete open base pairs within the partial chain $\mathbf{u}_k$ can be paired using nucleotides that have not yet been assigned (A, U, C, G pairing constraints considered). If so, condition 3 is satisfied.

If all conditions are satisfied, $U_k \leftarrow U_k \cup u^n_k$.

4. Numerical Results

In this section, we compare the performance of ExpertRNA and RNAfold used in isolation. We highlight that judging the quality of a proposed structure is generally not a trivial problem, and this is the main motivation at the basis of data-driven approaches that allow us to consider rewards different from free energy. In this analysis, we consider the sequence-structure pairs in the data sets as the "ground truth." Under this working assumption, the better method is the one that produces a sequence-structure pair that is more similar to the one in the database. Similarity measures are discussed.

Figure 5. (Color online) Distribution of RNA Length in (a) RNASTRAND, (b) Mathews, and (c) Rfam

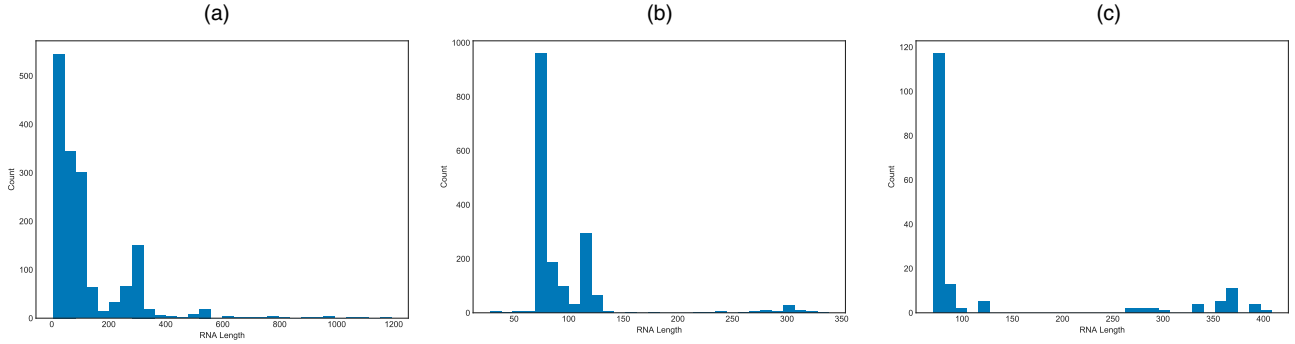


Table 1. Aggregate MCC Results Produced by ExpertRNA for the Rfam Data Set

Branch	$\overline{MCC}$	MCC^{+}	$\overline{\Delta}_{MCC}$	$\sigma(\Delta_{MCC})$	p-value
1	0.1135	0.1552	0.0566	0.1536	0.0579
2	0.1247	0.1751	0.0679	0.1481	0.0243
3	0.1327	0.1659	0.0651	0.1459	0.0267
4	0.1378	0.1637	0.0703	0.1497	0.0196

Table 2. Aggregate Statistics on Prediction Quality by ExpertRNA for the Mathews Data Set

Branch	$\overline{MCC}$	MCC^{+}	$\overline{\Delta}_{MCC}$	$\sigma(\Delta_{MCC})$	p-value
1	0.4274	0.4960	0.0058	0.1891	0.7077
2	0.4182	0.4902	−0.0034	0.1918	0.8211
3	0.4234	0.4950	−0.0050	0.1943	0.7430
4	0.4216	0.4875	−0.0074	0.1951	0.6235

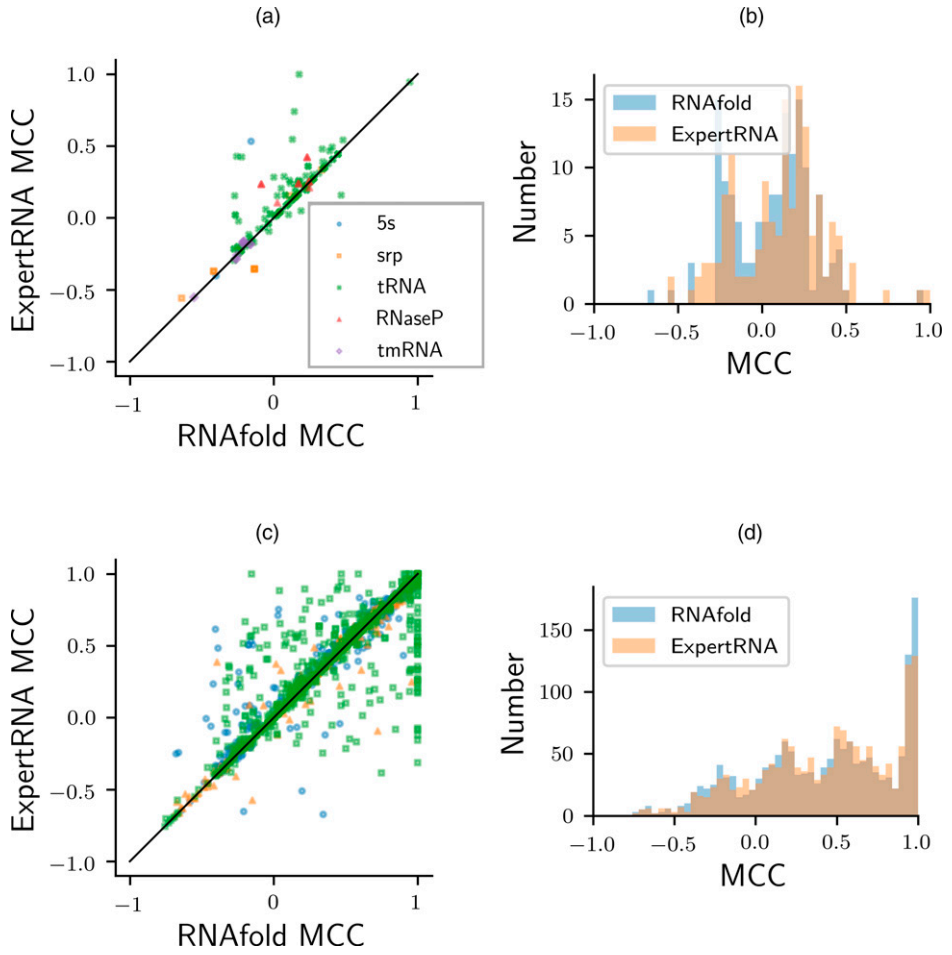
4.1. Implementation and Package Usage Details

Experiments were executed on the Agave HPC structure at Arizona State University (https://cores.research.asu.edu/research-computing/user-guide), equipped with 800 FP64 CPU Teraflops, 498 Compute Nodes, and 128-256GB RAM on most CPU nodes.

The ExpertRNA package that we used for the experiments can be downloaded at https://github.com/MenghanLiu212/RL-RNA. We included the data sets used for testing and training with a detailed explanation available at https://github.com/Menghan Liu212/RL-RNA/blob/master/README.md. ExpertRNA is written in Python 3.7, and the code has three main components:

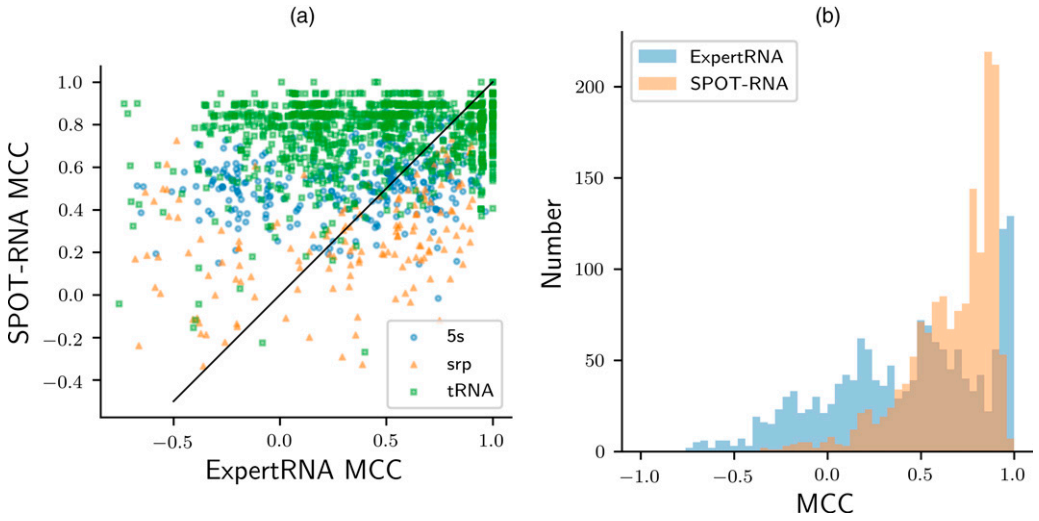
1. `ExpertRNA.py` is the entry point to run the code and generates final rewards for the results.
2. `ExpertRNA_main.py` contains the main algorithm.
3. `ExpertRNA_toolbox.py` includes the feasible action-generation function and RNA structure transforming functions.

Figure 6. (Color online) Performance of ExpertRNA Compared with RNAfold on the Two Data Sets Broken Down by RNA Class



Notes. (a) and (c) MCC scores for the structures generated from sequences in the Rfam database and Mathews database, respectively. (b) and (d) Distribution of MCC scores for the Rfam and Mathews database, respectively. In the scatterplots, the line of equal scores is also plotted (thin black line) as a reference.

Figure 7. (Color online) Performance of ExpertRNA Compared with SPOT-RNA on the Mathews Data Set Broken Down by RNA Class



Notes. (a) MCC scores for the structures generated from sequences in the Mathews database. In the scatterplots, the line of equal scores is also plotted (thin black line) as a reference (b) Distribution of MCC scores for the Mathews database.

In order to run the code, the user needs to download or clone the Github repository and provide the following information:

1. Execution mode: ExpertRNA can be executed by the user in two modalities: test mode (`-t`) and run mode (default). In both cases, the input is a path to a folder containing files with the sequences to fold. There should be one file per sequence. When running in test mode, both the sequences and secondary structures of the target RNAs need to be provided to ExpertRNA as input following the dot–bracket notation and stored in a `.dbn` file (same as a fasta with an additional line containing the structure). If run mode is selected instead, the user only needs to provide sequence information in `.fasta` files.

2. Folding software name and type (`-f`): Currently, the only available options are "RNAfold" and "nonspecific," which instantiate the version of RNAFold that accepts "open constraints"—that is, when a base pair is opened or closed, we do not need to specify the corresponding nucleotide pair involved.

3. Expert name and number of branches for that expert (`-e`): At the moment, we have two Expert options in the current packet: (a) "ENTRNA_MFE," corresponding to the version of ENTRNA with MFE, and (b) "ENTRNA_NFE," corresponding to the version of ENTRNA without MFE. These can also be combined, for example, to have two branches for each expert.

4. Minimum distance requirement for base pairing (`-m`): The default value is four because this is the requirement for RNAFold, which is our folding tool. RNA structures exist with spacing of three; however, we find that RNAFold does not always produce an output when the spacing of three is allowed.

As an example, to execute in testing mode ExpertRNA using RNAfold as a folding tool and ENTRNA_NFE as Expert with four branches, the command from terminal is

```
$python expertRNA.py path_to_input_folder
-e ENTRNA_NFE 4 -f RNAfold nonspecific -m 4 -t.
```

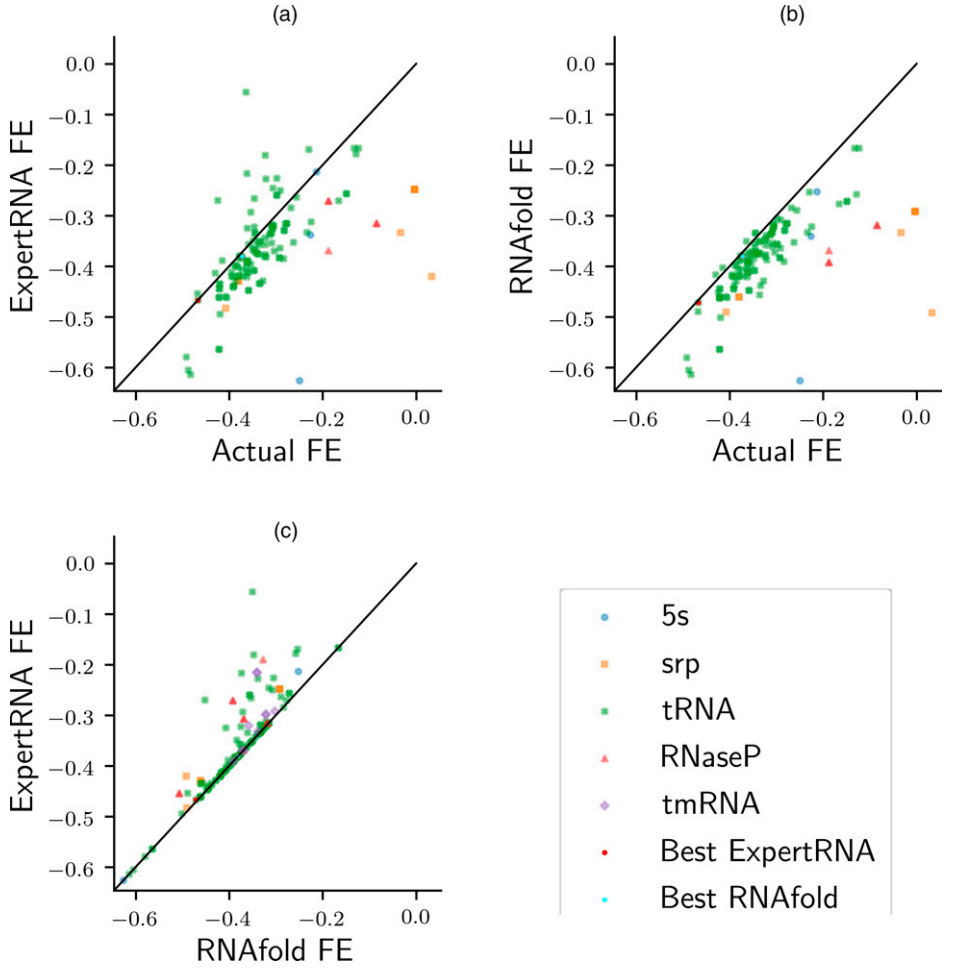
In testing mode, the results are saved as a .csv formatted file containing structure prediction(s) produced by the folding software alone as well as ExertRNA along with metrics that characterize the RNA structure (e.g., GC percentage, BP number). The detailed description of the output can be found within the package repository at https://github.com/MenghanLiu212/RL-RNA. In run mode, the output is a modified `.dbn` file, in which each sequence has a number of potential structures based on the number of branches requested and the score for the structure generated by the expert for each branch.

4.2. Experimental Settings

4.2.1. Data Set for Training. When training ENTRNA, we adopted 1,024 pseudoknot-free RNA molecules from the RNASTRAND database (Andronescu et al. 2008). The length of the sequences within the database ranges from 4 to 1,192 nucleotides (Figure 5(a) shows the distribution of the sequence length).

4.2.2. Data Set for Testing. ExpertRNA was tested on two data sets. The first data set was obtained from the

Figure 8. (Color online) FE per Nucleotide Scores of Structures from the Rfam Data Set as Calculated by RNAfold for the Actual Structure, the Predictions from ExpertRNA, and RNAfold



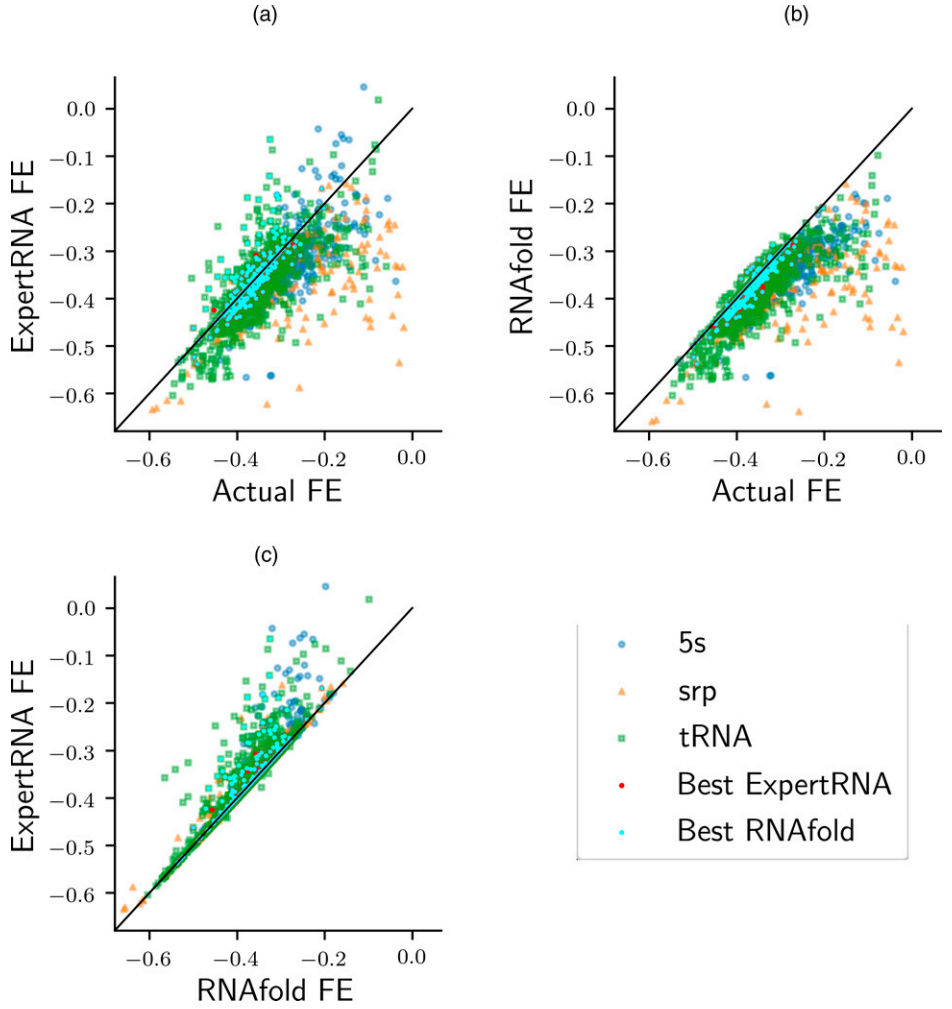
Notes. In each graph, input sequences are broken down by sequence type. Sequences in which ExpertRNA did particularly well (MCC ≥ 0.9) are highlighted with dark grey pips; there were no structures in this data set in which RNAfold performed well. The line of equal scores is also plotted (thin black line) as a reference. (a) ExpertRNA-predicted structures compared with the actual structure. (b) RNAfold-predicted structures compared with the actual structure. (c) ExpertRNA-predicted structures compared with RNAfold-predicted structures.

Rfam/revdatabase (Burge et al. 2013) by randomly selecting a subset of seed structures from distinct ncRNA families. Specifically, we used 147 sequences of length ranging from 71 to 408 nucleotides (Figure 5(c)). The second data set is the benchmark data set of the RNAStructure tool (Reuter and Mathews 2010) as populated by Mathews' laboratory (Ward et al. 2017), which consists of natural RNA sequences with known secondary structures, comprising 1,559 sequences of lengths ranging from 28 to 338 nucleotides (Figure 5(b)). Each data set was culled to remove structures with greater than 80% sequence identity with structures in the ENTRNA training data set using CD-HIT-EST-2D (Huang et al. 2010).

4.2.3. Metrics. As previously mentioned, in this analysis, we consider better the algorithm that produces a structure that is close to the one in the database. More specifically, to evaluate the quality of the produced structures, we look into three indicators:

- FE: FE is a standard metric to evaluate the quality of an RNA structure. RNAfold is an FE minimizer. Hence, the ExpertRNA solution may have higher associated FE. The free energy calculation used here is from the ViennaRNA package and is the same one used by RNAfold (Lorenz et al. 2011).
- Matthews correlation coefficient (MCC): MCC (Matthews 1975) is a popular metric in the RNA structure prediction field used to score the confusion matrix

Figure 9. (Color online) FE of Structures from the Mathews Database as Calculated by RNAfold Calculated for the Actual Structure and the Predictions from ExpertRNA and RNAfold



Notes. In each graph, input sequences are broken down by sequence type. Sequences in which ExpertRNA and RNAfold did particularly well (MCC ≥ 0.9) are highlighted with smaller cyan and dark grey pips, respectively. The line of equal scores is also plotted (thin black line) as a reference. (a) ExpertRNA-predicted structures compared with the actual structure. (b) RNAfold-predicted structures compared with the actual structure. (c) ExpertRNA-predicted structures compared with RNAfold-predicted structures.

of a binary prediction. It takes into account the number of correctly and incorrectly predicted paired/unpaired bases and returns a score between −1 and 1, where −1 is a totally incorrect structure, 0 is the expected value of random assignment, and 1 is a totally correct structure. The formal definition is

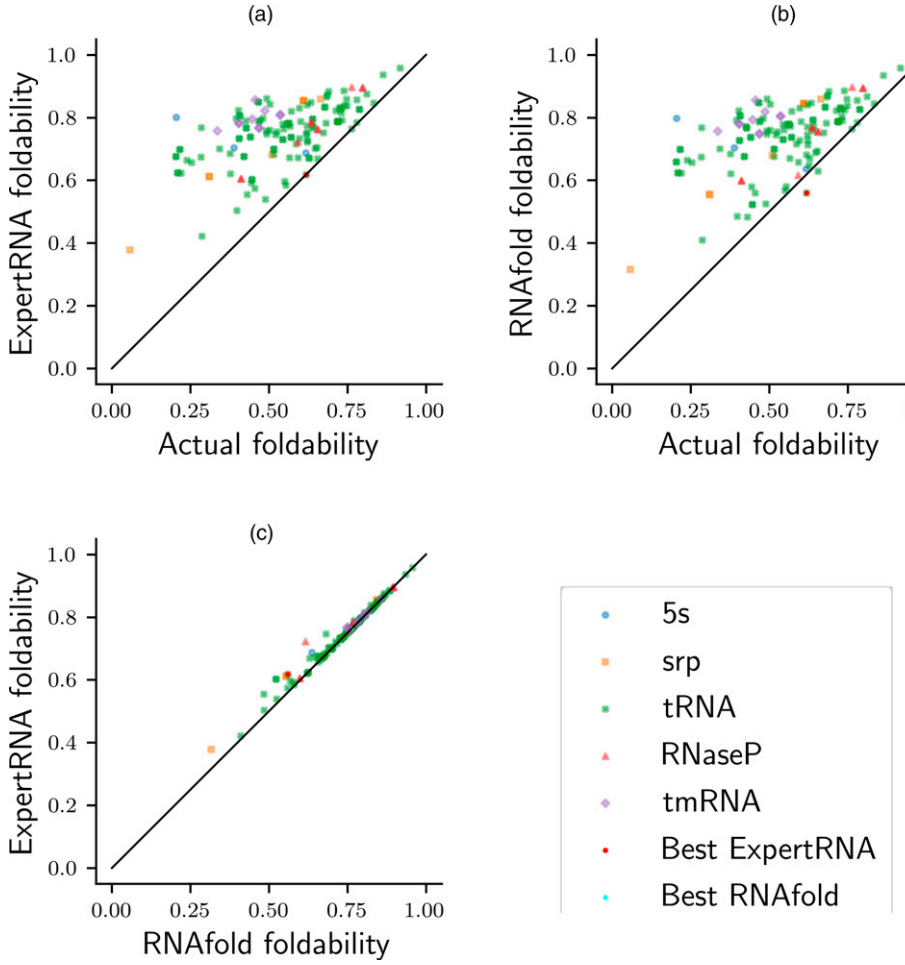
$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP) \cdot (TP + FN) \cdot (TN + FP) \cdot (TN + FN)}}, \tag{4}$$

where TP, TN are the correctly identified paired and unpaired bases, respectively. FP, FN are the incorrectly paired and unpaired bases, respectively.

- Foldability: This metric is a score in the interval [0,1] quantified by ENTRNA. The higher the foldability, the more likely the sequence-structure pair is to fold (Section 2).

4.2.4. ExpertRNA Settings. We created an extension of ENTRNA for this work, namely, the *No Free Energy ENTRNA (ENTRNA-NFE)*. Specifically, we retrained the ENTRNA classifier, removing the free energy from the set of features used to train the support vector machine at the basis of the expert. The rationale behind this modification was to allow us to search for solutions (structures) that do not necessarily have low free energy. ENTRNA-NFE was a better performing

Figure 10. (Color online) Foldability Scores for Structure Predictions from the Rfam Data Set as Calculated by the Expert, ENTRNA-NFE



Notes. In each graph, input sequences are broken down by sequence type. Sequences in which ExpertRNA did particularly well (MCC ≥ 0.9) are highlighted with dark grey pips; there were no structures in this data set in which RNAfold performed well. The line of equal scores is also plotted (thin black line) as a reference. (a) ExpertRNA-predicted structures compared with the actual structure. (b) RNAfold-predicted structures compared with the actual structure. (c) ExpertRNA-predicted structures compared with RNAfold-predicted structures.

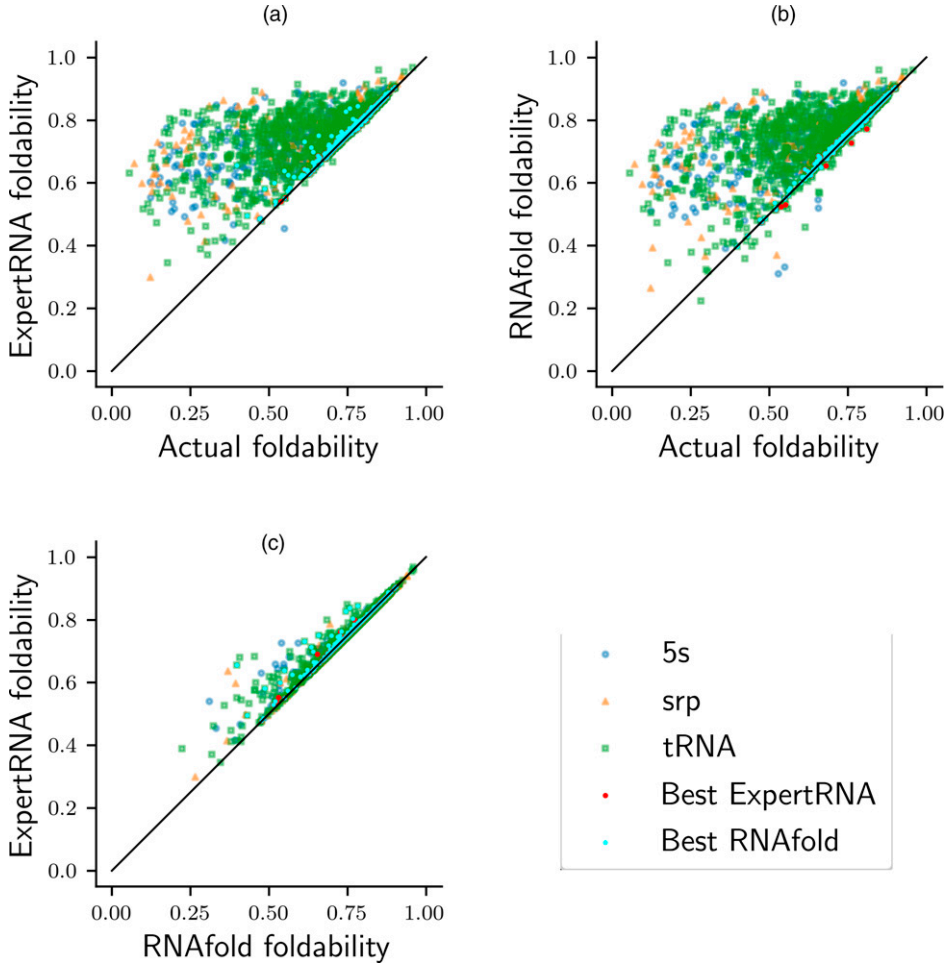
expert than the original ENTRNA as well as the ExpertRNA using *both* ENTRNA and ENTRNA-NFE simultaneously. We ran these several variants of ExpertRNA, maintaining four branches at each iteration. In the case in which we used both experts, we allowed each expert to maintain its two highest scoring branches at each iteration. As a result, at each iteration k of ExpertRNA, we have a number $B = 4$ of active branches, and $B = 4$ structure foldings are performed. To reduce the computational demand, the foldings can be parallelized. All results reported were obtained from ExpertRNA with the ENTRNA-NFE expert. Nonetheless, similar performances were obtained with the ENTRNA expert and when using both experts together.

4.3. Performance Analysis

In the following, we analyze first the MCC as the main discerning metric to individuate the characteristics of the proposed algorithm against the state-of-the-art RNAfold. We then look into free energy and foldability to provide additional insights on the difference between the two approaches.

4.3.1. MCC Analysis. Tables 1 and 2 show the aggregate results in terms of the returned MCC. In particular, for each instance within the database, we ranked the four resulting branches by foldability (best to worst) and report the average and median MCC ($\overline{MCC}, MCC^{\dagger}$) across the data set for each branch: the average improvement in MCC over the RNAfold prediction ($\overline{\Delta}_{\mathrm{MCC}}$), the

Figure 11. (Color online) Foldability Scores for Structure Predictions from the Mathews Data Set as Calculated by the Expert, ENTRNA-NFE



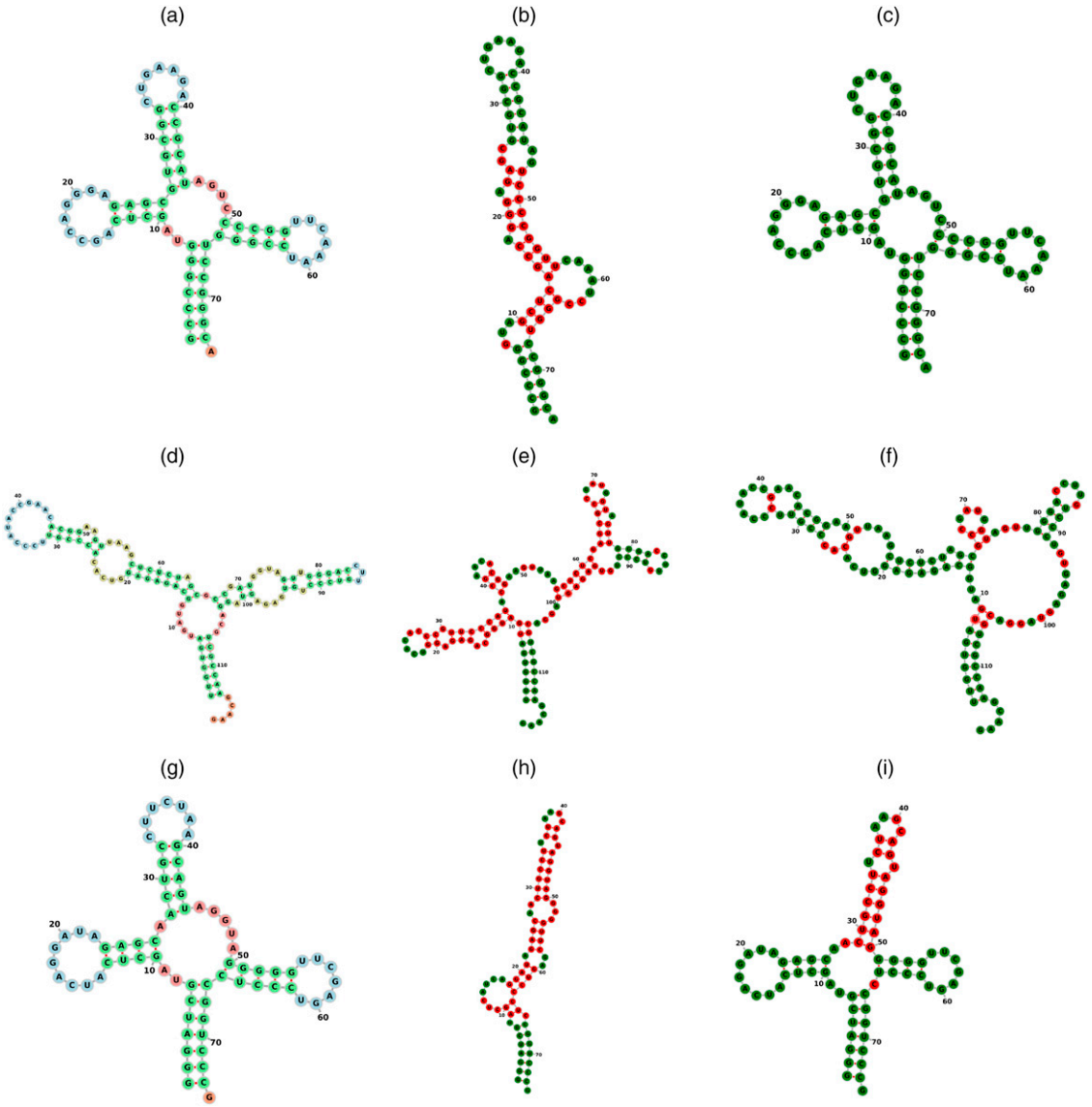
Notes. In each graph, input sequences are broken down by sequence type. Sequences in which ExpertRNA did particularly well (MCC ≥ 0.9) are highlighted with smaller cyan and dark grey pips, respectively. The line of equal scores is also plotted (thin black line) as a reference. (a) ExpertRNA-predicted structures compared with the actual structure. (b) RNAfold-predicted structures compared with the actual structure. (c) ExpertRNA-predicted structures compared with RNAfold-predicted structures.

standard deviation of the improvement ($\sigma(\Delta_{MCC})$), and the p-value associated to the hypothesis $\overline{\Delta}_{MCC} \neq 0$ computed with a two-sided t-test. Here, Δ_{MCC} = MCC (ExpertRNA) – MCC(RNAfold), and we want this to be positive. Two main observations can be drawn from the analysis of the aggregate data. First, the first branch solutions are always showing positive $\overline{\Delta}_{MCC}$; second, it is apparent that ExpertRNA has a better performance over RNAfold in the Rfam data set as compared with the results from the Mathews data set.

Figure 6 shows the disaggregated MCC results. Two interesting observations emerge from the MCC distributions. First, for the Rfam data set (Figure 6, (a) and (b)), RNAfold never correctly identifies the correct structure, whereas ExpertRNA gets the complete or almost complete structure in a couple of cases. In Figure 6(b), for the Rfam data set, the distribution of scores is shifted toward the right for ExpertRNA compared with RNAfold alone with ExpertRNA performing worse in very few cases, indicating a better performance for ExpertRNA. The results are less clear-cut for the Mathews data set (Figure 6, (c) and (d)), with which ExpertRNA does generally better on structures on which RNAfold does very poorly. However, there is also a significant population of structures that RNAfold gets correct or almost correct, whereas ExpertRNA misses. These two features balance each other out, resulting in no substantial improvement for ExpertRNA. In general, both ExpertRNA and RNAfold perform better on the Mathews data set compared with the Rfam data set.

It is important to notice the role played by the Expert software in the performance of the proposed

Figure 12. (Color online) Example Structure from the Rfam Data Set with the Best Performance Improvement Between ExpertRNA and RNAfold



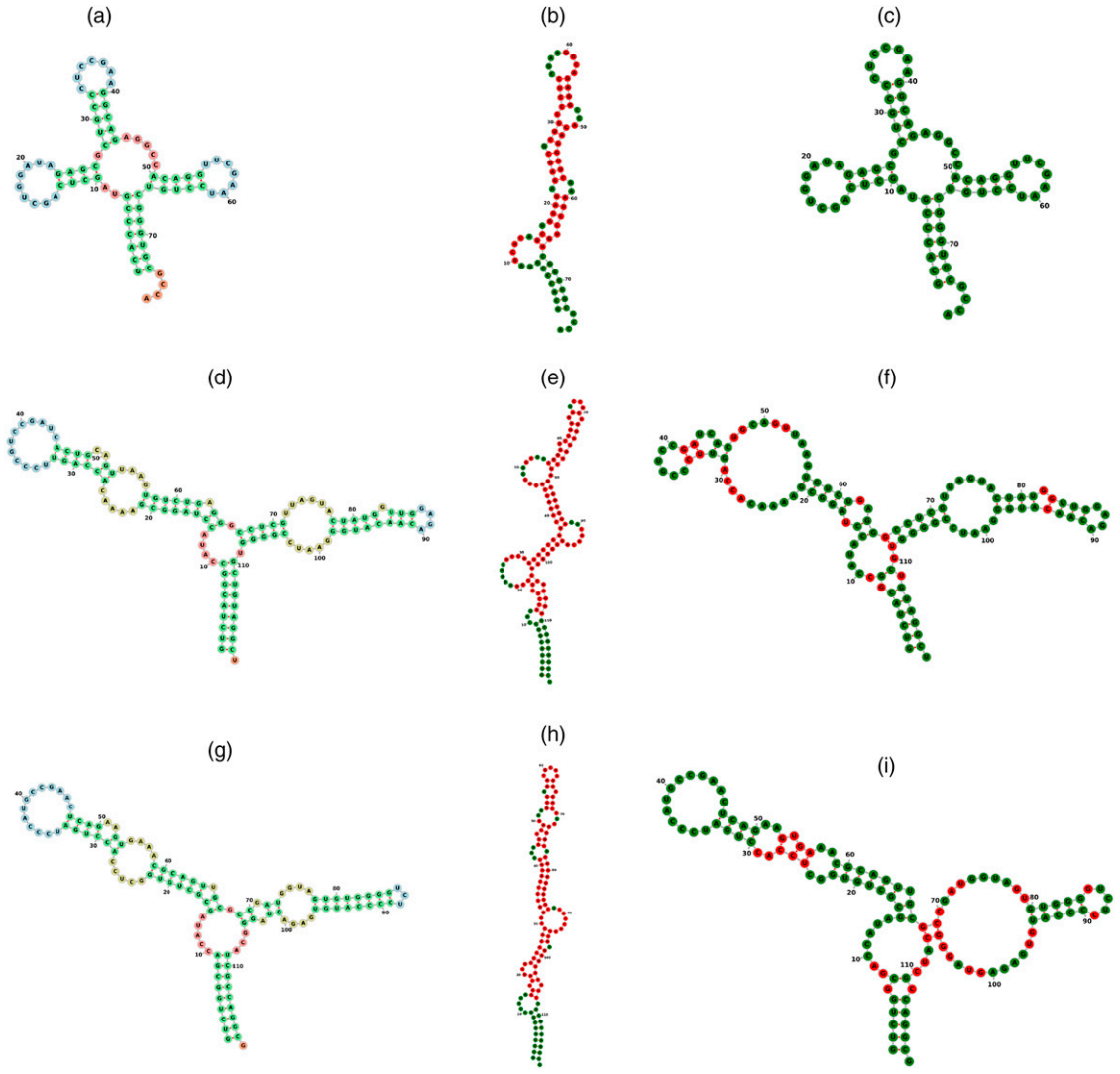
Notes. Each row contains the actual structure ((a), (d), (g)), the RNAfold-generated structure ((b), (e), (h)), and the ExpertRNA-generated structure ((c), (f), (i)). In the second and third columns, the nucleotides whose pairing was predicted incorrectly are shown in dark grey, whereas correct pairings are light grey; also, the numbers in parenthesis are foldability of the structure as evaluated by ENTRNA-NFE, free energy as evaluated by RNAfold, and MCC compared with the actual structure (in bold). The RNA in (a) is a phenylalanine tRNA from Archaeoglobus fulgidis; in (d), we find a 5s rRNA from Bacillus Thurigiensis; in (g), we find arginine tRNA from Treponema denticola.

algorithm. Whereas the design of a new expert is not in the scope of this paper, Figure 7(b) shows the potential in performance improvement when neural network models are used to evaluate the sequence-structure pair. Nonetheless, as SPOT-RNA is heavily trained on tRNA examples, the advantage is mainly because of this class of RNAs (Figure 7(a)), and generalizing this result to other RNA classes is a challenge.

4.3.1.1. Minimum Free Energy Analysis. Figures 8, (a) and (b), and 9, (a) and (b), show how both ExpertRNA and RNAfold-generated structures generally have lower FE as calculated by RNAfold than the actual structure. Figures 8(c) and 9(c) show that RNAfold returns a structure with equal or lower free energy than the one returned by ExpertRNA.

This is expected as RNAfold finds the lowest FE structure as defined by the Turner model, and it also reflects the result that ExpertRNA is able to predict non-MFE structures that are often more accurate by leveraging other measures of structure quality: in this case, the knowledge-based metrics used by ENTRNA,

Figure 13. (Color online) Example Structures from the Mathews Data Set with the Best Performance Improvement Between ExpertRNA and RNAfold



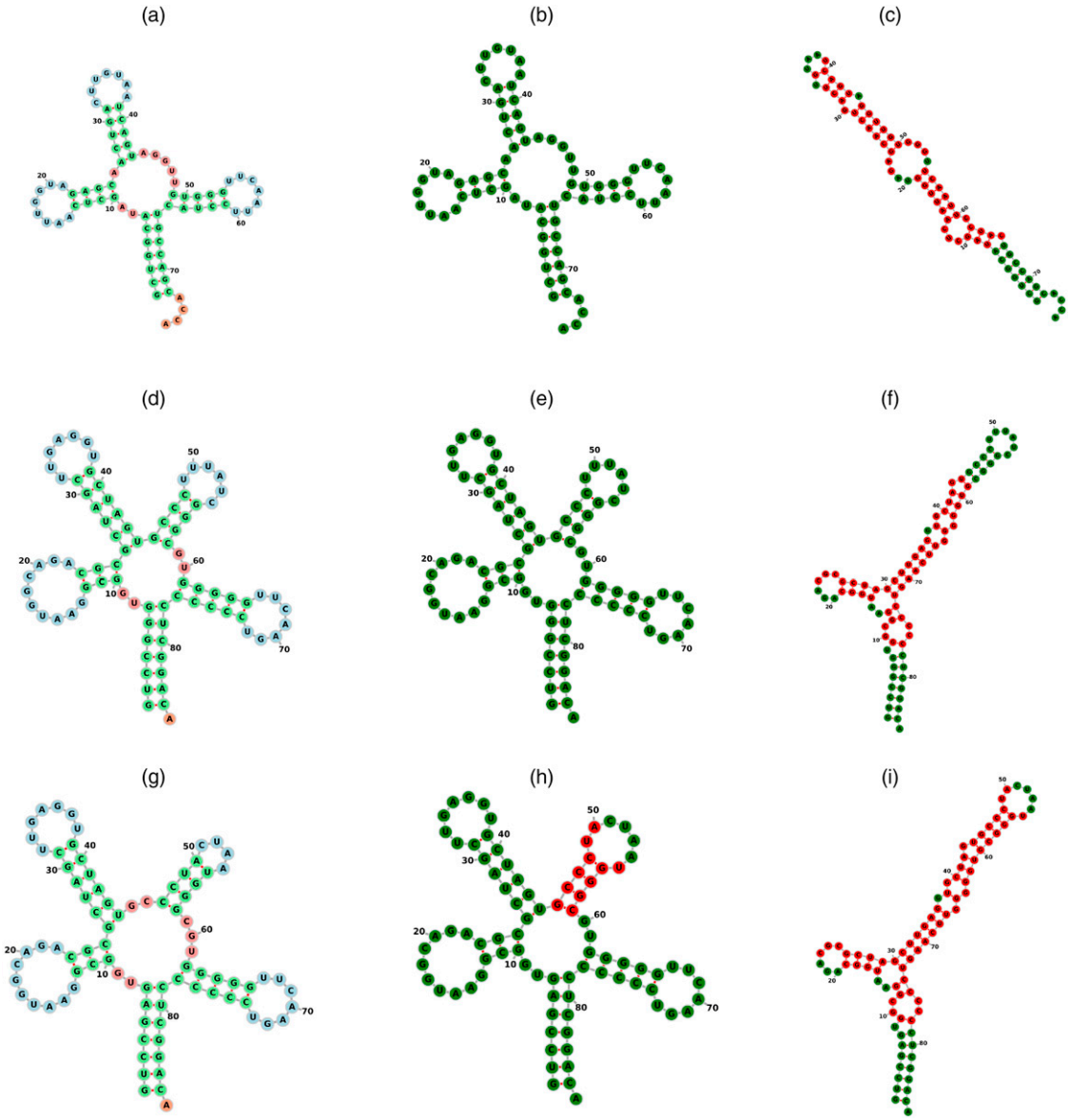
Notes. Each row contains the actual structure ((a), (d), (g)), the RNAfold-generated structure ((b), (e), (h)), and the ExpertRNA-generated structure ((c), (f), (i)). In the second and third columns, the nucleotides whose pairing was predicted incorrectly are shown in dark grey, whereas correct pairings are light grey; also, the numbers in parenthesis are foldability of the structure as evaluated by ENTRNA-NFE, free energy as evaluated by RNAfold, and MCC compared with the actual structure (in bold). The RNA in (a) is an agrinine tRNA from Brucella suis; in (d), we find a 5s rRNA from Schizosaccharomyces pombe; in (g), we have a 5s rRNA from Photobacterium profundum.

which may help correct deficiencies in the Turner model or reflect tertiary contacts and chemical modification present in the training data that RNAfold does not take into account.

4.3.1.2. Impact of the Expert Score. An important element of ExpertRNA is the expert adopted. Whereas, in principle, the algorithm can make use of any expert, it is important to understand the impact of the adopted expert on the solution. As detailed in Section 2, in this implementation, the adopted expert is a machine learning algorithm that, given a set of sequence-structure pairs, returns a score for each of them based on how "likely" the pairs are, and such likelihood is referred to as foldability in the original paper (Su et al. 2019). As shown in Figures 10 and 11, as expected, ExpertRNA predictions have the higher foldability. The fact that ExpertRNA performed better on the Rfam data set suggests that ENTRNA foldability is a slightly better expert than RNAfold's MFE measure. However, the many incorrect structures with higher foldability than the actual structures points to the need for more research in developing better performing expert software.

4.3.1.3. Predicted Structures. We show a sample of the structures generated by ExpertRNA and compare

Figure 14. (Color online) Example Structures from the Mathews Data Set with the Worst Performance Degradation Between ExpertRNA and RNAfold



Notes. Each row contains the actual structure ((a), (d), (g)), the RNAfold-generated structure ((b), (e), (h)), and the ExpertRNA-generated structure ((c), (f), (i)). Nucleotides whose pairing was predicted incorrectly are shown in dark grey in the second and third columns, whereas correct pairings are light grey. Also, in the second and third columns, the numbers in parenthesis are foldability of the structure as evaluated by ENTRNA-NFE, free energy as evaluated by RNAfold, and MCC compared with the actual structure (in bold). The RNA in (a) is a threonine tRNA from Clostridium felsineum; in (d), we have a leucine tRNA from a Frankia; the RNA in (g) is a leucine tRNA from Mycobacterium tuberculosis.

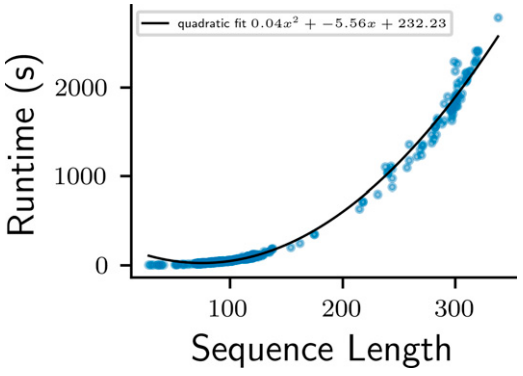
them to the prediction of the RNAfold algorithm alone as well as the actual structure. In particular, we look into cases in which the predicted structure generated by RNAfold significantly differs from the known structure. Visualizations were generated using the publicly available Forna software (Kerpedjiev et al. 2015). Figure 12 shows the structures for which RNAfold is particularly underperforming in the Rfam database.

Figure 13 shows the structures for which RNAfold is particularly underperforming in the Mathews data set, whereas Figure 14 shows structures for which ExpertRNA evaluated correct predictions from RNAfold as incorrect.

4.4. Computational Complexity

Another important aspect of structure folding is associated to the complexity of the algorithm. For the current implementation, ExpertRNA has a complexity $O(\sum_{i=1}^{N}(N-i)^2)$, where $(N-i)^2$ is the RNAfold complexity for a sequence of size $(N-i)$. Such quadratic behavior is evident in Figure 15 (the results are for the

Figure 15. (Color online) Computational Effort for ExpertRNA Across the Mathews Data Set with Quadratic Fit Shown



Mathews data set, and similar results were obtained for the Rfam data set). Whereas computational efficiency is a known challenge in secondary structure prediction, a promising avenue is to investigate multiagent rollout (Bertsekas 2020) as a way to boost the efficiency.

4.5. Summary

Compared with the RNAfold tool, ExpertRNA performs better on the data set of known structures from Rfam and approximately equivalently on the data set from the Mathews laboratory. By construction, the ExpertRNA structures had higher free energy (compared with RNAfold) according to the nearest neighbor model. However, they are closer to the actual known structure. Structures predicted by both ExpertRNA and RNAfold almost always have lower free energy than the actual structure, further indicating the limitations of free energy approximations in identifying correctly folded structures. In most cases, when ExpertRNA found a structure in better agreement with the actual known structure, the FE of that structure was only slightly higher than the FE of RNAfold. However, in a few cases (e.g., Figure 13, (a)–(c) and (g)–(i)), the solutions in which ExpertRNA had much better agreement with the correct structure, the RNAfold solution had much lower free energy. ExpertRNA, hence, shows the ability to identify the correct structure even if its predicted FE is quite higher than the MFE prediction by integrating other experts that may implicitly or explicitly account tertiary contact and kinetic information.

5. Conclusions

In this paper, we propose a new framework, ExpertRNA, for the automatic folding of nonpseudoknotted secondary structures for RNA molecular compounds. ExpertRNA builds upon the fortified rollout algorithm and generalizes the architecture to allow for the consideration of multiple experts that can evaluate, at each iteration, the solutions generated by the base heuristic. ExpertRNA allows one to control the growth of the rollout branches by enabling only a fixed number of alternatives at each iteration to be maintained. This differs from the traditional parallel expert implementation in which each expert is responsible for independent branches leading to the exponential growth of the maintained branches. Differently, ExpertRNA allows the folding algorithms to interact and the experts to evaluate all the solutions generated by all the folders at each iteration. This feature of ExpertRNA balances out different strengths and weaknesses of folding tools and experts. Numerical results show the advantage of the proposed approach especially over the data set in which the heuristic alone does not perform well. We show that, for several natural sequences, ExpertRNA is able to correctly predict secondary structures whose free energy is much higher than the MFE structure predicted by RNAfold.

Further investigation is being carried out to employ multiple different folding algorithms. We argue that, in the process of generating the solution, perhaps one of the folders will be performing very well, but after some iterations, a different folder may produce superior results. In some sense, ExpertRNA tracks the best folder, online, as the sequence is being constructed. Also, we are investigating the opportunity to use more/different experts. In light of this, a possible future direction is to include more features in ENTRNA training or a combination with additional expert scoring (e.g., based on covariation information of homologous sequences and SHAPE experimental data) would improve the prediction accuracy.

Finally, the introduced framework is not limited to RNA secondary structure and can be extended to de novo predictions of RNA tertiary or protein structures. Most of the currently used models for structure prediction allow specifying contact constraints that denote pairs of residues that are in contact. They are, hence, amenable to be used in ExpertRNA, which requires each folding tool to start from a partially formed structure. There are currently more than 200 different methods that participate in the critical assessment of structure prediction challenge that compares different folding algorithms of protein structure prediction (Moult et al. 2018) and more than 10 different methods in its RNA counterpart, RNA-puzzle (Cruz et al. 2012), and these algorithms can be used as folders in the ExpertRNA framework. The experts in such implementation can correspond to the system energy calculated, for example, with Rosetta (Rohl et al. 2004) and other knowledge-based potentials. The ExpertRNA tool is freely available at https://github.com/MenghanLiu212/RL-RNA. The scripts used to process the output of ExpertRNA into Figures 6 and 7 are freely available in the "analysis_scripts" directory of the repository.

References

Aghaeepour N, Hoos HH (2013) Ensemble-based prediction of RNA secondary structures. *BMC Bioinformatics* 14(1):139.

Andronescu M, Bereg V, Hoos HH, Condon A (2008) RNA STRAND: The RNA secondary structure and statistical analysis database. *BMC Bioinformatics* 9(1):340.

Andronescu M, Condon A, Hoos HH, Mathews DH, Murphy KP (2007) Efficient parameter estimation for RNA secondary structure prediction. *Bioinformatics* 23(13):i19–i28.

Andronescu M, Condon A, Hoos HH, Mathews DH, Murphy KP (2010) Computational approaches for RNA energy parameter estimation. *RNA* 16(12):2304–2318.

Antczak M, Popenda M, Zok T, Zurkowski M, Adamiak RW, Szachniuk M (2018) New algorithms to represent complex pseudoknotted RNA structures in dot-bracket notation. *Bioinformatics* 34(8):1304–1312.

Bertsekas DP (2019) *Reinforcement Learning and Optimal Control* (Athena Scientific, Belmont, MA).

Bertsekas DP (2020) *Rollout, Policy Iteration, and Distributed Reinforcement Learning* (Athena Scientific, Belmont, MA).

Bertsekas DP, Tsitsiklis JN (1996) *Neuro-Dynamic Programming* (Athena Scientific, Belmont, MA).

Bertsekas DP, Tsitsiklis JN, Wu C (1997) Rollout algorithms for combinatorial optimization. *J. Heuristics* 3(3):245–262.

Burge SW, Daub J, Eberhardt R, Tate J, Barquist L, Nawrocki EP, Eddy SR, Gardner PP, Bateman A (2013) Rfam 11.0: 10 years of RNA families. *Nucleic Acids Res.* 41(D1):D226–D232.

Calonaci N, Jones A, Cuturello F, Sattler M, Bussi G (2020) Machine learning a model for RNA structure prediction. *NAR Genomics Bioinformatics* 2(4):lqaa090.

Carell T, Brandmayr C, Hienzsch A, Müller M, Pearson D, Reiter V, Thoma I, Thumbs P, Wagner M (2012) Structure and function of noncanonical nucleobases. *Angewandte Chemie Internat. Edition* 51(29):7110–7131.

Chowdhury FRR, Zhang H, Huang L (2019) Learning to fold RNAs in linear time. Preprint, submitted November 24, https://www.biorxiv.org/content/10.1101/852871v1.

Cruz JA, Blanchet MF, Boniecki M, Bujnicki JM, Chen SJ, Cao S, Das R, et al. (2012) RNA-Puzzles: A CASP-like evaluation of RNA three-dimensional structure prediction. *RNA* 18(4):610–625.

Dieckmann T, Suzuki E, Nakamura GK, Feigon J (1996) Solution structure of an ATP-binding RNA aptamer reveals a novel fold. *RNA* 2(7):628–640.

Do CB, Woods DA, Batzoglou S (2006) Contrafold: RNA secondary structure prediction without physics-based models. *Bioinformatics* 22(14):e90–e98.

Elliott D, Ladomery M (2017) *Molecular Biology of RNA* (Oxford University Press, New York).

Geary C, Rothemund PW, Andersen ES (2014) A single-stranded architecture for cotranscriptional folding of RNA nanostructures. *Sci.* 345(6198):799–804.

Green AA, Silver PA, Collins JJ, Yin P (2014) Toehold switches: De-novo-designed regulators of gene expression. *Cell* 159(4):925–939.

Guo P (2010) The emerging field of RNA nanotechnology. *Nature Nanotechnology* 5(12):833–842.

Han D, Qi X, Myhrvold C, Wang B, Dai M, Jiang S, Bates M, et al. (2017) Single-stranded DNA and RNA origami. *Sci.* 358(6369):eaao2648.

Hanson J, Litfin T, Paliwal K, Zhou Y (2020) Identifying molecular recognition features in intrinsically disordered regions of proteins by transfer learning. *Bioinformatics* 36(4):1107–1113.

Hochrein LM, Schwarzkopf M, Shahgholi M, Yin P, Pierce NA (2013) Conditional dicer substrate formation via shape and sequence transduction with small conditional RNAs. *J. Amer. Chemical Soc.* 135(46):17322–17330.

Hofacker IL (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.* 31(13):3429–3431.

Huang Y, Niu B, Gao Y, Fu L, Li W (2010) CD-HIT suite: A web server for clustering and comparing biological sequences. *Bioinformatics* 26(5):680–682.

Isambert H, Siggia ED (2000) Modeling RNA folding paths with pseudoknots: Application to hepatitis delta virus ribozyme. *Proc. Natl. Acad. Sci. USA* 97(12):6515–6520.

Kerpedjiev P, Hammer S, Hofacker IL (2015) Forna (force-directed RNA): Simple and effective online RNA secondary structure diagrams. *Bioinformatics* 31(20):3377–3379.

Li XL, Yu PS, Liu B, Ng SK (2009) Positive unlabeled learning for data stream classification. *Proc. 2009 SIAM Internat. Conf. Data Mining* (SIAM), 259–270.

Lorenz R, Bernhart SH, Zu Siederdissen CH, Tafer H, Flamm C, Stadler PF, Hofacker IL (2011) Viennarna package 2.0. *Algorithms Molecular Biol.* 6(1):26.

Low JT, Weeks KM (2010) Shape-directed RNA secondary structure prediction. *Methods* 52(2):150–158.

Lucks JB, Mortimer SA, Trapnell C, Luo S, Aviran S, Schroth GP, Pachter L, Doudna JA, Arkin AP (2011) Multiplexed RNA structure characterization with selective 2′-hydroxyl acylation analyzed by primer extension sequencing (shape-seq). *Proc. Natl. Acad. Sci. USA* 108(27):11063–11068.

Mathews DH, Sabina J, Zuker M, Turner DH (1999) Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Molecular Biol.* 288(5):911–940.

Matthews BW (1975) Comparison of the predicted and observed secondary structure of t4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)–Protein Structure* 405(2):442–451.

Miao Z, Adamiak RW, Antczak M, Batey RT, Becka AJ, Biesiada M, Boniecki MJ, et al. (2017) RNA-puzzles round III: 3d RNA structure prediction of five riboswitches and one ribozyme. *RNA* 23(5):655–672.

Moult J, Fidelis K, Kryshtafovych A, Schwede T, Tramontano A (2018) Critical assessment of methods of protein structure prediction (CASP)—Round XII. *Proteins* 86(1):7–15.

Qi X, Liu X, Matiski L, Rodriguez Del Villar R, Yip T, Zhang F, Sokalingam S, et al. (2020) RNA origami nanostructures for potent and safe anticancer immunotherapy. *ACS Nano* 14(4):4727–4740.

Reuter JS, Mathews DH (2010) RNAstructure: Software for RNA secondary structure prediction and analysis. *BMC Bioinformatics* 11(1):129.

Rohl CA, Strauss CE, Misura KM, Baker D (2004) Protein structure prediction using Rosetta. *Methods Enzymology* 383:66–93.

Seetin MG, Mathews DH (2011) Automated RNA tertiary structure prediction from secondary structure and low-resolution restraints. *J. Comput. Chemistry* 32(10):2232–2244.

Singh J, Hanson J, Paliwal K, Zhou Y (2019) RNA secondary structure prediction using an ensemble of two-dimensional deep neural networks and transfer learning. *Nature Comm.* 10(1):1–13.

Sloma MF, Mathews DH (2017) Base pair probability estimates improve the prediction accuracy of RNA non-canonical base pairs. *PLOS Comput. Biol.* 13(11):e1005827.

Su C, Weir JD, Zhang F, Yan H, Wu T (2019) ENTRNA: A framework to predict RNA foldability. *BMC Bioinformatics* 20(1):373.

Sun Tt, Zhao C, Chen SJ (2018) Predicting cotranscriptional folding kinetics for riboswitch. *J. Physical Chemistry B* 122(30):7484–7496.

Tan ZJ, Chen SJ (2010) Predicting ion binding properties for RNA tertiary structures. *Biophysical. J.* 99(5):1565–1576.

Wang L, Liu Y, Zhong X, Liu H, Lu C, Li C, Zhang H (2019) DMFold: A novel method to predict RNA secondary structure with pseudoknots based on deep learning and improved base pair maximization principle. *Frontiers Genetics* 10:143.

Ward M, Datta A, Wise M, Mathews DH (2017) Advanced multi-loop algorithms for RNA secondary structure prediction reveal that the simplest model is best. *Nucleic Acids Res.* 45(14):8541–8550.

Watkins AM, Rangan R, Das R (2020) FARFAR2: Improved de novo Rosetta prediction of complex global RNA folds. *Structure* 28(8): 963–976.

Wayment-Steele HK, Kladwang W, Participants E, Das R (2020) RNA secondary structure packages ranked and improved by high-throughput experiments. Preprint, submitted May 31, https://www.biorxiv.org/content/10.1101/2020.05.29.124511v1.

Westhof E, Auffinger P (2006) RNA tertiary structure. *Encyclopedia of Analytical Chemistry: Applications, Theory and Instrumentation*.

Xia T, SantaLucia J Jr, Burkard ME, Kierzek R, Schroeder SJ, Jiao X, Cox C, Turner DH (1998) Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* 37(42): 14719–14735.

Yu AM, Gasper PM, Cheng L, Lai LB, Kaur S, Gopalan V, Chen AA, Lucks JB (2021) Computationally reconstructing cotranscriptional RNA folding from experimental data reveals rearrangement of non-native folding intermediates. *Molecular Cell* 81(4):870–883.

Zadeh JN, Steenberg CD, Bois JS, Wolfe BR, Pierce MB, Khan AR, Dirks RM, Pierce NA (2011) Nupack: Analysis and design of nucleic acid systems. *J. Comput. Chemistry* 32(1):170–173.

Zuker M, Stiegler P (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.* 9(1):133–148.

Zuker M, Mathews DH, Turner DH (1999) Algorithms and thermodynamics for RNA secondary structure prediction: A practical guide. *RNA Biochemistry and Biotechnology* (Springer), 11–43.