1 Subjects

- What is RNA 2^{nd} structure?
- Computing a pseudo-knot free RNA 2^{nd} structure.

2 Notes

2.1 RNA and second structure

Messenger RNA is often described as a linear, unstructured sequence, only interesting for the protein amino acid sequence that it encodes.

However, many non-coding RNA's exist which adopt sophisticated three-dimensional structures, and even catalyse biochemical reactions. RNA is typically produced as a single stranded molecule, which then folds to form a number of short base-paired stem, this is what we call the seconday structure of the RNA.

RNA is a polymer of four different nucleotide subunits, we abbreviate them A, C, G and U. In DNA, thymine T replaces uracil U. $G-C$ and $A-U$ form hydrogen bonded base pairs. $G-C$ form three hydrogen bonds and tend to be more stable than $A-U$ pairs which form only two. Some non-canonical pairs also forms, like the $G-U$ pair, and others which distort regular A-form RNA helices.

Base pairs are approximately coplanar and are almost always *stacked* onto other base pairs. Such contiguous stacked base pairs are called *stems*. In three dimensional space, the stems generally form a regula (A-form) double helix. We typically represent the RNA 2^{nd} structure in two-dimensional pictures.

Single stranded subsequences bounded by base pairs are called loops. A loop at the end of a stem is called a *hairpin loop*. Simple substructures which just consist of a stem and a loop is called *stem loops* or *hairpins*. Single stranded bases occuring within a stem are culled a *bulge* or *bulge loop* if the single stranded bases on only on side of the stem. it's called an *interior loop* if there is a bulge on both sides. If a loop connects three or more stems, then it is called a *multibranched loop*.

Base pairs almost always occur in a nested fashion in RNA secondary structure. Base pairs are nested if we can draw arcs over them, and none of the arcs intersect. Formally, if i, j is a base pair and i', j' is a base pair then $i < i' < j' < j$. If it happens that these arcs would cross, then they are called *pseudo-knots*.

Just to spell it out, this is nested:

This is juxtaposed:

This is overlapping (pseudo-knot):

If there are no pseudo-knots, then we can represent it as a planar graph, and in general it is easier to find the compute the secondary structure with the least "free energy" without the pseudo-knots. Fortunately, there are very few pseudo-knots compared to the number of base pairs in nested secondary structure, so it is usually acceptable to sacrifice the information in pseudo-knots in return of efficient algorithms.

2.2 Predicting 2^{nd} RNA structure

Usually, when we want to predict the secondary stucture, we will try to minimize the amount of "free energy". The first example we will look at, bases the prediction on the primary structure (the simple sequence) only. For this we have Nussinov and Zuker's Mfold algorithm. Other methods use comparative structure prediction which is based on a prior alignment. As well as probabilistic methods.

2.2.1 Nussinov

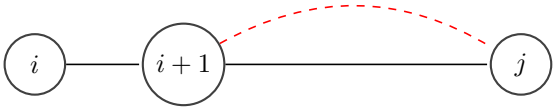
When we need to predict the secondary structure, there are many plausible secondary structures. An RNA of length 200 has over 10^{50} possible base-paired structures. Therefore, we need both a function that assigns the correct structure the highest score, and an algorithm for evaluating the scores.

Nussinov attempts to find the structure with the most base pairs, it is a dynamic programming approach, which calculate the best structure for small subsequences and work outwards. Let's first introduce som notations:

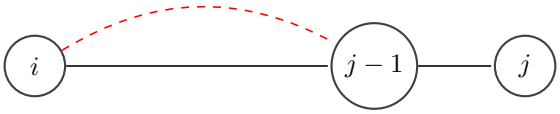
- seq the RNA sequence of $\{A, C, G, U\}$
- $seq[i, j]$ the RNA sequence from position i to j
- str the best 2^{nd} structure for seq of $\{(,), .\}$
- $str[i, j]$ the best 2^{nd} structure for $seq[i, j]$
- $score[i, j]$ the number of base pairs in $str[i, j]$

In the Nussinov algorithm we look at four cases:

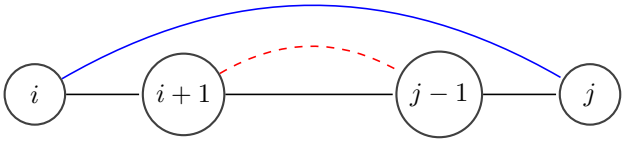
i being unpaired and $str[i+1, j]$, that is we just prepend i to the rest of the structure:



j being unpaired and $str[i, j-1]$, that is we just append j to the rest of the structure:



$seq[i] \cdot seq[j]$ and $str[i+1, j-1]$, that is we add the base-pair i, j to the rest of the structure:



$str[i, k]$ and $str[k+1, j]$ for some $i < k < j$, that is we just concatenate the two structures:

We then find the one of these cases which returns the highest score, which can be described formally as:

$$score[i,j] = \begin{cases} 0 \text{ if } j-1 < 2 \\ \max \begin{cases} score[i+1, j] \\ score[i, j-1] \\ score[i+1, j-1] + 1 \text{ if } seq[i] \cdot seq[j] \\ max_{i<k<j-1}(score[i,k] + score[k+1, j]) \end{cases} \end{cases}$$

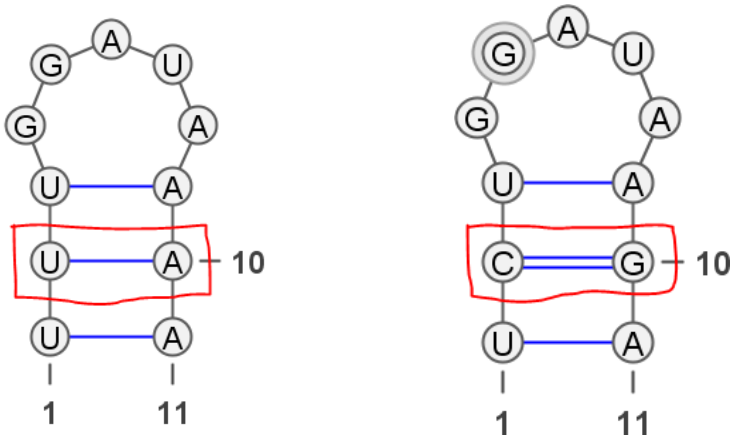
We have to save all the results in table of space $\mathcal{O}\left(n^2\right)$ and it will take $\mathcal{O}\left(n^3\right)$ to compute. We then simply start at the top-right corner of the table produced by the algorithm (the index corresponding to the first and last index) and traceback through the table. The path we trace back through the table is the optimal structure.

This can also be described as a stochastic CFG:

$$\begin{aligned} S &\to aS|cS|gS|uS && (i \text{ unpaired}) \\ S &\to Sa|Sc|Sg|Su && (j \text{ unparied}) \\ S &\to aSu|cSg|gSc|uSa && (\text{i,j pair}) \\ S &\to SS && \text{bifurcation} \end{aligned}$$

2.2.2 RNA evolution

RNA's can have a common 2^{nd} structure without sharing a siginificant sequence similarity. For example, look at the image below a mutation has happened, and in order to maintain the base-pairing complementarity a comensatory mutation has happened at the other end of the base-pair.



In a structurally correct multiple alignment of RNAs, conserved base pairs are often revealed by the presence of frequent correlated compensatory mutations. Intuition: in order to conserve the base pairs, compensatory mutations happen. Therefore if compensatory mutations happen, there must be some base pairs we want to conserve.

Therefore we will measure the pairwise sequence co-variation between two aligned columns i and j by:

$$M_{ij} = \sum_{x_i, x_j} f_{x_i, x_j} \log_2 \frac{f_{x_i, x_j}}{f_{x_i} \cdot f_{x_j}}$$

Where:

- f_{x_i} is the frequency of one of the five possible characters observed in column i
- $_{x_i, x_j}$ is the joint frequency of the pairs observed in columns i and j

For example, say we have the three alignments:

$$seq1 = GUCUGGAC$$
$$seq2 = GACUGGUC$$
$$seq3 = GGCUGGCC$$

Recall that the frequency of an event i is the number n_i of times it occured, and the relative frequency is the number n_i divided by the total number of events N_i.

Then the columns 2, and 7 which represents compensatory mutations can be computed as:

$$M_{2,7} = \sum_{x_i, x_j \in \{seq1, seq2, seq3\}} \frac{1}{3} \log_2 \frac{1/3}{1/3 \cdot 1/3} = \log_2 3 \approx 1.59$$

Therefore, we get the following properties of the mutual information M_{ij}:

- M_{ij} is maximum if i and j appear completely random, but are perfectly correlated
- If i and j are uncorrelated, then the mutual information is 0
- If either i or j are highly conserved positions, then we get little or no mutual information

Think of mutual information, as what we know about j if we know i.

Using this comparative analysis, this is how we would find the secondary structure:

- Start with a multiple alignment
- Predict 2^{nd} structure base on alignment
- Refine alignment based on 2^{nd} structure
- Repeat

In order to compare the sequences they must be:

- Sufficiently similar that they can be initially aligned by primary sequence
- Sufficiently dissimilar that a number of co-varying substitutions can be detected

How to build 2^{nd} structure based on alignment, we can do a greedy method:

- Choose the pair of columns that have the highest M_{ij}
- Make a base pair
- Carry on with the second highest M_{ij}

The problem with this solution is that columns might end up in more than one base pair.

Another solution is to modify the Nussinov algorithm to take alignments into account. We introduce some new notation for the Nussinov algorithm:

- aln the RNA alignment
- aln_k the k^{th} sequence in the alignment
- $aln[i, j]$ the RNA alignment from position i to j

- str the best 2^{nd} structure for aln
- $str[i,j]$ the best 2^{nd} structure for $aln[i,j]$
- $score[i,j]$ the number of base pairs in $str[i,j]$
- $aln[i] \cdot aln[j]$ if for all k, $aln_k[i] \cdot aln_k[j]$

We then increment the score by $1 + M_{ij}$ instead of just by 1 in order to favour base pairs between columns with high mutual information.

2.2.3 Zuker folding algorithm (MFold)

Zuker's algorithm assumes that the correct structure is the one with the lowest *equilibrium free energy* (ΔG). The ΔG of an RNA secondary structure is approximated as the sum of individual contributions from loops, base pairs on other elements. It can then be solved in pretty much the same way as Nussinov, i.e. using a dynamic programming algorithm.

2.2.4 The grammatical approach

As mentioned earlier, we can describe these dynamic programming algorithms as stochastic CFG's (SCFG). SCFG's work like CFG's we simply assign probabilities to each rule. For example we could write:

$$S \to {}^{0.25}a \mid {}^{0.75}b$$

For the grammar that writes an a with 25% probability and b with 75% probability.

If we have such a grammar, for example the one for Nussinov:

$S \to aS \mid cS \mid gS \mid uS$	(i unpaired)
$S \to Sa \mid Sc \mid Sg \mid Su$	(j unparied)
$S \to aSu \mid cSg \mid gSc \mid uSa$	(i,j pair)
$S \to SS$	bifurcation

Then we can convert it to Chomsky Normal Form and use the CYK algorithm to find the most probable structure for a RNA sequence, but it's usually better to use a specialized algorithm for your grammar in order to improve efficiency.

2.2.5 Pseudo-knots are NP-Hard

There are methods to handle pseudo-knots but the problem itself is NP-Hard.