RNAiFold: A web server for RNA inverse folding and molecular design

Juan Antonio Garcia-Martin, Peter Clote*, Ivan Dotu

Department of Biology, Boston College, Chestnut Hill, MA 02467, USA,

Running title: RNAiFold web server

Key words: RNA, inverse folding, molecular design, synthetic biology, RNA secondary structure.

* Corresponding authors: Email: clote@bc.edu Phone: (617) 552-1332, Fax: (617) 552-2011.

Abstract

Synthetic biology and nanotechnology are poised to make revolutionary contributions to the 21st century. In this paper, we describe a new web server to support *in silico* RNA molecular design. Given an input *target* RNA secondary structure, together with optional constraints, such as requiring GC-content to lie within a certain range, requiring the number of strong (GC), weak (AU), and wobble (GU) base pairs to lie in a certain range, the `RNAiFold` web server determines one or more RNA sequences, whose minimum free energy secondary structure is the target structure. `RNAiFold` provides access to two servers: `RNA-CPdesign`, which applies *constraint programming*, and `RNA-LNSdesign`, which applies the *large neighborhood search* heuristic, and hence is suitable for larger input structures. Both servers can also solve the RNA inverse hybridization problem; i.e. given a representation of the desired hybridization structure, `RNAiFold` returns two sequences, whose minimum free energy hybridization is the input target structure.
Availability: The web server is publicly accessible at `http://bioinformatics.bc.edu/clotelab/RNAiFold`, which provides access to two specialized servers: `RNA-CPdesign` and `RNA-LNSdesign`. Source code for the underlying algorithms, implemented in COMET and supported on linux, can be downloaded at the server web site.

1 Introduction

Given a target RNA secondary structure, the *RNA inverse folding problem* consists of determining one or more sequences, whose minimum free energy (MFE) structure, with respect to the Turner energy model **?**, is the target structure. `RNAiFold` is a web server, that provides access to two new algorithms, `CPdesign` and `LNSdesign`, that solve the RNA inverse folding problem. These algorithms are described in **?**, where applications are given, along with benchmarking studies against other methods.

Although there are other approaches for solving RNA inverse folding problem, such as **??????**, all of these algorithms can be classified as heuristic methods, which start with an initial sequence that is iteratively modified until it either folds into the target structure or some stopping criterion is reached. In contrast, `CPdesign` uses *Constraint Programming* (CP) to provide the first *complete* solution of inverse folding, capable of generating all solutions, as well as of determining that there are no solutions to a given inverse folding problem. `LNSdesign` uses CP to solve subproblems, but employs the *Large Neighborhood Search* (LNS) heuristic, to tackle larger inverse folding problems; however, by using a heuristic, `LNSdesign` can no longer generate all solutions, or determine that there is no solution to a given inverse folding problem. `RNAiFold` represents a step forward in RNA design, since both `CPdesign` and `LNSdesign` allow the user to specify a wide range of design constraints (both nucleotide and base-pairing constraints) in solving inverse folding. Both `CPdesign` and LNSdesign allow the user flexibility in specifying design constraints, such as sequence template for stipulating the occurrence of certain nucleotides at certain positions, GC-content, upper and lower bounds for base pairs of certain types (strong-GC, weak-AU, and wobble-GU), the maximum number of consecutive nucleotides (e.g. disallowing runs of five or more occurrences of 'A'), as well as requiring the returned sequence to be compatible (or not to be compatible) with certain base pairs located at certain positions. Additionally, the user can stipulate folding temperature and treatment of stacked, single-stranded nucleotides (also known as *dangles*). Both `CPdesign` and `LNSdesign` allow for *inverse co-folding* for a target hybridization structure for two RNAs, i.e. determining two RNA sequences, whose minimum free energy hybridization structure is the given target structure. The supplementary data includes:

1. Figure of the web server input page with file upload selected.
2. Figure of input page with paste input expanded.
3. Figure of input page with verbose input expanded.
4. Definitions of all structural measures.

2 Web server

`RNAiFold` is a web server, that provides access to two tools: `RNA-CPdesign`, which applies Constraint Programming (CP), and `RNA-LNSdesign`, which applies Large Neighborhood Search (LNS). The web server currently allows jobs to run for at most 30 minutes. For design problems requiring more computation time, the user is urged to download and locally run our software. As explained, `RNA-CPdesign` and `RNA-LNSdesign` provide somewhat different functionality; however, the input and output format is identical for both, hence will be treated together. Throughout the paper, when we mention the input and output format of `LNSdesign`, the reader should realize that the same is true of `CPdesign`.

Input

Though not required, it is best for the user to enter an email address. For jobs that require long computation time, the results will automatically be sent to the user, if the email address is present; in any case, the browser window will be provided a link to open later when the computation has terminated.

Figure S1 depicts the input form for both web servers `CPdesign` and `LNSdesign`. The user may upload a file, paste the input (Figure S2), or choose to fill in text fields in a more verbose manner (Figure S3).

Below, we discuss in detail the input format for file uploads and for pasting in the large text field. After this discussion, it will be clear how to fill in fields for the more verbose manner of entering a query. `RNAiFold` solves the RNA inverse folding problem, so the mandatory minimal query is a desired target secondary structure. The target structure may be pasted into the text field, or a file containing the target structure may be uploaded, or entered into the verbose form. Optional FASTA comment and optional *sequence constraints* may be included. The input must be in the following form:

- > FASTA comment (optional)
- sequence constraints (optional)
- target structure in dot bracket notation (required)

The server does not require sequence constraints to appear before the target structure, and any permutation of the above three lines is acceptable. Note that if sequence constraints are given, then the input sequence length must be the same as that of the structure. An example is given below.

Example: The target structure is that of a transfer RNA along with FASTA comment and sequence constraints, as follows:

```
> tRNA structure with sequence constraints
GAGCUUGNNNNNNNNNNNNNNNNNNNNNNNNNNNACGNNNNNNNNNNNNNNNNNNNNNNNNNNNNCUGGCUCG
(((((.(..((((.........)))).(((((.......)))))).....(((((.......))))))..)))))
```

Note the presence of 'GAGCUUG' in positions 1-7, 'ACG' in positions 35-37 and 'CUGGCUCG' in positions 67-74. The sequence constraints ensure that any sequence returned by `LNSdesign` have the stipulated nucleotides at the indicated positions ('N' mean any nucleotide). Allowable sequence constraints are all 1-character IUPAC codes.

In this example, the desired target structure is the cloverleaf structure of a transfer RNA, and the sequence constraints ensure an 'ACG' anticodon, corresponding the the CGU codon for arginine. As well, the first seven 'GAGCUUG' last eight nucleotides 'CUGGCUCG', together comprise the *acceptor stem*, known to be critical for proper binding of aminoacyl synthetase. The sequence constraints and structure come from the sequence and consensus structure the Rfam database `http://rfam.sanger.ac.uk/` for *Chlamydomonas reinhardtii* chloroplast transfer RNA, with EMBL accession code L13782.1/442-515. This example is interesting, since the Rfam consensus secondary structure of L13782.1/442-515 is different from its minimum free energy structure, which is not a cloverleaf structure. The output of `LNSdesign` is discussed below, in the section on *Output* and depicted in Figure **??** .

`RNA-LNSdesign` allows the user to give additional constraints on all returned sequences. The constraints are the following.

- Limit canonical base pairs: Minimum and maximum number of base pairs in the MFE structure of any returned sequence. Values can be stipulated for strong (GC), weak (AU) and wobble (GU) pairs. Default values are 0 for the minimum and - (no limit) for the maximum.
- Limit consecutive nucleotides: Minimum and maximum number of consecutive nucleotides of each type (A,C,G,U) in all returned sequences. Note that the value 0 (zero) indicates that the MFE structure (target structure) of any returned sequence *cannot* contain any base pair of the specified type. Default - (no limit) for the maximum.
- Limit GC content: Minimum and maximum percent of GC-content (number of either G or C nucleotides divided by sequence length). Default is 0% for minimum and 100% for maximum.

Compatible base pairs: `RNA-LNSdesign` allows the user to specify those positions in the returned sequence(s) that *can* form a base pair, even if these base pairs are not part of the target structure. In this fashion, one could design an RNA whose MFE structure is the given target structure, but which is compatible with

another structure. To specify these positions, the input (pasted in the text field, or uploaded in a file, or specified in the verbose form) should include the desired 'compatible' structure, given in dot bracket notation, preceded by the 'at' symbol ('@').

Incompatible base pairs: `RNA-LNSdesign` allows the user to specify positions in the returned sequence(s) that *cannot* form a base pair. To specify these positions, the input (pasted in the text field, or uploaded in file, or specified in the verbose form) should include the disallowed base pairs, given in dot bracket notation, preceded by the 'pound' symbol ('#'). Note that if disallowed base pairs appear in the target structure, then `LNSdesign` will return no solutions.

Verbose Form: An alternate input format is given in the verbose form, depicted in Figure S3. Here, the user may enter the formerly described target structure, sequence constraints, and constraints for compatible and incompatible base pairs. Note that no leading symbol '@' or '#' is required, respectively, for compatible and incompatible base pairs, and that any field may be left blank, with the exception of the field for the target structure.

In addition, the user can stipulate the temperature at which folding occurs (default is 37° C), and treatment of dangles. Since `LNSdesign` uses a COMET plug-in for Vienna RNA Package, the treatment of dangles (stacked, single-stranded nucleotides) follows that of the Vienna Package – the user is referred to Vienna Package web site for further explanation. Default dangle treatment for `LNSdesign` is 1, which is the default for Vienna Package `RNAfold`. However, the user can choose dangle treatment 0 (no dangle), 1 (default for Vienna Package minimum free energy structure computation), 2 (default for Vienna Package partition function computation), 3 (Vienna Package treatment of dangles and coaxial stacking).

Inverse cofolding: `RNA-LNSdesign` allows for *cofolding*: given a hybridization structure, together with optional constraints, `LNSdesign` returns two sequences, whose MFE hybridization is the input, target structure. Note that a hybridization may involve inter-molecular base pairs, as well as intra-molecular base pairs, provided that there are no pseudoknots.

The input format for the RNA inverse cofolding problem consists of the following form:

- > FASTA comment (optional)
- target hybridization structure, with ampersand ('&') (required)
- sequence constraints with ampersand ('&') (optional)

The target hybridization structure is represented in *dot bracket notation*, following the convention of Vienna RNA Package `RNAcofold`. Namely, an ampersand ('&') separates the two single-molecule portions of the hybridization structure. Sequence constraints are represented by the concatenation of two sequences, separated by ampersand, where as before, the user may indicate that certain positions in all returned sequences must have specific nucleotides in indicated positions. The sequence constraint is optional, but if provided, it must have the same length as the input hybridization structure including the ampersand.

In order to represent the hybridization of an unknown sequence of 20 nt with another unknown sequence of 10 nt, where there is a stem-loop structure from positions 1-10 for the unknown sequence of 20 nt, while positions 11-20 of the first sequence hybridize with positions 1-10 of the second sequence, we give the dot bracket notation (with ampersand): `(((....))).((((((((((&))))))))).`

Output

If the user gives an email address, then after submitting the job, an email will be sent as soon as the results are available. Three possible types of results can be returned:

1. Solution found: A file of results will be attached, with the following information:
    - Input target structure
    - A listing of the sequence(s) found.
    - The numbers of base pairs of each type (strong, weak and wobble).

- Computation time
- Structural measures

2. No solution found: time is limited to 30 minutes, so a solution may not be found within this limit.

3. No possible solution: If the target structure (with specified constraints) has no solution and the time limit has not been reached, then `LNSdesign` will state this in the results file.

On the earlier-explained example of a target transfer RNA structure with the sequence constraints of 'GAGCUUG' in positions 1-7, anticodon 'ACG' in positions 35-37, 'CUGGCUCG' in positions 67-74, `LNSdesign` returns an RNA sequence, whose MFE structure is the target structure, an image of the sequence and structure, such as depicted in the right panel of Figure **??**:

Figure 1: *(Left)* Results page for sequence determined by `LNSdesign` in 22 seconds, whose MFE is the input target structure. The output sequence meets the user-defined constraints of 'GAGCUUG' in positions 1-7, 'ACG' in positions 35-37 and 'CUGGCUCG' in positions 67-74. These constraints ensure that the acceptor stem and anticodon are identical to those of *Chlamydomonas reinhardtii* chloroplast transfer RNA, with EMBL accession code L13782.1/442-515. *(Left)* Results page of a target hybridization structure of an unknown sequence of 20 nt with another unknown sequence of 10 nt, given in dot bracket notation (with ampersand), by `(((....))).((((((((((&))))))))))`. Note that Varna cannot display hybridization structures and thus, a concatenated structure is presented.

Here, the objective structure and sequence pattern constitute the user-defined target structure and sequence constraints (if any). The solution sequence is then given, followed by its minimum free energy (MFE) structure – which must be the target structure. The number of strong (GC), weak (AU) and wobble (GU) pairs are given, followed by CPU computation time in seconds. Free energy of the MFE structure is given in kcal/mol, followed by the *expected base pair distance* from the target structure, the *ensemble defect*, which is another measure of distance between low energy structures in the Boltzmann ensemble and the target structure, the *average pointwise entropy* and two measures of structural diversity. All of these values provide information on the extent to which the Boltzmann ensemble of low energy structures for the output RNA sequence resemble the target structure. These measures are explained in **?** and in supplementary data S4.

Cluster and software specs

The web server `http://bioinformatics.bc.edu/clotelab/RNAiFold` runs on a Linux cluster with head and file server nodes, and 25 compute nodes, including 6 Dell Power Edge 1850, 2 x Intel Xeon P4 (2.80 GHz), 2 GB RAM, 11 Dell Power Edge 1750, 2 x Intel Xeon P4 (2.80 GHz), 4 GB RAM, and 8 Dell Power Edge 1950, 2 x Intel Xeon E5430 Quad core (2.80 GHz), 16 GB RAM. The `CPdesign` and `LNSdesign` software is written in version 2.1.1 of COMET, an optimization programming language available at `http://dynadec.com/support/downloads/`. `CPdesign` and `LNSdesign` employ a plug-in to version 1.8.5 of Vienna RNA Package, which employs the Turner 99 energy model. In the future, it is likely that `RNAiFold` later migrate to version 2.* of Vienna RNA Package, which employs the Turner 2004 energy model.

Comparison with other software and web servers

Table 1 compares current inverse folding software and web servers. To the best of our ability, we attempt to compare various aspects of existent RNA inverse folding software and web servers. This comparison table is intended only to provide an overview of how `RNAiFold` differs from existent software.

Our CP algorithm, `RNAiFold` **?**, whose source code is publicly available, can output millions of solution sequences; nevertheless, the `RNAiFold` web server is restricted to output up to 50 solution sequences. `RNAiFold` supports IUPAC sequence constraints, allows the user to stipulate GC-content, the number of GC, AU and GU base pairs, as well as the maximum number of successive nucleotides, such as the size upper bound of poly-A regions. `RNAiFold` additionally supports compatibility and incompatibility structural constraints, as described in the text, and can solve inverse folding for the hybridization of two structures.

Software	WS	PK	H	T	D	SeqC	StrC	O	Num
RNAiFold	✓	—	✓	✓	0,1,2,3	✓	✓	mfe	50
RNAinverse	✓	—	—	✓	0,1,2,3	IUPAC⋆	—	mfe, prob	100
RNA-SSD	✓	—	—	✓	3?	IUPAC⋆	—	mfe	10
INFO-RNA	✓	—	—	—	3?	IUPAC	—	mfe,prob	50
NUPACK	✓	—	✓*	✓	0,1,2	✓	—	ens def	10
MODENA	—	✓	—	—	?	—	—	mfe,prob	?
Inv	—	✓	—	—	?	—	—	mfe	?
Frnakenstein	—	—	—	✓	3?	—	✓	various	?

Table 1: Comparison table for RNA inverse folding software and web servers. Column titles: Soft (Software), WS (web server), PK (pseudoknots), H (hybridization), T (temperature), D (dangle), SeqC (sequence constraints), StrC (structural constraints), O (optimization strategy), Num (maximum number of sequences returned). Note that `NUPACK` optimization strategy uses ensemble defect (ens def) **?**, and there is an asterisk after '✓', to indicate that by an algorithmic *tour de force*, hybridization of more than two structures is supported. Note also that IUPAC⋆ refers to a limited subset of IUPAC (and non-IUPAC) symbols as explained in the text describing each software above.

Historically, `RNAInverse ??` is the first method described in the literature. It is available as part of the Vienna RNA Package web server `http://rna.tbi.univie.ac.at/cgi-bin/RNAinverse.cgi`. `RNAinverse` relies on a simple greedy strategy called *adaptive walk*, allows only limited sequence constraints restricted to A,U,C,G,T,X,K,I, as well as prohibiting GU wobble pairs, but does not allow structural constraints. The `RNAinverse` algorithm may strive to determine a sequence, whose minimum folding energy (MFE) structure is the target structure (objective is *mfe*), or alternatively, strive to determine a sequence, which takes on the target structure with probability greater than a user-defined parameter, with default 0.5 (objective is *prob*).

`RNA-SSD ?`, which is available through the web server `RNA Designer` at `http://www.rnasoft.ca/cgi-bin/RNAsoft/RNAdesigner/rnadesign.pl`, uses a sophisticated initialization procedure to choose an initial RNA sequence, and applies *stochastic local search* in place of an adaptive walk. Sequence constraints are limited to A,C,G,U,R,Y,N,X; no structural constraints are supported. In addition, GC-content for base-paired regions and (independently) for loop regions can be stipulated. No choice of dangle treatment is allowed; however, since `RNA-SSD` relies on `RNAfold` from the Vienna RNA Package, it seems likely the dangle treatment is that of `-d3`, which includes coaxial stacking.

The program, `INFO-RNA ?`, employs a novel initialization step, which uses a dynamic programming algorithm to choose a sequence having the lowest free energy among all sequences compatible with the target structure. `INFO-RNA ?` is available via web server at `http://rna.informatik.uni-freiburg.de:8080/INFORNA/Input.jsp`. The web server supports sequence constraints in the form of IUPAC symbols, and allows the user to stipulate an upper bound on the number of constraint violations.

The program, `NUPACK Design ?`, employs a similar approach to that of `RNA-SSD`, but, in this case, instead of finding sequences whose MFE structure is the given target structure, `NUPACK-DESIGN` attempts to find sequences having minimal *ensemble defect* **?**. Most remarkably, `NUPACK` solves inverse folding for the hybridization of two *or more* structures. IUPAC sequence constraints are supported, as well as control of GC-, AU- and GU-content, and the prohibition of occurrence of certain user-specified subsequences. It would appear that `NUPACK` supports treatment of dangles in a similar fashion to `-d` 0,1,2 option in the Vienna RNA Package. Moreover, inverse folding for both DNA and RNA is supported, where in the case of DNA, the user can stipulate Na^{+}, Mg^{++} concentrations. The `NUPACK` web server is available at `http://www.nupack.org/design/new`.

The program, `Inv ?`, uses a stochastic local search routine to determine a sequence whose minimum free energy *pseudoknotted* structure is a given target *3-noncrossing* RNA structure. `Inv` employs a dynamic programming method, that nevertheless requires time which is exponential in sequence length. There is no web server available.

The program, `MODENA ?`, uses a genetic algorithm to maximise proximity to the target structure and to minimize the free energy of a solution. Source code can be downloaded from `http://rna.eit.hirosaki-u.ac.jp/modena/`, but there is no web server available.

Finally, `Frnakenstein ?` is a recent Python program that calls Vienna RNA Package `RNAfold` and `RNAeval` within a genetic algorithm to evolve a collection of RNA sequences to have low energy structures

with respect to one *or more* target structures (since solution sequences are compatible with than one target structure, structural compatibility constraints are supported). Source code can be downloaded from `http://www.stats.ox.ac.uk/~anderson/Code/frnakenstein.html`; however, there is no web server. `Frnakenstein ?` allows the user to stipulate population size for its genetic algorithm, which thus determines the number of output sequences. A realistic upper bound on population size depends on run time, which is slow, since Python is an interpreted language.

Conclusion

`RNAiFold` is a web site that provides public access to the `CPdesign` and `LNSdesign` algorithms for solving the RNA inverse folding problem. `RNAiFold` supports a number of user-defined constraints concerning the nucleotide and base-pairing constitution of the sequences returned. The web site includes links to download source code as well as sequences, structures and benchmarking results from extensive testing of the software. Our group has confirmed the utility of `RNAiFold` in RNA molecular design, by experimentally validating a novel construct described in a forthcoming publication.

Acknowledgements

This work was supported by by National Science Foundation grant DMS-1016618. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

References

Xia, T., J. SantaLucia, J., Burkard, M., Kierzek, R., Schroeder, S., Jiao, X., Cox, C., and Turner, D. (1999) *Biochemistry* **37**, 14719–35.

Garcia-Martin, J., Clote, P., and Dotu, I. (2013) *J Bioinform Comput Biol* **11(2)**, 1350001 in press.

Hofacker, I., Fontana, W., Stadler, P., Bonhoeffer, L., Tacker, M., and Schuster, P. (1994) *Monatsch. Chem.* **125**, 167–188.

Andronescu, M., Fejes, A., Hutter, F., Hoos, H., and Condon, A. (2004) *J Mol Biol.* **336**, 607–624.

Busch, A. and Backofen, R. (2006) *Bioinformatics* **22(15)**, 1823–1831.

Taneda, A. (2011) *Advances and Applications in Bioinformatics and Chemistry* **4**, 1–12.

Zadeh, J. N., Steenberg, C. D., Bois, J. S., Wolfe, B. R., Pierce, M. B., Khan, A. R., Dirks, R. M., and Pierce, N. A. January 2011 *J. Comput. Chem.* **32(1)**, 170–173.

Gao, J., Li, L., and Reidys, C. (2010) *Algorithms for Molecular Biology* **5(27)**.

Zadeh, J. N., Wolfe, B. R., and Pierce, N. A. February 2011 *J. Comput. Chem.* **32(3)**, 439–452.

Hofacker, I. The rules of the evolutionary game for RNA: A a statistical characterization of the sequence to structure mapping in RNA Ph.D. disssertation from University of Vienna (1994).

Gruber, A., Lorenz, R., Bernhart, S., Neubock, R., and Hofacker, I. (2008) *Nucleic Acids Research* **36**, 70–74.

Dirks, R., Lin, M., Winfree, E., and Pierce, N. (2004) *Nucleic. Acids. Res.* **32(4)**, 1392–1403.

Lyngso, R. B., Anderson, J. W., Sizikova, E., Badugu, A., Hyland, T., and Hein, J. (2012) *BMC. Bioinformatics* **13**, 260.