Lecture 17

RNA Secondary Structure Prediction (continued)

March 2, 2000
Notes: Don Patterson

17.1. Recurrence Relations

The core of the dynamic programming algorithm for RNA secondary structure prediction lies in the recurrence relations used to fill the arrays introduced in Section 16.5. This section develops the recurrence relations for W, V, VBI, and VM, which are interdependent.

17.1.1. $W(j)$

$$\begin{aligned} W(0) &= 0 \\ W(j) &= \min(W(j-1), \min_{1 \le i < j}(V(i,j) + W(i-1))), \text{ for } j > 0 \end{aligned}$$

The terms in the second equation correspond to choosing the structure for bases $s_1, s_2, \ldots, s_j$ having the lesser free energy of two possible structures:

- The base s_j does not pair with any other base and is therefore an external base (see Figure 16.1). The recurrence for $W(j)$ makes the implicit assumption that the external bases do not contribute to the overall free energy of the structure. In this case the total energy is therefore $W(j-1)$.
- The base s_j pairs with some other base s_i in $s_1, s_2, \ldots, s_{j-1}$, where i is chosen to minimize the resulting free energy. That energy is the sum of the energy $V(i,j)$ of the compound structure closed by $i \cdot j$, plus the energy $W(i-1)$ of the remainder $s_1, s_2, \ldots, s_{i-1}$.

17.1.2. $V(i,j)$

$$V(i,j) = \begin{cases} +\infty, & \text{for } i \ge j \\ \min(\mathrm{eH}(i,j), \mathrm{eS}(i,j) + V(i+1,j-1), VBI(i,j), VM(i,j)), & \text{for } i < j \end{cases}$$

The terms in the second equation correspond to choosing the minimum free energy structure among the following possible solutions:

- $i \cdot j$ is the exterior pair in a hairpin loop, whose free energy is therefore given by $\mathrm{eH}(i,j)$.

- $i \cdot j$ is the exterior pair of stacked pair. In this case the free energy is the energy $\mathrm{eS}(i,j)$ of the stacked pair, plus the energy $V(i+1, j-1)$ of the compound structure closed by $(i+1) \cdot (j-1)$. We know in this case that $(i+1) \cdot (j-1)$ forms a base pair because $i \cdot j$ is the exterior pair of a stacked pair.
- $i \cdot j$ is the exterior pair of a bulge or internal loop, whose free energy is therefore given by $VBI(i,j)$.
- $i \cdot j$ is the exterior pair of a multibranched loop, whose free energy is therefore given by $VM(i,j)$.

17.1.3. $VBI(i,j)$

$$VBI(i,j) = \min_{\substack{i',j' \\ i<i'<j'<j}} (\mathrm{eL}(i,j,i',j') + V(i',j'))$$

In this case, $i \cdot j$ is the exterior pair of a bulge or interior loop, and we must search all possible interior pairs $i' \cdot j'$ for the pair that results in the minimum free energy. For each such interior pair, the resulting free energy is sum of the energy $\mathrm{eL}(i,j,i',j')$ of the bulge or internal loop, plus the energy $V(i',j')$ of the compound structure closed by $i' \cdot j'$. It is easy to see that this search for the best interior pair is computationally intensive, simply because of the number of possibilities that must be considered. We will see later how to speed up this calculation, which is the new contribution of Lyngsø *et al.* [1].

17.1.4. $VM(i,j)$

$$VM(i,j) = \min_{\substack{k,i_1,j_1,i_2,j_2,\ldots,i_k,j_k \\ i<i_1<j_1<i_2<j_2<\ldots<i_k<j_k<j \\ k \geq 2}} (\mathrm{eM}(i,j,i_1,j_1,i_2,j_2,\ldots,i_k,j_k) + \sum_{h=1}^{k} V(i_h,j_h))$$

In the same way that the recurrence for VBI requires a search for the best structure among all the possible interior pairs, the calculation for VM is even more intensive, requiring a search for k interior pairs $i_h \cdot j_h$, each of which closes its own branch out of the multibranched loop and contributes free energy $V(i_h,j_h)$. A direct implementation of the calculation shown for VM is infeasibly slow. Section 17.3 will discuss simplifying assumptions about multibranched loops that allow us to speed this up substantially.

17.2. Order of Computation

The interdependence of these recurrences requires a careful ordering of the calculations to ensure that we only rely on array entries whose values have already been determined. Specifically, the entries are computed in order from interior pairs to exterior pairs. This corresponds to filling the arrays V, VBI, and VM in order of increasing values of $j - i$. An inspection of the recurrences in Sections 17.1.2 – 17.1.4 reveals that this order will always guarantee that the needed array entries have been computed.

Within the calculations involving a given value $j - i$, we compute $VBI(i,j)$ and $VM(i,j)$ before $V(i,j)$, in order to accommodate the recurrence in Section 17.1.2. Note that the calculations for the three tables are interleaved: we calculate the entry in each table for a given pair i,j before advancing to the next pair.

Because none of these entries depend on the values of entries in W, the computation of W can be deferred until the other three tables have been completed.

17.3. Speeding Up the Multibranched Computation

As mentioned in Section 16.4, the actual free energy values of multibranched loops are not yet well understood. Given this state, the approximation we will describe is driven more by a desire to reduce the running time of the dynamic program than to produce a very accurate physical model of the loop.

For this approximation, we assume that the free energy of a multibranched loop is given by an affine linear function of the number k of branches and the size of the loop (measured as the number of unpaired bases):

$$\mathrm{eM}(i, j, i_1, j_1, \ldots, i_k, j_k) = a + bk + c((i_1 - i - 1) + (j - j_k - 1) + \sum_{h=1}^{k-1}(i_{h+1} - j_h - 1)),$$

where a, b, and c are constants. (Lyngsø *et al.* [1] suggest that it would be more accurate to approximate the free energy as a logarithmic function of the loop size.)

Assuming this linear approximation, we can devise a much more efficient dynamic programming solution for computing VM than the one given in Section 17.1.4. This solution requires an additional array WM, where $WM(i, j)$ gives the free energy of an optimal structure for $s_i, \ldots, s_j$, assuming that s_i and s_j are on a multibranched loop. $WM$is defined by the following recurrence relation:

$$\begin{aligned} WM(i,i) &= c \\ WM(i,j) &= \min(V(i,j) + b, \min_{i<h\leq j}(WM(i,h-1) + WM(h,j))), \text{ for } i < j \end{aligned}$$

The terms in the second equation correspond to the following possible solutions:

- $i \cdot j$ forms a base pair and therefore defines one of the k branches, whose free energy is $V(i, j)$.
- s_i and s_j are not paired with each other, so the free energy is given by the minimum partition of the sequence into two contiguous subsequences.

Calculating VM then reduces to partitioning the loop into at least two pieces with the minimum total free energy:

$$VM(i,j) = \min_{i+1<h\leq j-1}(WM(i+1,h-1) + WM(h,j-1) + a)$$

17.4. Running Time

The running time to fill in each of the complete tables (assuming the values on which it depends have already been computed and stored in their tables, and that we are using the multibranched approximation of Section 17.3) is determined as follows:

- W: $O(n^2)$. Each of n entries requires the computation of the min of $O(n)$ terms.
- V: $O(n^2)$. Each of $O(n^2)$ entries requires the computation of the min of 4 terms.

- VBI: $O(n^4)$. Each of $O(n^2)$ entries requires the computation of the min of $O(n^2)$ terms.
- WM: $O(n^3)$. Each of $O(n^2)$ entries requires the computation of the min of $O(n)$ terms.
- VM: $O(n^3)$. Each of $O(n^2)$ entries requires the computation of the min of $O(n)$ terms.

With the speedup of the multibranched loop computation described in Section 17.3, the new bottleneck has become the $O(n^4)$ time computation of the free energy of bulges and internal loops. We will see next how to eliminate this bottleneck.

References

[1] Rune B. Lyngsø, Michael Zuker, and Christian N. S. Pedersen. Internal loops in RNA secondary structure prediction. In *RECOMB99: Proceedings of the Third Annual International Conference on Computational Molecular Biology*, pages 260–267, Lyon, France, April 1999.