RNA Macrostates: Theory and Tools

Michael Flynn

A Thesis
submitted in partial fulfillment
of the requirements for the
Degree of Bachelor of Arts with Honors
in Physics

WILLIAMS COLLEGE
Williamstown, MA

Acknowledgments

This is the acknowledgements section.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The current free energy model of RNA is the result of evolution from simple model. In the first iterations of the RNA, energy would be determined by counting hydrogen bonds of cannonically paired. This would mean that GC pairs are given three units, AU and GU pairs are both given 2 [TODO: this is almost a quote from turner model paper, not sure if I could word it differently, what to do?]. This was a useful model to use as a baseline, and indeed it was used to design the first minimum free energy algorithm [TODO: cite?]. However, it was not very accurate, on average only 20.5correctly predicted, and 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 are called sequence dependent parameters. RNA is a polymer, something that bends and is flexible, and that bending costs energy. Many of the nucleotides in a given RNA sequence are not going to be paired to another base, and the energies contributed by these bases as they make up the loops of the secondary structure are going to be nonzero (unlike in the hydrogen bond model). In fact, the energies of these loop regions are experimentally found to be very different depending on what letters make up the subsequence that defines the loop [Todo: find citation

in Mathews paper]. Thus, what the Turner model does is find the energies of all the loop regions and add them up. Because the energy model treats the loop energies as independent of one another it is called a (or the) nearest neighbor model for RNA [TODO: this may not be precisely right, consult prof aalberts].

[Todo: loop region figure]

The energy of a loop is dependent on what type of loop it is. Different loops have different number of enclosing pairs, and this has consequences in the energy model. For example, a base pair stack, also called a helical region, is where we have two bases that are adjacent to each other pairing to two other bases that are also adjacent to each other. A general internal loop happens when there are one or more unpaired bases in between the would-be adjacent base pairs (no other pairs in between). An internal loop has a completely different energy model compared to a helical region, even though they are very similar. A multiloop can happen if we have a loop that is enclosed by more than 2 base pairs, and this has a different energy model still. In addition to these terms, energies associated with unpaired bases next to paired bases, called dangling bases, are included as well. Also a penalty for helices ending in AU's and general miscellanous terms as more papers were written and more energies were duct taped into the model [TODO: include kinder wording]. The specifics of the energy calculations for different loops are included in the following paragraphs, including how they are derived from experiments.

1.1 UV Melting Experiments

The thermodynamic behavior of an RNA strand can be determined by subjecting it to melting curve analysis. When a folded RNA strand denatures, or unfolds because it is heated, its absorbance of UV radiation changes. A physical model of this melting process can be developed to interpret these curves.

$$\Delta G = \Delta H + T\Delta S \tag{1.1}$$

$$T_M^{-1} = \frac{R}{\Delta H} \log\left(C_T/a\right) + \frac{\Delta S}{R} \tag{1.2}$$

The free energy of a loop structure, or rather its ΔG relative to the unfolded state,

Stacked Pairs The energy parameters for stacked pair loops were computed in a series of optical melting experiments (estimating parameters by UV absorbtion) by Xia et al (1998). For each combination of 2 sets of 2 paired bases, the change in energy at 37 Kelven was computed by fitting ΔH and ΔS to the data. [Todo: decide whether to have an in depth discussion of this]

For GU, a non-cannonical base pair that happens nontheless, the free energy is calculated by subtracting the free energy of a CGUACG strand from the free energy of a CGUUGACG strand, both of whose free energies are determined by optical melting experiments.

Dangling ends and terminal mismatches [TODO: read serra turner 1995]

bases adjacent to GU pairs are treated the same as bases adjacent to AU pairs

Hairpin loops The energy function for a hairpin loop is a similar table lookup.

[TODO: finish this]

- Tetraloop bonus

Bulge loops

Internal loops - 2x2 tandem mismatches - 2x1 internal loops - 1x1 (single mismatch

Chapter 2

Partition Function Computation and Improvements

2.1 Introduction

As a quick review, the partition function for a thermodynamic system of fixed volume, in contact with an environment with temperature T, is

$$Z = \sum_s e^{E(s)/RT} \tag{2.1}$$

where s denotes a particular state of the system, $E(s)$ is the energy of that state, and RT is the gas constant multiplied by the temperature, specified above. Each particular term in the sum is called that state's Boltzmann factor. The probability of a state is then said to be its Boltzmann factor divided by the partition function, or

$$P(s) = \frac{e^{E(S)/RT}}{Z}. \tag{2.2}$$

For an RNA molecule, we want to compute the probability of a particular folding or group of foldings, so we treat it as a thermodynamic system and sum up the

energies of each state, which is a particular folding. The energy that we assign to an RNA folding is determined by the Turner Free energy model, mentioned in the introduction. According to this model, the energy of an RNA folding is the sum of the energies of the loops that are created by the folding. These energies are added, for the most part, linearly. This means if the partition function for some small segment of the strand is computed, it will have the same contribution to the partition function of any larger segment that contains it. So we can spare ourselves from enumerating every single folding by using an approach that saves the results of these sub-computations in a table, such as dynamic programming.

The dynamic programming algorithm for computing the partition function of an RNA strands has several versions. If you ignore psuedoknots, and if you make an approximation that internal loops will never exceed a certain length, there is a general agreement that the fastest algorithm runs in $O(n^3)$, where n is the length of the strand. We believe that we can streamline this computation even more, taking advantage of the fact that empirically, the number of probable base pairs of a strand of length n seems to grow like n, not n^2. This is the same result we used to speed up the stochastic traceback algorithm and [TODO: see if this actually works].

2.2 Motivation

In certain situations, such as partition function clustering, the partition function is computed and recomputed several times. If the partition function takes on the order of hours or days to compute, this can make partition function clustering a bad option. However in these situations it is also true that the partition function is recomputed with almost the same properties, just certain pairs restricted. This motivates a method of computing the partition function using a known pairs heueristic to prune away unneccesary computation.

This concept has already been implemented to great success in the stochastic traceback algorithm. We've been able to show via experiment that the partition function only admits roughly $O(n)$ pairs with probabilities above thresholds around the machine precision limit. If we have the partition function already computed, we can recompute it by only adding in pairs that have sufficient probability. We can also extend this method: if a good heuristic appears in the future, one that can eliminate a large number of pairs, while being computationally cheap, we should be able to use the results to speed up the partition function computation.

2.3 Computation

The standard way of computing the partition function involves filling out a table where the (i, j) member represents the partition function for the substrand from base i to base j. Because the energy model for RNA is (mostly) linear, the partition function from i to j can be expressed as a function of nearby members of this table. This function is the recurrence relation for the partition function of RNA. Because the free energy model is so complicated and has gone through many iterations, differenct RNA folding software packages implement different versions of the recurrence relation, and they vary widely in complexity.

The definitive representation of the recurrence relation for RNA was formulated in 1990 by J.S. McCaskill in his landmark paper *The Equilibrium Partition Function and Base Pair Binding Probabilities for RNA Secondary Structure* [TODO: cite?]. The formula is also presented better and explained well by a later paper by Dirks and Pierce in 2003 (Dirks Peirce 2003). Starting at the outermost layer of this relation, the formula for the partition function of the strand from base i to base j is:

$$Q(i,j) = 1 + \sum_{i \leq d < e \leq j} Q(i, d-1) Q^b(d, e) \tag{2.3}$$

The theory behind this formula is that the partition function is a sum of the empty state (the first term, 1) and the state with at least 1 pair, the furthest pair to the right being pair (d, e). The term $Q^b(d, e)$ is the partition function assuming that base d and base e are paired. This function has the following recursion relation:

$$Q^b(i,j) = e^{-\frac{Hairpin(i,j)}{RT}} + \sum_{i \leq d < e \leq j} e^{\frac{Interior(i,d,e,j)}{RT}} Q^b(d,e) + \sum_{i \leq d < e \leq j} Q^m(i+1, d-1) Q^b(d,e) e^{-\frac{\alpha_1 + 2\alpha_2 + \alpha_3(j-e-1)}{RT}} \tag{2.4}$$

The theory behind this formula is that the partition function for a strand assuming i and j are paired includes 3 cases:

1. There are no bases paired between i and j, the loop is a hairpin and uses the energy function for a hairpin loop, we call $Hairpin(i, j)$, which consists of data table lookups.

2. There is an internal loop between i and j and a second pair d and e. This uses a different energy model, we call $Internal(i, j)$ and also consists of data table lookups.

3. There is a multiloop formed by the pair i and j, which must be carefully accounted for using a special model for multiloops.

The multiloop partition function, $Q^m(i, j)$ is the last piece of the puzzle. The formula is:

$$Q^m(i,j) = \sum_{i \leq d < e \leq j} e^{-\frac{\alpha_2 + \alpha_3(d-i) + \alpha_3(j-e)}{RT}} Q^b(d,e) + Q^m(i, d-1) Q^b(d,e) e^{-\frac{\alpha_2 + \alpha_3(j-e)}{RT}} \tag{2.5}$$

In english, this just means we sum up all the ways to just have 1 pair, and then all the ways to have more than one pair. The case with no pairs is not included, as

in the original recursion in Q^b, $Q^m(i+1, d-1)Q^b(d, e)$ must yield at least 2 pairs. Since Q^b makes one, then Q^m must make at least 1.

For example, the UNAFold software package implements a particularly hairy recurrence relation. Define $Q(i, j)$ as the partition function from i to j, $Q'(i, j)$ to be the partition function from i to j, assuming i and j are paired, and define $Q^1(i, j)$ to be the partition function from i to j, assuming exactly 1 pair happens on that interval, and that pair happens with base i. The recurrence relation is therefore

[recurrence relation]

Note the terms Z_{ND}, $Z_{3'D}$, $Z_{5'D}$, and Z_{DD} are extra free energy terms corresponding to 'dangle energies' which are the results of an experiment later implemented in the model to improve it from the standard energy model. In addition there are AU penalty terms appended to where pairs are made, as AU and GU pairs have penalties associated with forming. These additional energy terms improve the model's predictive ability and bring the model closer to the "truth", however it unfortunately makes the partition function seem very threatening.

Our new partition function relation has the following theory behind it: Assume we have the functions $I : B \to \{B\}$ and $J : B \to \{B\}$ that return the set of all probably pairs for a base i or a base j, respectively. The recurrence relation can be reformulated in the following way:

[new recurrence relation]

Note that for Q and in many places for Q', instead of a sum over the known k that could possibly begin a leftmost pair, we see a double sum. One of them over k that could end a leftmost pair, and this sum is limited to a certain length below j. This is just making the same assumption that the internal loop computation makes: there are not arbitrarily long strands without base pairs, after a certain number of bases it becomes overwhelmingly more likely to make a base pair that we can virtually ignore the energy of the the cases of length beyond a certain L.

As for the seocnd sum, since the number of probable pairs for a base i has been shown empirically to be roughly constant, regardless of length, the second sum is essentially constant. What this all means is that all $O(n^2)$ computations of $Q(i, j)$'s are roughly constant time. This means that the overall algorithm is $O(n^2)$, an improvement over the previous algorithms asymptotic bound by and order of n!

2.4 results

Chapter 3

Stochastic Traceback Algorithm and Improvements

3.1 Introduction

The stochastic traceback algorithm was introduced by Ye Ding and Charles Lawrence (2003) as a means to explore the energy landscape of RNA by sampling structures according to their Boltzmann probabilities. This was important because the minimum free energy structure was very sensitive to errors in the parameters of the free energy model, and although algorithms existed for generating suboptimal structures, they either sampled a very limited set of states (Zuker 1989), or had exponential runtime and did not correspond to the physical ensemble of states (Wuchty et al 1999).

The method uses the partition function algorithm as a forward-fill step, then it traces back over the contents of the tables allocated during that algorithm. Specifically, the tables $Q(i, j)$, $Q'(i, j)$, etc. now contain information about the conditional probabilities of bases pairing. The general principle of the backwards trace is that, presented with several possibilities for the structure along a sequence from i to j, the sampling probability for a case is the contribution to the partition function by that

case's partition function.

[TODO: figure of stochastic traceback algorithm]

The specific algorithm requires two stack data structures. A stack can be thought of as a literal "stack", like papers stacked on a desk, except instead of paper their are items of data. There are two basic operations, one to put an item on the top of the stack, and another to retrieve an item off the top. These are called "push" and "pop" operations, respectively, in Computer Science. The data items we will be pushing on to the first stack, A, are of the form $\{(i, j), b\}$ where i and j are indexes along the strand and b is either $True$ if we have determined that i and j are paired, or $False$ otherwise. The second stack, B, is where we'll collect pairs and unpaired bases for one sample.

The initialization of the algorithm is to push $\{(1, n), False\}$ onto the stack. From there the algorithm repeats the following steps:

1. Pop an element, $\{i, j, b\}$ off stack A.
2. Case b of $False$:
   (a) Pick a (k, l) where $i \leq k < l \leq j$ which is to be the rightmost pair on the segment, with the appropriate probability
   (b) Push $\{(i, k-1), False\}$ onto stack A, because the structures to the left of (k, l) are not yet determined
   (c) Push $\{(k, l), True\}$ onto stack A, because we need to determine what type of loop (k, l) encloses
   (d) Push (k, l) onto stack B as a pair
   (e) Push all m such that $l < m \leq j$ onto stack B as unpaired bases, as (k, l) is the rightmost pair
3. Case b of $True$:

(a) Choose what type of loop (i, j) is from {HAIRPIN, STACK, BULGE/INTERIOR, MULTI-LOOP} with the appropriate probability

(b) Push the appropriate elements onto the stack for that loop type, see figure.

4. If stack A is empty, the pairs and unpaired bases in stack B become a sampled structure. Reinitialize for additional samples.

In the preceding algorithm, I have reference the "appropriate probability" for each of the different choices. As stated before, these are the contributions to the partition function by these cases. In the framework of UNAfold, with the matrices $Q(i, j)$, $Q'(i, j)$, etc., the specific probabilities would be:

[TODO: note these are pretty much taken from Markham's thesis, but they are accurate, sooo?]

$$P_k(i, j) = \frac{\left(Q(i, k-1) + e^{-\frac{b(k-i)}{RT}}\right) Q^1(k, j)}{Q(i, j)} \tag{3.1}$$

which is the probability to pick any k for our rightmost pair (k, l), where $i \leq k < j$. Note that when summed over all values of k, the top becomes the definition of $Q(i, j)$, therefore these probabilities sum to 1.

$$[TODO : clarifyequationswithAalberts] \tag{3.2}$$

(3.3)

(3.4)

(3.5)

The probabilities are normalized and sampled from to output the first pair. From then on, if a pairs (h, l) is chosen, the algorithm chooses the type of structure that (h, l) encloses with the probabilities:

(3.6)

(3.7)

(3.8)

(3.9)

(3.10)

The algorithm continues from there on in a similar fashion, choosing from each case according to their partition function. Our innovation is to reduce the number of unnecesary computations. The partition function has already been calculated, so we already know which bases that might be paired and which bases are almost certainly not paired. By only checking the pairs that we know can happen, we see a large speedup.

3.2 Motivation

In the past 10 years, the stochastic traceback algorithm has become an increasingly central part of RNA secondary structure prediction algorithms (Ding et al 2005, [TODO: cite more]). This is because they present many advantages over the minimum free energy prediction. It can be shown that the minimum free energy state, even though it is the most probable state, can still have astronomically unlikely probabilities on average for typical strands of reasonable length ([TODO: cite, figure]). The more important concept in understanding the physical behavior of an RNA strand is therefore the overall shape of the energy landscape. Althogh the probability of any individual structure might be infinitesimally small, there can be shown to be relatively few large basins containing clusters of similar foldings. The consensus structures and the difference between the consensus structures of these basins define the function of the RNA molecule.

The way the stochastic algorithms probe that is by providing structures to group into these basins, and since the stochastic traceback algorithm samples states with the exact probability defined by the partition function, we know that the macrobehavior of these samples match what we would probably see in reality. There is one catch and that is statistical error. However, the error can be reduced and the landscape can be further explored the more stochastic samples we make.

The need to sample large numbers of secondary structures makes a speedup very convenient, and that is what motivates our current expedition.

3.3 Methods

Taking advantage of the empirical fact that the number of probable base pairs for an RNA strand tend to grow very slowly, we can restrict our traceback to only explore bases that we know can pair with one another.

3.4 Results

As one can see from the tables, the speedup is enormous. For randomly sampled sequences up to lengths in the thousands, the old stochastic timing grows quadratically, while the new method flatlines below it.

[TODO: add speedup figure]

A good question to ask would be, how do we know that this new algorithm is outputting structures with the correct probabilities. Verification plots here attempt to answer that question.

[TODO: add verification plot]

What we would expect to see from these plots, is that for a given base, we would expect to see it pair with other bases with probabilities given by the partition function as one can see. Of course there is sampling error, so each bin represents a sampling from a Bernoulli distribution. For $n^2 samples, we would expect [Todo: find out what error we expect] error. The number of samples that violate the bounds, do not deviate much fro$ experiments, so I think we can confidently say that the new algorithm is making the correct computation.

Chapter 4

Nestor and Clustering

4.1 Motivation

Given that we know, for any given RNA strand, the probability of an individual state is very low [TODO: reference section], a much more important computation is the overall shape of the strand's free energy landscape. Even if the probability of an individual state is low, if we "integrate" over a basin of free energy, the probability of that set of states could be something tangible.

In the past 10 years, several groups have started to explore this concept. There are two approaches, in general, to define basins and classify structures into them. The first class of methods defines the basins from the top-down: given a number of stochastically sampled structures, we divide them into groups based on some kind of distance metric. These methods tend to be very similar to typical clustering algorithms used in computer science and data analysis.

Another approach is to start at local minima and climb up the energy barriers between minima using the metropolis-hastings algorithm to maintain the correct probabilities according to the partition function. These methods can be used to accurately compute the energies of the transition states between local minima and these

can tell you the kinetics of the structure. This technique was developed by [TODO: find Vienna people and cite them].

Enabled by their stochastic sampling algorithm Ye Ding and Charles E. Lawrence clustering algorithms. [TODO: evaluate this].

Chapter 5

Reactivity Experiments

5.1 Methods

[Pretty rudimentary description of Das's process] Das Lab at Stanford perform chemical mapping experiments on RNA molecules. An RNA strand of interest is selected and is from there on called the wildtype strand (abbreviated WT). Then for each nucleotide in the strand a mutant is created switching out that particular base with its Watson and Crick opposite. This is intended to perturb the energy landscape in such a way that dominant loops may become less prominent and other foldings become more stable. SHAPE analysis is then done on each strand to prob which bases are paired and which are not.

Data was obtained in the form of RDat files from Stanford's RNA mapping database. SHAPE reactivity is extracted from these files for the WT and each of its mutants. To normalize the reactivity trace of a strand to a probability on $[0, 1]$ first the partition function is calculated for this strand, then probability of each base being paired is computed using the formula

$$basepair formula,$$

and finally these probabilities are rank sorted and fitted to a fermi distribution using a least squares gradient decent fit [figure here]. We believe that the measured reactivity should relate [correlate, correspond?] to the probability that a base is unpaired, so the reactivities are reverse rank sorted and mapped to the fermi distribution found by our fit.

From here, using the assumption that each mutation changes the relative energies of each macrostate without changing their internal structures, we fit this data to a model of k clusters each with n nucleotide probabilities, with then $k*(n+1)$ cluster probabilities. Therefore we have a model with $k(2*n+1)$ paramters fitting to $n*(n+1)$ data entries. A boxed gradient decent is used to minimize a cost function:

$Cost function$

This fit results in k fitted clusters with $k*(n+1)$ cluster probabilities.

These fitted clusters are compared to k nests generated by Nestor. The nests are created using the methods desribed in [Nestor chapter] for each strand. Since these nests are created independant of any other strands, nests for different strands must be matched to each other in order to compare to the fitted clusters.

[paragraph on the matching process, still investigating]

Once these matches are made we can compare the cluster vs nest probabilites for the WT and each mutant and see how they correlate, as well as investigate other clusters that may be found.

Chapter 6

Conclusion

Conclude conclude conclude

Appendix A

Tables

Table A.1: Armadillos

Armadillos	are
our	friends

Appendix B

Figures

Figure B-1: Armadillo slaying lawyer.

Figure B-2: Armadillo eradicating national debt.