Design of RNAs: comparing programs for inverse RNA folding.

Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, Danny Barash

▶ **To cite this version:**

Alexander Churkin, Matan Drory Retwitzer, Vladimir Reinharz, Yann Ponty, Jérôme Waldispühl, et al.. Design of RNAs: comparing programs for inverse RNA folding.. Briefings in Bioinformatics, Oxford University Press (OUP), 2018, 19 (2), pp.350–358. 10.1093/bib/bbw120 . hal-01392958

HAL Id: hal-01392958

https://hal.inria.fr/hal-01392958

Submitted on 5 Nov 2016

Design of RNAs: Comparing Programs for inverse RNA folding

Alexander Churkin [1] Matan Drory Retwitzer [2] Vladimir Reinharz [2,3]
Yann Ponty [4] Jérôme Waldispühl [3]
Danny Barash [2]*

[1] Shamoon College of Engineering and Physics Department at Ben-Gurion University, Beer-Sheva 84105, Israel;
[2] Department of Computer Science, Ben-Gurion University, Beer-Sheva 84105, Israel;
[3] School of Computer Science, McGill University, Montréal QC H3A 0E9, Canada;
[4] Laboratoire d'informatique, École Polytechnique, Palaiseau 91128, France;

This article has been accepted for publication in Briefings in Bioinformatics:

Churkin A, Retwitzer MD, Reinharz V, Ponty Y, Waldispühl J, and Barash D. *Design of RNAs: Comparing Programs for inverse RNA folding.* To appear in Briefings in Bioinformatics, 2017

Abstract. Computational programs for predicting RNA sequences with desired folding properties have been extensively developed and expanded in the past several years. Given a secondary structure, these programs aim to predict sequences that fold into a target minimum free energy secondary structure, while considering various constraints. This procedure is called inverse RNA folding. Inverse RNA folding has been traditionally used to design optimized RNAs with favorable properties, an application that is expected to grow considerably in the future in light of advances in the expanding new fields of synthetic biology and RNA nanostructures. Moreover, it was recently demonstrated that inverse RNA folding can successfully be used as a valuable pre-processing step in computational detection of novel non-coding RNAs. This review describes the most popular freeware programs that have been developed for such purposes, starting from `RNAinverse` that was devised when formulating the inverse RNA folding problem. The most recently published ones that consider RNA secondary structure as input are `antaRNA`, `RNAiFold` and `IncaRNAfbinv`, each having different features that could be beneficial to specific biological problems in practice. The various programs also employ distinct approaches, ranging from ant-colony optimization to constraint programming, in addition to adaptive walk, simulated annealing and Boltzmann sampling. This review compares between the various programs and provides a simple description of the various possibilities that would benefit practitioners in selecting the most suitable program. It is geared for specific tasks requiring RNA design based on input secondary structure, with an outlook towards the future of RNA design programs.

*To whom correspondence should be addressed: dbarash@cs.bgu.ac.il

1 Introduction

The inverse RNA folding problem for designing sequences that fold into a given RNA secondary structure was introduced in the early 1990's in Vienna [1]. Mathematically, much like the typical situation with inverse problems, it is not a well-posed problem by the standard definition of Hadamard, which makes it even more challenging to solve. As the well-known mathematician Andrey Tikhonov once noted, the class of ill-posed problems includes many classical mathematical problems and, most significantly, that such problems have important applications. Indeed, new emerging sub-fields that are of significant importance to a variety of functional RNAs [2, 3, 4, 5, 6], such as RNA synthetic biology [7, 8] and RNA nanostructure [9, 10], are fast developing and are utilizing in their arsenal the methods for solving inverse RNA folding [11, 12, 13, 14, 15, 16, 17, 18].

A brute force approach that searches all the possible sequences is not a viable option because the number of sequences grows exponentially as 4n, where n is the length of the sequence, while the number of valid designs can be arbitrarily small. This upper bound can be refined by noting that paired positions have to form valid base pairs under the standard A-U, C-G, G-U base pairing scheme. This implies that $6^{p/2}4^{u}$ sequences are compatible with a secondary structure having respectively u unpaired and p paired nucleotides. As a practical consequence, a typical 74 nts-long tRNA, including $p = 40$ paired and $u = 34$ unpaired ones, would require investigating $\sim 10^{36}$ compatible sequences.

RNA inverse folding also has deep connections with theoretical evolutionary studies, where the sequence/structure relationship in RNA is a popular model for studying genotype/phenotype maps [19, 20]. For instance, the identification of undesignable motifs [21] in empirical design studies immediately implies that only a negligible, exponentially decreasing on the length, proportion of secondary structures can be designed. Conversely, neutral evolution provides theoretical foundations for the practice of RNA design, and studies of the neutral network confirm a highly-variable numbers of admissible designs within structures of the same length [22]. It is interesting to note that the distribution of the neutral network [19] could help us understand how to further develop efficient local search strategies to reach the target structure.

The approach to solve the inverse RNA folding problem by stochastic optimization relies on the solution of the direct problem using software available in RNA folding prediction web servers, e.g. the `RNAfold` server [23, 24] or mfold/UNAFold [25, 26] as well as `RNAStructure` [27], by performing energy minimization with thermodynamic parameters [28, 29]. It should be noted that in principle, although far less popular in practice in the context of inverse RNA folding, other programs based on probabilistic models and posterior decoding that have been benchmarked in [30], e.g. `PFold` [31] and `CentroidFold` [32], can also be used to solve the direct problem. Initially, a seed sequence is chosen, after which a local search strategy is used to mutate the seed and apply repeatedly the direct problem of RNA folding prediction by energy minimization. Then, in the vicinity of the seed sequence, a designed sequence is found with desired folding

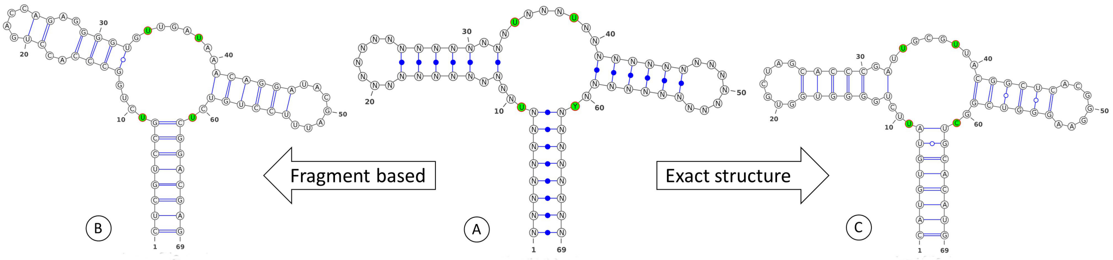

Figure 1: The standard inverse RNA folding problem and the generalized inverse RNA folding problem that is shape aware and fragment-based (i.e., fragment selection enabled) are illustrated on the purine riboswitch aptamer in the middle (A). The predicted structure of an output designed sequence is shown on the right (C) for the standard inverse folding problem and on the left (B) for the generalized inverse RNA folding problem.

properties according to the objective function in the optimization problem formulation. Chronologically, algorithmic improvements that relate to this approach which was pioneered in Vienna's RNAinverse [1] have been worked out in INFO-RNA [33], RNA-SSD [34] and NUPACK:Design [35, 36]. Alternative methods to the adaptive random walk [1] and the stochastic local-search [33, 34] include genetic algorithms (belonging to the class of evolutionary algorithms) [37, 38, 39], an improved evolutionary algorithm [40, 41], constraint programming [42, 43] and ant-colony optimization [44, 45]. On the methodological side, two separate ideas that were gradually developed and investigated were to sample the sequence space more efficiently and to include user-selected fragments in the design. The first idea started in [33] and then by presenting a global sampling approach named RNA-ensign [46], followed by a weighted sampling algorithm called incaRNAtion [47] that is a glocal methodology combining the global sampling approach with local search strategies. The second idea started by generalizing the inverse RNA folding problem to include RNA designed sequences that are predicted to fold into a prescribed shape [37, 48], a utility named RNAexinv that considers the coarse-grain tree graph [49] for shape representation (or potentially abstract shapes [50]). It culminated with a method called RNAfbinv [51] that allows a user-selected prescribed fragment to be preserved exactly by secondary structure (in addition to its shape) whereas the rest of the structure is designed by the generalized shape-based approach. These two ideas were recently merged into an RNA design webserver called IncaRNAfbinv [52] that provides more flexibility in the design as compared to the aforementioned methods. An example of the benefit of IncaRNAfbinv is illustrated in Figure 1 where it is shown that the designed sequence on the left (Figure 1(B)) is a feasible purine riboswitch candidate as much as the designed sequence on the right (Figure 1(C)), both containing the essential nucleotides for either guanine or adenine binding, but the Figure 1(B) solution cannot be reached to-date by other programs besides IncaRNAfbinv because its secondary structure is different than the input structure depicted in Figure 1(A), although its tree graph shape is the same.

As was mentioned above and exemplified in the shape aware capability, different computational frameworks for the inverse RNA folding have been implemented in the various programs. In addition to the strict definition that restricts solutions to those sequences whose mean-free-energy structure is exactly the target structure, more relaxed frameworks like ensemble defect optimization in NUPACK have been introduced and developed [35], in addition to minimizing the classical cost function given by the ”structure distance” between the structure of the test sequence and the target structure [1]. For a candidate sequence and a given target secondary structure, the ensemble defect is the average number of incorrectly paired nucleotides at equilibrium evaluated over the ensemble of unpseudoknotted secondary structures [36]. It could also be possible, although not currently done by any available software, to add prescribed kinetic properties (fast folding, absence of kinetic traps) to the objectives of design, as explored in earlier studies [35]. As will be noticed in the quantitative comparison performed in the continuation, these different computational frameworks will also make it impossible to draw a conclusion as to which program is better for use based on a benchmark. Instead, when facing an application that requires inverse RNA folding, the goal is to try and identify which is the most suitable program for each case.

Some inverse RNA folding programs are geared towards more specific biological problems, compared to the ones mentioned up until now that are general in their application scope. For example, nanostructure design including pseudoknots was performed with `Nanofolder` for the case of multi-stranded RNA secondary structure [12]. In another example, for designing the most stable and unstable mRNA sequences which code for a target protein, an algorithm was developed in [53]. On a similar topic, in [54], an algorithm for designing a protein-coding sequence with the most stable secondary structure called CDSfold is provided. For designing multiple-target artificial miRNAs, a tabu search was used in conjunction with biochemical considerations in [55]. For designing RNA sequences that fold into multiple target structures, which makes it possible to efficiently design multi-stable RNA sequences, a program called RNAdesign was introduced in [56]. Another specialized application, given here for completeness, is to allow game players to propose a set of rules for RNA design, as part of the Eterna project. Based on experimental results, Eterna players came up with a set of design rules and EternaBot was developed to design a sequence based on those rules [57]. Finally, for the problem of fixed backbone 3D design of RNA, the webserver RNA-redesign was put forth [58]. To the best of our knowledge, this is the first implementation of an RNA design program that considers tertiary structure, although it is a local design. In this respect, in secondary structure, sequences that are generated by point mutations performed on an input sequence to optimize a certain objective function may also be considered a design procedure in the weaker (local) sense. Example of such programs are RNAmutants [59] that employs an efficient sampling procedure based on the Boltzmann-weighted ensembles of mutants and RNAmute [60] that utilizes suboptimal solutions of the RNA free energy minimization prediction [61, 62] for simulating only selective mutations from all possible ones. These procedures could potentially also be integrated into RNA design tools in the future, as was already done by incorporating RNAmutants ingredients into `incaRNAtion` [47].

Inverse RNA folding programs are not only used for the design of artificial sequences to mimic natural ones. Recently, they have been used for the detection of novel naturally occurring RNAs as a pre-processing step before sequence-based searches [63, 64]. In both of these separate works, they have shown to find attractive candidates for naturally occurring RNAs that are not available in RFAM [65] and are missed by standard methods for RNA detection. This approach can well contribute to ongoing efforts aiming for de novo discovery of structured non-coding RNAs in genomic sequences [66].

2 Details of use

Installing the different programs or accessing the webservers is not a difficult task. They all contain a ReadMe file or a manual that is easy to follow with no prior knowledge assumed. However, not all programs provide both webserver and source code. Table 1 lists the various RNA design programs according to four categories: general purpose programs, shape aware programs, adaptive sampling programs and specialized programs. It then indicates the availability of webserver or source code for each program, including extended features and general remarks that relate to their capability, use, or strategy without providing more details on their specific methodology.

From all programs listed in Table 1, we picked five programs for further description on their details of use based on the following selection criteria: the programs have both a webserver and a source code available, and they were already used in the literature in a biological meaningful way by either a "wet-lab" experiment or the identification of a putative new non-coding RNAs. These criteria are of interest to practitioners who are considering the use of RNA design programs. The selection yielded the programs `RNAinverse`, `RNAiFold`, `NUPACK` and `IncaRNAfbinv`. In addition, the `antaRNA` program was added because it was published quite recently without a chance yet for practical use but it is considered promising as can also be observed by the program comparisons provided in next section and its overall strategy that allows much flexibility. Finally, although `Nanofolder` could not be added because of a lack of source code and our recommendation for the interested user would be to contact its authors [12], it is worthwhile noting the significant practical experience that has been accumulated by `Nanofolder` as a specialized program for RNA nanostructure design. There are sequence design rules implemented in `Nanofolder` that have been formulated based on the concept of same-length sequence fragments called "critons" [12], which have been successfully applied beforehand to the design of DNA nanostructures. These special rules with employed penalty scoring terms have been formulated to avoid unstable RNA designs and optimize the designed sequences. User experience with the five selected programs was performed on two example input secondary structures in dot-bracket notation (the first is a toy-problem, an artificial structure; the second is the structure of the guanine-binding riboswitch aptamer, a natural structure):

1. `((((...(((....)))...((((....))))...))))`

2. `((((((((...(.(((((.......))))).)........((((((.......))))))..))))))))`

Programs	Webserver	Source code	Extension to pseudoknots	Multi-target capability	Remarks
General					
antaRNA [44, 45]	●	●	●		
RNAiFold [42, 43]	●	●		●	Experience in biology ”wet-lab”
RNAinverse [1]	●	●			First program developed; Experience in biology ”wet-lab”
NUPACK [35, 36]	●	●			Optional multi-stranded target structures; experience in biology ”wet-lab”
INFO-RNA [33]	●	●			
RNA-SSD [34]	●				
Frnakenstein [39]		●		●	
ERD [40, 41]	●	●	●		
MODENA [38]		●	●	●	
Shape aware					
IncaRNAfbinv [52]	●	●			Fragment selection enabled; Experience in RNA detection
RNAfbinv [51]		●			Fragment selection enabled
RNAexinv [48]		●			No user-selected fragment
Adaptive sampling					
IncaRNAfbinv [52]	●	●			Global-local approach; Experience in RNA detection
IncaRNAtion [47]		●			Global-local approach
RNA-ensign [46]	●				Global approach
Specialized					
Nanofolder [12]	●		●		Nanostructures; multi-stranded RNA; experience in biology ”wet-lab”
CDSfold [54]	●	●			Design of protein-coding seq
RNAdesign [56]		●		●	
EternaBot [57]	●				Design rules set by Eterna players
RNA-redesign [58]	●				3D: fixed backbone

Table 1: A tabular overview with some basic information about the various RNA design programs.

2.1 RNAinverse in detail

RNAinverse [1] was the first program developed for RNA design. It is available as a webserver at

http://rna.tbi.univie.ac.at/cgi-bin/RNAinverse.cgi

and as a standalone version in the Vienna RNA package [24]. The algorithm employs an adaptive random walk to minimize base pair distance. The distance is calculated by comparing the minimal energy folding of a mutated sequence (its predicted structure) to the target structure. To avoid folding the entire sequence, small substructures are optimized and then elongated. The algorithm also supports designing sequences which are more probable based on the partition function. Those sequences may be more stable but mostly differ from the target structure.

The server receives a secondary structure in dot bracket notation. An optional start sequence can be inserted; any lower case letter in the sequence will be conserved in the final result. The server also supports multiple energy models, folding temperature and number of sequences to generate. Once the form is submitted the result page will appear with a list of designed sequences and the calculated minimum energy for them. It may also show designed sequences that did not match the exact structure and their base pair distance away. Another set of results that will appear below that are designed sequences using the partition function if selected. The webserver also shows links to RNAfold [24] for extensive information on a specific result. The command line used to run the design in the stand-alone version is also written.

The stand-alone version of RNAinverse is part of the Vienna RNA package. The package is a C code library that includes several stand-alone programs. This means RNAinverse can be accessed both from command line and through the package API. The simple API allows the user to design and test the sequences for individual purposes directly through C. The stand-alone version is simple and examples can be generated by using the webserver as discussed above.

2.2 RNAiFold in detail

RNAiFold [42, 43] is a well-established program available as a webserver at

http://bioinformatics.bc.edu/clotelab/RNAiFold/

It employs constraint programming and is available in two modes. RNA-CP Design take as simplest input a secondary structure and returns up to 50 sequences or the maximum found in 2 hours, optimized for one of three different criteria: yielding the target as the minimum free energy (MFE) structure, the free energy or the ensemble defect. RNAiFold ensures the optimality of the solutions. Additional constraints can be provided such as a target amino acid sequence, and limits on the amount of GC, AU and GU base pairs as a list of admissible and forbidden base pair. The nucleotide distribution and energy model are customizable. A special mode, RNA Synthetic Design, leverages RFAM [65] to add constraints from sequence conservation. A drop down menu allows the selection of any family and automatic extraction of the constraints. The consensus structure can be automatically selected as target. The constraints from RNA-CP Design are also available.

The output presents for each resulting sequence a number of different statistics such as its GC content, energy and entropy, and its amino acid sequence. It also provides a link to MPMATH [67] the resulting sequence. Structure (2) was tested on the webserver. It took a short time before the query was allowed to run and returned 23 sequences after 2 hours. A version can be downloaded in source code or binaries for Linux or Mac OS X, the source requires the Vienna Package as the open source Google optimization library OR-Tools. The binary was tested on Ubuntu 12.04 while the Mac binaries still need to be updated for OS 10.11.5. It provides a simple input by command line argument or through a file in a custom format to fine-tune a few more arguments of the algorithm itself, and provides a similar output.

2.3 antaRNA in detail

antaRNA [44, 45] is a very recent program available since 2015, as a webserver at

http://rna.informatik.uni-freiburg.de/antaRNA/

It employs ant colony optimization to allow the design of structures, allowing pseudoknotted structures using hardcoded energy parameters. A sequence constraint can be provided in the IUPAC format. A visualization of the secondary structure with the sequence constraint is dynamically shown. Additionally a target GC constraint can be set. The parameters of the ant colony algorithm can be modified through advanced options.

Each output sequence, up to a 100, is shown with its MFE structure and its distance from the target GC, target structure and target sequence. A click on a sequence will show a visualization of its secondary structure with the sequence embedded. Testing for structure (2) while requesting 100 sequences took a few minutes on the webserver. All the results contained all the requested base pairs, a few additional base pairs were sometimes present. A convenient link to download all the sequences in FASTA format with or without their MFE is provided.

The python script requires the Vienna RNA Package and provides the same options as the webserver. For pseudoknots prediction a finer control is provided requiring the user to have installed one of the programs RNAShapes studio [68] or HotKnots [69] or IPKnot [70]. All the options from the webserver are present and must be given through command line arguments. The script removes the limit of sequences sampled.

2.4 NUPACK in detail

NUPACK [35, 36] is a recent program developed in 2011. It is available as a webserver at

http://www.nupack.org/design/new

Its objective is to minimize the ensemble defect for a pseudoknot free structure. Its interface allows the user to specify a target secondary structure as a preference for DNA or RNA. An interesting feature of NUPACK is the ability to define unwanted motif by providing a list of forbidden sequence motifs, in IUPAC format. The webserver allows to choose between two energy models, Turner95 and Mathews99, dangles, and setting the temperature. A maximum of 10 sequences can be designed concurrently. The output presents the designed sequences. Tested on structure (2), in a few minutes 10 sequences were generated. An analysis of the sequences can be launched immediately to compute the MFE and base pair probabilities. A range of temperature can additionally be provided for this step. A link is provided to download the MFE secondary structure representation and the base pair probabilities. NUPACK is also available as command line software and was tested on a MAC OS X 10.11.5. Some options are not available on the webserver, such as setting a seed sequence or specifying the concentration of salt and magnesium. A few parameters of the algorithm can also be modified from the command line as its random seed, which is necessary to generate different sequences.

2.5 IncaRNAfbinv in detail

IncaRNAfbinv [52] is a very recent program for fragment-based design. It is available as a webserver at

https://www.cs.bgu.ac.il/IncaRNAfbinv/

The webserver combines two base applications: incaRNAtion [47] and RNAfbinv [51]. Both applications are available as a stand-alone client. RNAfbinv uses simulated annealing with a 4-nt look ahead local search function. The function includes biologically meaningful constraints such as sequence constraints, fragment based design, and a variety of optional features. The resulting sequences fit a coarse-grained tree graph shape of the original target structure, thus allowing for flexibility. incaRNAtion augments the local search method. It uses a global sampling approach and weighted sampling techniques. The sequences generated by incaRNAtion are used as seed sequences for the local search. The incorporation of those seeds forces highly distributed results and better control of GC content.

The webserver takes as input the target secondary structure in dot bracket representation. It then converts it to coarse-grained tree graph shape for future comparison. After inserting a structure, an image will appear, as well as a list of structural motifs from which the user can select a desired motif. Additional optional parameters are: sequence constrains, target fold energy, mutational robustness, and GC content. Submission of the query leads to a web page showing design progress. Once all the results are ready, a list will appear with the designed sequences as well as their predicted structure, folding energy, mutational robustness, GC content, and distance to the original structure. The distances are calculated for both base pairs and structural motifs. Finally, a link is available for an image of the designed structure.

The stand-alone versions of both tools can be run locally, links can be found in the webserver. incaRNAtion is a simple to run Python script, thus requiring Python installed and recommends adding the MPMATH library for long sequences. It receives as input a file containing the target structure and an optional multiple sequence alignment (MSA). The user is also required to enter a value between 0 and 1, where 1 means only to regard the structure and 0 the MSA. Additional variables are available for GC content control and specific sequence constraints. A single run will generate a large amount of sequences; a minimal value for the number of outputs can also be set as input. The output is a list of sequences separated by lines.

RNAfbinv is a C application but it is also available wrapped in a Java interface. Once the Java application is running, the user must first insert the secondary structure. The user then has an option to select a specific motif to preserve as is. After inserting the structure, the user arrives to a new screen where additional control variables are available such as target folding energy and mutational robustness. The results appear in a new screen as a list including the base pair distance from the input structure.

2.6 Using the programs

The main task for using these programs is to insert an RNA secondary structure into one of them and generate sequences accepting this target structure as the MFE structure, with possible generalizations that are closely related to this framework. All programs offer the possibility of additional parameters to be chosen by the user, with default values displayed at the beginning. Some programs offer more flexibility in their constraints than others, which is indicated in Table 1 for some selected features that are shared by several programs and are general in scope. As final output, all programs offer description of parts of the analysis of designed sequences as well. RNAiFold displays the results in one webpage per solution, seemingly in the order generated and selectable through a drop down menu. It presents the MFE structure of each sequence, as an ensemble of statistics, and provides an option to MPMATH them. antaRNA directly displays each generated sequence with its MFE structure. A click on each of the solutions creates a figure of the secondary structure with the sequence. Any constraint violation is represented in red in the figure. The list of sequences can be downloaded in FASTA. NUPACK presents each sequence by increasing ensemble defect. Each sequence has a link to the analysis tool, which computes the MFE structure and base pair probabilities. In addition, the textual output provided in all these programs is substantially contributing to the analysis, as an essential step before the graphical output. In general, most of the programs are user friendly for the novice. Especially the programs that have a webserver capability (see Table 1) can also be worked out by a non-specialist user along with the corresponding manuals and instructions that are available in the websites.

3 Discussion

The programs listed in Table 1 have been developed in the past several years, following the first program named RNAinverse that was introduced more than twenty years ago, and offer some interesting prospects for RNA sequence design. They all in one way or the other rely at present on thermodynamic parameters corresponding to the nearest-neighbor model and therefore structures that are known to be well predicted by energy minimization, for example the secondary structure of the guanine-binding riboswitch aptamer that is illustrated in the second test case example of previous section and in Figure 1, are the best to work with as inputs to these programs in order to achieve reliable results. Though exceptional cases exist, in general the upper range estimate

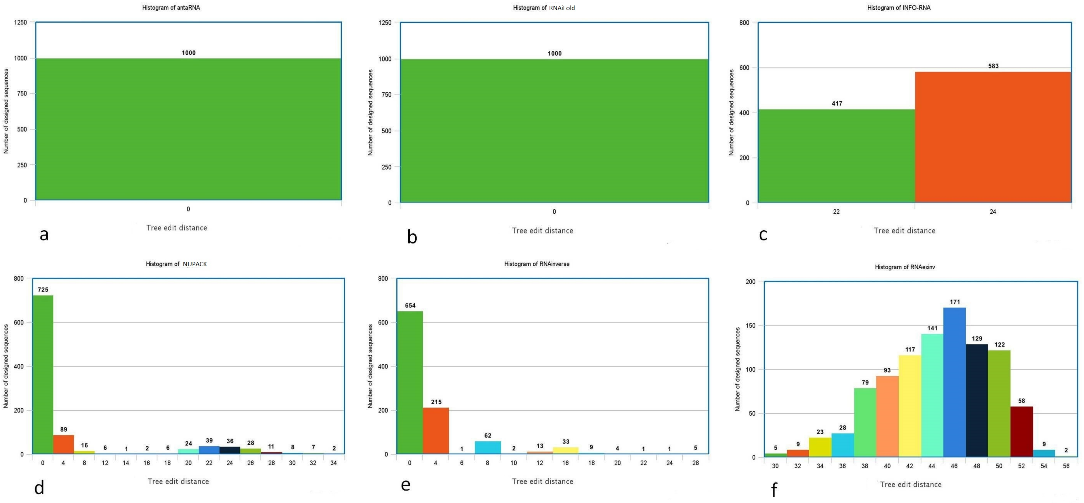

Figure 2: Histogram comparison between the six selected programs available in Table 2 for the example test case that is designated as (1) in the Details of Use Section and for which the runtimes are reported in the first column of Table 2.

for the sequence length that these programs are useful for is around 150 nt [71]. It is expected that in future, having more experimental structures elucidated, the number of RNA sequences with a well-predicted secondary structure by energy minimization techniques will grow significantly and more biological systems involving RNAs will be designed by the aid of these programs.

Runtime can be a critical issue concerning the usage of these tools. A runtime comparison of six programs is provided in Table 2 for the two test case examples that were provided in the previous section in dot-bracket notation (the first is a toy-problem, the second is the structure of the guanine-binding riboswitch aptamer). The times reported are in minutes. Standard parameter values were used in the comparison. Because runtimes are measured in downloadable source code and cannot be measured in programs that require user interactive intervention such as with IncaRNAfbinv and RNAfbinv, we replaced IncaRNAfbinv that was discussed in the previous section by the simpler and less developed program RNAexinv. The justification is that RNAexinv is shape aware (preserving the same coarse-grain tree graph shape in the output as in the input) without the user's interactive selection of a fragment for preserving its secondary structure exactly like in IncaRNAfbinv, therefore RNAexinv contains the shape aware feature itself for inclusion in the comparison. RNAexinv is still much slower than the rest of the programs because it solves a more general inverse RNA folding problem that is shape aware. By our past experience, the programs RNAfbinv and IncaRNAfbinv are about 10% more computationally expensive than RNAexinv. Additionally, we inserted the program INFO-RNA because it is known to be the most computationally efficient among all programs, as is also observed in Table 2. Correspondingly, for the two test cases measured in Table 2 by 1000 runs, a histogram is plotted in Figure 2 for the first test case

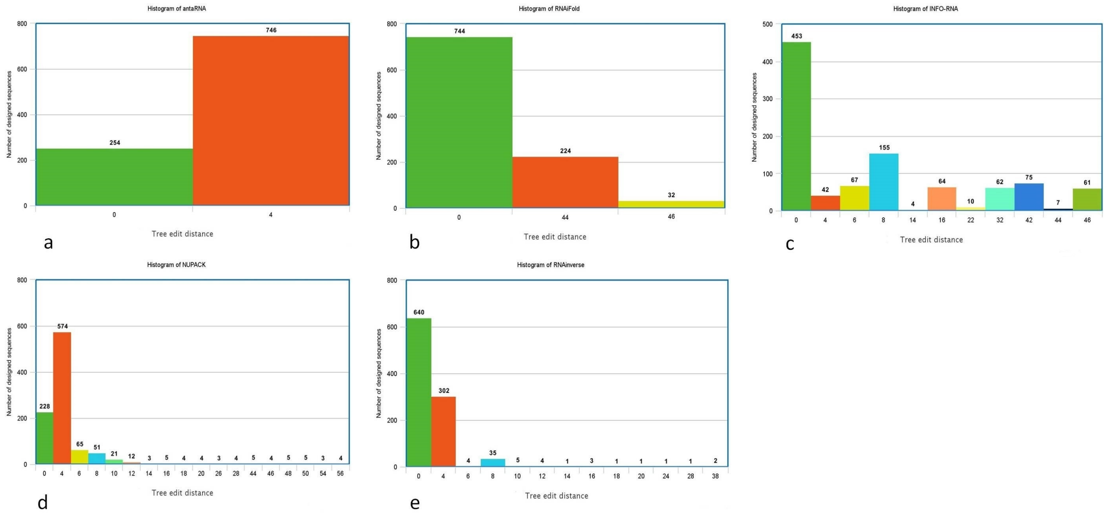

Figure 3: Histogram comparison between the five selected programs available in Table 2 for the example test case that is designated as (2) in the Details of Use Section and for which the runtimes are reported in the second column of Table 2.

and in Figure 3 for the second test case in order to examine how far away the predicted structures of the designed sequences are from the input structure that is initially given. The distance between the two secondary structures was measured by the `RNAdistance` routine available in the Vienna RNA package that calculates by default the tree edit distance.

While it is impossible to draw conclusions from Table 2 and the associated Figures 2,3 as to which program is better for use, because a program such as `RNAexinv` belonging to the shape aware category (see Table 1) is expected to be much slower along with histograms that are more wide-spread compared to the rest of the programs in a notably beneficial way for its purpose, some trends can be observed. For example, `INFO-RNA` is the quickest as expected, but its histograms are more widespread compared to most of the other programs and this correlates with the known result that sequences designed with `INFO-RNA` tend to be more biased to high GC-contents [47]. In contrast, the newly introduced `antaRNA` program from the same laboratory is both relatively fast and achieving histograms with results that are very close to the input structure. `RNAiFold` is also showing some fairly balanced outcomes between efficiency and proximity of the results to the input structure. Finally, `RNAinverse` shows impressively that although it was written more than 20 years ago and it features less constraints compared to the newer programs, it is still both fast and faithful to the input structure.

The above comparison is by no means exhaustive and can be supplemented by the additional references [43, 44, 45, 52]. These references could benefit a reader interested in the topic of run time comparisons, design capabilities and properties of the output sequences produced by each method. The `RNAiFold` webserver article [43] contains a section on comparison with other software. Because this article does not include the

Program	Time (minutes) Figure 2	Time (minutes) Figure 3
antaRNA	6.5	7.8
RNAiFold	41.4	6.4
INFO-RNA	0.5	0.8
NUPACK	37.5	217
RNAinverse	3.15	3.7
RNAexinv	231	N/A

Table 2: Runtime comparison between six selected programs with availability of source code. The times are reported in minutes and for 1000 runs in each of the two input structure examples. The left column is for the toy-problem designated as (1) in the *Details of Use* Section, while the right column is for the guanine-binding riboswitch aptamer designated as (2) in the *Details of Use* Section.

most recent methods antaRNA and IncaRNAfbinv, the interested reader can find more information about the performance of these algorithms in [44, 45] and [52], respectively. It should also be noted that the use of tree edit distance to the target structure in Figure 3 as a performance measure may not consider that some of the methods included do not necessarily use the same energy model and dangle treatment as used herein. The one used herein for computing the tree edit distance is the Turner 2004 model [29] included in the Vienna RNA package 2 [24]. While NUPACK results could be moderately accurate in any case since ensemble defect optimization mitigates the slight differences between energy models, the results of RNAinverse, RNAiFold, INFO-RNA and the rest of the programs could be affected by the energy model of choice.

3.1 New Prospects: designed RNAs for structure-based search

As was mentioned in the Introduction, a major new application of inverse RNA folding programs is the discovery of novel, structured and functional RNAs in transcriptomic data. We briefly describe the concept and refer the interested reader to [63, 64] for more information.

Sequence-based search tools like MPMATH [67] have been used extensively for the detection of novel RNAs of interest, such as riboswitches, in newly sequenced data. They are easily available, highly efficient, and can partially address this task. However, when the search is restricted to only sequence-based considerations, it is rather limited. The idea to augment MPMATH search with inverse RNA folding for including structure-based considerations has been developed independently for identifying IRES-like structural subdomains [63] and riboswitch aptamer domains [64], where in the first reference the findings were also verified experimentally and in the second reference the experimental verifications are ongoing. In both of these works, this strategy has been shown to yield attractive candidates that are beyond the reach of well-established methods like Infernal [72]. Consequently, an idea was even suggested by the authors of Infernal to augment their own tool by the inverse RNA folding pre-processing step. Various combinations should be tried and in any case, it is expected that in the future, inverse RNA folding would become useful not only for the design of synthetic RNAs but also for the search of naturally occurring RNAs by the use of designed RNAs as a pre-processing step.

3.2 Concluding remarks

The various programs, especially the ones who are gaining experiences in biological meaningful problems and are being improved as a consequence by updated versions, should best be examined along with the constraints they allow and their orientation purposes. There are already several programs that were described in detail and offer both a webserver implementation and source-code availability, along with a proven experience in biological meaningful problems. Other programs should strive to achieve these goals. Practitioners should then select which program is more suitable for their needs according to the specific constraints and capabilities that are advertised in each one of the programs.

4 Acknowledgements

The authors would like to thank the project students Maya Bechler-Speicher and Sarah Damyahoo, co-guided by DB together with Chen Keasar from Ben-Gurion University who assisted with his expertise, for help with the program comparisons. This work was supported by the Lynn and William Frankel Center for Computer Sciences at Ben-Gurion University and ISF within the ISF-UGC joint research program framework [9/14]. VR is grateful for being supported by an Azrieli and FQRNT postdoctoral fellowships.

References

[1] Hofacker IL, Fontana W, Stadler PF, et al. Fast folding and comparison of RNA secondary structures. Monatsh Chem 1994;125:167–188.

[2] Taft RJ, Pang KC, Mercer TR, et al. Non-coding RNAs: regulators of disease. J. Pathol. 2010;220(2):126-139.

[3] Hammann C, Westhof E. Searching genomes for ribozymes and riboswitches. Genome Biol 2007;8(4):210.

[4] Strobel SA, Cochrane JC. RNA catalysis: ribozymes, ribosomes, and riboswitches. Curr Opin Chem Biol. 2007;11(6):636-643.

[5] Breaker RR. Prospects for riboswitch discovery and analysis. Mol Cell 2011;43(6): 867-879.

[6] Serganov A, Nudler E. A decade of riboswitches. Cell 2013;152(1-2):17-24.

[7] Isaacs FJ, Dwyer DJ, Collins JJ. RNA synthetic biology. Nat. Biotechnol 2006;24(5): 545-554.

[8] Chappell J, Watters KE, Takahashi MK, et al. A renaissance in RNA synthetic biology: new mechanisms, applications and tools for the future. Curr Opin. Chem. Biol 2015;28:47-56.

[9] Jaeger L, Westhof E, Leontis NB. TectoRNA: modular assembly units for the construction of RNA nano-objects. Nucleic Acids Res 2001;29:455-463.

[10] Guo P. The emerging field of RNA nanotechnology. Nature Nanotechnol 2010;5:833-842.

[11] Mueller S, Coleman JR, Papamichail D, et al. Live attenuated influenza virus vaccines by computer-aided rational design. Nature Biotechnol 2010;28(7):723-726.

[12] Bindewald E, Afonin K, Jaeger L, at al. Multi-stranded RNA secondary structure prediction and nanostructure design including pseudoknots. ACS Nano 2011;5(12): 9542-9551.

[13] Afonin K, Kasprzak WK, Bindewald E, et al. In silico design and enzymatic synthesis of functional RNA nanoparticles. Acc. Chem. Res 2014;47(6):1731-1741.

[14] Fernandez-Chamorro J, Lozano G, Garcia-Martin JA, et al. Designing synthetic RNAs to determine the relevance of structural motifs in picornavirus IRES elements. Sci Rep 2016;6:24243.

[15] Dotu I, Garcia-Marin JA, Slinger BL, et al. Complete RNA inverse folding: computational design of functional hammerhead ribozymes. Nucleic Acid Res 2014;42(18):11752-11762.

[16] Findeiß S, Wachsmuth M, Mörl M, et al. Design of transcription regulating riboswitches. Methods Enzymol 2015;550:1-22.

[17] Wachsmuth M, Domin G, Lorenz R, et al. Design criteria for synthetic riboswitches acting on transcription. RNA Biol 2015;12(2):221-231.

[18] Ben-Yehezkel T, Atar S, Zur H, et al. Rationally designed, heterologous S. cerevisiae transcripts expose novel expression determinants. RNA Biol 2015;12(9):972-984.

[19] Schuster P, Fontana W, Stadler PF, et al. From sequences to shapes and back: a case study in RNA secondary structures. Proc. Biol. Sci.1994;255(1344):279-284.

[20] Greenbury SF, Schaper S, Ahnert SE, et al. Genetic correlations greatly increase mutational robustness and can both reduce and enhance evolvability. PLOS Comput Biol 2016;12(3):e1004773.

[21] Aguirre-Hernández R, Hoos HH, Condon A. Computational RNA secondary structure design: empirical complexity and improved methods. BMC Bioinformatics 2007;8:34.

[22] Jörg T, Martin OC, Wagner A. Neutral network sizes of biological RNA molecules can be computed and are not atypically small. BMC Bioinformatics 2008; 9:464.

[23] Hofacker IL. Vienna RNA secondary structure server. Nucleic Acids Res 2003;31:3429–3431.

[24] Lorenz R, Bernhart SH, Höner Zu Siederdissen C, et al. ViennaRNA Package 2.0. Algorithms Mol Biol 2011;8(6):938-946.

[25] Zuker M. Mfold web server for nucleic acid folding and hybridization prediction. Nucleic Acids Res 2003;31:3406–3415.

[26] Markham NR, Zuker M. UNAFold: software for nucleic acid folding and hybridization. Methods Mol Biol 2008;453:3-31.

[27] Mathews DH. RNA secondary structure analysis using `RNAStructure`. Current Protocols in Bioinformatics 2014;46:12.4.1-12.4.22.

[28] Mathews DH, Sabina J, Zuker M, et al. Expanded sequence dependence of thermodynamic parameters provides improved prediction of RNA secondary structure. J Mol Biol 1999;288:911-940.

[29] Mathews DH, Disney MD, Childs JL, et al. Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure. Proc. Natl. Acad. Sci. USA 2004;101(19), 7287-7292.

[30] Puton T, Kozlowski LP, Rother KM, et al. CompaRNA: a server for continuous benchmarking of automated methods for RNA secondary structure prediction. Nucleic Acids Res 2013;41(7):4307-4323.

[31] Knudsen B, Hein J. PFold: RNA secondary structure prediction using stochastic context free grammars. Nucleic Acids Res 2003;31:3423-3428.

[32] Sato K, Hamada M, Asai K, Mituyama T. CentroidFold: a webserver for RNA secondary structure prediction. Nucleic Acids Res 2009;37(2):W277-W280.

[33] Busch A, Backofen R. INFO-RNA-a fast approach to inverse RNA folding. Bioinformatics 2006;22(15), 1823-1831.

[34] Andronescu M, Fejes AP, Hutter F, et al. A new algorithm for RNA secondary structure design. J Mol Biol 2004;336(3):607-624.

[35] Dirks RM, Lin M, Winfree E, et al. Paradigms for computational nucleic acid design. Nucleic Acids Res 2004;32(4):1392-1403.

[36] Zadeh, JN, Wolfe BR, Pierce NA. Nucleic acid sequence design via efficient ensemble defect optimization. J. Comput. Chem 2011;32(3), 439-452.

[37] Dromi N, Avihoo A, Barash D. Reconstruction of natural RNA sequences from RNA shape, thermodynamic stability, mutational robustness, and linguistic complexity by evolutionary computation. J. Biomol. Struct. Dyn. 2008;26(1), 147-162.

[38] Taneda A. Multi-objective genetic algorithm for pseudoknotted RNA sequence design. Front. Genet 2012;3:36.

[39] Lyngsø RB, Anderson J WJ, Sizikova E, et al. Frnakenstein: multiple target inverse RNA folding. BMC Bioinformatics 2012;13:260.

[40] Esmaili-Taheri A, Ganjtabesh M, Mohammad-Noori, M. Evolutionary solution for the RNA design problem. Bioinformatics 2014;30(9):1250-1258.

[41] Esmaili-Taheri A, Ganjtabesh M. ERD: a fast and reliable tool for RNA design including constraints. BMC Bioinformatics 2015;16: 20.

[42] Garcia-Martin JA, Clote P, Dotu I. RNAiFold: A constraint programming algorithm for RNA inverse folding and molecular design. J Bioinform Comput Biol 2013;11(2):1350001.

[43] Garcia-Martin JA, Dotu I, Clote P. RNAiFold 2.0: A web server and software to design custom and Rfam-based RNA molecules. Nucleic Acids Res 2015;43(W1):W513-W512.

[44] Kleinkauf R, Mann M, Backofen R. antaRNA: Ant colony-based RNA sequence design. Bioinformatics 2015;31(19):3114-3121.

[45] Kleinkauf R, Houwaart, Backofen R, et al. antaRNA—Multi-objective inverse folding of pseudoknot RNA using ant-colony optimization. BMC Bioinformatics 2015;16:389.

[46] Levin A, Lis M, Ponty Y, et al. A global sampling approach to designing and reengineering RNA secondary structures. Nucleic Acids Res. 2012;40(20):10041-10052.

[47] Reinharz V, Ponty Y, Waldispühl J. A weighted sampling algorithm for the design of RNA sequences with targeted secondary structure and nucleotides distribution. Bioinformatics 2013;29(13):i308-i315.

[48] Avihoo A, Churkin A, Barash, D. RNAexinv: An extended inverse RNA folding from shape and physical attributes to sequences. BMC Bioinformatics 2011;12:319.

[49] Shapiro BA. An algorithm for comparing multiple RNA secondary structures. Comput Appl Biosci 1988;4:387-393.

[50] Giegerich R, Voss B, Rehmsmeier M. Abstract shapes of RNA. Nucleic Acids Res. 2014;32(16):4843-4851.

[51] Weinbrand L, Avihoo A, Barash D. RNAfbinv: An interactive Java application for fragment-based design of RNA sequences. Bioinformatics 2013;22(12):2938-2940.

[52] Drory Retwitzer M, Reinharz V, Ponty Y, et al. IncaRNAfbinv: A webserver for the fragment-based design of RNA sequences. Nucleic Acids Res. 2016; 44(W1):W308-W314.

[53] Cohen B, Skiena S. Natural selection and algorithmic design of mRNA. J. Comp. Biol. 2003;10:419-432.

[54] Terai G, Kamegai S, Asai K. CDSfold: an algorithm for designing a protein-coding sequence with the most stable secondary structure. Bioinformatics 2016;32(6):828-834.

[55] De Guire V, Caron M, Scott N, et al. Designing small multiple-target artificial RNAs. Nucleic Acids Res 2010;38(13):e140.

[56] Höner Zu Siederdissen C, Hammer S, Abfalter I, et al. Computational design of RNAs with complex energy landscapes. Biopolymers 2013;99(12):1124-1136.

[57] Lee J, Kladwang W, Lee M, et al. RNA design rules from a massive open laboratory. Proc Natl Acad Sci USA, 111(6):2122-2127.

[58] Yesselman J, Das, R. RNA-Redesign: a web-server for fixed backbone 3D design of RNA. Nucleic Acids Res 2015;43:W498-W501.

[59] Waldispühl J, Devadas S, Berger B, et al. Efficient algorithms for probing the RNA mutation landscape. PLoS Comput Biol 2008;4:e1000124.

[60] Churkin A, Barash D. An efficient method for the prediction of deleterious multiple- point mutations in the secondary structure of RNAs using suboptimal folding solutions. BMC Bioinformatics 2008;9:222.

[61] Zuker M. On finding all suboptimal folding of an RNA molecule. Science 1989;244:48-52.

[62] Wuchty S, Fontana W, Hofacker IL, et al. Complete suboptimal folding of RNA and the stability of secondary structures. Biopolymers 1999;49:145-165.

[63] Dotu I, Lozano G, Clote P, et al. Using RNA inverse folding to identify IRES-like structural subdomains. RNA Biol 2013;10(12):1842-1852.

[64] Drory Retwitzer M, Kifer I, Sengupta S, et al. An efficient minimum free energy structure-based search method for riboswitch identification based on inverse RNA folding. PLOS One 2015;10(7):e0134262.

[65] Nawrocki EP, Burge SW, Bateman A, et al. Rfam 12: Updates to the RNA families database. Nucleic Acids Res 2014;43(D1):D130-D137.

[66] Ruzzo WL, Gorodkin J. De novo discovery of structured ncRNA motifs in genomic sequences. Methods Mol Biol 2014;1097:303-318.

[67] Altschul SF, Madden TL, Schäffer AA, et al. Gapped MPMATH and PSI-MPMATH: a new generation of protein database search programs. Nucleic Acids Res 1997;25(17):3389-3402.

[68] Janssen S, Giegerich R. The RNA shapes studio. Bioinformatics 2015;31(3):423-425.

[69] Ren J, Rastegari B, Condon A. HotKnots: heuristic prediction of RNA secondary structures including pseudoknots. RNA 2005;11(10):1494-1504.

[70] Sato K, Kato Y, Hamada M, et al. IPKnot: fast and accurate prediction of RNA secondary structures with pseudoknots using integer programming. Bioinformatics 2011;27(13):i85-i93.

[71] Lange SJ, Maticzka D, Möhl M, et al. Global or local? Predicting secondary structure and accessibility in mRNAs. Nucleic Acids Res 2012;40(12):5215-5226.

[72] Nawrocki EP, Eddy SR. Infernal 1.1: 100-fold faster RNA homology search. Bioinformatics 2013;29(22): 2933-2935.

Key Points

- RNA design programs should be made user friendly and accessible to biologists as much as possible, both in terms of ease of use and simplification of the input and output such that it becomes understandable to the non-specialist.
- In most cases, a balanced tradeoff between efficiency and performance in terms of the quality of the designed sequences would be the best option for the design.
- From the algorithmic standpoint, the weighted sampling approach to sample the sequence space efficiently and the fragment based design approach are desired directions that can be further developed to yield more flexibility in the design procedure.
- Programs for RNA design should aim to accumulate practical experience in biological meaningful problems, be it experimental design or computational searches for novel non-coding RNAs.

Biographical Notes:

Alexander Churkin
is a computational scientist, a research fellow in the physics department at Ben-Gurion University and an instructor at the Shamoon College of Engineeing. His research interests include computational biology, RNA structure predictions, and scientific computing.

Matan Drory Retwitzer
is a computational scientist who is pursuing his Ph.D. studies in the computer science

department at Ben-Gurion University. His research interests include computational biology, RNA structure predictions and RNA structure probing by experimental methods.

Vladimir Reinharz
is a computational scientist and a PhD graduate of the computer science department at McGill University. He is the recipient of an Azrieli and FQRNT postdoctoral fellowships to be pursued at Ben-Gurion University. His research interests include computational biology, RNA structure predictions, and scientific computing.

Yann Ponty
is a computational scientist and a faculty member of the bioinformatics group in the computer science department (LIX) of École Polytechnique. His research interests include computational biology, RNA folding and visualization, combinatorics and discrete algorithms.

Jérôme Waldispühl
is a computational scientist and a faculty member in the computer science department at McGill University. His research interests include computational biology, RNA and Protein structure predictions, as well as crowd-computing systems.

Danny Barash
is a computational scientist and a faculty member in the computer science department at Ben-Gurion University. His research interests include computational biology, RNA structure predictions, and scientific computing.

Supplementary material

5 Command Line Interfaces

5.1 RNAinverse

The command line interface of `RNAinverse` allows subtle optimizations while the main parameters are inserted upon software request. More advanced options exists for custom alphabet, energy parameters and base pairing. Those would not be discussed here as they are a very rare usecase. For the average user, the following are the ones that will be most used.

`-T` Rescale energy parameters for a given temperature.

`-F` Select the minimization algorithm. `m` for energy minimization or `p` for partition function.

`-R` The number of output sequence to output for the same structure. Negative number will force the software to continue until a perfect match.

`--noGU` Do not allow GU pairs.

`--noClosingGU` Do not allow GU pairs at the end of helices.

Therefore, to find a maximum of 50 solutions, using both partition and energy minimization algorithms, for the structure

```
((((...(((....)))...((((....))))...))))
```

allowing any sequence with a mandatory `GC` base between the first and last nucleotide, at 25 Celsius, the command would be as follows:

```
./RNAinverse -R50 -Fmp -T25
```

Once the software begins, it will request an input structure and starting sequence. Lowercase letters will be forced into the sequence while uppercase will be considered a starting sequence. If no sequence is inserted a random seed sequence will be used.

```
((((...(((....)))...((((....))))...))))
gNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNc
```

5.2 incaRNAfbinv

incaRNAfbinv is a combination of two seperate programs, both have specific command line interfaces. It is recommended for most users to use the webserver as it already combines the two interfaces.

incaRNAtion generates the seed sequences later inserted into RNAfbinv. To run it, a Python distribution must be installed. The command line interface includes many fine tuning parameters while the main structure input must be inserted in a file. The input file must contain a target structure. In addition to the structure, a multiple sequence alignment (MSA) may be added to allow for sequence information. For the average user, the following are the ones that will be most used.

-d The path for an input file containing secondary structure and optional MSA.

-a A number between 0 and 1 used by the algorithm as a weight. 1 takes into account on the structure while 0 only considers the MSA.

-m Maximum penalty for an invalid pair.

-s_gc This is followed by 2 numbers. The first, between 0 and 1, forces a given GC content while se second show the minimal number of output sequences required.

-gc_max_err A number between 0 and 1 with the maximal GC difference between the output sequences and the requested number. 0.1 by default.

-c Sequence constraint to output sequences

To generate at least 50 seed sequences, for the structure

```
((((...(((....)))...((((....))))...))))
```

allowing any sequence with a mandatory GC base between the first and last nucleotide and a GC content of 70%, should be done as follows.

First create a file INPUT_FILE, where INPUT_FILE can be admissible file name later given to the -d option, containing the line:

```
((((...(((....)))...((((....))))...))))
```

Then, call the incaRNAtion script:

```
python IncaRNAtion -d INPUT_FILE -a 1 \
   -m 20 -s_gc 0.7 59 \
   -c GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNC
```

The output will be a list a seed sequences. Those seed sequences can later be insrted into RNAfbinv using the command line.

To use the incaRNAtion seeds, download the RNAfbinv extended version. The package includes a java GUI interface. The command line option allow for the same options as the GUI version. For the average user, the following are the ones that will be most used.

-i The number of simulated annealing iterations for a single sequence design.

-t Look ahead depth: The maximum number of consecutive mutations that generate a lower score sequence possible before a single simulated annealing iteration is over.

-c A starting sequence. This is where the incaRNAtion seed should be inserted.

To generate a single sequence, for the structure

```
((((...(((....)))...((((....))))...))))
```

allowing any sequence , starting from a given incaRNAtion seed, aiming at -23 dG (Kcal/mol) and target mutational robustness 0.8 given 50 iteration and a 4 nucleotide look ahead depth, the command should be a followed:

```
./RNAexinv -i 50 -t 4 -c <incaRNAtion seed>
```

Once the programs starts the following parameters should be inserted:

```
((((...(((....)))...((((....))))...)))) \
-23 \
0.8
```

5.3 RNAiFold

The command line interface of RNAiFold has over 50 options allowing for an extremely fine tuning of the desired output. For the average user, the following are the ones that will be most used. There is two way to enable those options, or through a file, where the option name is on a line preceded by a # instead of a -, followed on the next line by the desired option. Usually, the option can be simply given as argument on the command line.

-RNAscdstr The target structure. Multiple target can be set, they must be on the same line separated by the pipe | symbol. The structures must have the same length.

-RNAseqcon The admissible sequences, in IUPAC format. It must be one string the same length as the structure.

-maxGCcont The maximal GC content admissible in the sequences.

-minGCcont The minimal GC content admissible in the sequences.

-TimeLimit The amount of time allowed to run (default 600 seconds).

-MAXsol The maximum number of solutions to be reached under the time limit

Therefore, to find a maximum of 50 solutions, under an hour, for the structure

```
((((...(((....)))...((((....))))...))))
```

allowing any sequence with a mandatory GC base between the first and last nucleotide, and with a GC content between 60% and 70%, the command would be as follows:

```
./RNAiFold -RNAscdstr "((((...(((....)))...((((....))))...))))" \
           -RNAseqcon GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNC \
           -maxGCcont 70 -minGCcont 60 -MAXsol 50 -TimeLimit 3600
```

5.4 antaRNA

The distribution of antaRNA as a Python2.7 executable and all options are given as arguments. A similar ensemble of constraints exists.

-Cstr The target structure in the dot bracket notation. A fuzzy notation can be used to define blocks allowed to base pair together using any lowercase and uppercase letter.

-Cseq The admissible sequences, in IUPAC format. It must be one string the same length as the structure.

-tGC Target GC content, in $[0, 1]$, which also serves as a minimum.

-tGCmax Maximal GC content admissible in the sequences.

-tGCvar Variance (σ^2) in the case of normal distribution, -tGC serves as the expected value μ.

-t The amount of time allowed to run (default 600 seconds).

-n Number of solutions to be produced.

Therefore, to find a maximum of 50 solutions, under an hour, for the structure

((((...(((....)))...((((....))))...))))

allowing any sequence with a mandatory GC base between the first and last nucleotide, and with a GC content between 60% and 70%, the command would be as follows:

```
python antaRNA.py -Cstr "((((...(((....)))...((((....))))...))))" \
            -Cseq GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNC \
            -tGC 0.6 -tGCmax 0.7 -n 50 -t 3600
```

Pseudoknotted structures can be considered with the -p parameter if pKiss_mfe or HotKnots or IPKnot is installed,.

In addition, all parameters of the ant colony search algorithms can be directly modified through the command line, from the random seed to initiate the search -s, the number of ants exploring (-aps, default 10), the pheromone evaporation rate(-er, default 0.2), and a wealth of others.

5.5 NUPACK

The NUPACK program provides an ensemble of tools, design being the application for inverse folding. It has less options than the previous programs but with his focus for designing long sequences viable *in vitro*, it can extrapolate the energy parameters for a given concentration of sodium and magnesium.

The program loads the target structure and admissible sequences, in IUPAC format, from a file PREFIX.fold. The PREFIX can be any name chosen by the user but the extension .fold must be given. Additional parameters are:

-material which can be set as rna1995 to use Turner95 energy or rna1999 for Mathews99 energy parameters.

-sodium The sodium concentration.

-magnesium The magnesium concentration.

-prevent The name of a file, which can contain one subsequence per line forbidden in the design.

-loadseed PREFIX.init A file containing one number, the random seed to be used. Each execution of the software will choose a different random seed, but the program is deterministic and will always return the same output for a given seed. Note that the name of the file *must* be the same as the one with the target sequence, followed by the extension .init.

To design any sequence with a mandatory GC base pair between the first and last nucleotide, for the structure

```
((((...(((....)))...((((....))))...))))
```

using Turner95 energy parameters should be done as follows.

First create a file PREFIX.fold, where PREFIX can be admissible file name, containing the two lines:

```
((((...(((....)))...((((....))))...))))
GNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNNC
```

Then, call the function design.

```
./design -material rna1995 PREFIX
```

Note that the suffix .fold is not given. To generate a different sequence launch the program again.