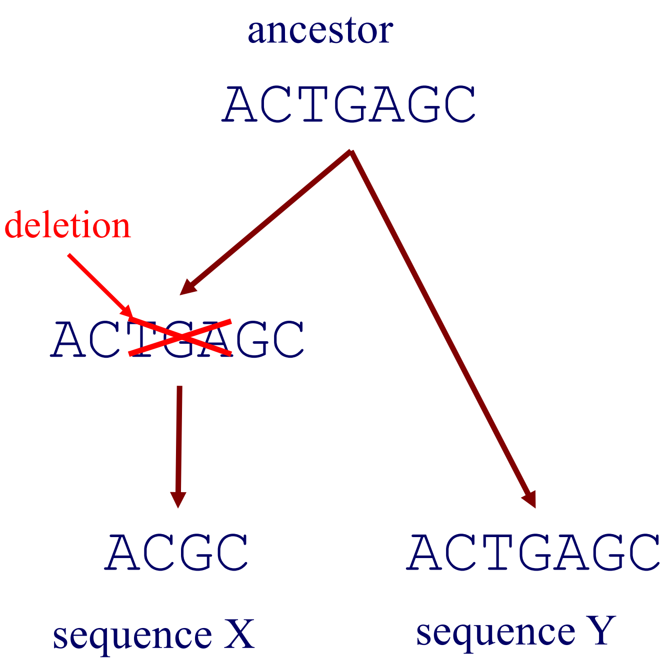
Sequence alignment

Alignment with affine gap penalty functions

Outline

- Affine gap penalty functions
- Affine gap global alignment algorithm
- Example run of the affine gap global alignment algorithm
- Affine gap local alignment algorithm
- More general gap penalty functions

Motivation for more complex gap penalty functions



With linear gap scoring scheme:
match = +1, mismatch = -1, space = -2

Alignment 1

```
AC-G--C
ACTGAGC
```

Alignment 2

```
AC---GC
ACTGAGC
```

Both alignments have score -2, but is one more biologically plausible than the other?

More complex gap penalty functions

- a gap of length *k* is more probable than *k* gaps of length 1
  - a gap may be due to a single mutational event that inserted/deleted a stretch of characters
  - separated gaps are probably due to distinct mutational events
- a linear gap penalty function treats these cases the same
- it is more common to use gap penalty functions involving two terms
  - a penalty *g* associated with <u>opening</u> a gap
  - a smaller penalty *s* for <u>extending</u> the gap

Gap Penalty Functions

- linear

$$w(k) = sk$$

- affine

$$w(k) = g + sk$$

Dynamic Programming for the Affine Gap Penalty Case

- to do in $O(n^2)$ time, need 3 matrices instead of 1

$M(i, j)$	best score given that $x[i]$ is aligned to $y[j]$
$I_x(i, j)$	best score given that $x[i]$ is aligned to a gap
$I_y(i, j)$	best score given that $y[j]$ is aligned to a gap

Why Three Matrices Are Needed

- consider aligning the sequences **`FW`** and **`WFP`** using g = -4, s = -1 and the following values from the BLOSUM-62 substitution matrix:

  S(**`F`**, **`W`**) = 1 S(**`W`**, **`W`**) = 11
  S(**`F`**, **`F`**) = 6 S(**`W`**, **`P`**) = -4
  S(**`F`**, **`P`**) = -4

- the matrix shows the highest-scoring partial alignment for each pair of prefixes

		W	**F**	**P**
	0	-5	-6	-7
F	-5	1	1	-4
W	-6	6	2	0

```
FW--
-WFP
```

optimal alignment

```
FW
WF
```

best alignment of these prefixes; to get optimal alignment, need to also remember

```
FW-
-WF
```

Global Alignment DP for the Affine Gap Penalty Case

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

$$I_x(i,j) = \max \begin{cases} M(i-1, j) + g + s \\ I_x(i-1, j) + s \end{cases}$$

$$I_y(i,j) = \max \begin{cases} M(i, j-1) + g + s \\ I_y(i, j-1) + s \end{cases}$$

Global Alignment DP for the Affine Gap Penalty Case

- initialization

  $$M(0,0) = 0$$

  $$I_x(i,0) = g + s \times i$$

  $$I_y(0,j) = g + s \times j$$

  other cells in top row and leftmost column $= -\infty$

- traceback
  - start at largest of $M(m,n), I_x(m,n), I_y(m,n)$
  - stop at any of $M(0,0), I_x(0,0), I_y(0,0)$
  - note that pointers may traverse all three matrices

Global Alignment Example
(Affine Gap Penalty)

g = -3, s = -1

		A	C	A	C	T
	M : 0 I_x : -3 I_y : -3	-∞ -∞ -4	-∞ -∞ -5	-∞ -∞ -6	-∞ -∞ -7	-∞ -∞ -8
A	-∞ -4 -∞	1 -∞ -∞	-5 -∞ -3	-4 -∞ -4	-7 -∞ -5	-8 -∞ -6
A	-∞ -5 -∞	-3 -3 -∞	0 -9 -7	-2 -8 -4	-5 -11 -5	-6 -12 -6
T	-∞ -6 -∞	-6 -4 -∞	-4 -4 -10	-1 -6 -8	-3 -9 -5	-4 -10 -6

Global Alignment Example (Continued)

		A	C	A	C	T
	M : 0 I_x : -3 I_y : -3	-∞ -∞ -4	-∞ -∞ -5	-∞ -∞ -6	-∞ -∞ -7	-∞ -∞ -8
A	-∞ -4 -∞	1 -∞ -∞	-5 -∞ -3	-4 -∞ -4	-7 -∞ -5	-8 -∞ -6
A	-∞ -5 -∞	-3 -3 -∞	0 -9 -7	-2 -8 -4	-5 -11 -5	-6 -12 -6
T	-∞ -6 -∞	-6 -4 -∞	-4 -4 -10	-1 -6 -8	-3 -9 -5	-4 -10 -6

three optimal alignments:

```
AA--T    A--AT    --AAT
ACACT    ACACT    ACACT
```

Local Alignment DP for the Affine Gap Penalty Case

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

$$I_x(i,j) = \max\begin{cases} M(i-1,j) + g + s \\ I_x(i-1,j) + s \end{cases}$$

$$I_y(i,j) = \max\begin{cases} M(i,j-1) + g + s \\ I_y(i,j-1) + s \end{cases}$$

Local Alignment DP for the Affine Gap Penalty Case

- initialization

  $M(0,0) = 0$

  $M(i,0) = 0$

  $M(0, j) = 0$

  cells in top row and leftmost column of $I_x, I_y = -\infty$

- traceback
  - start at largest $M(i, j)$
  - stop at $M(i, j) = 0$

Gap Penalty Functions

- linear: $$w(k) = sk$$
- affine:

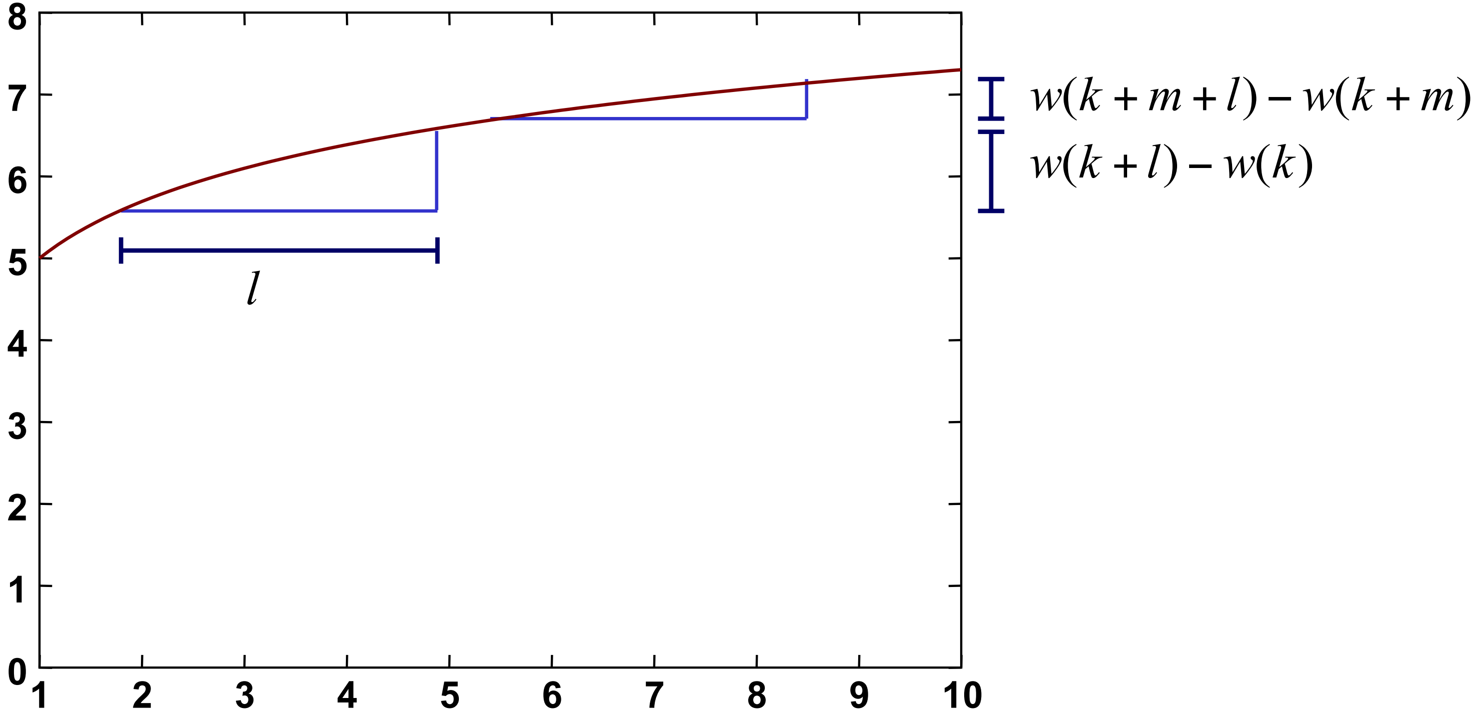
$$w(k) = g + sk$$

- concave: a function for which the following holds for all k, l, m $\geq$ 0

$$w(k+m+l) - w(k+m) \leq w(k+l) - w(k)$$

e.g. $$w(k) = g + s \times \log(k)$$

Concave Gap Penalty Functions



$$w(k+m+l) - w(k+m) \leq w(k+l) - w(k)$$

Computational Complexity and Gap Penalty Functions

- linear: $O(n^2)$
- affine: $O(n^2)$
- concave $O(n^2)$
- general: $O(n^3)$

* assuming two sequences of length *n*

Alignment (Global) with General Gap Penalty Function

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

consider every previous element in the row

consider every previous element in the column

Summary

- Affine gap penalty functions are more biologically realistic
- Similar dynamic programming algorithms are available for the affine gap case
  - involve three matrices instead of one
- The time complexity remains $O(n^2)$ for the affine gap and even concave gap cases
- Only an $O(n^3)$ algorithm is available for arbitrary gap functions