4. Alignement

Quand on a cloné et séquencé un ou plusieurs gènes/cDNA, la première étape est souvent une recherche dans les bases de données publiques pour savoir:

• Si les séquences sont déjà connues
• Si des séquences proches existent.
• Quelles sont les informations déjà connues sur ces séquences similaires.

La recherche d'identité consiste à retrouver les zones identiques entre deux séquences. La comparaison de séquences biologiques, ainsi que leur alignement, nécessite la mise en oeuvre de procédures de calcul et de modèles biologiques permettant de quantifier la notion de ressemblance ou similitude entre ces séquences.
Une ressemblance entre séquences peut indiquer par exemple :
- une fonction biologique proche
- une structure tridimensionnelle semblable
- une origine commune
- etc....

Aligner deux séquences, c'est rechercher le maximum d'appariements entre les lettre qui les composent (nucléotides ou résidus d'acides aminés) avec le minimum de mésappariement et de brèches (gap).
Les pénalités des brèches doivent être suffisamment coûteuses pour éviter les alignement sans signification biologiques.
Le coût d'extension d'une brèche déjà ouverte est généralement plus faible par rapport a celui de son ouverture
La recherche de similitude entre séquences nécessite la détermination d'un score de similarité
Score= ∑ scores élémentaires – ∑ pénalités

4.1. Evaluation d'un Alignement

Système de scores

Score simple

- Un moyen simple (mais pas le meilleur) d'estimer un alignement est de compter 1 pour chaque match et 0 pour chaque mismatch.

```
CGAGGCACAACGTCA
|||  |||   ||||||
CGATGCAAGACGTCA
```

Score : 12

```
ATTGGACAGCAATCAGG
|     ||   |     |
ACGATGCAAGACGTCAG
```

Score : 5

Exemple; (matrice unitaire)

- 1 pour chaque match, 0 pour chaque mismatch et -1 pour une brèche

```
ATGACTGGGCCACT        ATGACTGGGCC- ACT
ATACTGGGACAACT        AT- ACTGGGACAACT
```

Bonus/Malus pour ouverture/extension de gap

Coûts de création d'un gap.

- L'ouverture d'un gap est peu fréquente dans l'évolution, on la pénalise.
- Par contre, on peut avoir eu un événement qui a délété plusieurs résidus d'un coup. Le coût de l'extension est donc moins fort.

Exemple

- Deux alignements avec le même nombre d'indels, : mais avec une distribution différente. L'alignement de gauche sera donc préféré.

```
CGATGCAGCAGCAGCATCG    CGATGCAGCAGCAGCATCG
| ||||       |||||||   ||  ||  ||||  ||   ||  |
CGATGC------AGCATCG    CG-TG-AGCA-CA--AT-G
```

Malus pour ouverture d'indel : Compté chaque fois qu'on crée un indel.
Malus pour extension d'indel : Compté chaque fois qu'on allonge un indel d'une position.

4.2. Alignement de deux séquences
Les principales méthodes

- ***Needleman-Wunch*** : L'alignement selon Needleman-Wunch: l'alignement global.
- ***Smith-Waterman*** : L'algorithme de Smith et Waterman (1981): peut renvoyer des alignements locaux.
- **Les méthodes heuristiques :** Les méthodes précédentes sont exactes mais très gourmandes en temps calcul. Des méthodes approchées (heuristiques) qui vont trouver soit la meilleure réponse, soit une très bonne réponse ont donc été développées.
  1. FASTA : Algorithme très performant et rapide.
  2. *BLAST :* (Altschul *et al.*, 1990), Alignement performant et très rapide. De nombreuses versions sont disponibles pour des recherches spécialisées.

Alignement global et alignement local
1. Méthodes globales

Ce sont des méthodes qui considèrent les séquences dans leur totalité et aboutissent à un alignement de toute la première séquence avec la seconde. Si les longueurs des séquences sont différentes, alors des insertions devront être faites dans la séquence la plus petite pour arriver à aligner les deux séquences d'une extrémité à l'autre.

Exemple Needleman et Wunsch
Ce fût le premier programme de comparaison de séquences, publié en 1970. Il ne calcule pas la différence entre deux séquences mais la similarité.

2. Méthodes locales
Cependant dans un alignement global, si uniquement de courts segments sont très similaires entre deux séquences, les autres parties des séquences risquent de diminuer le poids de ces régions. C'est pourquoi d'autres algorithmes d'alignements, dits locaux, basés sur la localisation des similarités sont nés. Le but de ces alignements locaux est de trouver sans prédétermination de longueur les zones les plus similaires entre deux séquences. L'alignement local comporte donc une partie de chacune des séquences et non la totalité des séquences comme dans les alignements globaux.

Matrices de substitution
Dans tous les programmes de ressemblance, un système de score qui attribue un coût aux opérations élémentaires (identité, substitution, délétion et insertion) est défini.
Ces matrices seront donc fonction :

- de la nature des séquences (nucléique ou protéique)
- de la définition de la ressemblance : soit distance, soit similarité
- des propriétés ou des relations des lettres (nucléotide ou aminoacide) de la séquence que l'on veut mettre en évidence dans la ressemblance: par exemple des propriétés physicochimiques, des relations de structure, des relations d'homologie, etc.

Les Matrices de Substitution :
Dans les protéines certaines mutations sont plus tolérables que d'autres.
Dans l'ADN, les transitions sont plus probables que les transversions (mais cela est peu utilisé)
Les matrices de substitution indiquent le score qui sera retenu pour chaque possibilité de substitution.

1. Matrices pour l'ADN

Les plus utilisées sont:

Matrice unitaire identité:

Dans cette matrice, on attribue la valeur de 1 lorsque les deux nucléotides sont identiques et zéro s'ils ne le sont pas.

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

Matrice de transition/transversion :

- identité : score de 3
- transition : purine (A, G)/ purine, pyrimidine(C,T)/ pyrimidine : score de 1
- transversion : purine/pyrimidine : score de 0

	A	C	G	T
A	**3**	0	***1***	0
C	0	**3**	0	***1***
G	***1***	0	**3**	0
T	0	***1***	0	**3**

Recherche de segments identiques : La matrice de points

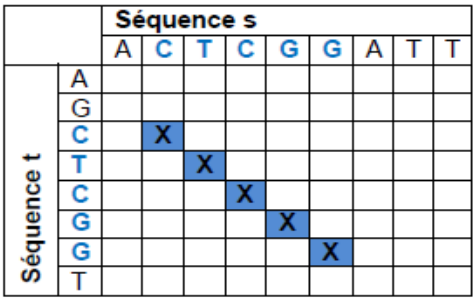
Elle permet une vue (méthode visuelle) englobant les similarités entre les régions des séquences à comparer.

Exemple :
Seq x=ACTCGGATT
seq y=AGCTCGGT

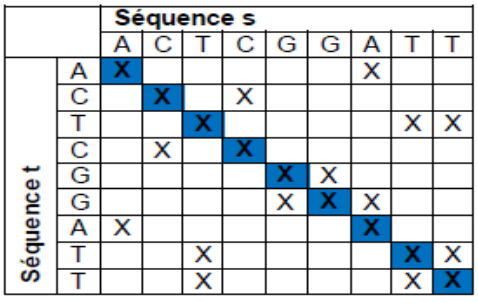
		Séquence s								
		A	C	T	C	G	G	A	T	T
Séquence t	A	X						X		
	G					X	X			
	C		X		X					
	T			X					X	X
	C		X		X					
	G					X	X			
	G					X	X			
	T			X					X	X

Cette méthode consiste à créer une matrice qui va contenir les deux séquences (la séquence x en horizontal et la séquence y en vertical) et de cocher les cases de cette matrice pour le seul cas où les nucléotides sont identiques (Match). Quand il n'ya pas identité on parle de Mismatch.

Sur cette matrice, constatons qu'il y a une diagonale formée de cinq cases. Donc le segment identique le plus long entre les deux séquences x et y contient cinq nucléotides identiques et consécutifs qui sont: **CTCGG**



Dans le cas où les deux séquences sont complètement identiques, le résultat est une diagonale :



2. Matrices pour les protéines

Plusieurs pondérations ont été proposées pour élaborer des matrices de substitution, basées sur:

- propriétés chimiques des chaînes latérales.
- fréquence d'apparition des aminoacides dans les structures secondaires.
- distance génétique: en relation avec le nombre de base à modifier dans le codon pour la substitution.
- fréquence de substitution observée après superposition de structure 3D
- fréquence de substitution observée dans une série de protéines homologues

Les matrices les plus utilisées et reconnues comme les plus performantes sont :

- la série des PAM (Dayhoff 1978)
- la série des BLOSUM (Henikoff et Henikoff 1992)

Pour les deux séries, les auteurs choisissent un **lot de séquences homologues** et pour chaque paire d'aminoacides, ils vont étudier le nombre de substitutions observées, en déduire une probabilité de cette mutation en la pondérant par la fréquence d'apparition de chacun des deux aminoacides.
La différence essentielle entre ces deux séries repose essentiellement sur le choix des lots de séquences et la façon de les aligner.

2.1. **PAM (Dayhoff)** (Percent Accepted Mutation)

Elles représentent les échanges possibles ou acceptables d'un acide aminé par un autre lors de l'évolution des protéines. Dayhoff a utilisé 1572 séquences protéiques groupées en 71 familles très semblables (moins de 15% de différence) avec un total de 1600 mutations. De ces alignements, une **matrice de probabilité** a été calculée où chaque élément de la matrice donne la probabilité qu'un acide aminé A soit **remplacé** par un acide aminé B **durant une étape d'évolution**. Ex : PAM250

	C Cys	S Ser	T Thr	P Pro	A Ala	G Gly	N Asn	D Asp	E Glu	Q Gln	H His	R Arg	K Lys	M Met	I Ile	L Leu	V Val	F Phe	Y Tyr	W Trp
C Cys	12																			
S Ser	0	2																		
T Thr	−2	1	3																	
P Pro	−3	1	0	6																
A Ala	−2	1	1	1	2															
G Gly	−3	1	0	−1	1	5														
N Asn	−4	1	0	−1	0	0	2													
D Asp	−5	0	0	−1	0	1	2	4												
E Glu	−5	0	0	−1	0	0	1	3	4											
Q Gln	−5	−1	−1	0	0	−1	1	2	2	4										
H His	−3	−1	−1	0	−1	−2	2	1	1	3	6									
R Arg	−4	0	−1	0	−2	−3	0	−1	−1	1	2	6								
K Lys	−5	0	0	−1	−1	−2	1	0	0	1	0	3	5							
M Met	−5	−2	−1	−2	−1	−3	−2	−3	−2	−1	−2	0	0	6						
I Ile	−2	−1	0	−2	−1	−3	−2	−2	−2	−2	−2	−2	−2	2	5					
L Leu	−6	−3	−2	−3	−2	−4	−3	−4	−3	−2	−2	−3	−3	4	2	6				
V Val	−2	−1	0	−1	0	−1	−2	−2	−2	−2	−2	−2	−2	2	4	2	4			
F Phe	−4	−3	−3	−5	−4	−5	−4	−6	−5	−5	−2	−4	−5	0	1	2	−1	9		
Y Tyr	0	−3	−3	−5	−3	−5	−2	−4	−4	−4	0	−4	−4	−2	−1	−1	−2	7	10	
W Trp	−8	−2	−5	−6	−6	−7	−4	−7	−7	−5	−3	2	−3	−4	−5	−2	−6	0	0	17
	C Cys	S Ser	T Thr	P Pro	A Ala	G Gly	N Asn	D Asp	E Glu	Q Gln	H His	R Arg	K Lys	M Met	I Ile	L Leu	V Val	F Phe	Y Tyr	W Trp

The PAM-250 Log Odds Substitution Matrix

2.2. **BLOSUM: (BLOcks SUbstitution Matrix)**

Henikoff et Henikoff (1992) ont étudié les substitutions d'aminoacides dans les blocs ("blocks") qui sont un alignement multiple d'une partie des séquences sans "gap" (insertion- délétion) dans les régions les plus conservées des familles de protéines. Ces auteurs ont utilisé plus d'une centaine de familles de protéines pour un total de blocs supérieurs à 2000. Ces blocs sont utilisés pour regrouper tous les segments de séquences ayant un **pourcentage d'identité** minimum au sein de leur **bloc**. On en déduit des **fréquences de substitution** pour chaque **paire d'acides aminés** et l'on calcule ensuite une matrice logarithmique de probabilité dénommée **BLOSUM**. Ex : **BLOSUM62**

4.3. Alignement multiple: Alignements de plus de deux séquences

Principe: alignement portant sur plusieurs séquences à la fois et dans leur intégralité. Il permet de mettre en évidence les relations entre séquences que l'on ne peut pas visualiser en comparant les séquences 2 à 2.

1. Identification de sites fonctionnels importants (conservés): L'alignement multiple de séquences entre espèces éloignées permet l'identification rapide des sites dont la conservation est requise pour la fonction.

2. Prédiction de fonction: Détecter des résidus identiques ou similaires ayant un rôle fonctionnel ou structural.

3. Prédiction de structure: La structure 3D des protéines étant plus conservée que la séquence primaire, on aura une prédiction de structure.

4. Recherche d'amorces de PCR: Pour dessiner des amorces (primers) de PCR pour amplifier le gène dans une nouvelle espèce.

6. Caractériser une nouvelle famille de protéines

7. Détecter une homologie entre différentes séquences

8. Etablir une phylogénie

Alignement multiple : ClustalW

Clustal : un programme très utilisé

Clustal est une des programmes d'alignements multiples les plus utilisés, relativement performant et présent sur de très nombreux serveurs.

L'ALGORITHME SUIVI PAR CLUSTAL

Les alignements multiples sont réalisés au moyen de trois étapes successives.

1. Calculer les distances entre toutes les paires de séquence
2. En utilisant cette matrice de distance, calculer un arbre phylogénétique avec la méthode de neighbor joining
3. Aligner les séquences en progressant dans l'arbre

· Aligner deux à deux les séquences voisines
· Répéter jusqu'à la fin :
· Faire les séquences consensus de ces alignements
· Collapser dans l'arbre les proches voisins
· Aligner les paires de séquences proches

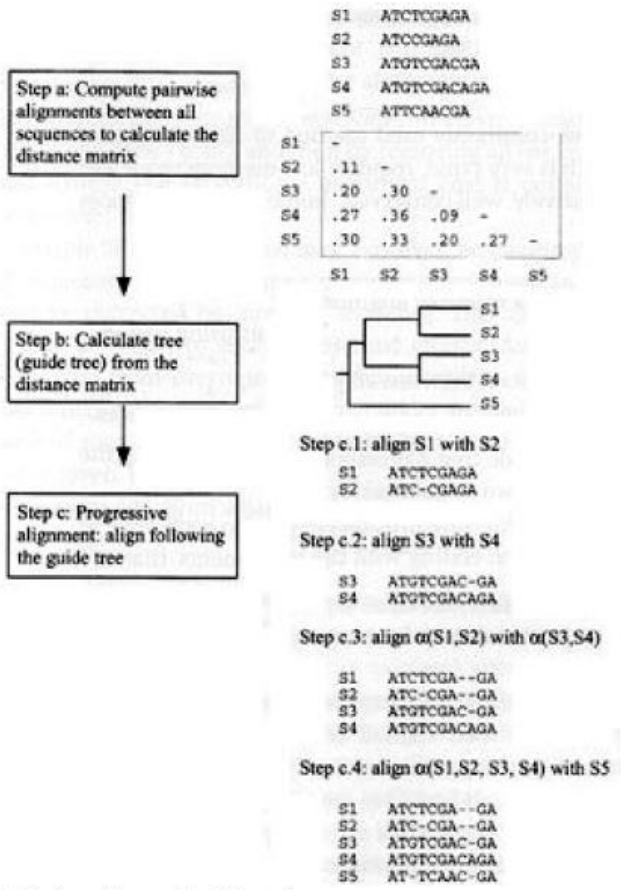


Schéma de l'algorithme de Clustal

4.4. Programmes de comparaison avec les banques

Recherches de similitudes dans les banques de séquences, Pourquoi ?

- Savoir si ma séquence ressemble à d'autres déjà connues
- Trouver toutes les séquences d'une même famille
- Rechercher toutes les séquences qui contiennent un motif donné

La taille sans cesse croissante des banques de séquences a nécessité **l'élaboration d'algorithmes spécifiques** pour effectuer **la comparaison d'une séquence avec une banque de données** car les algorithmes standards de comparaison entre deux séquences sont généralement trop longs sur des machines classiques.

La plupart de ces programmes constituent des méthodes heuristiques. Leur but est de filtrer les données de la banque en étapes successives car peu de séquences vont avoir des similitudes avec la séquence comparée. Ces méthodes heuristiques utilisent certaines approximations pour éliminer rapidement les situations sans intérêt et ainsi repérer les séquences de la banque susceptibles d'avoir une relation avec la séquence recherchée. Ces programmes permettent de calculer un **score** pour mettre en évidence les **meilleures similitudes locales** qu'ils ont observées.

Les deux types de programme les plus utilisés par les biologistes qui sont les logiciels
FASTA (Pearson et Lipman, **1988**) et
BLAST (Altschul et al., **1990**, **1997**).

1. FASTA

L'algorithme est basé sur **l'identification rapide des zones d'identité** entre la séquence recherchée et les séquences de la banque. Cette reconnaissance est essentielle car elle permet de considérer uniquement les **séquences présentant une région de forte similitude** avec la séquence recherchée. On peut ensuite, à partir de la meilleure zone de ressemblance, appliquer localement à ces séquences un algorithme d'alignement optimal. Le logiciel regroupe en fait plusieurs programmes de recherche avec les banques de données:

- Le programme FASTA qui compare respectivement une séquence nucléique avec une base nucléique ou une séquence protéique avec une base protéique.

- Les programmes TFASTA ou TFASTX qui comparent une séquence protéique avec des bases nucléiques traduites.
- Les programmes FASTX ou FASTY qui comparent une séquence nucléique traduite avec des bases protéiques.

2. BLAST: Basic Local Alignment Search Tool

BLAST est l'abréviation de « Basic Local Alignment Search Tool » ou, en français, L'outil de recherche basique d'alignement local. BLAST, quand à lui, cherche les bases de données des protéines et ADNs pour des séquences (sujets) qui ressemblent à notre séquence (requête) utilisée comme mot clé.
En ce qui concerne BLAST, il utilise l'alignement local pour comparer les séquences. Il divise la séquence en question « requête » en morceaux composées de trois acides aminés (en cas des protéines) ou 11 nucléotides (en cas d'ADNs). Ces morceaux sont nommés mots. En cherchant les bases de données de séquences avec ces mots on trouvera plusieurs mots (mots voisins) qui ressemble à ceux de la requête. Les mots voisins appartiennent à un ou plusieurs séquences sujets.

L'unité fondamentale de BLAST est le HSP (High-scoring Segment Pair) (fragments similaires). C'est un couple de fragments identifiés sur chacune des séquences comparées, de longueur égale mais non prédéfinie, et qui possède un score significatif. En d'autres termes, un HSP correspond à un **segment commun, le plus long possible**, entre deux séquences qui correspond à une similitude sans insertion-délétion ayant au moins un score supérieur ou égal à un score seuil.

La stratégie de la recherche consiste à trouver tous les HSPs entre la séquence recherchée et les séquences de la base.

- Pour déterminer un HSP, des mots de longueur fixe sont identifiés dans une **première étape** entre la séquence recherchée et la séquence de la banque.
- Dans une **deuxième étape**, on cherche à étendre la similitude dans les deux directions le long de chaque séquence, à partir du mot commun, de manière à ce que le score cumulé puisse être amélioré.
- Dans une **troisième étape**, la signification des segments similaires obtenus est évaluée statistiquement. Le score de la similarité est normalisé et évalué en unité standard d'information (bit). Ensuite la probabilité (E-value) d'avoir un tel score au hasard est calculé pour cette longueur de segment (m) dans une banque contenant au total (n) nucléotides ou acides aminés. Seuls seront conservés et classés les HSP significatifs, c'est à dire ceux dont la probabilité est la plus faible.

Il existe en fait deux versions de l'algorithme une sans insertion délétion, BLAST 1.0 (1990) et l'autre avec insertion délétion, BLAST 2.0 (1997).

Ce logiciel possède en fait plusieurs programmes de comparaison avec les bases de données :

- **BLASTN** (pour comparer une séquence nucléique contre base nucléique),
- **BLASTP** (Pour comparer une séquence protéique contre base protéique),
- **BLASTX** (comparaison de séquence nucléique (traduite en 6 phases) contre base protéique),
- **TBLASTN** (comparaison de séquence protéique contre base nucléique (traduite en 6 phases)),
- **TBLASTX** (comparaison de séquence nucléique (traduite dans les 6 phases) contre base nucléique (traduite dans les 6 phases)).

E-value

La signification statistique des alignements produits par un BLAST est mesurée par E-value (expected-value). Elle indique le nombre d'alignements différents ayant le même degré de similitude et que l'on peut espérer trouver par hasard dans la banque, même s'il n'existait pas de vraie séquence similaire.
Si $E=10^{-2}$ cela signifie que 1 alignement sur 100 sera trouvé par hasard.
E-value est donnée par la formule:

$$e = k \,.\, m \,.\, n \,.\, e^{-\lambda . S}$$

Elle dépend donc de la taille de la séquence (m), de la taille de la banque (n) et du score d'alignement (S).
K et λ sont des paramètres caractérisant la banque de données.