

Fitness functions for RNA structure design

Max Ward^{1,2,*}, Eliot Courtney² and Elena Rivas¹

¹Department of Molecular and Cellular Biology, Harvard University, Cambridge, MA, USA and ²Department of Computer Science & Software Engineering, University of Western Australia, Western Australia, Australia

Received June 25, 2022; Revised January 30, 2023; Editorial Decision January 31, 2023; Accepted February 22, 2023

ABSTRACT

An RNA design algorithm takes a target RNA structure and finds a sequence that folds into that structure. This is fundamentally important for engineering therapeutics using RNA. Computational RNA design algorithms are guided by fitness functions, but not much research has been done on the merits of these functions. We survey current RNA design approaches with a particular focus on the fitness functions used. We experimentally compare the most widely used fitness functions in RNA design algorithms on both synthetic and natural sequences. It has been almost 20 years since the last comparison was published, and we find similar results with a major new result: maximizing probability outperforms minimizing ensemble defect. The probability is the likelihood of a structure at equilibrium and the ensemble defect is the weighted average number of incorrect positions in the ensemble. We find that maximizing probability leads to better results on synthetic RNA design puzzles and agrees more often than other fitness functions with natural sequences and structures, which were designed by evolution. Also, we observe that many recently published approaches minimize structure distance to the minimum free energy prediction, which we find to be a poor fitness function.

INTRODUCTION

Ribonucleic Acid (RNA) is a versatile molecule (1). It is fundamental to life (2) and has a breadth of roles including transcription and translation (3), catalyzing reactions (4), gene regulation (5), maintaining telomeres (6) and beyond. The functions of these RNAs are often governed by their structure (7). Additionally, synthetic RNAs can easily be constructed (8). A consequence of these properties is that RNA is a popular tool for bioengineering therapeutics (including mRNA vaccines (9)) and biomachines (10–12).

RNAs have the quality that fast and (relative to other molecules) accurate algorithms exist to predict their struc-

tures *in silico* (13–17). Since structure usually determines function in biology, a multitude of attempts have been made to develop algorithms to automatically design RNAs with specific structures (18). We refer to this as the *RNA design* problem. Algorithmic RNA design dates back to at least the early 1990s (19). These algorithms have led to some promising, practical successes (20–23). However, reliably solving RNA design puzzles is an open problem. No algorithm has solved the entire EteRNA100, a widely used RNA design benchmark that was hand-crafted by humans (24). Recent results suggest the RNA design problem is NP-hard by proving that the most general version of the problem is hard (25) as well as a simplified model (26,27).

Many techniques have been applied to the RNA design problem. We give a non-exhaustive sample for context. The first method was likely RNAinverse (19), which applied an adaptive random walk. Later methods used stochastic local search (28) with seeding heuristics (29), genetic algorithms (30–32), constraint programming (33–35), simulated annealing (36), hierarchical decomposition (37,38), Monte Carlo methods (39–41), global sampling (42,43) and ant colony optimisation (44). The EteRNA project has turned RNA design into a game in which solutions to RNA design puzzles can be crowd sourced from human players (45). Recently, deep learning has also been applied to the problem in several ways including learning from human solutions (46), and by applying reinforcement learning (47,48).

Each of these RNA design algorithms generally have three components:

- (1) A computational *model* (encompassing multiple components (17)), that can predict the structure of an RNA given its sequence.
- (2) A *fitness function*, which determines the quality of a potential solution with reference to a target structure.
- (3) A *search algorithm*, which explores the sequence space to find a desirable solution under the fitness function. The choice of possible search algorithms is often limited by the choice of fitness function.

The thermodynamic ‘nearest neighbor’ model developed by Turner, Mathews, Tinoco and many others (49) is ubiquitous in the RNA design literature. The implementation of the nearest neighbor model in the ViennaRNA package was

*To whom correspondence should be addressed. Tel: +61 8 6488 2281; Email: max.ward@uwa.edu.au

used in our work. This is the most widely used implementation (13). However, it should be noted that other models exist and may offer new options for different fitness functions and search algorithms (17).

Generally speaking, design algorithms require a *fitness function* to guide their search. This is a measure of how promising an intermediate solution is. The fitness functions used are almost as numerous as the algorithms. However, most algorithms use fitness functions that are a variant of one of the four main types. These will be defined formally later. They comprise *structure distance minimization*, *free energy minimization*, *probability maximization* and *ensemble defect minimization*.

Our aims are to summarize information about fitness functions for RNA design, and to provide an objective analysis of them. The latter is achieved by experiments. We use a fitness function agnostic search algorithm to do a fair benchmark. Both our aims are relevant to future works developing computational approaches to RNA design. The only similar work we are aware of is by Dirks *et al.* (50). We significantly expand and modernize the methods used, and find new results that differ somewhat from those of Dirks *et al.* We find that probability maximization performs better than any other fitness function, which differs from the previous result. This has implications for algorithms for RNA design, many of which currently use other fitness functions.

The computational RNA design problem

Before we begin our discussion of fitness functions, we must understand the rules of the game: what is RNA design?

An RNA is represented by a string $p \in \{A, U, G, C\}^*$ where the letters correspond to nucleotides. We refer to this string p as the *sequence*. A secondary structure s is a (possibly empty) set of pairings between the indexes of p such that an index appears in s at most once and there are no two elements $(i, j) \in s$ and $(k, l) \in s$ such that $i < k < j < l$. In short, s is a properly nested set of pairings in p . In nature, some RNA structures contain improperly nested pairings called pseudoknots. Also, some natural structures contain a single nucleotide binding to more than one other nucleotide forming a triplex or quadruplex. Since the computational model we use does not support these well, we do not consider them. Henceforth we elide *secondary structure* to just *structure* for brevity.

Call $S(p)$ the set of all possible structures for a sequence p . Define a *model* $m(p, s)$ as a function that assigns every $s \in S(p)$ a score. Since the structure represents bonds that form as the RNA sequence folds, we call the process of finding the optimal structure *folding* and define $\text{FOLD}(m, p) = \arg \min_{s \in S(p)} m(p, s)$. Ties for the $\arg \min$ are broken arbitrarily. The solution to $\text{FOLD}(m, p)$ is the best guess of the true structure of the RNA under the model. Assuming the widely used thermodynamic model (49), $\text{FOLD}(m, p)$ can be computed in $O(|p|^3)$ time (16,51).

The algorithmic RNA design problem is essentially to invert the folding problem. The folding problem takes a sequence p and finds an optimal structure s . The design problem inverts this relationship by taking a structure s and finding a sequence p that folds into s . Formally, we are given a structure s and a model m . Define the inverse folding func-

tion $\text{FOLD}^{-1}(m, s) = \{p \in \{A, U, G, C\}^{|s|} \mid \text{FOLD}(m, p) = s\}$. Any sequence $p \in \text{FOLD}^{-1}(m, s)$ is a solution to the RNA design problem.

Sometimes, there can be ties for the optimal fold $\text{FOLD}(m, p)$. This happens when there are multiple structures with the minimum score. This issue is often overlooked in the literature. There are many published results where a folding algorithm is run and whatever arbitrary structure it produces in the case of ties is taken—a non-exhaustive list: (18,39,44,45). There are two ways to address the issue. We can say that a sequence p is a correct design for a structure s when it is tied with the optimal fold: $m(p, s) = m(p, \text{FOLD}(m, p))$. Alternatively, we can say that p is a correct design iff $\text{FOLD}(m, p) = s$ and there are no ties. We adopt the latter tie-breaking strategy because it is more stringent and should lead to more accurate designs *in vivo*. If there are ties for an optimal structure, it means the model cannot determine which is the most likely. An ideal design should ensure the target structure is non-ambiguously selected under the model.

Fitness functions

Success on the algorithmic RNA design problem as we have defined it is binary. Either a design is correct, or it is not. This is not directly usable for most search algorithms, which need to be able to make incremental progress toward a solution. This is why fitness functions are important. Their use is to estimate the effectiveness of intermediate sequence solutions towards a target structure. Next, we describe the typical fitness functions used in RNA design.

Structure distance minimization. The key idea in structure distance minimization is to make $\text{FOLD}(p, m)$ as similar to the target structure as possible. Given a distance function $d(s_1, s_2)$ between two structures, the structure distance for a given potential solution sequence p and target structure t is $d(t, \text{FOLD}(p, m))$. Minimizing the structure distance can lead to a correct design. This category of fitness function seems to be the most widely used (19,28,30,36,39,40,48).

It is important to choose a good distance function. In our experiments, we use three that are widely used in the literature. The first is base pair distance as implemented in the ViennaRNA software package (13). Base pair distance is the number of base pairs that occur in exactly one of the two compared structures. The second is the number of indexes in p where the structure differs from the target, defined as the number of incorrect nucleotides in (50). The third is Interaction Network Fidelity (INF) applied to secondary structures, which is the Mathews correlation coefficient of predicted and true base pairs (52,53).

A weakness of using structure distance is that it is unclear what to do in the case of ties for the optimal fold. The typical approach in the literature seems to be not to address this edge case and accept the arbitrary result of FOLD for structure distance comparison. We try several methods of dealing with ties including using the average distance and minimum distance. Another weakness is that structure distance myopically considers only the results of FOLD while ignoring all other structures. This can make structure distance a poor guide when the target structure is close to op-

timal, but not an optimal fold. Due to these weaknesses, we expected that structure distance is not an effective fitness function. The expectation that structure distance is a poor metric may be considered controversial, since it is widely used. However, the previous work on fitness functions by Dirks *et al.* found structure distance (they call it MFE satisfaction) to have middling efficacy (50).

Free energy minimization. Most existing RNA design algorithms use a thermodynamic nearest neighbour model. This model assigns each structure $s \in S(p)$ a free energy score. Lower free energy structures are more likely at equilibrium and therefore more likely to be the true structure. Free energies can be used directly for RNA design. The utility of a potential design p for a target structure t can be defined as $m(p, t)$. It seems natural that we want to select designs that minimize the free energy of our target structure. This fitness function is often used in conjunction with other metrics (29,30,39), but it has been used independently too (50,54).

Despite its intuitive appeal, free energy minimization has a major weakness when used for algorithmic RNA design. Consider two sequences p_1 and p_2 and a structure $s \in S(p_1) \cap S(p_2)$ where $m(p_1, s) > m(p_2, s)$. It is possible that $\text{FOLD}(m, p_1) = s$ and $\text{FOLD}(m, p_2) \neq s$. In words, s may be more stable in p_2 compared to p_1 , but not the most stable among all structures with respect to p_2 while being the most stable over all structures for p_1 . This is because a ‘good’ score in $S(p_1)$ may not be a ‘good’ score in $S(p_2)$. In essence, we cannot sensibly compare scores drawn from different thermodynamic ensembles. Additionally, if free energy minimization was a perfectly effective fitness function, then either algorithmic RNA design would not be NP-hard or $\text{NP} = \text{P}$ since a sequence with the minimum possible free energy for a structure can be found in polynomial time (29). Due to this weakness, we expect that free energy minimization is not an effective fitness function. Dirks *et al.* previously found free energy minimization to be a relatively poor fitness function (50).

Probability maximization. The thermodynamic equilibrium partition function for RNAs under the thermodynamic model can be computed efficiently, requiring $O(|p|^3)$ time, the same as computing $\text{FOLD}(m, p)$ (55). This allows us to compute the conditional probability $\mathbf{P}(s | p)$ of a structure s given a sequence p . This probability can be maximized and used as a fitness function for RNA design. This fitness function has been used for RNA design, (19,31,50) albeit less widely than some other fitness functions.

Using probability maximization is similar to free energy minimization, but it addresses many of the major shortcomings. Because the probabilities are normalized to the same scale, it is more sensible to compare two probabilities taken from different sequences. In this sense, probability can be thought of as a ‘better’ free energy minimization. There is still a weakness, however. Suppose we are comparing two sequences p_1 and p_2 as potential designs for a target structure t . It is possible that t is the best ranked (by probability) for p_1 , but has a higher probability and a worse rank in p_2 . This can happen when the probability distributions look dissimilar between sequences, for example p_1 could have a

flat distribution and p_2 could have an exponential distribution.

This weakness is similar to the weakness of free energy minimization. However, it should have a lower negative impact, since it requires the distributions of structure probabilities for sequences to look dissimilar. For this reason, we expected that probability will be a better fitness function than free energy. Dirks *et al.* found that probability maximization was an effective fitness function (50).

Ensemble defect minimization. Ensemble defect was originally introduced by Dirks *et al.* who found it to be the best fitness function they tested (50). They called it ‘average number of incorrect nucleotides,’ which describes what it measures. It was later called ensemble defect (37). In essence, the ensemble defect of a sequence p with respect to a target structure t is the sum of a certain distance function over the ensemble of structures $S(p)$ weighted by probability.

A structure s is a set of paired nucleotide locations. For convenience, let $s_i = j$ if i and j are paired, and $s_i = i$ if i is not paired to any nucleotide. Let our distance function be the average number of incorrect nucleotides between a structure s and a target structure t such that $|s| = |t|$, similar to a Hamming distance:

$$d(s, t) = \sum_{1 \leq i \leq |s|} \begin{cases} 0 & \text{if } s_i = t_i \\ 1 & \text{otherwise} \end{cases} \quad (1)$$

Define $\mathbf{P}(s | p)$ as the probability of the structure s given the sequence p . The ensemble defect of a sequence p for a target structure t is defined as:

$$\mathcal{D}(p, t) = \sum_{s \in S(p)} \mathbf{P}(s | p) \times d(s, t) \quad (2)$$

It is possible to compute $\mathcal{D}(p, t)$ in $O(|p|^3)$ time, making it no less efficient to compute than comparable fitness functions (50). While not as widely used as structure distance minimization, ensemble defect is well known (33,37,38,50,56). It appears to combine the strengths of probability maximization and structure distance minimization while avoiding the weaknesses of both. We expect that it will be the most effective fitness function.

MATERIALS AND METHODS

We ran two different experiments. The first used synthetic RNAs, and the second used real RNAs. In the first, we used a fitness function agnostic search algorithm to test the performance of each fitness function type on synthetic RNAs. In the second, we took real RNA sequences with conserved structures and determined if the fitness functions predict that the true sequence is a good design for the true structure.

We used the ViennaRNA package 2.5.0 (13) as the model and the RNAfold program as the folding algorithm for all our experiments.

Synthetic RNAs

A search algorithm is needed to test the fitness functions. It must be compatible with all fitness functions. We used an

Adaptive Random Walk (ARW), since it supports all fitness functions, is widely known, and is relatively simple. Dirks *et al.* (50) used an ARW, as does the RNAinverse program in ViennaRNA (13,19). The algorithm is described in Algorithm 1, and makes a sequence of mutations that maintain base pairs and accepting only those that improve the fitness function score. Our introduction points out the diversity of search algorithms for the RNA design problem. By restricting ourselves to a single search algorithm, we limit the conclusions we can make about different search algorithms. However, we have picked a search algorithm that is similar to local search subroutines in many algorithms (19,38,40,56), and is also comparable to techniques that have elements of local search like Monte Carlo methods, genetic and evolutionary algorithms, and simulated annealing.

Algorithm 1 Adaptive Random Walk.

```

function WALK( $t, f, \text{steps}$ )
   $p \leftarrow$  a random sequence where  $t \in S(p)$ 
  loop steps iterations
     $p' \leftarrow$  MUTATE( $p, t$ )
    if  $f(p') \geq f(p)$  then
       $p \leftarrow p'$ 
    end if
  end loop
  return  $p$ 
end function

function MUTATE( $p, t$ )
   $i \leftarrow$  a random index in the range  $[1, |p|]$ 
   $p'[i] \leftarrow p$ 
   $p'[i] \leftarrow$  a random nucleotide in  $\{A, U, G, C\}$ 
  if  $i$  is paired in  $t$  then
     $j \leftarrow$  the index paired to  $i$  in  $t$ 
     $p'[j] \leftarrow$  a random valid paired nucleotide with  $p'[i]$ 
  end if
  return  $p'$ 
end function
  
```

The ARW was run using each fitness function. ARW was run for 1000 steps on each target structure. A result sequence p was considered correct for a target structure t if it was the unique solution; that is it was the unique minimum free energy structure as judged by ViennaRNA (13). If there were ties or if p did not fold into the target structure, then a result was considered incorrect. For each fitness function, we counted the number of correct and incorrect results. Better fitness functions should guide the ARW to good solutions more often, so the rate of correct results was used to measure fitness function efficacy.

Three synthetic data sets were used. A 40nt, 80nt, and 120nt set. In all sets, target structures were generated similarly. First, sequences were generated by picking each nucleotide with equal probability. A target structure was picked randomly for each sequence from the set of structures within T kcal/mol of the minimum free energy. A different T value was picked in different data sets to adjust difficulty. We use this method because always picking a minimum free energy structure tended to generate structures that were easily solved by the ARW. The resulting set of target structures for each sequence was used as our testing data set. Wuchty's algorithm (57) in ViennaRNA (13) was used to generate all suboptimal structures within the free energy window. We ensured that a data set contained no duplicate structures.

For the 80nt and 120nt data sets, 1600 structures of lengths 80 and 120 respectively were generated with $T = 5.0$ kcal/mol. For the 40nt data set, 1600 structures of length 40 with $T = 1.0$ kcal/mol were generated. These values (1600, 80, 120, 5, 40, 1) were picked before examining the perfor-

mance results. Running the ARW for 1000 steps was an arbitrary parameter that was also chosen before examining performance results. Length 120 was picked because it was the largest length that ran in time on an AMD 5950X processor. Lengths 80 and 40 were picked arbitrarily. A window of $T = 5.0$ kcal/mol was chosen because 6.0 was too slow on our processor, and 1.0 was picked because it was significantly smaller than 5.0.

The code used to generate the data sets is provided in the *Data Availability* section. These can be used to generate the same set of structures used in our experiments. Statistics about the structural motifs in our data set are given in Table 1.

The previous work by Dirks *et al.* tested on 11 structures (50). Of these, 9 were variants of the a tRNA inspired three-way multiloop structure with varying stem lengths and numbers of unpaired nucleotides. Of the two remaining, one is a larger multiloop, and the other is a small pseudoknot. In contrast, we use 4800 structures with a much greater degree of diversity due to our method of generation. In particular, we expect our 80nt and 120nt data sets to contain much more difficult to solve structures. A major issues with the Dirks test set in addition to its small size is that all structures were solved by both probability and ensemble defect. We believe harder puzzles are needed to distinguish between these two.

We do not believe that our structure generation method has a bias toward producing structures that are easier for any particular fitness function, because the results using synthetic data are consistent across all data sets and with those from real RNAs. However, we could not find a way to ensure no bias in data generation for this or any other method except picking structures uniformly at random. Uniform random structure generation could not be used because we found it almost always generates structures no fitness function can solve.

Recall that the ARW algorithm starts with a random seed sequence and modifies it. This sequence was picked uniformly at random. We tested all fitness functions on the three data sets with different seeding strategies for the ARW. This was to assess the sensitivity of the fitness functions to the seed sequence. Seed sequences were tested with 25% GC-content and 75% as well as picking sequences uniformly at random, which have 50% GC-content.

Structure distance. We tried several variants of structure distance. We used 'base pair distance' (BPD), which is a standard distance measure included in ViennaRNA (13) and defined as the number of base pairs that occur in exactly one of the two compared structures. We also used a structural Hamming distance (HD) defined the same as in Equation (1). Finally, Interaction Network Fidelity (INF) was used (52,53).

For all of these distance functions, we tried three different ways of dealing with ties for the minimum free energy structure. This issue is described previously in **Fitness Functions**. We tried using the single arbitrary structure ViennaRNA returns; taking the average distance over all ties; and taking the minimum distance over all ties.

Both BPD and HD give distance values in the range from 0 to n for n nucleotides. Because our ARW implementation

Table 1. Statistics about the structural motifs in the Nearest-Neighbor model across the three synthetic data sets. Each data set contains 1600 structures. Average counts over the data set (/1600) are included in brackets. We define a helix as a sequence of consecutive base pairs with no loops

| Data set | % Paired | Helices | Hairpin loops | Bulge loops | Internal loops | Multiloops |
|----------|----------|-------------|---------------|-------------|----------------|------------|
| 40nt | 44% | 4184 (2.62) | 1978 (1.24) | 868 (0.54) | 1331 (0.83) | 7 (0.00) |
| 80nt | 50% | 9865 (6.17) | 3374 (2.11) | 2321 (1.45) | 3803 (2.38) | 367 (0.23) |
| 120nt | 54% | 14724 (9.2) | 4488 (2.81) | 3622 (2.26) | 5735 (3.58) | 879 (0.55) |

maximizes fitness values, we used $n - d(s_1, s_2)$ as our fitness value where $d(s_1, s_2)$ represents some distance function between the structures s_1 and s_2 . INF gives a value between 0 and 1, so $1 - d(s_1, s_2)$ was used.

Free energy. The free energy as evaluated by the ViennaRNA package (13) was used. Because a lower free energy is more stable, and our ARW maximizes fitness, we use $-\Delta G$ as the fitness value, where ΔG is the free energy change.

Free energy minimization produces sequences with a high GC-content (50). This is known to cause issues in RNA design (28,29,43). To address this, we test a variant of free energy minimization with a GC-content constraint which penalises excess GC exponentially. The ARW maximizes Equation (3) where g is the target GC-content ratio (0.5 in our experiments) and \hat{g} is the actual GC-content.

$$-\Delta G - 10 \times \frac{e^{\max(0, \hat{g} - g)} - 1}{e - 1} \times |\Delta G| \quad (3)$$

Probability. The probability $\mathbf{P}(s | p)$ for a structure s given a sequence p as evaluated by the ViennaRNA package (13) was used. This was achieved by computing the partition function for the sequence p . Probability was maximized directly by the ARW.

Ensemble defect. The normalized ensemble defect was calculated using the method provided in the ViennaRNA package (13). This is the same as in Equation (2), but normalized by sequence length: $\frac{\mathcal{D}(p, t)}{|p|}$. Because our ARW maximizes fitness, we optimized $1 - \frac{\mathcal{D}(p, t)}{|p|}$.

Real RNAs

Many RNAs in nature have conserved structures. Consider a natural RNA sequence p with a known structure s . A fitness function f can be used to rank every structure in $S(p)$ as a design for p . We can assign each structure $s' \in S(p)$ a score $f(p, s')$ then order $S(p)$ according to these scores. A good fitness function is expected to rank the true structure s highly. Natural sequences have been ‘designed’ through natural selection to fold into their structure. While the computational folding algorithm only approximates this, we do expect that the real structure for a natural sequence is seen as a good design. A subtlety worth emphasizing is that we are not comparing multiple sequences for a single target structure, as with synthetic RNAs. Instead, we are looking at all the structures for a single, natural sequence that has a single true structure.

Using natural sequences is important to fairly compare ensemble defect. It is claimed that ensemble defect is expected to make good predictions on natural sequences due

minor base pairing fluctuations *in vivo* and the observed low ensemble defect of natural helices (37,38,50). Despite this, the previous work on fitness functions by Dirks *et al.* (50) did not test on natural sequences, so we consider this analysis novel.

We used *ArchiveII*, a curated collection of 3948 natural RNA sequences with conserved structures (58). For each, suboptimal folds were generated using the implementation of Wuchty’s algorithm (57) in ViennaRNA (13). The suboptimal free energy window started at 0 kcal/mol and was increased in increments of 0.2 kcal/mol until at least 200 000 suboptimal structures were generated or the window exceeded 10 kcal/mol. The structures were ranked by their corresponding fitness function values. The rank of the RNA’s true structure was found in the ranked list of structures. If it was not found, the closest structure by base pair distance (13) was used. If the closest structure differed by >5% of the sequence length, the sequence was excluded from the experiment. A total of 1719 sequences were excluded this way. The 5% similarity threshold was picked because often the true structure does not appear in the suboptimal list (e.g., it contains a base pair not allowed in the nearest neighbour model), but a nearly identical structure does appear.

Probability, ensemble defect, and structure distance were considered in our experiment. We opted not to test free energy. In the special case of this experiment, free energy necessarily always has the same ranking as probability because the sequence is fixed. For structure distance, we considered only Hamming distance breaking ties for the minimum free energy by averaging because it is representative of other structure distance methods on synthetic RNAs.

RESULTS

Synthetic RNAs

Tables 2-4 summarize the results for synthetic RNAs. The first table shows probability and ensemble defect with the strongest performances on the easier synthetic data set. Ensemble defect and probability perform similarly. In order, the performance seems to be free energy, structure distance, ensemble defect, probability. This is somewhat similar to the results found by Dirks *et al.* (50) who found that both probability and ensemble defect solved their entire test set.

The second table contains results from the 80nt synthetic data set and shows probability with the highest correct rate and number of unique solutions. The same performance hierarchy is maintained, but probability pulls away by a larger margin. The number of unique solves is of particular interest where probability dominates all other fitness functions.

Table 2. Results on synthetic structures of length 40. The ‘# Correct’ is the number of correct solutions out of 1600. The ‘Correct Rate’ is the ratio of the number of correct solutions and 1600. The ‘Unique Solver’ column contains the number of structures for which a fitness function was the only fitness function to find a correct solution

| Fitness function | # Correct | Correct rate | GC-percent | Unique solver |
|--|-----------|--------------|------------|---------------|
| Ensemble defect | 1530 | 0.96 | 0.52 | 1 |
| Free energy | 249 | 0.16 | 0.74 | 0 |
| Free energy (GC-controlled) | 546 | 0.34 | 0.49 | 0 |
| Probability | 1565 | 0.98 | 0.52 | 14 |
| Structure distance (BPD; arbitrary tie breaking) | 1260 | 0.79 | 0.51 | 0 |
| Structure distance (BPD; average tie breaking) | 1437 | 0.90 | 0.51 | 0 |
| Structure distance (BPD; minimum tie breaking) | 1101 | 0.69 | 0.50 | 0 |
| Structure distance (HD; arbitrary tie breaking) | 1248 | 0.78 | 0.51 | 1 |
| Structure distance (HD; average tie breaking) | 1425 | 0.90 | 0.51 | 1 |
| Structure distance (HD; minimum tie breaking) | 1098 | 0.69 | 0.50 | 0 |
| Structure distance (INF; arbitrary tie breaking) | 1262 | 0.79 | 0.51 | 0 |
| Structure distance (INF; average tie breaking) | 1427 | 0.89 | 0.50 | 0 |
| Structure distance (INF; minimum tie breaking) | 1107 | 0.69 | 0.50 | 0 |

Table 3. Results on synthetic structures of length 80. The ‘# Correct’ is the number of correct solutions out of 1600. The ‘Correct Rate’ is the ratio of the number of correct solutions and 1600. The ‘Unique Solver’ column contains the number of structures for which a fitness function was the only fitness function to find a correct solution

| Fitness function | # Correct | Correct rate | GC-percent | Unique solver |
|--|-----------|--------------|------------|---------------|
| Ensemble defect | 906 | 0.57 | 0.58 | 23 |
| Free energy | 14 | 0.01 | 0.77 | 0 |
| Free energy (GC-controlled) | 74 | 0.05 | 0.50 | 0 |
| Probability | 1185 | 0.74 | 0.59 | 218 |
| Structure distance (BPD; arbitrary tie breaking) | 353 | 0.22 | 0.50 | 0 |
| Structure distance (BPD; average tie breaking) | 406 | 0.25 | 0.50 | 0 |
| Structure distance (BPD; minimum tie breaking) | 268 | 0.17 | 0.50 | 1 |
| Structure distance (HD; arbitrary tie breaking) | 313 | 0.20 | 0.50 | 0 |
| Structure distance (HD; average tie breaking) | 428 | 0.27 | 0.50 | 2 |
| Structure distance (HD; minimum tie breaking) | 241 | 0.15 | 0.50 | 1 |
| Structure distance (INF; arbitrary tie breaking) | 331 | 0.21 | 0.50 | 1 |
| Structure distance (INF; average tie breaking) | 393 | 0.25 | 0.50 | 0 |
| Structure distance (INF; minimum tie breaking) | 258 | 0.16 | 0.50 | 2 |

Table 4. Results on synthetic structures of length 120. The ‘# correct’ is the number of correct solutions out of 1600. The ‘correct rate’ is the ratio of the number of correct solutions and 1600. The ‘unique solver’ column contains the number of structures for which a fitness function was the only fitness function to find a correct solution

| Fitness function | # Correct | Correct rate | GC-percent | Unique solver |
|--|-----------|--------------|------------|---------------|
| Ensemble defect | 721 | 0.45 | 0.61 | 29 |
| Free energy | 2 | 0.00 | 0.78 | 0 |
| Free energy (GC-controlled) | 12 | 0.01 | 0.50 | 0 |
| Probability | 1149 | 0.72 | 0.62 | 370 |
| Structure distance (BPD; arbitrary tie breaking) | 128 | 0.08 | 0.50 | 0 |
| Structure distance (BPD; average tie breaking) | 203 | 0.13 | 0.50 | 3 |
| Structure distance (BPD; minimum tie breaking) | 111 | 0.07 | 0.50 | 0 |
| Structure distance (HD; arbitrary tie breaking) | 137 | 0.09 | 0.50 | 0 |
| Structure distance (HD; average tie breaking) | 213 | 0.13 | 0.50 | 1 |
| Structure distance (HD; minimum tie breaking) | 87 | 0.05 | 0.50 | 0 |
| Structure distance (INF; arbitrary tie breaking) | 136 | 0.09 | 0.50 | 0 |
| Structure distance (INF; average tie breaking) | 197 | 0.12 | 0.50 | 0 |
| Structure distance (INF; minimum tie breaking) | 110 | 0.07 | 0.50 | 0 |

The third table contains results for the 120nt synthetic data set. We see similar results as in the 80nt but more exaggerated. All fitness functions have lower performance, but probability is barely impacted, while the performance of other fitness functions falls precipitously.

We tried several variations of structure distance. The results suggest that HD, BPD and INF are all comparable. Further, using an average to break ties is much better than arbitrary tie breaking (which is the norm in the literature) and minimum tie breaking.

The GC-content of seed sequences was tested using 25% and 75% GC-content seeds in addition to our default seeding strategy, which produces roughly 50% GC-content. Table 5 summarizes the results. While there is some minor variation, the seeds did not seem to have a significant effect on the number of solves or the final GC-content of the ARW for any fitness function. Note that Hamming distance with an average tie breaking strategy was used as representative for structure distance.

Table 5. Results on synthetic structures of all lengths with different GC-content percentages used for the seed sequences. For example, 80nt/25% indicates the 80nt dataset was used and the seed sequence for the adaptive walk was generated with 25% of the nucleotides as G or C. The reported results are the number of correct solves and the final GC-content of the solution in brackets. For example, 512 (0.49) indicates 512 correct solutions and the average GC-content of the final sequences found by adaptive walk using this fitness function was 49%

| Fitness function | 40nt/25% | 40nt/75% | 80nt/25% | 80nt/75% | 120nt/25% | 120nt/75% |
|----------------------------------|-------------|-------------|-------------|-------------|-------------|-------------|
| Ensemble defect | 1522 (0.52) | 1526 (0.52) | 906 (0.59) | 921 (0.58) | 775 (0.61) | 725 (0.61) |
| Free energy | 273 (0.74) | 226 (0.75) | 15 (0.76) | 8 (0.78) | 5 (0.77) | 2 (0.79) |
| Free energy (GC-controlled) | 545 (0.49) | 503 (0.49) | 79 (0.50) | 78 (0.50) | 12 (0.50) | 18 (0.50) |
| Probability | 1556 (0.52) | 1561 (0.52) | 1167 (0.59) | 1195 (0.59) | 1143 (0.62) | 1152 (0.62) |
| Structure distance (HD; average) | 1412 (0.51) | 1440 (0.51) | 436 (0.50) | 435 (0.50) | 213 (0.50) | 183 (0.50) |

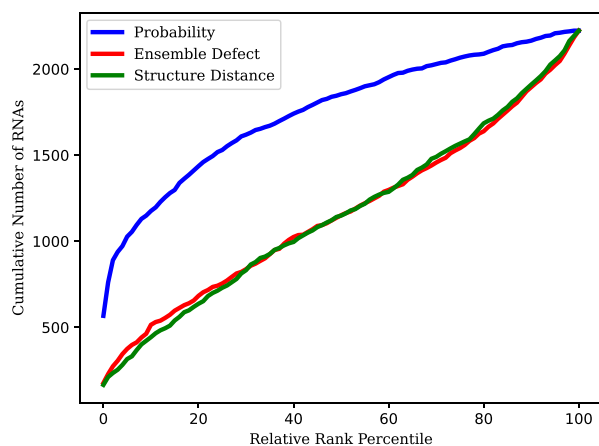


Figure 1. Comparison between probability, ensemble defect, and structure distance on real RNA sequences with known structures. Shows the cumulative number of RNAs for which the true structure (or the closest analog) was at or under a certain percentile when ranked by a fitness function against other structures.

Real RNAs

We consider the ranks of the true structure (or the closest analog) as a percentage. Consider an example percentage of 30.2%. This means, on average, the true structure was ranked at the position 30.2% below the highest rank. Higher ranks and therefore lower percentages are better as they correspond to the true structure being seen more favorably by the fitness function.

Probability achieved a mean rank of 21.2% with a standard deviation of 26.9% and a median of 8.0%. Ensemble defect achieved a mean rank of 48.0% with a standard deviation of 34.5% and a median of 47.8%. Structure distance achieved a mean rank of 48.2% with a standard deviation of 33.6% and a median of 47.9%.

Note that these results contain fewer solutions than on synthetic sequences. This is because the energy model used is imperfect, so natural sequence and structure pairs are not expected to be perfectly recognised as good designs. This is why we look at relative ranks rather than unique MFE solutions.

These results are reflected in Figure 1 where probability dominates the lower percentiles and ensemble defect and structure distance dominates the higher percentiles. It appears as though natural sequences are designed in a way

that correlates with probability, but not ensemble defect or structure distance.

A representative example RNA is considered to illustrate why ensemble defect performs poorly compared to probability on natural RNAs. Consider the tRNA from *Saccharomyces Cerevisiae* (59). Figure 2 shows a small but informative sample of structures from this RNA. The top ranked structures by probability are diverse with some three- and four-way multiloops, including the true structure. The top ranked structures by ensemble defect have low diversity and are all three-way multibranch loops. Ensemble defect tries to find the probability weighted center of the structure space under its distance function. In this case, the ensemble defect structures are a similar distance to the other structure clusters and ensemble defect picks this most central shape rather than the most accurate.

A second example is provided in Figure 3. It is a small SRP (60). This is another case where ensemble defect does poorly. The top ranked structures for probability include some diversity and contain several structures similar to the true structure. The top ranked by ensemble defect have low diversity. In this case, the top structures for ensemble defect are similar to the three-way multiloop structures found by probability. Ensemble defect prefers these structures since they are relatively high probability in the ensemble, and they are also roughly equidistant (by probability weighted defect) from the other two structural clusters (four-way multiloops and no multiloop). Ensemble defect makes a mistake by trying to pick a structure cluster that is central under its distance function, but being central does not appear to be correlated with being correct.

DISCUSSION

Ensemble defect did not perform the best in our experiments. Probability performed better. Dirks *et al.* reported that probability and ensemble defect both performed well (50). In their tests, both solved every RNA design challenge, with other fitness functions falling behind. We see a similar result in our 40nt data set (Table 2), which we expect has a similar difficulty to that of Dirks *et al.* However, the larger data sets (Tables 3 and 4) show probability having much stronger results. The explanation for the differences in our findings is that we tested on harder structures to design. If both fitness functions are solving every design challenge, then the challenges are not hard enough to properly distinguish the fitness functions.

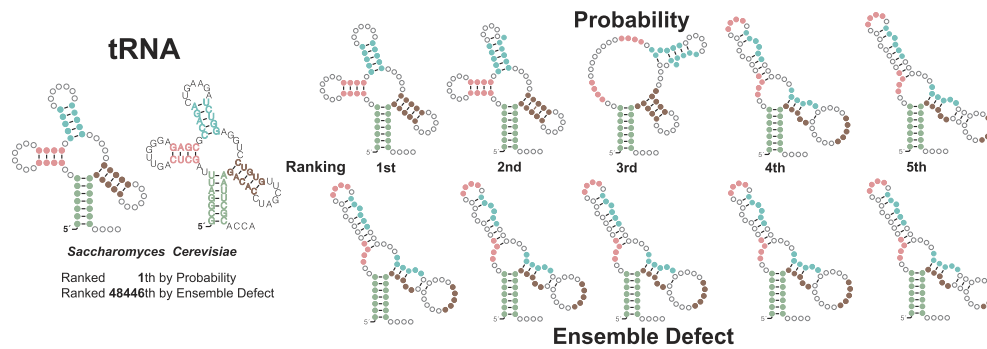


Figure 2. A sample of the structures for a tRNA (*Saccharomyces cerevisiae* $\text{t}_{\text{dbR00000083}}$). Colors correspond to nucleotides in the true structure depicted on the left. The top row is ranked by probability and the bottom row is ranked by ensemble defect.

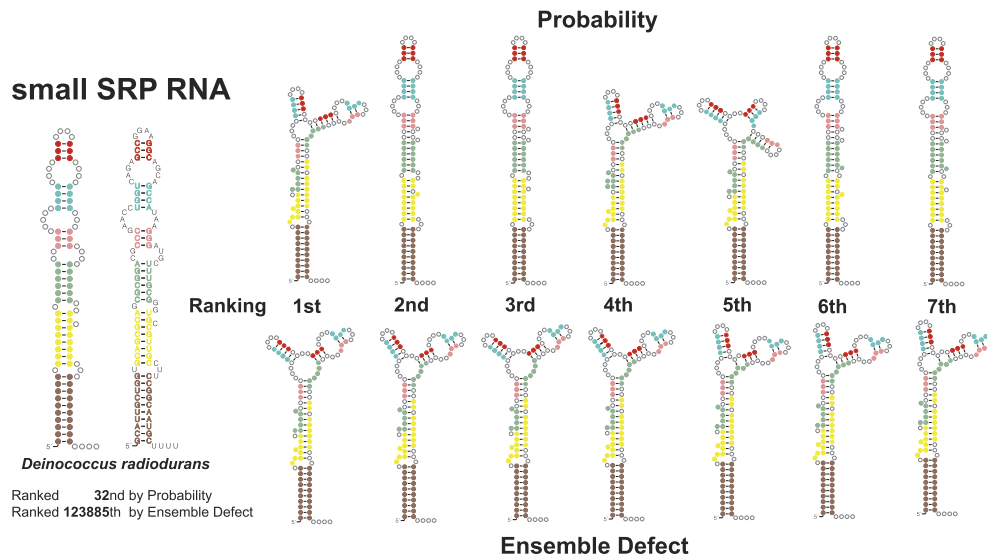


Figure 3. A sample of the structures for a small SRP (*Deinococcus radiodurans* $\text{Dein.radi.}_{\text{AE000513}}$). Colors correspond to nucleotides in the true structure depicted on the left. The top row is ranked by probability and the bottom row is ranked by ensemble defect.

We found the same pattern in real RNAs (Figure 1). Probability outperformed ensemble defect and structure distance. Our results seem to indicate that evolution has designed sequences in a way that correlates with probability, but not with ensemble defect or structure distance. Natural RNA sequences generally have low structural entropy (61,62). Maximising probability eventually minimises entropy, so these observations may be linked.

An examination of specific RNAs where ensemble defect performs worse than probability is revealing. If there is more than one cluster of similar structures that are high probability, ensemble defect will compromise between the two clusters, which can lead to unrealistic averaging as can be seen in Figures 2 and 3. We believe that a better interpretation when there are clusters is that a structure that at the center of any single cluster is likely, but one that is equidistant between two or more clusters is not. Probability captures this intuition.

It is possible that the issues with ensemble defect can be rectified by changing the distance function used (see Equation 1). However, a primary strength of ensemble defect is that it can be efficiently computed. This is because of the intelligent choice of distance function (50). It may be diffi-

cult to achieve such an efficient algorithm with a different distance function.

Tables 2-4 record GC-content. They suggest that free energy has a strong bias, which was expected given GC pairs have the lowest free energy change in the model. We expect to see 50% if the sequence was chosen randomly. Using GC-controlled free energy as a fitness function did improve performance, but only a little. Probability and ensemble defect appear to have a slight bias in GC-content. The increase in GC-content seems to be correlated with increased performance on the test as well as increased difficulty. This may be explained by the thermodynamic model predicting GC pairs are more stabilizing than other pairs. Thus, harder design challenges must rely on using the most stabilizing pairs in some situations, increasing the required GC-content. We note that Dirks *et al.* reported similar GC-contents, albeit slightly elevated, possibly due to their data set having a bias (50). Also, we found that changing the GC-content of the seed sequences for the ARW did not materially change our results (Table 5).

Structure distance appears to be the most widely used fitness function in the literature as we noted previously. The issue of minimum free energy tie breaking is not addressed

in the existing literature. We address it here, and our results suggest that taking the average among the ties has the best result. We tested base pair distance, structural Hamming distance, and Interaction Network Fidelity. We did not observe any significant differences in performance. Overall, no variant of structure distance came close to the level of probability. As such, we recommend that probability is preferred in the future over structure distance despite its popularity.

In Figure 1, we see that structure distance performs similarly to ensemble defect when used to rank real RNAs. However, it is less effective than ensemble defect on synthetic RNAs as seen in Tables 3 and 4. We speculate that it could be the case that the evolutionary pressures on natural sequences are not captured by ensemble defect or structure distance, but ensemble defect is still a better guide than structure distance.

In the work presented here, we examined fitness functions in isolation. However, some existing techniques combine multiple fitness functions (31,39). Future work might test the performance of composite fitness functions.

Probability is the best performing fitness function, but it has a theoretical weakness described previously: in short probabilities taken from different distributions may not be comparable. Future work may show that, in practice, RNA sequences of the same length have similar probability distributions over the corresponding structure ensemble. If this is true, then it would explain why comparing probabilities between different sequences works well in practice. Additionally, a better fitness function may be found that outperforms probability and does not have this issue.

We give three major components comprising an RNA design algorithm: the *computational model*, the *fitness function*, and the *search algorithm*. It is interesting that the choice of computational model informs the space of possible fitness functions. For example, probability and ensemble defect require that the model assigns the ensemble of structures probabilities. However, structure distance only requires a folding function. The choice of fitness function also constrains the search algorithm. To illustrate, (37) uses the properties of ensemble defect to weight mutations. A computational model based on the nearest neighbour thermodynamic model and dynamic programming recursions from (16) is widely assumed in RNA design, but there are alternatives (17). We wonder if using a different paradigm altogether may affect the choice of fitness function and search algorithm.

The fitness function for design must lead the search algorithm toward good solutions. The goodness of those solutions is dependent on the computational model, which also has its own associated fitness function. Therefore, it must be the case that there is some relationship between the two fitness functions. The relationship between fitness functions for structure prediction and fitness functions for design is indirect. For example, most successful folding algorithms minimize free energy (16), however we have demonstrated that doing the same for design is poor. However, some prediction algorithms maximize probability (17), and probability also works well for design as we have demonstrated. The reason for the relationship being complex is likely because the space of sequences is not comparable to the space of structures. In structure prediction, we must select from

the space of all structures. For design, we must select from the space of all sequences, which is much larger. It is not clear that different fitness function have comparable interpretations in both spaces.

DATA AVAILABILITY

All source code and data is available at <https://github.com/maxward/fit-fns-for-rna-design>.

FUNDING

Funding for open access charge: University operating account.

Conflict of interest statement. None declared.

REFERENCES

1. Caprara, M.G. and Nilsen, T.W. (2000) RNA: versatility in form and function. *Nat. Struct. Biol.*, **7**, 831–833.
2. Bernhardt, H.S. (2012) The RNA world hypothesis: the worst theory of the early evolution of life (except for all the others). *Biol. Dir.*, **7**, 23.
3. Crick, F. (1970) Central dogma of molecular biology. *Nature*, **227**, 561–563.
4. Doudna, J.A. and Cech, T.R. (2002) The chemical repertoire of natural ribozymes. *Nature*, **418**, 222–228.
5. Serganov, A. and Patel, D.J. (2007) Ribozymes, riboswitches and beyond: regulation of gene expression without proteins. *Nat. Rev. Genet.*, **8**, 776–790.
6. Cohen, S.B., Graham, M.E., Lovrecz, G.O., Bache, N., Robinson, P.J. and Reddel, R.R. (2007) Protein composition of catalytically active human telomerase from immortal cells. *Science*, **315**, 1850–1853.
7. Tinoco Jr, I. and Bustamante, C. (1999) How RNA folds. *J. Mol. Biol.*, **293**, 271–281.
8. Reese, C.B. (2005) Oligo- and poly-nucleotides: 50 years of chemical synthesis. *Org. Biom. Chem.*, **3**, 3851–3868.
9. Pardi, N., Hogan, M.J., Porter, F.W. and Weissman, D. (2018) mRNA vaccines—a new era in vaccinology. *Nat. Rev. Drug Disc.*, **17**, 261–279.
10. Opalinska, J.B. and Gewirtz, A.M. (2002) Nucleic-acid therapeutics: basic principles and recent applications. *Nat. Rev. Drug Disc.*, **1**, 503–514.
11. Isaacs, F.J., Dwyer, D.J., Ding, C., Pervouchine, D.D., Cantor, C.R. and Collins, J.J. (2004) Engineered riboregulators enable post-transcriptional control of gene expression. *Nat. Biotech.*, **22**, 841–847.
12. Chappell, J., Watters, K.E., Takahashi, M.K. and Lucks, J.B. (2015) A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future. *Curr. Opin. Chem. Biol.*, **28**, 47–56.
13. Lorenz, R., Bernhart, S.H., Höner zu Siederdisen, C., Tafer, H., Flamm, C., Stadler, P.F. and Hofacker, I.L. (2011) ViennaRNA Package 2.0. *Algorithms for Mol. Biol.*, **6**, 26.
14. Reuter, J.S. and Mathews, D.H. (2010) RNAstructure: software for RNA secondary structure prediction and analysis. *BMC Bioinformatics*, **11**, 129.
15. Huang, L., Zhang, H., Deng, D., Zhao, K., Liu, K., Hendrix, D.A. and Mathews, D.H. (2019) LinearFold: linear-time approximate RNA folding by 5'-to-3' dynamic programming and beam search. *Bioinformatics*, **35**, i295–i304.
16. Zuker, M. and Stiegler, P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.
17. Rivas, E. (2013) The four ingredients of single-sequence RNA secondary structure prediction. A unifying perspective. *RNA Biol.*, **10**, 1185–1196.
18. Churkin, A., Retwitzer, M.D., Reinharz, V., Ponty, Y., Waldspühl, J. and Barash, D. (2018) Design of RNAs: comparing programs for inverse RNA folding. *Brief. Bioinform.*, **19**, 350–358.
19. Hofacker, I.L., Fontana, W., Stadler, P.F., Bonhoeffer, L.S., Tacker, M. and Schuster, P. (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.

20. Wachsmuth, M., Findeiß, S., Weissheimer, N., Stadler, P.F. and Mörl, M. (2013) De novo design of a synthetic riboswitch that regulates transcription termination. *Nucleic Acids Res.*, **41**, 2541–2551.
21. Chappell, J., Takahashi, M.K. and Lucks, J.B. (2015) Creating small transcription activating RNAs. *Nat. Chem. Biol.*, **11**, 214–220.
22. Dotu, I., Garcia-Martin, J.A., Slinger, B.L., Mechery, V., Meyer, M.M. and Clote, P. (2014) Complete RNA inverse folding: computational design of functional hammerhead ribozymes. *Nucleic Acids Res.*, **42**, 11752–11762.
23. Wayment-Steele, H.K., Kim, D.S., Choe, C.A., Nicol, J.J., Wellington-Oguri, R., Watkins, A.M., Parra Sperberg, R.A., Huang, P.-S., Participants, E. and Das, R. (2021) Theoretical basis for stabilizing messenger RNA through secondary structure design. *Nucleic Acids Res.*, **49**, 10604–10617.
24. Anderson-Lee, J., Fisker, E., Kosaraju, V., Wu, M., Kong, J., Lee, J., Lee, M., Zada, M., Treuille, A. and Das, R. (2016) Principles for predicting RNA secondary structure design difficulty. *J. Mol. Biol.*, **428**, 748–757.
25. Schnall-Levin, M., Chindelevitch, L. and Berger, B. (2008) Inverting the Viterbi algorithm: an abstract framework for structure design. In: *Proceedings of the 25th International Conference on Machine Learning*. pp. 904–911.
26. Bonnet, É., Rzażewski, P. and Sikora, F. (2020) Designing RNA secondary structures is hard. *J. Comput. Biol.*, **27**, 302–316.
27. Haleš, J., Héliou, A., Maňuch, J., Ponty, Y. and Stacho, L. (2017) Combinatorial RNA design: designability and structure-approximating algorithm in Watson–Crick and Nussinov–Jacobson energy models. *Algorithmica*, **79**, 835–856.
28. Andronescu, M., Fejes, A.P., Hutter, F., Hoos, H.H. and Condon, A. (2004) A new algorithm for RNA secondary structure design. *J. Mol. Biol.*, **336**, 607–624.
29. Busch, A. and Backofen, R. (2006) INFO-RNA—a fast approach to inverse RNA folding. *Bioinformatics*, **22**, 1823–1831.
30. Taneda, A. (2011) MODENA: a multi-objective RNA inverse folding. *Adv. Appl. Bioinform. Chem.*, **4**, 1.
31. Lyngso, R.B., Anderson, J.W., Sizikova, E., Badugu, A., Hyland, T. and Hein, J. (2012) Frnakenstein: multiple target inverse RNA folding. *BMC Bioinformatics*, **13**, 260.
32. Rubio-Largo, Á., Vanneschi, L., Castelli, M. and Vega-Rodríguez, M.A. (2018) Multiobjective metaheuristic to design RNA sequences. *IEEE Trans. Evol. Comput.*, **23**, 156–169.
33. Garcia-Martin, J.A., Clote, P. and Dotu, I. (2013) RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. *J. Bioinform. Comput. Biol.*, **11**, 1350001.
34. Minuesa, G., Alsina, C., Garcia-Martin, J.A., Oliveros, J.C. and Dotu, I. (2021) MoiRNAiFold: a novel tool for complex in silico RNA design. *Nucleic Acids Res.*, **49**, 4934–4943.
35. Yao, H.-T., Waldspühl, J., Ponty, Y. and Will, S. (2021) Taming disruptive base pairs to reconcile positive and negative structural design of RNA. In: *RECOMB 2021-25th International Conference on Research in Computational Molecular Biology*.
36. Sav, S., Hampson, D.J. and Tsang, H.H. (2016) SIMARD: A simulated annealing based RNA design algorithm with quality pre-selection strategies. In: *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, pp. 1–8.
37. Zadeh, J.N., Wolfe, B.R. and Pierce, N.A. (2011) Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem.*, **32**, 439–452.
38. Bellaousov, S., Kayedkhordeh, M., Peterson, R.J. and Mathews, D.H. (2018) Accelerated RNA secondary structure design using preselected sequences for helices and loops. *RNA*, **24**, 1555–1567.
39. Portela, F. (2018) An unexpectedly effective Monte Carlo technique for the RNA inverse folding problem. bioRxiv doi: <https://doi.org/10.1101/345587>, 14 July 2018, preprint: not peer reviewed.
40. Yang, X., Yoshizoe, K., Taneda, A. and Tsuda, K. (2017) RNA inverse folding using Monte Carlo tree search. *BMC Bioinformatics*, **18**, 468.
41. Cazenave, T. and Fournier, T. (2020) Monte Carlo inverse folding. In: *Monte Carlo Search International Workshop*. Springer, pp. 84–99.
42. Levin, A., Lis, M., Ponty, Y., O'donnell, C.W., Devadas, S., Berger, B. and Waldspühl, J. (2012) A global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Res.*, **40**, 10041–10052.
43. Reinharz, V., Ponty, Y. and Waldspühl, J. (2013) A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics*, **29**, i308–i315.
44. Kleinkauf, R., Mann, M. and Backofen, R. (2015) antaRNA: ant colony-based RNA sequence design. *Bioinformatics*, **31**, 3114–3121.
45. Lee, J., Kladwang, W., Lee, M., Cantu, D., Azizyan, M., Kim, H., Limpaecher, A., Gaikwad, S., Yoon, S., Treuille, A. et al. (2014) RNA design rules from a massive open laboratory. *Proc. Natl. Acad. Sci. U.S.A.*, **111**, 2122–2127.
46. Koodli, R.V., Keep, B., Coppess, K.R., Portela, F., participants, E. and Das, R. (2019) EternaBrain: Automated RNA design through move sets and strategies from an Internet-scale RNA videogame. *PLoS Comput. Biol.*, **15**, e1007059.
47. Eastman, P., Shi, J., Ramsundar, B. and Pande, V.S. (2018) Solving the RNA design problem with reinforcement learning. *PLoS Comput. Biol.*, **14**, e1006176.
48. Runge, F., Stoll, D., Falkner, S. and Hutter, F. (2018) Learning to design RNA. arXiv doi: <https://arxiv.org/abs/1812.11951>, 12 April 2019, preprint: not peer reviewed.
49. Turner, D.H. and Mathews, D.H. (2010) NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Res.*, **38**, D280–D282.
50. Dirks, R.M., Lin, M., Winfree, E. and Pierce, N.A. (2004) Paradigms for computational nucleic acid design. *Nucleic Acids Res.*, **32**, 1392–1403.
51. Lyngso, R.B., Zuker, M. and Pedersen, C.N. (1999) Internal loops in RNA secondary structure prediction. In: *Proceedings of the Third Annual International Conference on Computational Molecular Biology*. pp. 260–267.
52. Gorodkin, J., Stricklin, S.L. and Stormo, G.D. (2001) Discovering common stem–loop motifs in unaligned RNA sequences. *Nucleic Acids Res.*, **29**, 2135–2144.
53. Parisien, M., Cruz, J.A., Westhof, É. and Major, F. (2009) New metrics for comparing and assessing discrepancies between RNA 3D structures and models. *RNA*, **15**, 1875–1885.
54. Dadkhahi, H., Rios, J., Shanmugam, K. and Das, P. (2022) Fourier representations for black-box optimization over categorical variables. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. **36**, pp. 10156–10165.
55. McCaskill, J.S. (1990) The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers: Orig. Res. Biom.*, **29**, 1105–1119.
56. Zadeh, J.N., Steenberg, C.D., Bois, J.S., Wolfe, B.R., Pierce, M.B., Khan, A.R., Dirks, R.M. and Pierce, N.A. (2011) NUPACK: Analysis and design of nucleic acid systems. *J. Comput. Chem.*, **32**, 170–173.
57. Wuchty, S., Fontana, W., Hofacker, I.L. and Schuster, P. (1999) Complete suboptimal folding of RNA and the stability of secondary structures. *Biopolymers: Orig. Res. Biom.*, **49**, 145–165.
58. Ward, M., Datta, A., Wise, M. and Mathews, D.H. (2017) Advanced multi-loop algorithms for RNA secondary structure prediction reveal that the simplest model is best. *Nucleic Acids Res.*, **45**, 8541–8550.
59. Jühling, F., Mörl, M., Hartmann, R.K., Sprinzl, M., Stadler, P.F. and Pütz, J. (2009) tRNADB 2009: compilation of tRNA sequences and tRNA genes. *Nucleic Acids Res.*, **37**, D159–D162.
60. Rosenblad, M.A., Gorodkin, J., Knudsen, B., Zwieb, C. and Samuelsson, T. (2003) SRPDB: signal recognition particle database. *Nucleic Acids Res.*, **31**, 363–364.
61. Dethoff, E.A. and Weeks, K.M. (2019) Effects of refolding on large-scale RNA structure. *Biochemistry*, **58**, 3069–3077.
62. Rice, G.M., Shivashankar, V., Ma, E.J., Baryza, J.L. and Nutiu, R. (2020) Functional atlas of primary miRNA maturation by the microprocessor. *Mol. Cell*, **80**, 892–902.