

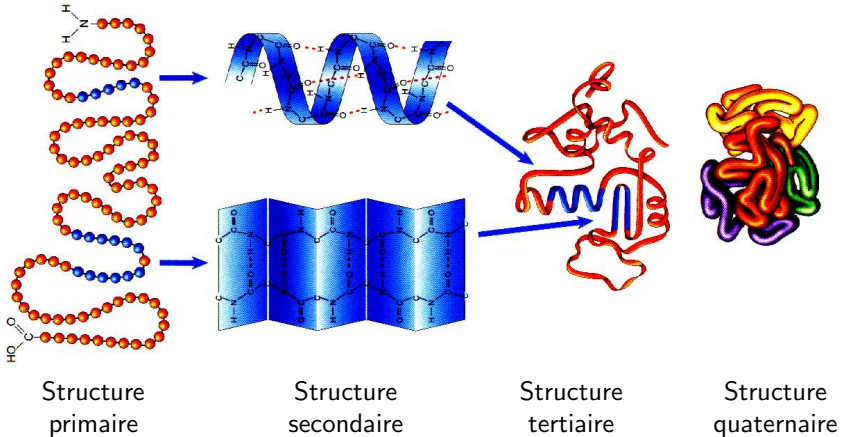
Pourquoi comparer des séquences ?

- ▶ Les programmes de séquençage fournissent des **séquences** qu'il faut annoter

“Public Collections of DNA and RNA Sequence Reach 100 Gigabases”
(août 2005)

- ▶ Recherche d'homologie
La similitude syntaxique est un signe de proximité fonctionnelle.
- ▶ Recherche de fonction commune
Les régions conservées correspondent à des régions fonctionnellement importantes.
- ▶ Prédiction de gènes

Structure des protéines



Swissprot:163235 séquences
TrEMBL:1449374 séquences

PDB : 243 structures

Structure des protéines

- ▶ Les structures 3D des protéines sont longues et coûteuses à déterminer

Cristallographie, résonance magnétique nucléaire

- ▶ Dans **PDB**, toutes les protéines avec plus de 25 % d'identité partagent la même structure

PDB: Protein Data Bank -

banque de structures de protéines résolues expérimentalement

Exemple : l'insuline

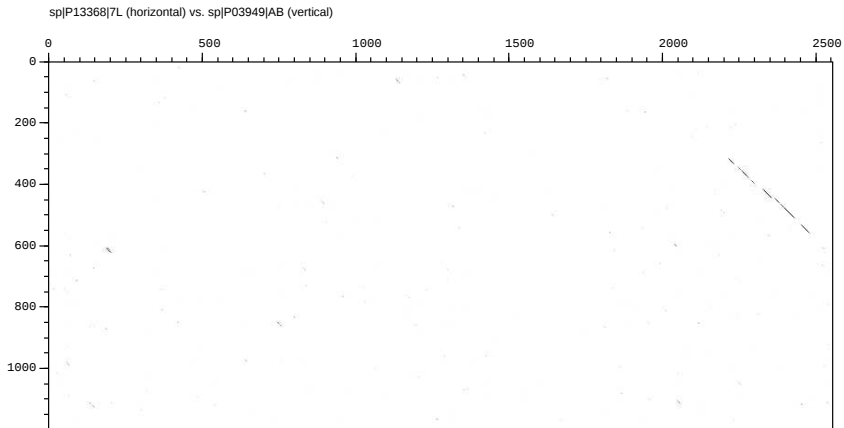
éléphant	FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN
hamster	FVNQHLCGSHLVEALYLVCGERGFFYTPKSGIVDQCCTSIQSLYQLENYCN
éléphant	FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN
baleine	FVNQHLCGSHLVEALYLVCGERGFFYTPKAGIVEQCCASTCSLYQLENYCN
éléphant	FVNQHLCGSHLVEALYLVCGERGFFYTPKTGIVEQCCTGVCSLYQLENYCN
alligator	AANQRLCGSHLVDALYLVCGERGFFYSPKGGIVEQCCHNTCSLYQLENYCN

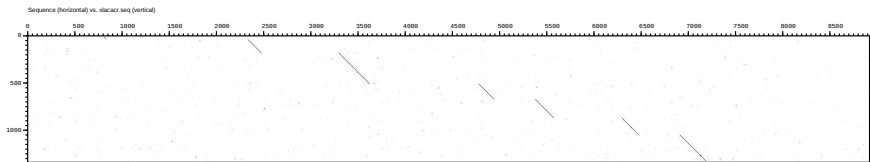


horizontalement : ADN codant pour la chaîne α de l'hémoglobine humaine

verticalement : ADN codant pour la chaîne β de l'hémoglobine humaine

Deux enzymes avec un domaine catalytique commun





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

- 3 événements mutationnels élémentaires

- *substitution*
 - *insertion*
 - *délétion*
- } indel

- Score d'une opération

- *substitution : score de similitude*
- *indel : pénalité*

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

- 2 séquences → plusieurs alignements possibles

```

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

```

```

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

```

- Bon/mauvais alignement? *matrices de substitutions*

Mismatch :	Match :	Indel :
	G, N : 6	
DN : 1	R, K : 5	—5
AV, LD : 0	A, I, L, S, V : 4	

- ▶ 2 séquences → plusieurs alignements possibles

```

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

```

```

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

```

Scores : 19, -11 et 25 respectivement

- ▶ Bon/mauvais alignement? *matrices de substitutions*

Mismatch :	Match :	Indel :
	G, N : 6	
DN : 1	R, K : 5	-5
AV, LD : 0	A, I, L, S, V : 4	

Alignement global

Needleman & Wunsch - 1970

Évaluation d'une ressemblance globale entre deux séquences

Données

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

Problème

Quel est l'alignement de score maximal ?

Algorithme

Aligner les séquences ACGGCTAT et ACTGTAC avec les scores $match = 2$, $mismatch = -1$ et $indel = -2$.

Que peut-il se passer pour la dernière opération?

► **Substitution** de T en C

ACGGCTA T
? ? ? |
ACTGTA C

score de

ACGGCTA
ACTGTA

 -1

► **Délétion** de T

ACGGCTA T
? ? ?
ACTGTAC -

score de

ACGGCTA
ACTGTAC

 -2

► **Insertion** de C

ACGGCTAT -
? ? ?
ACTGTA C

score de

ACGGCTAT
ACTGTA

 -2

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

► **Formule de récurrence :**

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

► **Méthode :** programmation dynamique

Programmation dynamique

- ▶ Un algorithme de *programmation dynamique* procède en réduisant le problème à plusieurs instances plus petites, elle-mêmes résolues par décomposition.
- ▶ Les résultats des calculs intermédiaires sont stockés dans une table.
- ▶ La solution est ensuite construite à partir de la table, en remontant celle-ci.

Ici :

$$\begin{array}{c} \textit{calculs intermédiaires} \\ = \\ \textit{scores d'alignements entre préfixes} \end{array}$$

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

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

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

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								
C	-4								
T	-6								
G	-8								
T	-10								
A	-12								
T	-14								

Cas de base - initialisation

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								
T	-6								
G	-8								
T	-10								
A	-12								
T	-14								

Remplissage ligne par ligne

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

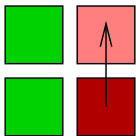
Remplissage ligne par ligne

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

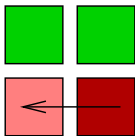
		A	C	G	G	C	T	A	T
	0	-2	-4	-6	-8	-10	-12	-14	-16
A	-2	2	0	-2	-4	-6	-8	-10	-12
C	-4	0	4	2	0	-2	-4	-6	-8
T	-6	-2	2	3	1	-1	0	-2	-4
G	-8	-4	0	4	5	3	1	-1	-3
T	-10	-6	-2	2	3	4	5	3	1
A	-12	-8	-4	0	1	2	3	7	5
T	-14	-10	-6	-2	-1	0	4	5	9

Étape 3 : *construction de l'alignement*

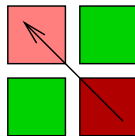
Sur le chemin des scores maximaux, on regarde quelle est l'opération correspondante.



insertion



délétion



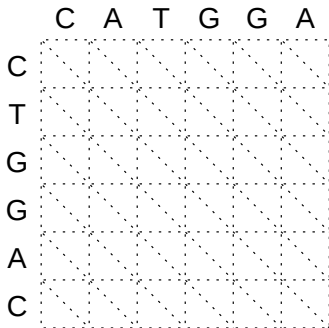
substitution
ou
identité

Résultat

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

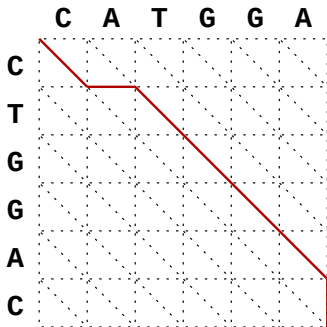
Graphe d'édition pour l'alignement de séquences

- ▶ Le problème peut également être vu comme la recherche d'un chemin optimal dans un graphe.
- ▶ Grille à deux dimensions
- ▶ Trois types d'arc : délétion, insertion et substitution



Graphe d'édition pour l'alignement de séquences

- ▶ Le problème peut également être vu comme la recherche d'un chemin optimal dans un graphe.
- ▶ Grille à deux dimensions
- ▶ Trois types d'arc : délétion, insertion et substitution



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

Complexité de l'algorithme

- ▶ Pour le calcul du score d'alignement : (**étape 1**)
 - ▶ $O(n \times m)$ en temps
 - ▶ $O(\min\{n, m\})$ en espace
- ▶ Pour la construction de l'alignement : (**étapes 1, 2 et 3**)
 - ▶ $O(n \times m)$ en temps et en espace
- ▶ Optimisation pour la construction de l'alignement avec espace linéaire (en $O(n)$). (*Myers & Millers - 1988*)

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 $T(j)$ ($j \in [1..n]$)

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

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

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

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

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

Calculable en espace linéaire

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

Problème symétrique au précédent - calculable en espace linéaire

- ▶ Fonction **Score**

Entrée : les deux séquences S et T , l'indice I dans S

Sortie : l'indice J dans T et le Cas, 1 ou 2 (Booléen)

Cas : **Vrai**, si l'alignement optimal correspond au cas 1
(substitution de $S(i)$)

Faux, s'il correspond au cas 2 (délétion de $S(i)$)

J : **indice** correspondant dans T

- ▶ Complexité de **Score** ?

Récapitulation

- ▶ Division du problème d'alignement entre S et T en deux sous-alignements, coupés en $S(i)$ 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 Aligned 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: substitution de S(i)
      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: deletion de S(i)
      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 ?

Alignement local

Smith & Waterman -1981

Données

- ▶ deux séquences (nucléotides ou acides aminés),
- ▶ des scores de similitude.

Problème

Quelles sont les régions de forte similarité entre les deux séquences ?

Exemple : GGCTGACCACCTTGTA et GATCACTTCCATGGCAGTA

► Alignement global :

1	G	G	C	T	G	A	C	C	A	C	C	-	T	T	G	T	A	-	-	-	16
1	G	A	-	T	C	A	C	T	T	C	C	A	T	G	G	C	A	G	T	A	19

Les séquences présentent une similarité que l'alignement global ne révèle pas.

Exemple : GGCTGACCACCTTGTA et GATCACTTCCATGGCAGTA

► Alignement global :

1	G	G	C	T	G	A	C	C	A	C	C	-	T	T	G	T	A	-	-	-	16
1	G	A	-	T	C	A	C	T	T	C	C	A	T	G	G	C	A	G	T	A	19

Les séquences présentent une similarité que l'alignement global ne révèle pas.

► Alignement local :

5	G	A	C	C	A	C	C	T	T	13	14	G	T	A	16
1	G	A	T	C	A	C	-	T	T	8	17	G	T	A	19

$\text{Loc}(i, j)$: score optimal entre un suffixe de $U(1..i)$ et un suffixe $V(1..j)$.

Formule de récurrence :

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

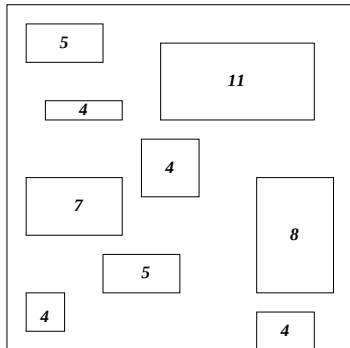
Implémentation, complexité : cf alignement global (programmation dynamique)

Recherche du résultat

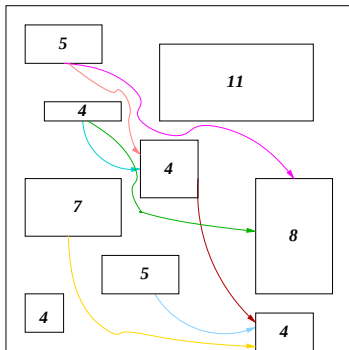
	G	G	C	T	G	A	C	C	A	C	C	T	T
G	2	1	0	0	2	1	0	0	0	0	0	0	0
A	1	1	0	0	0	4	3	2	2	1	0	0	0
T	0	0	0	2	1	0	3	2	1	0	0	2	2
C	0	0	2	1	1	0	2	5	4	3	2	1	1
A	0	0	1	1	0	3	2	4	7	6	5	4	3
C	0	0	2	1	0	2	5	4	6	9	8	7	6
T	0	0	1	4	3	2	4	4	5	8	8	10	9
T	0	0	0	3	3	2	3	3	4	7	7	10	12
C	0	0	2	2	2	2	4	5	4	6	6	9	11
C	0	0	2	1	1	1	4	6	4	6	8	8	10
A	0	0	1	1	0	3	3	5	8	7	7	7	9
T	0	0	0	3	2	2	2	4	7	7	6	9	11
G	2	2	1	2	5	4	3	3	6	6	6	8	10

- ▶ La zone de plus forte similarité : score maximal
- ▶ Les zones de similarités au delà d'un seuil
- ▶ **Plus difficile:** les zones de forte similarité compatibles entre elles

Les zones de forte similarité compatibles



Les zones de forte similarité compatibles



- ▶ Deux alignements sont compatibles si :
 - ▶ ils ne créent pas de croisement
 - ▶ ils ne se chevauchent pas
- ▶ Recherche du chemin de poids maximal dans un graphe