

Supplementary Data for RAG-Web: RNA Structure Prediction/Design using RNA-As-Graphs

Grace Meng¹, Marva Tariq², Swati Jain¹, Shereef Elmetwaly¹, and Tamar Schlick^{1,3,4,*}

¹Department of Chemistry, New York University, New York, NY 10003, USA

²Department of Chemistry, Smith College, Northampton, MA 01063, USA

³Courant Institute of Mathematical Sciences, New York University, New York, NY, USA

⁴NYU-ECNU Center for Computational Chemistry at NYU Shanghai, Shanghai, China

*Corresponding author: schlick@nyu.edu

S1 RAG background and tools

This section provides an overview of our RNA-As-Graphs (RAG) approach, including tree graph definitions and brief descriptions of the RAG tools utilized in our webserver.

S1.1 Tree graph representation

In RAG, RNA secondary (2D) structures are represented as undirected tree graphs [1, 2]. Unpaired single-stranded RNA regions, called *loops*, are represented by vertices, and base-paired helices with at least two canonical base pairs (AU, GC, or GU wobble) are represented by edges. Single isolated base pairs are ignored, as are single-nucleotide bulges and internal loops, and pseudoknots. Figure S1(a) shows the 6-vertex tree graph for a thi-box riboswitch. Using graph enumeration methods, we have previously generated a library of tree graphs up through 13 vertices (approximately 260 nucleotides) [3] (partly shown in Figure S1(b)). We have used clustering techniques to classify tree graph topologies as “existing” (corresponding to known RNA structures), “RNA-like” (more likely to correspond to yet undiscovered RNAs), and “non RNA-like” [3, 4]. The RNA-like topologies serve as candidates for sequence design (see Subsection “Sequence design with F-RAG”).

S1.2 Graph topology sampling with RAGTOP

Our RNA-As-Graphs Topology Prediction (RAGTOP) protocol uses hierarchical graph-sampling with Monte Carlo/Simulated Annealing (MC/SA) to generate candidate graph topologies [5]. Figure S1(c) shows the different RAGTOP steps. First, the coaxial stacking and helical arrangement for RNA junctions in the RNA 2D structure is predicted [6, 7], and then the corresponding 2D tree graph is scaled to a tertiary (3D) tree graph by defining additional vertices and edges [5]. Vertices are added to represent the 5' and 3' helical ends and internal loops and bulges with fewer than two nucleotides. The edges of a 3D tree graph now connect the loop and adjacent helical-end vertices or two end vertices of the same helix, and their lengths are scaled by the number of nucleotides in corresponding loops and helices.

For MC moves, the twist and bend angle around the internal loop vertices (with helical-end vertices on each side) are sampled, either within a restricted angle range (“restricted” sampling) or without restriction (“random” sampling). If the SA protocol is performed, the system temperature is gradually cooled with each iteration. The orientation of the junction is kept fixed during the MC/SA simulation. Graph topologies are scored using our knowledge-based potential that contains terms for bend and twist angles around internal loops and radius of gyration; the objective function was updated to treat k-turn or non k-turn internal loops differently [8]. If the given structure does not contain any internal loops, all generated graphs receive identical scores. The generated graphs are accepted or rejected based on the probability of their scores. The

accepted candidate graphs produced from the MC/SA protocol are used as targets for 3D atomic model generation (see Subsection “3D model building with F-RAG”).

S1.3 Graph partitioning and motif search with RAG-3D

Our RAG-3D graph partitioning identifies all possible subgraphs for an RNA tree graph, and our RAG-3D search tool finds similar submotifs in known RNA structures for each subgraph [9]. The search tool uses our RAG-3D database that contains subgraphs and corresponding atomic fragments from ≈ 1500 RNA 3D structures, catalogued for 51 different tree graph topologies. Using these tools, the tree graph is partitioned into smallest possible subgraphs that maintain junctions intact; the initial subgraphs that share a common vertex are then hierarchically merged to extract all possible distinct subgraphs (up to 13 vertices). Each subgraph is then assigned a graph ID. All subgraphs of the query tree graph are structurally aligned to all 3D graph fragments with the same graph ID in the RAG-3D database, and the fragments with the lowest Root Mean Square Deviations (RMSD) are identified as the best matching fragments.

The atomic fragments corresponding to the best matching graph fragments are used as input for fragment assembly to build atomic models for structure prediction, as shown in Figure S1(d). All atomic fragments with the same graph ID as target subgraphs are used as input to fragment assembly for sequence design, as shown in Figure S1(e).

S1.4 3D model building with F-RAG

Our fragment assembly for RNA-As-Graphs (F-RAG) algorithm produces atomic models for a target RNA 2D structure and corresponding 3D tree graph (generated by RAGTOP) [10]. Figure S1(d) shows the steps for building 3D models using F-RAG. First, RAG-3D partitioning and search is performed to partition the target graph into subgraphs and find the best matching atomic fragments for each subgraph (see above). These subgraph combinations and atomic fragments are used as inputs for F-RAG. To build a full atomic model, F-RAG assembles the atomic fragments, using nucleotides corresponding to common subgraph vertices. The length and sequence of fragment loops and helices are adjusted to match the target 2D structure (some junction or 5'/3' nucleotides may be missing). F-RAG also uses a separate template library to add extra nucleotides to internal loops and hairpins if necessary.

The 3D tree graph corresponding to each generated model is scored using the knowledge-based potential created for RAGTOP [5, 8]. The output of F-RAG includes the generated 3D atomic models and the corresponding number of nucleotides, 3D tree graphs, and their scores. The generated atomic models can be geometry optimized or energy minimized before further use. We have previously tested our F-RAG algorithm to generate 3D atomic models for 50 RNA structures. F-RAG produced atomic models with an RMSD of less than 10 Å for 46 of the 50 structures, and the best atomic models for 28 of 50 structures, when compared to two other similar 3D structure prediction programs [10].

S1.5 Sequence design with F-RAG

Our F-RAG algorithm also designs sequences and corresponding atomic models to fold onto a target RNA-like tree graph topology [11]. Figure S1(e) shows the steps for generating candidate sequences using F-RAG. First, the target tree graph topology is partitioned into subgraphs using RAG-3D, and atomic fragments corresponding to each subgraph ID from the RAG-3D database are used as the inputs for F-RAG (see above). These are then filtered based on the number and connectivity of the loops as compared to the target subgraph. F-RAG then assembles the remaining atomic fragments for each subgraph using common vertices. The identity and number of nucleotides in the fragments are left unchanged to provide additional flexibility to the design process and generate novel combination of sequences. Any unpaired nucleotides at the 5'/3' ends are removed.

Similar to structure prediction with F-RAG, the 3D tree graphs for the generated atomic models are scored using our knowledge-based statistical potential [5, 8]. The sequences of the generated models are produced as output, along with their 3D atomic models, scores, 3D tree graphs, and 2D structures. The generated sequences can be further screened using *in silico* 2D structure prediction programs. We have previously used our F-RAG design procedure to design sequences to fold onto 6 RNA-like topologies [11].

S2 Webserver description and usage guidelines

Figure S2 shows the “Home” page of our webserver that introduces its features. The “Information” page includes tree graph definitions and instructions on how to use the webserver, along with sample results. Citation details, formats for input files, webserver browser compatibilities, and licenses for external programs used are also provided. The “Get Started” page contains the links to run the three modules of the webserver – RAG Sampler, RAG Builder, and RAG Designer.

Figure 1 in the main article shows the input/output and different components for the three modules. Detailed descriptions are also provided on their respective main pages. On each page, the user provides the input files or tests our webserver with the sample input. After the initial processing of the input files, each run is assigned an ID, and the user is redirected to the Results page, which can be bookmarked.

The Results page is updated to display the results as they become available, and can only be accessed by the full weblink, thus keeping the user data private. All results are available for download, and information relevant to the run (e.g., run ID, run time, number of models generated) is displayed. The user is also given the option to enter an email address to receive a notification after the results page is updated. The user also has the option to cancel the job (while still running) and delete data from the Results page. Below we describe the input required and the output produced for each of three modules, along with details of additional processing and analysis done on the results before they are presented to the user.

S2.1 Module I: RAG Sampler

RAG Sampler uses RAGTOP (see Subsection “Graph topology sampling with RAGTOP”) to generate candidate 3D tree graph topologies from the given RNA 2D structure.

S2.1.1 Input:

The user is asked to provide a RNA sequence and 2D structure (in .bpseq file format as shown in Figure S3, details are also provided on the “Information” page of the webserver). *The size of the RNA 2D structures is limited to 200 nucleotides.* The user can run RAGTOP with default parameters of random moves with simulated annealing (as these produced the best convergence in our previous studies) or choose different ones (restricted moves or no simulated annealing), as explained on the file upload page. The parameters for simulated annealing are fixed at present.

S2.1.2 Processing:

RAGTOP MC/SA sampling is performed on the uploaded RNA 2D structure. The MC/SA is performed for 50,000 steps, and all accepted graphs are produced as output.

S2.1.3 Output:

All accepted graphs generated by RAGTOP are produced as candidate graphs (in PDB format), along with their scores (as text file). The Results page also contains plots that show the distribution of candidate graph scores, as shown in Figure S4 for an example RNA structure (VS ribozyme, PDB ID: 2MIS). The user also obtains a link to automatically feed the lowest scoring candidate graph as the target to build 3D atomic models using our RAG Builder module.

S2.2 Module II: RAG Builder

RAG Builder uses F-RAG (see Subsection “3D model building with F-RAG”) to build 3D atomic models for a given RNA sequence, 2D structure, and target 3D tree graph topology.

S2.2.1 Input:

The user is asked to provide a RNA sequence and 2D structure (in .bpseq file format as shown in Figure S3) and the corresponding target 3D tree graph topology generated by the RAG Sampler module above (file format details are also provided on the “Information” page of the webserver). The user can also use the link

provided on RAG Sampler’s Results page to feed the best (lowest) scoring 3D tree graph directly into this module. As with RAG Sampler, *the size of the target RNA is limited to 200 nucleotides.*

S2.2.2 Processing:

RAG-3D partitioning and search is used to partition the target 3D tree graph into all subgraphs and extract 10 best matching atomic fragments from the RAG-3D database (see Subsection “Graph partitioning and motif search with RAG-3D”). Next, all pairs of subgraphs that meet the following conditions are identified as potential F-RAG inputs: 1) the combination of two subgraphs forms the full graph, with the first subgraph containing the first vertex; and 2) the two subgraphs have exactly one common vertex. If the target graph is partitioned into two subgraphs, there is only one such subgraph pair. If the target graph is partitioned into more than two subgraphs, all such subgraph pairs are identified, and the first pair is selected as the default. The user can choose to run model building with the default selection or any of the other identified subgraph combinations (drop-down menu provided).

F-RAG uses the selected subgraph combination and corresponding atomic fragments to build 3D atomic models (see Subsection “3D model building with F-RAG”). After completion, duplicate models are removed and all remaining models are ranked based on their nucleotide number (as some models may have nucleotides missing) and score. Top 20 (or fewer) lowest scoring models with the required nucleotide number are selected as the best models.

S2.2.3 Output:

The best models selected above are produced as output (in PDB format), along with their 3D tree graphs (in PDB format) and scores (as text files). The Results page also shows two plots that display information about number of nucleotides and scores for all models generated by F-RAG, with the best models selected above highlighted in blue. The Results page also indicates any possible failure to produce models with the required number of nucleotides or no models at all. Figure S5 shows the results generated by RAG Builder for an example structure of 7S.S SRP RNA (PDB ID: 1LNG).

The best atomic models are available for download, along with the generated plots and all atomic models (with their 3D tree graphs and scores) generated by F-RAG. Note that the atomic models are not geometry optimized or energy minimized. For best results, we recommend generating models with multiple subgraph combinations, followed by geometry optimization and/or energy minimization of the best models before further use. See details in the original F-RAG paper [10].

S2.3 Module III: RAG Designer

RAG Designer uses F-RAG (see Subsection “Sequence design with F-RAG”) to design sequences and corresponding atomic models that fold onto target RNA-like tree graph topology.

S2.3.1 Input:

The user is asked to provide a target tree graph topology in the form of an adjacency matrix (an example shown in Figure S3, details are also provided on the “Information” page of the webserver). *The size of the target tree graph topology is limited to 13 vertices.*

S2.3.2 Processing:

The uploaded target graph is assigned a graph ID, and RAG-3D partitioning is used to partition it into all subgraphs (see Subsection “Graph partitioning and motif search with RAG-3D”). Similar to RAG Builder, all pairs of subgraphs that form the complete graph are identified as potential input for fragment assembly with F-RAG. The user can choose the default selection or any of the other identified subgraph combinations (drop-down menu provided). The selected subgraph combination, along with all atomic fragments with the corresponding graph ID from the RAG-3D database are used as input to run F-RAG for design (see Subsection “Sequence design with F-RAG”) to generate novel sequences and corresponding 3D models.

After F-RAG completion, duplicate models and sequences are removed, along with any models that have large chain breaks (more than 5Å). The remaining sequences are ranked based on their model scores and

the top 200 (or fewer) sequences are retained for further analysis. The 2D structures of these top sequences are predicted using two *in silico* programs RNAfold [12] and NUPACK [13], and the predicted structures are assigned graph IDs. The sequences that are predicted to fold onto the target graph topology by both RNAfold and NUPACK (i.e., their predicted structure have the same graph ID as that of target graph topology) are returned as successful sequences. Note that the designed sequences can have different 2D structures but will fold onto the target topology, which is more general than a 2D structure.

S2.3.3 Output:

All successful sequences identified above are produced as output (as text files), along with their scores (as text files), associated 3D atomic models (in PDB format) and 2D structures (in .bpseq format). The Results page also contains two plots that display information about score and number of nucleotides in the top 200 (or fewer) sequences generated by F-RAG, with the successfully designed sequences marked as blue diamonds. The user can also visualize the 2D structures of the successful sequences on the Results page (drop down menu provided). The Results page will also indicate if this F-RAG run does not produce any successful sequences. Figure S6 shows the results generated by RAG Designer for an example target tree graph topology, 8.7.

The successfully designed sequences, along with their scores, 2D structures, and atomic models are available for download. The results page also provides a link to download all top 200 sequences and tree graph topologies for the 2D structures as predicted by RNAfold and NUPACK. Note that the successfully designed sequences may not fold onto the target topology using any other 2D structure prediction program. For best results, we recommend running F-RAG for multiple orientations and subgraph combinations of the target RNA-like topology. See details in the original design paper [11].

S2.4 Server Architecture

This webserver was developed in Python 2.7, Perl, and HTML5. Some client-side visual components were developed using jQuery. The underlying code for RAGTOP and F-RAG is written in C++, whereas the RAG-3D search and partitioning is written in Python. Visual plots of results are generated using the Matplotlib library and the Jinja2 library is used for HTML templating. External programs used by the webserver include PyMOL [14] for RMSD calculation in RAG-3D search, and RNAfold (available via ViennaRNA package version 2.4.10) [12] and NUPACK (version 3.2.2) [13] for RNA 2D structure prediction.

S3 Webserver performance and limitations

We have previously tested our structure prediction and design protocols on 50 3D RNA structures and 6 RNA-like tree graph topologies, respectively. Of the 50 structures, F-RAG produced models with an RMSD of less than 10 Å for 46 structures. Compared to two similar RNA 3D structure prediction algorithms, Vfold3D and 3dRNA, F-RAG generated the best models for 28 of the 50 RNA structures (please see full details in our original F-RAG paper) [10]. Our design pipeline successfully generated sequences for 5 of the 6 target topologies without any restrictions, two of which were also experimentally verified using SHAPE-Map [11]. Our graph-based design pipeline designs to a graph topology, rather than a specific 2D structure (used in most RNA inverse-folding algorithms) which provides greater flexibility in the design process and leads to the design of a variety of sequences, with variable 2D structures. Our webserver produces similar results when tested with the same target topologies. Recent design improvements will be reported and implemented separately (manuscript in preparation).

As an application, we used the RAG Designer module to design sequences for 5 additional RNA-like topologies. Table S1 lists the 5 RNA-like topologies, along with the number of runs performed with either different orientations (as each end vertex of the tree graph can be considered as the 5'/3' vertex) or subgraph combinations (as more than one pair of subgraphs can be used to construct the full graph). RAG Designer successfully generated sequences that fold onto the target topology for 4 of the 5 topologies. The variation in the lengths of the successfully designed sequences showcases the advantage of designing to a graph topology rather than a specific 2D structure, and the capacity of our webserver to design a variety of sequences.

Due to time and memory restrictions of the underlying code, RNAs with more than 200 nucleotides and tree graph topologies with more than 13 vertices are not accepted. Similarly, the subgraph combination to construct the full target graph consists of only two subgraphs. Larger RNAs or the use of more than two subgraphs may be examined with our standalone tools, but other caveats may apply; e.g., F-RAG does not build atomic models for graphs that have more than one junction in a subgraph.

Many improvements to both our webserver and standalone tools can be envisioned. Improvements to our F-RAG procedure to build atomic models with more than one junction in a subgraph and to avoid missing nucleotides in junctions and 5'/3' dangling ends¹ can be made. We are also currently in the process of updating our classification of tree graph topologies into “RNA-like” and “non RNA-like”, and our RAG-3D database to include atomic fragments using recently solved RNA structures. Such updates will enlarge the number of fragments available to F-RAG for model building and sequence design. We are also working on creating a similar pipeline for structure prediction and sequence design using dual graphs (that represent helices as vertices and loop strands as edges) that can also represent more complex RNA features like pseudoknots [4, 15, 16].

¹A dangling end refers to unpaired nucleotides next to stems at the 5' or 3' ends of the RNA sequence.

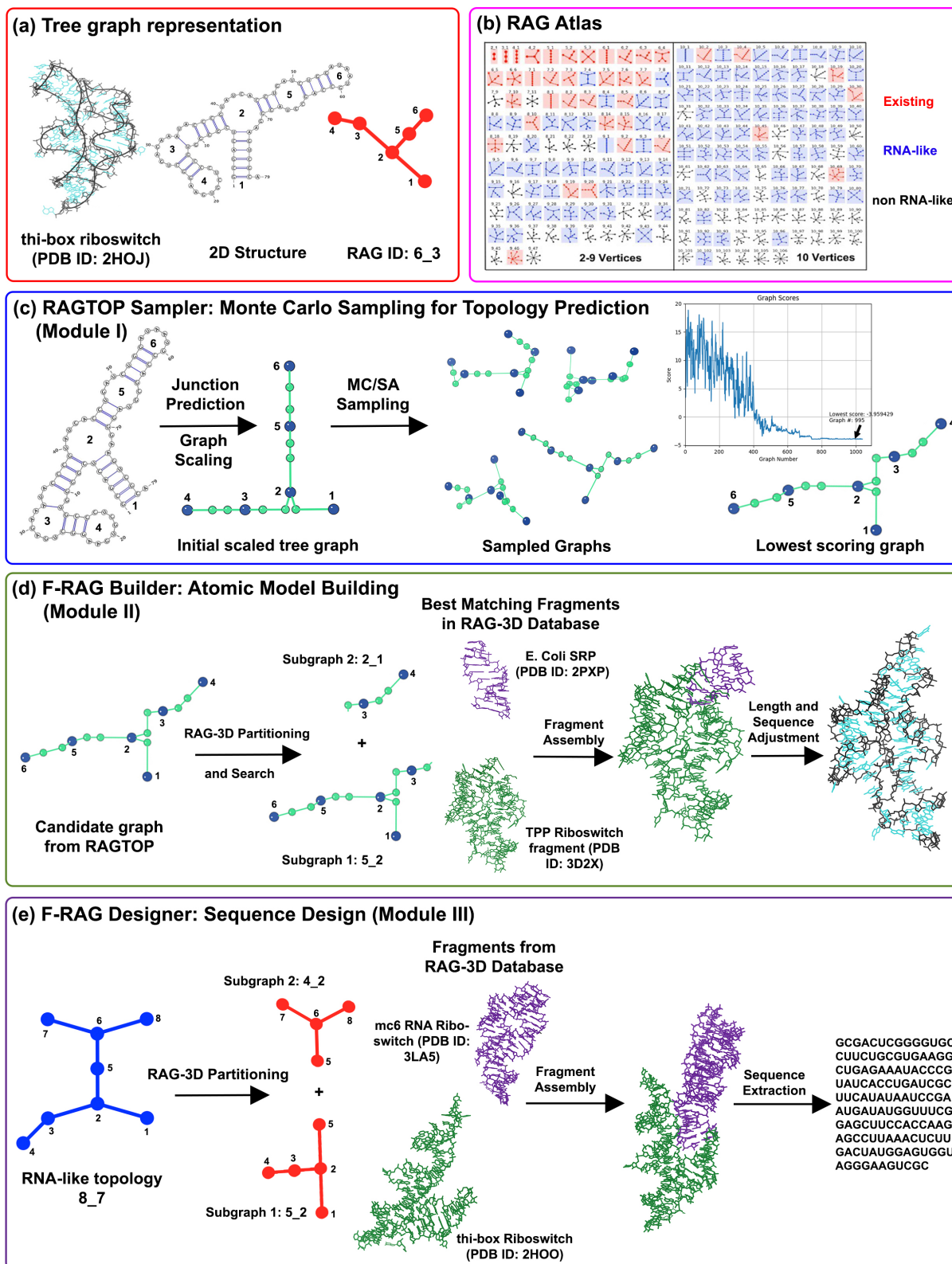


Figure S1: Depiction of RAG tree graph definitions and different components used in the webserver. (a) A 6-vertex tree graph shown for the thi-box riboswitch (PDB ID: 2HOJ). (b) The tree graph library is shown for 2-10 vertices. The graphs corresponding to existing RNA structures are shown in red, “RNA-like” graph topologies are shown in blue, and “non RNA-like” topologies are shown in black. (c) The steps followed by RAGTOP MC/SA sampling to generate candidate 3D tree graphs for a RNA 2D structure (shown here for the thi-box riboswitch). (d) The steps followed (using RAG-3D and F-RAG) to build 3D atomic models for candidate 3D tree graphs. (e) The steps followed (using RAG-3D and F-RAG) to design sequences for the target tree graph topology (shown here for RNA-like topology 8_7).

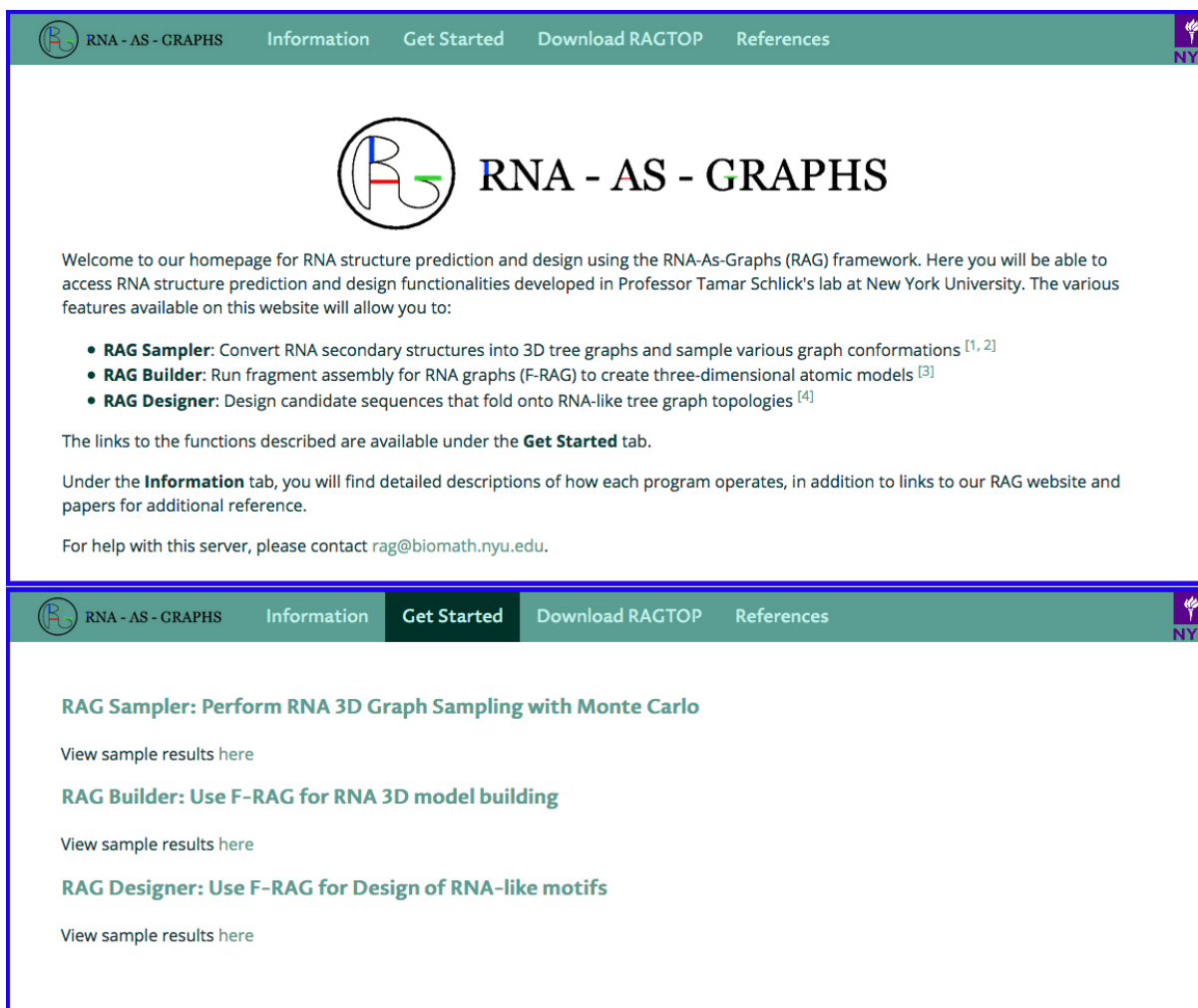
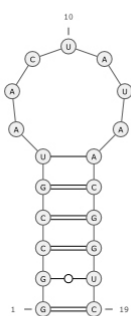


Figure S2: Screenshots of the “Home” page and “Get Started” page of our webserver with the links for running the three modules.

Table S1: Number of sequences generated by RAG Designer that are predicted to fold onto the 5 RNA-like tree graph topologies by both RNAfold and NUPACK.

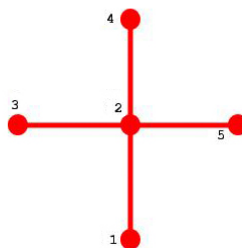
Target Topology	No. of Orientations and Subgraph Combos.	No. of Successful Sequences	Sequence Lengths
9_1	2	14	83 – 157
9_3	6	0	N/A
9_5	2	27	109 – 151
9_9	2	17	134 – 140
9_16	2	16	107 – 144

Example RNA and corresponding BPSEQ format file



1	G	19
2	G	18
3	C	17
4	C	16
5	G	15
6	U	14
7	A	0
8	A	0
9	C	0
10	U	0
11	A	0
12	U	0
13	A	0
14	A	6
15	C	5
16	G	4
17	G	3
18	U	2
19	C	1

Example tree graph topology and corresponding adjacency matrix



0	1	0	0	0
1	0	1	1	1
0	1	0	0	0
0	1	0	0	0
0	1	0	0	0

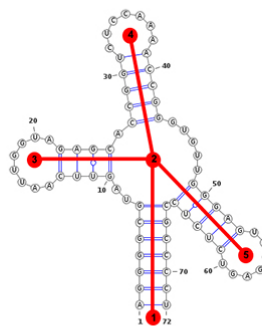
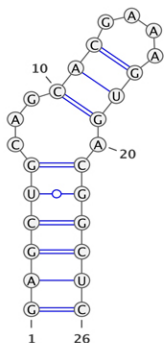


Figure S3: File formats used as input for our webserver. (a) The bpseq format file (.bpseq extension) consists of three columns; the first column provides the nucleotide number (starting from 1 and in increasing order); second columns provides the base type; and the third column gives the base-paired nucleotide, 0 if the nucleotide is not paired. (b) The adjacency matrix A for a V -vertex tree graph is a $V \times V$ dimension matrix, where each element a_{ij} is 1 if vertices i and j are connected, 0 otherwise. For RAG Designer, the vertices must be numbered in the 5' to 3' direction. A sample 2D structure corresponding to the adjacency matrix is also shown.

**Input 2D Structure
VS Ribozyme (PDB ID: 2MIS)**



Candidate graphs generated

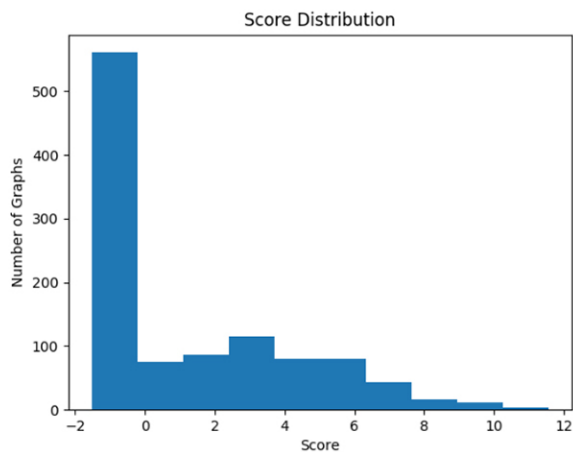
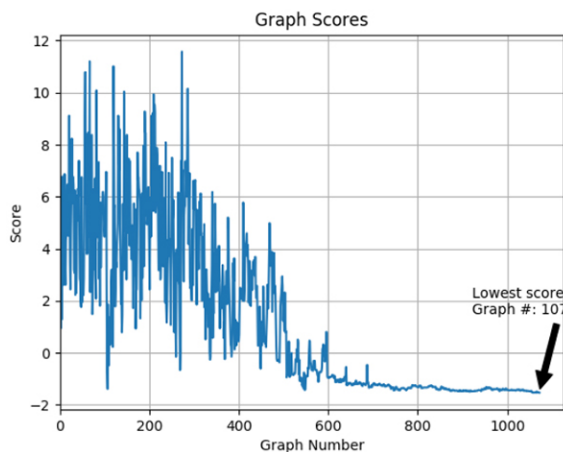
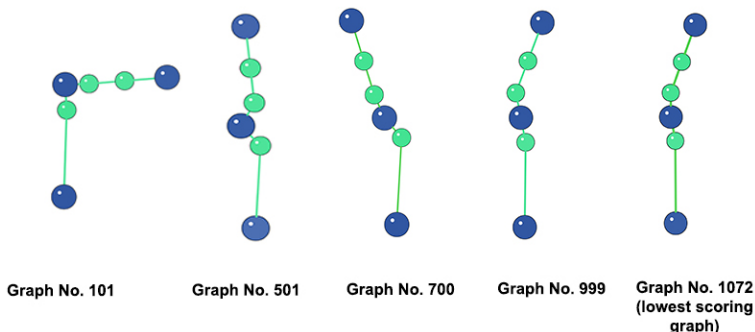


Figure S4: Input and output for RAG Sampler for a 26-nucleotide VS ribozyme fragment (PDB ID: 2MIS). The input 2D structure is shown, along with some of the candidate 3D tree graphs generated by RAGTOP. The plot titled “Graph Scores” shows the graph numbers plotted against their scores. This plot indicates the convergence of the MC/SA run as well as the position of the lowest scoring graph, which is marked. The second plot titled “Score Distribution” shows the distribution of number of candidate graphs according to their scores.

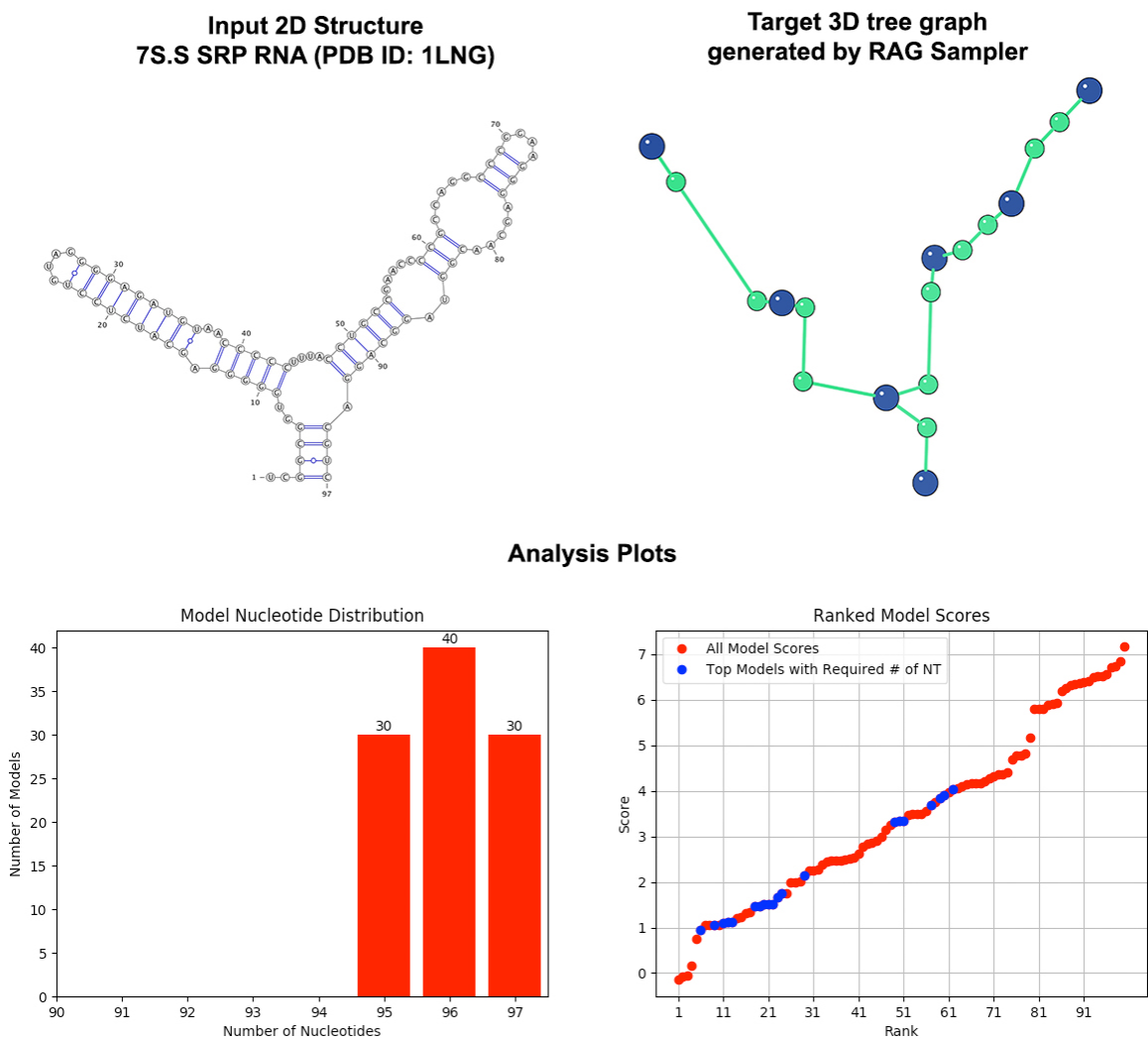
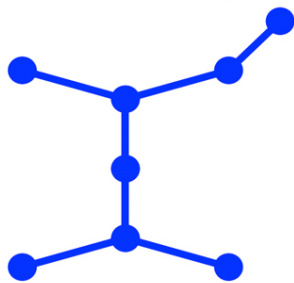


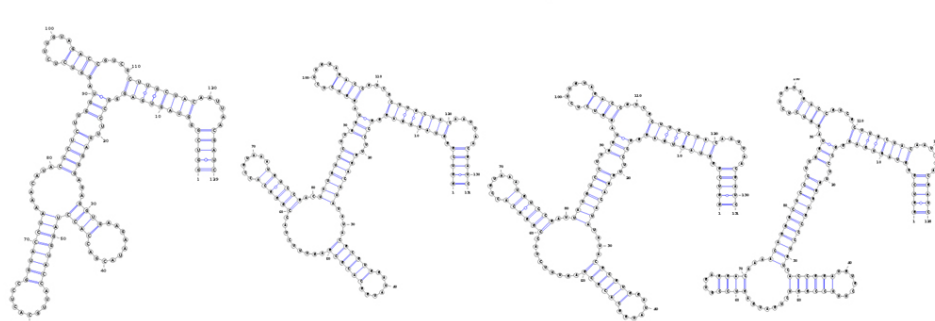
Figure S5: Input and output for RAG Builder for the structure of 7S.S SRP RNA (PDB ID: 1LNG) with 97 nucleotides. The input 2D structure is shown along with the target 3D tree graph (lowest scoring graph) generated by RAG Sampler. The plot titled “Model Nucleotide Distribution” shows the distribution of number of nucleotides for all generated models. This plot indicates how many models generated by F-RAG have all nucleotides and how many are missing nucleotides. The plot titled “Ranked Model Scores” shows the score of all generated models, with the best models with the required number of nucleotides highlighted in blue.

Input Graph Topology



RAG ID: 8_7

Successful Sequences



Analysis Plots

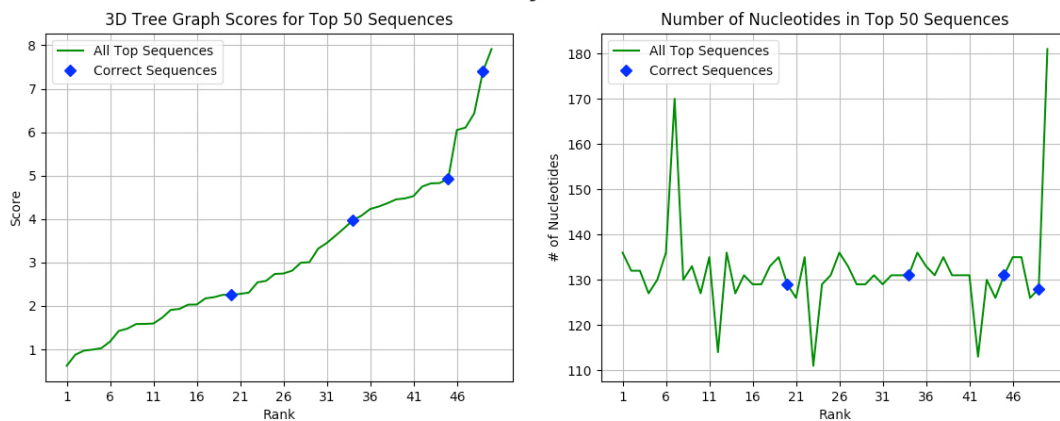


Figure S6: Input and output for RAG Designer for the RNA-like motif 8.7. The two plots show the score and nucleotide number for all top 200 (or less) unique sequences generated by F-RAG, with the successfully designed sequences (that fold onto the target topology with both RNAfold and NUPACK) marked as blue diamonds. The 2D structures of the 4 successfully designed sequences are also shown (available as a dropdown menu on the Results page).

References

- [1] H.H. Gan, S Pasquali, and T Schlick. Exploring the repertoire of RNA secondary motifs using graph theory; implications for RNA design. *Nucleic Acid Res*, 31(11):2926–2943, 2003.
- [2] D Fera, N Kim, N Shiffeldrim, J Zorn, U Laserson, H.H. Gan, and T Schlick. RAG: RNA-As-Graphs web resource. *BMC Bioinformatics*, 5(1):1, 2004.
- [3] N Baba, S Elmetwaly, N Kim, and T Schlick. Predicting large RNA-like topologies by a knowledge-based clustering approach. *J Mol Biol*, 428(5):811–821, 2016.
- [4] N Kim, N Shiffeldrim, H.H. Gan, and T Schlick. Candidates for novel RNA topologies. *J Mol Biol*, 341(5):1129–1144, 2004.
- [5] N Kim, C Laing, S Elmetwaly, S Jung, J Curuksu, and T Schlick. Graph-based sampling for approximating global helical topologies of RNA. *PNAS USA*, 111(11):4079–4084, 2014.
- [6] C Laing, D Wen, J.T.L. Wang, and T Schlick. Predicting coaxial helical stacking in RNA junctions. *Nucleic Acid Res*, 40(2):487–498, 2012.
- [7] C Laing, S Jung, N Kim, S Elmetwaly, M Zahran, and T Schlick. Predicting helical topologies in RNA junctions as tree graphs. *PLOS ONE*, 8(8):e71947, 2013.
- [8] C.S. Bayrak, N Kim, and T Schlick. Using sequence signatures and kink-turn motifs in knowledge-based statistical potentials for RNA structure prediction. *Nucleic Acid Res*, 45(9):5414–5422, 2017.
- [9] M Zahran, C.S. Bayrak, S Elmetwaly, and T Schlick. RAG-3D: a search tool for RNA 3D substructures. *Nucleic Acid Res*, 43(19):gkv823–9488, 2015.
- [10] S Jain and T Schlick. F-RAG: Generating atomic models from RNA graphs using fragment assembly. *J Mol Biol*, 429(23):3587–3605, 2017.
- [11] S Jain, A Laederach, S.B.V. Ramos, and T Schlick. A pipeline for computational design of novel RNA-like topologies. *Nucleic Acid Res*, 46(14):7040–7051, 2018.
- [12] R Lorenz, S.H. Bernhart, C Höner zu Siederdisen, H Tafer, C Flamm, P.F. Stadler, and I.L. Hofacker. Viennarna package 2.0. *Algo Mol Biol*, 6(1):26, 2011.
- [13] R.M. Dirks, J.S. Bois, J.M. Schaeffer, E Winfree, and N.A. Pierce. Thermodynamic analysis of interacting nucleic acid strands. *SIAM Rev*, 49(1):65–88, 2007.
- [14] Schrödinger, LLC. The PyMOL molecular graphics system. 2015.
- [15] J.A. Izzo, N Kim, S Elmetwaly, and T Schlick. RAG: An update to the RNA-As-Graphs resource. *BMC Bioinformatics*, 12(219), 2011.
- [16] S Jain, C.S. Bayrak, L Petingi, and T Schlick. Dual graph partitioning highlights a small group of pseudoknot-containing RNA submotifs. *Genes*, 9(8):371, 2018.