

# Sequence Alignment 2

BCB 567, Fall 2007

# Global Alignment (review)

- Input: Two strings, labeled  $A$  and  $B$ 
  - Length of  $A$  we call  $n$
  - Length of  $B$  we call  $m$
- Output: Two strings  $A_A$  and  $B_A$  such that:
  - $A_A$  and  $B_A$  are of equal length  $L$ ,
  - The characters of  $A_A$  consist only characters in  $A$  in the same order, with a special gap character '-' possibly inserted in some positions.
  - The characters of  $B_A$  consist only characters in  $B$  in the same order, with a special gap character '-' possibly inserted in some positions.
  - If  $A_A[i] = '-'$ , then  $B_A[i] \neq '-'$
  - If  $B_A[i] = '-'$ , then  $A_A[i] \neq '-'$

# Optimal Alignment (review)

- Optimal Alignment
  - Find an alignment that optimizes an evaluation function:
    - $E(A_A, B_A)$
- Finding the optimal alignment is hard problem (NP-hard) in general because of the exponential size of the solution space.
- When we make assumptions about the scoring function, we can solve the problem much faster.

# Simple Scoring Method (review)

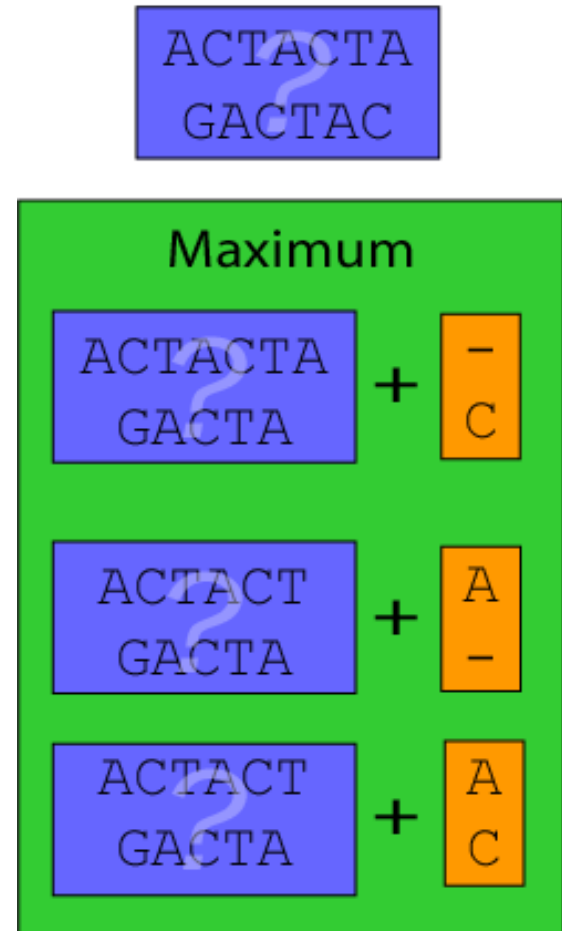
- Motivations
  - Sequencing Error
  - Evolutionary Model
- Gap Penalty:  $\gamma$ 
  - A missed or added base during reading
  - A insertion or deletion mutation event
- Mismatch Penalty:  $\beta$ 
  - A misread base
  - A substitution mutation event
- Match Bonus:  $\alpha$ 
  - Represents a proper read of a base
  - Represents no change between the organisms

# Recursive Formulation (review)

- Can be solved inductively by looking at three smaller problems

$$S(i, j) = \max \begin{cases} S(i, j-1) + \gamma \\ S(i-1, j) + \gamma \\ S(i-1, j-1) + \delta(i-1, j-1) \end{cases}$$

$$\delta(i, j) = \begin{cases} \alpha & A[i] = B[j] \\ \beta & A[i] \neq B[j] \end{cases}$$



# The Dynamic Programming Table

		Sequence B											
		A	G	C	T	A	A	G	C	T	A	A	
Sequence A	$s[0,0]$ →	0	-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11
	A	0	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
	G	-1	1	4	3	2	1	0	-1	-2	-3	-4	-5
	C	-2	0	3	6	5	4	3	2	1	0	-1	-2
	C	-3	-1	2	5	5	4	3	2	4	3	2	1
	T	-4	-2	$s[5,4]$ →	7	6	5	4	3	6	5	4	
	A	-5	-3	0	3	6	9	8	7	6	5	8	7
	G	-6	-4	-1	2	5	8	7	10	9	8	7	6
	C	-7	-5	-2	1	4	7	6	9	12	11	10	9
	C	-8	-6	-3	0	3	6	5	8	11	11	10	9
	A	-9	-7	-4	-1	2	5	8	7	10	10	13	12
A	-10	-8	-5	-2	1	4	7	6	9	9	12	15	

column



$$S[i,j] = S[i-1,j-1]$$



row

$$S[i,0] = i\gamma$$

$$S[0,j] = j\gamma$$

$$S[i,j] = \max \begin{cases} S[i-1,j-1] + \delta(i-1,j-1) \\ S[i,j-1] + \gamma \\ S[i-1,j] + \gamma \end{cases}$$

	$\lambda$	C	T	C	G	C	A	G	C
$\lambda$	0	-5	-10	-15	-20	-25	-30	-35	-40
C	-5	10	5						
A	-10								
T	-15								
T	-20								
C	-25								
A	-30								
C	-35								

+10 for match, -2 for mismatch, -5 for gap

# What are the time and space requirements of this Algorithm?

- $O(nm)$  time
- $O(nm)$  space

# Practice

$$S[i,0] = i\gamma$$

$$S[0,j] = j\gamma$$

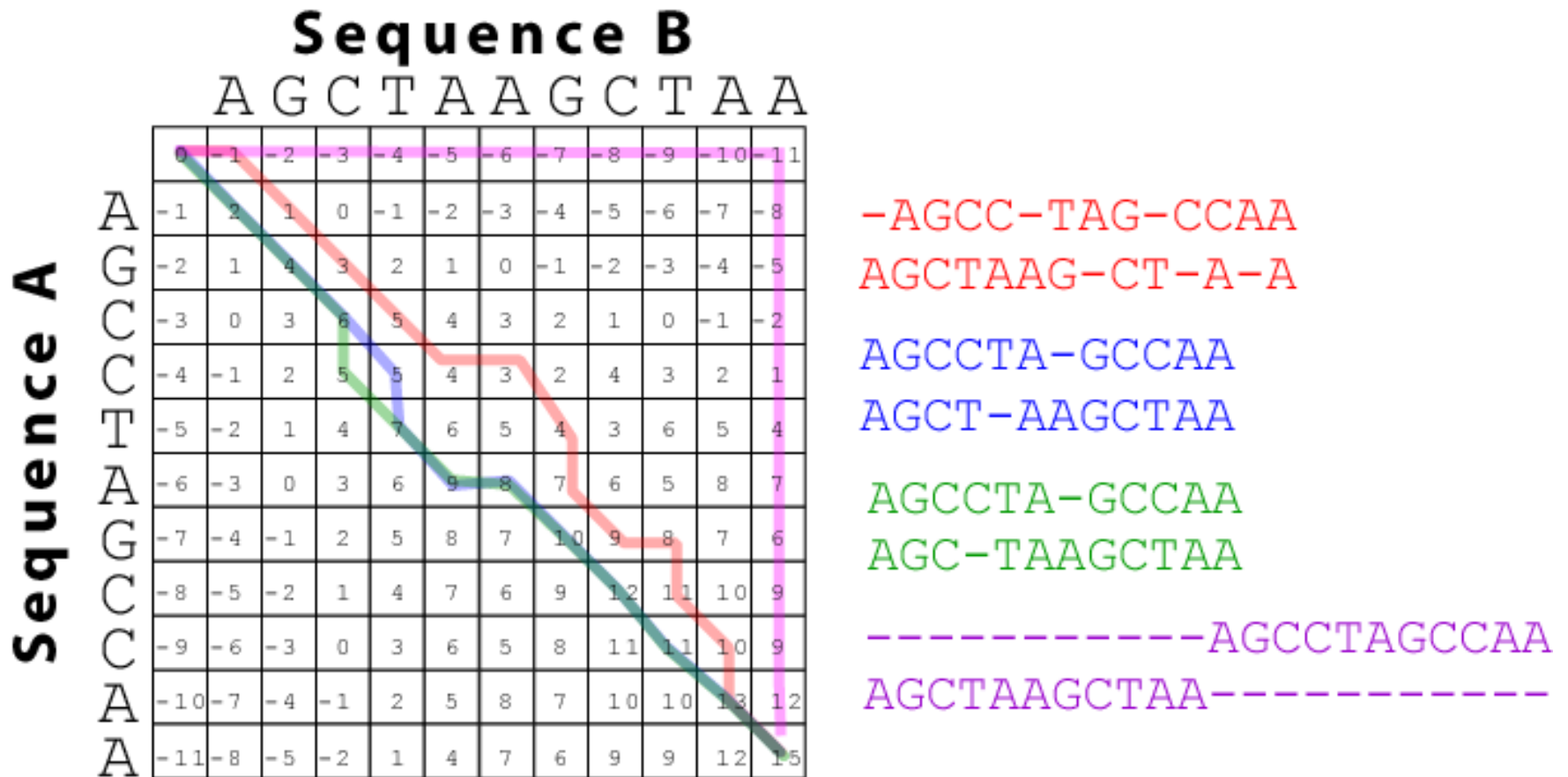
$$\alpha=5, \beta=-2, \gamma=-1$$

A=AGCATTA  
B=ACATTTAG

$$S[i,j] = \max \begin{cases} S[i-1, j-1] + \delta(i-1, j-1) \\ S[i, j-1] + \gamma \\ S[i-1, j] + \gamma \end{cases}$$

Find the score of the best alignment

# Meaning of Paths



# Finding the Optimal Path

**Sequence B**

A G C T A A G C T A A

		-1	-2	-3	-4	-5	-6	-7	-8	-9	-10	-11	
<b>Sequence A</b>	A	-1	2	1	0	-1	-2	-3	-4	-5	-6	-7	-8
	G	-2	1	4	3	2	1	0	-1	-2	-3	-4	-5
	C	-3	0	3	6	5	4	3	2	1	0	-1	-2
	C	-4	-1	2	5	5	4	3	2	4	3	2	1
	T	-5	-2	1	4	7	6	5	4	3	6	5	4
	A	-6	-3	0	3	6	9	8	7	6	5	8	7
	G	-7	-4	-1	2	5	8	7	10	9	8	7	6
	C	-8	-5	-2	1	4	7	6	9	12	11	10	9
	C	-9	-6	-3	0	3	6	5	8	11	11	10	9
	A	-10	-7	-4	-1	2	5	8	7	10	10	13	12
	A	-11	-8	-5	-2	1	4	7	6	9	9	12	15

AGCCTA-GCCAA  
AGC-TAAGCTAA

$$\begin{aligned}
 S[i, j] &= S[i-1, j-1] + \delta(i-1, j-1) && \cdot \\
 S[i, j] &= S[i-1, j] + \lambda && \dots \\
 S[i, j] &= S[i, j-1] + \lambda && \vdots
 \end{aligned}$$

	$\lambda$	C	T	C	G	C	A	G	C
$\lambda$	0	-5	-10	-15	-20	-25	-30	-35	-40
C	-5	10	5	0	-5	-10	-15	-20	-25
A	-10	5	8	3	-2	-7	0	-5	-10
T	-15	0	15	10	5	0	-5	-2	-7
T	-20	-5	10*	13	8	3	-2	-7	-4
C	-25	-10	5	20	15	18	13	8	3
A	-30	-15	0	15	18	13	28	23	18
C	-35	-20	-5	10	13	28	23	26	33

Traceback yields both optimal alignments in this example

# Practice

$$S[i,0] = i\gamma$$

$$S[0,j] = j\gamma$$

$$S[i,j] = \max \begin{cases} S[i-1, j-1] + \delta(i-1, j-1) \\ S[i, j-1] + \gamma \\ S[i-1, j] + \gamma \end{cases}$$

$$\alpha=5, \beta=-2, \gamma=-1$$

A=AGCATT A

B=ACATTTAG

$$S[i,j] = S[i-1, j-1] + \delta(i-1, j-1) \quad \cdot \cdot$$

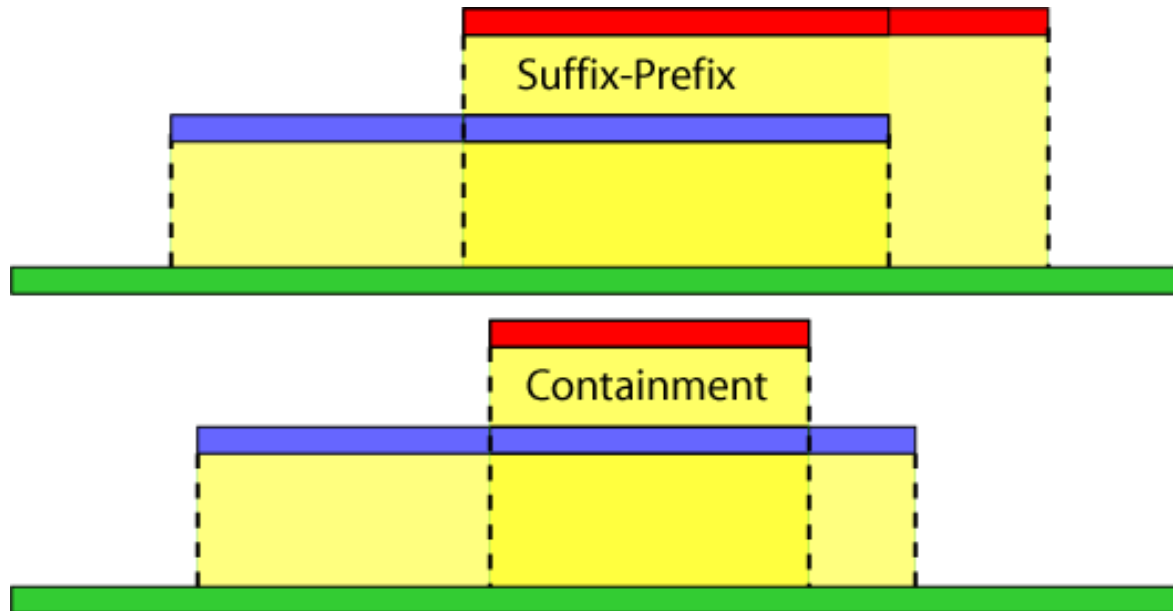
$$S[i,j] = S[i-1, j] + \gamma \quad \dots$$

$$S[i,j] = S[i, j-1] + \gamma \quad \vdots$$

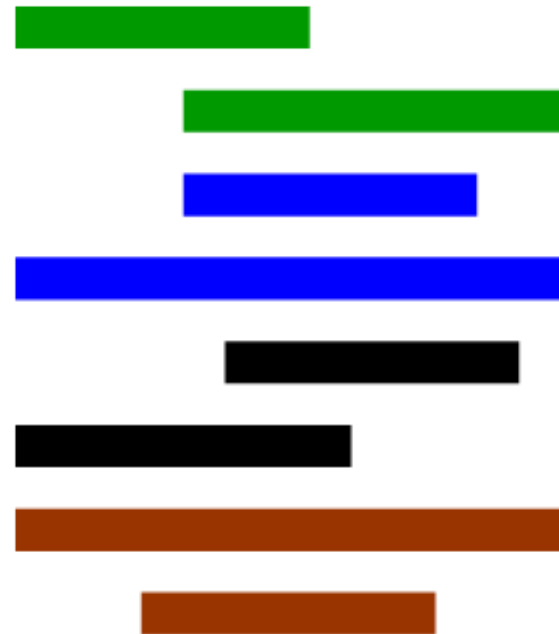
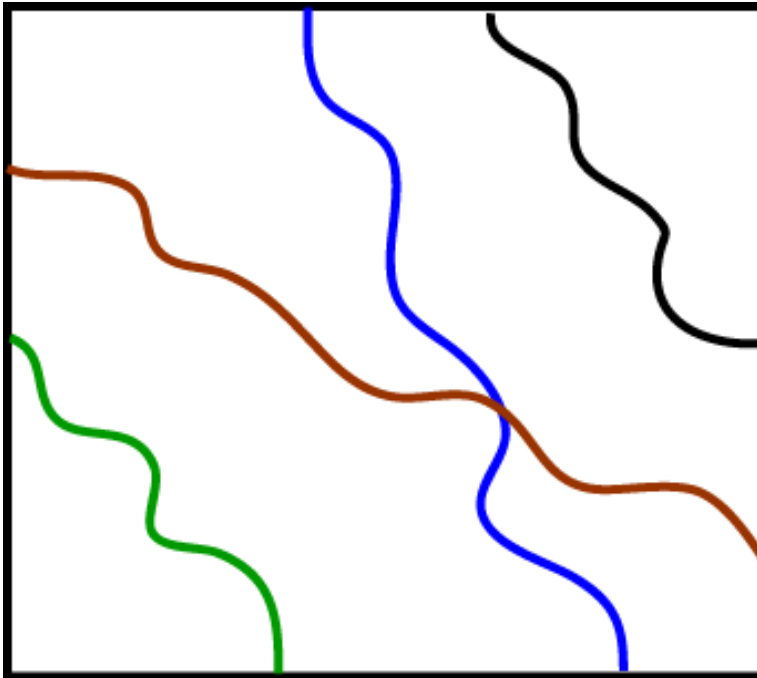
Find an optimal alignment

# Semi Global Alignment

- Motivation:
  - The sequence assembly problem
- Corresponds to deciding if both sequences are pieces of some larger sequence.
- “Allowed to ignore a prefix of A or a prefix of B, but not both.”
- “Allowed to ignore a suffix of A or a suffix of B, but not Both”



# Semi-global Paths



# Modifying the endpoints of an alignment path

- The starting point of the path:
  - Choosing the maximum
- The end point of a path
  - Finding a zero entry
  - Must set an entry to zero that would otherwise be negative.

	$\lambda$	C	T	C	G	C	A	G	C
$\lambda$	0	0	0	0	0	0	0	0	0
C	0	10	5	10	5	10	5	0	10
A	0	5	8	5	8	5	20	15	10
T	0	0	15	10	5	6	15	18	13
T	0	-2	10	13	8	3	10	13	16
C	0	10	5	20	15	18	13	8	23
A	0	5	8	15	18	13	28	23	18
G	0	0	3	10	25	20	23	38	33

+10 for match, -2 for mismatch, -5 for gap

# Practice

$$S[i,0] = 0$$

$$S[0,j] = 0$$

$$S[i,j] = \max \begin{cases} S[i-1, j-1] + \delta(i-1, j-1) \\ S[i, j-1] + \gamma \\ S[i-1, j] + \gamma \end{cases}$$

$$\alpha=5, \beta=-2, \gamma=-1$$

A=ACTAG  
B=TACACT

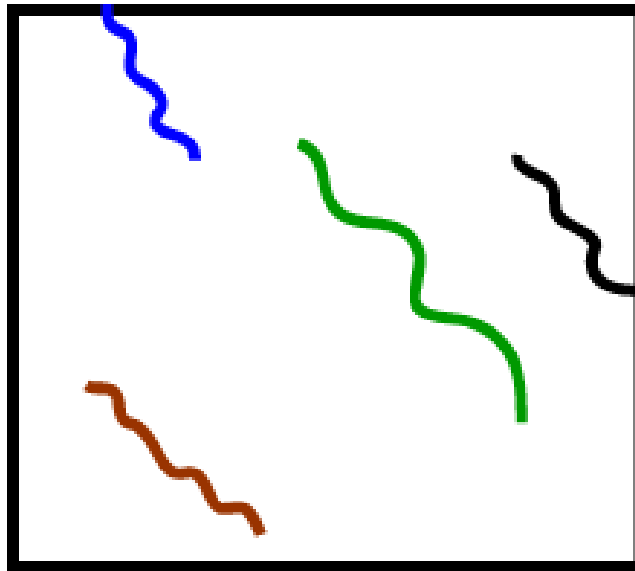
$$S[i,j] = S[i-1, j-1] + \delta(i-1, j-1) \quad \ddots$$

$$S[i,j] = S[i-1, j] + \gamma \quad \dots$$

$$S[i,j] = S[i, j-1] + \gamma \quad \vdots$$

Find an optimal semi-global alignment

# Local Alignment Paths



# Local Alignment

We must modify this recursion to allow for alignments to stop at any point in the table.

$$S[i,0] = 0$$

$$S[0,j] = 0$$

$$S[i,j] = \max \begin{cases} S[i-1, j-1] + \delta(i-1, j-1) \\ S[i, j-1] + \gamma \\ S[i-1, j] + \gamma \\ 0 \end{cases}$$

	$\lambda$	C	T	C	G	C	A	G	C
$\lambda$	0	0	0	0	0	0	0	0	0
C	0	1	0	1	0	1	0	0	1
A	0	0	0	0	0	0	2	0	0
T	0	0	1	0	0	0	0	1	0
T	0	0	1	0	0	0	0	0	0
C	0	1	0	2	0	1	0	0	1
A	0	0	0	0	1	0	2	0	0
C	0	1	0	1	0	2	0	1	1

+1 for a match, -1 for a mismatch, -5 for a gap

# Practice

$$S[i,0] = 0$$

$$S[0,j] = 0$$

$$S[i,j] = \max \left\{ \begin{array}{l} S[i-1, j-1] + \delta(i-1, j-1) \\ S[i, j-1] + \gamma \\ S[i-1, j] + \gamma \\ 0 \end{array} \right.$$

$$\alpha=5, \beta=-2, \gamma=-1$$

A=AAACTAGT

B=TCACTAC

$$S[i,j] = S[i-1, j-1] + \delta(i-1, j-1) \quad \ddots$$

$$S[i,j] = S[i-1, j] + \gamma \quad \dots$$

$$S[i,j] = S[i, j-1] + \gamma \quad \vdots$$

Find an optimal local alignment

# Some Results

- Most pairwise sequence alignment problems can be solved in  $O(mn)$  time.
- Space requirement can be reduced to  $O(m+n)$ , while keeping run-time fixed [Myers88].
- Two highly similar sequences can be aligned in  $O(dn)$  time, where  $d$  is a measure of the distance between the sequences [Landau86].