Pairwise Alignment

- Given two strings S and T over some alphabet. Length of S is $|S|$ and that of T is $|T|$.

  Define $a(i, j)$ to be the value of an **optimal alignment** of strings:

  $S[1], S[2], \ldots, S[i]$ and

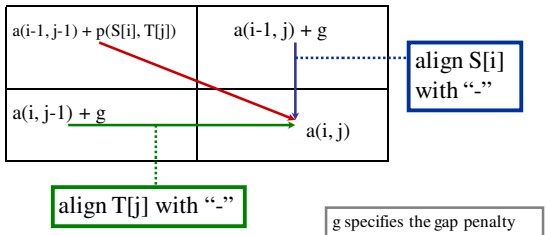
  $T[1], T[2], \ldots\ldots, T[j]$

  $a(|S|, |T|)$ is the value of an optimal alignment of S and T.

Filling in the DP Table

- DP uses a table of size (|S|+1) × (|T|+1).
- $a(i, j)$ corresponds to the optimal alignment of the i^{th} prefix of S with the j^{th} prefix of T.
- The dynamic programming algorithm fills in the entries of the table (matrix) by computing the values of $a(i, j)$ from top to bottom, left to right.
- The value of the optimal alignment is given by $a(|S|,|T|)$.

Filling Entry a(i,j) in the Table



g specifies the gap penalty
p(S[i], T[j]) = 1 if S[i] = T[j]
p(S[i], T[j]) = 0 otherwise

DP: Bookkeeping and Retracing

- Draw lines crossing the entries in the matrix to show from which entry in the matrix we derived the maximum score for each entry a(i, j).
- To determine the solution of the optimal alignment, simply retrace the steps from entry $a(|S|, |T|)$ to entry a(0, 0).

DP for Pairwise Alignment

Algorithm *Similarity*
input: sequences s and t
output: similarity between s and t
$m \leftarrow |s|$
$n \leftarrow |t|$
for $i \leftarrow 0$ **to** m **do**
  $a[i, 0] \leftarrow i \times g$
for $j \leftarrow 0$ **to** n **do**
  $a[0, j] \leftarrow j \times g$
for $i \leftarrow 1$ **to** m **do**
  for $j \leftarrow 1$ **to** n **do**
    $a[i, j] \leftarrow \max(a[i-1, j] + g,$
    $a[i-1, j-1] + p(i, j),$
    $a[i, j-1] + g)$
return $a[m, n]$

Algorithm for filling in the DP table row by row, from top to bottom, left to right. g specifies the gap penalty.

Introduction to Computational Molecular Biology by Setubal et al.

Pairwise Alignment: Traceback

Algorithm *Align*
input: indices i, j, array a given by algorithm *Similarity*
output: alignment in *align-s*, *align-t*, and length in *len*
if $i = 0$ **and** $j = 0$ **then**
  $len \leftarrow 0$
else if $i > 0$ **and** $a[i, j] = a[i-1, j] + g$ **then**
  Align$(i-1, j, len)$
  $len \leftarrow len + 1$
  align-s$[len] \leftarrow s[i]$
  align-t$[len] \leftarrow$ -
else if $i > 0$ **and** $j > 0$ **and** $a[i, j] = a[i-1, j-1] + p(i, j)$ **then**
  Align$(i-1, j-1, len)$
  $len \leftarrow len + 1$
  align-s$[len] \leftarrow s[i]$
  align-t$[len] \leftarrow t[j]$
else // has to be $j > 0$ and $a[i, j] = a[i, j-1] + g$
  Align$(i, j-1, len)$
  $len \leftarrow len + 1$
  align-s$[len] \leftarrow$ -
  align-t$[len] \leftarrow t[j]$

Introduction to Computational Molecular Biology by Setubal et al.