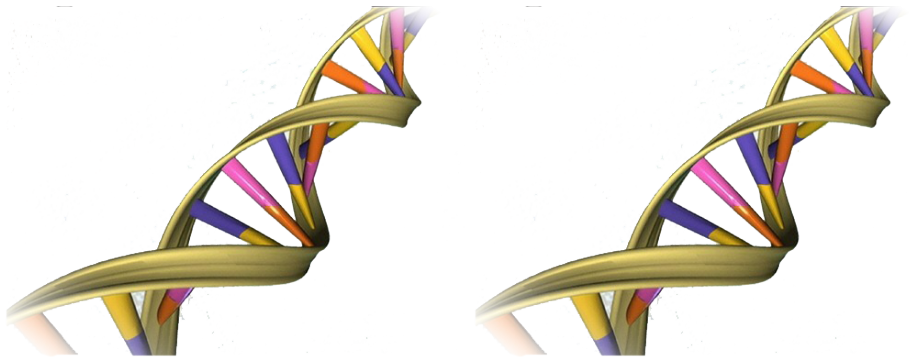
Alignements de deux séquences

Module Informatique Génomique - Licence 3



http://igm.univ-mlv.fr/~fsikora/ens/2008-2009/L3/genomique/

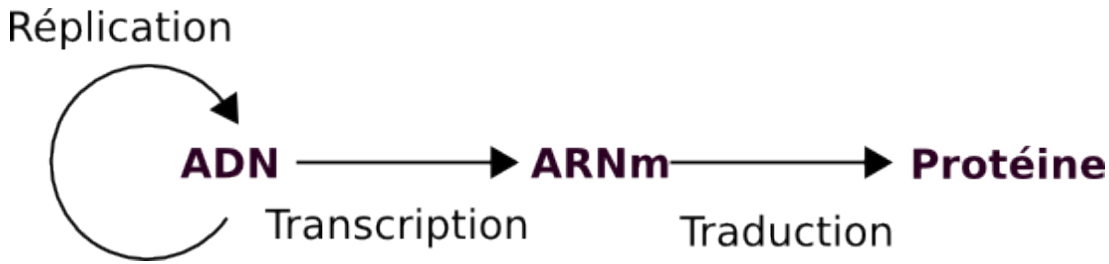
Plan

Plan

Qu'est-ce qu'une séquence ?

Rappels sur les séquences génomiques

- Enchaînement de nucléotides le long d'une macromolécule d'ADN.
- Représentée par une chaîne de caractères utilisant l'alphabet {A,C,G,T} - qui distingue les quatre types de nucléotides.

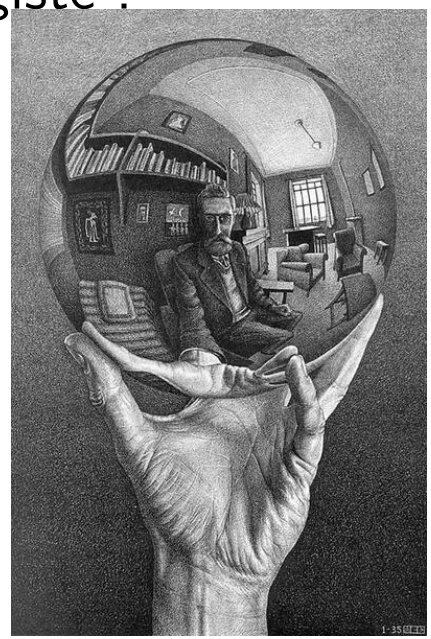


Pourquoi comparer des séquences ?

- Déterminer les similitudes éventuelles entre deux séquences génomiques ou protéiques.
- But : inférer des connaissances sur une nouvelle séquence à partir des connaissances sur d'autres séquences proches
  - ce qui se ressemble s'assemble ...
  - si la fonction d'une séquence est connue, la fonction de la seconde peut s'en déduire.

Pourquoi comparer des séquences ?

- Stockage des connaissances dans des bases de données.
- Lors de la découverte d'une sequence par un biologiste : recherche des séquences similaires.
- La comparaison de séquences permet également
  - de prédire des gènes
  - de déterminer la fonction d'une protéine
  - de prédire des structures
  - ...



D'où viennent les similitudes entre séquences ?

- Sources des différences entre les séquences :
  - substitution d'un nucléotide par un autre
  - disparition d'un nucléotide
  - apparition d'un nucléotide
- Propagation de ces modification au sein des populations par héritage génétique.
- Evolution des génomes au cours du temps.

D'où viennent les similitudes entre séquences ?

Histoire des espèces

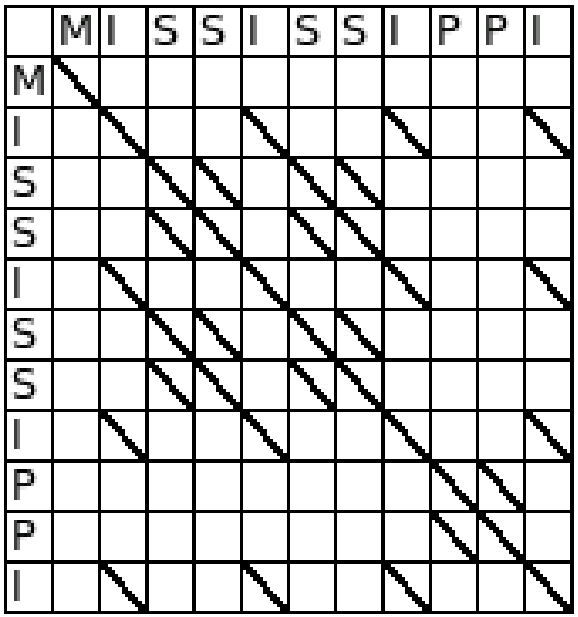
- Représentable par un arbre dont les feuilles sont les espèces actuelles.
- Deux espèces sont considérées d'autant plus proches que leur espèce ancestrale commune est récente.

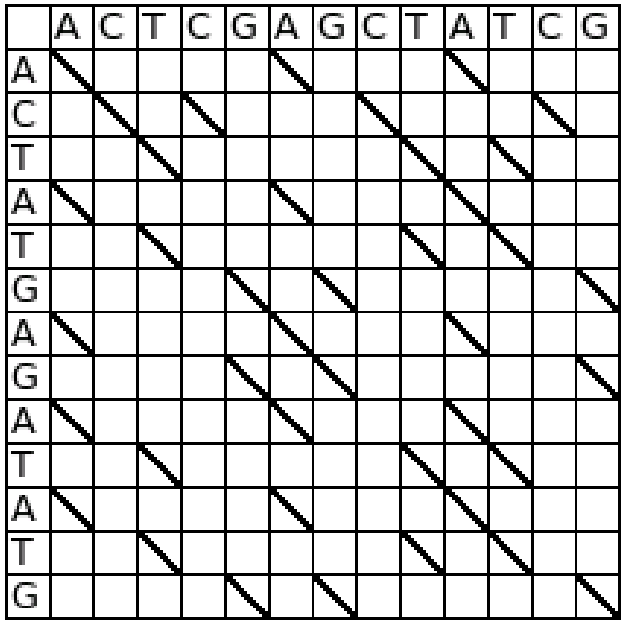
Plan

Les mesures de similarité

Vers une première solution - Dotplot

- Staden 1982 : Dotplot (matrice de points)

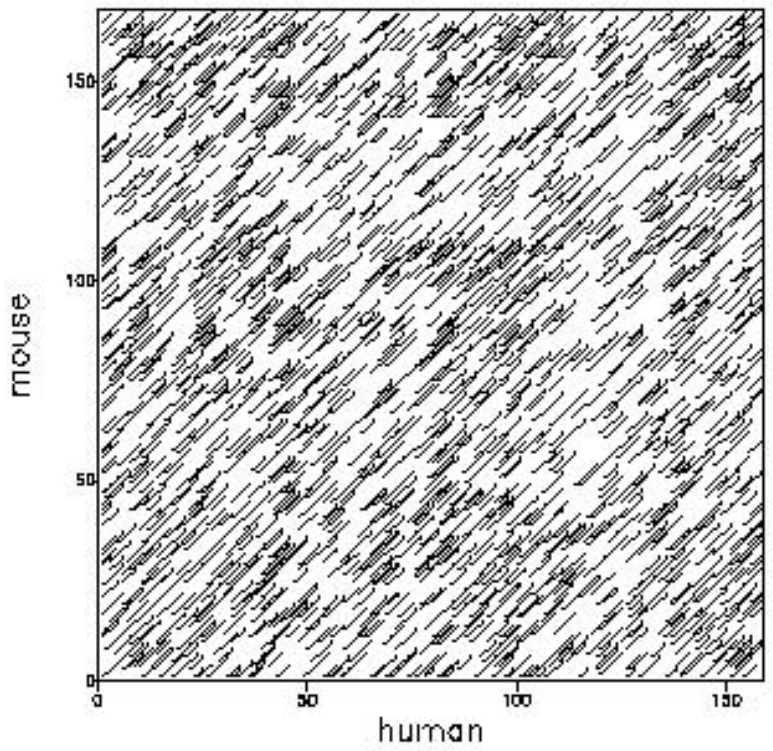




Les mesures de similarité

Limites de Dotplot

- Principalement, la lisibilité !
- Les petites similarités créent du *bruit*.
- Il existe des techniques pour le limiter
- Il suffit par exemple de ne considérer que les diagonales de taille $> k$



Les mesures de similarité

Distance de Hamming

- C'est la mesure de similarité triviale entre deux séquences de même longueur (nombre de différences).
- Formellement : $d(a,b) = \sum_{i=0}^{n-1}(a[i] \oplus b[i])$

Exemple

Comparaison simple d'une paire de séquences

- *(a)* `AGTATC` *et* `AGATGC`*; 3 différences*
- *(b)* `AGTTTC` *et* `AGATTC`*; 1 différence*

Les séquences de la paire (b) sont considérées comme étant plus similaires que celles de la paire (a) si on considère la distance de Hamming.

Les mesures de similarité

Limite de la distance de Hamming

- Les séquences à comparer ont rarement la même longueur.
- Même si c'est le cas, rien ne dit qu'elles doivent être comparées sur cette longueur exactement.
- Dans le cadre de séquences génomiques, des nucléotides ou des acides aminés ont pu s'insérer ou au contraire disparaître au cours de l'évolution (cascade de différences).

Alignement et distance d'édition

- Alignement = mise en correspondance de 2 séquences lettre à lettre.
- Pour tenir compte des insertions ou délétions éventuelles, il faut introduire un caractère particulier, appelé *gap* et noté $-$.

Exemple

$$(a)\quad \begin{matrix} AG-ATGCT \\ AGTAT-C- \end{matrix} \qquad (b)\quad \begin{matrix} AGAT-GCT \\ AG-TATC- \end{matrix}$$

Coût d'un alignement

Calcul du coût

- Il faut fixer le coût d'insertion d'un gap (appelé *indel*).
- Le coût d'un alignement (S', T') de S et T correspond alors à $cout(S', T') = \sum_{i=0}^{|S'|-1} \alpha(S'[i], T'[i])$ *tq*

$$\alpha(S'[i], T'[i]) = \begin{cases} sub, & si\ S'[i] \neq T'[i] \\ indel, & S'[i]\ ou\ T'[i]\ est\ un\ gap \\ 0, sinon & \end{cases}$$

Exemple

Avec $indel = 2$ *et* $sub = 1$,

S'	AG-ATGCT	AGAT-GCT
T'	AGTAT-C-	AG-TATC-
Coût	6	7

Alignement optimal

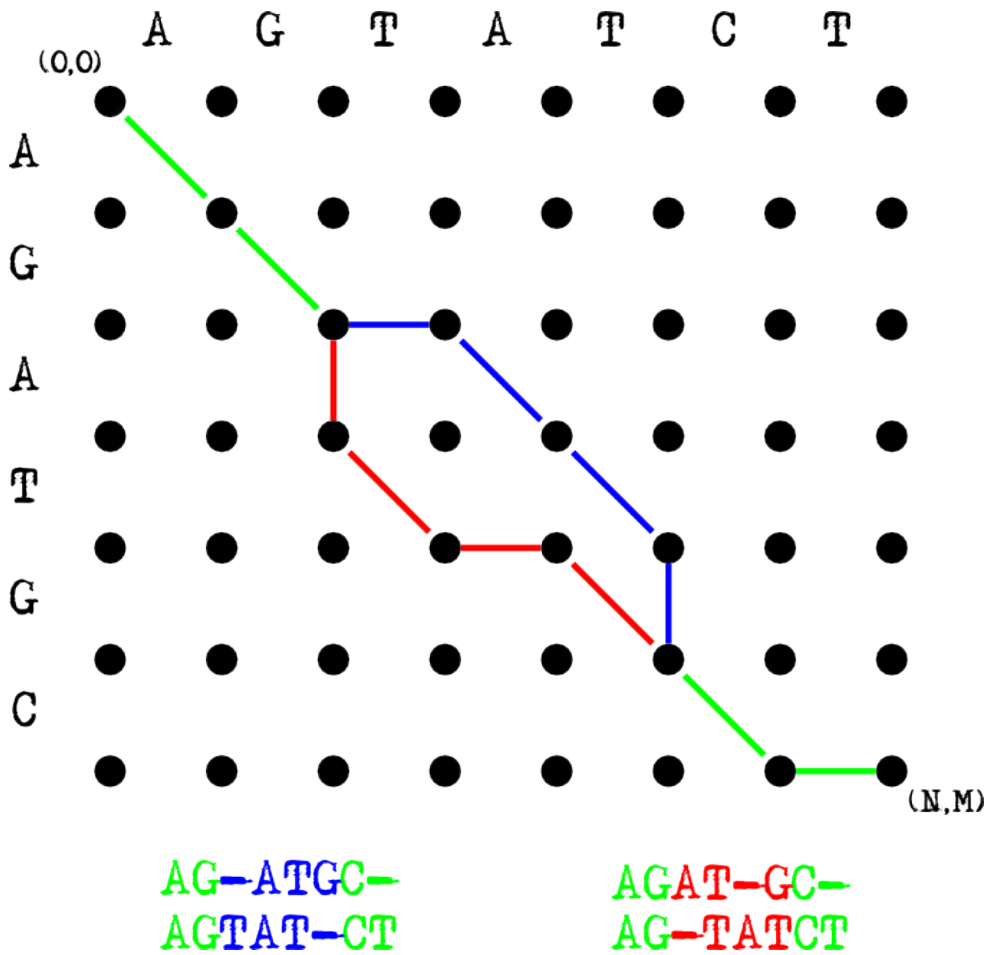
Calcul des alignements optimaux

- Parmi tous les alignements possibles, le principe est de retenir ceux (il peut y en avoir plusieurs) dont le coût est minimal.
- Un alignement de coût minimal est dit optimal.
- La similarité des deux séquences se mesure alors par ce coût minimal.

Plan

Une représentation graphique d'un alignement

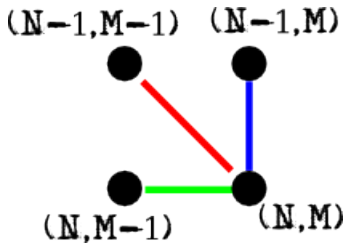
Le graphe d'édition



Algorithme de recherche

Algorithme

- Un alignement (ou chemin) optimal de longueur L s'obtient en calculant un alignement partiel (ou sous-chemin) optimal de longueur L-1.
- Or, il n'existe que trois sous-chemins de longueur L-1 du chemin arrivant au noeud (N,M) :
  - celui venant du noeud (N,M-1) ;
  - celui venant du noeud (N-1,M-1) et
  - celui celui venant du noeud (N-1,M).



Exemple

ACGGCTA T
????? |
ACTGTA T

ou

ACGGCTA T
?????
ACTGTAT −

ou

ACGGCTAT −
?????
ACTGTA T

Algorithme de recherche

Formule de récurrence

On note $Cout(i, j)$ le coût optimal entre $S[0..i]$ et $T[0..j]$

- $Cout(0, 0) = 0$
- $Cout(0, j) = Cout(0, j-1) + indel$
- $Cout(i, 0) = Cout(i-1, 0) + indel$
- $$Cout(i, j) = min \begin{cases} Cout(i-1, j-1) + sub(S[i], T[j]) \\ Cout(i, j-1) + indel \\ Cout(i-1, j) + indel \end{cases}$$

Que se passe-t-il si on implémente ces formules de manière récursive ?

Plan

Alignement optimal

Programmation dynamique – Neddleman et Wunsch (1970)

- Avec la méthode récursive, coût d'un même chemin calculé plusieurs fois inutilement : complexité exponentielle.
- Utilisation de la programmation dynamique : stockage des résultats intermédiaires dans une table : complexité polynomiale.
- Reste une méthode exacte.

Alignement optimal

Programmation dynamique

- Idée : déterminer la solution optimale d'un problème à partir de la solution optimale d'un sous-problème.
- Parfait dans notre cas : l'alignement optimal de deux séquences de longueur L est égal à l'alignement optimal de ces séquences de longueur L-1, plus l'alignement de la lettre L.
- Méthode algorithmique souvent utilisée ("fibonnaci", voyageur de commerce, multiplication de matrices,...)

Alignement optimal

L'alignement global

Evaluation d'une ressemblance globale entre deux séquences.

- Données :
  - Deux séquences
  - Des coûts pour les opérations d'édition
- Problème :
  - Trouver un alignement optimal ?

Alignement optimal

1ere étape

- Consiste à créer une table indéxée par les deux séquences
- On remplit chaque case en utilisant les données calculées dans les cases précédentes.

		A	G	T	A	T
	0	1	2	3	4	5
A	1					
G	2					
A	3					
T	4					

- Exemple avec deux séquences de tailles N et M
- Coût indel et coût substitution de 1.

Alignement optimal

1ere étape

		A	G	T	A	T
	0	1	2	3	4	5
A	1	0	1	2	3	4
G	2	1	0	1	2	3
A	3	2	1	1	1	2
T	4	3	2	1	2	1

Alignement optimal

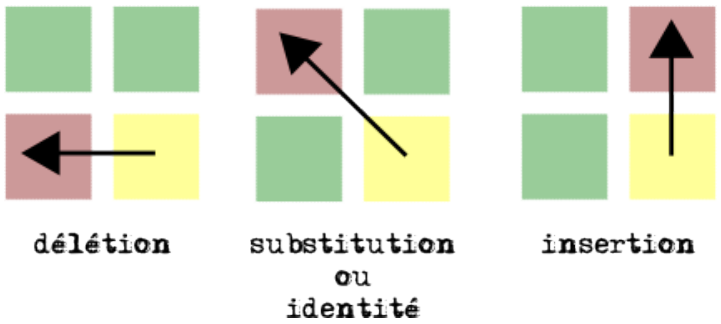
1ere étape

- Finnalement, on obtient un tableau à $N+1$ lignes et $M+1$ colonnes.
- En case (i, j) : coût du meilleur alignement impliquant les i premiers caractères de la première séquence et les j premiers de la seconde.
- Donc, en (N, M) : coût de tout alignement optimal des séquences complètes.
- Mais ce tableau ne dit pas explicitement quels sont les alignements optimaux.

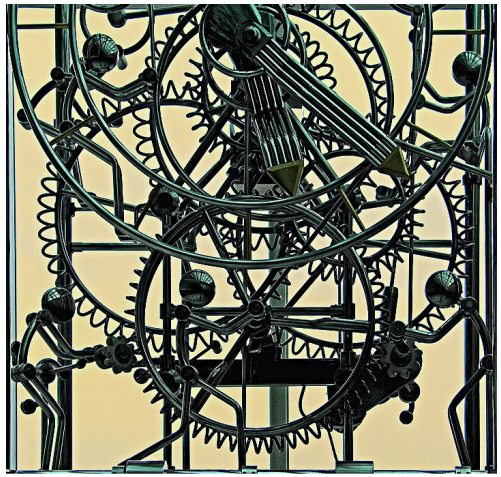
Alignement optimal

2nde étape

- Pour connaître l'alignement optimal, il faut appliquer un algorithme de *backtracking*.
- Partir de (N, M) en choisissant à chaque fois la direction qui ramène au noeud précédent jusqu'à arriver en $(0, 0)$
- Le chemin ainsi tracé fournit l'alignement.



Alignement optimal



Complexité de l'algorithme

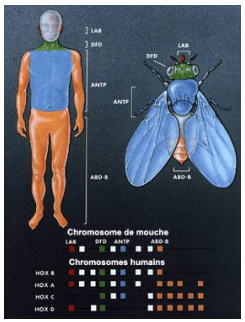
- Pour le calcul du score d'alignement :
  - (Etape 1) – $O(nm)$ en temps, $O(min\{n, m\})$[1] en espace
- Pour la construction de l'alignement :
  - (Etapes 1 et 2) – $O(nm)$ en temps et en espace

[1] Deux lignes ou colonnes sont effectivement nécessaires

Plan

Conclusion

- Comparaison de séquences très utilisé en bioinformatique.
- Une méthode exacte pour l'alignement global par programmation dynamique à complexité polynomiale.
- Méthode similaire pour l'alignement local.
- Existe des heuristiques plus rapides (BLAST, FASTA,...).
- Généralisation de la programmation dynamique à l'alignement de k séquences impossible en pratique.



Bibliographie

Références

1. D. Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.
2. M.S. Waterman. *Introduction to Computational Biology*. Chapman & Hall, 1995.
3. J. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing Co, 1997.
4. M. Crochemore, C. Hancart and T. Lecroq. *Algorithmique du texte*. Vuibert, 2001.