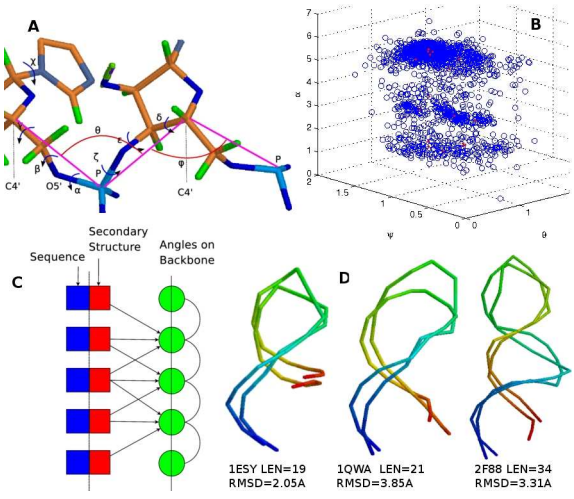
A New Probabilistic Model for RNA 3D Structure Prediction Using High-throughput Computing

Zhiyong Wang
Toyota Technological Institute at Chicago
zywang@ttic.edu

Motivation and Question Definition

RNA is becoming an important research subject in recent years and there is an increasing study of non-coding RNA in biology and health. Its growing important role appears in various life domains and processes including regulating gene expression [7, 9], interaction with other ligands [1, 2], and stabilizing itself [8]. To elucidate the function of RNA molecules, it is essential to determine their 3D structures. Experimental techniques for RNA structure determinations are time-consuming, expensive and sometimes technically challenging. As the computing technology advances, especially the high-throughput computing, computational methods are more often resorted for RNA structure determination.

Lots of papers have been published for RNA 2nd structure prediction and RNA 3rd structure prediction starts to gain attention in recent years. Methods for RNA 3rd structure prediction consist of two major components: an algorithm for conformation sampling and an energy function that can tell apart native from decoys. Fragment assembly, a method widely used in protein folding, has been implemented in FARNA [3] for RNA 3rd structure prediction. However, this method has several limitations: 1) the RNA conformation space formed by fragments is discrete and thus, may not contain a decoy close enough to the native; 2) there is no guarantee that a good fragment can always be identified for RNA without structure to be predicted; and 3) sequence information is not used for conformation sampling. MC-Fold [6] is another fragment assembly method, which uses a library of nucleotides cyclic motifs (NCM) to construct RNA structures. MC-Fold has a time complexity exponential with respect to the RNA length (i.e., the number of nucleotides in RNA), so MC-Fold may not be used to predict 3rd structure for a very large RNA. Recently, Frellsen et al [4] proposed a probabilistic model (BARNACLE) of RNA conformation space. BARNACLE uses a dynamic Bayesian network (DBN) to model backbone conformation. Although BARNACLE can sample conformations in a continuous space, BARNACLE does not make use of sequence information in sampling a backbone angle. In addition, BARNACLE only models the interdependency between two adjacent nucleotides, but not among more nucleotides.



Our Method

We propose a new machine learning method conditional random fields (CRF) [5,10] to model RNA sequence-structure relationship.

Different from the DBN method, our CRF method can model the sophisticated relationship among the primary sequence, 2nd structure and local conformation. That is, we can sample backbone angles using sequence and 2nd structure. Our method can also model RNA structure in a continuous space. Our method may be used to predict structures of large RNAs since our method can generate a conformation in linear time.

Our method uses a simplified representation of RNA backbone. That is, we model RNA backbone structure using torsion and planar angles on virtual bonds (indicated by pink lines in Figure A). We cluster the angles collected from PDB into twenty groups. The clustering result is shown in Figure B, in which the group centers are colored in red and the angles are in blue. We also use a probabilistic density function to model the angle distribution in each group. Finally, we use CRF to model the relationship between sequence and angles, as shown in Figure C. In our CRF method, the occurring probability of each group center at a given nucleotide not only depends on sequence information, but also on the angles at three adjacent nucleotides. Thus, we define a probability distribution on the structure space, which is used for the following sampling process.

High-throughput Computing

As shown in Figure E, A single computing job is to sample a RNA conformation with this CRF model trained from a set of known structures. First, we sample the group to which the angle at each nucleotide belongs from the distribution defined by our CRF model. Then we sample the real-valued backbone angles using the probabilistic density function of the sampled group and rebuild a 3D model from the sampled angles. This new 3D model is then subject to evaluation by a simple energy function (induced from only base-pairing information). The algorithm converges at a decoy with a local lowest energy value.

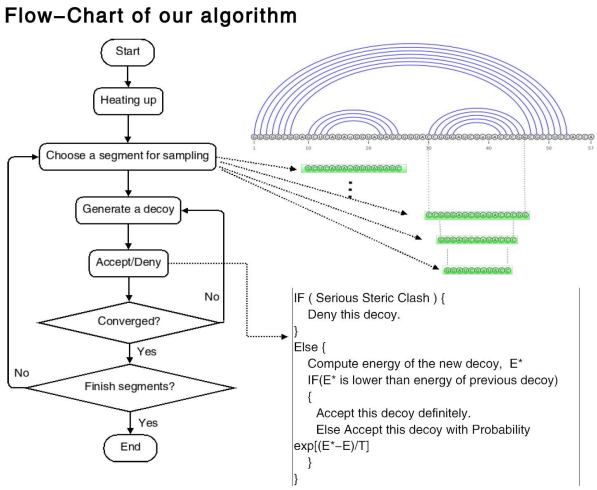


Figure E: The flow-chart of a single sampling process. The blue arcs are hydrogen bonds in RNA stems, and green highlighted parts show the order of substructures to be sampled.

We perform many replicates of this single process independently to generate enough decoys. Because of the inaccuracy of the energy function we used, we need to employ an extra step to select a good centroid structure from those decoys by clustering them. This step is also called consensus method, since it is suggested that a good structure be robust and have many good neighbors. Practically, we need more than 10,000 independent computing jobs. The independence among jobs enables them to be run separately on the computing nodes joined in OSG.

Considering the heterogeneous nodes in OSG, we code our algorithm in ANSI C++ which is

widely compatible with most of platforms and compile into 32bits binary with all necessary libraries statically linked. However, sometimes our program still will fail due to incompatibility or other unknown reasons. To make sure that a certain number of decoys will be generated, we use a DAG schema which includes a post-process node as the common child of all sampling jobs. All the failure jobs will be repeated before the post-process is ready to run.

The post-process performs the consensus step including clustering all the decoys by a RMSD distance, which is defined by RMSD value between two structures. The best structure among five centroids from the clustering is output as the final result. By far, the post-process can be done on a single computing node in several minutes. However, in case of more than 100,000 decoys, parallel computing of the distance matrix is very necessary. Since the distance matrix is consist of RMSD values of pairs of decoy structures, the computation of it can be discomposed into independent processes directly.

The large number of computing nodes joined in the OSG improves the efficiency of models comparison significantly. In our research, we compare scores of different models, such as models trained using different features or with different parameters. The comparison can answer the question that which factor in the model contributes more to the structure prediction. However, it will increase the running time by folds if running on a single workstation or a cluster with limited computing nodes. This can be done quickly, if there are many idle computing nodes accessible from OSG. To achieve this, we add a layer in the DAG script. The Figure F shows the comparison among three models. The first layer contains three pre-processes of three models, which set up parameters and prepare directories for the sampling program. The second layer does sampling decoys, which contains tons of jobs. Jobs in the third layer collect results, filter decoys by their energy values and cluster them. In case of large number of decoys, the third layer can also be discomposed into parallel jobs and an extra layer for collecting results of those jobs.

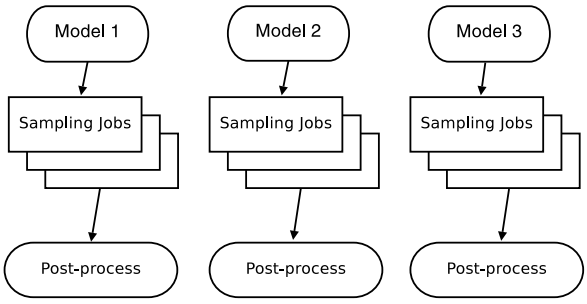


Figure F: The three layers of jobs in DAG submission to perform a model comparison. Each model generates a certain number of decoys which are clustered in the post-process steps.

Future Improvements

There are several methods to accelerate our research we plan to do. 1) We plan to setup a Condor system on our own workstation to reduce transferring large mount of data, such as generated decoys, which contains a large number of files. This also brings a great convenience to us in managing data and results. 2) We may shift the compiling from submitting node to computation nodes to increase the success rate of running. 3) Including more information as input features, we can develop a more sophisticated model to improve the efficiency of sampling structures, such as a CRF based on cyclic graph. However, it will cost more CPU time.

For all above, without OSG, I cannot do so many tests on my model and get good structures comparable to other methods. I appreciate all the people working for OSG. Thanks.

References

[1] Christopher S. Badorrek, Costin M. Gherghe, and Kevin M. Weeks. PNAS, 103(37):13640-13645, 2006.

[2] Amy H. Buck, Alexei V. Kazantsev, Andrew B. Dalby, and Norman R. Pace. Nat Struct Mol Biol, 12(11):958-964, Nov 2005.

[3] Rhiju Das and David Baker. PNAS, 04(37):14664-14669 2007.

[4] Jes Frellsen, Ida Moltke, Martin Thiim, Kanti V. Mardia, Jesper Ferkingho-Borg, and Thomas Hamelryck. PLoS Comput Biol ,5(6):e1000406, Jun 2009.

[5] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. In ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning , pages 282-289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[6] Marc Parisien and Francois Major. Nature, 452(7183):51-55, Mar 2008.

[7] Partho Sarothi Ray, Jie Jia, Peng Yao, Mithu Majumder, Maria Hatzoglou, and Paul L. Fox. Nature, 457(7231):915-919, Feb 2009.

[8] Cedric Reymond, Jean-Denis Beaudoin, and Jean-Pierre Perreault. Cellular and Molecular Life Sciences, 66(24):3937-3950, Dec 2009.

[9] David Solnick. Cell , 43(3, Part 2):667-676, 1985.

[10] Feng Zhao, Shuaicheng Li, Beckett W. Sterner, and Jinbo Xu. PROTEINS ,73(1):228-240, 2008.