A simple evolutionary algorithm guided by local mutations for an efficient RNA design.

Nono S. C. Merleau*

nonosaha@mis.mpg.de

Max Planck Institute for Mathematics in the Sciences
Leipzig, Saxony, Germany

Matteo Smerlak

smerlak@mis.mpg.de

Max Planck Institute for Mathematics in the Sciences
Leipzig, Saxony, Germany

ABSTRACT

Inverse folding an RNA secondary structure of length L can be understood as searching in space $S = \{A, C, G, U\}^L$, one or many RNA sequence(s) that fold(s) into that structure. Solving this problem is a prerequisite for RNA design and is still a real challenge in bio-engineering. To improve the RNA inverse folding, we investigate an evolutionary algorithm implementing local mutations taking into account the nucleotide and canonical base pair distributions. We implement our algorithm as a simple open-source python program called aRNAque¹. The empirical results show that aRNAque can solve some target structures previously unsolvable by the existing evolutionary algorithms benchmarked in this work. On the two benchmark datasets, $\approx 72\%$ of the EteRNA100 dataset (respectively $\approx 92\%$ using the Turner1999 energy parameter sets) and $\approx 87\%$ of the non-EteRNA100 dataset and the designed sequences are of better quality. In comparison to existing tools, our EA implementation also shows a significant computational time improvement.

CCS CONCEPTS

• **Mathematics of computing** → **Discrete mathematics**; • **Computing methodologies** → **Discrete space search**.

KEYWORDS

RNA design, inverse RNA folding, Evolutionary Algorithm (EA).

ACM Reference Format:

Nono S. C. Merleau and Matteo Smerlak. 2021. A simple evolutionary algorithm guided by local mutations for an efficient RNA design.. In *2021 Genetic and Evolutionary Computation Conference (GECCO '21)*, July 10–14, 2021, Lille, France. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3449639.3459280>

1 INTRODUCTION

An RNA (or ribonucleic acid) molecule is a polymer chain over a four-letter alphabet. Each letter represents a particular nucleotide base: A for adenine, U for uracil, C for cytosine, G for guanine, and RNAs play essential roles in all cellular processes (information storage, regulation, and catalysis). For an RNA molecule to perform

¹The source code is publicly available at <https://github.com/strevol-mpi-mis/aRNAque.git>

a biological function, it has to fold in a three-dimensional structure which strongly depends on its secondary structure. In nature, RNA folding is thought to be hierarchical [2, 20] where a stable secondary structure forms first and subsequently there is a refinement to the tertiary one. Here we consider only the first relationship where an RNA sequence folds into a stable secondary structure.

More formally, an RNA molecule consists of an ordered sequence of nucleotides that can be represented as:

$$x = (x_1, \dots, x_L) \text{ where } x_i \in \{A, C, G, U\}$$

x is often known as the primary structure of RNA. For simplicity, a secondary structure of such sequence x is a list of base pairs (i, j) on x with the following constraints:

- (1) A nucleotide (sequence position) can only belong to a single pair,
- (2) No pseudoknots: No pairs (i, j) and (k, l) with $i < k < j < l$,
- (3) If (i, j) is a pair then $x_i x_j \in \{GC, CG, AU, UA, GU, UG\}$,
- (4) If (i, j) is a base pair, then $j - i > 3$.

Even though the pseudoknots occur in RNA structures, the above definition classifies them as tertiary structure motifs and the constraints ((1), (2)) exclude any tertiary structure motifs; this is because the RNA folding tool we use in this work does not deal with pseudoknots.

There are several graphical ways to represent such secondary structure. Figure 1 presents four representations mostly used, but we can also have tree, mountain and forest representations.

Figure 1: Graphical representation. (A) Dot-bracket representation where each base pair corresponds to the open-close parentheses and the dot correspond to unpaired nucleotide. (B) Circular or circle representation. (C) Conventional secondary structure graph representation of planar representation. (D) linear representation.

This work is licensed under a Creative Commons Attribution International 4.0 License.
GECCO '21, July 10–14, 2021, Lille, France
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8350-9/21/07.
<https://doi.org/10.1145/3449639.3459280>

Figure 2: Different types of planar secondary structure loops. (A) **Stacking pair:** interior loop of size 0. (B) **Hairpin:** loop of degree 1. (C) **Multi-loop:** loop with degree > 2 . (D) **Interior loop:** loop with degree 2. (E) **Bulge:** a special case of interior loop in which there are unpaired bases only on one side. (F) **Exterior loop:** all the positions are not interior of any pair.

The secondary structures can be uniquely decomposed into *loops* where loops are the faces of the planar drawing of the structure [11]. Each loop is characterized by its length (the number of unpaired nucleotides in the loop) and its degree (the number of base pairs delimiting the loop, including the closing pairs). Figure 2 presents the six types of loops. The loop decomposition allows building the basis of the standard energy model (on which the folding tool we are using in this work is based) for RNA secondary structures. The total free energy of a structure is assumed to be a sum over its constituent loops' energies. The model is often called the *nearest neighbor* model [21].

For a given sequence x of length L , let Γ be the secondary structural ensemble of x . The stable RNA secondary structure of that ensemble is often the one with the minimum free energy. We write:

$$s^{MFE}(x) = \arg \min_{s \in \Gamma} E(x, s) \quad (1)$$

where Γ is the ensemble of secondary structures of x and $E(x, s)$ the free energy.

Advanced tools for the free energy minimization to predict RNA secondary structures have been gradually developed, and most of them rely on empirical (or experimental) data such as energy parameters [14, 21, 23]. The work in this paper makes use of the ViennaRNA folding package [13] since it deals with the structure conformation considered in this work and, it is the most accurate one to the best of our knowledge.

Neutral Network. The size of the RNA structural ensemble has been analytically computed through tools developed by Stein and Waterman [18], and it yields an upper bound of $S_L \approx 1.48 \times L^{-\frac{3}{2}} 1.85^L$ structure vis-a-vis 4^L sequences. Compared to the total number of sequences, the number of structures is much smaller, which means there is indeed a high possibility that many sequences fold into the same structure. In case that happens, the set of those sequences are said to be neutral. Therefore, a neutral network of RNA sequences

Figure 3: A minimum free energy secondary structure produced using ViennaRNA package 2.1.4. (A) **The standard graph representation with the different base pair interactions.** (B) **The base pair matrix probabilities.**

is a graph in which vertices are neutral sequences and, two vertices are adjacent when they differ by a single nucleotide.

Inverse RNA folding problem. Unlike the folding problem, here we consider the inverse folding problem for which the goal is to find one or many RNA sequences that fold into a given target RNA secondary structure. In other terms, it is an optimization problem where a target RNA secondary structure is given, and the goal is to determine a set of RNA sequences for which the distance between their corresponding 'mfe' secondary structures and the target one is minimized. In the field of modern bioengineering, to design an RNA molecule that performs a specific function, one must first solve the RNA inverse folding problem [8, 9, 22].

Related works. Several methods or algorithms addressing this problem have been proposed in the literature. Most of them perform a stochastic search optimization where initial potential solutions are generated and, refined over a finite number of iterations or generations [4–6, 15, 19]. Among the existing tools, both ERD and MODENA are evolutionary algorithms but implementing different strategies. MODENA uses a multi-objective function that measures the stability of the folded sequence and its similarities to the target. It starts from a population of randomly generated sequences, and the objective is optimized through tournament selection and random mutation at non-closing loop positions. In contrast, ERD starts by decomposing the target structure into loops and independently uses an evolutionary algorithm to minimize each constituent's energy. Finally, it recombines the different fragments to form the designed sequence. Some algorithms are performing different strategies from the population-based approach: sentRNA [17] is a computational agent that uses a set of information and strategies collected from the EteRNA game players to train a neural network model that will be able to design a set of sequences for a given target structure. NEMO [15] is a recently developed tool, combining a Nested Monte Carlo Search (NMCS) technique with domain-specific knowledge to create a novel algorithm. It surpasses all the existing tools on the EteRNA100 benchmark datasets by solving $\approx 95\%$ of the targets using the Turner1999 energy parameter sets. antaRNA [10] utilizes an

“ant-colony” optimization, in which an initial sequence is generated via a weighted random search, and the *fitness* of that sequence is then used to refine the weights and improve subsequences over generations. NUPACK [24] uses a tree decomposition technique and the ensemble defect as objective function to design qualitatively good sequences. IncaRNA [16] generates a GC-weighted partition function for the target structure, and then adaptively samples sequences from it to match the desired GC-content. RNAinverse [13], one of the oldest inverse folding tool included in the ViennaRNA package. It uses a Monte Carlo Search technique in the space of compatible sequences to find a set of sequences that fold into a given target structure. Finally, RNAiFold [7] employs constraint programming that exhaustively searches over all possible sequences compatible with a given target.

Benchmarks. In general, to evaluate the RNA design tools’ performance, we use benchmark datasets [3]. Two benchmark datasets are mostly used in the literature—(i) Rfam²: a collection of RNA families, each represented by multiple sequence alignments, consensus secondary structures and covariance models—(ii) EteRNA100: a collection of hundred RNA secondary structures extracted from the EteRNA Puzzle game³. The EteRNA game is built on an old version of ViennaRNA package that used the Turner1999 energy parameter sets [21]. When the benchmark datasets rely on a particular energy parameter set, the performance of a given inverse RNA folding tool evaluated on these datasets will also be related to the choice of the RNA folding tool’s energy parameter set. In case the benchmark datasets do not rely on a particular energy parameter set, the robustness of the inverse RNA tool will be its capability to perform well on different energy parameter sets.

Limits of existing methods. The existing approaches have some advantages and disadvantages, depending on the techniques implemented. NUPACK for example—despite its well-defined objective function—still has difficulty designing sequences for large targets and most of the EteRNA100 targets. In contrast, ERD because of its strong decomposition method which allows to deal quickly with large targets (On Rfam 1.0 with target’s length between 400 – 1400) but still a big challenge to solve more than 65% of the EteRNA100 using the Turner2004 energy parameter sets. On another side, NEMO, one of the most recent tools, can solve more than 90% of the EteRNA100 dataset using an old version of ViennaRNA package, which is based on Turner1999 energy parameter sets [21]. The sentRNA’s machine learning model also relied on the same old version of ViennaRNA package and, by adding a refinement on the machine learning model, sentRNA solves 78% of EteRNA100. Without this refinement, sentRNA can only solve 48% of EteRNA100’s targets, which is also a clear limitation. For the EAs ERD and MODENA, none of them can solve more than 65% of EteRNA100 using the Turner2004 energy parameter sets.

Contribution. In this work, we investigate the choice of the selection function and a mutation mode for which an EA’s performance can be significantly improved. As a result, we propose a simple evolutionary algorithm implementing most of the objective functions used by previous tools as a selection function and

a local mutation scheme considering the nucleotide and base-pair distributions. We also propose a refinement of the sequences’ ensemble defect found through a neutral walk in sequence space. Finally, benchmarking our method on existing benchmark datasets shows apparent robustness to both Turner1999 and Turner2004 energy parameters. On EteRNA100, we obtain 72% of success when using the Turner2004 energy parameter set for the RNA folding (respectively 92% when using the Turner1999 energy parameter sets) and 87% on the 63 targets non-EteRNA datasets using the Turner2004 energy parameter sets.

In the first part of our work, we describe the steps of our evolutionary algorithm. In the second part, we describe the experimental setup and the benchmark datasets. Finally, we present our algorithm’s performance on existing benchmark datasets, and we compare its performance to existing methods in the literature.

2 THE PROPOSED ALGORITHM BASED ON EVOLUTIONARY COMPUTATION

2.1 Objective functions

Our EA reaches its performance through the minimization of three objective functions:

- Hamming distance from the target structure: since the main goal of the inverse folding problem is to find sequences that fold into a given target secondary structure σ^* , the simple fitness measurement f of an RNA sequence ϕ can be defined as follows:

$$f(\phi, \sigma^*) = \frac{1}{1 + h(s^{MFE}(\phi), \sigma^*)} \quad (2)$$

where $h(\cdot, \cdot)$ is the hamming distance on the structure space (structure are in dot and bracket representation)

- Normalized Energy Distance (NED): the difference between the energy of a given sequence ϕ evaluated to fold into a target structure σ^* and the minimum free energy of the sequence in its structural ensemble Γ . The value is normalized over all the sequences in a given population P .

$$NED(\phi, \sigma^*) = [1 - \Delta E_{norm}(\phi, \sigma^*)]^P \quad \forall p > 1 \quad (3)$$

where,

$$\Delta E_{norm}(\phi, \sigma^*) = \frac{\Delta E(\phi, \sigma^*)}{\sum_{\phi \in P} \Delta E(\phi, \sigma^*)} \quad (4)$$

and,

$$\Delta E(\phi, \sigma^*) = E(\phi, \sigma^*) - \arg \min_{s \in \Gamma} E(\phi, s) \quad (5)$$

- Ensemble defect (ED) [24]: Here, we use the ED as a second objective function for refinement after having at least one sequence that folds into the target in the current population. It is defined as follows:

$$\begin{aligned} ED(\phi, \sigma^*) &= \sum_{\sigma \in \Gamma} p(\phi, \sigma) d(\sigma, \sigma^*) \\ &= L - \sum_{1 \leq i, j \leq L} P_{i,j}(\phi) S_{i,j}(\sigma^*) \end{aligned} \quad (6)$$

²The Rfam database <https://rfam.xfam.org/>

³The EteRNA game <https://eternagame.org/>

where $P_{i,j}$ is the base pair probability matrix and $S(s)$ is the structure matrix with entries $S_{i,j} \in \{0, 1\}$. If the structure contains pair $\{i, j\}$, then $S_{i,j}(s) = 1$ otherwise $S_{i,j}(s) = 0$.

To minimize the *NED* (Eq. (3)) and the hamming distance (or maximize the fitness (Eq. (2)) of a population of RNA sequences, instead of combining both measurements to form a multi-objective function, we use them separately at a different level of our EA. We use the *NED* as a selection weight for the sequences that will be mutated, and the hamming distance is used as a weight to elite ten best sequences that will always move to the next generation. Therefore the selection method we use is the *fitness proportionate selection*, also known as roulette wheel selection [12]. Once we have at least one sequence that folds into the given target in the current population (for the successful case), we walk through its neutral network by minimizing the ensemble defect function (Eq. (6)). This ED minimization is simply performed by replacing the *NED* objective function by the *ED* function the EA (Step 4).

2.2 Mutation operator

To explore the search space of compatible sequences (sequences with canonical base pairs at the corresponding open and closed bracket positions) efficiently, we introduce a mutation step that depends on the nucleotide and canonical base pair probability distribution.

Let $N_{ucl.} = \{A, U, C, G\}$ be the set of nucleotides weighted respectively by the probabilities:

$$P_{nucl.} = \{P_A, P_U, P_C, P_G\}$$

and $C_{bp} = \{AU, UA, CG, GC, UG, GU\}$ be the set of canonical base pairs weighted respectively by the probabilities:

$$P_{can.} = \{P_{AU}, P_{UA}, P_{CG}, P_{GC}, P_{UG}, P_{GU}\}$$

where

$$\sum P_N = 1, \sum P_C = 1$$

and

$$P_{AU} = P_{UA}, P_{UG} = P_{GU}, P_{CG} = P_{GC}$$

Our evolutionary algorithm relies on the flexibility of the mutation rate parameters. It allows the user to explicitly set the design constraints through the mutation parameters P_C and P_N , which allow the control of the nucleotide distribution during the evolutionary process.

Figure 4 shows an illustration of the mutation algorithm.

2.3 The algorithm

For a given population size N and a target structure σ^* of length l , an initial population P is generated randomly as follows:

- (1) Select randomly l nucleotides in $N_{ucl.}$.
- (2) Identify the base pair position (i, j) in the random sequence, select randomly a base pair in the set of canonical base pairs C_{bp} and fix the first nucleotide of the selected canonical base pair at the position i and the second at position j .
- (3) Repeat 2. for all base pair positions in the target structure
- (4) Repeat 1. 2. and 3 N -times.

Let T be the maximum number of generations and F_t the set of all sequences found at a given time t . After the initial population of

Figure 4: Mutation step illustration. (a) is a given target structure and (b) is a random compatible sequence from a population of RNA sequences. (c) is the mutated sequence where the non-base pair positions (in black color) are mutated independently of the base pair positions. One non-base pair position (3) and two base pair positions $\{(2, 7); (11, 16)\}$ are mutated.

RNA sequences is generated, our algorithm is performed as follows:

While $(t < T)$ or $|F_t| < n_f$ do:

- Evaluate the fitness (Eq. (2)) of every sequence $s \in P$
- Add to F_t all the sequences with fitness equal to 1
- Choose 10 best sequences of P based on their fitness (Eq. (2)) values and add them to P' .
- Sample N sequences with respect to their *NED* (Eq. (3)) values and store them in S .
- For each RNA sequence $s \in S$:
 - (1) $s' = \text{copy}(s)$
 - (2) Select a random nucleotide r_n in $N_{ucl.}$ respect to the probabilities $P_{nucl.}$.
 - (3) $s'[i] = r_n$, where i is a random integer chosen uniformly from 0 to $l - 1$
 - (4) Select a random base pair b_p in C_{bp} respect to the probabilities $P_{can.}$.
 - (5) $s'[i] \leftarrow b_p[0]$ and $s'[j] \leftarrow b_p[1]$, where (i, j) is a random tuple of integers chosen uniformly from the set of base pair positions.
 - (6) Add s' in P'
- $P \leftarrow P'$
- $t \leftarrow t + 1$

3 EXPERIMENTAL SETUP AND BENCHMARK DATA

To estimate our EA's performance, we use two different benchmark datasets and compare the results to several existing algorithms. The two datasets were recently used by the deep learning tool SentRNA [17] and most of the algorithms we mentioned earlier:

- The non-EteRNA dataset: a set of 63 experimentally synthesized targets that Garcia-Martin et al. [7] recently used to

benchmark a set of ten inverse folding algorithms, which from our knowledge, is the most recent and comprehensive benchmark of current state-of-the-art methods. The dataset is collected from 3 sources: the first dataset called **dataset A** which contains 29 targets collected from Rfam and also used in [5, 19] and the second called **dataset B** is a collection of 24 targets used in [5] and added to that the 10 structures used in [17].

- The EteRNA100: it contains a set of 100 target structures extracted from the EteRNA puzzle game and classified by their degree of difficulty. The EteRNA puzzle game was designed using ViennaRNA 1.8.5. which relies on Turner1999 energy parameters.

The success rate measurement considered here is simply the number of designed sequences folding into the target secondary structure over the total number of designed sequences.

In the experiments carried out, we used the python implementation of our EA aRNAque which depends on ViennaRNA Package 2.0 [13]. The package supports different energy parameters, and each of them may change the energy landscape completely and make the design problem harder for some targets or easier for others.

For the non-EteRNA dataset, only the Turner2004 energy parameters were used. The maximum number of generations was set to be 5000. The mutation parameters (P_C and P_N) were chosen to be close to the nucleotide distribution of the RNA sequence in nature [5]. For the EteRNA100, two benchmarks have been carried out: the first one using the Turner1999 energy parameters and a second one using the Turner2004 energy parameters.

The benchmark datasets, the code source, and all the necessary scripts to reproduce the results presented in this work are available on aRNAque's github repository ⁴.

4 EXPERIMENTAL RESULTS

4.1 Performance on non-EteRNA100 dataset

For the first benchmark, we use the non-EteRNA dataset. Compared to other tools, the statistical results are presented in Table 1.

The results show that our method surpasses 8/10 of other tools. ERD solved 2 more targets than our method because of its strong decomposition capacity, which allows it to solve the entire **dataset B**. With the advantage that our evolutionary algorithm also allows us to fit the nucleotide distribution parameters taken from natural RNA directly in the mutation parameters, we can solve 21/24 targets from the **dataset B**. For the **dataset A** we solve 24/29 targets which means 2 more than the previous tools, and for the 10 last targets we solve 7 targets. Adding all these solved targets together, we obtain a result of 52/63 as presented in Table 1.

4.2 Performance on EteRNA100 dataset

In this second benchmark, we compare our EA's performance on the EteRNA100 benchmark datasets. For each target structure in the dataset, 5 sequences are designed. For each tool benchmarked,

Table 1: Summary of performance of aRNAque vs the 10 other algorithms benchmarked on the non-EteRNA100 by Anderson-Lee et al. [1]

Methods	Number of puzzles solved
SentRNA, NN + full moveset	57/63
ERD	54/63
SentRNA, NN + GC pairing	53/63
SentRNA, NN + All pairing	53/63
aRNAque	52/63
RNA-SSD	47/63
SentRNA, NN only	46/63
INFO-RNA	45/63
MODENA	32/63
NUPACK	29/63
IncaRNAation	28/63
Frnakenstein	27/63
RNAinverse	20/63
RNAfbinv	0/63

Table 2: Summary of performance of aRNAque vs the 7 other algorithms benchmarked on EteRNA100 by Anderson-Lee et al. [1]

Methods	Number of puzzles solved
aRNAque	72/100
RNAinverse	66/100
Learna	66/100
ERD	65/100
SentRNA, NN + full moveset	60/100
MODENA	54/100
NEMO	50/100
INFO-RNA	50/100
NUPACK	48/100
DSS-Opt	47/100
RNA-SSD	27/100

if at least one sequence out of 5 folds into the corresponding target secondary structure, the target is said to be successfully solved by the tool. Therefore, each tool's success rate is the number of solved targets over the dataset's total number of targets.

- Benchmark using energy parameters Turner2004.

In Table 2, we have benchmarked four tools (SentRNA, RNAinverse, NEMO, and ERD) and the rest of the results are taken from [17].

We can notice in Table 2 that the evolution algorithm we propose can solve $\approx 72\%$ of the dataset, and it surpasses the 4 methods we benchmarked and all the tools already benchmarked in [17]. We can also solve approximately 23 targets more than NUPACK which is also minimizing the ensemble defect and that shows the importance of population-based algorithm. Compared to the existing GA-based algorithms, our EA can solve approximately 18 targets more than MODENA and 7 targets more than ERD.

- Benchmark using energy parameters Turner1999.

⁴The source code is publicly available at <https://github.com/strevol-mpi-mis/aRNAque>. git

Most of the recent tools used the Turner1999 energy parameters while performing their benchmark on the Eterna100 dataset. To compare our benchmarking results to the one in the literature, we found it worthwhile to perform the second benchmark using the Turner1999 energy parameter sets. With this configuration and using the same setting as in the first benchmark, we can solve $\approx 92\%$ of the dataset, which allows our method to stand behind NEMO which is the only one solving 95% of the dataset while using the Turner1999 energy parameters. This may explain why NEMO's performance using a different version of ViennaRNA was worse and, in contrast, demonstrates the capacity of our method to keep its performance even when the energy parameters of the RNA folding change.

4.3 Quality of the designed sequences

Designing RNA molecules does not only require solving the inverse folding, but it also requires finding qualitatively good sequences. To assess the quality of the sequences we designed in this work, we use two different metrics: the minimum free energy for the folded sequence's stability at the equilibrium and the ensemble defect for the structural ensemble quality.

• NUPACK vs. aRNAque on Rfam.

Since another essential aspect of our method is to allow ED's refinement, we also need to check how well our approach minimizes the ensemble defect function and the free energy. Among the existing tools to which we compare our method, only NUPACK minimizes the ensemble defect function (Eq. (6)). For that reason, we decided to compare the ensemble defect of the 30 first solved structures to the one found by NUPACK. Figure 5 shows the median ED and MFE over 5 runs for each of the targets wherein blue is our method and in red is NUPACK, we can notice our EA (named aRNAque) designs sequences with lower ensemble defect and minimum free energy than NUPACK.

Figure 5: The first 30 Rfam sequences designed using NUPACK and aRNAque. (A) The median ensemble defect of the 5 sequences designed by the two tools for each Rfam target structure. (B) The median minimum free energy of the 5 sequences designed by the two tools for each Rfam target structures. (C) The distributions of hamming distances between the MFE structures and the different target structures.

• RNAinverse, sentRNA, ERD vs. aRNAque on Eterna100.

To assess the quality of the sequences designed by each of the tools, the ensemble defect and minimum free energy were computed. Figure 6 shows the Hamming distance distributions of the MFE structures (only for the first 35 Eterna targets) from different targets and their corresponding ensemble and minimum free energy medians. The percentage of success and the quality (in terms of ED and MFE) of our designed sequences surpasses all of the other methods, except in few cases where the MFEs of RNAinverse's sequences are lower but, the hamming distance distributions show that most of the folded sequences are far from the targets.

Figure 6: The first 35 Eterna100 sequences designed using RNAinverse, SentRNA, ERD and aRNAque. (A) The median ensemble defect of the 5 sequences designed by the four tools for each Eterna100 target structure. (B) The median minimum free energy of the 5 sequences designed by the four tools for each Eterna100 target structure. (C) The hamming distance distribution of the MFE structures from different target structures.

4.4 Complexity and CPU time comparison

An evolutionary algorithm's complexity is often considered to be the number of fitness evaluations needed to solve a given problem. That complexity depends on the number of generations and the population size parameters. For our evolutionary algorithm, we have three objective functions. Therefore, in the worst case the number of evaluation needed is:

$$f(T, t, N) = (2T + t) \times N$$

Where T is the number of generation need to get at least one sequence that folds into the target and t the number of generations for performing the ensemble defect refinement.

4.4.1 Accuracy and speed comparisons on the dataset A. After benchmarking our algorithm on existing datasets and comparing the results obtained to previous tools, we also deemed it is important to address our methods' accuracy and speed comparisons. Here we present time and accuracy comparison on **dataset A**. The formula

used to compute $E_t(s)$ is the same used in [5] and it is given by:

$$E_t(s) = \frac{\text{Total Execution Time}}{Sc}$$

In Table 3, our results are compared to the benchmark performed in [5] and the best results are indicated in bold. As we can see, aRNAque solves 2 targets (**RF00009** and **RF00028**) previously unsolved by INFO-RNA and MODENA, and 2 other targets unsolved by ERD (**RF00030** and **RF00018**). This result allows our algorithm to obtain the best accuracy over all other existing algorithms found in the literature. We can notice that even though for a very few cases we have gotten better speed, the computational time is much more expensive than ERD and INFO-RNA. This can be understood by the fact that when we use a selection function different from the fitness function, we evaluate every sequence twice at each generation, which consequently increases our algorithm's computation time.

4.4.2 CPU time comparison: RNAinverse vs. aRNAque on Eterna100.

Since our previous benchmarks on Eterna100 using the Turner2004 energy's parameters reveal that RNAinverse, one of the oldest inverse folding tools, stands behind aRNAque solving 66% of the dataset; we also compare its computational time to our implementation. Figure 7 shows the CPU time in seconds needed to design for each target in the Eterna100, 5 sequences. As the RNAinverse time increases exponentially with the length of the target, the aRNAque one does not.

Figure 7: CPU time: RNAinverse vs. aRNAque. Each bubble corresponds to a target structure in Eterna100 dataset and, their colours are proportional to the length of the targets. In the legend, MHD stands for Median Hamming distance, and the different markers represent—('o') 100% success for both tools—('+') 100% success for aRNAque and not for RNAinverse—('-') for the case both tools fail to find at least one sequence that folds into the target. Underlying the CPU time difference is the inside plot that shows the CPU time (in seconds) as a target length function.

5 CONCLUSION

We investigated an evolutionary approach to improve the existing solutions to the RNA inverse folding problem in this work. As a discovery, we proposed a new EA python tool called aRNAque. The

benchmark results showed that our program performs better than the existing evolutionary algorithms in the literature. It surpassed 10 other methods benchmarked by us and ER et al.[17] on the Eterna100 dataset using the most recent ViennaRNA package first and in the dataset of 29 biological target structures used to measure the algorithm's performance of previous existing algorithms second. The benchmark on Eterna100 also revealed that taking into account the folding energy parameter can shift the difficulty of designing some targets from hard to easier. Finally our results showed significant improvements in CPU time and in sequence quality.

Despite the simplicity and the advantages of the RNA design tool we presented in this work, aRNAque still presents also some limitations. First, it will be interesting to consider the secondary structures with pseudoknots interactions to see if the algorithm will still perform well and make it more useful for real-life applications. Secondly, the benchmarks carried out in this work revealed that the existing programs are often based on different folding algorithm's energy parameters, making their results not comparable to aRNAque in some cases. Some tools like RNAinverse, SentRNA, and NEMO seem to show different performance compared to the one found in the literature [1, 3]. To overcome the limitations mentioned above, we propose building a new benchmark dataset that does not always rely on energy parameters and re-benchmark existing tools using this dataset and the different energy parameter sets.

6 ACKNOWLEDGMENTS

We thank the Structure of the Evolution group at MPI MiS for useful discussions, especially Vincent Messow and Vaitea Opuu, working on the RNA bioinformatics project. The Alexander von Humboldt Foundation provided funding for this work in the Sofja Kovalevskaja Award framework endowed by the German Federal Ministry of Education.

REFERENCES

- [1] Jeff Anderson-Lee, Eli Fisker, Vineet Kosaraju, Michelle Wu, Justin Kong, Jeehyung Lee, Minjae Lee, Mathew Zada, Adrien Treuille, and Rhiju Das. 2016. Principles for predicting RNA secondary structure design difficulty. *Journal of molecular biology* 428, 5 (2016), 748–757.
- [2] Philippe Brion and Eric Westhof. 1997. Hierarchy and dynamics of RNA folding. *Annual review of biophysics and biomolecular structure* 26, 1 (1997), 113–137.
- [3] Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldspühl, and Danny Barash. 2017. Design of RNAs: comparing programs for inverse RNA folding. *Briefings in bioinformatics* 19, 2 (2017), 350–358.
- [4] N Dromi, A Avihoo, and D Barash. 2008. Reconstruction of natural RNA sequences from RNA shape, thermodynamic stability, mutational robustness, and linguistic complexity by evolutionary computation. *Journal of Biomolecular Structure and Dynamics* 26, 1 (2008), 147–161.
- [5] Ali Esmaili-Taheri and Mohammad Ganjtabesh. 2015. ERD: a fast and reliable tool for RNA design including constraints. *BMC bioinformatics* 16, 1 (2015), 20.
- [6] Ali Esmaili-Taheri, Mohammad Ganjtabesh, and Morteza Mohammad-Noori. 2014. Evolutionary solution for the RNA design problem. *Bioinformatics* 30, 9 (2014), 1250–1258.
- [7] Juan Antonio Garcia-Martin, Peter Clote, and Ivan Dotu. 2013. RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. *Journal of bioinformatics and computational biology* 11, 02 (2013), 1350001.
- [8] Michael S Goldberg, Deyin Xing, Yin Ren, Sandra Orsulic, Sangeeta N Bhatia, and Phillip A Sharp. 2011. Nanoparticle-mediated delivery of siRNA targeting Parp1 extends survival of mice bearing tumors derived from Brca1-deficient ovarian cancer cells. *Proceedings of the National Academy of Sciences* 108, 2 (2011), 745–750.
- [9] Chenhui Hao, Xiang Li, Cheng Tian, Wen Jiang, Guansong Wang, and Chengde Mao. 2014. Construction of RNA nanocages by re-engineering the packaging RNA of Phi29 bacteriophage. *Nature communications* 5, 1 (2014), 1–7.

Table 3: Summary of performance of aRNAque vs the 4 other algorithms benchmarked on the dataset B by A.Esmaili-Taheri et al. [5]

ID	Length(nt)	aRNAque		ERD		INFO-RNA		MODENA		NUPACK	
		$E_t(s)$	SC	$E_t(s)$	SC	$E_t(s)$	SC	$E_t(s)$	SC	$E_t(s)$	SC
RF00008	54	0.115	50	0.16	50	0.011	50	41.905	49	7.005	50
RF00029	73	0.371	50	0.373	50	0.033	50	71.502	38	41.415	41
RF00005	74	0.227	50	0.045	50	0.032	50	36.046	47	24.85	50
RF00027	79	0.2088	50	0.1	50	0.119	50	61.825	40	20.732	50
RF00019	83	0.336	50	0.136	50	0.091	50	50.813	35	9.211	50
RF00014	87	0.2914	50	0.103	50	0.034	50	63.707	35	6.389	50
RF00006	89	0.4	50	0.23	50	0.255	50	37.926	41	51.291	50
RF00026	102	0.36	50	0.038	50	11.294	50	57.124	44	209.624	50
RF00001	117	1.51	50	2.389	49	0.255	50	53.41	46	25423.278	1
RF00021	118	0.556	50	0.3	50	0.164	50	59.139	46	18.411	50
RF00020	119	∞	0	∞	0	∞	0	∞	0	∞	0
RF00016	129	∞	0	∞	0	∞	0	∞	0	4028.944	7
RF00015	140	2.46	50	2.218	49	2.885	50	77.096	42	∞	0
RF00022	148	2.01	50	2.846	50	6.384	50	77.024	44	1870.129	14
RF00002	151	3.76	50	7.831	49	11.294	50	64.602	39	∞	0
RF00007	154	2.64	50	1.429	50	0.975	50	85.534	43	531.148	32
RF00003	161	34.62	34	23.325	43	379.851	14	∞	0	∞	0
RF00013	185	5.92	50	3.014	50	8.601	50	104.908	46	18.1	50
RF00004	193	6.24	50	3.177	50	5.883	50	84.767	41	814.047	14
RF00025	210	11.34	50	7.584	50	7.361	50	106.894	48	∞	0
RF00012	215	13.8	50	3.164	50	44.89	49	111.521	48	345.226	46
RF00017	301	144	50	4.331	50	1.373	50	350.1	48	69.3605	50
RF00030	340	10	50	∞	0	207.593	50	294.552	36	∞	0
RF00028	344	8.7	50	147.091	44	∞	0	∞	0	∞	0
RF00009	348	10	50	268.613	28	∞	0	288.867	45	∞	0
RF00010	357	∞	0	∞	0	∞	0	∞	0	∞	0
RF00018	360	34.37	38	∞	0	296.386	50	293.777	39	∞	0
RF00011	382	∞	0	∞	0	∞	0	∞	∞	0	0
RF00024	451	∞	0	∞	0	∞	0	∞	0	1387.425	12
SUM	-	305.6852	1172	478.367	1062	986.298	1046	2467.039	940	34876.594	682

- [10] Robert Kleinkauf, Martin Mann, and Rolf Backofen. 2015. antaRNA: ant colony-based RNA sequence design. *Bioinformatics* 31, 19 (2015), 3114–3121.
- [11] Josef Leydold and Peter F Stadler. 1998. Minimal cycle bases of outerplanar graphs. *The Electronic Journal of Combinatorics* 5, R16 (1998).
- [12] Adam Lipowski and Dorota Lipowska. 2012. Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications* 391, 6 (2012), 2193–2196.
- [13] Ronny Lorenz, Stephan H Bernhart, Christian Höner Zu Siederdisen, Hakim Tafer, Christoph Flamm, Peter F Stadler, and Ivo L Hofacker. 2011. ViennaRNA Package 2.0. *Algorithms for molecular biology* 6, 1 (2011), 26.
- [14] David H Mathews, Matthew D Disney, Jessica L Childs, Susan J Schroeder, Michael Zuker, and Douglas H Turner. 2004. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. *Proceedings of the National Academy of Sciences* 101, 19 (2004), 7287–7292.
- [15] Fernando Portela. 2018. An unexpectedly effective Monte Carlo technique for the RNA inverse folding problem. (2018), 345587 pages.
- [16] Vladimir Reinharz, Yann Ponty, and Jérôme Waldspühl. 2013. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotide distribution. *Bioinformatics* 29, 13 (June 2013), i308–i315. <https://doi.org/10.1093/bioinformatics/btt217> arXiv:<https://academic.oup.com/bioinformatics/article-pdf/29/13/i308/18534655/btt217.pdf>
- [17] Jade Shi, Rihui Das, and Vijay S Pande. 2018. SentRNA: Improving computational RNA design by incorporating a prior of human design strategies. (2018).
- [18] Paul R Stein and Michael S Waterman. 1979. On some new sequences generalizing the Catalan and Motzkin numbers. *Discrete Mathematics* 26, 3 (1979), 261–272.
- [19] Akito Taneda. 2011. MODENA: a multi-objective RNA inverse folding. *Advances and applications in bioinformatics and chemistry: AABC* 4 (2011), 1.
- [20] Ignacio Tinoco Jr and Carlos Bustamante. 1999. How RNA folds. *Journal of molecular biology* 293, 2 (1999), 271–281.
- [21] Douglas H. Turner and David H. Mathews. 2009. NNDB: the nearest neighbor parameter database for predicting stability of nucleic acid secondary structure. *Nucleic Acids Research* 38, suppl1 (2009), D280–D282.
- [22] Maung Nyan Win and Christina D Smolke. 2008. Higher-order cellular information processing with synthetic RNA devices. *Science* 322, 5900 (2008), 456–460.
- [23] Tianbing Xia, John SantaLucia Jr, Mark E Burkard, Ryszard Kierzek, Susan J Schroeder, Xiaoqi Jiao, Christopher Cox, and Douglas H Turner. 1998. Thermodynamic parameters for an expanded nearest-neighbor model for formation of RNA duplexes with Watson-Crick base pairs. *Biochemistry* 37, 42 (1998), 14719–14735.
- [24] Joseph N Zadeh, Brian R Wolfe, and Niles A Pierce. 2011. Nucleic acid sequence design via efficient ensemble defect optimization. *Journal of computational chemistry* 32, 3 (2011), 439–452.