

CMSC 423:

Sequence Alignment

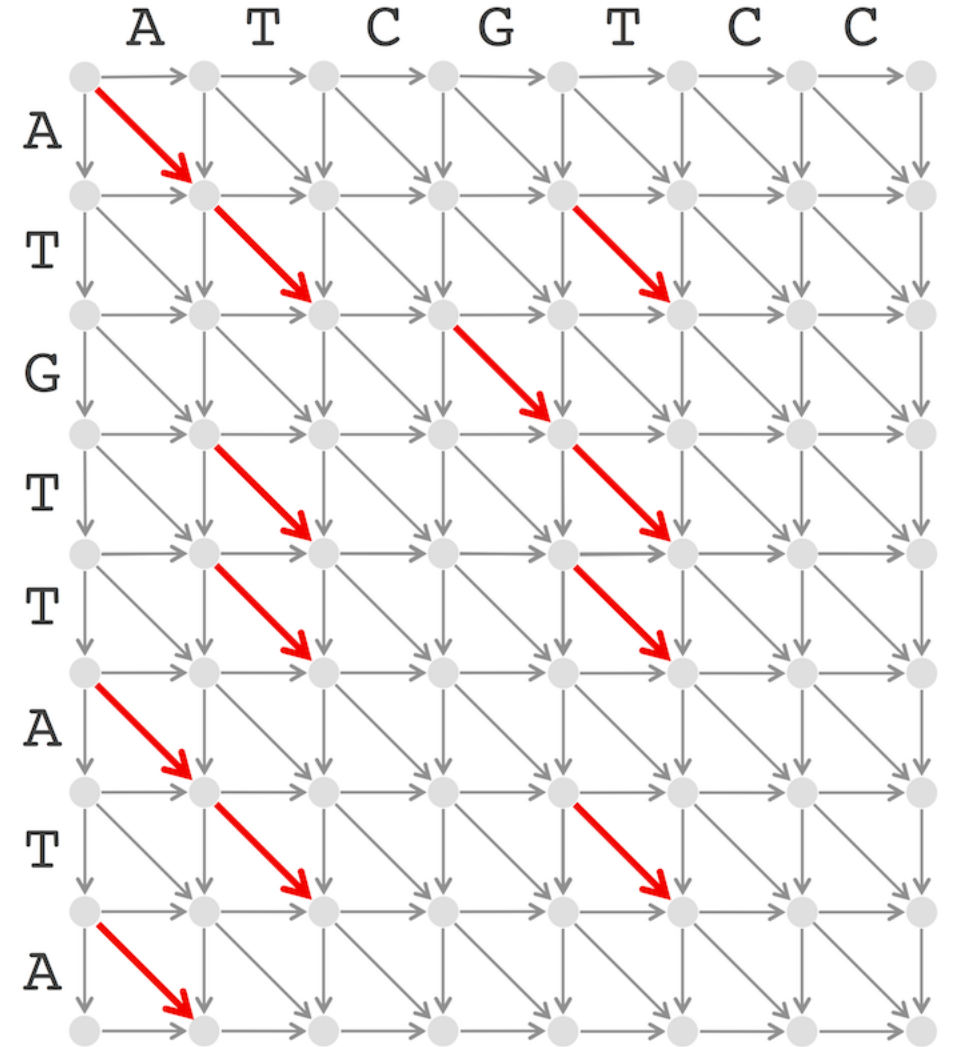
Part 3



and Think

Can we use the
alignment graph to find
a longest common
subsequence of two
strings?

Yes, with dynamic
programming!



Dynamic Programming

Idea: re-use previously computed information

Solve each smaller problem once, rather than billions of times

		i							
		-	A	G	C	G	T	A	G
j	-	0							
	G								
	T								
	C								
	A								
	G								
	A								

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{i,j} = \max \left\{ \begin{array}{l} s_{i-1,j-1} + 1, \\ \quad \text{if } v_i = w_j \\ s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \end{array} \right.$$

		i							
		-	A	G	C	G	T	A	G
j	-	0							
	G	0							
	T	0							
	C	0							
	A	0							
	G	0							
	A	0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{0,j} = \max \left\{ \begin{array}{l} s_{-1,j-1} + 1, \\ \quad \text{if } v_i = w_j \\ s_{-1,j} + 0 \\ s_{0,j-1} + 0 \end{array} \right.$$

		i							
		-	A	G	C	G	T	A	G
	-	0	0	0	0	0	0	0	0
j	G	0							
	T	0							
	C	0							
	A	0							
	G	0							
	A	0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{0,j} = \max \left\{ \begin{array}{l} s_{i-1,-1} + 1, \\ \text{if } v_i = w_j \\ s_{i-1,0} + 0 \\ s_{i,-1} + 0 \end{array} \right.$$

		i							
		-	A	G	C	G	T	A	G
	-	0	0	0	0	0	0	0	0
	G	0							
j	T	0							
	C	0							
	A	0							
	G	0							
	A	0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$A \neq G$

$s_{1,1} = \max$

$s_{0,0} + 1,$
if $v_1 \neq w_1$

$s_{0,1} + 0$

$s_{1,0} + 0$

		i						
	-	A	G	C	G	T	A	G
-	0	0	0	0	0	0	0	0
j	G	0						
	T	0						
	C	0						
	A	0						
	G	0						
A	0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{1,1} = \max \left\{ \begin{array}{l} s_{0,0} + 1, \\ \text{if } v_1 = w_1 \\ s_{0,1} + 0 \quad 0 \\ s_{1,0} + 0 \end{array} \right.$$

		i							
		-	A	G	C	G	T	A	G
-		0	0	0	0	0	0	0	0
G		0	0						
T		0							
C		0							
A		0							
G		0							
A		0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{1,1} = \max \left\{ \begin{array}{l} s_{0,0} + 1, \\ \quad \text{if } v_1 = w_1 \\ s_{0,1} + 0 \quad 0 \\ s_{1,0} + 0 \quad 0 \end{array} \right.$$

		i							
		-	A	G	C	G	T	A	G
	-	0	0	0	0	0	0	0	0
	G	0	0	1					
j	T	0							
	C	0							
	A	0							
	G	0							
	A	0							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$G = G$

$$s_{2,1} = \max \left\{ \begin{array}{ll} s_{1,0} + 1, & 1 \\ & \text{if } v_1 = w_1 \\ s_{1,1} + 0 & 0 \\ s_{2,0} + 0 & 0 \end{array} \right.$$

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{i,j} = \max \left\{ \begin{array}{l} s_{i-1,j-1} + 1, \\ \quad \text{if } v_i = w_j \\ s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \end{array} \right.$$

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{i,j} = \max \begin{cases} s_{i-1,j-1} + 1, & \text{if } v_i = w_j \\ s_{i-1,j} + 0 \\ s_{i,j-1} + 0 \end{cases}$$

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

G
—

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

A G
A -

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

T A G
- A -

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

G T A G
G - A -

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

- G T A G
A G - A -

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

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

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

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

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

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

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

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

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

Extending to sequence alignment

- When solving for the LCS, mis-alignments are free
- What happens if we pay for our "mistakes"? (this also allows us to account for "similar" amino acids)

Value[Match] = 10

Value[Mismatch] = -5

Value[Gap] = -2

Match = [A,A], [C,C], [G,G], [T,T]

Mismatch = [A,G], [A,C], [A,T], ...

Gap = [A,-], [-,A], ...

The same dynamic programming algorithm works!

	-	A	G	C	G	T	A	G
-	0	-2	-4	-6	-8	-10	-12	-14
G	-2	-4	8	6				
T	-4	-6	6	4				
C	-6	-8	4	16				
A	-8							
G	-10							
A	-12							

$v = \text{AGCGTAG}$

$w = \text{GTCAGA}$

$$s_{i,j} = \max \left\{ \begin{array}{l} s_{i-1,j-1} + \text{Value}[v[i], w[j]] \\ s_{i-1,j} + \text{Value}[v[i], -] \\ s_{i,j-1} + \text{Value}[-, w[j]] \end{array} \right.$$

Match = 10

Mismatch = -5

Gap = -2