

# Les méthodes d'alignement dans l'analyse des séquences

# But d'un alignement

- Déterminer si des séquences ont divergé depuis un ancêtre commun
- Degré de similarité  $\rightarrow$  homologie ?
- Nombre minimum d'opérations d'édition (substitution, insertion, suppression) pour transformer  $x$  en  $y$

# Exemples d'alignements

$\alpha$ -globine HBA\_HUMAN

GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSDLHAHKL  
GNPKVKAHGKKVLGAFSDGLAHLDNLKGTFATLSELHCDKL

$\beta$ -globine HBA\_HUMAN

$\alpha$ -globine HBA\_HUMAN

GSAQVKGHGKKVADALTNAVAHV - - - D - - DMPNALSALSDLHAHKL  
NNPELQAHAGKVFVKLVYEAAIQLQVTGVVVTDATLKNLGSVHVSCKG

leghamoglobine lapin LGB2\_LUPLU

$\alpha$ -globine HBA\_HUMAN

GSAQVKGHGKKVADALTNAVAHVDDMPNALSALSD - - - - LHAHKL  
GSGYLVGDSLTFVDLL - - VAQHTADLLAANAALLDEFPQFKAHQE

Nématode S-transférase F11G11.2

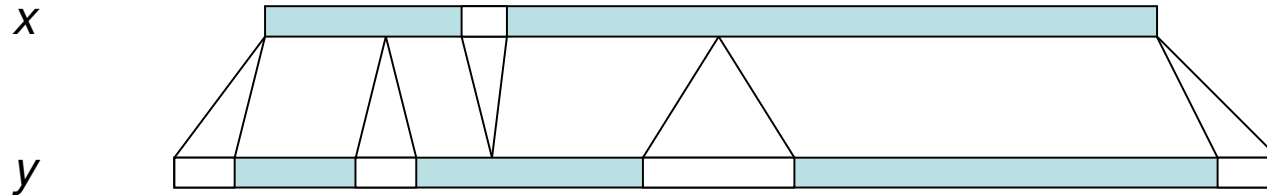
Extrait de [DEKM 1998]

# Méthodologie

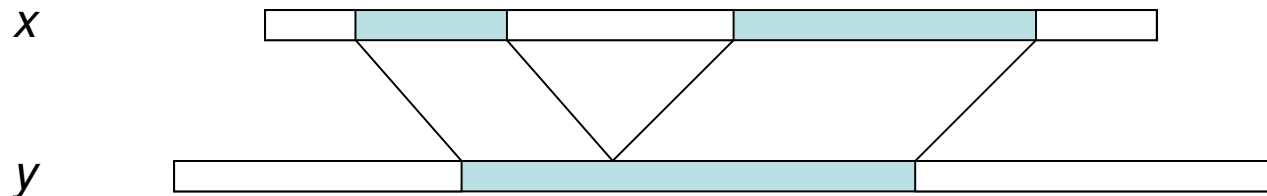
- Quel type d'alignement ?
- Quel score ?
- Quel algorithme ?
- Quelle méthode statistique ?

# Quel type d'alignement ?

brèche



Global



Local

# Matrices de substitution

- PAM (Point Accepted Mutation)  
[Dayhoff *et al* 1978]
- BLOSUM (Blocks Substitution Matrices)  
[Henikoff & Henikoff 1992]
- autres

# PAM 250

|   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
|---|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|----|----|--|
| C | 12 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| S | 0  | 2  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| T | -2 | 1  | 3  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| P | -3 | 1  | 0  | 6  |    |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| A | -2 | 1  | 1  | 1  | 2  |    |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| G | -3 | 1  | 0  | -1 | 1  | 5  |    |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| N | -4 | 1  | 0  | -1 | 0  | 0  | 2  |    |    |    |    |    |    |    |    |    |    |   |    |    |  |
| D | -5 | 0  | 0  | -1 | 0  | 1  | 2  | 4  |    |    |    |    |    |    |    |    |    |   |    |    |  |
| E | -5 | 0  | 0  | -1 | 0  | 0  | 1  | 3  | 4  |    |    |    |    |    |    |    |    |   |    |    |  |
| Q | -5 | -1 | -1 | 0  | 0  | -1 | 1  | 2  | 2  | 4  |    |    |    |    |    |    |    |   |    |    |  |
| H | -3 | -1 | -1 | 0  | -1 | -2 | 2  | 1  | 1  | 3  | 6  |    |    |    |    |    |    |   |    |    |  |
| R | -4 | 0  | -1 | 0  | -2 | -3 | 0  | -1 | -1 | 1  | 2  | 6  |    |    |    |    |    |   |    |    |  |
| K | -5 | 0  | 0  | -1 | -1 | -2 | 1  | 0  | 0  | 1  | 0  | 3  | 5  |    |    |    |    |   |    |    |  |
| M | -5 | -2 | -1 | -2 | -1 | -3 | -2 | -3 | -2 | -1 | -2 | 0  | 0  | 6  |    |    |    |   |    |    |  |
| I | -2 | -1 | 0  | -2 | -1 | -3 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | 2  | 5  |    |    |   |    |    |  |
| L | -6 | -3 | -2 | -3 | -2 | -4 | -3 | -4 | -3 | -2 | -2 | -3 | -3 | 4  | 2  | 6  |    |   |    |    |  |
| V | -2 | -1 | 0  | -1 | 0  | -1 | -2 | -2 | -2 | -2 | -2 | -2 | -2 | 2  | 4  | 2  | 4  |   |    |    |  |
| F | -4 | -3 | -3 | -5 | -4 | -5 | -4 | -6 | -5 | -5 | -2 | -4 | -5 | 0  | 1  | 2  | -1 | 9 |    |    |  |
| Y | 0  | -3 | -3 | -5 | -3 | -5 | -2 | -4 | -4 | -4 | 0  | -4 | -4 | -2 | -1 | -1 | -2 | 7 | 10 |    |  |
| W | -5 | -2 | -5 | -6 | -6 | -7 | -4 | -7 | -7 | -5 | -3 | 2  | -3 | -4 | -5 | -2 | -6 | 0 | 0  | 17 |  |
|   | C  | S  | T  | P  | A  | G  | N  | D  | E  | Q  | H  | R  | K  | M  | I  | L  | V  | F | Y  | W  |  |

# Définition

Un **alignement** de  $x, y \in A^*$  est un mot  $w$  sur l'alphabet

$$(A \cup \{-\}) \times (A \cup \{-\}) - \{(-,-)\}$$

tel que  $\Pi_1(w) = x$  et  $\Pi_2(w) = y$



# Exemple

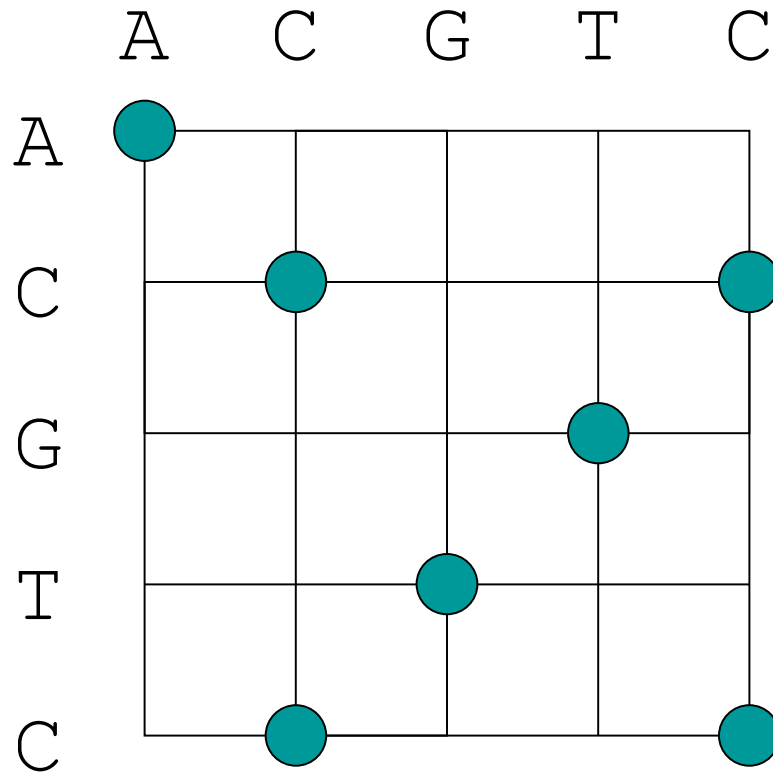
$x = \text{ACTGC}$  et  $y = \text{ACGTC}$

$w = (\text{A,A}) (\text{C,C}) (\text{T,-}) (\text{C,G}) (-,\text{T}) (\text{C,C})$

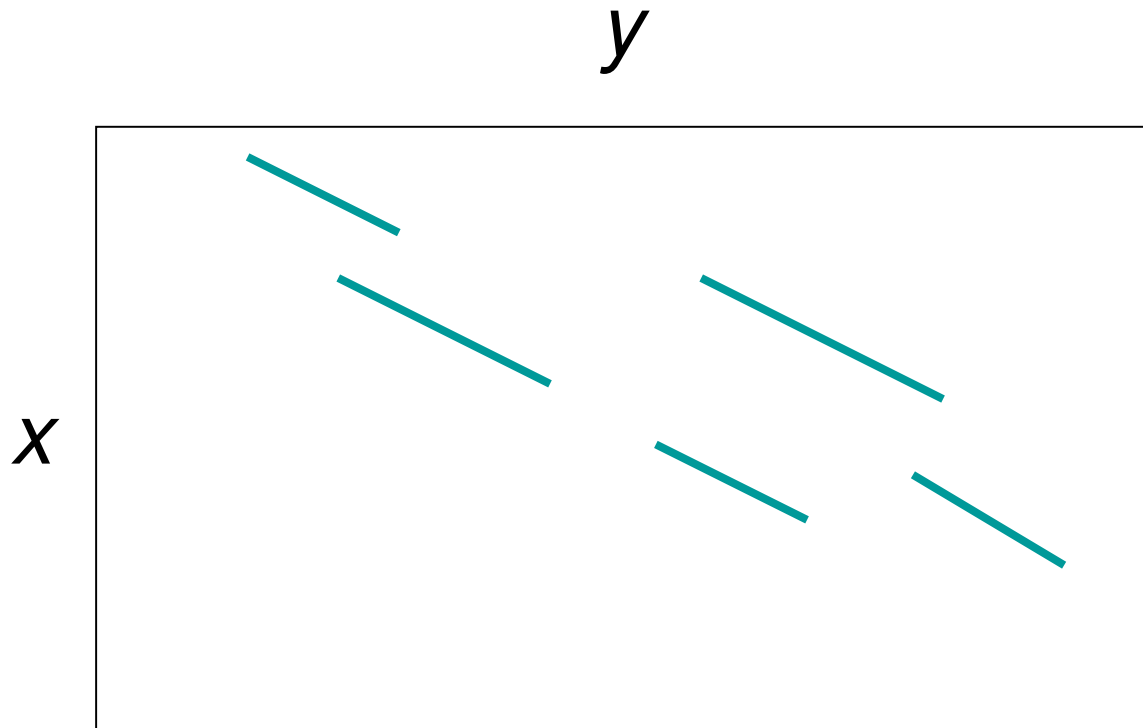
$w = \begin{pmatrix} \text{A} \\ \text{A} \end{pmatrix} \begin{pmatrix} \text{C} \\ \text{C} \end{pmatrix} \begin{pmatrix} \text{T} \\ - \end{pmatrix} \begin{pmatrix} \text{C} \\ \text{G} \end{pmatrix} \begin{pmatrix} - \\ \text{T} \end{pmatrix} \begin{pmatrix} \text{C} \\ \text{C} \end{pmatrix}$

$w = \begin{pmatrix} \text{A} & \text{C} & \text{T} & \text{C} & - & \text{C} \\ \text{A} & \text{C} & - & \text{G} & \text{T} & \text{C} \end{pmatrix}$

# Tracé ou « dotplot »



# Tracé ou « dotplot »



# Score d'un alignement

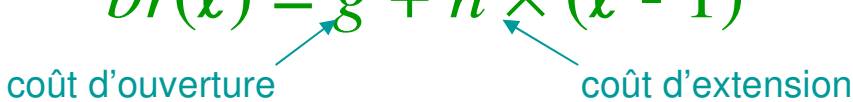
$$score(w) = \sum_{i=0}^{|w| - 1} score(w[i])$$

$score(a, b)$  est indépendant  
des positions de  $a$  et  $b$

# score ou distance ?

|                           | score | distance |
|---------------------------|-------|----------|
| $f(a,a) = f(b,b)$         | non   | oui      |
| inégalité<br>triangulaire | non   | oui      |
| global                    | oui   | oui      |
| local                     | oui   | non      |

# Coût des brèches

- constant  $br(\ell) = c$
- linéaire  $br(\ell) = e \times \ell$
- affine  $br(\ell) = g + h \times (\ell - 1)$   


coût d'ouverture

coût d'extension
- concave
- autres

# Calcul par programmation dynamique

- alignement global, coût linéaire des brèches
- table  $T$  de dimension  $(|x| + 1) \times (|y| + 1)$
- $T[i,j] = \text{score}(x[0..i], y[0..j])$

# Programmation dynamique

## Proposition

Pour  $i = 0, 1, \dots, |x| - 1$  et  $j = 0, 1, \dots, |y| - 1$  :

$$T[-1, -1] = 0$$

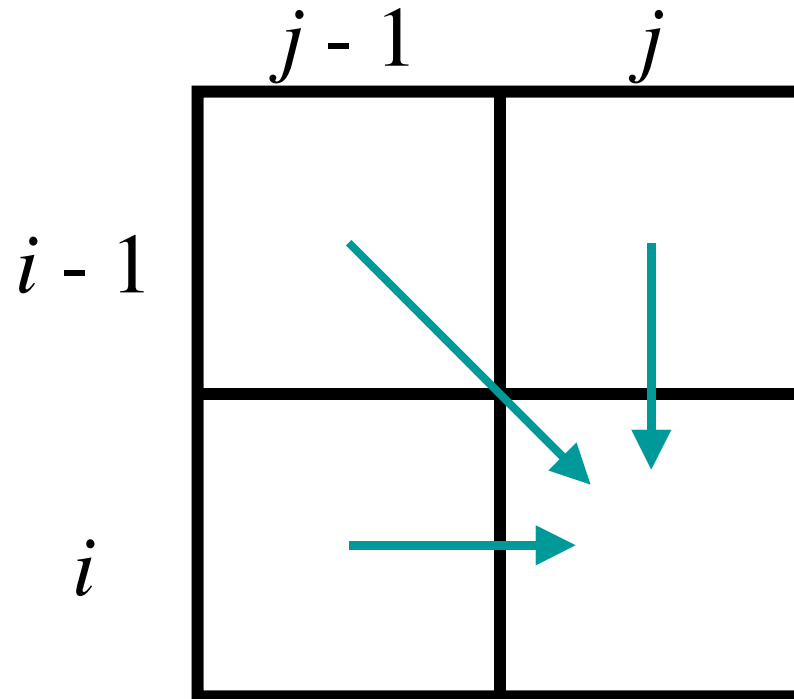
$$T[i, -1] = T[i - 1, -1] + \textit{Dél}(x[i])$$

$$T[-1, j] = T[-1, j - 1] + \textit{Ins}(y[j])$$

$$T[i, j] = \max \begin{cases} T[i - 1, j - 1] + \textit{Sub}(x[i], y[j]) \\ T[i - 1, j] + \textit{Dél}(x[i]) \\ T[i, j - 1] + \textit{Ins}(y[j]) \end{cases}$$



# Programmation dynamique



# Complexités

Le calcul du score s'effectue

- en temps  $O(|x| \times |y|)$
- en espace  $O(\min(|x|, |y|))$

[Needleman & Wunsch 1970] cubique

[Gotoh 1982]

# Calcul d'un alignement optimal

- tracé arrière (« traceback »)
- temps  $O(|x| \times |y|)$
- espace  $O(|x| \times |y|)$     2 bits par case

# Calcul de tous les alignements optimaux

3 bits par case

# Réduction de l'espace

méthode « diviser pour régner » due à  
[Hirschberg 75] pour le calcul d'un plus long  
sous-mot commun étendu au calcul d'un  
alignement par Myers & Miller

# Exemple

Sub

|   |    |    |    |   |
|---|----|----|----|---|
| A | 1  |    |    |   |
| C | -1 | 1  |    |   |
| G | -1 | -1 | 1  |   |
| T | -1 | -1 | -1 | 1 |
|   | A  | C  | G  | T |

$$\text{Del}(a) = \text{Sup}(a) = -1$$

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

|   |   | A | C | G | T | C |
|---|---|---|---|---|---|---|
|   | 0 |   |   |   |   |   |
| A |   |   |   |   |   |   |
| C |   |   |   |   |   |   |
| T |   |   |   |   |   |   |
| G |   |   |   |   |   |   |
| C |   |   |   |   |   |   |



|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 |    |    |    |    |    |
| C | -2 |    |    |    |    |    |
| T | -3 |    |    |    |    |    |
| G | -4 |    |    |    |    |    |
| C | -5 |    |    |    |    |    |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  |    |    |    |    |
| C | -2 |    |    |    |    |    |
| T | -3 |    |    |    |    |    |
| G | -4 |    |    |    |    |    |
| C | -5 |    |    |    |    |    |

|   |    | A    | C  | G  | T  | C  |
|---|----|------|----|----|----|----|
|   | 0  | -1   | -2 | -3 | -4 | -5 |
| A | -1 | 1 -2 |    |    |    |    |
| C | -2 |      |    |    |    |    |
| T | -3 |      |    |    |    |    |
| G | -4 |      |    |    |    |    |
| C | -5 |      |    |    |    |    |

|   |    | A          | C  | G  | T  | C  |
|---|----|------------|----|----|----|----|
|   | 0  | -1         | -2 | -3 | -4 | -5 |
| A | -1 | 1 -2<br>-2 |    |    |    |    |
| C | -2 |            |    |    |    |    |
| T | -3 |            |    |    |    |    |
| G | -4 |            |    |    |    |    |
| C | -5 |            |    |    |    |    |

|   |    | A          | C  | G  | T  | C  |
|---|----|------------|----|----|----|----|
|   | 0  | -1         | -2 | -3 | -4 | -5 |
| A | -1 | 1 -2<br>-2 |    |    |    |    |
| C | -2 |            |    |    |    |    |
| T | -3 |            |    |    |    |    |
| G | -4 |            |    |    |    |    |
| C | -5 |            |    |    |    |    |

|   |    | A            | C  | G  | T  | C  |
|---|----|--------------|----|----|----|----|
|   | 0  | -1           | -2 | -3 | -4 | -5 |
| A | -1 | 1 -2<br>-2 1 |    |    |    |    |
| C | -2 |              |    |    |    |    |
| T | -3 |              |    |    |    |    |
| G | -4 |              |    |    |    |    |
| C | -5 |              |    |    |    |    |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |



|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

The diagram illustrates a sequence alignment problem using a 6x6 matrix. The rows and columns are labeled with nucleotides: A, C, G, T, and C. The values in the matrix represent the scores for each alignment. The diagonal elements (1, 2, 1, 2, 1, 2) are circled in blue, and arrows indicate the path of maximum score from the bottom-right to the top-left.

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

|   |    | A  | C  | G  | T  | C  |
|---|----|----|----|----|----|----|
|   | 0  | -1 | -2 | -3 | -4 | -5 |
| A | -1 | 1  | 0  | -1 | -2 | -3 |
| C | -2 | 0  | 2  | 1  | 0  | -1 |
| T | -3 | -1 | 1  | 1  | 2  | 1  |
| G | -4 | -2 | 0  | 2  | 1  | 1  |
| C | -5 | -3 | -1 | 1  | 1  | 2  |

2 alignements  
optimaux

**ACTG - C**  
**AC - GTC**

**AC - TGC**  
**ACGT - C**

# Coût affine sur les brèches

$$br(\ell) = g + h \times (\ell - 1)$$

pour  $i = 0, 1, \dots, |x| - 1$  et  $j = 0, 1, \dots, |y| - 1$

- $D[i, j] = \max \{ D[i-1, j] - h, T[i-1, j] - g \}$
- $I[i, j] = \max \{ I[i, j-1] - h, T[i, j-1] - g \}$
- $T[i, j] = \max \begin{cases} D[i, j] \\ I[i, j] \\ T[i-1, j-1] + Sub(x[i], y[j]) \end{cases}$

+ phase d'initialisation marginale



# Coût affine sur les brèches

## Calcul

- en temps  $O(|x| \times |y|)$
- en espace  $O(|x| \times |y|)$

tracé arrière délicat

[Gotoh 1982]

# Alignement local

pour  $i = 0, 1, \dots, |x| - 1$  et  $j = 0, 1, \dots, |y| - 1$

- $T[-1, -1] = T[i, -1] = T[-1, j] = 0$
- $T[i, j] = \max \begin{cases} T[i-1, j] + \textit{Dél}(x[i]) \\ T[i, j-1] + \textit{Ins}(y[j]) \\ T[i-1, j-1] + \textit{Sub}(x[i], y[j]) \\ 0 \end{cases}$

# Alignement local

Tracé arrière à partir du maximum dans  $T$

Calcul

- en temps  $O(|x| \times |y|)$
- en espace  $O(|x| \times |y|)$

[Smith & Waterman 1981]

- score
- estimation
- méthodes
  - sensibilité pour de petites séquences
  - manipulation de grandes séquences

# Alignement d'une séquence requête avec une banque de séquences

- séquence requête  $x$
- banque de séquence  $Y = \{ y_0, y_1, \dots, y_{k-1} \}$
- méthodes heuristiques

# BLAST [Altschul *et al* 1990]

## Basic Local Alignment Search Tool

- construction de l'automate fini déterministe des voisins fréquentables ( $\geq T$ ) des facteurs de longueur  $W$  de  $x$
- recherche d'un segment de  $y$  contenant le plus de voisins fréquentables de  $x$
- extension de la zone

# Exemple de Blast

- $W = 2$
- $T = 14$
- $x = \text{YWCQPGK}$
- $y = \text{LAWYQQKPGKA}$

facteurs de x

YW

WC

CQ

QP

PG

GK

voisins fréquentables

AW,CW,FW,HW,IW,LW,MW,

NW,SW,TW,VW,WW,YW

RC,WA,WC,WG,WH,WI,

WP,WS,WT,WV,WY

CD,CE,CH,CQ



# Différentes versions

|         | requête                   | banque                    |
|---------|---------------------------|---------------------------|
| BLASTP  | protéine                  | protéine                  |
| BLASTN  | nucléique                 | nucléique                 |
| BLASTX  | protéine                  | nucléique<br>(traduction) |
| TBLASTN | nucléique<br>(traduction) | protéine                  |
| TBLASTX | nucléique<br>(traduction) | nucléique<br>(traduction) |

# Améliorations

PSI-BLAST : Position-Specific Iterative :  
crée un profil pendant la recherche et le raffine  
au fur et à mesure des itérations

# FASTA [Pearson & Lipman 1988]

- tracé ou « dotplot » des facteurs de longueur *ktup* à l'aide de technique de hachage
- sélection de la diagonale contenant le plus de points
- programmation dynamique autour de cette diagonale sur une bande de largeur  $s$

# Exemple

$x = \text{YWCQPGK}$ ,  $y = \text{LAWYQQKPGKA}$ ,  $ktup = 2$

|    | LA | AW | WY | YQ | QQ | QK | KP | PG | GK | KA |
|----|----|----|----|----|----|----|----|----|----|----|
| YW |    |    |    |    |    |    |    |    |    |    |
| WC |    |    |    |    |    |    |    |    |    |    |
| CQ |    |    |    |    |    |    |    |    |    |    |
| QP |    |    |    |    |    |    |    |    |    |    |
| PG |    |    |    |    |    |    |    | X  |    |    |
| GK |    |    |    |    |    |    |    |    | X  |    |

|   |  |   |    |    |    |    |    |    |    |    |    |    |
|---|--|---|----|----|----|----|----|----|----|----|----|----|
|   |  | L | A  | W  | Y  | Q  | Q  | K  | P  | G  | K  | A  |
|   |  |   | -1 | -2 | -3 |    |    |    |    |    |    |    |
| Y |  |   |    | -2 | -1 | -2 |    |    |    |    |    |    |
| W |  |   |    |    | -2 | -2 | -2 |    |    |    |    |    |
| C |  |   |    |    |    | -3 | -3 | -4 |    |    |    |    |
| Q |  |   |    |    |    |    | -2 | -3 | -4 |    |    |    |
| P |  |   |    |    |    |    |    | -3 | -2 | -3 |    |    |
| G |  |   |    |    |    |    |    |    | -3 | -1 | -2 |    |
| K |  |   |    |    |    |    |    |    |    | -2 | 0  | -1 |

YWCQ - PGK

YQ - QKPGK

YWCQ - PGK

Y - QQKPGK

# Alignement multiple

aligner  $k$  séquences  $x_0, x_1, \dots, x_{k-1}$

|       |   |   |   |   |   |   |   |   |   |   |
|-------|---|---|---|---|---|---|---|---|---|---|
| $x_0$ | Y | D | G | G | A | - | V | E | A | L |
| $x_1$ | Y | D | G | G | - | - | - | E | A | L |
| $x_2$ | F | E | G | G | I | L | V | E | A | L |
| $x_3$ | F | D | - | G | I | L | V | Q | A | V |
| $x_4$ | Y | E | G | G | A | V | V | Q | A | L |

# Alignement multiple

aligner  $k$  séquences  $x_0, x_1, \dots, x_{k-1}$

maximiser les sommes des paires

Matrice  $M$  à  $k$  lignes et  $c$  colonnes

$$Score(M) = \sum_{j=0}^{c-1} \sum_{0 \leq i < \ell \leq k-1} score(M[i,j], M[\ell,j])$$