

# RNAsoft: a suite of RNA secondary structure prediction and design software tools

Mirela Andronescu, Rosalía Aguirre-Hernández, Anne Condon\* and Holger H. Hoos

Department of Computer Science, University of British Columbia, Vancouver, BC V6T 1Z4, Canada

Received February 15, 2003; Revised and Accepted April 7, 2003

## ABSTRACT

**DNA and RNA strands are employed in novel ways in the construction of nanostructures, as molecular tags in libraries of polymers and in therapeutics. New software tools for prediction and design of molecular structure will be needed in these applications. The RNAsoft suite of programs provides tools for predicting the secondary structure of a pair of DNA or RNA molecules, testing that combinatorial tag sets of DNA and RNA molecules have no unwanted secondary structure and designing RNA strands that fold to a given input secondary structure. The tools are based on standard thermodynamic models of RNA secondary structure formation. RNAsoft can be found online at <http://www.RNAsoft.ca>.**

## INTRODUCTION

Software tools that predict the secondary structure of a DNA or RNA strand from the base sequence, such as mfold (1) and RNAfold from the Vienna RNA Package (2), are widely used to shed insight on nucleic acid structure and function. The secondary structure is the set of base pairs formed when the (single) strand folds on itself, with each base occurring in, at most, one pair. Increasingly, RNA molecules are designed for new purposes, as catalysts, probes on arrays, molecular bar codes or information storage media in biomolecular computations (3–5). Accordingly, new variants of the secondary structure prediction and design tasks for nucleic acids arise.

RNAsoft provides new tools for three such tasks:

- **PairFold** predicts the minimum free energy secondary structure formed by two input DNA or RNA molecules. PairFold can be used, for example, to predict interactions between a probe and target RNA molecule or between pairs of strands in biomolecular nanostructures.
- **CombFold** predicts which strand, out of a combinatorial set formed from DNA or RNA input strands, folds to a secondary structure with the lowest minimum free energy. CombFold can, for example, efficiently test that no strand in a large tag library [such as the Brenner's library of size  $8^8$  (3)] forms unwanted secondary structure.

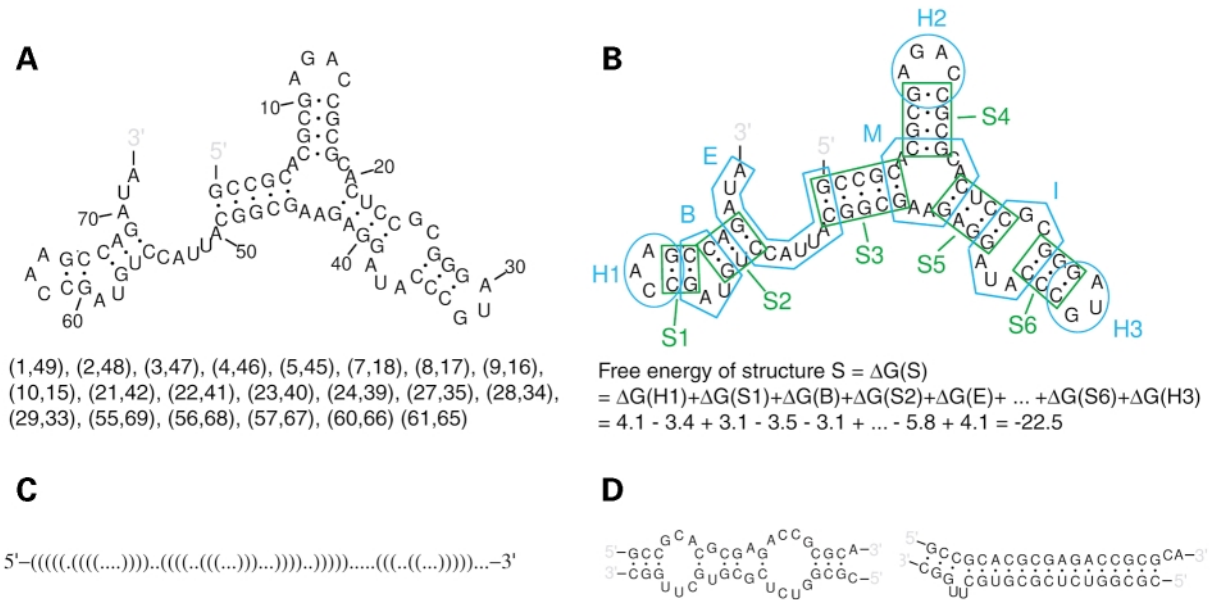
- **RNA Designer** designs an RNA sequence that folds to a given input secondary structure. The tool is intended for designers of RNA molecules with particular structural or functional properties.

The RNAsoft web site, at <http://www.RNAsoft.ca>, provides online access to all three tools. Following a brief overview of RNA secondary structure modeling and representation, we describe the function and output of each of the online services.

All tools are based on a standard free energy model (1), which provides a measure of thermodynamic stability for possible secondary structures that a molecule or molecules could form, as a function of the base sequence and temperature, in a 1 M NaCl solution. Here, we provide a brief overview of the model. If the bases of a strand  $R$  are indexed consecutively starting from the 5' end, then the secondary structure can be represented as a set of base pair indexes (Fig. 1A). If there are no base pairs  $(i, j)$  and  $(i', j')$  with  $i < i' < j < j'$ , then the structure is *pseudoknot free*. The standard free energy model pertains only to pseudoknot free structures, in which the base pairs define component loops and stacked pairs of the structure (Fig. 1A). The free energy of a structure is calculated as the sum of the energies of its component loops and stacked pairs, which in turn are calculated using experimentally obtained thermodynamic data (6,7) (Fig. 1B). The free energy model predicts that, under fixed conditions such as temperature, a DNA or RNA molecule will fold to the structure that minimizes the free energy. Throughout, when we refer to the minimum free energy structure for a molecule, we mean the minimum free energy pseudoknot free structure for the molecule. [Methods for prediction of pseudoknotted structures can be found in the work of Rivas and Eddy (8) and Gulyaev *et al.* (9).]

A pseudoknot free secondary structure for a molecule with  $n$  bases can be uniquely represented as a sequence of length  $n$  whose  $i$ th symbol is either '.' if the  $i$ th base is unpaired, '(' if the  $i$ th base is paired with a base of higher index and ')' if the  $i$ th base is paired with a base of lower index (Fig. 1C). We refer to this as dot-parenthesis notation. All three RNAsoft tools use this notation to represent DNA or RNA secondary structures in their input and output. Where possible, output structures are also provided in Connectivity Table (CT) and RNAML formats. An RNA sequence is represented as a string over the alphabet {A, C, G, U}, representing the adenine, cytosine, guanine and uracil bases, with the 5' end corresponding to the

\*To whom correspondence should be addressed: Tel: +1 6048228175; Fax: +1 604 822 5485; Email: [condon@cs.ubc.ca](mailto:condon@cs.ubc.ca)



**Figure 1.** RNA Secondary structure. **(A)** Graphical depiction of the predicted minimum free energy secondary structure  $S$  for sequence  $R = \text{GCCGCACGCG-AGACCGCGCACUCCGCGGGAUG CCCAUAGGAGAAGCGGCAUUACCUGUAGCCAAGCCAGUA}$ , plus the list of associated base pairs.  $S$  is pseudoknot free. **(B)** Free energy calculation for structure  $S$ . Contributions of the labeled stems and loops (hairpins, bulges, internal loops, external loop and multiloops) to the free energy are summed to obtain the total free energy, which for this strand is the minimum free energy. **(C)** Dot-parenthesis representation of secondary structure  $S$ . **(D)** Graphical depiction of two possible secondary structures for the pair of sequences  $(R_1, R_2) = (\text{GCCGCACGCGAGACCGCGCA}, \text{CGCGGUCUC-GCGUGCUUGGC})$ . The structure on the left is obtained by pairing bases in matching positions of the two strands and has free energy  $-19.14$  kcal/mol, as predicted by HYTHER<sup>TM</sup>, but this is not the minimum free energy structure. The structure on the right is predicted by PairFold to be the minimum free energy structure, having free energy  $-36.70$  kcal/mol.

left end of the sequence. A DNA sequence is represented in the same way, except that T, for thymine, replaces U.

**PAIRFOLD**

The PairFold program predicts the secondary structure of a pair of RNA or a pair of DNA molecules. Extending the notion of secondary structure for a single molecule, a secondary structure for a pair of molecules  $(R_1, R_2)$  is a set of base pairs, with each base of  $R_1$  and  $R_2$  occurring in, at most, one pair. The model for measuring the minimum free energy of a secondary structure for two RNA molecules at a given temperature is very similar to that for a single molecule, except that an intermolecular initiation penalty is added (Fig. 1D). The secondary structure can be represented as a pair of sequences  $(S_1, S_2)$  in dot-parenthesis notation. We define the structure  $(S_1, S_2)$  for the pair of molecules  $(R_1, R_2)$  to be pseudoknot free if, and only if, the structure  $S_1S_2$  for the single molecule  $R_1R_2$  (the concatenation of the sequences  $R_1$  and  $R_2$ ) is pseudoknot free.

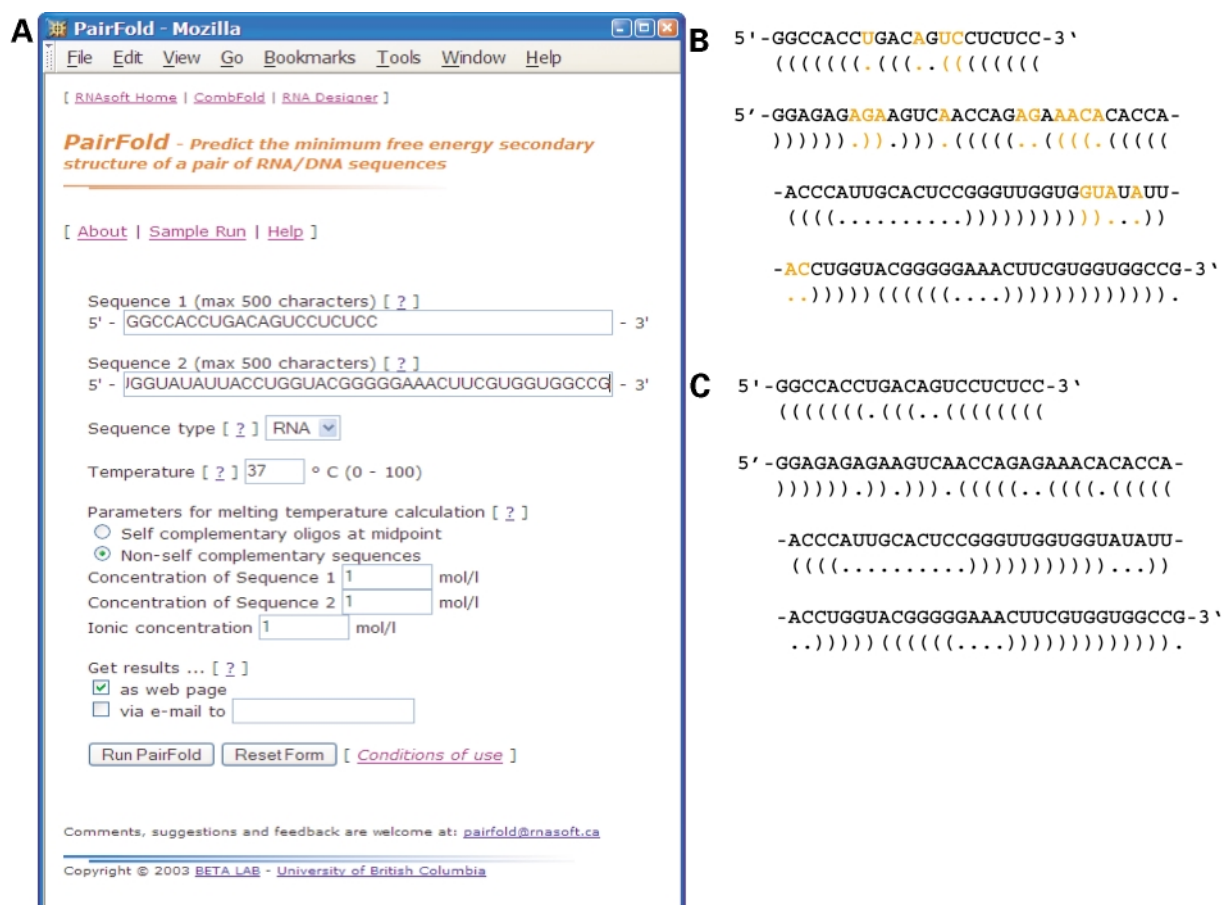
PairFold takes two RNA or two DNA sequences,  $R_1$  and  $R_2$ , plus a temperature value in the range  $0-100^\circ\text{C}$ . It calculates and outputs the minimum free energy (pseudoknot free) secondary structure of the pairs  $(R_1, R_2)$ ,  $(R_1, R_1)$  and  $(R_2, R_2)$  at the input temperature, along with their minimum free energy values. Predictions of the enthalpy, entropy and melting temperature of minimum free energy structure are also provided.

Figure 2 shows the user interface for PairFold, illustrating its use in predicting the secondary structure of a hairpin ribozyme

(10). As with all three online services, the user can choose whether to view the output on a dynamically generated web page or to receive the output via email or both. If a computation takes more than 1 min, the web interface notifies the user that the result will be sent out via email.

PairFold differs from mfold and RNAfold in that it takes as input two sequences, rather than one sequence. The HYTHER<sup>TM</sup> software tool (available at <http://ozone2.chem.wayne.edu/Hyther/hythermain.html>) does take two sequences as input, but is more limited than PairFold because it calculates the free energy of the single secondary structure obtained by pairing those thermodynamically favorable base pairs at the corresponding positions in the two input sequences, as in the first secondary structure of Figure 1D, rather than finding the minimum over the class of all pseudoknot free structures. Moreover, PairFold can accept sequences of different lengths, whereas the sequences input to HYTHER<sup>TM</sup> must have the same length. Other related programs include OligoWalk of Mathews *et al.* (11) and ProbeSelect of Li and Stormo (12), which design probe oligos for RNA targets or genes, respectively. An algorithm that is essentially the same as PairFold is incorporated into OligoWalk, but is not available on the web.

PairFold uses a dynamic programming algorithm that is a straightforward variant of the Zuker–Stiegler algorithm for single molecules (13). If the lengths of the sequences  $R_1$  and  $R_2$  are  $n_1$  and  $n_2$ , respectively, then the running time of PairFold is  $O((n_1 + n_2)^3)$ , meaning that the number of computer operations needed on input  $R_1, R_2$  is bounded by a constant time  $(n_1 + n_2)^3$ . In practice, the computation for sequences of  $<500$  bases each takes less than 500 CPU seconds on our



**Figure 2.** PairFold user interface. (A) The sequences  $R_1 = \text{GGCCACCUGACAGUCCUCUC}$  and  $R_2 = \text{GGAGAGAGAAGUCAACCAGAGAAACACACCAACCCAUUGCACUCCGGGUUGGUGGUAUAUUACCUGGUACGGGGGAAACUUCGUGGUGGCCG}$  are entered in simple text boxes. In this example, the sequences form a hairpin ribozyme (8). (B) The dot-parenthesis representation of the minimum free energy secondary structure as predicted by PairFold. (C) The dot-parenthesis representation of the secondary structure found in the crystallization construct, as reported by Rupert and Ferré-D'Amaré (10). The structure reported by PairFold for the pair ( $R_1, R_2$ ) contains the four stems found in the crystallization construct. The difference in the secondary structure for ( $R_1, R_2$ ) of (B) and the secondary structure of (C) (highlighted in orange in B) can be attributed to the fact that the non-standard base pairings A-A, G-A and C-A found in the true structure are not accounted for in the Turner thermodynamic parameters and thus cannot be predicted by software based on these parameters.

current server (for information on our server see section on Implementation). When allowing a maximum of 1 CPU hour, PairFold can handle sequences of length up to 1000. To keep the load on our server manageable and response times of the online services reasonably low, we limit the length of the input sequences for the online version of PairFold to 500 bases and the maximal processing time to 1 h of wall-clock time (using wall-clock time in this context means that the maximal CPU time available for processing the given input may be reduced when the server load is high).

Applications of PairFold include predicting the stability of a duplex formed by a probe oligo and target molecule, predicting how a ribozyme binds with its RNA target (10) or verifying that no unwanted secondary structure forms between two words (short single strands) in an RNA computation (5) or between two molecular tags in a polymer library (3). In the future, PairFold will be incorporated into a tool for the design of short DNA or RNA strands for use in biomolecular computations or DNA nanotechnology and will also be extended to take more than two sequences as input.

## COMBFOLD

CombFold predicts which strand from a combinatorial set of strands has the lowest minimum free energy secondary structure. Combinatorial sets of DNA strands are used as molecular bar codes in applications such as massively parallel signature sequencing (3), as well as for information storage in DNA computations (5). For example, Brenner *et al.* (3) use  $8^8$  DNA tags, each of length 32 (Fig. 3). In these applications, it is important that none of the tags or strands fold on themselves in the temperature range at which they are used. CombFold provides an efficient way to test whether this is indeed the case, avoiding the need to run a program like mfold on each tag individually.

CombFold takes as input a description of a combinatorial set of DNA or RNA strands. We define a combinatorial set of strands as follows. Let  $S_1, S_2, \dots, S_k$  be sets of strands for some  $k$  (see Fig. 3 for examples). Within each  $S_i$ , all strands should have the same length. We denote the length of the strands in set  $S_i$  by  $n_i$  and the number of strands in  $S_i$  by  $N_i$ . Then the combinatorial

A								B					
1	2	3	4	5	6	7	8	1	2	3	4	5	6
TTAC	TTAC	TTAC	TTAC	TTAC	TTAC	TTAC	TTAC	AU	G	CAAUGC	G	A	AUGCAC
AATC	AATC	AATC	AATC	AATC	AATC	AATC	AATC	A			C	G	
TACT	TACT	TACT	TACT	TACT	TACT	TACT	TACT					C	
ATCA	ATCA	ATCA	ATCA	ATCA	ATCA	ATCA	ATCA					U	
ACAT	ACAT	ACAT	ACAT	ACAT	ACAT	ACAT	ACAT						
TCTA	TCTA	TCTA	TCTA	TCTA	TCTA	TCTA	TCTA						
CTTT	CTTT	CTTT	CTTT	CTTT	CTTT	CTTT	CTTT						
CAAA	CAAA	CAAA	CAAA	CAAA	CAAA	CAAA	CAAA						

**Figure 3.** Combinatorial sets of strands. (A) The DNA tag set of Brenner *et al.* (3) Each tag is obtained by concatenating eight 4mers, one chosen from each of the eight columns. Thus, each tag has length 32 and there are  $8^8 = 16\,777\,216$  tags in total. One such tag is AATCTTACCAAAACATACATTCTACTTTAATC. Column  $i$  lists the strands in the set  $S_i$ , for  $i$  between 1 and 8. Also in this example,  $n_i = 4$  and  $N_i = 8$  for each  $i$ . (B) A simple combinatorial set used in the screen shots of Figure 4. This set can also be represented using IUPAC code, as 5'-AURCAAUGC SNAUGCAC-3'.

set formed from  $S_1, S_2, \dots, S_k$  is the set of all strands of the form  $S_1 S_2 \dots S_k$  where  $S_i$  is a strand in  $S_i$ , for each  $i$  between 1 and  $k$ . Note that the length of a strand in the combinatorial set is  $n = n_1 + n_2 + \dots + n_k$  and the number of strands in the combinatorial set is  $N_1 \times N_2 \times \dots \times N_k$ .

A combinatorial set may be specified in one of two ways in the CombFold user interface. The first way is to use IUPAC code. For example, 5'-ARNG-3' represents the combinatorial set of 4mers in which the first and last bases are A and G, respectively, the second is a purine (A or G) and the third may be any one of A, C, G, U or T. In the second way to specify a combinatorial set (Fig. 4) the strands in set  $S_1$  are entered, one per line, in a simple text area, followed by a line containing a \*, then the strands in  $S_2$  are entered, one per line, followed by a line containing a \*, and so on until the strands in  $S_k$  are entered. The set of strands can easily be pasted into the box from a file on the user's computer. The folding temperature may be specified by the user. In addition, the user may specify the desired number,  $N$ , of output sequences. CombFold will return the  $N$  strands with the lowest minimum free energy secondary structures at the given temperature, along with the corresponding structures and free energy values.

CombFold uses a dynamic programming algorithm to identify its output, thereby avoiding the need to calculate the free energy of each individual strand in the combinatorial set (14,15). The running time of CombFold is  $O(N^2 \times n^3)$ , where  $N$  is the maximum of the  $N_i$  and  $n$  is the length of the strands in the combinatorial set. In contrast, an algorithm that folds each possible strand in the combinatorial set requires time proportional to  $N_1 \times N_2 \times \dots \times N_k \times n^3$ . In applications of CombFold,  $N$  is often small compared to  $k$ —for example, in the combinatorial set of Braich *et al.* (5) all of the  $N_i = 2$  (and thus  $N = 2$ ) and  $k = 20$ —in which case CombFold is significantly faster than the naive approach. Still, in practice the running time of CombFold increases quickly with  $N$ ,  $k$  and  $n$ . On an input with 10 sets, two strands per set and all strands having length 16, the computation time is <400 CPU seconds. On an input with eight sets, eight strands in each set and all strands having length 4 [as in Brenner's combinatorial set (3)], the computation time is <700 CPU seconds. To keep the load on our server manageable and response times reasonably low, we limit the size of CombFold's inputs to 10 sets, four strands per set and strand length 20.

CombFold can be used to test whether all strands in a combinatorial set for use in a DNA or RNA computation (5,16) or as molecular tags (3) have no unwanted secondary structure. CombFold can also be useful to gain insight on the range of

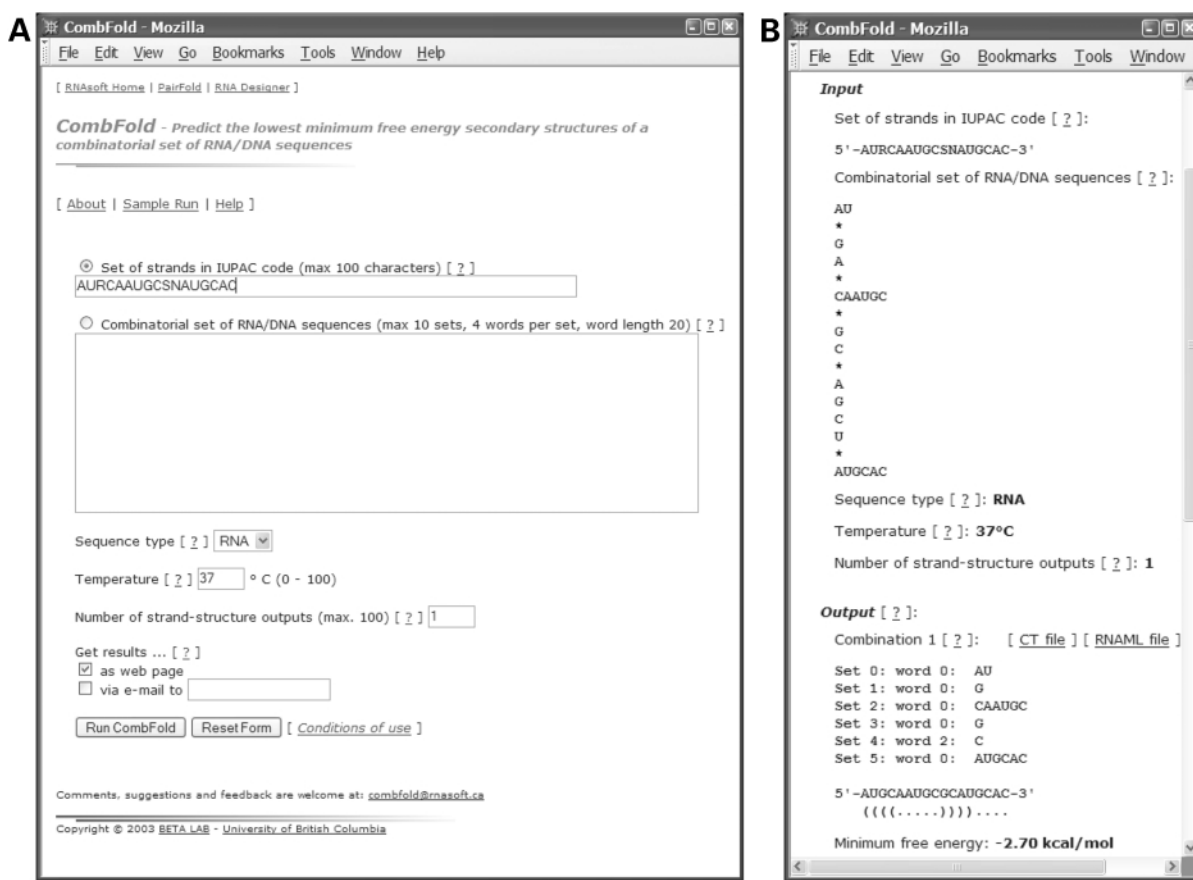
secondary structures that can be formed from a consensus RNA sequence or in identification of good starting points for mutagenesis or SELEX experiments. Another application is to determine which of the many RNA strands that code for a particular protein has the most stable secondary structure (14). In the future, CombFold and an extended version of PairFold will be combined to determine the minimum free energy structure that can be formed from multiple strands taken over all of the possible orderings of the individual strands.

## RNA DESIGNER

RNA Designer takes as input a secondary structure description and outputs an RNA strand that is predicted to fold to that secondary structure. RNA Designer can be used to design RNA molecules with certain structural properties, as part of the development of molecules with novel functional properties, or more fundamentally in order to understand which secondary structure elements are critical to specific functions of cellular RNAs.

RNA Designer uses a stochastic local search algorithm, which decomposes the input structure in a hierarchical fashion, finds strands that fold to the resulting substructures and then attempts to combine the strands for substructures into a strand for the overall structure. A pseudo-random number generator is used to implement the stochastic steps of the algorithm and by fixing the seed for this generator, the behavior of RNA Designer becomes completely deterministic. If desired, any run of the algorithm may be reproduced by running the algorithm again with the respective seed value. RNA Designer uses the *fold* routine from the Vienna RNA Package as part of its implementation (2). Due to the stochastic nature of the algorithm, RNA Designer can be used to probabilistically sample the solution space by performing multiple runs of the algorithm (with different seed values). This allows the user to easily obtain sets of sequences that all fold to the desired structure, from which further selections according to specific criteria can be made manually.

As shown in Figure 5, the input secondary structure is specified using dot-parenthesis representation and is entered in a simple text box. The temperature, number of output strands and optionally a seed for the pseudo-random number generator may all be specified. Furthermore, the user may additionally impose primary sequence constraints by providing a sequence specification using a subset of IUPAC code. Two additional parameters can be used to control the target GC content of



**Figure 4.** CombFold user interface. (A) Web interface for CombFold input. The input is the combinatorial set of Figure 3B specified using IUPAC code. (B) Partial view of web interface for CombFold output, reporting that for the given input, the sequence AUGCAAUGCGCAUGCAC has the minimum free energy structure, namely (((.....)))...., with free energy  $-2.70$  kcal/mol.

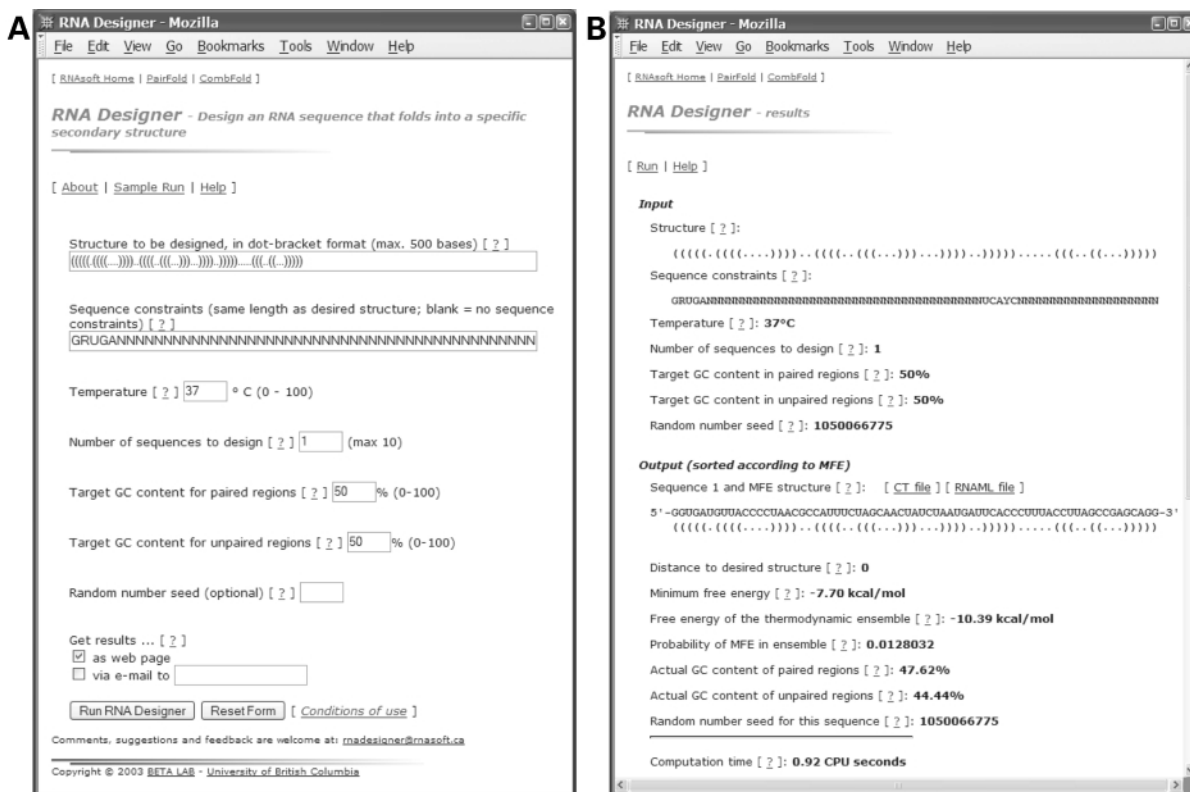
paired and unpaired bases in the given structure. In our current implementation, these parameters allow the user to specify a probabilistic bias towards the desired GC content in the search process. In cases where many sequences exist for a given structure (or substructure), RNA Designer can be expected to produce results that closely match the target GC content. However, in cases where finding a correctly folding sequence with the target GC content is hard or impossible, the search is allowed to drift from the GC target.

RNA Designer outputs the specified number of RNA strands and, for each, the structure to which the strand is predicted to fold at the user specified temperature, its free energy and the random seed used to produce the strand. If the user does not provide a random seed, a seed is chosen based on the time at which the program starts to run. In most cases, the output strands fold to the input structure. However, if the program fails to find a strand which folds to the input structure (e.g. because no strand exists that would fold into the given structure), it outputs the strand found whose MFE structure is of minimum distance from the input structure, where the distance is the number of bases in the output strand that are not paired (or unpaired) in the output secondary structure as specified in the input secondary structure.

The Vienna Package also contains a tool for design of RNA strands that fold to a given input secondary structure. However,

RNA Designer is consistently able to solve structures for which the Vienna Package program, RNAinverse, is unable to find solutions and its running time is significantly lower than RNAinverse. Furthermore, RNAinverse does not support primary sequence constraints. We empirically evaluated RNA Designer on 24 computationally predicted structures of biological sequences from the Ribosomal Database Project. RNA Designer found solutions to all 24 instances (length: 260–1475 bases), while RNAinverse solved only seven of the instances. Examples of sequences whose predicted structures were not solved by RNAinverse include *Bradyrhizobium* sp. str. 283A, *Spirochaeta* sp. and *Methanocaldococcus fervens* str. AG86 DSM 4213 (T). RNA Designer also performed better than the RNAinverse on 420 artificially generated structures. Further details can be found in a report that is linked from the RNA Designer website.

RNA Designer, when running on our current server, typically finds RNA sequences of length  $\sim 300$  bases for biologically plausible secondary structures in  $<60$  CPU seconds. When allowing a maximum runtime of 1 CPU hour, RNA Designer can handle structures with up to 1500 bases. To keep the load on our server manageable and to ensure reasonably low response times, we limit the length of the input sequences for the online version of RNA Designer to 500 and the maximal processing time to 1 h of wall-clock time.



**Figure 5.** RNA Designer user interface. (A) Web interface for RNA Designer input, with input structure as in Figure 1C. (B) Partial view of web interface for RNA Designer output.

Possible applications for RNA Designer include the design of RNAs with specific secondary structures which may be used in the study and characterization of biological RNAs and their function as well as in building artificial RNA structures that can be used for modulation of cellular RNA functions, nanostructure design or biomolecular computations.

Future plans for RNA Designer include: (i) to support better energy models of RNA structure (in particular, models including pseudoknots which provide the basis for the algorithmic design of strands with pseudoknotted secondary structures); (ii) to support the design of RNAs with bi- or multistable secondary structures; and (iii) to support the design of complexes consisting of two or more RNA strands.

## IMPLEMENTATION

At the basis of the RNAsoft web services are the three standalone programs PairFold, CombFold and RNA Designer. These have been developed and implemented in C++ under RedHat Linux, version 2.4.18 SMP. RNA Designer uses routines from the Vienna RNA Package (see <http://www.tbi.univie.ac.at/~ivo/RNA/RNALib.html>). These programs are accessed from the RNAsoft web site via HTML pages and CGI scripts implemented in Perl. Our current server machine is a PC with two Intel Xeon 2 GHz CPUs with 512 KB CPU cache each and 4 Gb of RAM, running Redhat Linux, version 2.4.18 SMP; this machine runs an Apache web server.

Thermodynamic parameters for RNA were provided by Doug Turner (6) and Michael Zuker (<http://rna.chem.rochester.edu/>, <http://www.bioinfo.rpi.edu/~zukerm/>) and parameters for DNA were provided by John SantaLucia Jr (7) (<http://ozone.chem.wayne.edu>).

## ACKNOWLEDGEMENTS

We gratefully acknowledge the permission of John SantaLucia Jr, Doug Turner and Michael Zuker for use of their DNA and RNA thermodynamic parameters, and Ivo Hofacker *et al.* (2) for making the Vienna RNA Package publicly available, parts of which are used in RNA Designer. In addition, we thank our colleagues Barry Cohen, Danielle Dees, Anthony Fejes, Firas Hamze, Frank Hutter, Steven Skiena, Laura Slaybaugh and Shelly Zhao, who collaborated with us on development of algorithms and on earlier versions of this software. We thank our collaborators Dan Tulpan, Rob Corn and Lloyd Smith for their support in the development of PairFold, and Dave Brent for his technical assistance. We thank the anonymous reviewers for their valuable suggestions for improving the functionality of the tools and for pointing out additional useful applications. This work was funded by the Natural Sciences and Engineering Research Council (NSERC) of Canada, the US National Science Foundation under grant number 144-KN87, the Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory, Air Force Materiel Command, USAF, under agreement number

F30602-01-2-0555. The US Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Defense Advanced Research Projects Agency (DARPA), the Air Force Research Laboratory or the US Government.

## REFERENCES

- Zuker,M., Mathews,D.H. and Turner,D.H. (1999). Algorithms and thermodynamics for RNA secondary structure prediction: a practical guide. Barciszewski,J. and Clark,B.F.C. (eds), In *RNA Biochemistry and Biotechnology*, NATO ASI Series, Kluwer Academic Publishers.
- Hofacker,I.L., Fontana,W., Stadler,P.F., Bonhoeffer,S., Tacker,M. and Schuster,P. (1994) Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, **125**, 167–188.
- Brenner,S., Williams R.S., Vermaas,E.H., Storck,T., Moon,K., McCollum,C., Mao,J-I., Luo,S., Kirchner,J.J., Eletr,S. *et al.* (2000) *In vitro* cloning of complex mixtures of DNA on microbeads: physical separation of differentially expressed cDNAs. *Proc. Natl Acad. Sci. USA*, **97**, 1665–1670.
- Shoemaker,D.D., Lashkari,D.A., Morris,O., Mittman,M. and Davis,R.W. (1996) Quantitative phenotypic analysis of yeast deletion mutants using a highly parallel molecular bar-coding strategy. *Nature Genet.*, **16**, 450–456.
- Braich,R.S., Chelyapov,N., Johnson,C., Rothmund,P.W.K. and Adleman,L. (2002) Solution of a 20-variable 3-SAT problem on a DNA computer. *Science*, **296**, 499–502.
- Serra,M.J., Turner,D.H. and Freier,S.M. (1995) Predicting thermodynamic properties of RNA. *Methods Enzymol.*, **259**, 243–261.
- SantaLucia,J.,Jr (1998) A unified view of polymer, dumbbell, and oligonucleotide DNA nearest-neighbor thermodynamics. *Proc. Natl Acad. Sci. USA*, **95**, 1460–1465.
- Rivas,E. and Eddy,S.R. (1999) A dynamic programming algorithm for RNA structure prediction including pseudoknots. *J. Mol. Biol.*, **285**, 2053–2068.
- Gulyaev,A.P., van Batenburg,F.H.D. and Pleij,C.W.A. (1995) The computer simulation of RNA folding pathways using a genetic algorithm. *J. Mol. Biol.*, **250**, 37–51.
- Rupert,P.B. and Ferré-D’Amaré,A.R. (2001) Crystal structure of a hairpin ribozyme-inhibitor complex with implications for catalysis. *Nature*, **410**, 780–786.
- Mathews,D.H., Burkard,M.E., Freier,S.M., Wyatt,J.R. and Turner,D.H. (1999) Predicting oligonucleotide affinity to nucleic acid targets. *RNA*, **5**, 1458–1469.
- Li,F. and Stormo,G.D. (2001) Selection of optimal DNA oligos for gene expression arrays. *Bioinformatics*, **17**, 1067–1076.
- Zuker,M. and Stiegler,P. (1981) Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Res.*, **9**, 133–148.
- Cohen,B. and Skiena,S. (2002) Designing RNA structures: natural and artificial selection. *Proc. Sixth Annual Int. Conf. Comp. Biol.*, ACM Press, New York, pp. 109–116.
- Andronescu,M., Dees,D., Slaybaugh,L., Zhao,Y., Cohen,B., Condon,A. and Skiena,S. (2003) Algorithms for testing that sets of DNA words concatenate without secondary structure. *Proc. Eighth International Workshop on DNA Based Computers. Lecture Notes in Computer Science*, Springer-Verlag, **2568**, 182–195.
- Faulhammer,D., Cukras,A.R., Lipton,R.J. and Landweber,L.F. (2000) Molecular computation: RNA solutions to chess problems. *Proc. Natl Acad. Sci. USA*, **97**, 1385–1389.