Sequence alignment

The Needleman-Wunsch algorithm

Outline

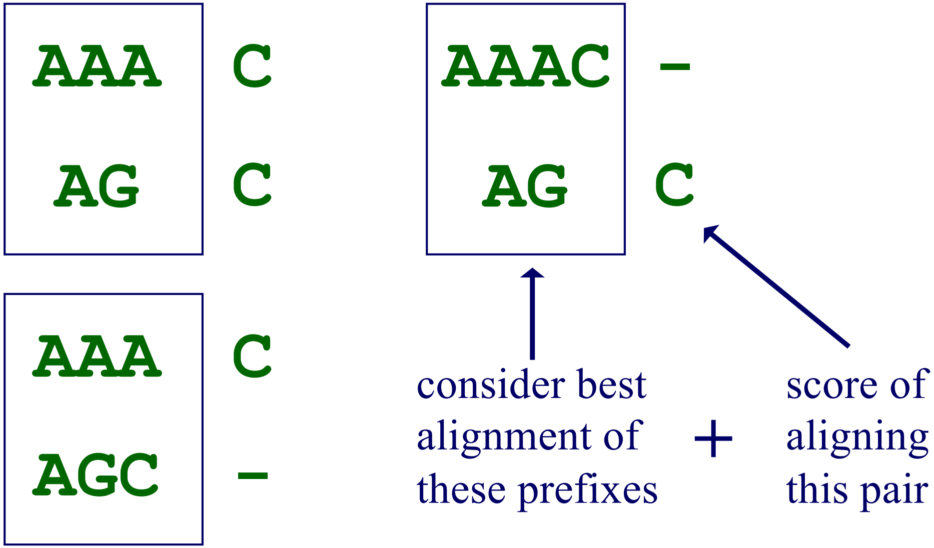
- The Needleman-Wunsch (NW) algorithm
  - Solves *global* pairwise alignment task
- Example run of the algorithm
- Computational complexity of NW

Global Pairwise Alignment Via Dynamic Programming

- first algorithm by Needleman & Wunsch, *Journal of Molecular Biology*, 1970
- *dynamic programming algorithm:* determine best global alignment of two sequences by determining best alignment of all prefixes of the sequences

Dynamic Programming Idea

- consider the last column of the optimal alignment of **AAAC** with **AGC**
- three possible options; in each we'll choose a different pairing for end of alignment, and add this to the best alignment of previous characters



AAA C
AG C

AAAC –
AG C

AAA C
AGC –

consider best alignment of these prefixes + score of aligning this pair

DP Algorithm for Global Alignment with Linear Gap Penalty

- Subproblem: *F(i,j)* = score of best alignment of the length *i* prefix of *x* and the length *j* prefix of *y*.

Main recurrence:

$$F(i,j) = \max \begin{cases} F(i-1,j-1) + S(x_i, y_j) \\ F(i-1,j) + s \\ F(i,j-1) + s \end{cases}$$

Base cases

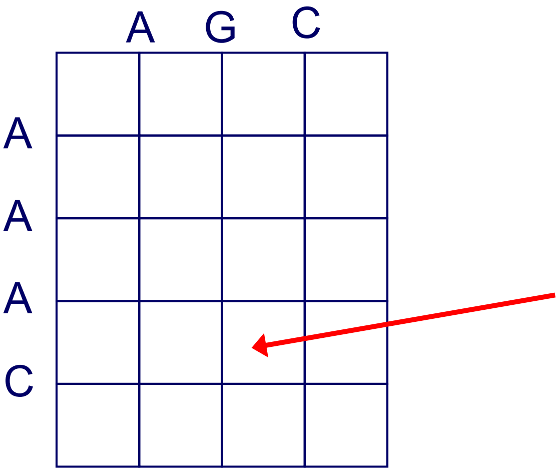
$F(0,0) = 0$ Alignment of two empty strings

$F(i, 0) = i \times s$ Alignment of length i string to empty string

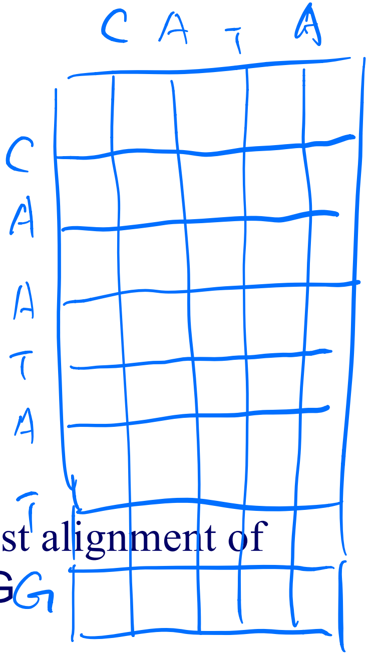
$F(0, j) = j \times s$ Alignment of length j string to empty string

Dynamic Programming Implementation

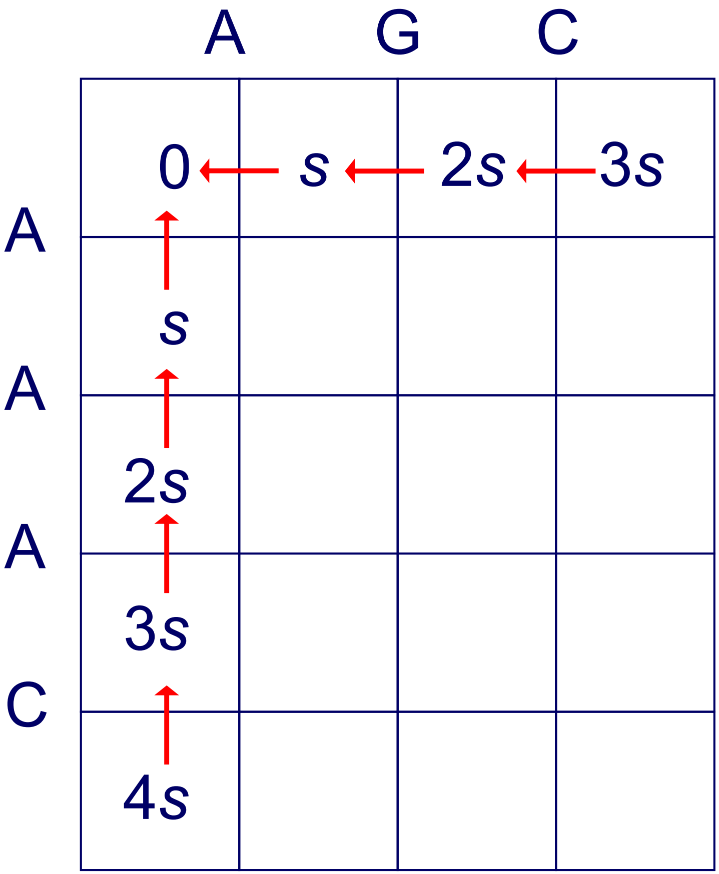
- given an m-character sequence x, and an n-character sequence y
- construct an $(m+1) \times (n+1)$ matrix F
- $F(i, j)$ = score of the best alignment of $x[1 \ldots i]$ with $y[1 \ldots j]$



score of best alignment of AAA to AG



Initializing Matrix: Global Alignment with Linear Gap Penalty



		A	G	C
	0	← *s*	← 2*s*	← 3*s*
A	↑ *s*			
A	↑ 2*s*			
A	↑ 3*s*			
C	↑ 4*s*			

DP Algorithm Sketch: Global Alignment

- initialize first row and column of matrix
- fill in rest of matrix from top to bottom, left to right
- for each $F(i, j)$, save pointer(s) to cell(s) that resulted in best score
- $F(m, n)$ holds the optimal alignment score; trace pointers back from $F(m, n)$ to $F(0, 0)$ to recover alignment

Global Alignment Example

- suppose we choose the following scoring scheme:

$$S(x_i, y_i) =$$

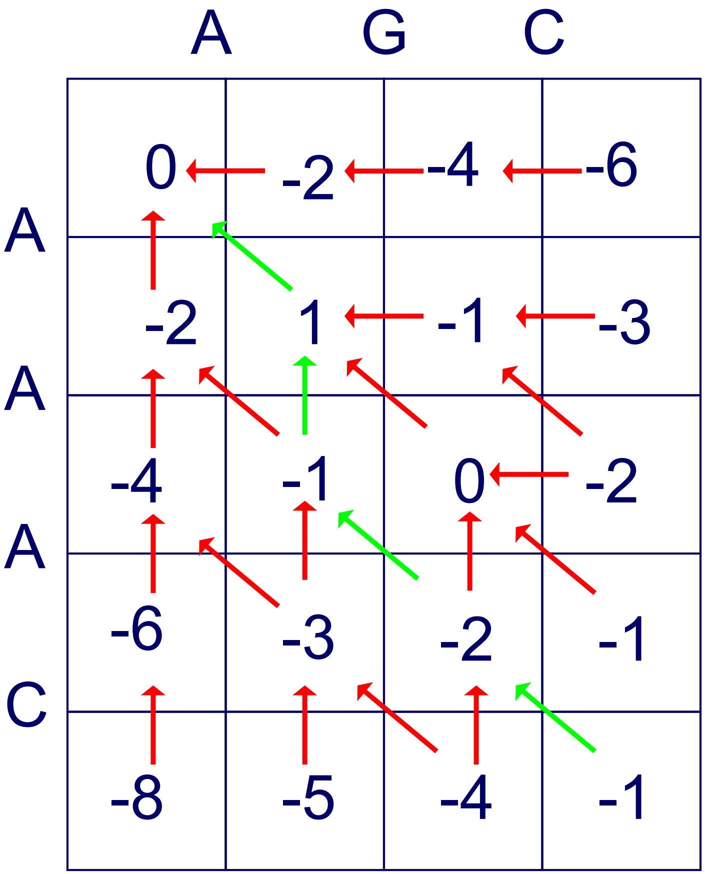
$$\begin{array}{ll} +1 & \text{when } x_i = y_i \\ -1 & \text{when } x_i \neq y_i \end{array}$$

s (penalty for aligning with a space) = -2

Global Alignment Example

		A	G	C
A				
A				
A				
C				

Global Alignment Example



		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

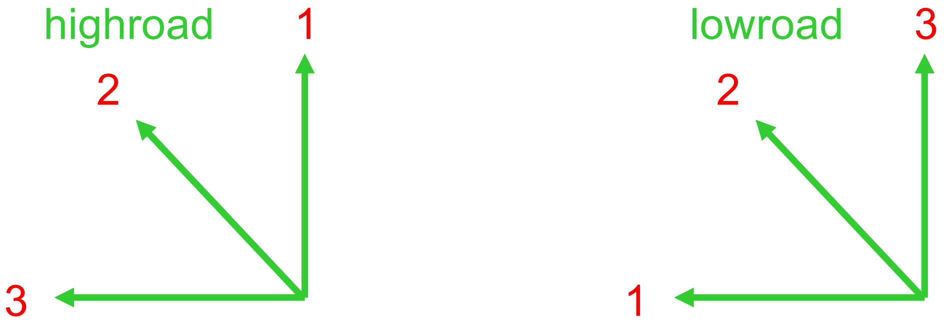
one optimal alignment

x: A A A C
y: A - G C

but there are three optimal alignments here (can you find them?)

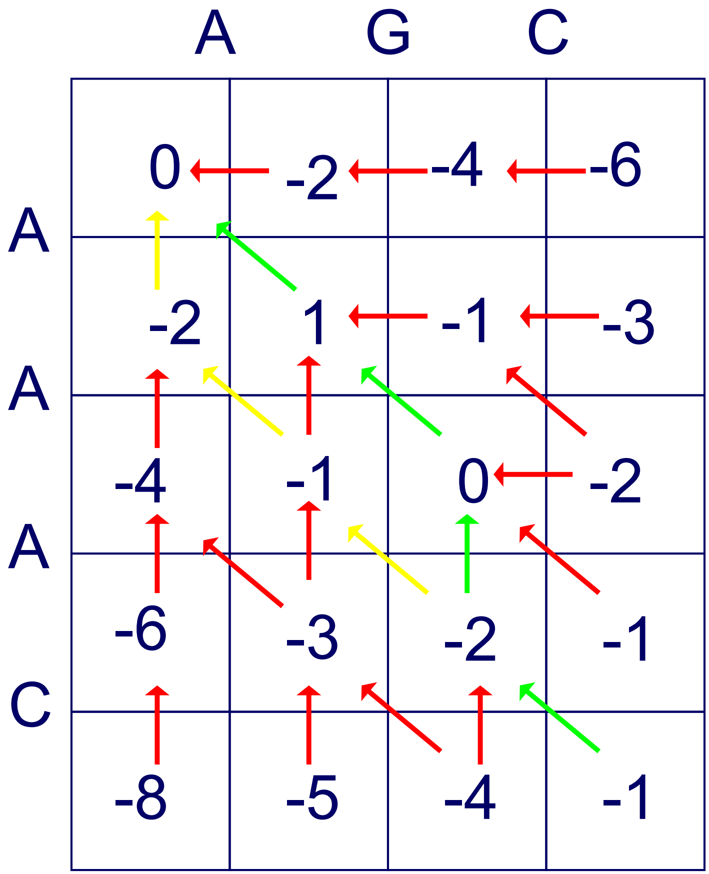
Equally Optimal Alignments

- many optimal alignments may exist for a given pair of sequences
- can use preference ordering over paths when doing traceback



- *highroad* and *lowroad* alignments show the two most different optimal alignments

Highroad & Lowroad Alignments



		A	G	C
	0	-2	-4	-6
A	-2	1	-1	-3
A	-4	-1	0	-2
A	-6	-3	-2	-1
C	-8	-5	-4	-1

highroad alignment

x:	A	A	A	C
y:	A	G	-	C

lowroad alignment

x:	A	A	A	C
y:	-	A	G	C

Computational Complexity

- initialization: $O(m)$, $O(n)$ where sequence lengths are m, n
- filling in rest of matrix: $O(mn)$
- traceback: $O(m + n)$
- hence, if sequences have nearly same length, the computational complexity is

$$O(n^2)$$

Dynamic Programming Analysis

- recall, there are

$$\binom{2n}{n} = \frac{(2n)!}{(n!)^2} \approx \frac{2^{2n}}{\sqrt{\pi n}}$$

possible global alignments for 2 sequences of length *n*

- but the DP approach finds an optimal alignment efficiently

DP Comments

- works for either DNA or protein sequences, although the substitution matrices used differ
- finds an optimal alignment
- the exact algorithm (and computational complexity) depends on gap penalty function (we’ll come back to this issue)

Summary

- Needleman-Wunsch algorithm is a dynamic programming algorithm for solving the global alignment task
- *Key point:* Needleman-Wunsch breaks the problem into a function of the solutions to three subproblems.
- Needleman-Wunsch is a $O(n^2)$ algorithm even thought the space of alignments is exponential in size.