

# Comparaison et alignement de séquences

# Séquence Biologique

- Séquence AND( A,C,G,T)

Exemple : TATTTACAAC

- Séquence ARN ( A,C,G,U)

Exemple : UAUACAAGG

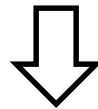
- Séquence protéique ARNDCQEGHILKM

# L'évolution des séquences

**Temps  $t = t_0$**

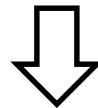
S1 TATACATTAG

S2 TATACATTAG



S1 TAT**T**ACATTAG

S2 TATACATTG



**Temps  $t = t$**

S1 TATTACATTAG

S2 TATACATTG

Déduire la  
correspondance

# Des questions sur les séquences

1.

Question biologique : Quelles sont les similitudes entre le génome des Homo sapiens et les chimpanzé

Question informatique : soit deux séquences **S1** et **S2**. calculer le de similarité  $\text{sim}(S1, S2)$ .

2.

Question biologique :

ce gène cause le cancer dans les souris, les humains ont-ils ce gène.

Question informatique : soit une séquence **S** (le gène de la souris ) et D la base de donnée de tous les gènes des Êtres humains, trouver la séquence **R** ou le  $\text{sim}(\mathbf{R}, \mathbf{S})$  est supérieure au seuil.

---

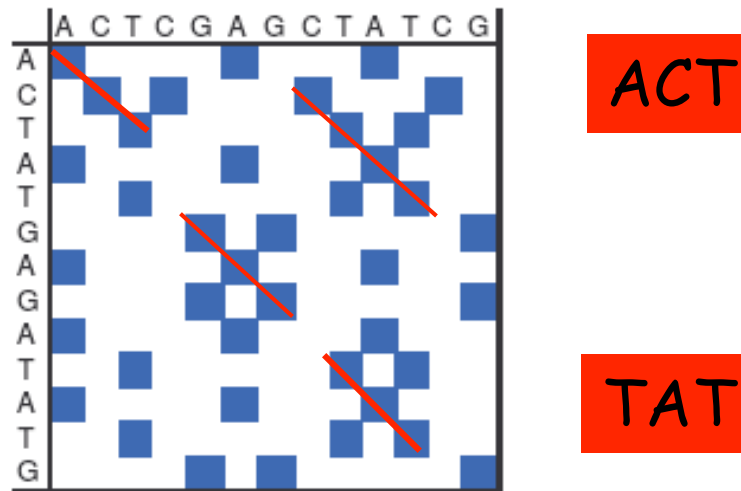
Cliquez pour ajouter un titre

- Comment comparer deux séquences ?

# Le Dot-Plot

*Maizel et Lenk 1981 – Staden 1982*

- Tableau indexé par les caractères des deux séquences
- Identité : ■      Non Identité: □



Les similarités (ressemblances locales) apparaissent le long des segments diagonaux

# Le Dot-Plot (en pratique ...)

Sans fenêtre

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

# Le Dot-Plot (en pratique ...)

Fenêtre de taille 2

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



# Le Dot-Plot (en pratique ...)

Fenêtre de taille 3

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

# Le Dot-Plot (en pratique ...)

Fenêtre de taille 3,  
Seuil identité  $\geq 2/3$

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

# Le Dot-Plot (en pratique ...)

Fenêtre de taille 3,  
Seuil identité  $\geq 2/3$

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

# Le Dot-Plot (en pratique ...)

Fenêtre de taille 3,  
Seuil identité  $\geq 2/3$

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

# Le Dot-Plot (en pratique ...)

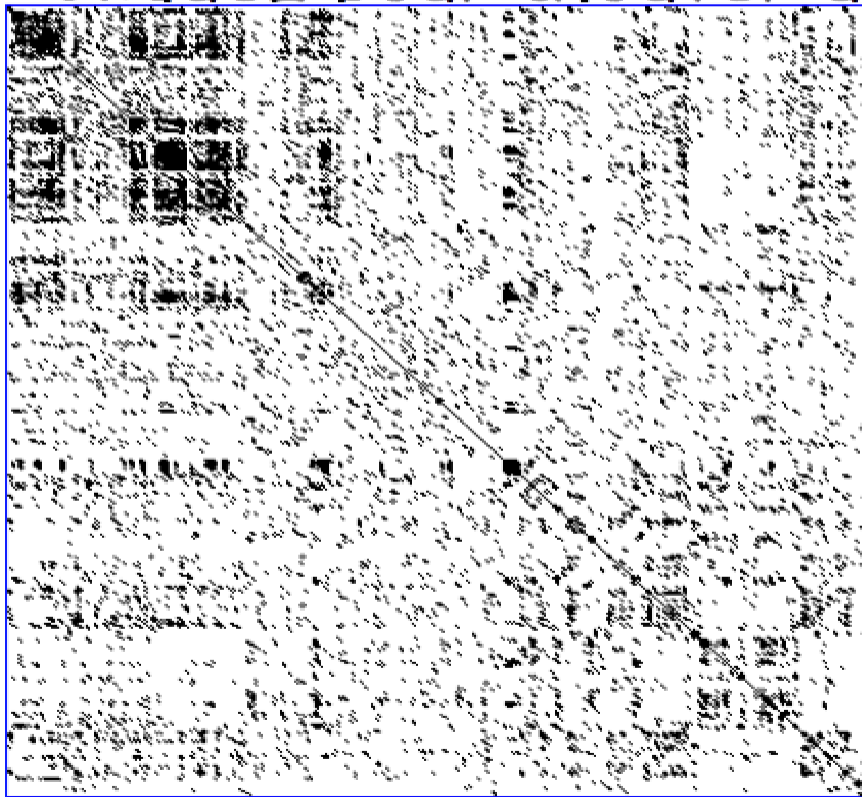
- Beaucoup de « bruit »
- Utiliser une **fenêtre** pour « lisser »
- Choisir un **seuil** au dessus duquel la similarité dans la fenêtre génère un point
- => Trouver un équilibre en faisant varier la taille de la fenêtre et le seuil

# Le Dot-Plot (en pratique ...)

- Autre critère de « bruit » = nature des séquences.
- ADN 4 lettres => beaucoup de bruit.
- Protéines 20 aa => moins de bruit.
- En protéines, mutations silencieuses non visibles ne « brulent » pas l'alignement.

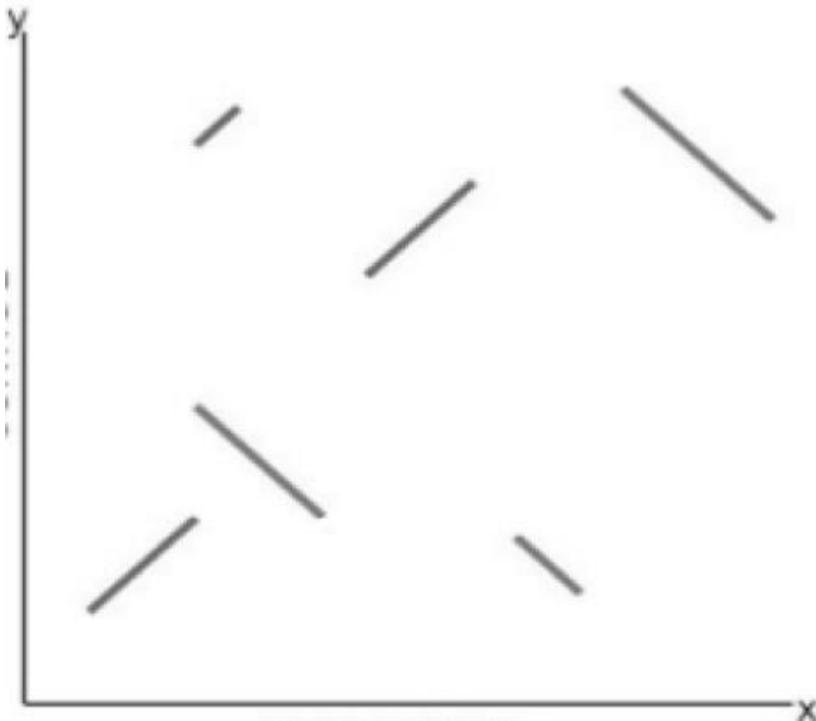
# Le Dot-Plot : exemple

- Cliquez pour ajouter un texte



*Diagonale = 2 séquences  
« identiques »  
Pavés sombres =  
répétitions*

*ADN ?*



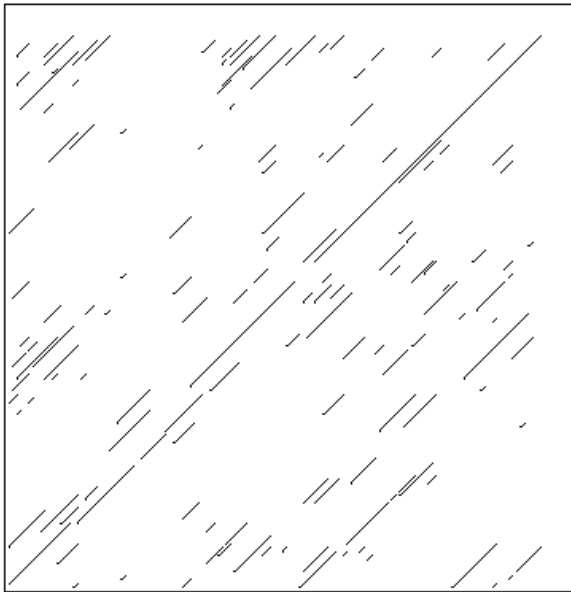
*Orientation différente =  
une inversion.*

*Moins de point si  
protéique, et/ou si  
fenêtre plus grande et  
seuil plus haut.*

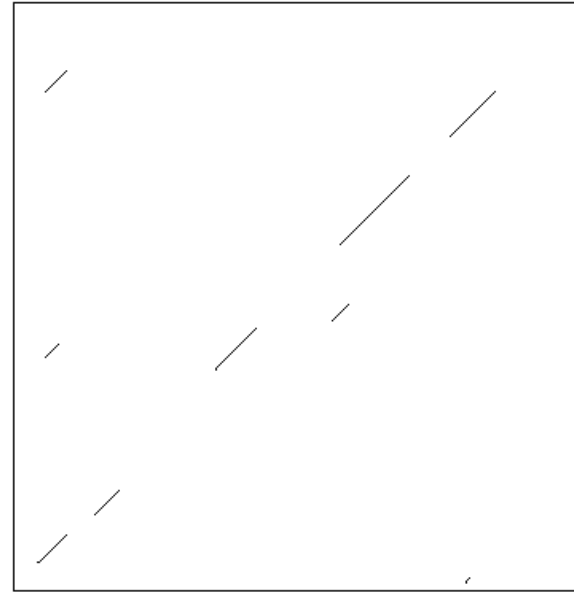
*Pas d'inversion en  
protéines*



## *Exemple des Hémoglobines Humaines*



Identités = 3/10



Identités = 5/10

# Avantages et inconvénients du Dot-Plot

- Simple et rapide, graphique mais ...
- L'interprétation visuelle rend impossible l'utilisation du Dot-Plot dans le cadre d'une « comparaison massive »
  - *i.e.*, une séquence protéique *versus* la banque UniProtKB qui contient environ 21 millions de protéines (02-Avril-12)

---

Alignement de séquences 2 à 2  
(nucléotides ou acides aminés)

---

# Alignement

- Mise en correspondance de deux séquences (ADN ou protéines) pour faire apparaître les similarités, *i.e.*, segments communs

**AAAATTTTTTGGCCTTTAA et AAAAGCCCAA**

**AAAATTTTTTGGCCTTTAA**

**AAAAGCCCAA**

**AAAATTTTTTGGCCTTTAA**

**AAAA                      GCCC              AA**

# Alignement

- Mise en correspondance de deux séquences (ADN ou protéines)
- 3 événements élémentaires :
  - ▣ Correspondance (match)
  - ▣ Substitution (mismatch)
  - ▣ Indel (Insertion/Délétion)

ACGGCTAT

ACGGCTAT

ACGGCTAT

| | |

| | | | |

ACTGTAT

ACTGTAT-

ACTG-TAT

# Alignement

- Chaque alignement a 1 **Score**
- Il dépend des « pénalités » fixées pour les événements élémentaires
- Par exemple :
  - Correspondance/Match : +2
  - Substitution/Mismatch : -1
  - Indel : -2

Le score de l'alignement est la somme des scores des événements élémentaires

# Alignement

## ■ Alignement des deux séquences nucléiques **ACGGCTAT** et **ACTGTAT**

■ Correspondance: +2, Substitution: -1, Indel: -2

**ACGGCTAT**

| | |

**ACTGTAT-**

$$\text{Score} = 2+2-1+2-1-1-1-2 = 0$$

**ACGGCTAT**

| | | | |

**ACTG-TAT**

$$\text{Score} = 2+2-1+2-2+2+2+2 = 9$$

# Code Python

```
seq1 = 'ACGGCTAT'  
seq2 = 'ACTGTAT-'
```

```
score = 0
```

```
for a, b in zip(seq1, seq2):  
    if a == b:  
        score += 2  
    elif a == '-' or b == '-':  
        score = score - 2  
    else:  
        score = score - 1
```

```
print(score)
```



# Alignement Global

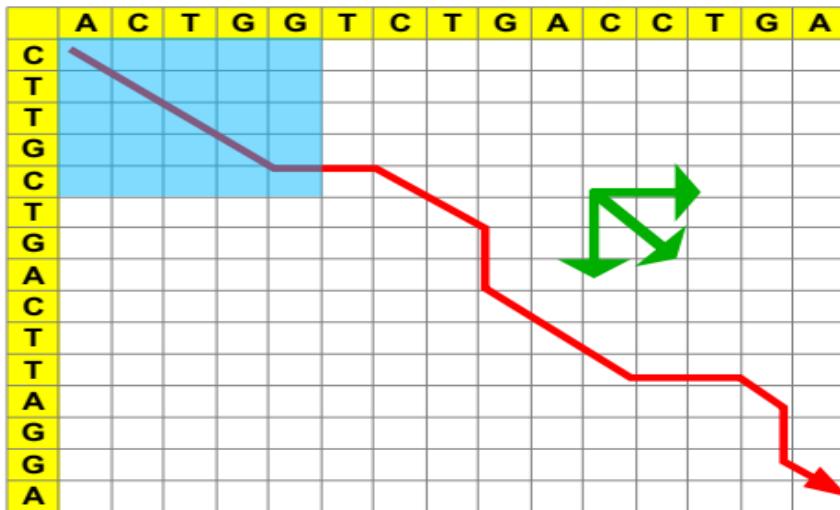
*Needleman & Wunsch - 1970*

- But : Evaluation d'une ressemblance globale entre deux séquences = sur toute la longueur
- Problème :
  - ▣ Quel est l'alignement de score maximal ?

# Alignement Global: Programmation dynamique

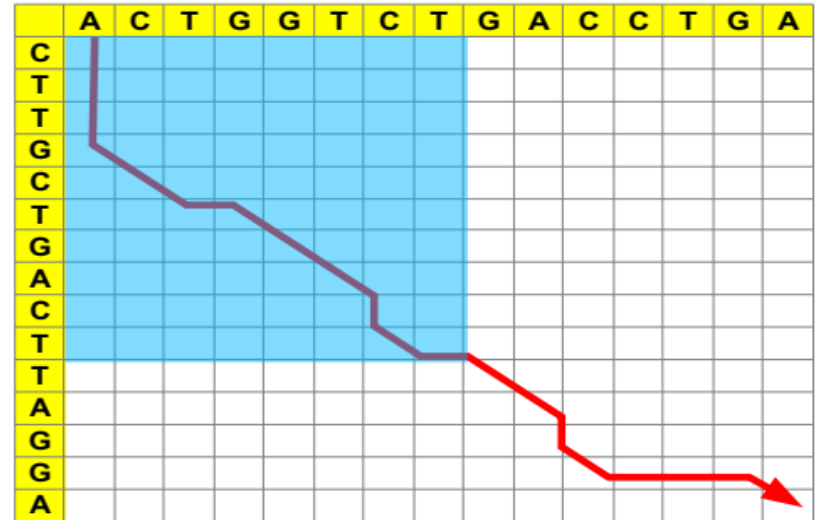
*Needleman & Wunsch - 1970*

Un alignement = un chemin dans la matrice  
À chaque chemin est associé un score



L=19

ACTGGTCT--GACCTG--A  
CTTG--CTGACTT--AGGA



---ACTGGT- GA-CCTGA  
CTCTTGCTGACTTAGG---A

L=21

Nous cherchons l'alignement avec le mieux score

# Alignement Global: Programmation dynamique

*Needleman & Wunsch - 1970*

Diviser pour mieux aligner


cet alignement de taille L aura le meilleur score ...

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

... à condition que cet alignement de taille L-1  
ait le meilleur score !

# Alignement Global: Programmation dynamique

		A	C	T	G
C					
T					
T					
G					



- **Règle 1:**  
chaque case va contenir un score; le score de l'alignement sera celui de la case en bas à droite
- **Règle 2:**  
le score d'une case se déduit à partir de celui des cases au-dessus, à gauche ou en diagonale
- **Règle 3:**  
un pas horizontal/vertical coûte 1 gap  
un pas diagonal coûte 1 position alignée (match ou mismatch)

# Alignement Global: Programmation dynamique

*Needleman & Wunsch - 1970*

		A	C	T	G	
		0	-4	-8	-12	-16
C		-4				
T		-8				
T		-12				
G		-16				

Gap penalty -4

# Alignement Global: Programmation dynamique

*Needleman & Wunsch - 1970*

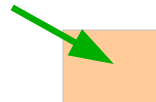
## Etape 2:

on remplit toutes les cases en gardant en mémoire le mouvement qui donne le meilleur score

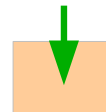
		A	C	T	G
	0	-4	-8	-12	-16
C	-4				
T	-8				
T	-12				
G	-16				

Score:

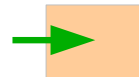
gap: -4    mismatch: -4



alignement AC  $\rightarrow$  score =  $0 - 4 = -4$



insertion de gap  $\rightarrow$  score =  $-4 - 4 = -8$



insertion de gap  $\rightarrow$  score =  $-4 - 4 = -8$

# Alignement Global: Programmation dynamique

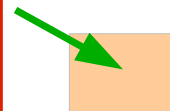
## Etape 2:

on remplit toutes les cases en gardant en mémoire le mouvement qui donne le meilleur score

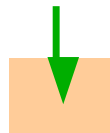
Score:

gap: -4    mismatch: -4

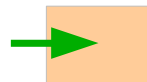
		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4			
T	-8				
T	-12				
G	-16				



alignement AC  $\rightarrow$  score =  $0 - 4 = -4$



insertion de gap  $\rightarrow$  score =  $-4 - 4 = -8$



insertion de gap  $\rightarrow$  score =  $-4 - 4 = -8$

# Alignement Global: Programmation dynamique

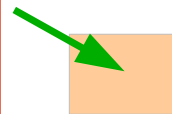
*Needleman & Wunsch - 1970*

		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4	0		
T	-8				
T	-12				
G	-16				

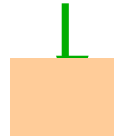
Score:

gap: -4    mismatch: -4

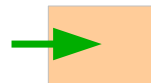
match: +4



alignement CC  $\rightarrow$  score =  $-4+4 = 0$



insertion de gap  $\rightarrow$  score =  $-8-4 = -12$



insertion de gap  $\rightarrow$  score =  $-4-4 = -8$



# Alignement Global: Programmation dynamique

*Needleman & Wunsch - 1970*

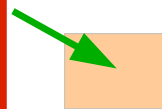
		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4	0		
T	-8	-8			
T	-12				
G	-16				

Score:

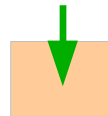
gap: -4

mismatch: -4

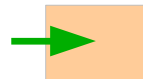
match: +4



alignement AT  $\rightarrow$  score =  $-4 - 4 = -8$



insertion de gap  $\rightarrow$  score =  $-4 - 4 = -8$



insertion de gap  $\rightarrow$  score =  $-8 - 4 = -12$

# Alignement Global: Programmation dynamique

*Needleman & Wunsch - 1970*

		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4	0	-4	-8
T	-8	-8	-4	4	0
F	-12	-12	-8	0	0
G	-16	-16	-12	-4	4

meilleur score

Score:

gap: -4

mismatch: -4

match: +4

# Alignement Global: Programmation dynamique

## Etape 3:

On part du score en bas à droite, et on remonte le cours des flèches pour trouver l'alignement (« **backtracking** »)

		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4	0	-4	-8
T	-8	-8	-4	4	0
T	-12	-12	-8	0	0
G	-16	-16	-12	-4	4

### Bilan:

- 24 scores calculés
- $3^{4+4} = 6561$  chemins possibles

2 chemins =  
2 alignements **optimaux**:  
AC-TG      ACT-G  
-CTTG      -CTTG      score: +4

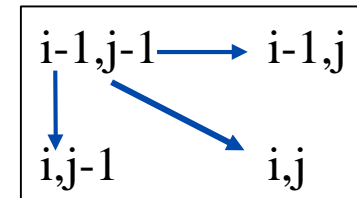
Alignement **global**:  
on aligne les 2 séquences  
du début à la fin

# Algorithme de « programmation dynamique »

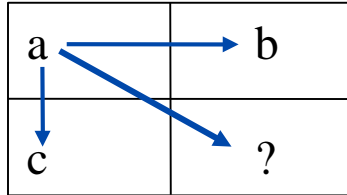
■ 2 séq  $A = (a_1, \dots, a_n)$  et  $B(b_1, \dots, b_m)$

■  $S_{i,j}$  = score maximum entre 2 séquences alignées du début jusqu'aux résidus  $a_i$  et  $b_j$  tel que :

■ 
$$S_{i,j} = \max \begin{cases} S_{i-1,j-1} + w(a_i, b_j) \\ S_{i-1,j} + g \\ S_{i,j-1} + g \end{cases}$$



# Récurrance



? = MAX

$\left\{ \begin{array}{l} a + \text{Subs. ou Corresp.} \\ b + \text{Indel} \\ c + \text{Indel} \end{array} \right.$

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

# Example

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

# Example : Initialisation

→ Indel = -2      ↓      Indel = -2

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

# Exemple : Remplissage ligne par ligne

**max** [  $\downarrow -2-2=-4 \rightarrow -2-2=-4 \searrow 0+2=2$  ]

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

Correspondance=2 ou substitution=-1

Indel=-2

Indel=-2



# Exemple : Remplissage ligne par ligne

		A	C	G	G	C	T	A	T
	0	-2	-4	-6	-8	-10	-12	-14	-16
A	-2	2	0	-2	-4	-6	-8	-10	-12
C	-4	0	4	2	0	-2	-4	-6	-8
T	-6	-2	2	3	1	-1	0	-2	-4
G	-8	-4	0	4	5	3	1	-1	-3
T	-10	-6	-2	2	3	4	5	3	1
A	-12	-8	-4	0	1	2	3	7	5
T	-14	-10	-6	-2	-1	0	4	5	9

↙ Correspondance=2 ou substitution=-1

↓ Indel=-2



Indel=-2

## Exemple : Recherche du chemin des scores maximaux

		A	C	G	G	C	T	A	T
	0	-2	-4	-6	-8	-10	-12	-14	-16
A	-2	2	0	-2	-4	-6			
C	-4	0	4	2	0	-2			
T	-6	-2	2	3	1	-1			
							0	-2	-4
G	-8	-4	0	4	5	3	1	-1	-3
T	-10	-6	-2	2	3	4	5	3	1
A	-12	-8	-4	0	1	2	3	7	5
T	-14	-10	-6	-2	-1	0	4	5	9

**ACGGCTAT**

- | | | | |

- ACTG-TAT

Correspondance=2 ou substitution=-1

Indel=-2

Indel=-2

# Sensibilité aux paramètres

ACGGCT-ATC

|| | | ||

ACTG-TAATG

Correspondance : +2

Substitution : -1

Indel : -1

ACGGCTATC

|| | ||

ACTGTAATG

■ Correspondance : +1

■ Substitution : -1

■ Indel : -2

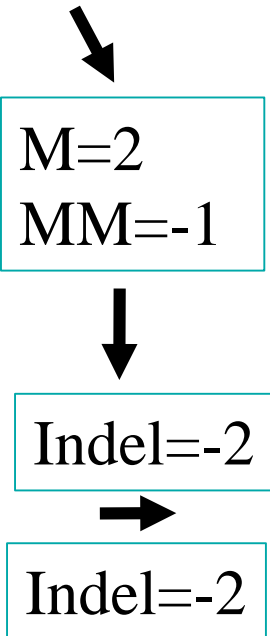
L'alignement optimal dépend de :

- du coût des matches/mismatches
- des pénalités pour les indels, etc.

# Local : Remplissage ligne par ligne

**max**  $\left[ \downarrow 0-2=-2 \rightarrow 0-2=-2 \searrow 0+2=2 \quad 0 \right]$

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



# Local : Remontée

**max**  $\left[ \downarrow -2-2=-4 \rightarrow -2-2=-4 \searrow 0+2=2 \quad 0 \right]$

		A	C	G	G	C	T	A	T
	0	0	0	0	0	0	0	0	0
A	0	2	0	0	0	0	0	2	1
G	0	0	1	2	2	0	0	0	1
C	0	0	2	0	1	4	2	0	0
T	0	0	0	1	0	2	6	4	2
T	<div> <b>GCTAT</b>           <b>GCTTT</b> </div>						4	5	6
T							2	3	7
C							0	1	5

Score =  $2+2+2-1+2=7$

