Vol. 29 no. 22 2013, pages 2938–2940
doi:10.1093/bioinformatics/btt494

Structural bioinformatics

Advance Access publication August 23, 2013

RNAfbinv: an interactive Java application for fragment-based design of RNA sequences

Lina Weinbrand[1], Assaf Avihoo[2] and Danny Barash[1,*]

[1]Department of Computer Science, Ben Gurion University of the Negev, Beer Sheva 84105, Israel and [2]Microsoft Research Israel, Herzliya 46733, Israel

Associate Editor: Ivo Hofacker

ABSTRACT

Summary: In RNA design problems, it is plausible to assume that the user would be interested in preserving a particular RNA secondary structure motif, or fragment, for biological reasons. The preservation could be in structure or sequence, or both. Thus, the inverse RNA folding problem could benefit from considering fragment constraints. We have developed a new interactive Java application called RNA fragment-based inverse that allows users to insert an RNA secondary structure in dot-bracket notation. It then performs sequence design that conforms to the shape of the input secondary structure, the specified thermodynamic stability, the specified mutational robustness and the user-selected fragment after shape decomposition. In this shape-based design approach, specific RNA structural motifs with known biological functions are strictly enforced, while others can possess more flexibility in their structure in favor of preserving physical attributes and additional constraints.

Availability: RNAfbinv is freely available for download on the web at http://www.cs.bgu.ac.il/∼RNAexinv/RNAfbinv. The site contains a help file with an explanation regarding the exact use.

Contact: dbarash@cs.bgu.ac.il

Received on March 11, 2013; revised on June 28, 2013; accepted on August 19, 2013

1 INTRODUCTION

The inverse RNA folding problem for designing sequences that fold into a given RNA secondary structure was introduced in the early 1990s in Vienna (Hofacker *et al.*, 1994). The approach to solve it 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 (Hofacker, 2003) or mfold (Zuker, 2003), by performing energy minimization with thermodynamic parameters (Mathews *et al.*, 1999). 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 properties according to the objective function in the optimization problem formulation. Performance improvements of this approach have been worked out in INFO-RNA (Busch and Backofen, 2006), RNA-SSD (Aguirre-Hernández *et al.*, 2007) and NUPACK:Design (Zadeh *et al.*, 2011). Some general extensions of the problem that are evolutionary motivated were suggested in Dromi *et al.* (2008) and were carried out using evolutionary computation without efficiency considerations. Other trials to solve related problems, such as the design of RNA with multiple target structures, include Frnakenstein (Lyngsø *et al.*, 2012). Some interesting analogies to the coding theory were described in the study by Heitsch *et al.* (2003). Recently, a constraint programming approach was developed in RNAiFOLD (Garcia-Martin *et al.*, 2013), and a global sampling approach was introduced in RNA-ensign (Levin *et al.*, 2012). Remaining within the framework of a local search strategy, using a coarse-grain tree-graph representation of the RNA secondary structure, we extended the inverse RNA folding problem to include constraints such as thermodynamic stability and mutational robustness, developing a program called RNAexinv (Avihoo *et al.*, 2011). The program improved the efficiency with respect to the study by Dromi *et al.* (2008), but it had no preference to a particular RNA secondary structure motif that the user would like to preserve.

To motivate the need for enabling the user to specify a preference for particular RNA secondary structure motifs that the practitioner would like to preserve, one can think of an example from the area of RNA viruses. It is well known that particular hepatitis C virus stem-loops are important for virus replication (e.g. Yi and Lemon, 2003). In the future, the functional importance of more such RNA structural motifs in diverse types of RNAs will be revealed. The ability of the user to design sequences according to such given information should become useful.

RNAfbinv (RNA fragment-based inverse) is an interactive Java application that allows the user to insert an RNA secondary structure and design sequences based on user-selected fragments by solving an extended inverse RNA folding problem. Given an RNA secondary structure, the RNAfbinv program decomposes the structure into fragments that are RNA secondary structure motifs. The user can then select to preserve certain fragments that are presumed to be of biological importance, after which the program solves the extended inverse RNA folding problem by considering the additional constraints of mutational robustness and thermodynamic stability (Avihoo *et al.*, 2011). In addition, nucleotide constraints on the sequence can be inserted. The decomposition into structural motifs based on coarse-grain tree-graphs (Shapiro, 1988) allows a bottom-up approach in which the RNA-designed sequences are constructed from the fragments, from sequence constraints and from evolutionary and physical attributes. The first use of coarse-grain tree-graphs in this context was suggested in the study by Dromi *et al.* (2008), in

*To whom correspondence should be addressed.

which the physical attributes were inserted as constraints, inspired by the review of Higgs (2000) about the physical aspects of RNA secondary structure. RNAfbinv is a natural extension of the coarse-graining idea that uses the coarse-grain tree-graph representation to decompose the structure into fragments. The user can then select structure constraint fragments that should be preserved, and together with user-selected sequence constraint fragments, they comprise building blocks on which the sequences are designed. For coarse-graining, we have chosen to work with tree-graphs (Shapiro, 1988), but similarly, one can use abstract shapes (Giegerich *et al.*, 2004).

The application consists of three separate programs: RNAattributes, RNAfbinv and RNAfbinv-flex that supports flexible length-designed sequences. The RNAattributes program is an aid tool that accepts an RNA sequence and provides its predicted minimum free energy secondary structure and its minimum free energy and mutational robustness. The RNAfbinv program is the main program that accepts an RNA structure, thermodynamic stability, mutational robustness and a choice of user-selected sequence and structure fragments based on coarse-grain tree-graph representation. The output consists of designed sequences that are ranked by the base pair distance of their predicted structure from the input structure. The RNAfbinv-flex program is an extension to RNAfbinv that also outputs designed sequences but of variable lengths. Because of this flexibility, RNAfbinv-flex does not accept sequence constraints in the current implementation, as the locations of sequence constraints become obscured when the sequences vary in size. In the future, more sophisticated programs could perhaps allow inserting sequence constraints in RNAfbinv-flex, but this requires special development. In the current implementation of RNAfbinv-flex, up to 2 nt are deleted, and it is possible to consider sequence constraints by filtering the results obtained in a post-processing step.

2 METHODOLOGY

RNAfbinv uses a similar methodology to RNAexinv (Avihoo *et al.*, 2011) except for a major difference in the objective function to be minimized. In general, the solution method of RNAfbinv comprises two phases: (1) identify a good initial candidate: if the motif does not exist in the initial candidate, it is re-chosen and (2) simulated annealing with a 4-nt look-ahead local search function. The two phases are elaborated in more detail as follows. For the first phase, it was found for our needs that using RNAinverse (Hofacker *et al.*, 1994) with a random start point is preferable over a deterministic first stage like the one used in INFO-RNA (Busch and Backofen, 2006). A random start point will produce different starting sequences for the extended search rather than a fixed starting point. The purpose is to avoid repeatedly reaching the same local minimum each time. In the second phase, as described in more detail for RNAexinv (Avihoo *et al.*, 2011), iterative mutating is performed to search for local minima and a simulated annealing approach with a 4-nt look-ahead local search function is used to sample the vicinity of the sequence (the default look ahead is 4, not exhaustive, only sampling). In the objective function, an extra term relative to RNAexinv that outputs a binary value is added in RNAfbinv to account for the existence (or inexistence) of a user-selected motif:

$$\begin{aligned}
&f(\mathit{initial}, \mathit{target}) = \\
&|\mathit{neutrality}_{\mathit{target}} - \mathit{neutrality}_{\mathit{initial}}| * 100 + |\mathit{dG}_{\mathit{target}} - \mathit{dG}_{\mathit{initial}}| \\
&+ \mathit{target\_motif\_exists}(\mathit{initial}) * 1000 \\
&+ \mathit{tree\_edit\_distance\_shapiro\_representation}(\mathit{target}, \mathit{initial}) * 100 \\
&+ \mathit{base\_pair\_distance\_dotBracket\_representation}(\mathit{target}, \mathit{initial}) * 0.01
\end{aligned}$$

The extra term for the target motif existence is the most important constraint that should be fulfilled exactly without any compromise. Therefore, a much larger weight of 1000 relative to all others is chosen for this term. The other terms are the same as in RNAexinv (Avihoo *et al.*, 2011) and are described as follows. The neutrality for measuring mutational robustness is a number between 0 and 1; therefore, a weight of 100 is assigned. The minimum free energy dG is for measuring thermodynamic stability in kilocalorie per mole; therefore, a unity weight is assigned. The distances between secondary structures are calculated using RNAdistance in the Vienna RNA package (supporting both the coarse-grain tree-graphs that are called Shapiro representation, and the dot bracket representation of the secondary structure). For the tree edit distance between Shapiro representations, a relatively large weight of 100 is chosen for shape preservation, whereas for the base pair distance in the last term, a very small weight of 0.01 is assigned. This last term is the one used in the original RNAinverse for preserving the exact secondary structure, and its purpose here is to protect the solutions from being too shape-based dominated, as explained in Avihoo *et al.* (2011).

3 IMPLEMENTATION

RNAfbinv is currently available on a Linux platform (Ubuntu, OpenSuse and Fedora were checked), Mac and Windows. All preparations and compilations should be performed with Java and ‘GNU CC’ compiler installed. The package content includes programs from the Vienna RNA package. RNAfbinv is free under the GNU license.

Detailed instructions on preparation and compilation are available in the ReadMe file that can be easily accessed from the web pointer. In the input screen, the user inserts a structure in dot-bracket notation and on pressing ‘Fragment’, the program performs structure decomposition into fragments. The next screen presented in Figure 1 appears, which then lets the user select the structural motif (fragment) that should be preserved by scrolling a combo-box dropdown list that contains all the structural motifs. After the user selects a structural motif, on pressing ‘Process’ in the input screen where the structure in dot-bracket was given, a new window appears with the dot-bracket at the top followed by physical parameters and nucleotide constraint information that can easily be filled. To summarize, the list of input data and parameters contains:

(1) Input structure in dot-bracket notation.

(2) User-selected structural motif (fragment).

(3) Physical parameters: desired free energy in kilocalorie per mole and desired neutrality, which is a number between 0 and 1.

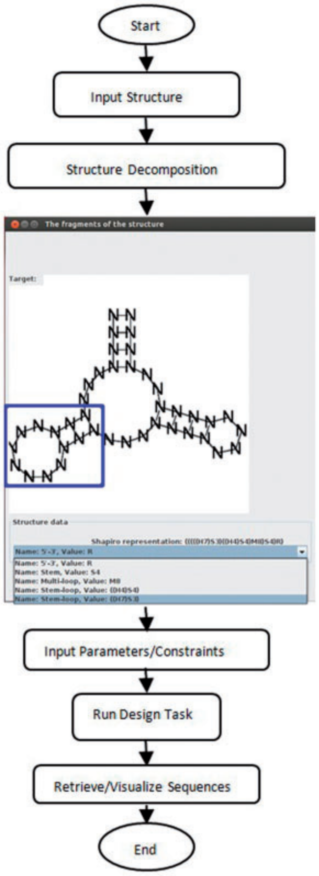
Start
Input Structure
Structure Decomposition
The fragments of the structure
Target:
Structure data
Shapiro representation: ((((H7)S3)(H4)S4)M8)S4)R)
Name: 5'-3', Value: R
Name: 5'-3', Value: R
Name: Stem, Value: S4
Name: Multi-loop, Value: M8
Name: Stem-loop, Value: (H4)S4
Name: Stem-loop, Value: (H7)S3
Input Parameters/Constraints
Run Design Task
Retrieve/Visualize Sequences
End

Fig. 1. Structural decomposition to motifs and selection screen

(4) Nucleotide constraints: desired nucleotide(s), start location and a possibility to allow a single-nucleotide slippage.

On pressing 'Results', execution of the program starts. The output screen with 20 different possible designed sequences appears. The user can press a button below each sequence to obtain useful information. The entire work flow is illustrated in Figure 1.

Computational time depends on the sequence length that the user selects as input. For the example in Figure 1, run-time is ~10 min on a standard PC. The default number of iterations is set to 100. On average, RNAfbinv is 10% more costly than RNAexinv (Avihoo *et al.*, 2011). It is possible to reduce computational time by ~6% when disregarding the non-structural constraints.

4 RESULTS

The output from RNAattributes, RNAfbinv and RNAfbinv-flex can be easily checked by running RNAfold (Hofacker, 2003) or mfold (Zuker, 2003) on the designed sequences.

To test whether the inverse problem solutions obtained from our program meet the desired constraints convincingly, we ran it on artificial examples as well as on natural ones. For example, on an miRNA precursor example that was taken from Krol *et al.* (2004) and illustrated in Dromi *et al.* (2008) and Avihoo *et al.* (2011), we were able to obtain designed sequences that fulfill all constraints imposed and full preservation of the selected fragments.

5 FUTURE DEVELOPMENTS

The basic implementation of RNAfbinv is generic and can be applied to solve a variety of RNA design problems. It can also be used to search RNA patterns that are more general than sequences using sequence-based searches, by selecting RNA-designed sequences that conform to several constraints. In the future, it will be possible to extend RNAfbinv to handle more constraints and also to improve its efficiency by using more sophisticated sampling approaches, e.g. Levin *et al.* (2012) and by other means.

ACKNOWLEDGEMENTS

The authors thank Alexander Churkin and Idan Gabdank for their technical assistance.

Funding: The Lynne and William Frankel Center for Computer Sciences, Ben-Gurion University.

Conflict of interests: None declared.

REFERENCES

Aguirre-Hernández,R. *et al.* (2007) Computational RNA secondary structure design: empirical complexity and improved methods. *BMC Bioinformatics*, **8**, 34.

Avihoo,A. *et al.* (2011) RNAexinv: an extended inverse RNA folding from shape and physical attributes to sequences. *BMC Bioinformatics*, **12**, 319.

Busch,A. and Backofen,R. (2006) INFO-RNA-a fast approach to inverse RNA folding. *Bioinformatics*, **22**, 1823–1831.

Dromi,N. *et al.* (2008) Reconstruction of natural RNA sequences from RNA shape, thermodynamic stability, mutational robustness, and linguistic complexity by evolutionary computation. *J. Biomol. Struct. Dyn.*, **26**, 147–162.

Garcia-Martin,J.A. *et al.* (2013) RNAiFOLD: a constraint programming algorithm for RNA inverse folding and molecular design. *J. Bioinform. Comput. Biol.*, **11**, 1350001.

Giegerich,R. *et al.* (2004) Abstract shapes of RNA. *Nucleic Acids Res.*, **32**, 4843–4851.

Heitsch,C.E. *et al.* (2003) From RNA secondary structure to coding theory: a combinatorial approach. In: *Proceedings of the 8th international meeting on DNA based computers,* Vol. 2568, *Lecture Notes in Computer Science*, Springer-Verlag, pp. 215–228.

Higgs,P.G. (2000) RNA secondary structure: physical and computational aspects. *Q. Rev. Biophys.*, **33**, 199–253.

Hofacker,I.L. *et al.* (1994) Fast folding and comparison of RNA secondary structures. *Monatsh. Chem.*, **125**, 167–188.

Hofacker,I.L. (2003) Vienna RNA secondary structure server. *Nucleic Acids Res.*, **31**, 3429–3431.

Krol,J. *et al.* (2004) Structural features of microRNA (miRNA) precursors and their relevance to miRNA biogenesis and small interfering RNA/short hairpin RNA design. *J. Biol. Chem.*, **279**, 42230–42239.

Levin,A. *et al.* (2012) A global sampling approach to designing and reengineering RNA secondary structures. *Nucleic Acids Res.*, **40**, 10041–10052.

Lyngsø,R.B. *et al.* (2012) Frnakenstein: multiple target inverse RA folding. *BMC Bioinformatics*, **13**, 260.

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

Shapiro,B.A. (1988) An algorithm for comparing multiple RNA secondary structures. *Comput. Appl. Biosci.*, **14**, 387–393.

Yi,M. and Lemon,S.M. (2003) Structure-function analysis of the 3' stem-loop of hepatitis C virus genomic RNA and its role in viral RNA replication. *RNA*, **9**, 331–345.

Zadeh,J.N. *et al.* (2011) Nucleic acid sequence design via efficient ensemble defect optimization. *J. Comput. Chem*, **32**, 439–452.

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