Project topics for course CS-E4880

Machine Learning in Bioinformatics

RNA (Inverse) Folding Using Reinforcement Learning

March 29, 2019

Instructor: Mehdi Saman Booy (mehdi.samanbooy@aalto.fi)

Background

Biomolecular nanotechnology is a rapidly emerging area which provides the capability of making highly controlled nano-structural shapes for different purposes such as medical therapeutics. To this end, DNA and RNA are promising materials due to our ability to create synthetic linear strands which will then self-assemble into the desired shapes. The DNA origami technique, proposed by Paul Rothemund in 2006 [1], is a significant milestone which allows us to fabricate almost any 2D shape at nanoscale, recently extended also to 3D shapes. In biomolecular objects, structure defines function, thus being able to control their structure is an important goal.

Suppose we wish to create a nanoscale RNA or DNA object with a given structure. Then we need a sequence of DNA or RNA that folds, following fundamental biophysical principles, into the designed structure. Therefore, sequence design (also known as inverse folding) is crucial for achieving our goal. Mostly, sequence design is done for each problem separately regarding the structure and it takes a long time [2]. Moreover, as part of the design task, we need to predict the resulting structure (with minimum free energy) for a given sequence, a task known as secondary or tertiary structure prediction. Consequently, sequence design and structure prediction are two vital challenges in the design of RNA and DNA nanostructures.

Both of the aforementioned challenges are high-dimensional, and one cannot use simple search algorithms to find the desired answer. Secondary structure prediction can be solved in polynomial time [3, 4], while tertiary structure prediction (secondary structure with pseudoknots) is NP-complete [5]. One approach is complete tree search with pruning the answers to reduce the search space, by ignoring partial solutions that lead to invalid structures. Modern machine learning techniques can be used to make the search more intelligent. *Deep Reinforcement Learning* is great with respect to its learning capability and dealing with the lack of data. Two papers "Solving the RNA design problem with reinforcement learning [2]" and "Learn to Design RNA" [6] have used DeepRL for RNA inverse folding, although there is still plenty of room for improvement.

Goal

The goal of this project is implement a simple secondary structure prediction algorithm using Reinforcement Learning. You can choose to work on either of folding or inverse folding problems. After finishing this project, you have learnt about these things:

- Designing a environment for an RL framework
- Basic principles for RNA folding / inverse folding
- Approximation in RL using different methods like Neural Network

Due to complexity of general problem for folding / inverse folding, your program need to work on some sample sequences / structures and doesn't need to be general. Also, some constraints for the environment would be relaxed to make it easier. In general, you would try to test some estimator as policy function.

Materials and Methods

First of all, there is no dataset and the agent would learn with experiencing. Hence, you need to implement these classes:

- `Environment` which has principles of folding / inverse folding for RNA e.g. `pairing` function to connect to bases together or `free_energy` function to calculate free energy for a structure given a sequence.
- `Agent` which will learn what to do while interacting with your environment and getting feedback. For example, it has `step` function which does an action and return the reward and the next state.
- `Policy` is the internal model for the agent which help him/her decide what to do in each state.

You will get more in-detail information about these classes, their relationships, and their important functions. Besides, you will have important principles of RNA sequence and structure.

It is highly recommended to do your implementation in *Python*. To implement the policy you can use any estimator like Linear Regression or other things (e.g. using *scikit-learn*), but it is recommended to use a neural network. In this case, it is recommended to use *Pytorch* which is more straightforward to use, but feel free to use *Tensorflow* or any other frameworks (something familiar please :)).

* If you don't have any idea about using Pytorch and that is your reason to ignore this project, I can provide a simple neural network architecture also as an example.

Prerequisites

Programming skills (Python), Basic knowledge in Machine Learning, Some basic knowledge about RNA

References

[1] P. W. K. Rothemund, "Folding DNA to create nanoscale shapes and patterns," *Nature*, vol. 440, p. 297, 2006.

[2] P. Eastman, J. Shi, B. Ramsundar, and V. S. Pande, "Solving the RNA design problem with reinforcement learning," *PLOS Computational Biology*, pp. 1–15, 2018.

[3] J. S. McCaskill, "The equilibrium partition function and base pair binding probabilities for RNA secondary structure," *Biopolymers*, vol. 29, no. 6-7, pp. 1105–1119, 1990.

[4] E. Rivas and S. R. Eddy, "A dynamic programming algorithm for RNA structure prediction including pseudoknots.," *Journal of Molecular Biology*, vol. 285, no. 5, pp. 2053–2068, 1999.

[5] R. B. Lyngsø and C. N. S. Pedersen, "Pseudoknots in RNA secondary structures," in *Proceedings of the Fourth Annual International Conference on Computational Molecular Biology - RECOMB '00*, pp. 201–209, ACM Press, 2000.

[6] F. Runge, D. Stoll, S. Falkner, and F. Hutter, "Learning to design RNA," in *International Conference on Learning Representations*, 2019.