



Sequence Alignment

AGGCTATCACCTGACCTCCAGGCCGATGCCC
TAGCTATCACGACCGCGGTCGATTGCCCCGAC

–AGGCTATCACCTGACCTCCAGGCCGA–TGCCC––
TAG–CTATCAC–GACCGC–GGTCGATTGCCCCGAC



Why sequence alignment?

- Comparing genes or regions from different species
 - to find important regions
 - determine function
 - uncover evolutionary forces
- Assembling fragments to sequence DNA
- Compare individuals to looking for mutations



Alignments in two fields

- In Natural Language Processing
 - We generally talk about **distance** (minimized)
 - And **weights**
- In Computational Biology
 - We generally talk about **similarity** (maximized)
 - And **scores**



The Needleman-Wunsch Algorithm

- Initialization:

$$D(i, 0) = -i * d$$

$$D(0, j) = -j * d$$

- Recurrence Relation:

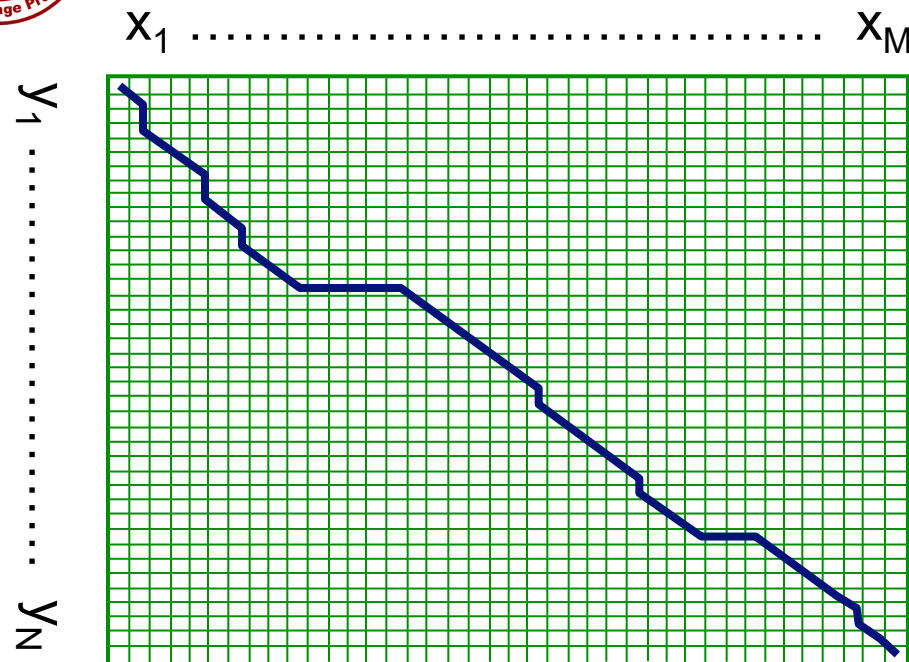
$$D(i, j) = \max \begin{cases} D(i-1, j) & - d \\ D(i, j-1) & - d \\ D(i-1, j-1) & + s[x(i), y(j)] \end{cases}$$

- Termination:

$D(N, M)$ is distance



The Needleman-Wunsch Matrix



(Note that the origin is at the upper left.)



A variant of the basic algorithm:

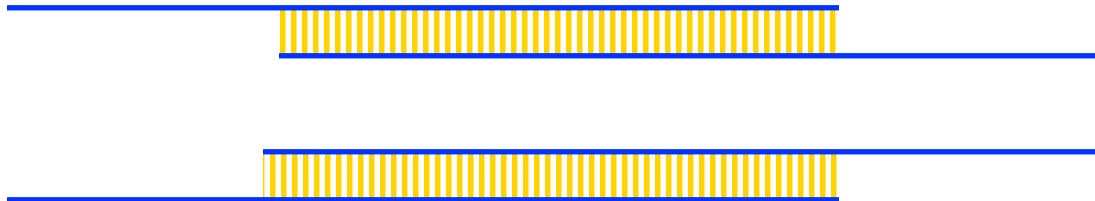
- Maybe it is OK to have an unlimited # of gaps in the beginning and end:

-----CTATCACCTGACCTCCAGGCCGATGCCCCCTTCCGGC
GCGAGTTCATCTATCAC--GACCGC--GGTCG-----

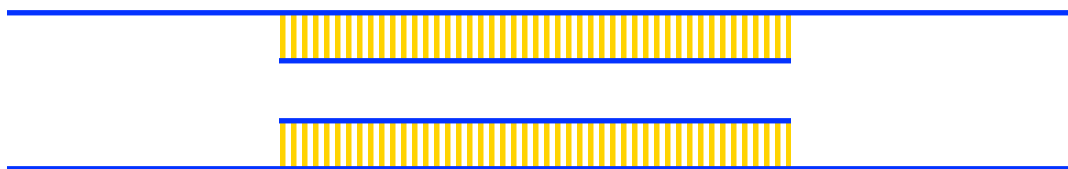
- If so, we don't want to penalize gaps at the ends



Different types of overlaps



Example:
2 overlapping “*reads*” from a sequencing project



Example:
Search for a mouse gene within a human chromosome



The Overlap Detection variant



Changes:

1. Initialization

For all i, j ,

$$F(i, 0) = 0$$

$$F(0, j) = 0$$

2. Termination

$$F_{\text{OPT}} = \max \left\{ \begin{array}{l} \max_i F(i, N) \\ \max_j F(M, j) \end{array} \right.$$

Slide from Serafim Batzoglou



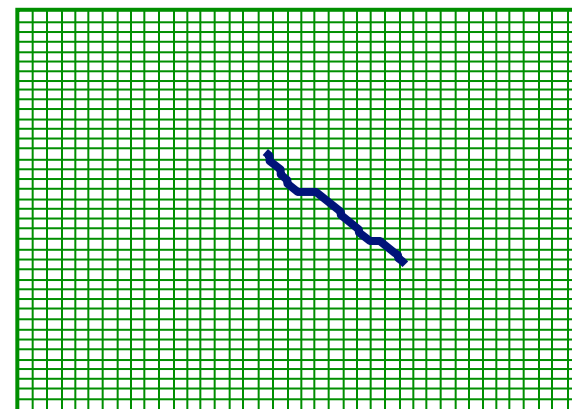
The Local Alignment Problem

Given two strings

$$x = x_1 \dots x_M,$$

$$y = y_1 \dots y_N$$

Find substrings x' , y' whose similarity
(optimal global alignment value)
is maximum



$x = \text{aaaacc}\text{cccggg}\text{gtta}$

$y = \text{ttcccggg}\text{aaccaacc}$



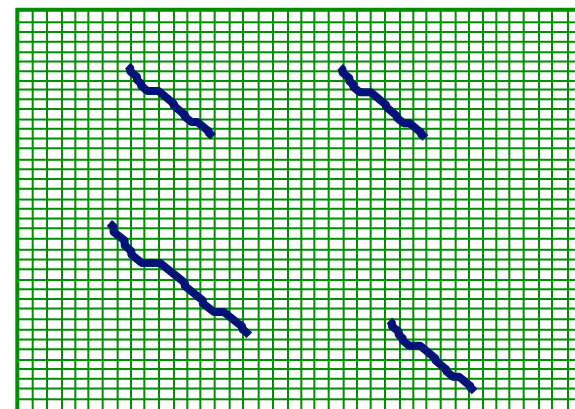
The Smith-Waterman algorithm

Idea: Ignore badly aligning regions

Modifications to Needleman-Wunsch:

Initialization: $F(0, j) = 0$
 $F(i, 0) = 0$

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





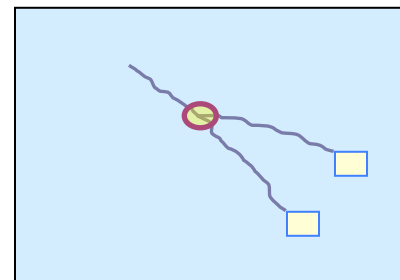
The Smith-Waterman algorithm

Termination:

1. If we want the **best** local alignment...

$$F_{\text{OPT}} = \max_{i,j} F(i, j)$$

Find F_{OPT} and trace back



2. If we want **all** local alignments **scoring** $> t$

?? For all i, j find $F(i, j) > t$, and trace back?

Complicated by overlapping local alignments

Slide from Serafim Batzoglou



Local alignment example

X = ATCAT

Y = ATTATC

Let:

$m = 1$ (1 point for match)

$d = 1$ (-1 point for del/ins/sub)

		A	T	T	A	T	C
	0	0	0	0	0	0	0
A	0						
T	0						
C	0						
A	0						
T	0						



Local alignment example

X = ATCAT

Y = ATTATC

		A	T	T	A	T	C
	0	0	0	0	0	0	0
A	0	1	0	0	1	0	0
T	0	0	2	1	0	2	0
C	0	0	1	1	0	1	3
A	0	1	0	0	2	1	2
T	0	0	2	0	1	3	2



Local alignment example

X = **ATCAT**

Y = **ATTAT**C

		A	T	T	A	T	C
	0	0	0	0	0	0	0
A	0	1	0	0	1	0	0
T	0	0	2	1	0	2	0
C	0	0	1	1	0	1	3
A	0	1	0	0	2	1	2
T	0	0	2	0	1	3	2



Local alignment example

X = **ATC**AT

Y = ATT**ATC**

		A	T	T	A	T	C
	0	0	0	0	0	0	0
A	0	1	0	0	1	0	0
T	0	0	2	1	0	2	0
C	0	0	1	1	0	1	3
A	0	1	0	0	2	1	2
T	0	0	2	0	1	3	2

