

Dynamic programming
algorithms for
pairwise alignment
and
maximum parsimony

Edit distance

- Given indel cost 1, and substitution cost C , and two sequences X and Y , find the minimum cost of any edit transformation of X into Y .
- How to solve this using Dynamic Programming?
- $M[i,j]$ is the minimum cost of any edit transformation of $X[1\dots i]$ into $Y[1\dots j]$

DP solution for edit distance

- $M[i,j]$ is the minimum cost of any edit transformation of $X[1\dots i]$ into $Y[1\dots j]$
- How to initialize?
- How to set each entry using previously computed entries?
- How to order the calculations?
- Where is the answer?

The DP solution

- Input strings: $X[1\dots n]$ and $Y[1\dots m]$.
- Indel cost 1, substitution cost $C > 0$
- $M[i,j]$ is the edit distance of the prefix x_1, x_2, \dots, x_i to the prefix y_1, y_2, \dots, y_j . We need to compute $M[i,j]$ for $i=0, 1, \dots, n$, and $j=0, 1, \dots, m$.
- $M[0,j]=M[j,0]=j$ for all j (why?)
- The solution is stored in $M[n,m]$ (why?).
- How do we compute $M[i,j]$ for the other values of i and j ?

The DP solution

- Indel cost 1, substitution cost $C > 0$
- $M[i,j]$ is the edit distance of the prefix x_1, x_2, \dots, x_i to the prefix y_1, y_2, \dots, y_j . How do we compute $M[i,j]$ for the other values of i and j ?
- If $x_i = y_j$ then $M[i,j] =$
$$\min\{M[i-1,j-1], M[i-1,j]+1, M[i,j-1]+1\}$$

Else $M[i,j] =$
$$\min\{M[i-1,j-1] + C, M[i-1,j]+1, M[i,j-1]+1\}$$

The DP solution

- Indel cost 1, substitution cost $C > 0$
- If $x_i = y_j$ then $M[i,j] =$
 $\min\{M[i-1,j-1], M[i-1,j]+1, M[i,j-1]+1\}$

else $M[i,j] =$

$$\min\{M[i-1,j-1] + C, M[i-1,j]+1, M[i,j-1]+1\}$$

We compute the entries of the matrix M row-by-row (or column-by-column).

The edit distance is stored in $M[n,m]$.

If we add arrows (from each box to the box(es) which gave the lowest edit distance, we can obtain the minimum cost transformation.

Maximum parsimony

- Fixed tree problem: given a tree T and sequences at the leaves, compute the “length” of the tree under an optimal assignment of sequences to the internal nodes.
- The “length” is the sum of the Hamming distances on the edges of the tree.

MP on a fixed tree (cont.)

- We solve the problem for sequences of length 1!
- Let $\text{Cost}(v,L)$ be the minimum cost of the tree rooted at v given that we label v by the letter L (so $L = A, C, T$ or G).
- What should $\text{Cost}(v,L)$ be if v is a leaf (and so already labelled)? (Infinity if L is the wrong label, and otherwise 0.)

MP on a fixed tree (cont.)

- Suppose v has two children, w and x . Then

$\text{Cost}(v,L) =$

$$\min_P\{\text{Cost}(w,L), \text{Cost}(w,P)+1 \text{ if } P \neq L\} +$$
$$\min_P\{\text{Cost}(x,L), \text{Cost}(x,P)+1 \text{ if } P \neq L\}$$

MP on a fixed tree (cont.)

- Compute $\text{Cost}(v,L)$ for every node v and every nucleotide L , as you go from the leaves to the root.
- $\min\{\text{Cost}(\text{root},A),\text{Cost}(\text{root},C), \text{Cost}(\text{root},T), \text{Cost}(\text{root},G)\}$ is the minimum cost achievable (i.e., the “length” of the tree for that site).
- Backtrack to get the actual assignment to internal nodes.