Flexible and universal multiple target sequence design

Stefan Hammer[1,2], Birgit Tschiatschek[2], Christoph Flamm[1], Ivo L. Hofacker[1,2,3] and Sven Findeiß[1,2]

1 University of Vienna, Department of Theoretical Chemistry, Vienna, A-1090, Austria
2 University of Vienna, Bioinformatics and Computational Biology Research Group, Vienna, A-1090, Austria
3 University of Copenhagen, Center for Non-coding RNA in Technology and Health, Copenhagen, DK-1870, Denmark

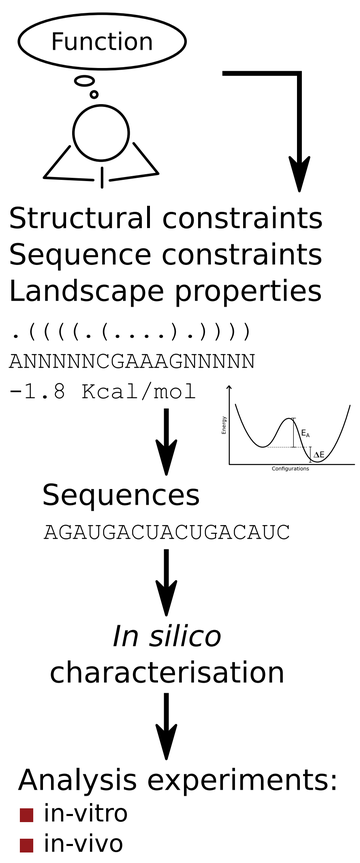
Abstract

Motivation: Due to its close structure to function relationship, the availability of good structure prediction methods and energy models, RNA is perfectly suited to design molecules with predefined properties. Currently available RNA design tools implement very specialized use-cases which cannot be easily adapted to new applications. Often, complicated sampling and optimization methods were developed to suit a specific RNA design goal.

Results: We therefore developed a C++ Library implementing a graph coloring approach to uniformly sample sequences compatible to structural and sequence constraints from the typically huge solution space. Uniform sampling from the solution space not only makes optimization runs much more performant, but for long optimization runs raises the probability to find better solutions. Scripting interfaces allow to easily adapt existing code to new scenarios which makes the whole design process very universal and flexible. We implemented novel design approaches written in Python to show the advantages of a scripting language in conjunction with the RNAdesign library for sequence sampling. We show that our software can be combined with any other software package to allow diverse RNA design applications.

Introduction To RNA design

To be able to *de novo* design RNA molecules it was necessary to develop methods to solve the inverse folding problem and approaches for an *in silico* characterization. This task can be generally split into three components:
(1) Generate valid sequences compatible to sequence and structural constraints, (2) formulate an optimization problem where the sampled sequences are optimized towards a minimal score calculated by the objective function and (3) use in silico characterization methods such as filtering and clustering with respect to additional features not included in the objective.



Fair sampling of RNA sequences

The developed software implements a graph theoretical approach to uniformly sample any sequence from the solution space [2].
A so called Dependency Graph is constructed from the given structural constraints. It is then decomposed into smaller components such that only paths remain (Figure 1). Therefore, the complex problem is split into two parts: (1) assigning bases with the right probabilities to paths [1] and (2) assigning bases to the connection points beween the paths. The algorithm is implemented as a dynamic programming approach, where probabilities for each position are calculated and stochastic backtracing is used to sample the nucleotide assignments.
The software handles multiple structural constraints and any sequence constraint in IUPAC annotation as input and generates sequences compatible to all constraints.

This work has been supported by the European Commission under the Environment Theme of the 7th Framework Program for Research and Technological Development (Grant agreement number 323987).

Furthermore, it is possible to get properties of the solution space and current search spaces, such as the number of solutions or characteristics of the underlying dependency graph. This feature can, for example, be used for mutational analyses. For efficient optimizations we guarantee to sample every solution with the same probability from the whole solution space as shown in Figure 2.

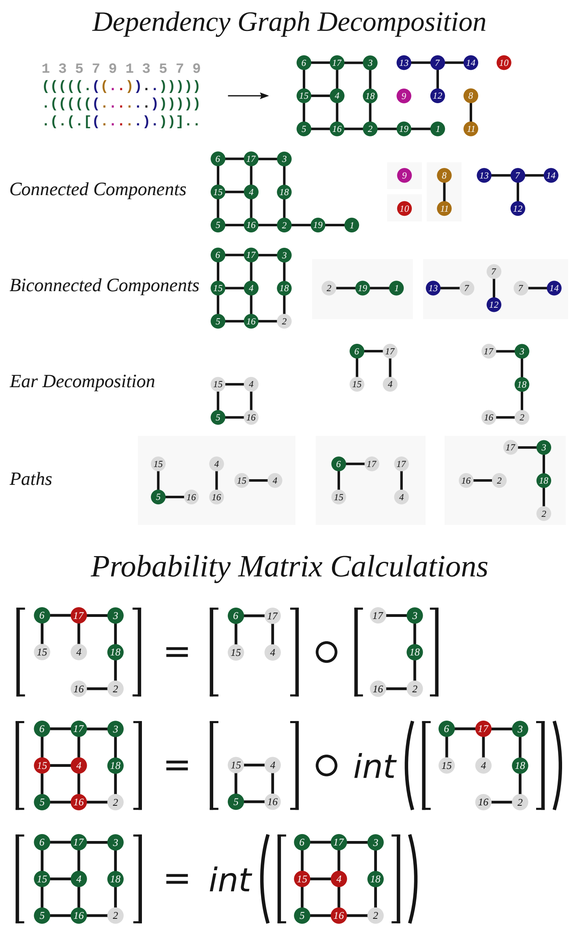
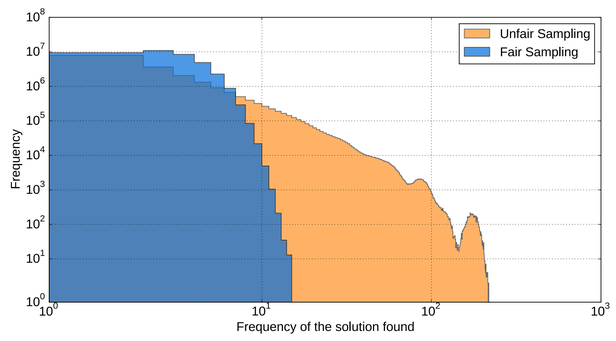


Figure 1: Graphical representation of the algorithmic implementation details. Left: Decomposition of the dependency graph into its subgraphs. Coloring is consistent between the input structural constraints and the connected components of the dependency graph. Gray boxed subgraphs are not decomposed further as we can get their probabilities with the path coloring approach. Gray notes represent special vertices. Right: Probabilities for all other graphs can be calculated as depicted. Gray nodes are special vertices, red nodes internal special vertices which can be removed from the matrix with the int() function and **o** is a matrix multiplication operator.



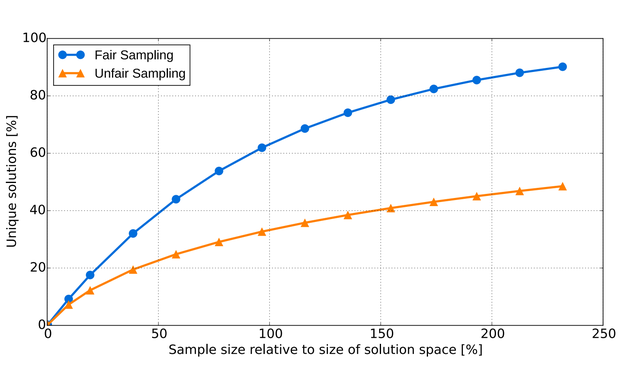


Figure 2: Differences in fair and unfair stochastic sampling shown on a small example with a rather complex dependency graph including biconnected components and blocks. Top: The histogram shows how frequent unique solutions were found when sampling completely new sequences using fair and unfair sampling. 9.6E+9 solutions were sampled from 4.1E+7 possible unique sequences (size of solution space). While fair sampling leads to a normal distribution with the mean (2.57) count being slightly above the relative sample size and the maximum number of times a solution is discovered being 15, unfair sampling leads to a warped distribution where a solution is found 4.78 times on average and 227 times maximal. Bottom: Even if we choose the sample size to be much bigger than the solution space (~230%), we still only get about 50% of all possible solutions with unfair sampling for this example, while we are able to sample about 90% with the fair method. The performance of the fair sampling is indipendent of the underlying problem whereas the curve of the unfair approach heavily depends on the properties of the dependency graph.

sRNA regulated gene expression

To show the advantages of our software, namely the flexibility, universality and efficiency, we implemented a regulatory 5'UTR that can control the expression of its downstream gene by responding to the presence of a small RNA. We use the RNAdesign library to sample sequences compatible to our structural and sequence constraints (Figure 3), formulated an adaptive walk optimization approach and came up with a novel objective function (Equation 1) which not only contains the accessibility of the RBS, but also concentrations of the formed sRNA/5'UTR complex for efficient binding.

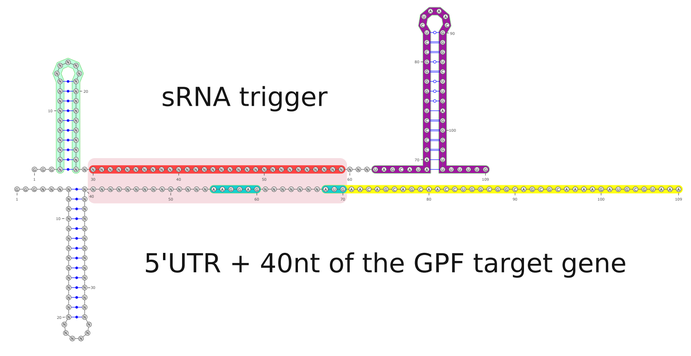


Figure 3: Designed structural and sequence constraints. The sRNA trigger containes a 5' stem for more stability (green) and a standard T7 terminator (lavender). The only sequence constraints used are a triple 5' G after the promoter and the terminator sequence. The mRNA consists of the retulatory 5'UTR and the first 40nt of the GFP reporter gene as context (yellow). We designed a 5' stem to enhance the accessibility of the ribosome binding site RBS (cyan) and the start codon (cyan). The sRNA/mRNA binding site (red) is included in the objective as the term S_{AB}.

$$
\begin{aligned}
f(x) = & \underbrace{1 - [S_{AB}]/[A_0]}_{\text{Efficient duplex formation}} + \beta \times \underbrace{(1 - P(RBS_{unpaired}))}_{\text{RBS accessibility}} \to 0 \\
[S_{AB}] = & [AB] \times P(S_{AB}|Z^{AB\prime}) \\
Z^{AB\prime} = & Z^{AB} - Z^{A} Z^{B} \\
P(S_{AB}|Z^{AB\prime}) = & \frac{Z_{S_{AB}}}{Z_{AB\prime}} = e^{-\frac{E_{S_{AB}} - E_{AB\prime}}{kT}} \\
P(RBS_{unpaired}) = & \frac{Z_{RBS_{unpaired}}}{Z_A} = e^{-\frac{E_{RBS_{unpaired}} - E_A}{kT}} \\
[S_{AB}] \leq & [A_0]
\end{aligned}
$$

Equation 1: Objective function characterizing a sRNA induced gene regulation. The first line shows the actual objective function used to optimize the sequences for the sRNA as well as the regulatory 5'UTR. A ... 5'UTR + 40nt of the GPF reporter gene; B... sRNA; AB... mRNA/sRNA complex; E_X... gibbs free energy; Z_X... partition function; S_{AB}... states containing input structure of AB; β... weighting factor; $[A_0]$... input concentration of mRNA ($[B_0]$ >> $[A_0]$)

Conclusion

We provide a software solution in form of the RNAdesign library which makes it possible to uniformly sample RNA sequences compatible to structural and sequence constraints. This makes it possible to efficiently sample from the whole solution space with avoiding the heavy re-evaluation of repeatedly generated solutions. Thereby it is possible to review much more solutions with the same effort, which leads to better results. Scripting interfaces make it easy to freely combine different optimization algorithms and incorporate evaluations of different software packages into the objective function. Our framework does not restrict the user to the ViennaRNA [5] and NUPACK [4] package only.
As it is possible to incorporate almost any software in the calculation of the objective function, it is now feasible to explore a much broader range of objectives. We showed one example on how to use the developed software to implement a design method using a *de novo* objective function that incorporates terms which were not available in other existing design frameworks [3].

References

[1] Christoph Flamm, I. L. Hofacker, S. Maurer-Stroh, P. F. Stadler, and M. Zehl. Design of multistable RNA molecules. RNA, 7(2):254 265, February 2001.
[2] Christian Höner zu Siederdissen, Stefan Hammer, Ingrid Abfalter, Ivo L. Hofacker, Christoph Flamm, and Peter F. Stadler. Computational design of RNAs with complex energy landscapes. Biopolymers, 99(12):1124–1136, 2013.
[3] Akito Taneda. Multi-objective optimization for RNA design with multiple target secondary structures. BMC Bioinformatics, 16(1):280, September 2015.
[4] Joseph N. Zadeh, Conrad D. Steenberg, Justin S. Bois, Brian R. Wolfe, Marshall B. Pierce, Asif R. Khan, Robert M. Dirks, and Niles A. Pierce. NUPACK: Analysis and design of nucleic acid systems. Journal of Computational Chemistry, 32(1):170–173, 2011.
[5] Ronny Lorenz, Stephan H. Bernhart, Christian Höner zu Siederdissen, Hakim Tafer, Christoph Flamm, Peter F. Stadler, and Ivo L. Hofacker. ViennaRNA Package 2.0. Algorithms for Molecular Biology, 6(1):26, November 2011.