

# Cours 2

Alignement global

# Pourquoi comparer des séquences ?

- ▷ mécanisme d'évolution
- ▷ protéines: même fonction = même moyen
- ▷ **But** : identifier des gènes, des fonctions communes etc.

## Exemple : l'insuline

éléphant FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN  
|||||  
hamster FVNQHLCGSHLVEALYLVCGERGFFYTPKSGIVDQCCTSICSLYQLENYCN

éléphant FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN  
|||||  
baleine FVNQHLCGSHLVEALYLVCGERGFFYTPKAGIVEQCCASTCSLYQLENYCN

éléphant FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN  
|| |||  
alligator AANQRLCGSHLVDALYLVCGERGFFYSPKGGIVEQCCHNTCSLYQLENYCN

# Alignement

- ▷ Mise en correspondance de deux séquences (ADN ou protéines)

```
R D I S L V - - - K N A G I
|   |   | |           | |   | |
R N I - L V S D A K N V G I
```

- ▷ Évolution : 3 événements mutationnels élémentaires

- substitution
  - insertion
  - délétion
- } *indel*

- ▷ Score

- substitution : matrice de similarité
- indel : pénalité

## Matrices de score pour l'ADN

match  $\rightarrow$  1  
mismatch  $\rightarrow$  0

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

match  $\rightarrow$  2  
mismatch  $\rightarrow$  -1

	A	C	G	T
A	2	-1	-1	-1
C	-1	2	-1	-1
G	-1	-1	2	-1
T	-1	-1	-1	2

# Alignement global

*Needleman & Wunsch - 1970*

## Données

- ▷ deux séquences (nucléotides ou acides aminés),
- ▷ des scores de similarité et des pénalités.

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

---

## Problème

Quel est l'alignement de score maximal ?

# Alignons les séquences

A C G G C T A T

A C T G T A T

avec les scores

*match* : 2

*mismatch* : -1

*indel* : -2

*Que peut-il se passer pour la dernière opération?*

## 1. Substitution de T en T

ACGGCTA T  
? ? ? |  
ACTGTA T

ACGGCTA
ACTGTA

 +2

## 2. Délétion de T

ACGGCTA T  
? ? ?  
ACTGTAT -

ACGGCTA
ACTGTAT

 -2

## 3. Insertion de T

ACGGCTAT -  
? ? ?  
ACTGTA T

ACGGCTAT
ACTGTA

 -2

$\text{Sim}(i, j)$  : score optimal entre  $U(1..i)$  et  $V(1..j)$ .

### Formule de récurrence :

$$\left| \begin{array}{l} \text{Sim}(0, 0) = 0 \\ \text{Sim}(0, j) = \text{Sim}(0, j - 1) + \text{Ins}(V(j)) \\ \text{Sim}(i, 0) = \text{Sim}(i - 1, 0) + \text{Del}(U(i)) \\ \text{Sim}(i, j) = \max \begin{cases} \text{Sim}(i - 1, j - 1) + \text{Sub}(U(i), V(j)) \\ \text{Sim}(i - 1, j) + \text{Del}(U(i)) \\ \text{Sim}(i, j - 1) + \text{Ins}(V(j)) \end{cases} \end{array} \right.$$

**Méthode** : programmation dynamique

**Complexité** :  $O(n \times m)$  en temps et en espace

**Étape 1:** *création d'une table indexée par les deux séquences.*

		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

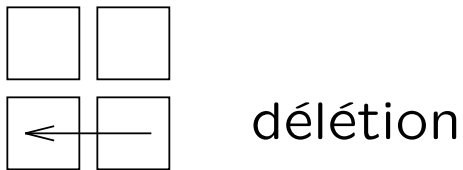
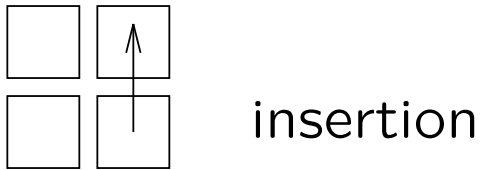
Case  $(i, j)$  : score entre les  $i$  premières bases de ACGGCTAT et les  $j$  premières bases de ACTGTAT.

**Étape 2** : recherche du chemin des scores maximaux dans la matrice.

		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

### Étape 3 : construction de l'alignement

Sur le chemin des scores maximaux, on regarde quelle est l'opération correspondante.



---

### Résultat

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

## Calcul du score avec espace linéaire

- ▷ Le score de l'alignement est connu dès l'**étape 1**.
- ▷ Au lieu de conserver toute la matrice, on travaille sur deux vecteurs : la dernière ligne calculée, et la ligne courante.

*On sait calculer le score entre  $U$  et tous les préfixes de  $V$  en temps quadratique et avec un espace linéaire.*

# Calcul de l'alignement avec espace linéaire

## *Diviser pour régner*

$S$ , séquence de longueur  $m$ ,  $T$  séquence de longueur  $n$

**A**: alignement optimal entre deux séquences

Que peut-il se passer pour  $S(i)$  ?

**Cas 1.**  $S(i)$  est aligné avec un certain  $T(j)$  ( $j \in [1..n]$ )

$$\mathbf{A} \left( \begin{array}{c} S(1..i-1) \\ T(1..j-1) \end{array} \right) \& \left( \begin{array}{c} S(i) \\ T(j) \end{array} \right) \& \mathbf{A} \left( \begin{array}{c} S(i+1..m) \\ T(j+1..n) \end{array} \right)$$

**Cas 2.**  $S(i)$  est déléte :  $S(i)$  est aligné avec un -, situé entre  $T(j)$  et  $T(j+1)$  ( $j \in [0..n]$ )

$$\mathbf{A} \left( \begin{array}{c} S(1..i-1) \\ T(1..j-1) \end{array} \right) \& \left( \begin{array}{c} S(i) \\ - \end{array} \right) \& \mathbf{A} \left( \begin{array}{c} S(i+1..m) \\ T(j..n) \end{array} \right)$$

- Comment déterminer le cas (**1** ou **2**) ?
- Comment déterminer la bonne valeur de  $j$  ?

- Similarité entre  $S(1..i - 1)$  et tous les préfixes de  $T$

Calculable en espace linéaire

- Similarité entre  $S(i + 1..m)$  et tous les suffixes de  $T$

Problème symétrique au précédent

---

**Score :**

Entrée : S,T: Sequence, I: Naturel

Sortie : J: Natural, Cas: Booléen

*Calcule les scores d'alignement de  $S(1..i - 1)$  avec tous les préfixes de  $T$  et les scores d'alignement de  $S(i + 1..m)$  avec tous les suffixes de  $T$ .*

Cas : **Vrai**, si l'alignement optimal correspond au cas 1,  
**Faux**, s'il correspond au cas 2.

J : **indice** correspondant dans  $T$

## Récapitulation

- ▷ Division du problème d'alignement entre  $S$  et  $T$  en deux sous-alignements, suivant  $S(i)$  et  $T(j)$
- ▷  $i$  est fixé et  $j$  est déterminé en fonction de  $i$
- ▷ Conclusion avec deux appels récursifs
- ▷ **Quel indice choisir pour  $i$  ?**

```

function Align(S,T:Sequence) return Alignement is
  M:Natural:= longueur de S;
  N:Natural:= longueur de T;
begin
  if M=0 then
    return (1..N =>'-', T);
  elsif N=0 then
    return (S, 1..M =>'-');
  else
    Score(S, T, M/2, J, Cas);
    if Cas then -- cas 1
      return Align(S(1..M/2-1),T(1..J-1))
        &(S(M/2), T(J))
        &Align(S(M/2+1..M),T(J+1..N));
    else -- cas 2
      return Align(S(1..M/2-1),T(1..J))
        &(S(M/2), '-')
        &Align(S(M/2+1..M),T(J+1..N));
    end if;
  end if;
end Align;

```

**Complexité ?**

# Comment construire proprement une matrice de scores ?

*Un exemple intermédiaire (en dimension 1)*

- ▷ Les séquences protéiques ont des zones **hydrophiles** et **hydrophobes** qui conditionnent la structure de la molécule.
- ▷ Ces zones sont caractérisées par un fort biais de composition :

hydrophobe → Ile, Leu, Val, Phe

hydrophile → Asn, Glu, Gln, His, Lys, Arg

## Données

- Modèle de base  $\mathcal{P}$ :  $p_i$ , fréquence d'apparition de chaque acide aminé dans une protéine
- Modèle cible  $\mathcal{Q}$ :  $q_i$ , fréquence d'apparition de chaque acide aminé dans une zone hydrophile

fréquences données par les échantillons

## Problème

Quel score attribuer à chaque acide aminé pour discriminer les zones hydrophiles des autres?

▷ La séquence  $U = u_1 \dots u_n$  vue comme un **message** à transmettre dans un alphabet de transmission (taille  $k$ )

▷ Longueur du codage optimal de  $U$  (Shannon):

$$- \sum_{i=1}^n \log_k(f_i)$$

$f_i$  fréquence d'apparition de  $u_i$ .

▷ Revenons aux protéines

$$U \in \mathcal{Q}: \text{codage}(U, \mathcal{Q}) < \text{codage}(U, \mathcal{P})$$

---

Score de  $u_i$  :  $\log(q_i/p_i)$

- ▷ Les positions dans  $U$  sont indépendantes
- ▷ Probabilité de  $U$  dans le modèle  $\mathcal{Q}$ :  $q_1 \times \dots \times q_l$
- ▷ Probabilité de  $U$  dans le modèle  $\mathcal{P}$ :  $p_1 \times \dots \times p_l$
- ▷  $U \in \mathcal{Q}$ :  $\text{proba}(U, \mathcal{P}) < \text{proba}(U, \mathcal{Q})$

---

Score de  $u_i$  :  $\log(q_i/p_i)$

# Application à l'alignement

- ▷ un alignement sans indels : mot composé de couples

```
U=  C A G G T C T A
      |   | | |   |
V=  C T G G T A T C
```

- ▷ modèle cible : zone de forte similarité
- ▷ modèle de base : alignement quelconque

Matrice "**log odd ratios**" :

$$\text{Score de } (u_i, v_j) : \log \frac{q_{i,j}}{p_i p_j}$$

$q_{i,j}$  : fréquence d'alignement de  $u_i$  avec  $v_j$  dans une zone de forte similarité

$p_i, p_j$  : fréquence d'apparition de  $u_i$ , de  $p_j$  dans le modèle de base

*Exemples* : matrices PAM, matrices BLOSUM

**Et pour les gaps ?** empirique