



**Mansoura University**  
**Faculty of Computers and Information**  
**Department of Computer Science**  
**First Semester: 2020-2021**



**[MED121] Bioinformatics: Sequence Alignment Algorithms**  
**Grade: Third Year (Medical Informatics Program)**

**Sara El-Metwally, Ph.D.**  
**Faculty of Computers and Information,**  
**Mansoura University,**  
**Egypt.**

# AGENDA

- Sequence Alignment
- Scoring Metrics
- Local Vs. Global Alignments
- Pairwise Vs. Multiple Alignments
- Needleman-Wunsch algorithm
- Smith-Waterman algorithm

# SEQUENCING TECHNOLOGIES



<https://ngisweden.scilifelab.se/technologies/pacific-biosciences/pacbio-sequel/>



<https://www.biocompare.com/23967-Ne>



<https://www.technologyreview.com/2016/02/24/8993/with-patent-suit-illumina-looks-to-tame-emerging-british-rival-oxford-nanopore/>

# NOTES

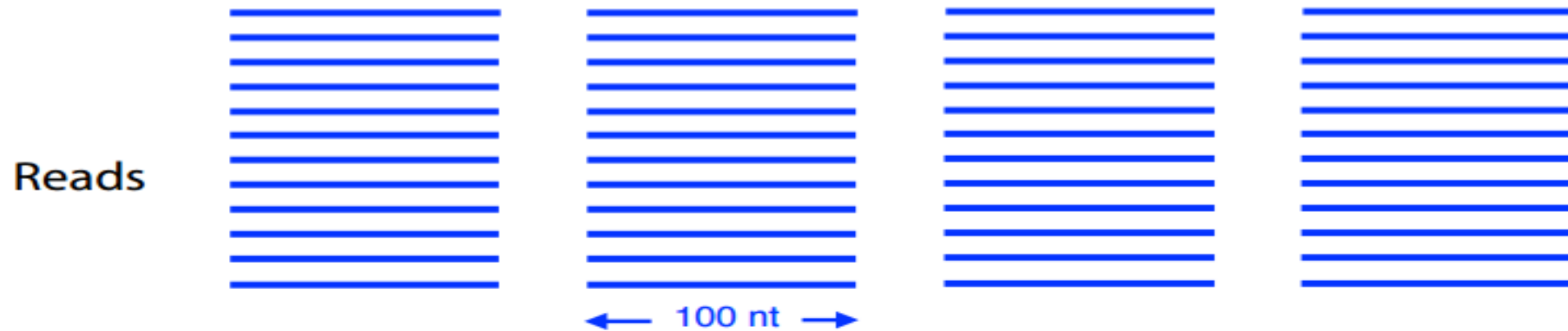
Reads

GTATGCACGCGATAG	TATGTCGCGAGTATCT	CACCCTATGTCGCAG	GAGACGCTGGAGCCG
TAGCATTGCGAGACG	GGTATGCACGCGATA	TGGAGCCGGAGCACC	CGCTGGAGCCGGAGC
TGTCTTTGATTCTG	CGCGATAGCATTGCG	GCATTGCGAGACGCT	CCTATGTCGCGAGTAT
GACGCTGGAGCCGGA	GCACCCTATGTCGCA	GTATCTGTCTTTGAT	CCTCATCCTATTATT
TATCGCACCTACGTT	CAATATTTCGATCATG	GATCACAGGTCTATC	ACCCTATTAACCACT
CACGGGAGCTCTCCA	TGCATTTGGTATTTT	CGTCTGGGGGGGTATG	CACGCGATAGCATTG
GTATGCACGCGATAG	ACCTACGTTCAATAT	TATTTATCGCACCTA	CCACTCACGGGAGCT
GCGAGACGCTGGAGC	CTATCACCCCTATTAA	CTGTCTTTGATTCT	ACTCACGGGAGCTCT
CCTACGTTCAATATT	GCACCTACGTTCAAT	GTCTGGGGGGGTATGC	AGCCGGAGCACCTA
GACGCTGGAGCCGGA	GCACCCTATGTCGCA	GTATCTGTCTTTGAT	CCTCATCCTATTATT
TATCGCACCTACGTT	CAATATTTCGATCATG	GATCACAGGTCTATC	ACCCTATTAACCACT
CACGGGAGCTCTCCA	TGCATTTGGTATTTT	CGTCTGGGGGGGTATG	CACGCGATAGCATTG

Your genome

**CGTCTGGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCTATGTCGCGAGTATCTGTCTTTGATTCTG**

# NOTES



Your genome



# What is Next?

## Sequence Analysis

### Alignment



### Assembly





# Sequence alignment

## Sequencing Reads

TATGTCGCAGTATCTGCGCAGTATCTG  
TATGTCGCAGTATCTT  
TATGTCGCAGTATCTG  
TATGTCGCAGTATCTG  
GTCGCAGTATCTGTCT  
CCGGACACCCCTATATATGTCGCAGTATCTT  
ACACCCTATGTCGCA  
ACACCCTATGTCGCA  
TATGTCGCAGTATCTG  
ACACCCTATGTCGCA  
TATGTCGCAGTATCTG  
CCGGACACCCCTATAT  
GTCGCAGTATCTGTC  
TGTGTCGCAGTATCTGTC



## Reference Genome

GATCACAGGTCTATCACCCCTATTAACCACTCACGGGAGCTCTCCATGCATTTGGTATTTT  
CGTCTGGGGGGTATGCACGCGATAGCATTGCGAGACGCTGGAGCCGGAGCACCCCTATGTC  
GCAGTATCTGTCTTTGATTCCCTGCCTCATCCTATTATTTATCGCACCTACGTTCAATATT  
ACAGGCGAACATACTTACTAAAGTGTGTTAATTAATTAATGCTTGTAGGACATAATAATA  
ACAATTGAATGTCTGCACAGCCACTTTCACACAGACATCATAACAAAAAATTTCCACCA  
AACCCCCCTCCCCGCTTCTGGCCACAGCACTTAAACACATCTCTGCCAAACCCCAAAA  
ACAAAGAACCCTAACACCAGCCTAACCCAGATTTCAAATTTTATCTTTTGGCGGTATGCAC  
TTTTAACAGTCACCCCCCACTAACACATTATTTCCCTCCCACTCCCACTACTACTAAT  
CTCATCAATACAACCCCCGCCCCATCCTACCCAGCACACACACACCCGCTGCTAACCCATA  
CCCCGAACCAACCAAAACCCCAAGACACCCCCACAGTTTATGTAGCTTACCTCCTCAA  
GCAATACACTGACCCGCTCAAACCTCCTGGATTTTGGATCCACCCAGCGCCTTGGCCTAAA  
CTAGCCTTTCTATTAGCTCTTAGTAAGATTACACATGCAAGCATCCCGTTCCAGTGAGT  
TCACCCTCTAAATCACCACGATCAAAAGGAACAAGCATCAAGCACGCAGCAATGCAGCTC  
AAAACGCTTAGCCTAGCCACACCCCCACGGGAAACAGCAGTGATTAACCTTTAGCAATAA  
ACGAAAGTTTAACTAAGCTATACTAACCCAGGGTTGGTCAATTTCTGCGCCAGCCACCGC  
GGTCACACGATTAACCCAGTCAATAGAAAGCCGGCGTAAAGAGTGTTTTAGATCACCCCC  
TCCCCAATAAAGCTAAAACTCACCTGAGTTGTAAAAAACTCCAGTTGACACAAAATAGAC  
TACGAAAGTGCTTTAACATATCTGAACACACAATAGCTAAGACCCAACTGGGATTAGA  
TACCCCACTATGCTTAGCCCTAAACCTCAACAGTTAAATCAACAAAACTGCTCGCCAGAA  
CACTACGAGCCACAGCTTAAAACTCAAAGGACCTGGCGGTGCTTCATATCCCTCTAGAGG  
AGCCTGTTCTGTAATCGATAAAACCCCGATCAACCTCACCACTCTTGCTCAGCCTATATA

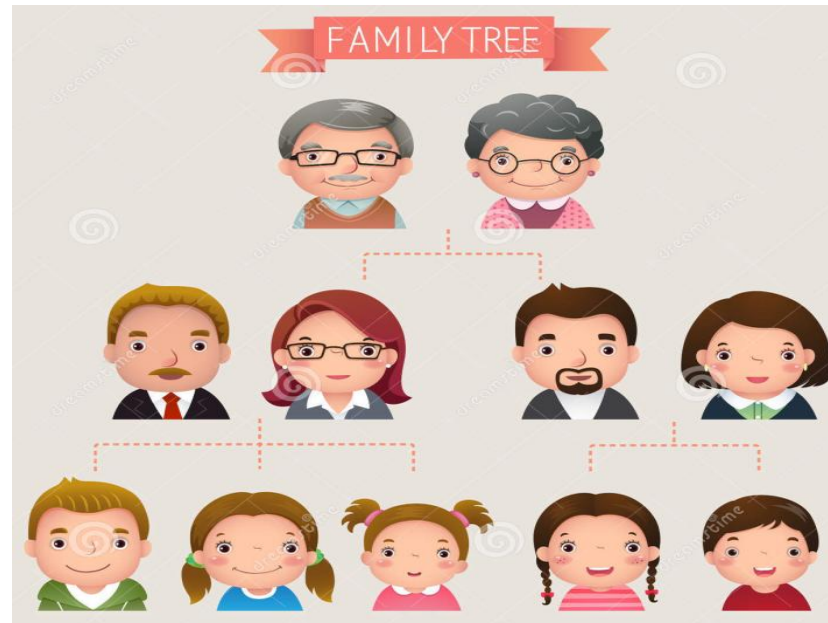
# Sequence alignment

- **Sequence alignment** or sequence comparison lies at heart of the bioinformatics, which describes the way of arrangement of DNA/RNA or protein sequences, in order to identify the regions of similarity among them.
- **Causes for sequence (dis) similarity:**
  - Mutation or substitution (i.e. ACT → AGT)
  - Insertion (i.e. ACT → ACGT)
  - Deletion (i.e. ACT → AT)
  - Indels (Insertion/Deletion)

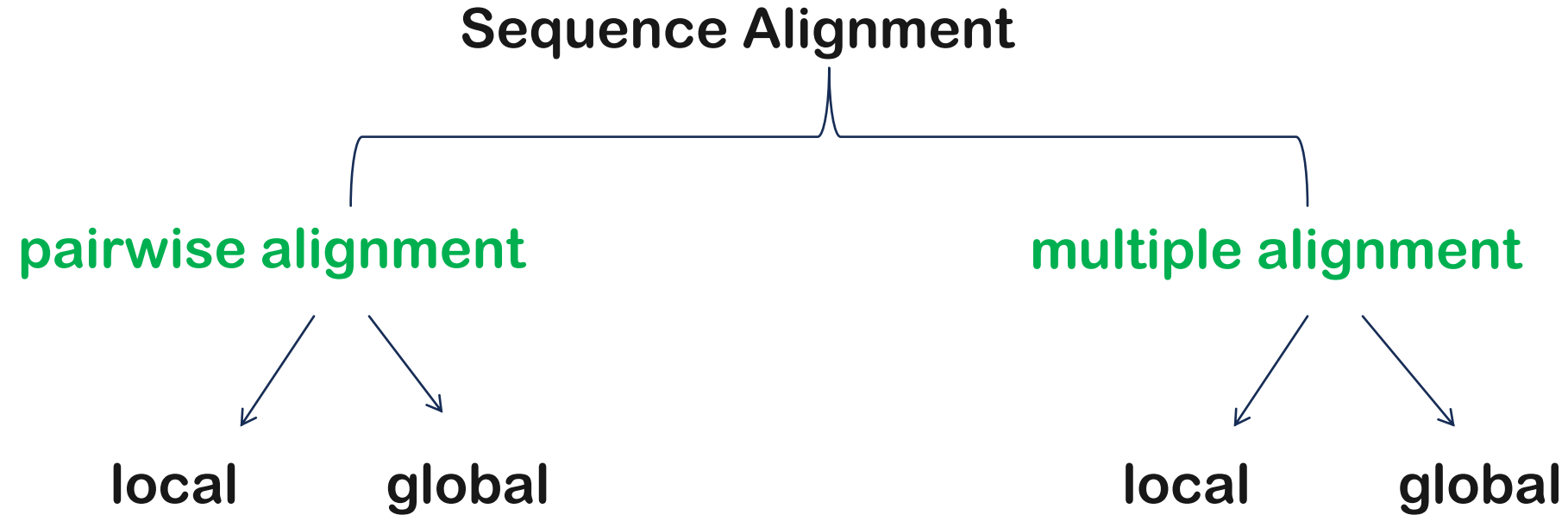


# Sequence alignment (Why)

- Predicting functions.
- Database searching.
- Gene finding.
- Sequence evolutionary.



# Sequence alignment



# Sequence alignment

- Try to align these two sequences: AGTCTT, ATCT?

AGTCTT

A-TCT-

- ▶ Is there a better alignment? How can we compare the “goodness” of two alignments?

# scoring an alignment

- Simple scoring scheme is **Identity score** which is defined by the following equation:

$$\text{id}(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases}$$

AGTCTT

A-TCT-

1+0+1+1+1+0 = 4

alignment score= 4

(4/6)%= 67% identical

# scoring matrix

- Simple scoring scheme is **Identity score** which is defined by the following equation:

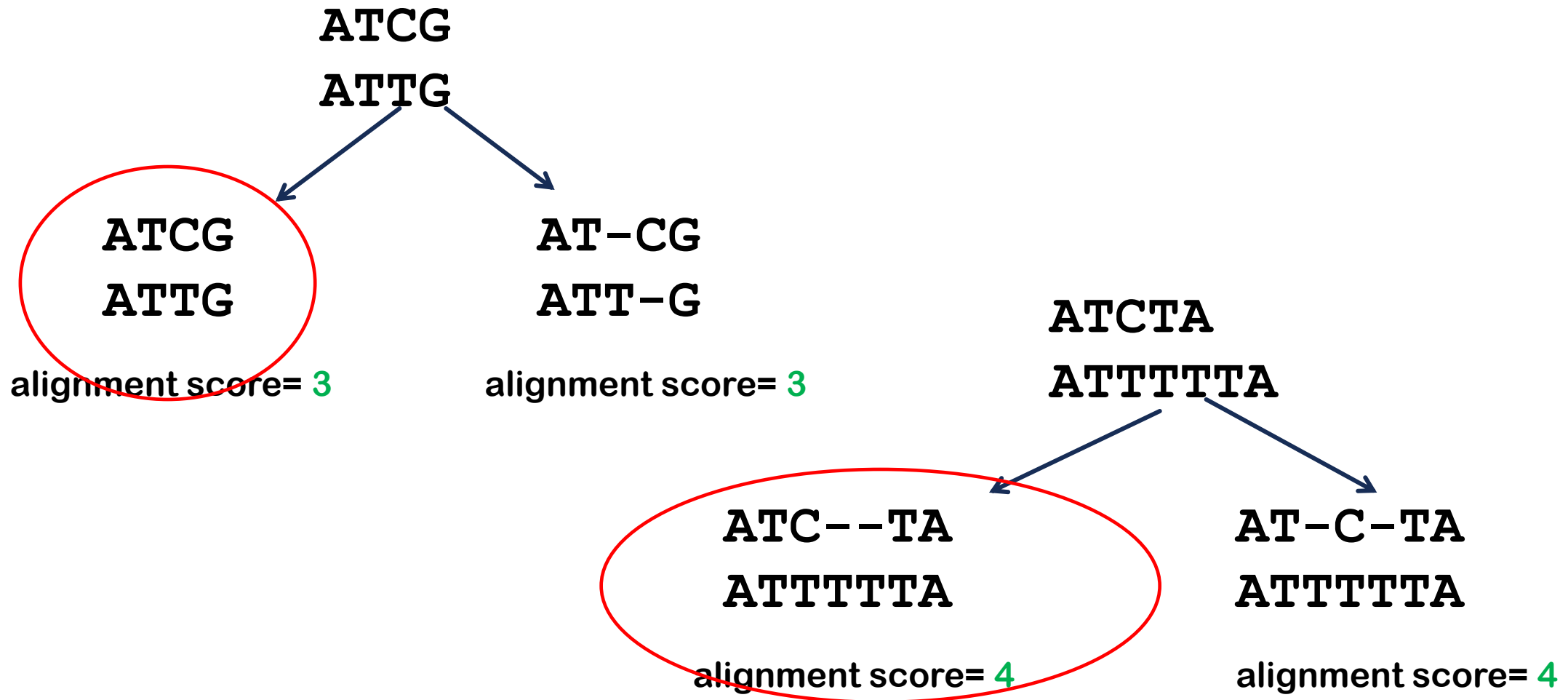
$$\text{id}(x, y) = \begin{cases} 1 & \text{if } x = y \\ 0 & \text{if } x \neq y \end{cases} \quad \text{as a matrix} \quad \longrightarrow$$

	x	y
x	1	0
y	0	1

If amino acids are used? RNA seq? DNA seq?

	A	C	G	T
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

# Identity Score (problems)



# substitution matrix

	A	C	G	T	
A	2	-1	1	-1	Purines
C	-1	2	-1	1	
G	1	-1	2	-1	Pyrimidine
T	-1	1	-1	2	

Purines

Pyrimidine



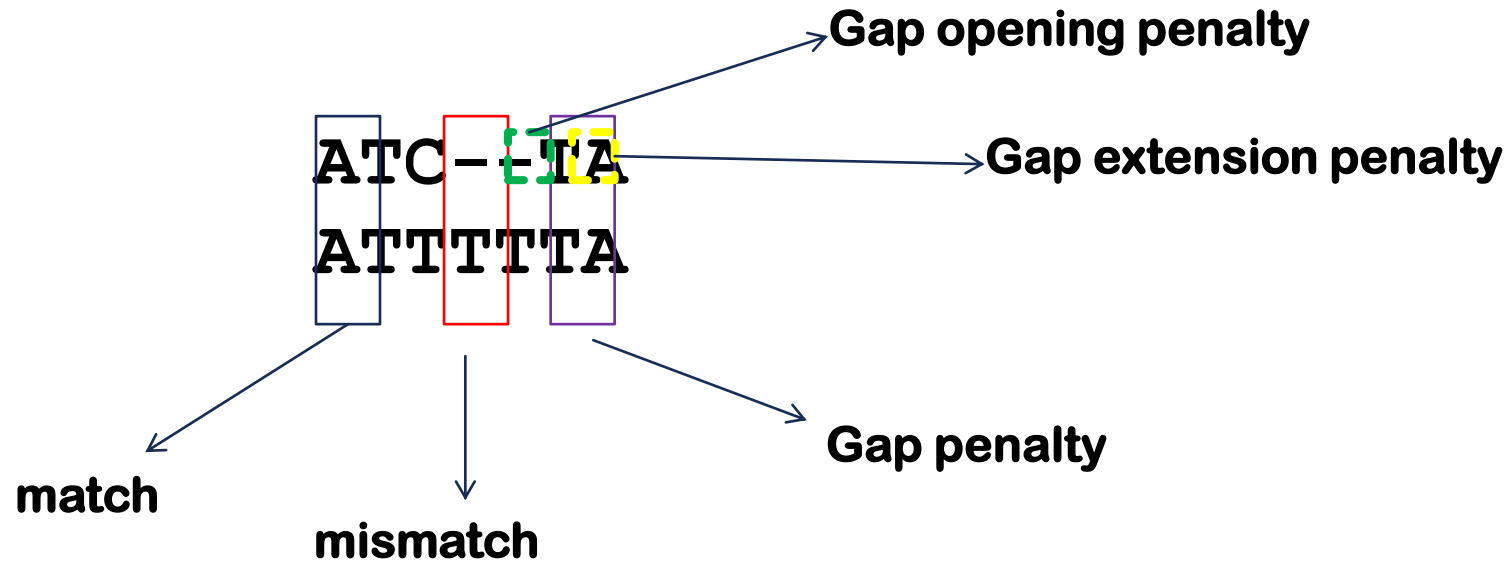
# substitution matrix

- A protein substitution matrix can be based on any property of amino acids: size, polarity, charge, etc.
- In practice the most important are evolutionary substitution matrices:
  - **PAM** ("point accepted mutation") family PAM250, PAM120, etc.
  - **BLOSUM** ("Blocks substitution matrix") family BLOSUM62, BLOSUM50, etc.
  - The substitution scores of both PAM and BLOSUM matrices are derived from the analysis of known alignments of closely related proteins.

# substitution matrix ( **blosum62** )

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	4	-1	-2	-2	0	-1	-1	0	-2	-1	-1	-1	-1	-2	-1	1	0	-3	-2	0	-2	-1	0	-4
R	-1	5	0	-2	-3	1	0	-2	0	-3	-2	2	-1	-3	-2	-1	-1	-3	-2	-3	-1	0	-1	-4
N	-2	0	6	1	-3	0	0	0	1	-3	-3	0	-2	-3	-2	1	0	-4	-2	-3	3	0	-1	-4
D	-2	-2	1	6	-3	0	2	-1	-1	-3	-4	-1	-3	-3	-1	0	-1	-4	-3	-3	4	1	-1	-4
C	0	-3	-3	-3	9	-3	-4	-3	-3	-1	-1	-3	-1	-2	-3	-1	-1	-2	-2	-1	-3	-3	-2	-4
Q	-1	1	0	0	-3	5	2	-2	0	-3	-2	1	0	-3	-1	0	-1	-2	-1	-2	0	3	-1	-4
E	-1	0	0	2	-4	2	5	-2	0	-3	-3	1	-2	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
G	0	-2	0	-1	-3	-2	-2	6	-2	-4	-4	-2	-3	-3	-2	0	-2	-2	-3	-3	-1	-2	-1	-4
H	-2	0	1	-1	-3	0	0	-2	8	-3	-3	-1	-2	-1	-2	-1	-2	-2	2	-3	0	0	-1	-4
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4	2	-3	1	0	-3	-2	-1	-3	-1	3	-3	-3	-1	-4
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4	-2	2	0	-3	-2	-1	-2	-1	1	-4	-3	-1	-4
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5	-1	-3	-1	0	-1	-3	-2	-2	0	1	-1	-4
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5	0	-2	-1	-1	-1	-1	1	-3	-1	-1	-4
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6	-4	-2	-2	1	3	-1	-3	-3	-1	-4
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7	-1	-1	-4	-3	-2	-2	-1	-2	-4
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4	1	-3	-2	-2	0	0	0	-4
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5	-2	-2	0	-1	-1	0	-4
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11	2	-3	-4	-3	-2	-4
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	-1	-3	-2	-1	-4
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4	-3	-2	-1	-4
B	-2	-1	3	4	-3	0	1	-1	0	-3	-4	0	-3	-3	-2	0	-1	-4	-3	-3	4	1	-1	-4
Z	-1	0	0	1	-3	3	4	-2	0	-3	-3	1	-1	-3	-1	0	-1	-3	-2	-2	1	4	-1	-4
X	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	-1	-1	-1	-1	-2	0	0	-2	-1	-1	-1	-1	-1	-4
*	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	-4	1

# scoring scheme updated!



# Gap Scoring Scheme (**Linear**)

ATC---TA  
ATTTTTA

match(1), mismatch(-1), gap(-1)

gap penalty \* gap length (L)  
 $= -1 * 3 = -3$

Total similarity score:  
 $= -3 + 4 - 1 = 0$

# Gap Scoring Scheme (**Affine**)

ATC---TA  
ATTTTTTA

match(1), mismatch(-1),  
gap opening (-2), gap extension (- 4)

gap opening penalty + (gap length (L)\* gap extension penalty)  
 $= -2 + (2 * -4) = -2 - 8 = -10$

Total similarity score:  
 $= 4 - 1 - 10 = -7$

# Gap Scoring Scheme (**Affine : Q**)

GAATTCCGTTA

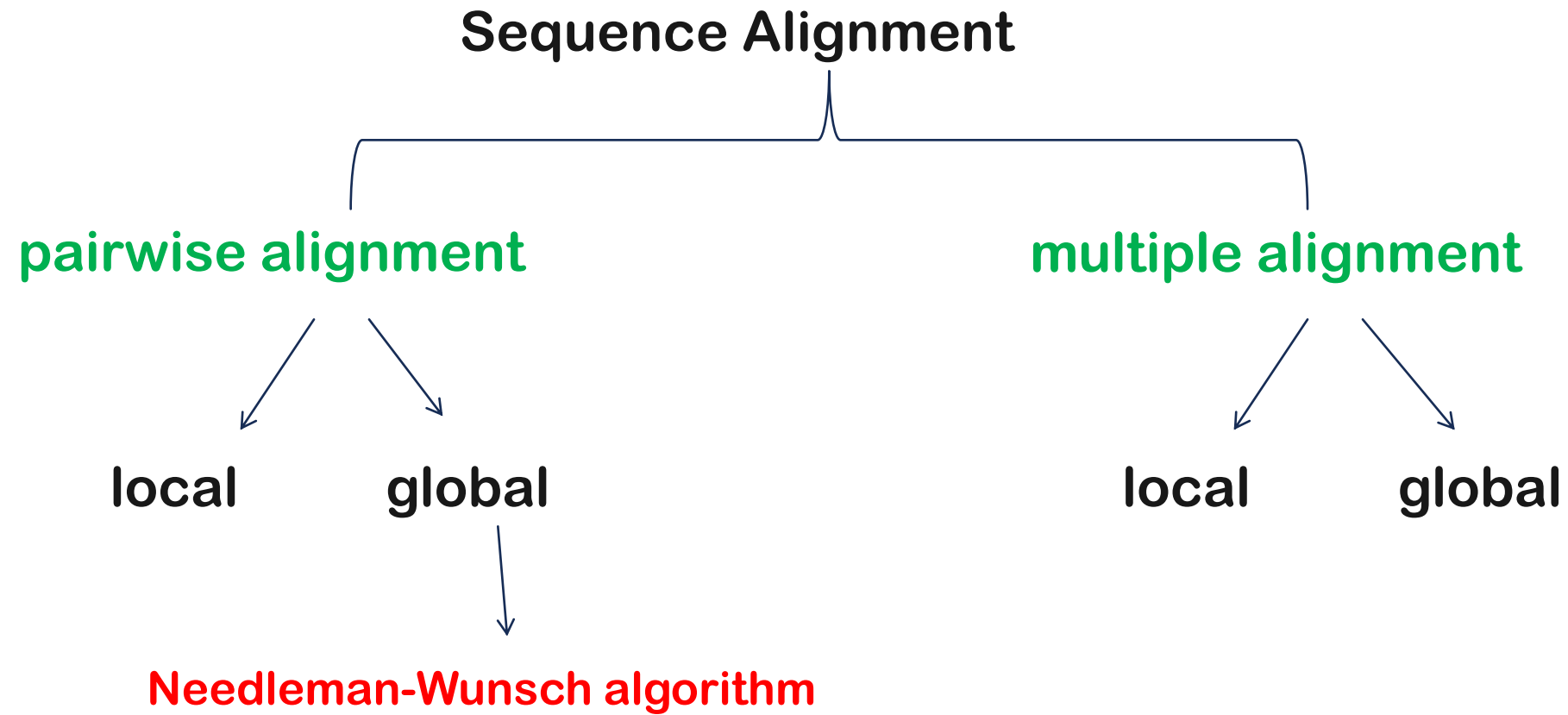
GGAT-C-G--A

match(2), mismatch(-1),  
gap opening (-2), gap extension (-3)

Total similarity score:

$$2-1+2+2-2+2+2-2-3+2=2$$

# Sequence alignment





# Needleman Wunch algorithm

- It is global pairwise sequence alignment algorithm.
- You need to know:
  - Sequences that you will align.
  - Scoring scheme that you will use.

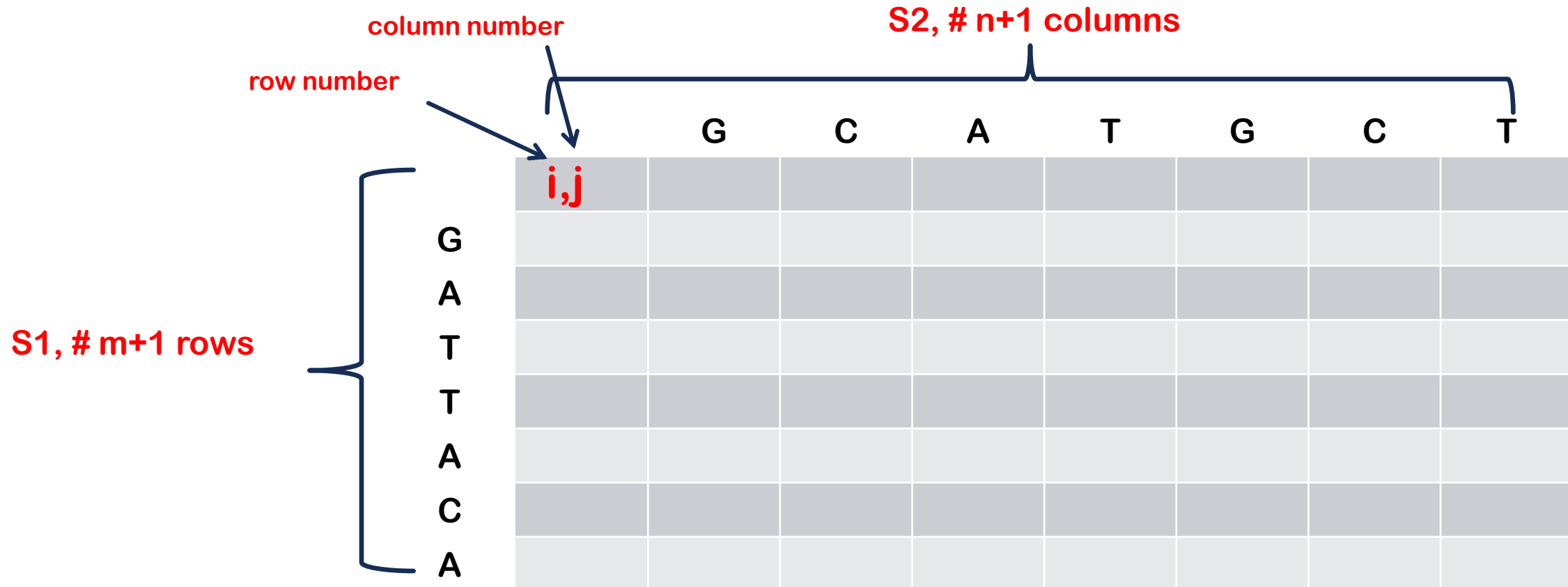
# Needleman Wunch algorithm

- **Example:** Suppose you have two sequences:
  - S1=GATTACA
  - S2=GCATGCT
  - Match (+1), mismatch(-1), Linear Gap Model (-1)
    - Length of S1=m=7chars.
    - Length of S2=n=7chars.
    - Create a matrix of rows (m+1) and columns (n+1)

# Needleman Wunch algorithm

S1=GATTACA  
S2=GCATGCT

Match (+1), mismatch(-1), Linear Gap Model (-1)



# Needleman Wunch algorithm

S1=GATTACA

S2=GCATGCT

Match (+1), mismatch(-1), Linear Gap Model (-1)

Step 1: Fill all (i,0) with  $i \times \text{Gap score}$

		G	C	A	T	G	C	T	
		0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	1,0	-1							
A	2,0	-2							
T	3,0	-3							
T	4,0	-4							
A	5,0	-5							
C	6,0	-6							
A	7,0	-7							

# Needleman Wunch algorithm

S1=GATTACA

S2=GCATGCT

Match (+1), mismatch(-1), Linear Gap Model (-1)

Step 2: Fill all (0,j) with j \* Gap score

		G	C	A	T	G	C	T
	0,0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0 -1							
A	2,0 -2							
T	3,0 -3							
T	4,0 -4							
A	5,0 -5							
C	6,0 -6							
A	7,0 -7							

# Needleman Wunch algorithm

S1=GATTACA

S2=GCATGCT

Match (+1), mismatch(-1), Linear Gap Model (-1)

Step 3: Fill (0,0) with 0

		G	C	A	T	G	C	T	
G A T T A C A	0,0	0	-1	-2	-3	-4	-5	-6	-7
	1,0	-1							
	2,0	-2							
	3,0	-3							
	4,0	-4							
	5,0	-5							
	6,0	-6							
	7,0	-7							







# Needleman Wunch algorithm

$$cell(1,1) = \max \left\{ \begin{array}{l} (0,0) + score(G,G) \\ (1,0) + -1 \\ (0,1) + -1 \end{array} \right\}$$

$$cell(1,1) = \max \left\{ \begin{array}{l} 0 + 1 \\ -1 + -1 \\ -1 + -1 \end{array} \right\} = 1$$

		G	C	A	T	G	C	T	
	0,0	0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0 -1	1,1 1							
A	2,0 -2								
T	3,0 -3								
T	4,0 -4								
A	5,0 -5								
C	6,0 -6								
A	7,0 -7								

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \begin{cases} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{cases}$$

$$\text{cell } (1,2) = \max \begin{cases} (0,1) + \text{score}(G, C) \\ (1,1) + -1 \\ (0,2) + -1 \end{cases} \quad \text{cell } (1,2) = \max \begin{cases} -1 + -1 \\ 1 + -1 \\ -2 + -1 \end{cases} = 0$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0 0	-1 1,1	-2 1,2	-3	-4	-5	-6	-7
A	-1							
T	-2							
T	-3							
A	-4							
C	-5							
A	-6							
C	-7							

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(\text{ch1}, \text{ch2}) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

$$\text{cell } (1,2) = \max \left\{ \begin{array}{l} -2 + -1 \\ 0 + -1 \\ -3 + -1 \end{array} \right\} = -1$$

		G	C	A	T	G	C	T	
	0,0	0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0	-1	1,1 1	1,2 0	-1				
A	2,0	-2							
T	3,0	-3							
T	4,0	-4							
A	5,0	-5							
C	6,0	-6							
A	7,0	-7							

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \begin{cases} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{cases}$$

		G	C	A	T	G	C	T	
	0,0	0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0	-1	1,1 1	1,2 0	-1	-2	-3	-4	-5
A	2,0	-2							
T	3,0	-3							
T	4,0	-4							
A	5,0	-5							
C	6,0	-6							
A	7,0	-7							

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i, j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(\text{ch1}, \text{ch2}) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3							
A	-4							
C	-5							
C	-6							
A	-7							

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1						
A	-4	-2						
C	-5	-3						
C	-6	-4						
A	-7	-5						

Match (+1), mismatch(-1), Linear Gap Model (-1)



# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(\text{ch1}, \text{ch2}) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T	
	0,0	0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0 -1	1,1 1	1,2 0	-1	-2	3	-4	-5	
A	2,0 -2	0	0	1	0	-1	-2	-3	
T	3,0 -3	-1	-1						
T	4,0 -4	-2	-2						
A	5,0 -5	-3	-3						
C	6,0 -6	-4	-2						
A	7,0 -7	-5	-3						

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2					
C	-5	-3	-3					
C	-6	-4	-2					
A	-7	-5	-3					

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1				
C	-5	-3	-3	-1				
C	-6	-4	-2	-2				
A	-7	-5	-3	-1				

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(\text{ch1}, \text{ch2}) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

		G	C	A	T	G	C	T	
	0,0	0	0,1 -1	0,2 -2	0,3 -3	0,4 -4	0,5 -5	0,6 -6	0,7 -7
G	1,0 -1	1,1 1	1,2 0	-1	-2	3	-4	-5	
A	2,0 -2	0	0	1	0	-1	-2	-3	
T	3,0 -3	-1	-1	0	2	1	0	-1	
T	4,0 -4	-2	-2	-1	1				
A	5,0 -5	-3	-3	-1	0				
C	6,0 -6	-4	-2	-2	-1				
A	7,0 -7	-5	-3	-1	-2				

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \begin{cases} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{cases}$$

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	-3	-4	-5
T	-2	0	-1	0	-1	-2	-3	-4
T	-3	-1	-2	-1	0	-1	-2	-3
A	-4	-2	-3	-2	-1	0	-1	-2
C	-5	-3	-4	-3	-2	-1	0	-1
A	-6	-4	-5	-4	-3	-2	-1	0
A	-7	-5	-6	-5	-4	-3	-2	-1

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + \text{score}(ch1, ch2) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{array} \right\}$$

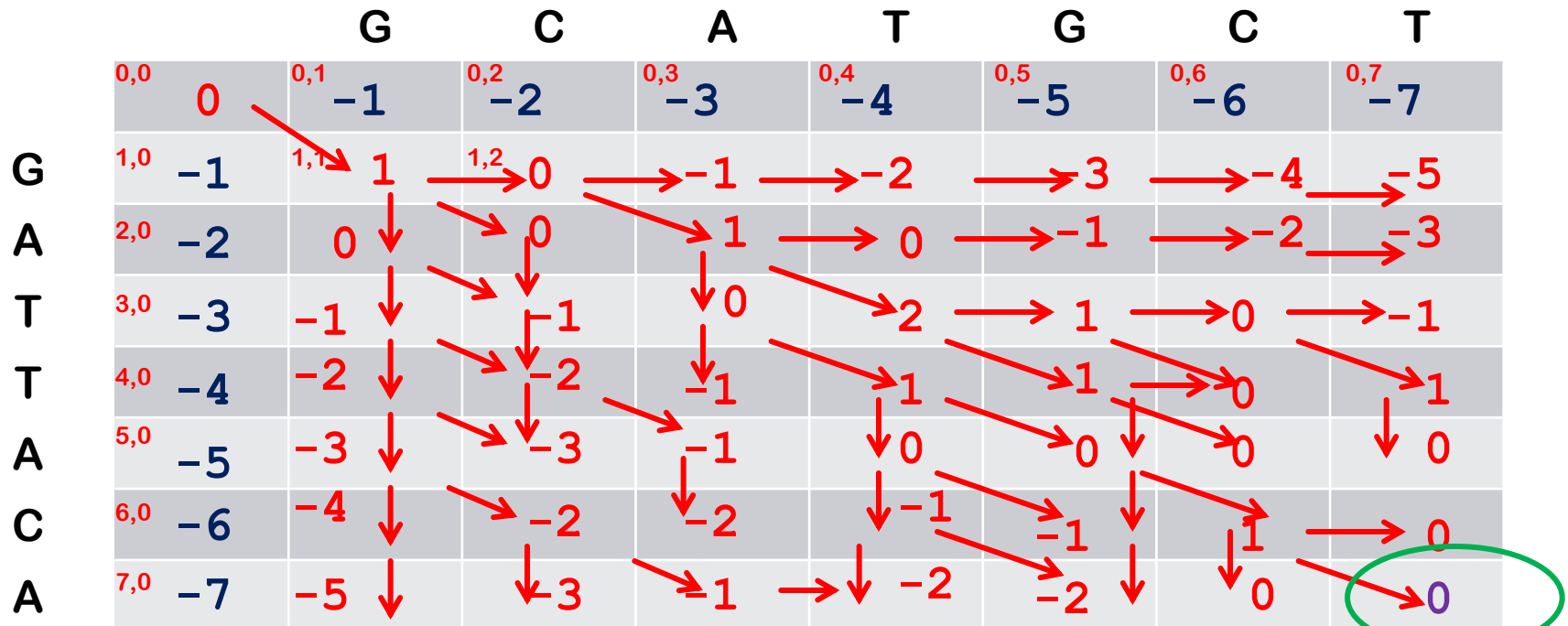
		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0

Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

$$\text{for cell } (i,j) = \max \begin{cases} (i-1, j-1) + \text{score}(\text{ch1}, \text{ch2}) \\ (i, j-1) + \text{gap score} \\ (i-1, j) + \text{gap score} \end{cases}$$

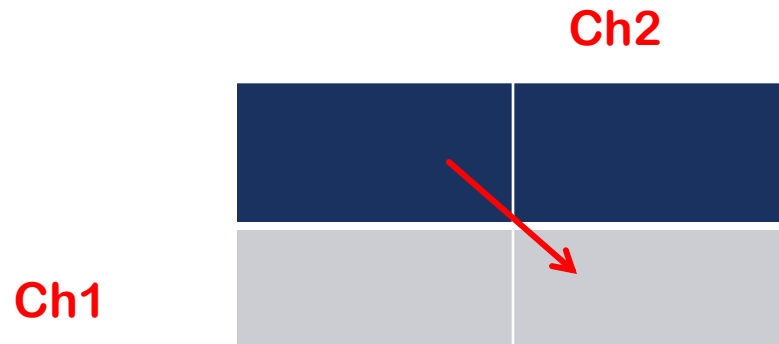
Similarity Score will be 0



Match (+1), mismatch(-1), Linear Gap Model (-1)

# Needleman Wunch algorithm

- To compute the global alignment, we need to trace back the matrix.
- Start from the bottom right corner and follow the arrows back.





# Needleman Wunch algorithm

- To compute the global alignment, we need to trace back the matrix.
- Start from the bottom right corner and follow the arrows back.



# Needleman Wunch algorithm

- To compute the global alignment, we need to trace back the matrix.
- Start from the bottom right corner and follow the arrows back.



# Needleman Wunch algorithm

A

T

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1	1	1	0	1
C	-5	-3	-3	-1	0	0	0	0
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

# Needleman Wunch algorithm

CA

CT

		G	C	A	T	G	C	T
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7
G	0	-1	-2	-3	-4	-5	-6	-7
A	-1	1	0	-1	-2	3	-4	-5
T	-2	0	0	1	0	-1	-2	-3
T	-3	-1	-1	0	2	1	0	-1
A	-4	-2	-2	-1	1	1	0	1
C	-5	-3	-3	-1	0	0	0	0
C	-6	-4	-2	-2	-1	-1	1	0
A	-7	-5	-3	-1	-2	-2	0	0

# Needleman Wunch algorithm

ACA

GCT

		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0

# Needleman Wunch algorithm

**TACA**  
**TGCT**

		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0

# Needleman Wunch algorithm

**TTACA**  
**-TGCT**

		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0

# Needleman Wunch algorithm

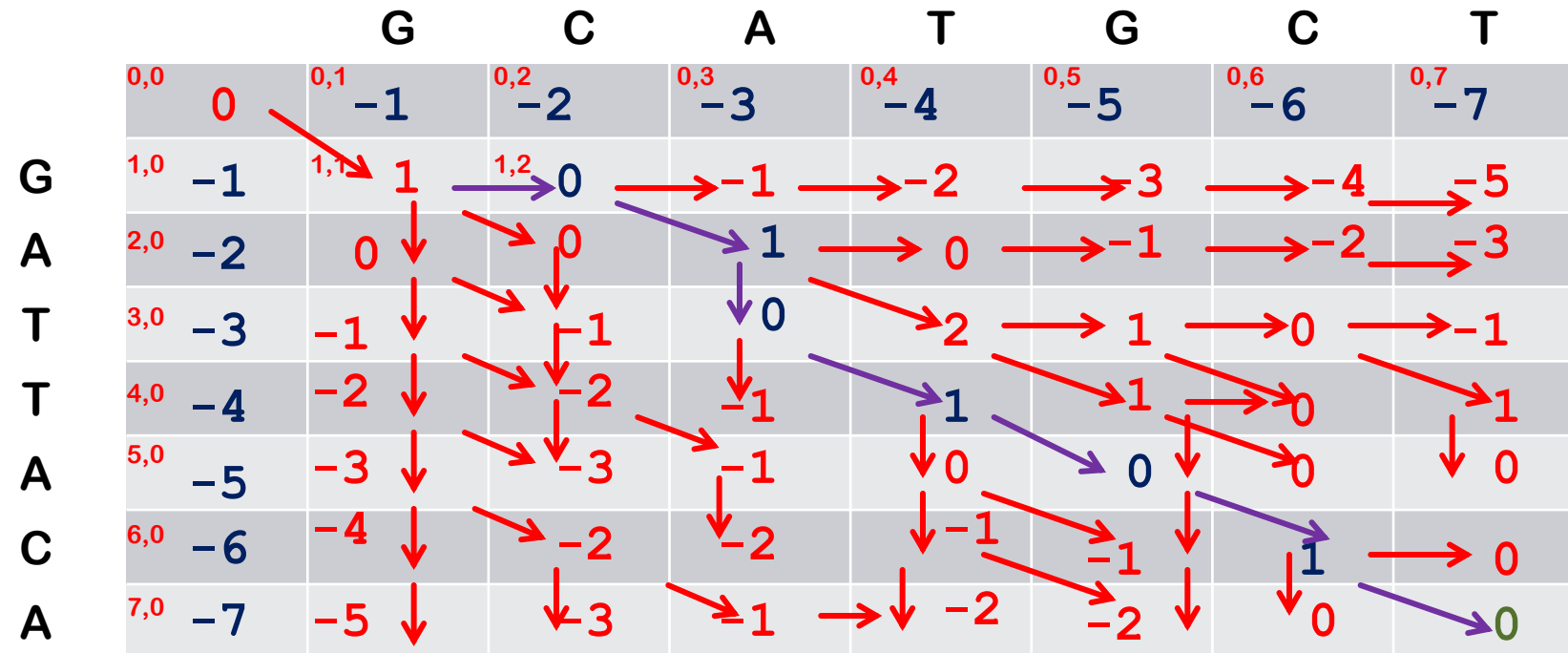
**ATTACA**  
**A-TGCT**

		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0



# Needleman Wunch algorithm

**-ATTACA**  
**CA-TGCT**



# Needleman Wunch algorithm

**G-ATTACA**

**GCA-TGCT**

		G	C	A	T	G	C	T	
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	
	0	-1	-2	-3	-4	-5	-6	-7	
G	1,0	-1	1	0	-1	-2	3	-4	-5
A	2,0	-2	0	0	1	0	-1	-2	-3
T	3,0	-3	-1	-1	0	2	1	0	-1
T	4,0	-4	-2	-2	-1	1	1	0	1
A	5,0	-5	-3	-3	-1	0	0	0	0
C	6,0	-6	-4	-2	-2	-1	-1	1	0
A	7,0	-7	-5	-3	-1	-2	-2	0	0

# Sequence alignment

## Sequence Alignment

pairwise alignment

multiple alignment

local

global

local

global

Smith Waterman algorithm

# Smith Waterman algorithm

- **Example:** Suppose you have two sequences:
  - S1=GGTTGACTA
  - S2=TGTTACGG
  - Match (+3), mismatch(-3), Linear Gap Model (-2)
    - Length of S1=m=9chars.
    - Length of S2=n=8chars.
    - Create a matrix of rows (m+1) and columns (n+1)

# Smith Waterman algorithm

S1=GGTTGACTA

S2=TGTTACGG

Match (+3), mismatch(-3), Linear Gap Model (-2)

Step 1: Fill all (i,0) with 0

		T	G	T	T	A	C	G	G	
		0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G		1,0 0								
G		2,0 0								
T		3,0 0								
T		4,0 0								
G		5,0 0								
A		6,0 0								
C		7,0 0								
T		8,0 0								
A		9,0 0								

# Smith Waterman algorithm

S1=GGTTGACTA

S2=TGTTACGG

Match (+3), mismatch(-3), Linear Gap Model (-2)

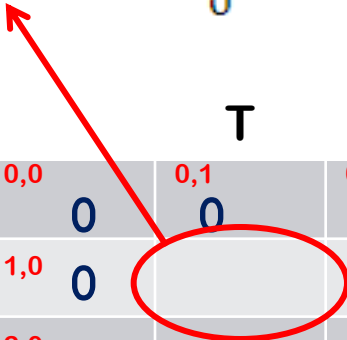
Step 1: Fill all (0,j) with 0

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0							
T	2,0	0							
T	3,0	0							
G	4,0	0							
A	5,0	0							
C	6,0	0							
T	7,0	0							
A	8,0	0							
	9,0	0							

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

$$cell(i,j) = \max \left\{ \begin{array}{l} (i-1, j-1) + score(ch1, ch2) \\ (i, j-1) + Gap\ Score \\ (i-1, j) + Gap\ Score \\ 0 \end{array} \right\}$$



		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0								
G	2,0								
T	3,0								
T	4,0								
G	5,0								
A	6,0								
C	7,0								
T	8,0								
A	9,0								

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
	0	0	0	0	0	0	0	0	0
G	1,0	0							
G	2,0								
T	3,0								
T	4,0								
G	5,0								
A	6,0								
C	7,0								
T	8,0								
A	9,0								



# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0	3	1	0	0	0	3	3
T	2,0	0	3	1					
T	3,0								
T	4,0								
G	5,0								
A	6,0								
C	7,0								
T	8,0								
A	9,0								

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0	3	1	0	0	0	3	3
G	2,0	0	0	3	1	0	0	3	6
T	3,0	0	3	1	6	4	2	0	1
T	4,0	0	3	1	4	9	7	5	3
G	5,0	0	1	6	4	7	6	4	8
A	6,0	0	0	4	3	5	10	8	6
C	7,0	0	0	2	1	3	8	13	11
T	8,0	0	3	1	5	4	6	11	10
A	9,0	0	1	0	3	2	7	9	8

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

C

C

		T	G	T	T	A	C	G	G
G	0,0	0	0	0	0	0	0	0	0
G	1,0	0	0	3	1	0	0	3	3
T	2,0	0	0	3	1	0	0	3	6
T	3,0	0	3	1	6	4	2	0	1
G	4,0	0	3	1	4	9	7	5	3
A	5,0	0	1	6	4	7	6	4	8
C	6,0	0	0	4	3	5	10	8	6
T	7,0	0	0	2	1	3	8	13	11
A	8,0	0	3	1	5	4	6	11	10
A	9,0	0	1	0	3	2	7	9	8

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

AC

AC

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
	0	0	0	0	0	0	0	0	0
G	1,0	0	0	3	1	0	0	3	3
G	2,0	0	0	3	1	0	0	3	6
T	3,0	0	3	1	6	4	2	0	1
T	4,0	0	3	1	4	9	7	5	3
G	5,0	0	1	6	4	7	6	4	8
A	6,0	0	0	4	3	5	10	8	6
C	7,0	0	0	2	1	3	8	13	11
T	8,0	0	3	1	5	4	6	11	10
A	9,0	0	1	0	3	2	7	9	8

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

GAC  
-AC

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0	3	1	0	0	0	3	3
G	2,0	0	3	1	0	0	0	3	6
T	3,0	0	3	1	6	4	2	0	1
T	4,0	0	3	1	4	9	7	5	3
G	5,0	0	1	6	4	7	6	4	8
A	6,0	0	0	4	3	5	10	8	6
C	7,0	0	0	2	1	3	8	13	11
T	8,0	0	3	1	5	4	6	11	10
A	9,0	0	1	0	3	2	7	9	8

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

TGAC

T-AC

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0	3	1	0	0	0	3	3
G	2,0	0	3	1	0	0	0	3	6
T	3,0	0	3	1	6	4	2	0	1
T	4,0	0	3	1	4	9	7	5	3
G	5,0	0	1	6	4	7	6	4	8
A	6,0	0	0	4	3	5	10	8	6
C	7,0	0	0	2	1	3	8	13	11
T	8,0	0	3	1	5	4	6	11	10
A	9,0	0	1	0	3	2	7	9	8

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

TTGAC

TT-AC

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	0	0	0	0	0	0	0
G	1,0	0	3	1	0	0	0	3	3
G	2,0	0	3	1	0	0	0	3	6
T	3,0	3	1	6	4	2	0	1	4
T	4,0	3	1	4	9	7	5	3	2
G	5,0	1	6	4	7	6	4	8	6
A	6,0	0	4	3	5	10	8	6	5
C	7,0	0	2	1	3	8	13	11	9
T	8,0	3	1	5	4	6	11	10	8
A	9,0	1	0	3	2	7	9	8	7

# Smith Waterman algorithm

Match (+3), mismatch(-3), Linear Gap Model (-2)

GTTGAC

GTT-AC

		T	G	T	T	A	C	G	G
	0,0	0,1	0,2	0,3	0,4	0,5	0,6	0,7	0,8
G	0	0	3	1	0	0	0	3	3
G	2,0	0	3	1	0	0	0	3	6
T	3,0	3	1	6	4	2	0	1	4
T	4,0	3	1	4	9	7	5	3	2
G	5,0	1	6	4	7	6	4	8	6
A	6,0	0	4	3	5	10	8	6	5
C	7,0	0	2	1	3	8	13	11	9
T	8,0	3	1	5	4	6	11	10	8
A	9,0	1	0	3	2	7	9	8	7





**Thank you!**