Chapter 1

Introduction

1.1 RNA Secondary Structure Prediction

In the nucleus of a cell, RNA is constantly being synthesized by transcribing sections of DNA onto a portable chain of the nucleic acids adenine, gaunine, cytozine, and uracil. These RNAs serve several functions inside of the cell, including acting as:

mRNA: 'messanger RNA', which travel to ribosomes to serve as blueprints for proteins,

tRNA: 'transfer RNA', which bond to and transport amino acids to the ribosome to be formed into proteins,

rRNA: 'ribosomal RNA', which make up ribosomes,

ncRNAs 'non-coding RNAs', which are not involved in the manufacturing of proteins, instead being used as tools of the cell for tasks including the regulation of gene expression.

The last group, ncRNAs, comprises the majority of RNAs synthesized and have mostly unknown functions. It is unlikely that they just float around the cell uselessly,

rather they are the tools the cell uses to build and run itself. It is also widely believed that the function of a strand of RNA is directly related to its structure. While DNA is composed of 2 seperate strands woven together in a double helix, RNA is most often found as a single strand that folds back onto itself. There are considered to be 3 main levels of structure of RNA. The first, the primary structure, is the sequence of nucleic acids that make up the strand of RNA i.e. 'GACCUUGGGGCCCC...'. The second, the secondary structure, is how these bases fold back on each other and form base pairs, most often of the Watson and Crick variety ('G-C', 'A-U', although sometimes 'G-U' is possible as well). The third, the tertiary structure, is how the structure bends on a larger scale as the stems and loops formed by the secondary structure interact with each other.

Primary structure can be readily observed by modern sequencing technology, however it is the secondary structure that determines the shape of the molecule, which is the most important when considering interactions with other biological molecules. The full specification of a secondary structure includes a list of every pair that is made by 2 of the nucleic bases in the primary structure making a bond. Finding the secondary structure of an RNA molecule is a different task than finding the primary structure of RNA because there is no single solution for one chemical chain. For an individual RNA molecule there are many valid pairings, in fact for a sequence of length n there are $O(1.8^n)$ secondary structures [TODO: cite].

To find which of these structures the molecule is likely to assume in nature, we turn to statistical mechanics. We approximate the RNA molecule as an isolated system in contact with a thermal resevoir that is the cell, with each secondary structure as a state of that system. In such a system the probability of any state s is its boltzmann factor divided by the partition function:

$$P(s) = \frac{1}{Z} e^{-\beta E(s)}, \tag{1.1}$$

where $\beta = \frac{1}{RT}$, R is the gas constant, T is the temperature, and

$$Z = \sum_{s} e^{-\beta E(s)}. \tag{1.2}$$

Initially researchers were satisfied with presenting the MFE (minimum free energy) state as the state the molecule assumes in nature, after all this state is the most probable. However it has become clear that this analysis is unsufficient, as MFE structures still are not very probable. Statistical procedures, sampling the Boltzmann distribution, are a more modern tool for predicting the secondary structure found in nature.

1.2 The Energy Model of RNA

Computing the partition function and probabilities is impossible without an energy model for RNA, $E(s)$. Setting the energy of the single-stranded (no pair) state as $E = 0$, the energy model must accurately estimate an energy for a folded secondary structure. In chemical experiments, this can be measured by using a strand with an known secondary structure and finding it's ΔG by deducing it from the relative concentrations of single-stranded states to double-stranded states in solution (see UV melting section).

The first energy models developed by researchers awarded energy bonuses to pairs formed. One of the first papers by Nussinov and Jacobson (1980) awarded the same amount of points to A-U and G-C pairs and found the MFE state, so the algorithm reduced to finding the legal folding with the most base pairs. After, it was discovered exprimentally that G-C pairs are more stable than A-U pairs. Because of this, in further iterations the energy would be determined by counting hydrogen bonds of cannonically paired bases, assigning each -1 kcal/mol of free energy. This would mean that GC pairs are given -3 kcals/mol, AU and GU pairs are both given -2.

The algorithm developed for this model minimized the free energy. Both this algorithm and the previous one had elementary dynamic programming solutions. They were useful models to use as a baseline, however, even the second iteration was not very accurate, on average only 20.5% of known base pairs are correctly predicted. Later energy models would use it as a control for the hypothesis that they increased secondary structure prediction accuracy (Mathews et al 1999).

Indeed, much improvement was made over the hydrogen bond model by expanding it to include what is now called the Nearest Neighbor model. Experiments made it clear that energy of an RNA folding is not just linearly dependent on the bonds that are made. There are significant interaction effects between nearby bases and bonds, this is called 'sequence dependence', and there are polymer physics based energy terms that scale logarithmically with the length of a loop (this can be thought of as the energy it takes to bend the strand). To handle these interaction effects in a model, we approximate that they are contained within loop regions. We divide our structure into its loops and compute the energy of each loop, with a seperate energy model for each.

1.2.1 Loop Parameters

One might wonder what the parameters, the terms, of this model would be. The model is simply that the energy of a structure, is the sum of it's loops:

$$E(s) = \sum_{l \in \text{loops}} E(l). \tag{1.3}$$

For example for a very predictably folding strand 'GGGAAACCC' (G's really want to pair with C's and A's resist pairing), we can decompose the energy as such:

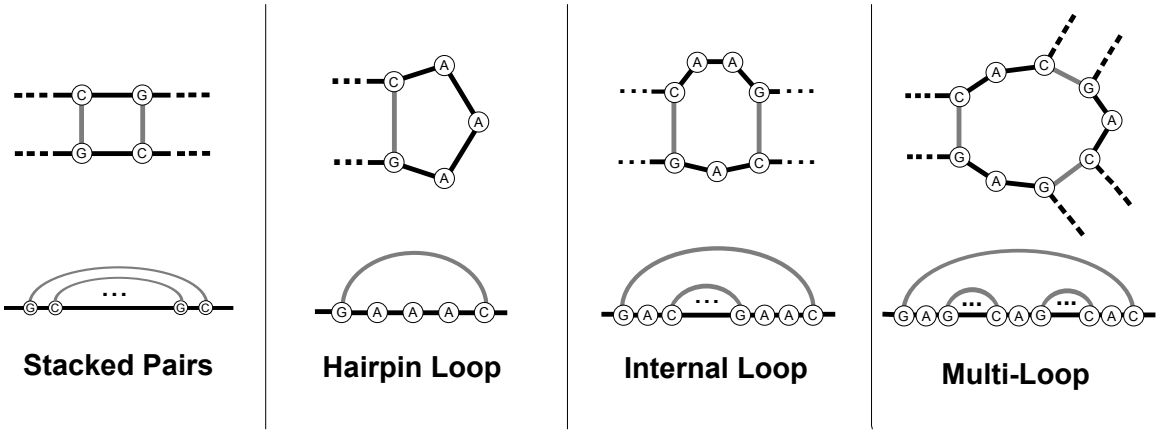


Figure 1-1: The 4 types of loops, black connections are bonds in the RNA backbone, grey connections are hydrogen bonds between bases. The types are: *Stacked Pairs*, adjacent pairs of bonded bases, *Hairpin Loops*, one bonding pair closing off a turn in the RNA backbone, *Internal Loops*, which can range from bulges to long loops, connecting to pairs with 2 chains of unpaired bases, and *Multi-Loops*, which connect 3 or more pairs.

$$\Delta G\left(\begin{array}{c}\text{G G G A}\\ \text{C C C A}\end{array}\right) = \Delta G_S\left(\begin{array}{c}\text{G G}\\ \text{C C}\end{array}\right) + \Delta G_S\left(\begin{array}{c}\text{G G}\\ \text{C C}\end{array}\right) + \Delta G_H\left(\begin{array}{c}\text{G A}\\ \text{C A}\end{array}\right) \tag{1.4}$$

Where ΔG_S is the function for energy of a stack loop and ΔG_H is the function that gives you the energy of a hairpin loop.

A model of loop energy should capture as much non-linearity as possible. For stack loops, the legal pairings limit the possible loops types. In fact, instead of worrying about how to model the loop's energy using polymer physics, we can just exhaust the possible stack loops, giving each a seperate parameter, i.e.

$$\Delta G_S\left(\begin{array}{c}\text{G G}\\ \text{C C}\end{array}\right) = \beta_{S1},\ \Delta G_S\left(\begin{array}{c}\text{G A}\\ \text{C U}\end{array}\right) = \beta_{S2},\ \Delta G_S\left(\begin{array}{c}\text{A A}\\ \text{U U}\end{array}\right) = \beta_{S3},\ \ldots \tag{1.5}$$

Where $\beta_{S1},\ \beta_{S2},\ \ldots,\ \beta_{Sn}$ are parameters to a linear regression model we fit to

ΔG of 'GGGAAACCC' and similar sequences (see section: UV Melting Experiments). This is what is done in practice.

For the other type of loops, the sizes can grow unbounded, so simply having a term for each loop will not work. However, at least for internal and hairpin loops, in keeping with the nearest-neighbor philosophy, we can keep a term for every combination of the bases that start the loop, and then have a general term for computing the length, addressing these both directly we have:

Hairpin Loop Loops with 3 and 4 unpaired bases are kept in special tables of triloop and tetraloop parameters, respectively. Each possible tetraloop and triloop has it's own energy determined by experiment. Beyond that, the general model is

$$\Delta G_H(n > 3) = \Delta G_{init}(n) + \Delta G_{HStack}(\text{Initializing Stack}) \tag{1.6}$$

$$+ \Delta G(\text{bonuses}). \tag{1.7}$$

As you can see there is an initialization term and a stack term that are both fitted via linear regression. The bonus term is for various special loops that have been experimentally found to be more stable. It's outside the scope of this thesis to get into them, but they are specified in the paper by Mathews et al (1999).

Internal Loop Much like stack loops, internal loops are given individual parameters for 1×1, 1×2, 2×2, internal loops, where $n \times m$ denotes one arm of the loop having n unpaired bases and the other having m. This results from extensive studies on the ΔG's of these internal loops. For the rest of internal loops there's a model of the form:

$$\Delta G_{int}(n \times m) = \Delta G_{init}(n+m) + \Delta G_{asymmetry}(|n-m|) \tag{1.8}$$

$$+ \Delta G(\text{bonuses}). \tag{1.9}$$

Where each term on the right is a regression parameter and the bonus term is similar to the hairpin bonus term for loops composed and ended with certain special kinds of bases determined experimentally to be more stable.

Multi-Loops Multi-Loops are harder to create experimental strands for and because of this there are no individual parameters for multi loops. In fact, for modeling mutli-loops we regress to a more simple linear model of the form:

$$\Delta G_{multi} = a_1 + b_1 n + c_1 h + \Delta G_{dangle} \tag{1.10}$$

Where a_1 is a penalty for starting a mutl-loop, n is the number of unpaired bases in the loop, b_1 is an energy penalty per unpaired base, h is the number of pairs in the structure, c_1 is the energy bonus per pair, and ΔG_{dangle} is a term similar to stack loop terms which include the energy of the unit composed of a pair and its 2 adjacent unpaired bases, if it has them.

[TODO: coaxial stacking?]

1.2.2 UV Melting experiments

Loop regions are given energies as parameters to linear regression models of free energy change in predictably folding strands. For example, the strand 'GGGAAACCC' folds predictably into a structure with all the G's paired to the C's and a 3-A hairpin turn (because G's pair very strongly to C's and A's tend to resist pairing) as seen in Equation 1.4. Large amounts of identical strands are synthesized and put into

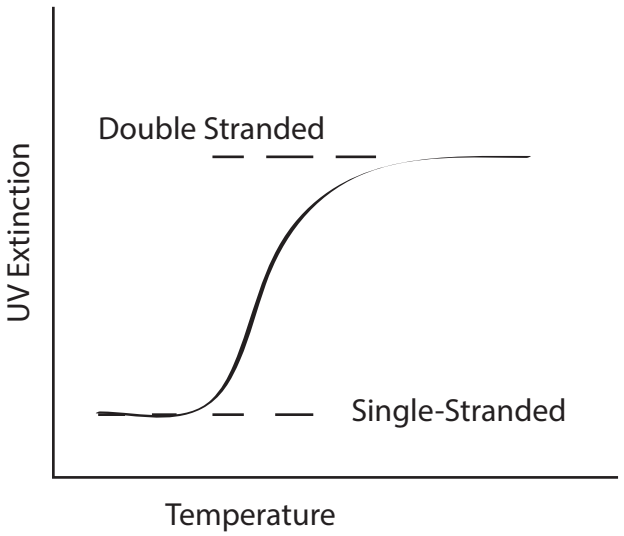
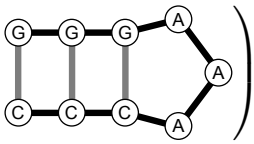
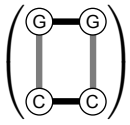
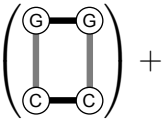
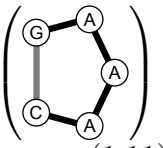


Figure 1-2: An example of the output of a UV melting experiment. We should expect to see 2 levels of extinction in the graph, one corresopoding to where single-strandedness is the equilibrium condition of the strand and the other where double strandedness is the equilibrium. As the temperature increases, the strand absorbs more energy from the environment allowing it to escape the double-stranded state and so there is a transition interval where as the temperature increases the UV extinction goes from the double-stranded extinction to the single stranded extinction.[TODO: this graph has double stranded and single stranded extinction switched]

---

solution and heated. A two-state assumption is made, either the RNA is folded in a 'double-stranded' state or unfolded in a 'single-stranded' state. As the solution heats, there is enough ambient energy to put all the strands in the unfolded, no-bonds state. RNA is an organic, aromatic molecule that absobes light in the UV spectrum in different amounts depending on whether it is in a folded state or unfolded state, so the UV absobtion is fit to a curve that then tells us about the relative concentrations of the single-stranded vs. double-stranded state, which in turn tells us about the free energy change between the two at a given temperature. This free energy change is extracted and treated as a function of the loop variables, which are then fitted to a linear model over experiments on many different such strands.

For example, if we wanted to compute the free energy change of a strand and fit it to it's loop parameters:

$$\Delta G \left(\text{  } \right) = \Delta G_S \left(\text{  } \right) + \Delta G_S \left(\text{  } \right) + \Delta G_H \left(\text{  } \right) \tag{1.11}$$

We synthesize large amounts of the strand 'GGGAAACCC', put them in solution and heat them and record their UV extinction. Observing the curve to be like something in Figure 1-2. The melting temperature T_m is defined to be where the concentrations of single stranded and double stranded molecules are equal, and this is taken to be the inflection point of the graph in Figure 1-2. From there Van't Hoof analysis is performed, where the concentration, C_T, is varied and this follows a model of the form:

$$\frac{1}{T_m} = \frac{R}{\Delta H} \log C_T + \frac{\Delta S}{\Delta H}. \tag{1.12}$$

This equation comes from the relation $\Delta G = -RT \log K_{eq}$ and plotting $1/T_m$ against $\log C_T$ and fitting a linear model gives us the parameters ΔS and ΔH from the slope and intercept and therefore ΔG through the relation

$$\Delta G = \Delta H - T\Delta S. \tag{1.13}$$

If we repeat this process over several strands, we can then fit the ΔG's to a linear model based on the many parameters described in the Loop Parameters section. From there, when we want to compute the energy of a folding, all we have to do is seperate it into loops and sum the energies of each based off the parameters of this model.

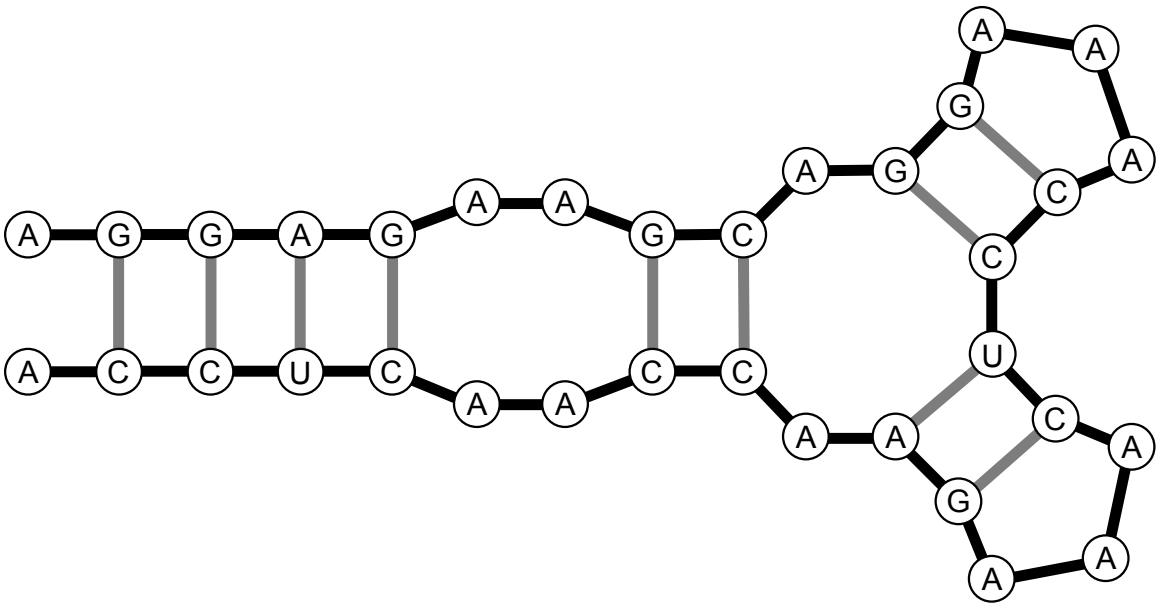


Figure 1-3: This is one possible folding out of the $O(1.8^n)$ secondary structures of the sequence 'AGGAGAAGCAGGAAACCUCAAAGAACCAACUCCA'.

1.2.3 Example Energy Computation

Here's an example of the energy computation of a secondary structure, pictured in Figure 1-3. There are 3 stack loop of the type:

$$\Delta G \left(\begin{matrix} G & G \\ C & C \end{matrix} \right) = -3.26 \text{ kcal/mol}, \tag{1.14}$$

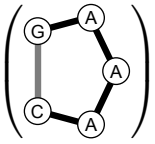
and one of each:

$$\Delta G \left(\begin{matrix} G & A \\ C & U \end{matrix} \right) = -2.35 \text{ kcal/mol}, \tag{1.15}$$

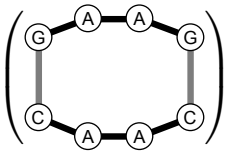
$$\Delta G \left(\begin{matrix} A & G \\ U & C \end{matrix} \right) = -2.35 \text{ kcal/mol}, \tag{1.16}$$

$$\Delta G \left(\begin{matrix} U & C \\ A & G \end{matrix} \right) = -2.08 \text{ kcal/mol}. \tag{1.17}$$

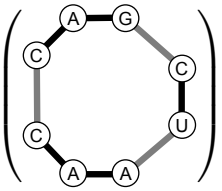
These are all lookups from linear regression parameters. Besides the stacks, there are 2 identical hairpins, an internal loop, an external loop, and a multi-loop. For the hairpins the energy can be looked up in a special parameter table designed specifically for hairpins with 3 unpaired bases called triloops:

$$\Delta G \left(\text{  } \right) = 5.8 \tag{1.18}$$

For the internal loop, there is a direct lookup for the 2×2 mismatch in a table:

$$\Delta G \left(\text{  } \right) = 0.5 \text{ kcals/mol.} \tag{1.19}$$

The multibranch loop has 3 pairs and 2 unpaired bases. The energy is therefore:

$$\Delta G \left(\text{  } \right) = 3.4 + 0 * 3 + 0.4 * 2 = 4.2 \text{ kcals/mol.} \tag{1.20}$$

If we sum up the contributions, we get that $\Delta G = -6.36$ kcals/mol. The energy of the MFE state can be computed using software, and I find it to be -3.06 with just the first stem and a large hairpin. [TODO: figure out what is wrong, make ct file, find energy].

1.3 Critique of the Energy Model

The energy model of RNA is not perfect, notably so. It is complex, there are linear models, non-linear models, lookup tables, and bonuses. Complexity is a very undesir-

able quality for a model, it makes it hard to implement and hard for new researchers to get started in the field. Not only that, but many parameters have very high error. For example, the initialization penalty for hairpin loops is supposed to be different for different hairpin lengths, but most of the terms aren't even 1σ away from each other. Because of this, it is not clear what advantage these extra parameters give, perhaps they could be reduced to one, to simplify the model greatly. Another problem is that the data of the original experiments has been lost, so no one can take a closer look at how well the parameters fit, and no one seems to be interested in running the experiments again. The only thing that seems to keep the current parameters in good standing is relatively solid results in prediction, with something like 80% of base pairs correctly predicted. However, to make any progress from this point, it is essential to get the energy model correct.

If there was a list of big problems to tackle in RNA secondary structure prediction, fixing the energy model should be a top priority. The new model should be consistent and physically based and confirmed by experiments. Some groups have made progress on creating energy models for multi-loops that may perform better than the current parameters [TODO: insert aalberts plug]. However, none of these changes have been implemented in the software packages that currently hold the grand majority of the secondary structure prediction market. There are reasons for this, including the fact that the software is massive, complicated, and poorly documented. Perhaps a new energy model could usher in a better software framework for RNA prediction that has:

1. Better naming conventions, with function, variable, and source file names describing precisely what they represent,

2. Consistently documented source files, perhaps with a full manual created with doxygen,

3. Simpler energy models, with less branching and complexity.

This is not to belittle the efforts of those that implemented Unafold and RNAStructure, after all they accomplished the great feat of implementing these complex software suites. Instead, it is to note that the job is not done, and the items listed above are things that need to be accomplished.

1.4 Applications of RNA Structure Prediction

RNA Secondary structure has many clinical applications including in sequence design.

[TODO: finish this section]