Sequence Alignment with Gap Opening and Extension

In the last lecture, we learned about a dynamic programming algorithm for sequence alignment. This algorithm uses a single fixed gap penalty when aligning amino acids against gaps. In this lecture, we introduce a variant of the dynamic programming algorithm that takes into account the gaps' tendency to cluster together.

Rather than using a single score for all gaps, our algorithm will use two gap scores: the *gap opening* score g_o, and the *gap extension* score g_e. The gap opening score g_o will be used for the first gap in each gap cluster; that is, the gap that immediately follows an amino acid in the aligned sequence. The gap extension score g_e will be used for each gap following the opening gap. Since gaps tend to bundle together, the presence of an opening gap makes it more likely that the next character in the alignment is also a gap. Therefore, g_o should be *higher* than g_e.

$$\begin{array}{lcccccccc} & & & \downarrow g_o & \downarrow g_e & & & & \\ \textbf{A:} & \mathrm{A} & \mathrm{C} & - & - & \mathrm{R} & \mathrm{T} & \mathrm{W} & \mathrm{Y} \\ \textbf{B:} & \mathrm{A} & \mathrm{C} & \mathrm{A} & \mathrm{I} & - & \mathrm{S} & \mathrm{W} & \mathrm{Y} \\ & & & & & \uparrow g_o & & & \end{array}$$

Figure 1: An example of gap opening vs. gap extension scores

How do we implement the two different gap scores into our sequence alignment algorithm? The idea of simply using g_o or g_e instead of the fixed gap score g does not work. Consider, for instance the following alignment.

$$\begin{array}{lcccccc} & & & & & \downarrow & \\ \textbf{A:} & \cdot & \cdot & \cdot & \cdot & \cdot & - \\ \textbf{B:} & \cdot & \cdot & \cdot & \cdot & \cdot & b_m \end{array}$$

Here, the characters prior to b_m have been aligned, and we want to align b_m against a gap. In order to determine whether to use the score g_o or g_e in aligning b_m against a gap in sequence A, we need to know whether the character preceding the gap in A was an amino acid or a gap. However, this information is not stored in the score matrix generated by our dynamic programming algorithm. Therefore, we will need to adopt a slightly modified approach to implement gap extensions and openings.

The key idea in our new approach is that at any point in the alignment process, we can extend our existing alignment in three possible ways: by aligning a_n against b_m, by aligning a_n against a gap, or by aligning b_m against a gap. Therefore, we will have three dynamic score matrices rather than just one. The matrices will be defined as follows:

$T(n,m)$: The matrix of optimal scores so that a_n is aligned against b_m.

$T_-(n,m)$: The matrix of optimal scores so that a_n is aligned against a gap, and b_m precedes that gap.

$T^-(n,m)$: The matrix of optimal scores so that b_m is aligned against a gap, and a_n precedes that gap.

$$
\begin{array}{r|c|l}
T(n,m) & \begin{array}{c}\cdots\cdots \\ \cdots\cdots\end{array} & \begin{array}{l}a_n \\ b_m\end{array} \\
T_-(n,m) & \begin{array}{c}\cdots\cdots \\ \cdots b_m\end{array} & \begin{array}{l}a_n \\ -\end{array} \\
T^-(n,m) & \begin{array}{c}\cdots a_n \\ \cdots\cdots\end{array} & \begin{array}{l}- \\ b_m\end{array}
\end{array}
$$

Figure 2: The three dynamic score matrices

In figure 3, we illustrate how to set up the recurrence relations for use with our new dynamic matrices. Each equation is shown, along with the picture of the alignment corresponding to that equation. These recurrence relations follow naturally from the definitions of our dynamic matrices. The initialization for each of the matrices is identical to the initialization of the dynamic matrix for sequence alignment with a single fixed gap penalty.

$$T(n+1,m+1)=\max\begin{cases} T(n,m)+S(a_{n+1},b_{m+1}) & \begin{array}{rcc}\hline \cdots & a_n & a_{n+1}\\ \cdots & b_m & b_{m+1}\\ \hline \end{array}\\ T_{-}(n,m)+S(a_{n+1},b_{m+1}) & \begin{array}{rcc} \cdots\cdots & a_n & a_{n+1}\\ \cdots b_m & - & b_{m+1}\\ \hline \end{array}\\ T^{-}(n,m)+S(a_{n+1},b_{m+1}) & \begin{array}{rcc} \cdots a_n & - & a_{n+1}\\ \cdots\cdots & b_m & b_{m+1}\\ \hline \end{array} \end{cases}$$

$$T_{-}(n+1,m+1)=\max\begin{cases} T(n,m+1)+g_o & \begin{array}{rcc}\hline \cdots & a_n & a_{n+1}\\ \cdots & b_{m+1} & -\\ \hline \end{array}\\ T_{-}(n,m+1)+g_e & \begin{array}{rcc} \cdots\cdots\cdots & a_n & a_{n+1}\\ \cdots b_{m+1} & - & -\\ \hline \end{array}\\ T^{-}(n,m+1)+g_o \quad (*) & \begin{array}{rcc} \cdots a_n & - & a_{n+1}\\ \cdots\cdots & b_{m+1} & -\\ \hline \end{array} \end{cases}$$

$$T^{-}(n+1,m+1)=\max\begin{cases} T(n+1,m)+g_o & \begin{array}{rcc}\hline \cdots & a_{n+1} & -\\ \cdots & b_m & b_{m+1}\\ \hline \end{array}\\ T_{-}(n+1,m)+g_o \quad (*) & \begin{array}{rcc} \cdots\cdots & a_{n+1} & -\\ \cdots b_m & - & b_{m+1}\\ \hline \end{array}\\ T^{-}(n+1,m)+g_e & \begin{array}{rcc} \cdots a_{n+1} & - & -\\ \cdots\cdots\cdots & b_m & b_{m+1}\\ \hline \end{array} \end{cases}$$

Figure 3: Recurrence relations for the dynamic score matrices

Note that in the equations marked with (*), the alignment segments $\binom{-\ \ a_{n+1}}{b_{m+1}\ \ -}$ and $\binom{a_{n+1}\ \ -}{-\ \ b_{m+1}}$ both contribute a score of $2g_o$, whereas aligning $\binom{a_{n+1}}{b_{m+1}}$ would contribute the score $S(a_n, b_m)$. Any reasonable value of g_o will be quite small, so that for any pair of amino acids, $S(a_n, b_m) > 2g_o$. Therefore, both equations marked by (*) are redundant, and the resulting recurrence relations are:

$$T(n+1,m+1) \;=\; \max\begin{cases} T(n,m) & + \quad S(a_{n+1},b_{m+1})\\ T_{-}(n,m) & + \quad S(a_{n+1},b_{m+1})\\ T^{-}(n,m) & + \quad S(a_{n+1},b_{m+1}) \end{cases} \tag{1}$$

$$T_{-}(n+1,m+1) = \max \begin{cases} T(n,m+1) + g_o \\ T_{-}(n,m+1) + g_e \end{cases} \tag{2}$$

$$T^{-}(n+1,m+1) = \max \begin{cases} T(n+1,m) + g_o \\ T^{-}(n+1,m) + g_e \end{cases} \tag{3}$$

Finally, to recover the optimal alignment, we would need to backtrack through our scoring matrices. To do that, we must first determine which of the three matrices has the maximum value at position (N, M), where N is the length of sequence A and M is the length of sequence B. Afterwards, we proceed according to the recurrence relation, identically to the single fixed gap sequence alignment algorithm. In the figure below, for example, we start off at $T(M, N)$ and backtrack to $T^{-}(M-1, N-1)$. This corresponds to the last two characters in the alignment being $\begin{pmatrix} - & A_N \\ B_{M-1} & B_M \end{pmatrix}$.

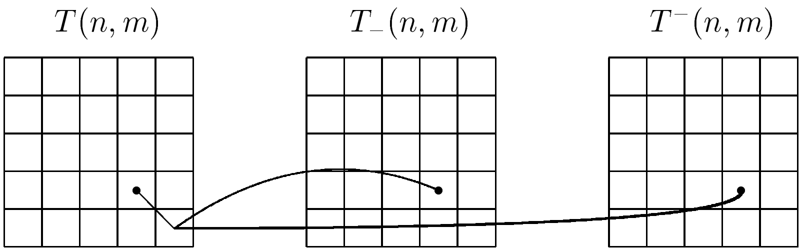

Figure 4: Back-tracking through the matrices