BIOINFORMATICS APPLICATIONS NOTE Vol. 29 no. 22 2013, pages 2941–2942
doi:10.1093/bioinformatics/btt496

Structural bioinformatics

Advance Access publication August 23, 2013

RNA secondary structure diagrams for very large molecules: RNAfdl

Nikolai Hecker[1,2,*], Tim Wiegels[3] and Andrew E. Torda[4]

[1]Centre for non-coding RNA in Technology and Health and [2]Department of Veterinary Clinical and Animal Sciences, University of Copenhagen, 1870 Frederiksberg, Denmark, [3]European Molecular Biology Laboratory—Hamburg Outstation, c/o DESY, 22603 Hamburg, Germany and [4]Centre for Bioinformatics, Institute for Biochemistry and Molecular Biology, University of Hamburg, 20146 Hamburg, Germany

Associate Editor: Ivo Hofacker

ABSTRACT

Summary: There are many programs that can read the secondary structure of an RNA molecule and draw a diagram, but hardly any that can cope with 10^3 bases. RNAfdl is slow but capable of producing intersection-free diagrams for ribosome-sized structures, has a graphical user interface for adjustments and produces output in common formats.

Availability and implementation: Source code is available under the GNU General Public License v3.0 at http://sourceforge.net/projects/rnafdl for Linux and similar systems or Windows using MinGW. RNAfdl is implemented in C, uses the Cairo 2D graphics library and offers both command line and graphical user interfaces.

Contact: hecker@rth.dk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on May 9, 2013; revised on July 31, 2013; accepted on August 19, 2013

1 INTRODUCTION

It might be a great simplification, but the most popular method to show RNA structure remains the classic radial base pair graph. We describe RNA force-directed layout (RNAfdl), whose strength is the ability to handle large molecules and to produce diagrams without intersections, without user intervention. This type of diagram is extremely regular with fixed distances between consecutive bases and between members of base pairs. Unpaired bases are placed on circle segments. The rules mean that diagrams are easy to interpret, but the rules cannot be enforced on larger molecules. They invariably lead to intersecting lines.

This means that programs either produce bad diagrams or bend the rules. NAView (Bruccoleri and Heinrich, 1988) relaxes the rules for loop regions, with the consequence that loops may be distorted in size and shape, and larger structures are still not free of line intersections (data not shown). Muller *et al.* (1993) treated the intersection problem with a backtracking method, but the code still cannot draw a large ribosomal subunit (Gaspin, 2001). A more recent program can draw large structures without intersections, but varies the distances between bases in loops (Byun and Han, 2009). There are no intersections, but the size of the loops may not reflect the number of bases.

If automatic approaches fail, one may allow or even encourage manual intervention. VARNA and RNAViz both use sophisticated methods for an initial layout, but rely on manual editing to get intersection-free plots (Darty *et al.*, 2009; De Rijk *et al.*, 2003).

2 METHODS

RNAfdl is able to generate intersection-free diagrams for structures as large as ribosomes. Like jVis.Rna (Wiese *et al.*, 2005), RNAfdl begins with a circle plot (Fig. 1A). When there are no pseudoknots, this is always intersection free. A penalty function is defined that quantifies how the current layout deviates from an ideal radial layout, and this is optimized by conjugate gradients minimization (Press *et al.*, 2007). The penalty function uses quadratic terms $\left(r_{ij} - r'_{ij}\right)^2$ to enforce distances between consecutive residues, paired bases and non-consecutive bases within unpaired loops. In each case, r_{ij} is the distance between bases i and j, and r'_{ij} is an ideal distance. In the case of unpaired regions, this is calculated from ideal circular geometry. A quartic term is used for repulsion between bases. Finally, and more unusually, a quartic term is used to enforce repulsion between bases and any line connecting consecutive or paired bases. This was a key ingredient to avoiding the need for manual intervention. A slight *ad hoc* change was made to the classic conjugate gradients method. First, the largest possible step size that does not lead to an intersection is calculated. A sweep line algorithm is used to test for intersections, and a ray-tracing method is used to detect loops that would be pushed into helices (Cormen *et al.*, 2009; de Berg *et al.*, 2008; Press *et al.*, 2007). Next, the textbook-style locally optimal step size is determined within this range. The implementation uses range and segment trees (de Berg *et al.*, 2008).

3 RESULTS AND DISCUSSION

Our test suite has >500 test structures from 100 to 700 nucleotides, as well as small and large ribosomal subunits downloaded from RNA STRAND (Andronescu *et al.*, 2008).All are drawn without intersections and without user intervention (Fig. 1B). Admittedly, the running time is long, ranging from seconds to half an hour for larger structures (~700 bases), to more than half a day for the large ribosomal subunit (2904 bases). The complexity per step is $O(n^2)$ (Supplementary Material), but the number of steps can be large for complicated structures. This pain is somewhat alleviated by two properties. First, the code can use a quick

*To whom correspondence should be addressed.

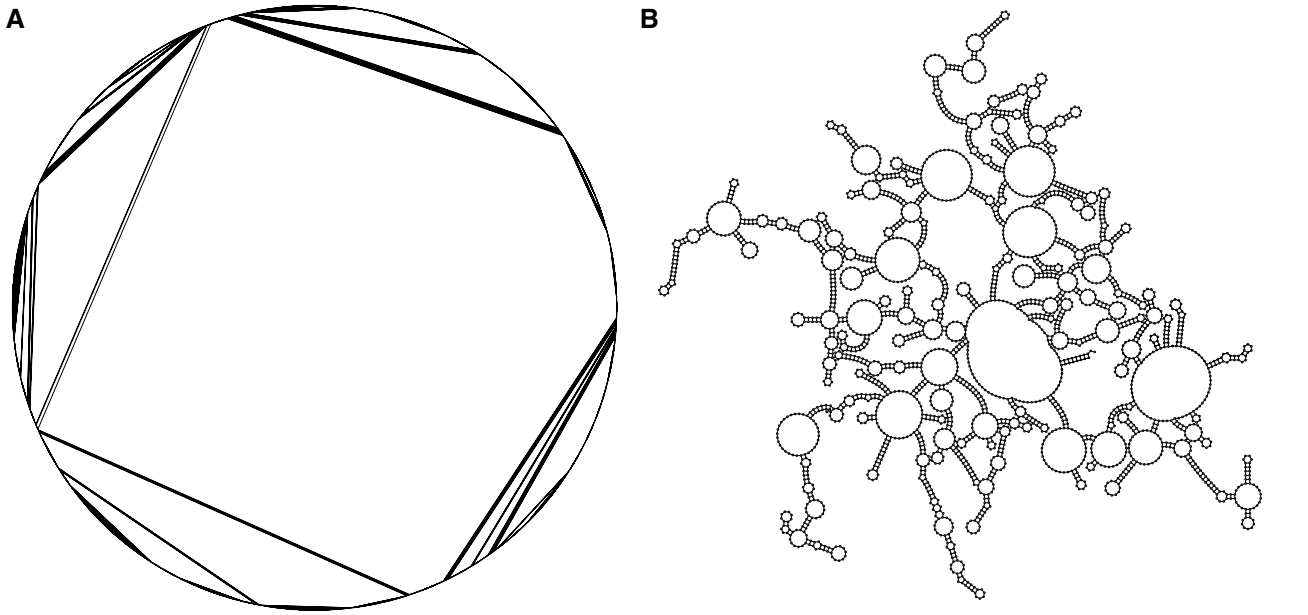

Fig. 1. Secondary structure of the 23S rRNA (*Escherichia coli*) with 2904 bases. (**A**) Circle plot, (**B**) radial layout using the gradient descent approach

radial layout method for smaller structures. Second, the graphical user interface allows one to stop a calculation, save the coordinates and resume later.

The graphical user interface offers manual editing, which can be useful even if there are no intersections to be repaired. RNAfdl is flexible here. Single bases or a set of selected bases can be moved without restrictions. Like other tools, RNAfdl allows bases to be colored; non-canonical base pairs can be incorporated; and many drawing parameters ranging from line width, the size of bases, to the interval of tick marks can be adjusted.

The program can, of course, be invoked from the command line, making it suitable for automation in pipelines or batch jobs. One can also generate circle plots, linear plots and mountain plots. The input follows the dot/bracket format used in the ViennaRNA Package (Lorenz *et al*., 2011). Output can be saved in several different formats defined by the underlying graphics library. The choices include PDF, SVG and popular bitmap formats such as PNG, so it is easy to incorporate diagrams in other files.

RNAfdl is a tool for people with patience, but there may be no other program that gives similar approximations to radial layouts for large structures. It is hoped that the program is already useful for publications and presentations. The performance will be improved. Parts of the cost function are independent and suitable for parallelizing. There is room for heuristics for better starting configurations. There is also room for more sophisticated treatment of pseudoknots and tertiary interactions.

ACKNOWLEDGEMENTS

We thank Marco Matthies, Tobias Schwabe, Jens Kleesiek and Peter Kerpedjiev for testing and suggestions; Stefan Bienert for test data and advice; Corinna Theis for testing and suggesting the integration of non-canonical base pairs.

Conflict of Interest: none declared.

REFERENCES

Andronescu,M. *et al*. (2008) RNA STRAND: the RNA secondary structure and statistical analysis database. *BMC Bioinformatics*, **9**, 340.

Bruccoleri,R.E. and Heinrich,G. (1988) An improved algorithm for nucleic acid secondary structure display. *Comput. Appl. Biosci*., **4**, 167–173.

Byun,Y. and Han,K. (2009) PseudoViewer3: generating planar drawings of large-scale RNA structures with pseudoknots. *Bioinformatics*, **25**, 1435–1437.

Cormen,T.H. *et al*. (2009) *Introduction to Algorithms*. 3rd edn. The MIT Press, Cambridge, Massachusetts, USA; London, UK.

Darty,K. *et al*. (2009) VARNA: interactive drawing and editing of the RNA secondary structure. *Bioinformatics*, **25**, 1974–1975.

de Berg,M. *et al*. (2008) *Computational Geometry: Algorithms and Applications*. Springer-Verlag Berlin Heidelberg, Germany.

De Rijk,P. *et al*. (2003) RnaViz 2: an improved representation of RNA secondary structure. *Bioinformatics*, **19**, 299–300.

Gaspin,C. (2001) RNA secondary structure determination and representation based on constraints satisfaction. *Constraints*, **6**, 201–221.

Lorenz,R. *et al*. (2011) ViennaRNA Package 2.0. *Algorithms Mol. Biol*., **6**, 26.

Muller,G. *et al*. (1993) Automatic display of RNA secondary structures. *Comput. Appl. Biosci*., **9**, 551–561.

Press,W.H. *et al*. (2007) *Numerical Recipes: The Art of Scientific Computing*. 3rd edn. Cambridge University Press, New York, USA.

Wiese,K.C. *et al*. (2005) JViz.Rna—a Java tool for RNA secondary structure visualization. *IEEE Trans. Nanobioscience*, **4**, 212–218.