## The SMITH-WATERMAN ALGORITHM (for Local Alignment)

$V(i,0) = 0, \ \forall i, \ 0 \le i \le m$

$V(0,j) = 0, \ \forall j, \ 0 \le j \le n$

For $i > 0$ and $j > 0$:

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

input: 2 sequences $X$ and $Y$ over some alphabet $\Sigma$ and scoring scheme $\delta$

find: 2 subsequences $\alpha$ and $\beta$ of $X$ and $Y$ whose optimal global alignment is MAXIMAL over all pairs of subsequences from $X$ and $Y$.

☆ The score of the optimal local alignment = value in matrix $V$ that is maximal

example:

$X$ = ACACTC

$Y$ = ACTCCA

$\delta$: identity: +1

mismatch: -1

indel: -2

| | | A | C | T | C | C | A |
|---|---|---|---|---|---|---|---|
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| A | 0 | 1 | 0 | 0 | 0 | 0 | 1 |
| C | 0 | 0 | 2 | 0 | 1 | 1 | 0 |
| A | 0 | 1 | 0 | 1 | 0 | 0 | 2 |
| C | 0 | 0 | 2 | 0 | 2 | 1 | 0 |
| T | 0 | 0 | 0 | 3 | 1 | 1 | 0 |
| C | 0 | 0 | 1 | 1 | 4 | 2 | 0 |

→ max local alignment:

A C T C
| | | |
A C T C

score: 4

(start backtrack @ $V^*$, stop when reach a zero)

- can think of 0's as restarts

## INTRO TO GAP ALIGNMENT

A C G - - - G G G
| | | | | | |
A T G A A T G G G

} biologists tend to prefer larger gaps as opposed to lots of little gaps

- A little bit of biological context:
  - DNA is transcribed into RNA, which is translated into proteins
  - RNA is spliced (non-coding introns are removed) before translation
  - SO, if you have cDNA (synthetic DNA transcribed from mRNA) and you want to align it with the genome, you want to use an alignment algorithm that is tolerant of long gaps to account for the RNA's missing introns
- How do we favor longer gaps algorithmically?
- Affine Gap Alignment has 2 parameters for gap penalties:
  - penalty for opening a gap: $\alpha$
  - penalty for continuing a gap: $\tau$

  } $\alpha > \tau$
- How do we score a gap cluster of size $n$?
  - logarithmic in $n$? Quadratic in $n$?

↳ most useful = linear in $n$:

score of gap cluster = $\alpha + n(\tau)$