RNA Secondary Structure

RNA can be viewed as a sequence of bases, where each base is either A, C, G, or U.

Let $\mathbf{b} = b_1, \ldots, b_n$

For example:

A C A U G A U G U G C C C A U G C A U

RNA forms secondary structure because bases tend to pair up. Formally, a secondary structure on $\mathbf{b}$ is a set $S$ of pairs of the form $(i, j)$ where $i, j \in \{1, 2, \ldots, n\}$ that satisfies:

(i) Any pair in $S$ is either to $
\{A, U\}$ or $
\{G, C\}$.
(ii) \( S \) is a matching: no bare appears in more than one pair.

(iii) No sharp turns: if \((i, j)\) is a pair in \( S \) and \( i < j \), then \( j > i + 4 \).

(iv) No crossings: if \((i, j)\) and \((k, l)\) are pairs in \( S \), we cannot have \( i < j \) and \( k < l \), we cannot have \( i < k < j < l \).

\[
\begin{align*}
b_i & \cdots b_k \cdots b_j \cdots b_l \\
\underline{\text{A crossing. Not allowed.}}
\end{align*}
\]
Here is a secondary structure on our example:

\[ \text{ACAU} \text{GAUGUC} \text{CCCAUGCAU} \]

Another representation of same structure.

Here is a different secondary structure on same example.

\[ \text{ACAU} \text{GAUGUC} \text{CCCAUGCAU} \]
Problem: Find the secondary structure that maximizes the number of base pairs.
Consider an optimal solution \( O \) for \( b_1, \ldots, b_n \).

If \( n \) is not paired in \( O \),
then \( O \) must be an optimal solution
for \( b_1, \ldots, b_{n-1} \).

Suppose \( (t, n) \) is a pair in \( O \).

\[ b_1, \ldots, b_{t-1}, b_t, b_{t+1}, \ldots, b_n \]

The remaining pairs in \( O \) can be partitioned into an optimal solution
for \( b_1, \ldots, b_{t-1} \), plus an optimal solution
for \( b_{t+1}, \ldots, b_{n-1} \).

Thus, \( O \) consists of an optimal structure
for \( b_1, \ldots, b_{t-1} \), an optimal structure
for \( b_{t+1}, \ldots, b_{n-1} \), plus the pair \( (t, n) \).
Let \( \text{Opt}(i, j) \) denote the number of base pairs in an optimal structure for \( b_i \ldots b_j \).

Of course, \( \text{Opt}(i, j) = 0 \) if \( j \leq i + 4 \).

\[ \text{Opt}(i, n) = \max \left\{ \text{Opt}(1, n-1), \right. \]
\[ \left. \max_{t} \left( 1 + \text{Opt}(1, t-1) \right) \right. \]
\[ \left. + \text{Opt}(t+1, n-1) \right. \]

where \( t \) ranges over indices between 1 and \( n \) such that

(a) \( n > t + 4 \)

(b) \( b_n \) and \( b_t \) are complementary - either \( \{A, U\} \) or \( \{C, G\} \).
In general, if \( j > i + 4 \),

\[
\text{Opt}(i, j) = \max \left\{ \ \text{Opt}(i, j-1), \max_{t} 1 + \text{Opt}(i, t-1) + \text{Opt}(t+1, j-1) \right\}
\]

where \( t \) ranges over indices similar to what is described above.

(Note that \( t = i \) is a special case - treat \( \text{Opt}(i, i+1) \) as 0.)

Here is the algorithm for computing the \( \text{Opt}(i, j) \)s. Have a 2D array \( M[i\ldots n, 0\ldots n] \), an \( n \times (n+1) \) array. \( M[i, i] \) will hold \( \text{Opt}(i, i) \).
Initialize $M[i,j] \leftarrow 0$ for $1 \leq i \leq n$, $0 \leq j \leq n$.

For $K \leq 5, 6, \ldots, n-1$

For $i \leq 1, 2, \ldots, n-K$

Set $j \leftarrow i + K$

best $\leftarrow M[i, j-1]$

For $t \leftarrow i$ to $j-5$ do

if $b_j$ and $b_t$ are complementary

best $\leftarrow \max \{ \text{best}, 1 + M[i, t-1] + M[t+1, j-1] \}$

endif

end for

end for

$M[i, j] \leftarrow \text{best}$

end for

end for