

Sequence alignment

The Smith–Waterman algorithm

Outline

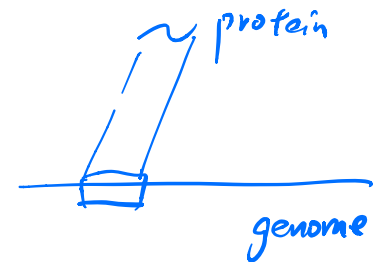
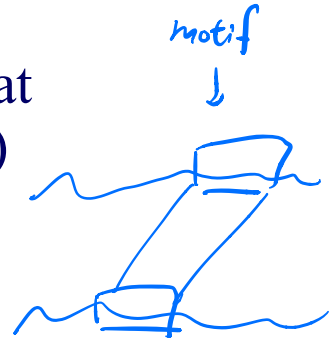
- The Smith–Waterman (SW) algorithm
 - Solves *local* pairwise alignment task
- Example run of the algorithm

Local Alignment

- so far we have discussed *global alignment*, where we are looking for the best matching between sequences from one end to the other
- sometimes, we will only want a *local alignment*, the best match between contiguous subsequences (substrings) of x and y

Local Alignment Motivation

- useful for comparing protein sequences that share a common *motif* (conserved pattern) or *domain* (independently folded unit) but differ elsewhere
- useful for comparing DNA sequences that share a similar *motif* but differ elsewhere
- useful for comparing protein sequences against *genomic DNA sequences* (long stretches of uncharacterized sequence)
- more sensitive when comparing highly diverged sequences




Local Alignment DP Algorithm

- original formulation: Smith & Waterman, *Journal of Molecular Biology*, 1981
- interpretation of array values is somewhat different
 - $F(i, j)$ = score of the best alignment of a suffix of $x[1 \dots i]$ and a suffix of $y[1 \dots j]$

Local Alignment DP Algorithm

- the recurrence relation is slightly different than for global algorithm

$$F(i, j) = \max \begin{cases} F(i-1, j-1) + S(x_i, y_j) \\ F(i-1, j) + s \\ F(i, j-1) + s \\ 0 \end{cases}$$


Local Alignment DP Algorithm

- initialization: first row and first column initialized with 0's
- traceback:
 - find maximum value of $F(i, j)$; can be anywhere in matrix
 - stop when we get to a cell with value 0

Local Alignment Example

		A	A	G	A	
T	0	0	0	0	0	0
T	0	0	0	0	0	0
A	0	0	0	0	0	0
A	0	←	←	←	←	1
A	0	←	←	←	←	1
G	0	1	2	0	1	
	0					

Match: +1

Mismatch: -1

Space: -2

Local Alignment Example

		A	A	G	A	
T	0	0	0	0	0	
T	0	0	0	0	0	
A	0	0	0	0	0	
A	0	1	1	0	1	
G	0	1	2	0	1	
	0	0	0	3	1	

x: A A G
 y: A A G

Match: +1

Mismatch: -1

Space: -2

Summary

- The Smith–Waterman algorithm is a dynamic programming algorithm for solving the *local* alignment task
- It is very similar to Needleman–Wunsch, with three major differences
 - Additional case in maximization recurrence
 - Initialization
 - Traceback