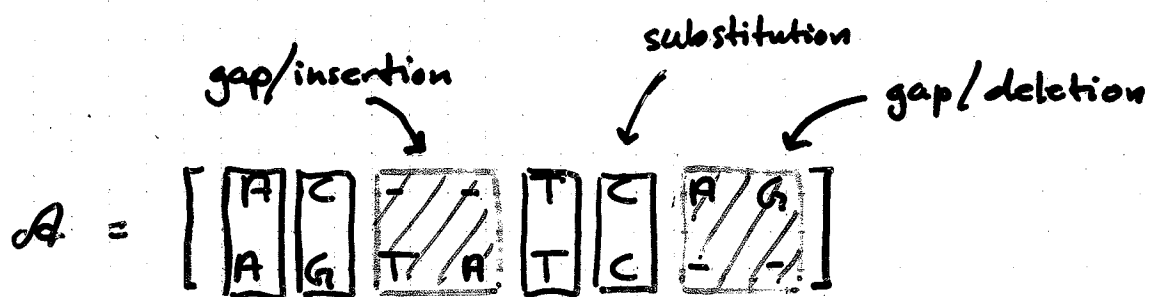


Global alignment of two strings



General cost:

$$\text{cost}(A) = \sum_{\text{blocks}} \text{"cost of block"}$$

Substitution cost: $s: \Sigma \times \Sigma \rightarrow \mathbb{R}$

gap cost: $g: \mathbb{N}_+ \rightarrow \mathbb{R}$

Problem

$$\text{Sim}(A, B) = \max_A \text{cost}(A)$$

$$A_{\text{opt}} = \underset{A}{\text{argmax}} \text{cost}(A)$$

Computing optimal cost with affine gap cost

$$g(k) = \alpha \cdot k + \beta, \quad \alpha, \beta \geq 0$$

Intuition

... as general gap cost but reduce time to compute $D(i, j)$ and $I(i, j)$. Let us consider $D(i, j)$...

Trick

Consider the best deletion-block, two possibilities:

① A new block

$$\begin{array}{cc} \overbrace{M(i-1, j) - (\alpha + \beta)} & \overbrace{I(i-1, j) - (\alpha + \beta)} \\ \left[\begin{array}{c} \sim A[i-1] \quad A[i] \\ \sim B[j] \quad - \end{array} \right] & \left[\begin{array}{c} \sim \quad - \quad A[i] \\ \sim \quad B[j] \quad - \end{array} \right] \end{array}$$

② Continuation of existing block

$$\overbrace{D(i-1, j) - \alpha}$$
$$\left[\begin{array}{c} \sim A[i-k] \quad \dots \quad A[i-1] \quad A[i] \\ \sim B[j] \quad - \quad - \quad - \end{array} \right]$$

Hence,

$$D(i, j) = \max \{ M(i-1, j) - (\alpha + \beta), I(i-1, j) - (\alpha + \beta), D(i-1, j) - \alpha \}$$

$$= \max \{ S(i-1, j) - (\alpha + \beta), D(i-1, j) - \alpha \}$$

Computing optimal cost with affine gap cost

$$S(i, j) = \max \begin{cases} 0 & i=0, j=0 \\ S(i-1, j-1) + s(A[i], B[j]) & i>0, j>0 \\ D(i, j) & i>0, j=0 \\ I(i, j) & i=0, j>0 \end{cases}$$

$$D(i, j) = \max \begin{cases} S(i-1, j) - (\alpha + \beta) & i>0, j>0 \\ D(i-1, j) - \alpha & i>0, j=0 \end{cases}$$

$$I(i, j) = \max \begin{cases} S(i, j-1) - (\alpha + \beta) & i>0, j>0 \\ I(i, j-1) - \alpha & i=0, j>0 \end{cases}$$

Can be implemented using dyn. prog./memorization
using 3 tables of size $\Theta(n \cdot m)$

Time:

Space:

What about retrieving an optimal alignment?