

Dynamic programming for global alignment of amino acid sequences (Simplified Needleman-Wunsch algorithm)

Procedure

Start in upper left corner. Fill in the table by utilizing a series of “moves”. Allowed moves into a given cell are from above, from the left, or diagonally from the upper-left.

Consider all possible moves into a cell. Calculate the resulting score as the score in the originating cell together with all points incurred due to the resulting matches, mismatches, or gaps that have been inserted. Write the best score in the new cell.

A move in the horizontal or vertical direction implies an added gap. Moves diagonally do not involve the addition of a gap.

Complete the entire table in this manner.

Determine the optimum alignments by finding the largest score along the last column or bottom row and tracing back through the matrix to decide how that score was arrived at. The trace-back process thus defines the optimal aligned sequence.

Simplifications that were applied in this example:

Multiple gaps – actually the possible moves into a cell from the above or left direction (not the diagonal) should allow for multiple gap insertions by picking the maximum score based on a move from the immediately adjacent cell (either directly above or to the left of the cell being calculated) and also the cells further away along that vertical or horizontal direction. Such moves will create multiple gap penalties, but might ultimately lead to an alignment further along in the matrix. Note that this simplification applies to inserting multiple gaps in the course of filling in one cell – even with this simplification, multiple gaps may arise in the final aligned sequence if an alignment path is followed horizontally or vertically over several cells.

Scoring matrix – this example used a simplified arrangement for assigning points to matches and mismatches. In actual use, these algorithms would assign different points to matches or mismatches depending upon the specific amino acids that become aligned. Thus the alignment of two cysteine amino acids might be worth +9 points whereas two leucine amino acids might only be worth 4 points. Also the mismatch of C with E might be a –4 point score, but a C and A mismatch only 0 points (no penalty at all).

Simplified scheme used for scoring in this example:

matches: 1 point
 mismatches: -1 point
 gaps: -1 point

	gaps	V	D	S	C	Y
gaps	0	-2	-4	-6	-8	-10
V	-2	1 -2 -3 1	-2 -3 -5 -1	-3 -5 -7 -3	-5 -7 -9 -5	-7 -9 -12 -7
E	-4	-3 -6 -6 -1	-1 -3 -2 0	-2 -4 -5 -2	-4 -6 -5 -4	-6 -9 -8 -6
S	-6	-5 -8 -7 -3	-3 -5 -2 -2	-2 -6 -3 1	-1 -3 -2 -1	-3 -5 -3 -3
L	-8	-7 -10 -6 -5	-4 -7 -3 -4	-3 -6 -1 -1	0 -3 -3 0	-2 -5 -2 -2
C	-10	-9 -12 -9 -7	-6 -9 -6 -6	-5 -8 -3 -3	0 -3 -4 0	-1 -2 -1 -1
Y	-12	-11 -14 -8 -9	-8 -11 -8 -8	-7 -10 -5 -5	-4 -7 -2 -2	1 -2 -1 1

Note: The column and row labeled “gaps” are present to allow for gaps to be inserted at the beginning of either sequence. The three tiny numbers in each cell show the score for each cell that would arise based upon each of the three possible moves.

Note in the previous figure how there are two trace back pathways and only one leads to the expected sequence alignment. Although this can occur, repeating the process using a less simplified scoring matrix (like BLOSUM62) would yield a single sequence as shown below.

	gaps	V	D	S	C	Y
gaps	0	-11	-22	-33	-44	-55
V	-11	4	-10	-23	-35	-47
E	-22	-9	6	-17	-32	-45
S	-33	-22	-5	10	-2	-15
L	-44	-35	-20	-3	9	-3
C	-55	-47	-34	-15	7	-7
Y	-66	-59	-48	-28	-6	14

Resulting alignment:

```
VDS-CY
VESLCY
```