BCB 5200 Introduction to Bioinformatics

Pairwise Sequence Alignment 2

Bioinformatics and Computational Biology
Saint Louis University

Outlines

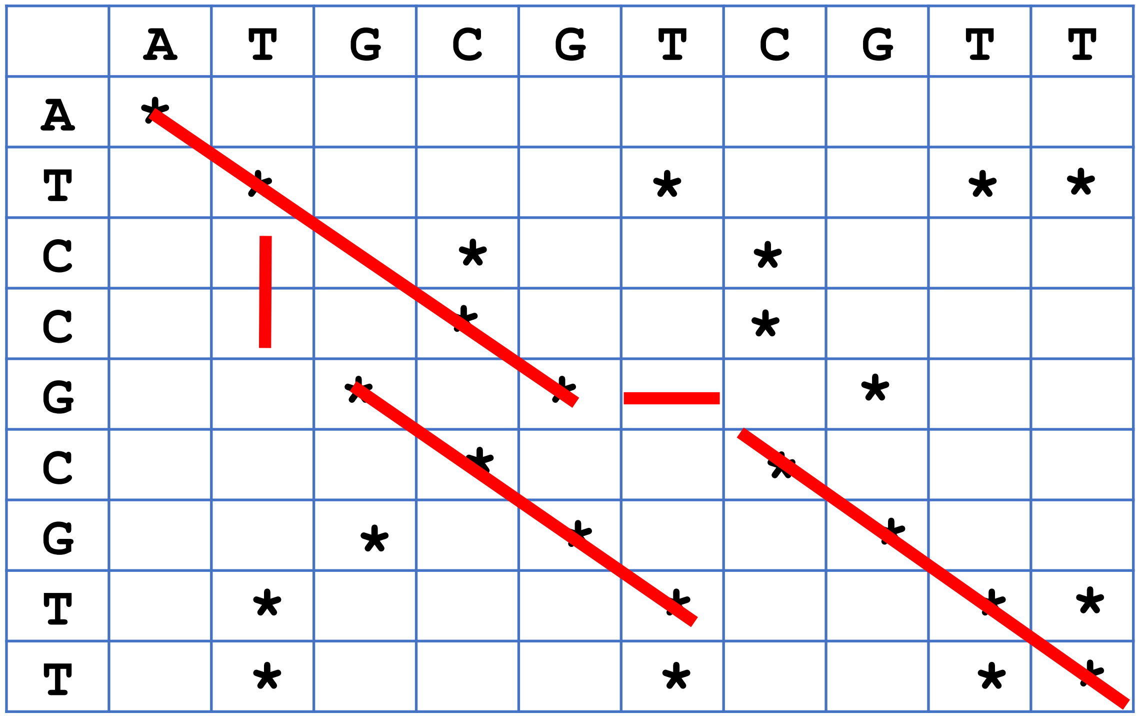
- Similarity & Homology
- Basic components of sequence alignment
  - Similarity or scoring Matrix
  - Gap penalties
- Dot matrix analysis
- Dynamic programming algorithm
  - Global sequence alignment: Needleman-Wunsch (NW) algorithm
  - Local sequence alignment: Smith-Waterman (SW) algorithm

Algorithms for sequence alignment

- Dot Matrix Method (Gibbs and McIntyre 1970)
- Dynamic programming

- Common steps:
  1. Setting up a two-dimensional matrix
  2. Matching or scoring the matrix
  3. Identifying the optimal alignment

Dot matrix analysis



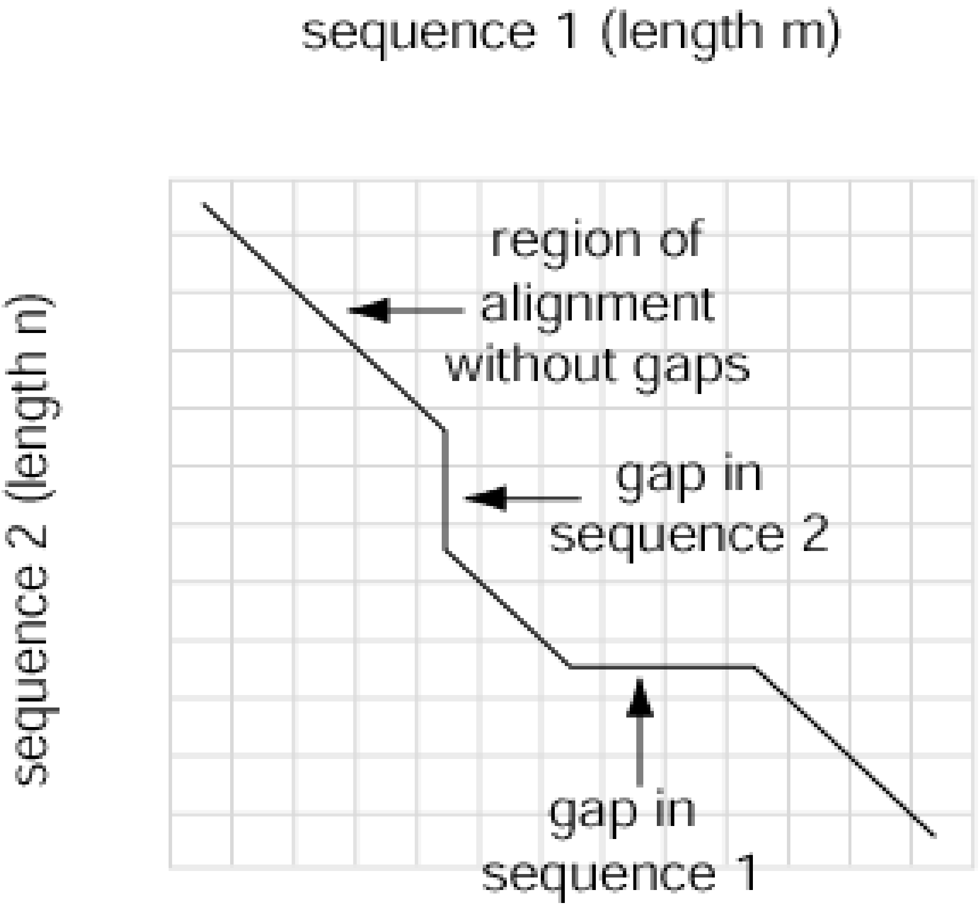
Sequence (A)

Sequence (B)

	A	T	G	C	G	T	C	G	T	T
A	*									
T		*				*			*	*
C				*			*			
C				*			*			
G			*		*			*		
C				*			*			
G			*		*			*		
T		*				*			*	*
T		*				*			*	*

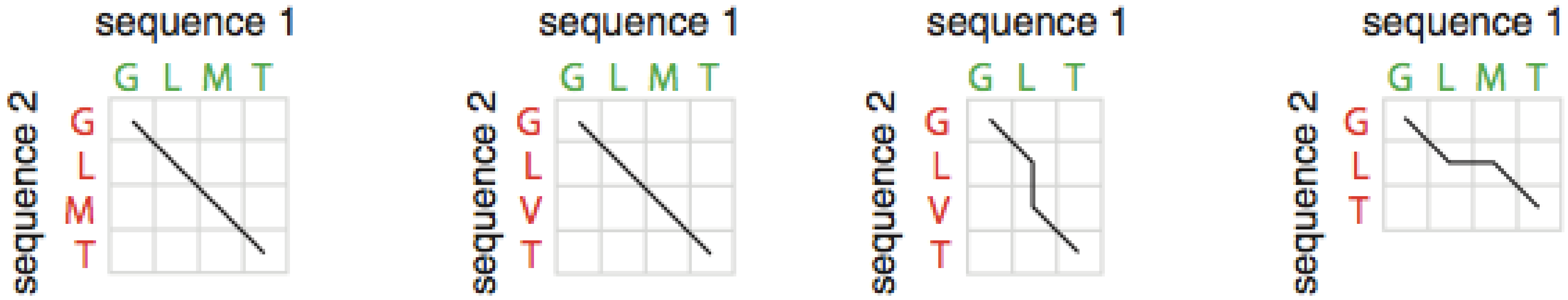
```
ATGCGTCGTT          AT--GCGTCGTT
||  ||  ||||        ||    ||||
ATCCG-CGTT          ATCCGCGTT---
```

Four possible outcomes in aligning two sequences



1. identity (stay along a diagonal)
2. mismatch (stay along a diagonal)
3. gap in one sequence 1 (move vertically!)
4. gap in the other sequence 2 (move horizontally!)

Four possible outcomes in aligning two sequences



```
1 GLMT
2 GLMT
```

```
1 GLMT
2 GLVT
```

```
1 GL-T
2 GLVT
```

```
1 GLMT
2 GL-T
```

Dynamic programming algorithm for pairwise sequence alignment

- Global alignment: Needleman-Wunsch (NW) algorithm (1970)

- Local alignment: Smith-Waterman (SW) algorithm (1981)

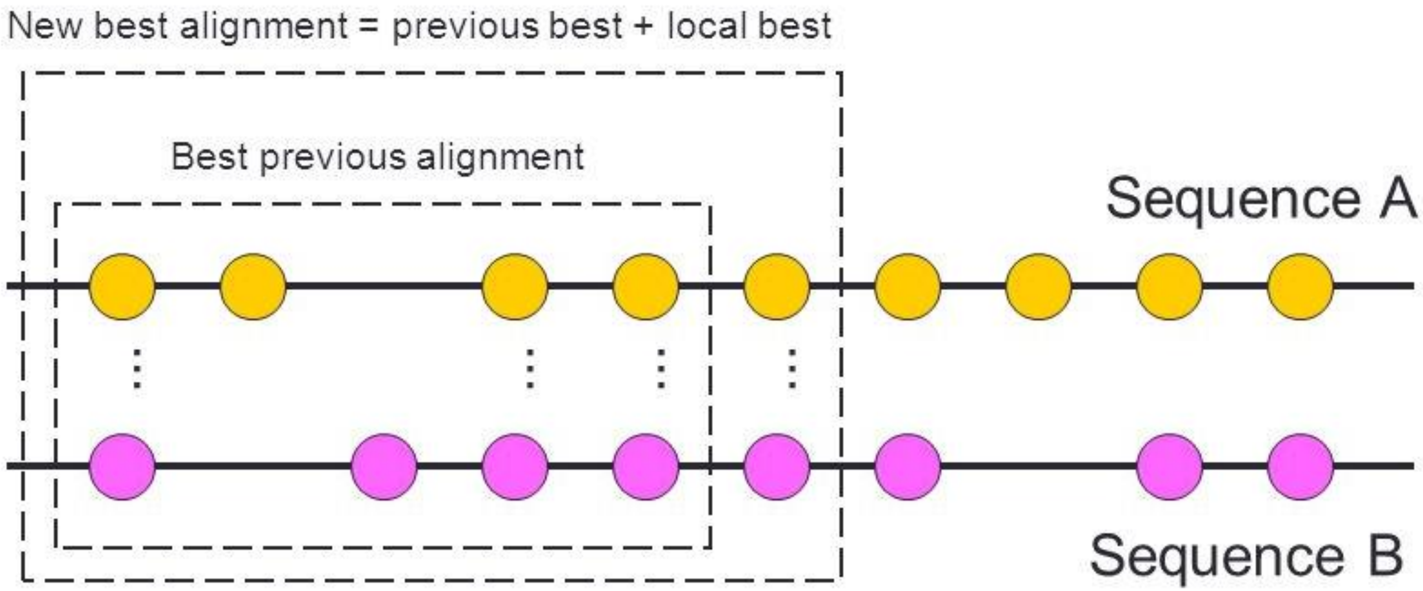
Dynamic programming (DP)

- Recursive approach, sequential dependency.

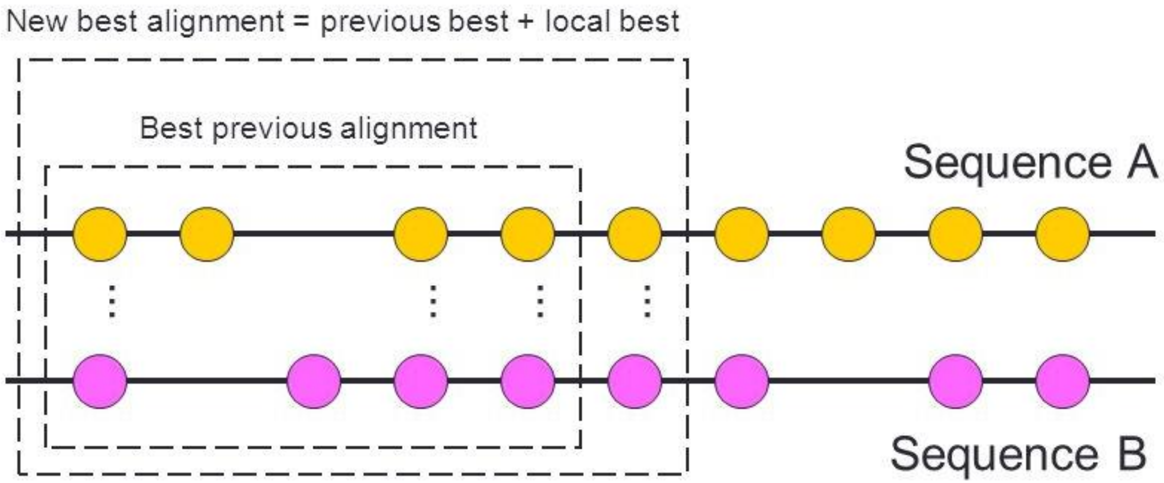
- DP solves problems by combining the solutions to sub-problems.
  - Break the problem into smaller sub-problems
  - Solve these sub-problems optimally recursively
  - Use these optimal solutions to construct an optimal solution for the original problem

Dynamic programming (DP)

- The best alignment that ends at a given pair of symbols is the best alignment of the sequences up to that point, plus the best alignment for the two additional symbols



Dynamic programming (DP)



- If you already have the optimal solution to:

X…Y
A…B

- Then you know the next pair of characters will either be:

X…YZ **A…BC**	or	**X…Y-** **A…BC**	or	**X…YZ** **A…B-**

- You can extend the match by determining which of these has the highest score

Sequence alignment with Dynamic Programming: the formula

- Align two sequences: x and y
  - D (*i-1, j-1*) is the score of the best alignment between $x_{1..i-1}$ and $y_{1..j-1}$
  - s (x_i, y_j) is the score for substituting i with j; g is the gap penalty

$$F\ (0,0) = 0$$

$$D(i,j) = \max \begin{cases} D(i-1, j-1) + s(x_i, y_j) \\ D(i-1, j) + g \\ D(i, j-1) + g \end{cases}$$

X…YZ

A…BC

X…Y-

A…BC

X…YZ

A…B-

Three steps for sequence alignment with the Needleman-Wunsch algorithm (1970)

[1] set up a matrix

[2] score the matrix

[3] identify the optimal alignment(s)

Needleman-Wunsch algorithm to find the best alignment of these two sequences

1. Setting up the matrix

Seq1:ACTGATTCA

Seq2:ACGCATCA

Seq1 length m = 9

Seq2 length n = 8

Draw a matrix (m+1) × (n+1)

It means a 10 × 9 matrix

Assign 0 to the top left cell

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

Needleman-Wunsch algorithm to find the best alignment of these two sequences

1. Setting up the matrix
2. Scoring the matrix

match = 2
mismatch = -3
gap = -2

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

Needleman-Wunsch algorithm to find the best alignment of these two sequences

1. Setting up the matrix
2. Scoring the matrix

1) Add gap penalties in the first row and column. Each gap position receives a score of -2

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2									
C	-4									
G	-6									
C	-8									
A	-10									
T	-12									
C	-14									
A	-16									

Needleman-Wunsch algorithm to find the best alignment of these two sequences

1. Setting up the matrix
2. Scoring the matrix

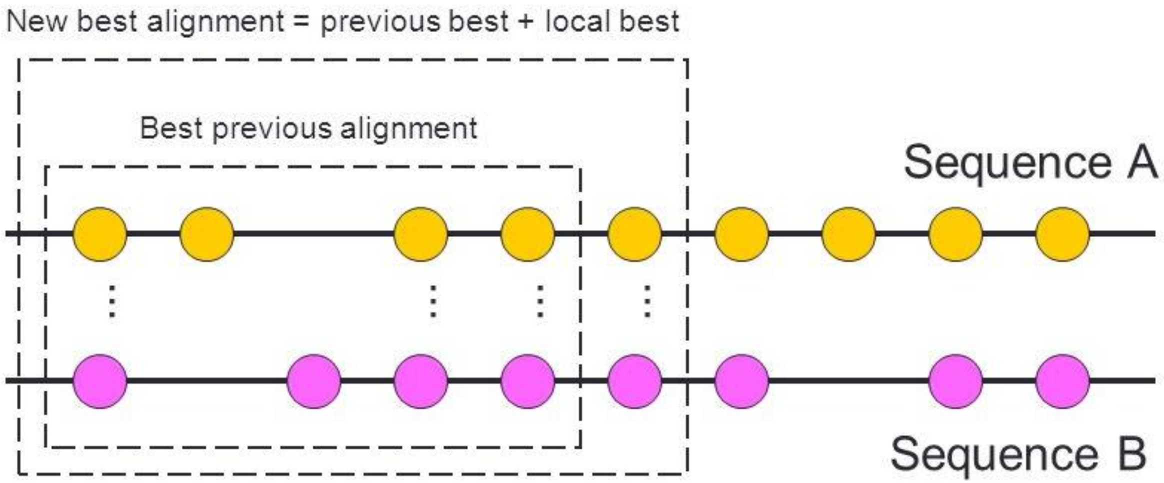
1) Add gap penalties in the first row and column. Each gap position receives a score of -2

```
ACTGATTCA--------
---------ACGCATCA
```

```
--------ACTGATTCA
ACGCATCA---------
```

		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2									
C	-4									
G	-6									
C	-8									
A	-10									
T	-12									
C	-14									
A	-16									

Dynamic programming (DP)



- If you already have the optimal solution to:

X…Y
A…B

- Then you know the next pair of characters will either be:

X…YZ **A…BC**	or	**X…Y-** **A…BC**	or	**X…YZ** **A…B-**

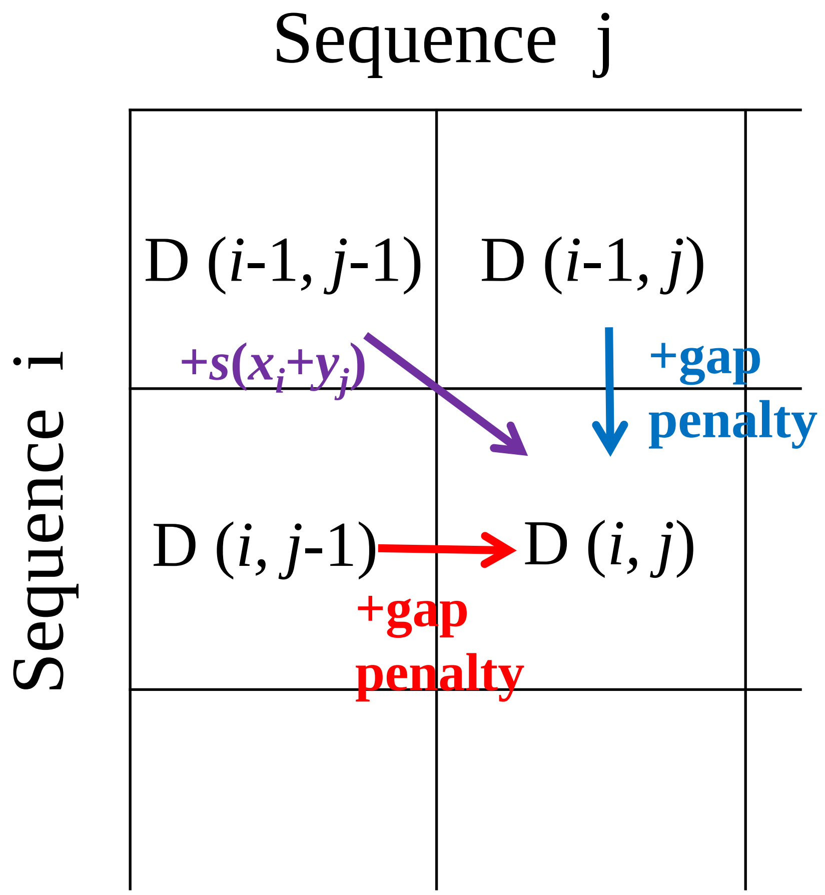
- You can extend the match by determining which of these has the highest score

Sequence alignment with Dynamic Programming: the formula

- A matrix D(i, j) indexed by residues of each sequence is built recursively, such that

$$D(i,j) = \max \begin{cases} D(i-1, j-1) + s(x_i, y_j) \\ D(i-1, j) + g \\ D(i, j-1) + g \end{cases}$$

s(I, j) is the substitution score for residues i and j, and g is the gap penalty. subject to a boundary conditions.



Sequence j

Sequence i

D (i-1, j-1)

D (i-1, j)

$+s(x_i+y_j)$

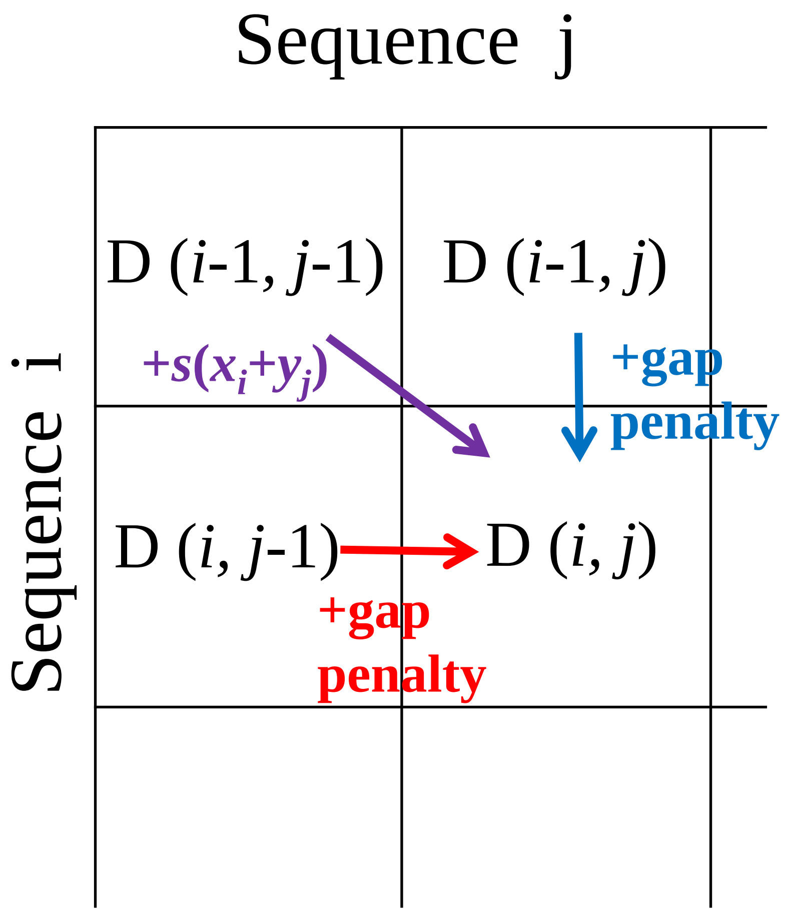
+gap penalty

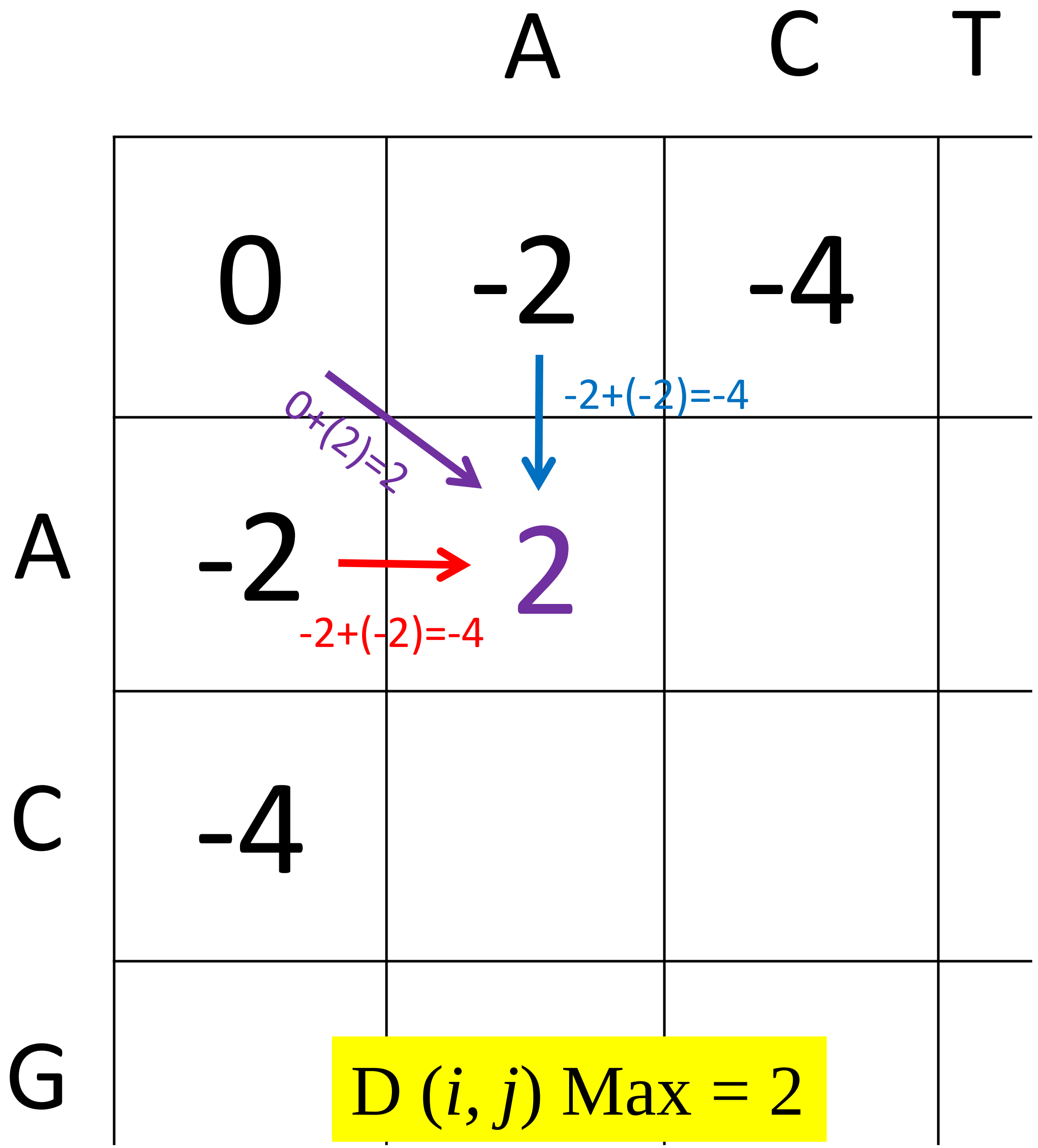
D (i, j-1)

D (i, j)

+gap penalty

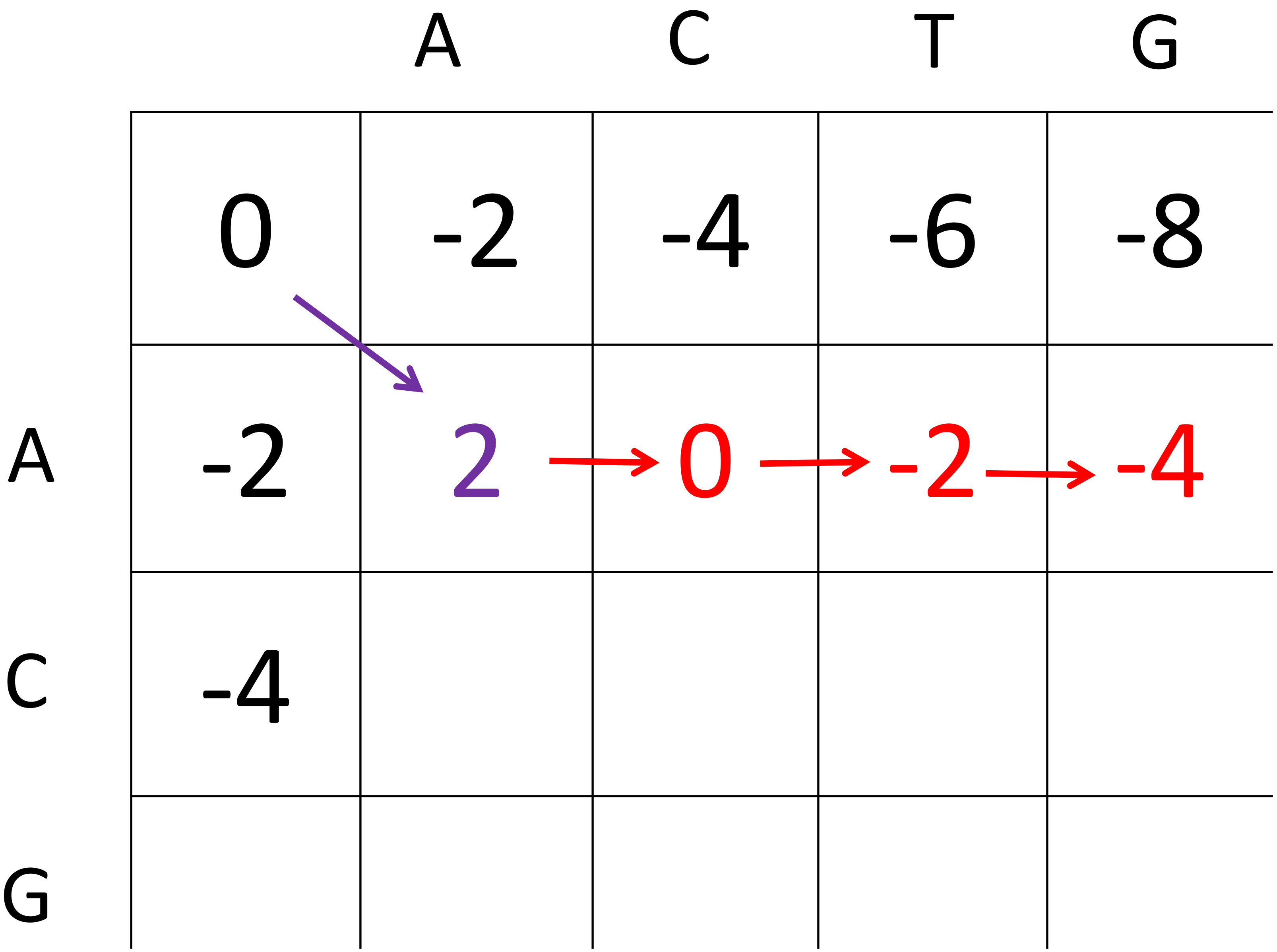
2) Assign best score in each position





D (*i*, *j*) Max = 2

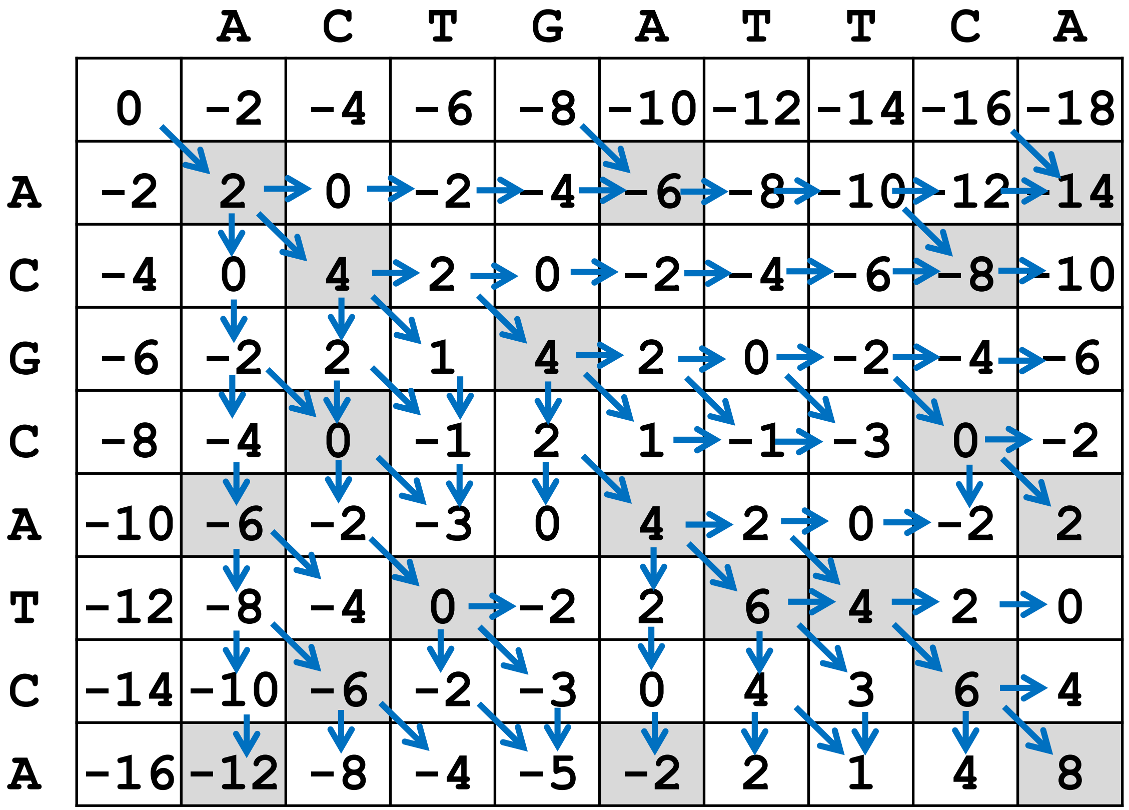
match = 2
gap = -2
mismatch = -3



		A	C	T	G
	0	-2	-4	-6	-8
A	-2	2	0	-2	-4
C	-4				
G					

Needleman-Wunsch algorithm to find the best alignment of these two sequences

match = 2, gap = -2 , mismatch = -3

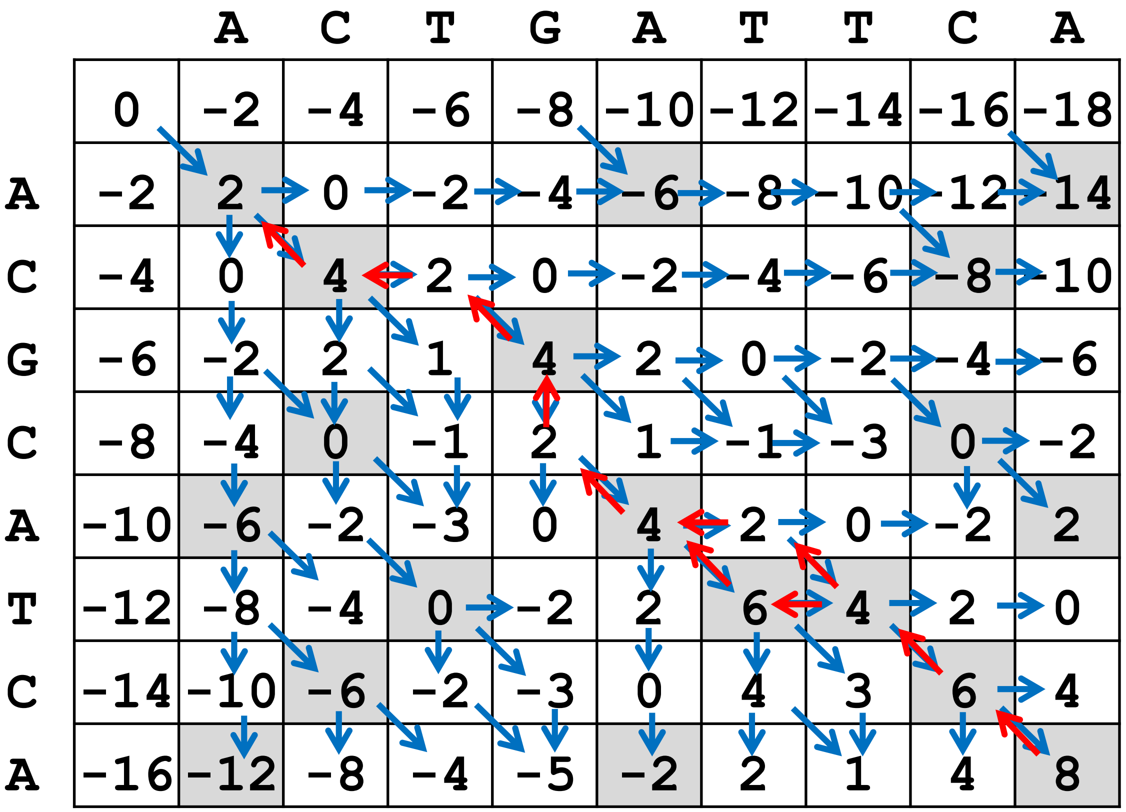


		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-3	0	4	3	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

If two or more arrows have identical values, keep all of them

3. Identifying the optimal alignment using a trace-back procedure

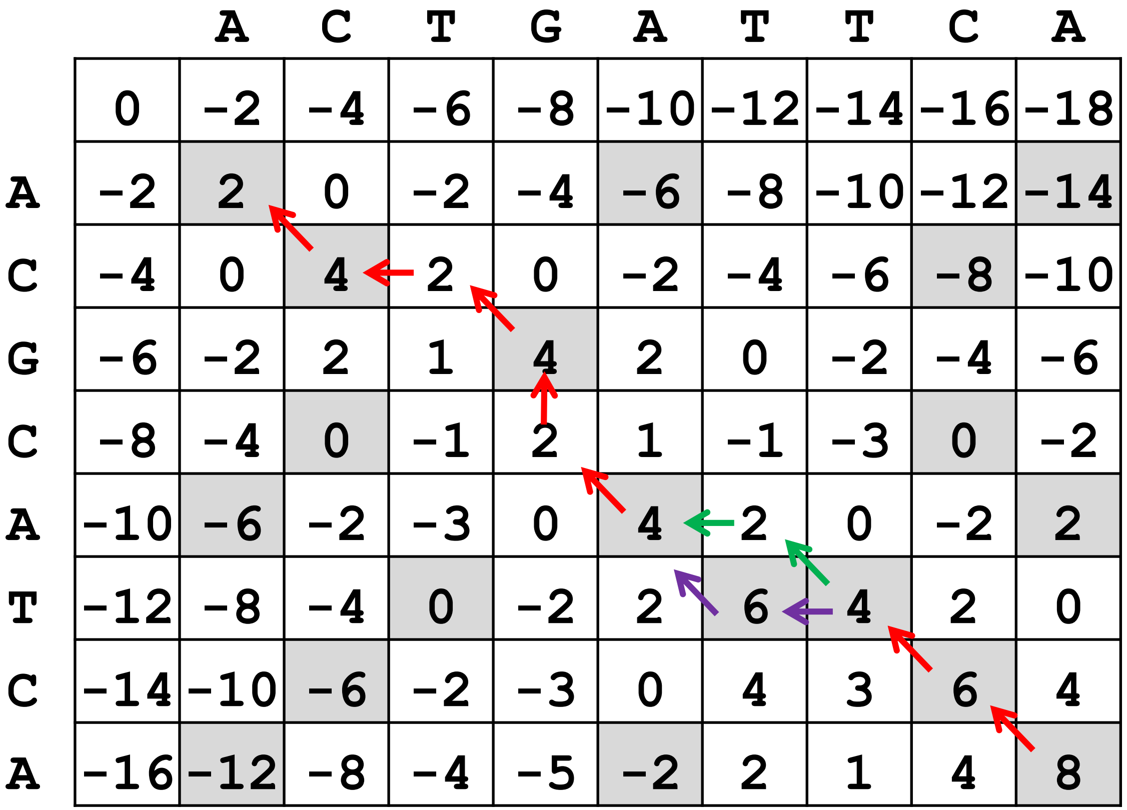
- Trace-back = the process of deduction of the best alignment from the traceback matrix
- The traceback always begins with the last cell to be filled with the score, i.e. the bottom right cell.
- We use arrows to point back the source of each cell's best score
- The traceback is completed when the first, top-left cell of the matrix is reached (”done” cell).



		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-3	0	4	3	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

Generate alignments

match = 2
gap = -2
mismatch = -3



		A	C	T	G	A	T	T	C	A
	0	-2	-4	-6	-8	-10	-12	-14	-16	-18
A	-2	2	0	-2	-4	-6	-8	-10	-12	-14
C	-4	0	4	2	0	-2	-4	-6	-8	-10
G	-6	-2	2	1	4	2	0	-2	-4	-6
C	-8	-4	0	-1	2	1	-1	-3	0	-2
A	-10	-6	-2	-3	0	4	2	0	-2	2
T	-12	-8	-4	0	-2	2	6	4	2	0
C	-14	-10	-6	-2	-3	0	4	3	6	4
A	-16	-12	-8	-4	-5	-2	2	1	4	8

```
ACTG-ATTCA
||  | || ||
AC-GCAT-CA
```

# match = 8
# gap = 3
# mismatch = 0
Score=8

OR

```
ACTG-ATTCA
||  |  | |||
AC-GCA-TCA
```

# match = 8
# gap = 3
# mismatch = 0
Score=8

Practice

Using the NW method to globally alignment the two sequences

```
Seq X = GCGTC
Seq Y = ACGAC

Match = 3
Mismatch = -1
Gap = -1
```

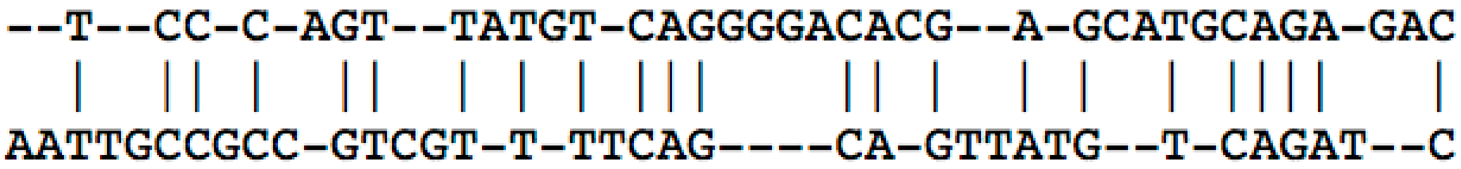
Needleman-Wunsch Global Alignment programs

- BLAST-Global Align Nucleotide Sequences

- Global Alignment: http://www.ebi.ac.uk/Tools/psa/
    - Needle
    - Stretcher

One example:

- Comparing two sequences:
  A: TCCCAGTTATGTCAGGGGACACGAGCATGCAGAGAC
  B: AATTGCCGCCGTCGTTTTCAGCAGTTATGTCAGATC



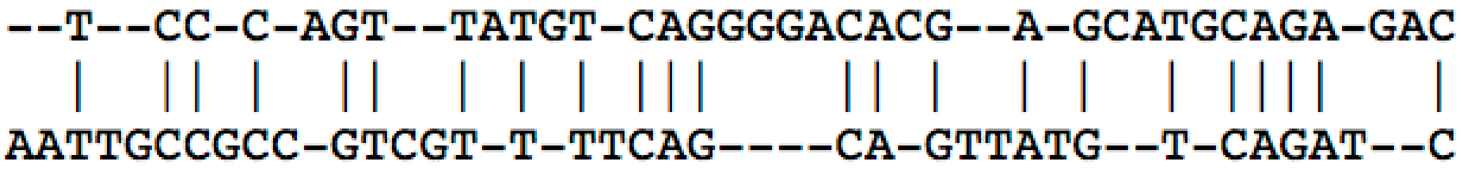
```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || |  | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

One problem:

- Comparing two sequences:

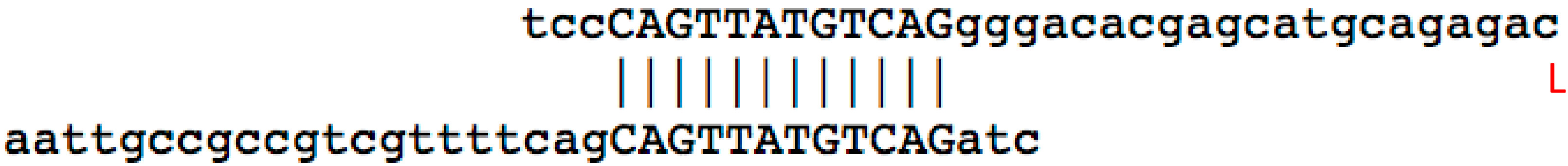
  A: TCCCAGTTATGTCAGGGGACACGAGCATGCAGAGAC

  B: AATTGCCGCCGTCGTTTTCAGCAGTTATGTCAGATC



```
--T--CC-C-AGT--TATGT-CAGGGGACACG--A-GCATGCAGA-GAC
  |  || |  ||  | | | |||    || |  | |  | ||||   |
AATTGCCGCC-GTCGT-T-TTCAG----CA-GTTATG--T-CAGAT--C
```

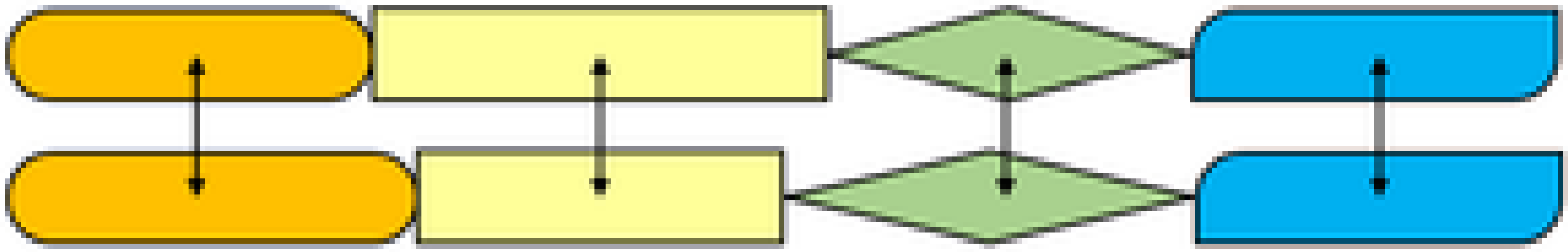
Global alignment



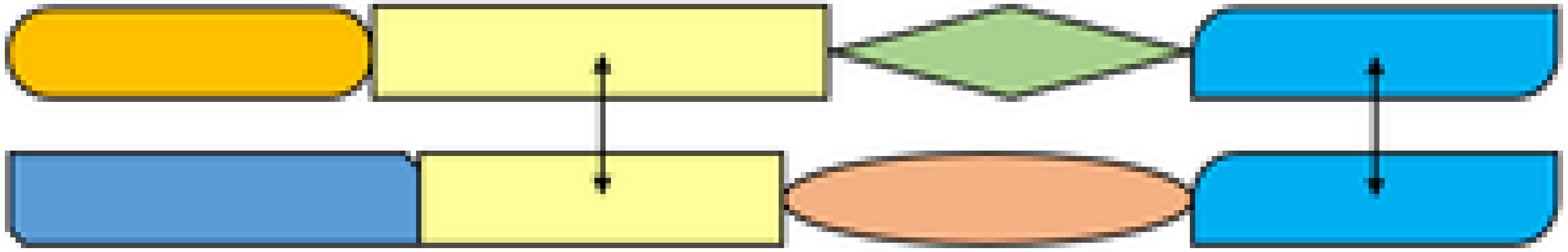
```
                  tccCAGTTATGTCAGgggacacgagcatgcagagac
                     ||||||||||||
aattgccgccgtcgttttcagCAGTTATGTCAGatc
```

Local alignment

Local vs global alignment



Global Alignment



Local Alignment

Complex protein domain architectures

Local alignment vs global alignment

- Global alignment: an attempt is made to align the entire sequence.
  - If two sequences have approximately the same length and are quite similar, they are suitable for the global alignment.

- Local alignment: to find the most similar regions in the two sequences being aligned ("paired subsequences")
  - Regions outside the area of local alignment are removed
  - More than one local alignment could be generated for any two sequences being compared
  - Best for sequences that share some similarity, or for sequences of different lengths

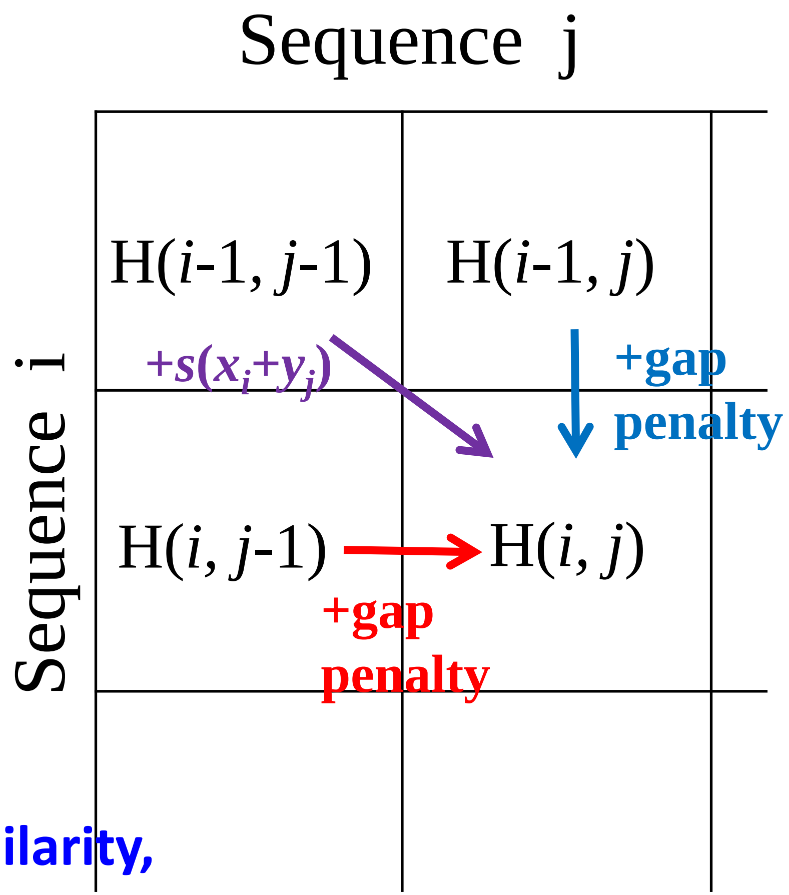
Smith-Waterman algorithm (1981)

For $0 \leq i \leq n$ and $0 \leq j \leq m$, define

$$H_{i,j} = \max\{\, H_{i-1,j-1} + s(a_i,b_j),\; H_{i-1,j} + g;\; H_{i,j-1} + g;\; 0\}$$

$s(a_i,b_j)$: similarity between a_i and b_j

g *or* W_k: Gap penalty



Sequence j

Sequence i

H(i-1, j-1)

H(i-1, j)

+$s(x_i+y_j)$

+gap penalty

H(i, j-1)

H(i, j)

+gap penalty

0 is included to prevent having a negative similarity, indicating no similarity up to a_i and b_j

Smith-Waterman: example

Seq 1:DESIGN
Seq 2:IDEAS

match = 5
gap = -1
mismatch = -1

Add 0 to every cell of row 1 and column 1

		D	E	S	I	G	N
	0	0	0	0	0	0	0
I	0						
D	0						
E	0						
A	0						
S	0						

Smith-Waterman: example

match = 5
gap = -1
mismatch = -1

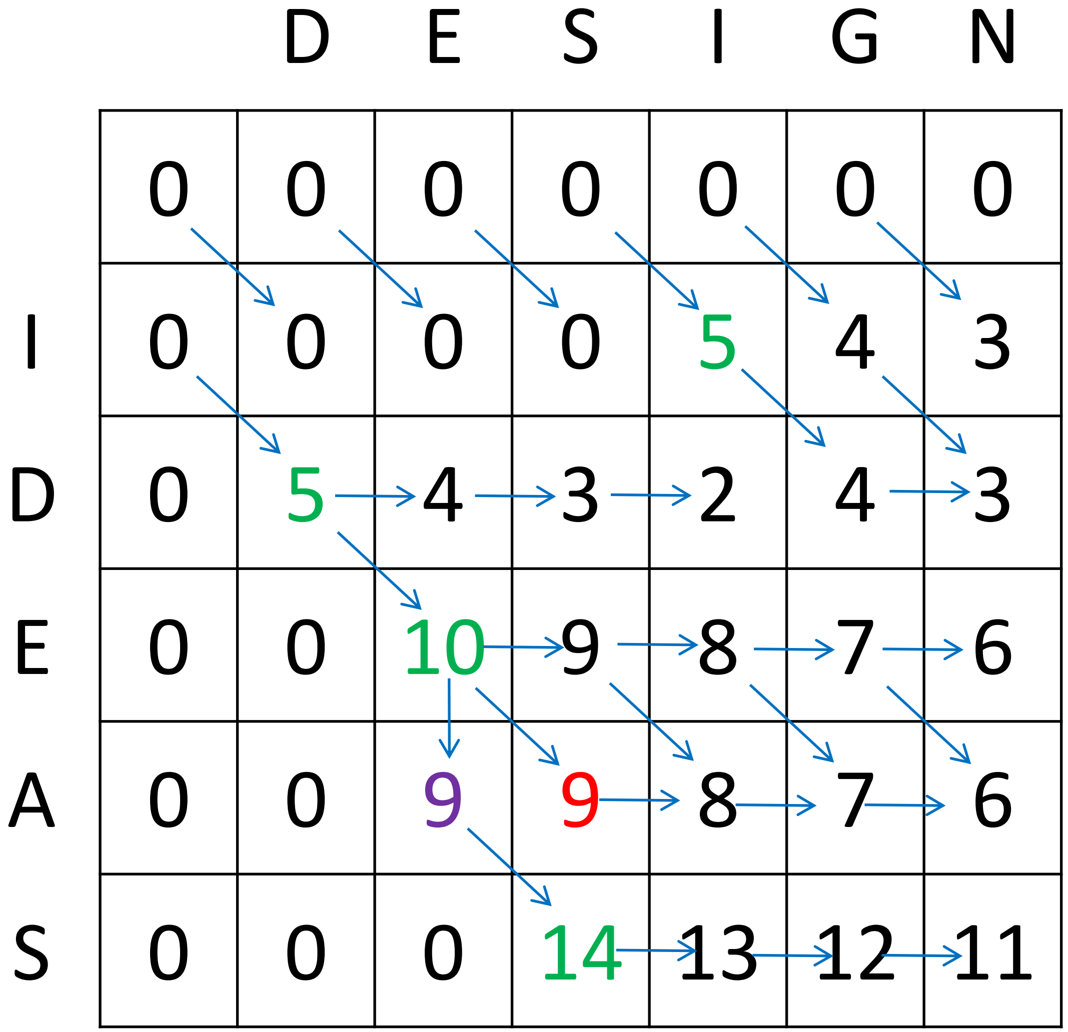
Add 0 if the value below 0;
So, no values below zero in a local alignment scoring matrix

		D	E	S	I	G	N
	0	0	0	0	0	0	0
I	0	0	0	0	5	4	3
D	0						
E	0						
A	0						
S	0						

Smith-Waterman: example

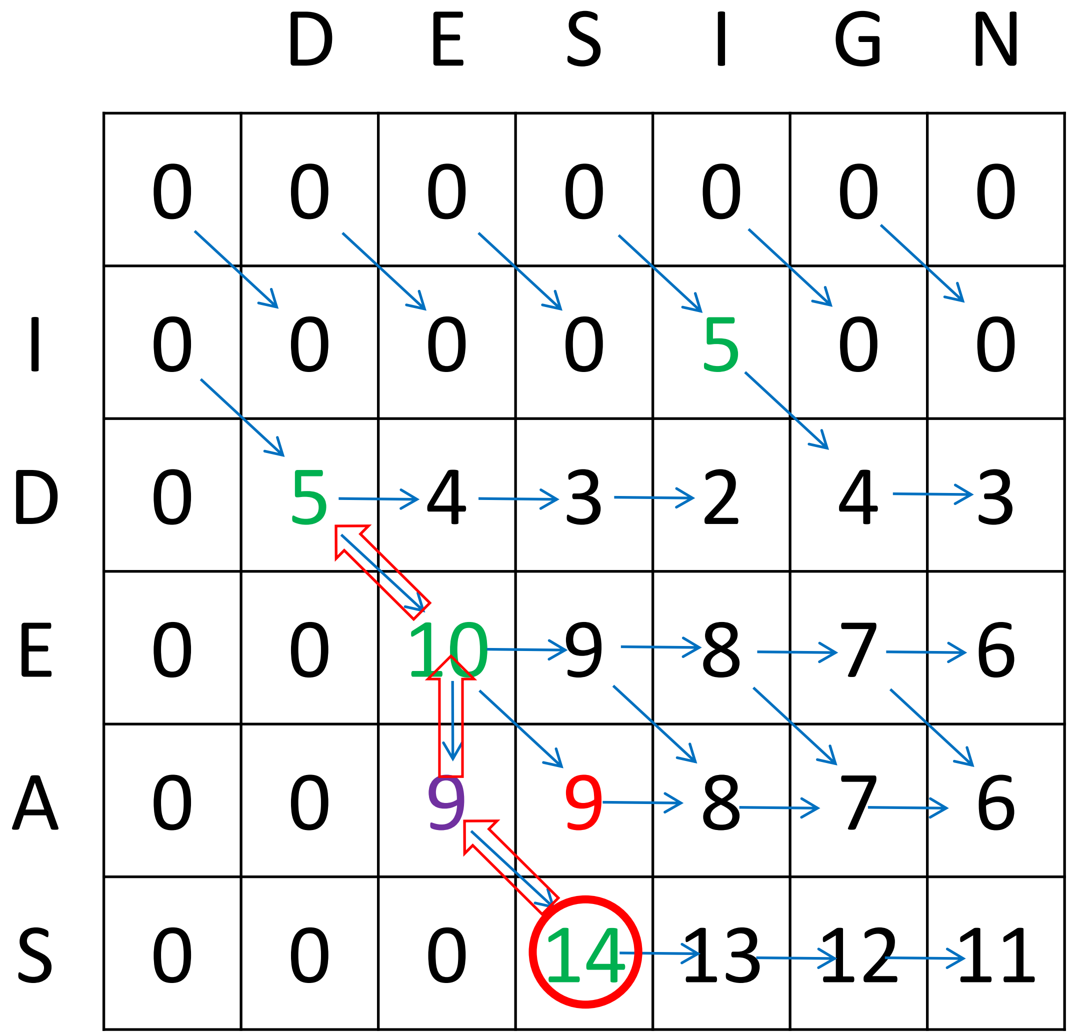
match = 5
gap = -1
mismatch = -1

Add 0 if the value below 0;
So, no values below zero in a local alignment scoring matrix



		D	E	S	I	G	N
	0	0	0	0	0	0	0
I	0	0	0	0	5	4	3
D	0	5	4	3	2	4	3
E	0	0	10	9	8	7	6
A	0	0	9	9	8	7	6
S	0	0	0	14	13	12	11

Smith-Waterman: example



		D	E	S	I	G	N
	0	0	0	0	0	0	0
I	0	0	0	0	5	0	0
D	0	5	4	3	2	4	3
E	0	0	10	9	8	7	6
A	0	0	9	9	8	7	6
S	0	0	0	14	13	12	11

Traceback begins at the highest value (which is also the alignment score)

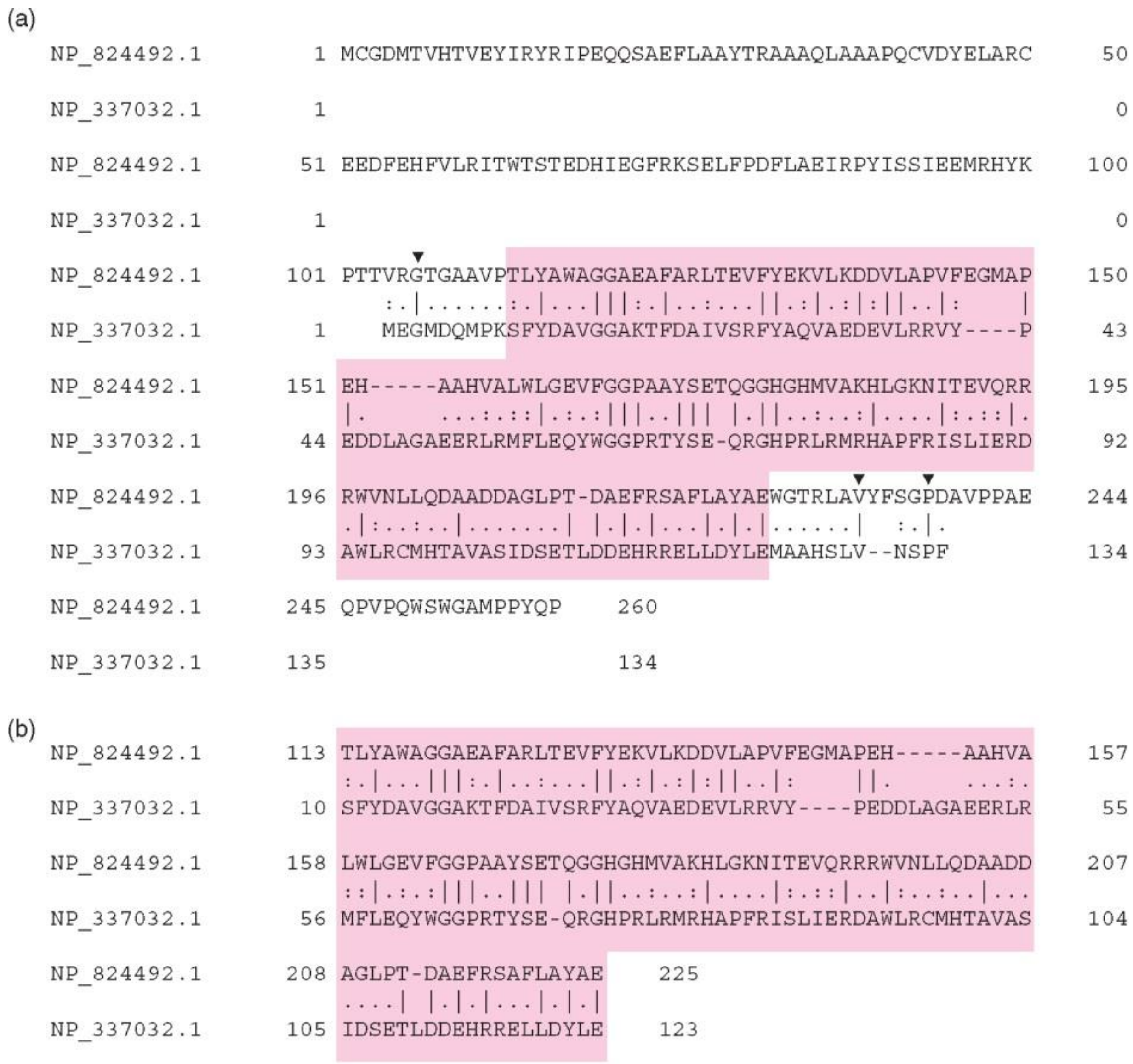
```
1:DESIGN
2:IDEAS
```

```
1:DE-S
  || |
2:DEAS
```

Smith-Waterman algorithm

- To identify the similar segments and produces the corresponding alignment.
- 0 is included to prevent having a negative similarity, indicating no similarity up to Ai and Bj.
- Trace back procedure:
  - Locate the maximum element of H.
  - Trace back along the maximum Hij values, ending with an element of H equal to 0.
- The pair of segments with the next best similarity is found by the second largest element of *H* not associated with the first trace-back.

Global alignment (top) includes matches ignored by local alignment (bottom)



```
(a)
NP_824492.1        1 MCGDMTVHTVEYIRYRIPEQQSAEFLAAYTRAAAQLAAAPQCVDYELARC     50

NP_337032.1        1                                                         0

NP_824492.1       51 EEDFEHFVLRITWTSTEDHIEGFRKSELFPDFLAEIRPYISSIEEMRHYK    100

NP_337032.1        1                                                         0
                          ▼
NP_824492.1      101 PTTVRGTGAAVPTLYAWAGGAEAFARLTEVFYEKVLKDDVLAPVFEGMAP    150
                        :.|.......:.|...|||:.|..:...||.:|.:|:||..|:    |
NP_337032.1        1    MEGMDQMPKSFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----P     43

NP_824492.1      151 EH-----AAHVALWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRR    195
                     |.     ...:.::|.:.:|||..||| |.||..:..:|....|:.::|.
NP_337032.1       44 EDDLAGAEERLRMFLEQYWGGPRTYSE-QRGHPRLRMRHAPFRISLIERD     92
                                                          ▼    ▼
NP_824492.1      196 RWVNLLQDAADDAGLPT-DAEFRSAFLAYAEWGTRLAVYFSGPDAVPPAE    244
                     .|:..:..|.......| |.|.|...|.|.|......|  :.|.
NP_337032.1       93 AWLRCMHTAVASIDSETLDDEHRRELLDYLEMAAHSLV--NSPF          134

NP_824492.1      245 QPVPQWSWGAMPPYQP    260

NP_337032.1      135                     134

(b)
NP_824492.1      113 TLYAWAGGAEAFARLTEVFYEKVLKDDVLAPVFEGMAPEH-----AAHVA    157
                     :.|...|||:.|..:...||.:|.:|:||..|:    ||.     ...:.
NP_337032.1       10 SFYDAVGGAKTFDAIVSRFYAQVAEDEVLRRVY----PEDDLAGAEERLR     55

NP_824492.1      158 LWLGEVFGGPAAYSETQGGHGHMVAKHLGKNITEVQRRRWVNLLQDAADD    207
                     ::|.:.:|||..||| |.||..:..:|....|:.::|..|:..:..|...
NP_337032.1       56 MFLEQYWGGPRTYSE-QRGHPRLRMRHAPFRISLIERDAWLRCMHTAVAS    104

NP_824492.1      208 AGLPT-DAEFRSAFLAYAE    225
                     ....| |.|.|...|.|.|
NP_337032.1      105 IDSETLDDEHRRELLDYLE    123
```

Global:
15% identity

Local:
30% identity

NP_824492, NP_337032

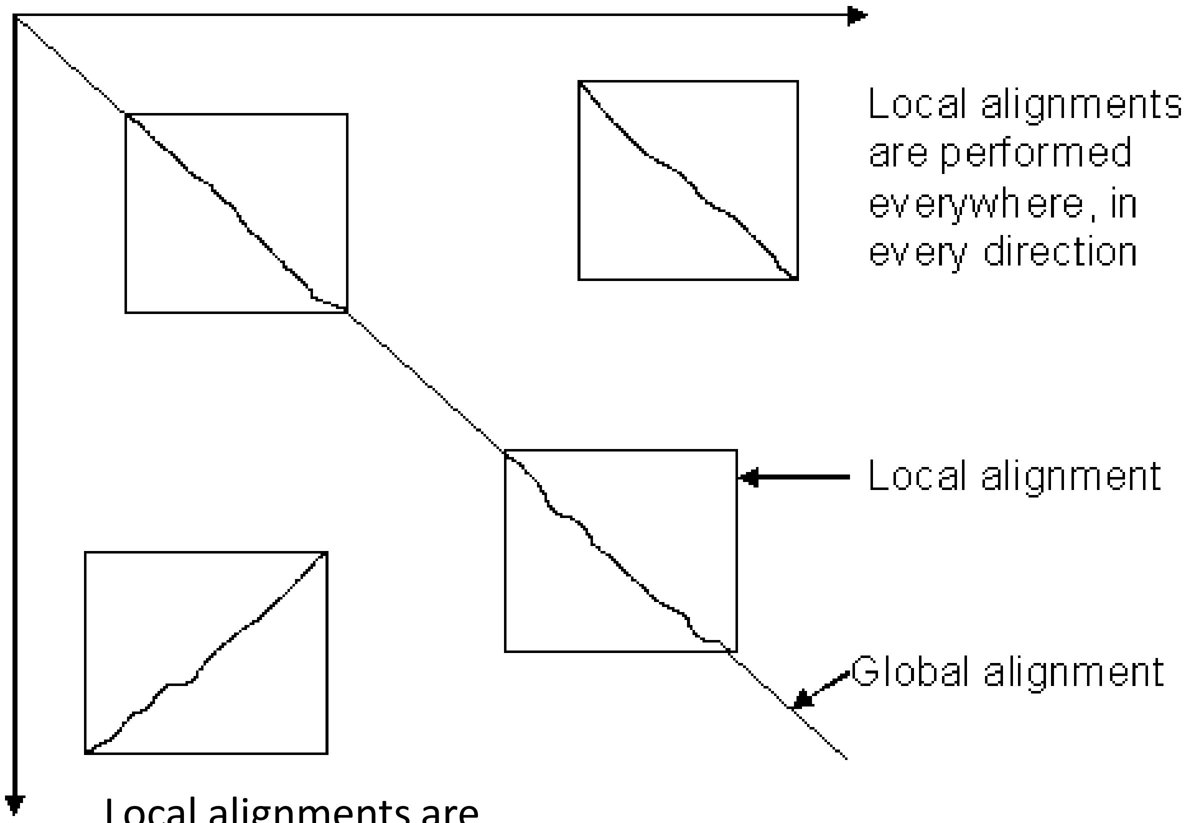
Pairwise alignment tools

- http://www.ebi.ac.uk/Tools/psa/

- Global Alignment
  - Needle
  - Stretcher

- Local Alignment
  - Water
  - Matcher

Global vs local alignment


Local alignments are performed everywhere, in every direction
Local alignment
Global alignment

Local alignments are performed everywhere possible along two sequences.