Université Batna 2 **Année universitaire : 2022-2023**

Faculté des Sciences de la nature et de la vie

Département d'Ecologie et environnement

Cours 2 : Alignement de séquences biologiques

Introduction :

Au cours de l'évolution naturelle, les mutations causent des erreurs au moment de la réplication de l'ADN car l'évolution se fait par mutations successives. Ces erreurs peuvent être :

- ✓ Des **substitutions** (changement ponctuel d'un nucléotide par un autre). On parle de transition ou de transversion,
- ✓ Des **insertions** (ajout d'un ou plusieurs nucléotides),
- ✓ Des **délétions** (suppression d'une base ou d'un segment d'ADN).

Il en découle alors des différences, plus ou moins importantes, dans les structures (primaire, secondaire, …) de ces séquences, d'où la divergence et la biodiversité des espèces.

En bioinformatique, la comparaison des séquences (ADN, ARN et/ou protéines…) repose essentiellement sur la notion de l'**alignement**[1], et permet de déterminer le degré de ressemblance entre celles-ci (similitude ou identité en révélant des régions proches dans leurs séquences primaires). Cela peut alors indiquer que :

- La structure (primaire, secondaire ou tertiaire) des deux séquences est semblable,
- La fonction biologique est proche ou différente (dans le cas de la dissémilarité),
- L'origine des séquences alignées est commune ou éloignée (notion d'homologie), …

Cependant, la comparaison pour l'obtention d'un alignement optimal entre deux séquences biologiques, nécessite néanmoins la mise en œuvre de procédures de calcul (algorithmes) et de modèles biologiques permettant de quantifier la notion de ressemblance entre ces séquences.

---

[1] **Alignement : opération de base en bioinformatique qui a pour but d'identifier des zones conservées entre séquences :**

- **Identifier des sites fonctionnels**
- **Prédire la ou les fonctions d'une protéine**
- **Prédire la structure secondaire (voir tertiaire ou quaternaire) d'une protéine**
- **Etablir une phylogénie (évolution : parenté entre les organismes).**

1. Définitions

Alignement : processus par lequel deux (ou n) séquences sont **comparées** afin d'obtenir **le plus de correspondances** (identités ou substitutions conservatives) possibles entre les lettres qui les composent.

- **Alignement local :** alignement des séquences sur une partie de leur longueur
- **Alignement global :** alignement des séquences sur toute leur longueur
- **Alignement optimal :** alignement des séquences qui produit le plus haut score possible
- **Alignement multiple :** alignement global de trois séquences ou plus

Brèches ou "gap" : espace artificiel introduit dans une séquence pour contre-balancer et matérialiser une insertion dans une autre séquence. Il permet d'optimiser l'alignement entre les séquences.

indel :

"in" = insertion

"del" = délétion

Similarité : c'est le pourcentage d'identités et/ou de substitutions conservatives entre des séquences. Le degré de similarité est quantifié par un score. Le résultat de la recherche d'une similarité peut être utilisé pour inférer l'homologie de séquences.

Homologie : 2 séquences sont homologues si elles ont un ancêtre commun.

mésappariement : non correspondance entre deux lettres. Un mésappariement peut être : soit la substitution d'un caractère par un autre, c'est-à-dire une mutation soi l'introduction d'un "gap"

Score : un score global permet de quantifier l'homologie. Il résulte de la somme des scores élémentaires calculés sur chacune des positions en vis à vis des deux séquences dans leur appariement optimal. C'est le nombre total de "bons appariements" pénalisé par le nombre de mésappariements.

2. TRAITEMENT DES SEQUENCES NUCLEIQUES (ADN ou ARN)

Notion de score : Le score élémentaire (noté "s") est une entité numérique que l'on attribue à chaque couple de nucléotides des deux séquences à comparer. Il prend la valeur de 1 lorsque les deux nucléotides des deux séquences sont identiques, et la valeur de zéro sinon.

Exemple :

Séquence1	A	G	C	T	A	C	C	T	G	T	Score global : Total des scores
Séquence2	A	A	G	T	A	G	C	T	T	T	
Point de comparaison	1	2	3	4	5	6	7	8	9	10	
Score élémentaire (s)	1	0	0	1	1	0	1	1	0	1	**1+0+0+1+1+0+1+1+0+1=6**

Dans cet exemple, constatez qu'au niveau du premier point de comparaison (ou site de comparaison), les deux séquences contiennent le même nucléotide A, donc le score élémentaire (s) à ce point prend la valeur de 1 (s = 1).

Au deuxième point de comparaison, la séquence 1 contient un G et la séquence 2 contient un A. Elles sont donc différentes en ce point d'où un score élémentaire de zéro (s = 0)…

Au 10ème point de comparaison, les deux séquences contiennent le même nucléotide T donc un score élémentaire de 1.

Constatons que la somme des scores élémentaires est égale à six (s = 6). Donc il y a six points identiques entre les deux séquences ; soit 60% d'identité entre les deux séquences ([6/10] x100). On dit alors que le score global entre les deux séquences est égal à six. Le score a donc permis de quantifier la ressemblance entre les deux séquences.

La relation entre le score global (S) et les scores élémentaires (s) pour deux séquences est de la forme :

$$S = \sum_{i=1}^{n} si$$

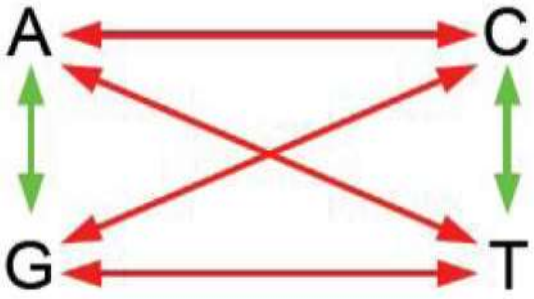
Il faut savoir qu'il existe une matrice (**matrice d'identité)** qui donne les valeurs de scores d'identité entre les séquences à comparer. Dans cette matrice, on attribue la valeur de 1 lorsque les deux nucléotides sont identiques et zéro s'ils ne le sont pas.

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

Matrice d'identité nucléique

Il existe une autre matrice de score, qui tient compte de l'analogie structurale entre purines (A et G) et pyrimidines (C, T et U) et affecte des scores en fonction de cette ressemblance :

C'est la matrice de transition/transversion : La substitution entre purines d'une part, et entre pyrimidines d'autre part est pondérée et n'a pas de score élémentaire nul au moment de la comparaison des séquences :



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

Matrice de transition/transversion

3. Recherche de segments identiques : La matrice de points (Dot plot)

Elle permet une vue (méthode visuelle) englobant les similarités entre les régions des séquences à comparer.
Exemple de réalisation: On donne deux séquences x et y :

x=ACTCGGATT et y=AGCTCGGT

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:

		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

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**

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

Remarque : Dans le cas où les deux séquences sont complètement identiques, le résultat est une diagonole principal.
Le dot-plot est utile pour déterminer de combien d'exons est composé un gène en le comparant à son ARNm et pour avoir une idée de la taille des introns et des exons.

Il existe un logiciel de dotplot interactif, Dotlet qui nécessite JAVA. Si JAVA n'est pas installé sur vos machines, vous pouvez utiliser Dottup.

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

4. Matrices protéiques

Notons tout d'abord que les matrices protéiques utilisées pour réaliser des alignements sont totalement différentes de celles des acides nucléiques (matrice d'identité et matrice de transition/transversion) et ce en raison du nombre des acides aminés (20 acides aminés et non 4 comme le cas des nucléotides) et de la nature physico-chimiques de ceux-ci.

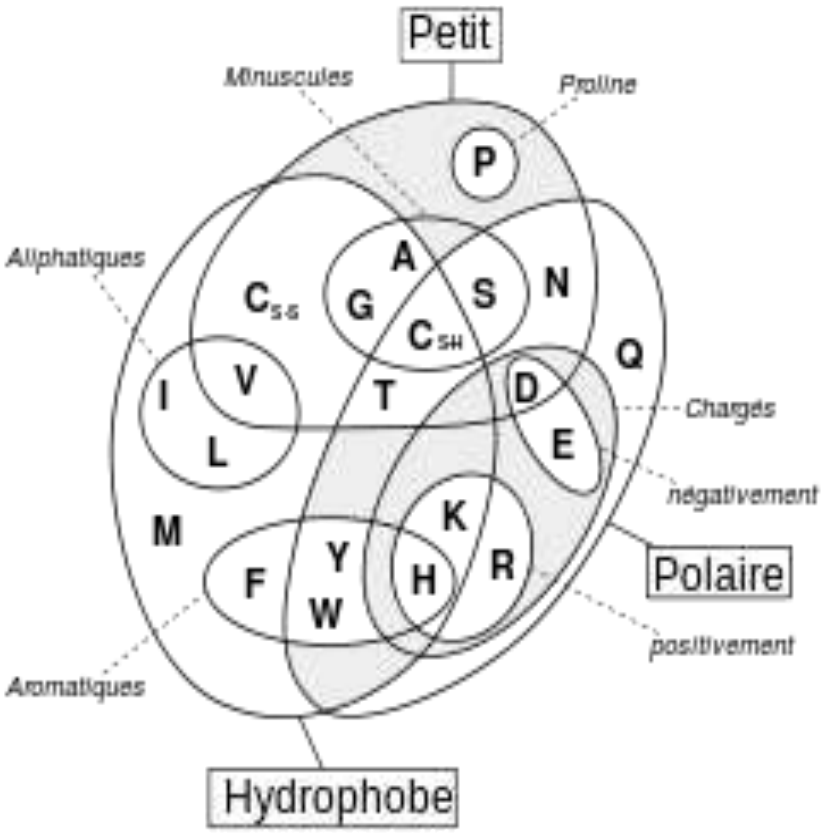
En effet, le système nucléique basé sur l'identité n'est pas approprié pour le cas des systèmes protéiques. Ceci est du au fait que certains acides aminés peuvent être remplacés par d'autres (à cause de leurs propriétés physicochimiques surtout) sans altérer le rôle et la fonction biologique de la protéine.

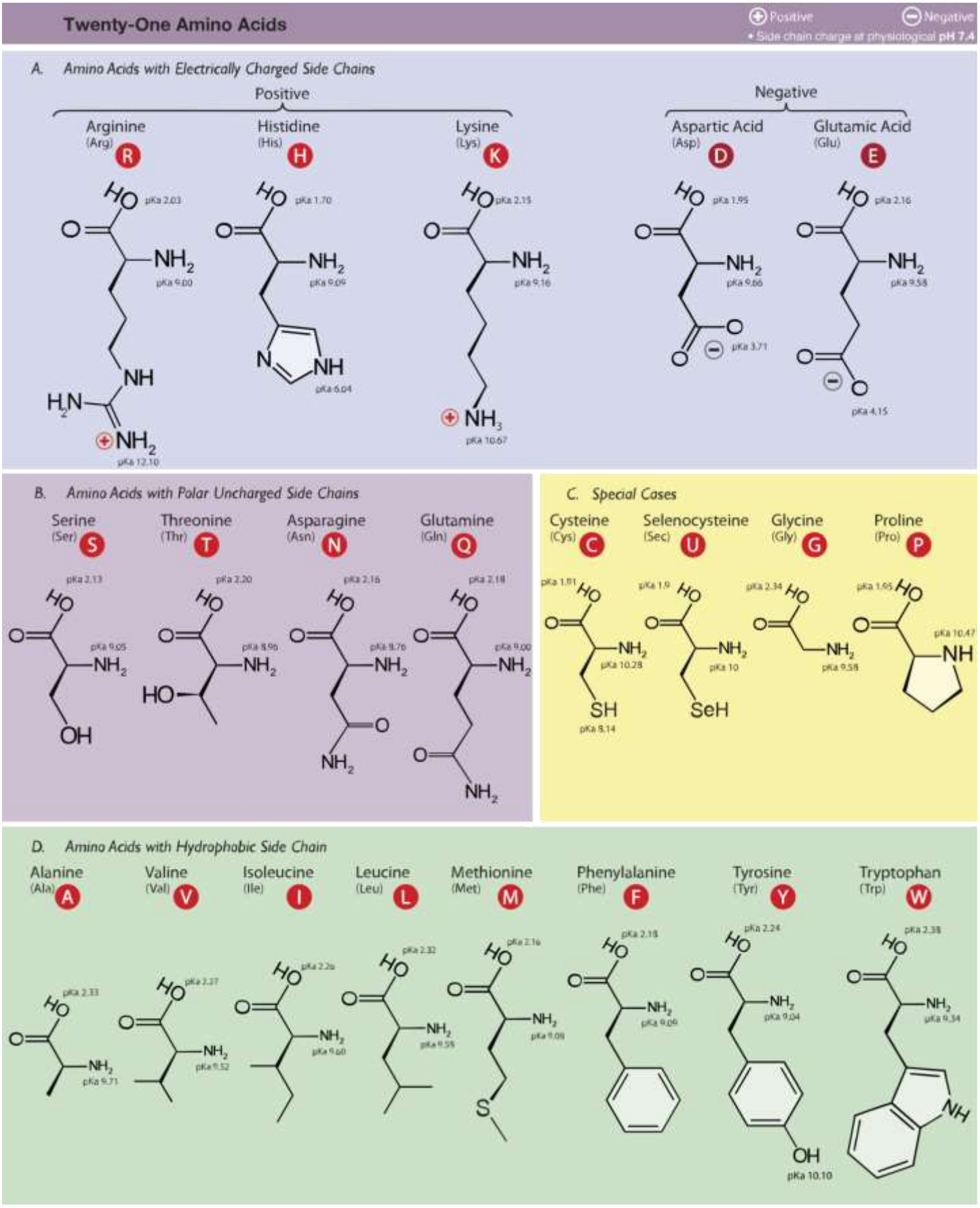
On peut donc classer les acides aminés en familles par rapport à leurs propriétés et obtenir ainsi un système de scores qui rende compte de l'affinité des résidus protéiques entre eux. C'est cette affinité qui permet à un acide aminé d'être substitué par un autre, et les deux structures protéiques ne seront pas identiques à ce point où la substitution a eu lieu mais on dira que les deux séquences sont **SIMILAIRES** et la fonction de la protéine reste conservée.

Dans l'exemple qui suit, on dispose de la structure primaire de deux séquences enzymatiques ayant la même fonction biologique, c'est-à-dire toutes les deux sont des amylases. Mais on se rend vite compte que l'amylase de la mouche ne ressemble pas à celle de l'abeille, et pourtant elles assurent toutes les deux l'hydrolyse de l'amidon.

Nom systématique	Numéro d'accession	Séquence : structure primaire
Mouche[8] *Musca domestica* 	AAY88830 319 aa	iareceeflaprgfagvqvspvtenvivanrpwweryqpisyklqtrsgtqqefsemcrrcnnvgiriy vdvllnhmaadqyqmavgtagsiadpaaksfpsvpyteldfhatceiwdwndryqvqncelvglk dldqsnewvrdclvefldhlvelgvagfrvdaakhmkasdleiiykrvrdlnvdhgfepnsrpfiyqev idhghetvskyeynllgavtefqfseeigrafrgnnqlkwlrnwgpqwgflpsdhalvfvdnhdnqr dggqvltyknskqykmatafalaypygitr imssfdftdrdqpppht ne
Abeille Apis mellifera[9] 	BAA86909 493 aa	mmpaivlllallltlaageiahndphfapghdaivhlfewkwndiakeceqflgpvgfggvqvspvqe nividkrpwweryqpisykwitrsgtreqfidmvarcnkagvriyvdvimnhmsgdrndahgtgns rantynfdypqvpytvknfhprcavnnyndpsnvrncelvglhdldqsqeyvrsklvdflndlvaigv agfrvdaakhmwpsdlrtiysrvrnlnrthgfpndaqpyifqevidygneaiskreyngigaviefkys yeisnafrgnnnlkwlvnwgeqwgflpskdslvfvdnhdtqrdnpqiltykyskrykmavafmlshp fgtprimssfdfqskdqgppndgngnilspsihdnicsngwicehrwrqiynmvrfrnlvkgtkidnw wdngsnqiafsrgcsgfvafngdqydlkknlkvclppgqycdvisgnlekgrctgkivtvgsdgnani eigageedgvlaihvkakma

Les acides aminés de même classe peuvent se substituer par simple mutation acceptable et répondre ainsi aux contraintes de la sélection évolutive. Il en découle alors des structures protéiques non identiques mais similaires




Twenty-One Amino Acids
Positive
Negative
• Side chain charge at physiological pH 7.4
A. Amino Acids with Electrically Charged Side Chains
Positive
Negative
Arginine (Arg) R
Histidine (His) H
Lysine (Lys) K
Aspartic Acid (Asp) D
Glutamic Acid (Glu) E
B. Amino Acids with Polar Uncharged Side Chains
Serine (Ser) S
Threonine (Thr) T
Asparagine (Asn) N
Glutamine (Gln) Q
C. Special Cases
Cysteine (Cys) C
Selenocysteine (Sec) U
Glycine (Gly) G
Proline (Pro) P
D. Amino Acids with Hydrophobic Side Chain
Alanine (Ala) A
Valine (Val) V
Isoleucine (Ile) I
Leucine (Leu) L
Methionine (Met) M
Phenylalanine (Phe) F
Tyrosine (Tyr) Y
Tryptophan (Trp) W

Les matrices protéiques peuvent être classées en deux catégories :

- Une catégorie qui regroupe les matrices issues d'études montrant le caractère de substitution des acides aminées au cours de l'évolution (matrices liées à l'évolution). Elles représentent les échanges possibles et acceptables d'un acide aminé par un autre lors de l'évolution des protéines.
- ❖ La deuxième est basée plus particulièrement sur les caractéristiques physicochimiques des acides aminés : caractère hydrophile ou hydrophobe des protéines, la structure secondaire ou tertiaire des protéines.

Ce sont les matrices liées à l'évolution qui seront utilisées pour réaliser les alignements des séquences protéiques.

4. 1. La matrice PAM250 (Purcent Accepted Mutation): La matrice de mutation de Dayhoff.

La plus courante, cette famille de matrices probabilistes a été calculée à partir d'une étude sur une famille de 71 protéines très semblables, que l'on pouvait facilement aligner. Chaque élément de la matrice représente alors la probabilité qu'un acide aminé se transforme en un autre dans un temps d'évolution donné. La matrice créée est une matrice 1PAM, on obtient une matrice XPAM en la multipliant par elle-même. Les probabilités associées sont alors les probabilités de mutation en un temps plus long. En prenant compte des fréquences relatives matrice PAM-X, utilisable directement dans les programmes. La matrice PAM-250 s'est avérée être optimale par rapport au problème biologique ce qui explique sa très grande fréquence d'utilisation10.

Les matrices de type PAM dérivent d'alignements globaux de protéines très semblables et représentent les échanges possibles et acceptables d'un acide aminé par un autre au cours de l'évolution des protéines : Les acides aminés entrant dans la composition d'une protéine peuvent avoir les mêmes propriétés physico-chimiques ou presque et la structure 3d va donc dépendre de ces caractéristiques. Cette similarité des propriétés physico-chimiques est donc suffisante pour permettre la substitution (la mutation) entre ces acides aminés sans pour autant perturber la fonction de la protéine.

4. 2. Les matrices BLOSUM ("BLOcks SUbstitution Matrix")

Elles sont postèrieures aux matrices PAM et ont été développées par Henikoff & Henikoff. Les matrices BLOSUM sont construites à partir de 2000 BLOCKS provenant de plus de 500 familles de protéines.

Le degré de substitution des acides aminés a été mesuré en observant des blocs d'acides aminés issus de protéines plus éloignées. Chaque bloc est obtenu par l'alignement multiple sans insertion/délétion de courtes régions très conservées. 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** (BLOcks SUbstitution Matrix). A chaque pourcentage d'identité correspond une matrice particulière. Ainsi la matrice BLOSUM50 est obtenue en utilisant un seuil d'identité de 50%.

Henikoff et Henikoff, (1992) ont réalisé un tel traitement à partir d'une base contenant plus de2000 blocs11 :

• observation de blocs d'acides aminés issus de protéines relativement éloignées ;

• chaque bloc provient d'alignements multiples sans insertions / délétions de courtes régions conservées ;

• les 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 ;
• on calcule une matrice logarithmique de probabilité ;
• à chaque pourcentage d'identité correspond une matrice :
• BLOSUM50 avec un seuil d'identité de 50 % ;
• BLOSUM62 avec un seuil d'identité de 62 %.

Matrice de PAM250

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

S tryptophane/cystéine = -8
S tyrosine/phénylalanine = 7
S tryptophane/tryptophane = 17

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
	Cys	Ser	Thr	Pro	Ala	Gly	Asn	Asp	Glu	Gln	His	Arg	Lys	Met	Ile	Leu	Val	Phe	Tyr	Trp
Hydrophobic	C			P	A	G								M	I	L	V			
Aromatic											H							F	Y	W
Polar		S	T				N			Q									Y	
Basic											H	R	K							
Acidic								D	E											

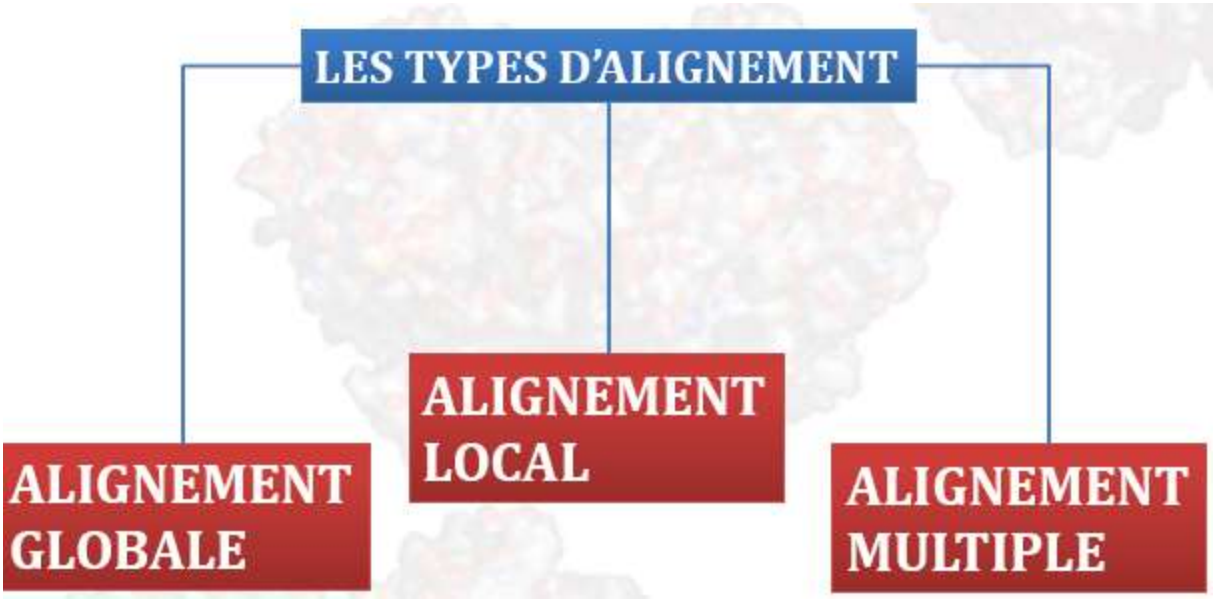
BLOSUM 62 scoring matrix

(positive values are shaded)

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	4																			
R	-1	5																		
N	-2	0	6																	
D	-2	-2	1	6																
C	0	-3	-3	-3	9															
Q	-1	1	0	0	-3	5														
E	-1	0	0	2	-4	2	5													
G	0	-2	0	-1	-3	-2	-2	6												
H	-2	0	1	-1	-3	0	0	-2	8											
I	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
L	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
K	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
M	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
F	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
P	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
S	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
T	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
W	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Y	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
V	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4

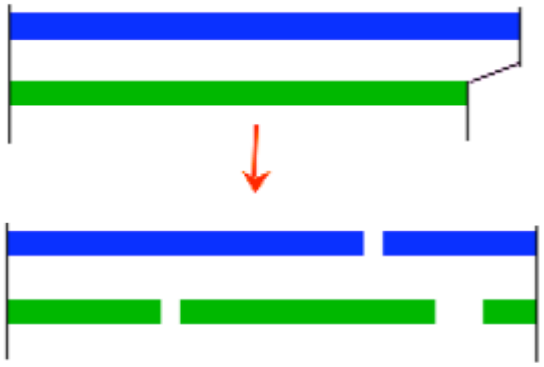
The values for amino acid substitutions were obtained from Henikoff S & Henikoff JG (1992) Amino acid substitutions matrices from protein blocks. *Proc. Natl. Acad. Sci.* **89**: 10915-10919.

5. Types d'alignement



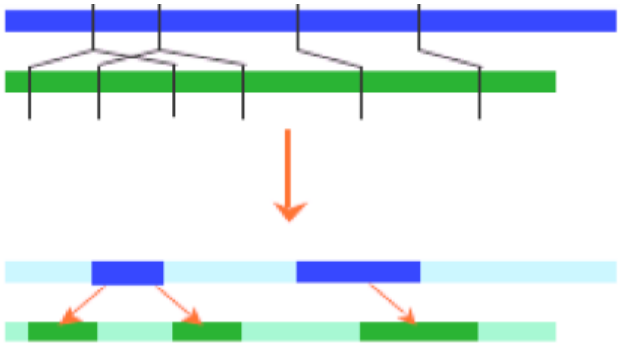
5. 1. Alignement global : Alignement de deux séquences sur la totalité de leur longueur en tenant compte de tous les résidus. Si les longueurs sont différentes, des insertions / délétions sont introduites pour aligner les deux extrémités des deux séquences.

Cet alignement permet de mesurer le degré de similitude entre deux séquences.

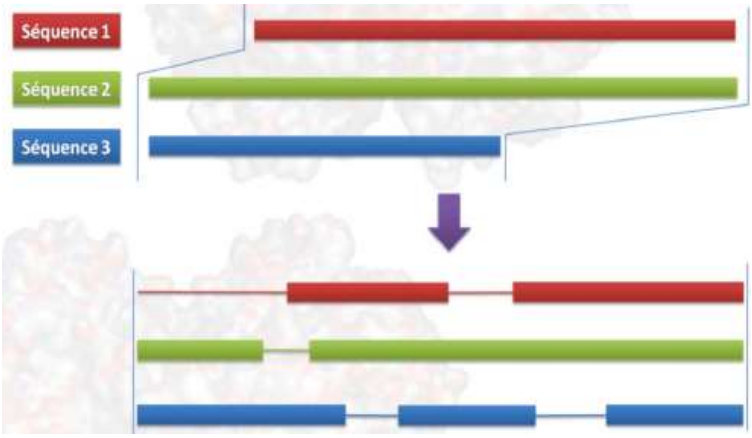


5. 2. Alignement local: Alignement de deux séquences sur des régions isolées et permettant de trouver des segments qui ont un haut degré de similitude.

Utilisé pour la recherche dans les bases de données (comparaison d'une séquence avec les séquences contenues dans la base).



5. 3. Alignement Multiple: 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 deux à deux.



À chaque type d'alignement est associé un programme informatique permettant d'optimiser le traitement

Alignement global :	Alignement local:	Alignement Multiple:
Needle Stretcher	BLAST (Basic Local Alignment Tool) FASTA	T-Coffee

Exemples sur les logiciels :

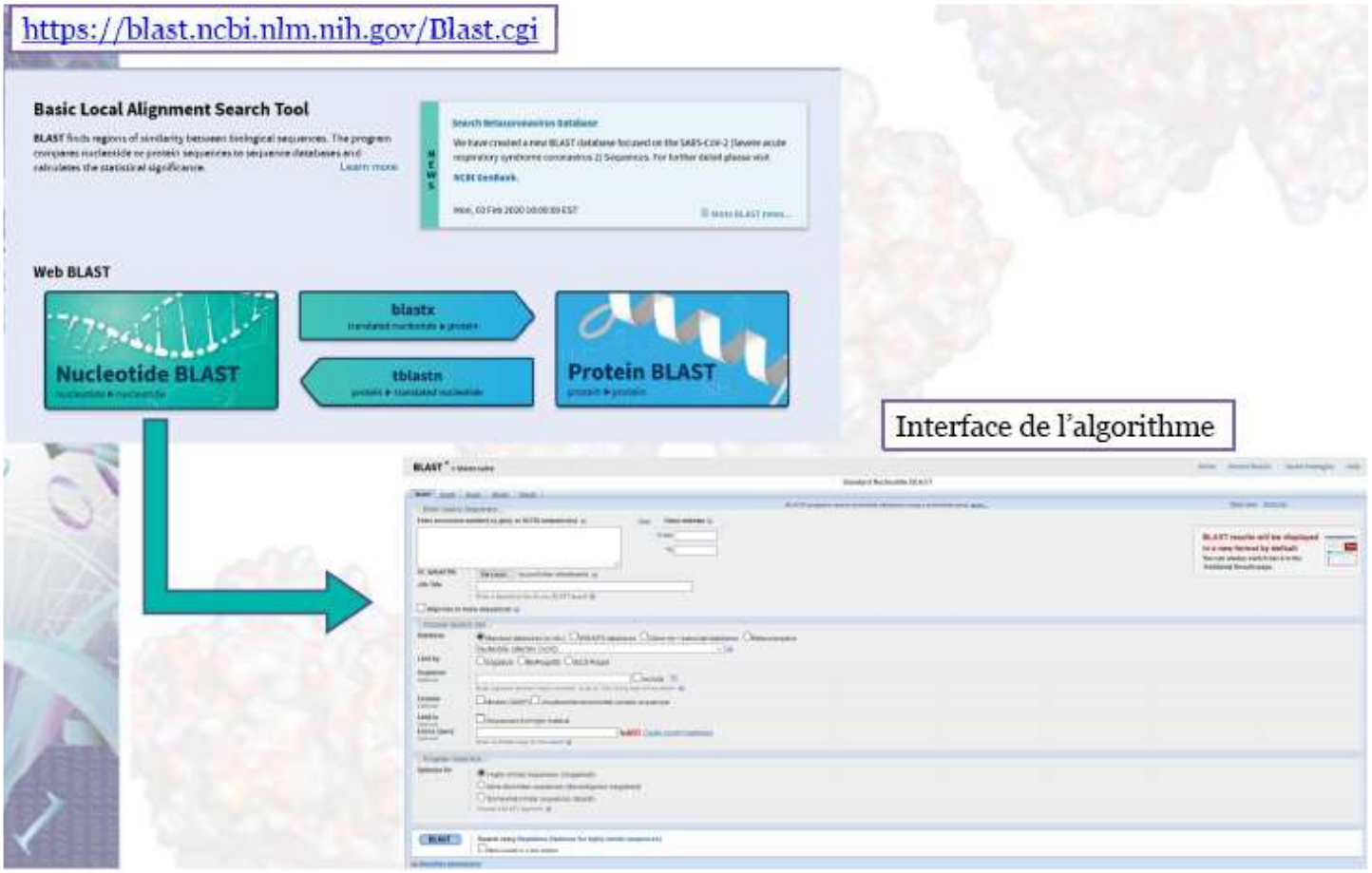
- **Alignement local :**

BLAST (Basic Local Alignment Search Tool) :

. Programme pour la recherche de similarités dans les bases de données
. Utilise un algorithme heuristique linéaire pour l'alignement local
. Séquences nucléiques et protéiques
. Disponible sur le Web
. Connecte aux principales banques de données

FASTA (FAST All):

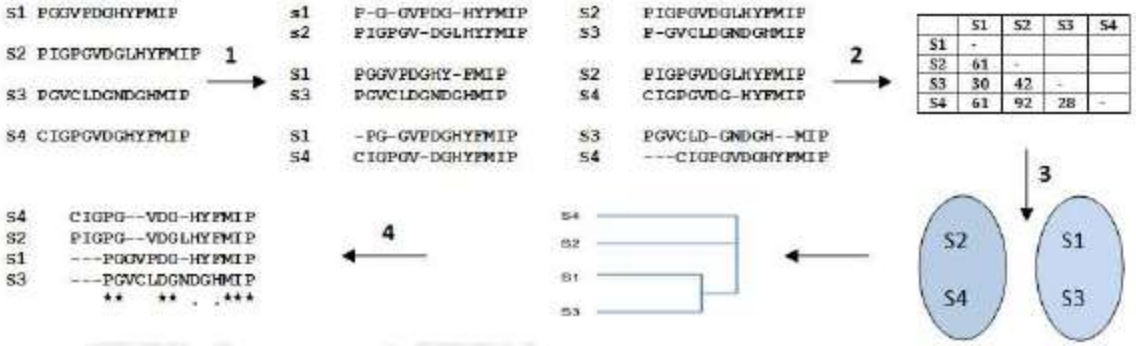
L'algorithme est basé sur l'identification rapide des zones d'identité entre la séquence recherchée et les séquences de la banque de données. Cette reconnaissance est essentielle car elle permet de considérer uniquement les **séquences présentant une région de forte similitude** avec la séquence recherchée.



- **Alignement multiple :**

CLUSTALW (Cluster Alignment). (Thompson, Higgins et Gibson, 1995)

Principe : **CLUSTALW** est fondé sur l'utilisation d'un algorithme d'alignement progressif. Les séquences les plus similaires sont alignées en premier puis l'alignement progresse vers les séquences les plus distantes. C'est également un programme de construction d'arbre phylogénétique.



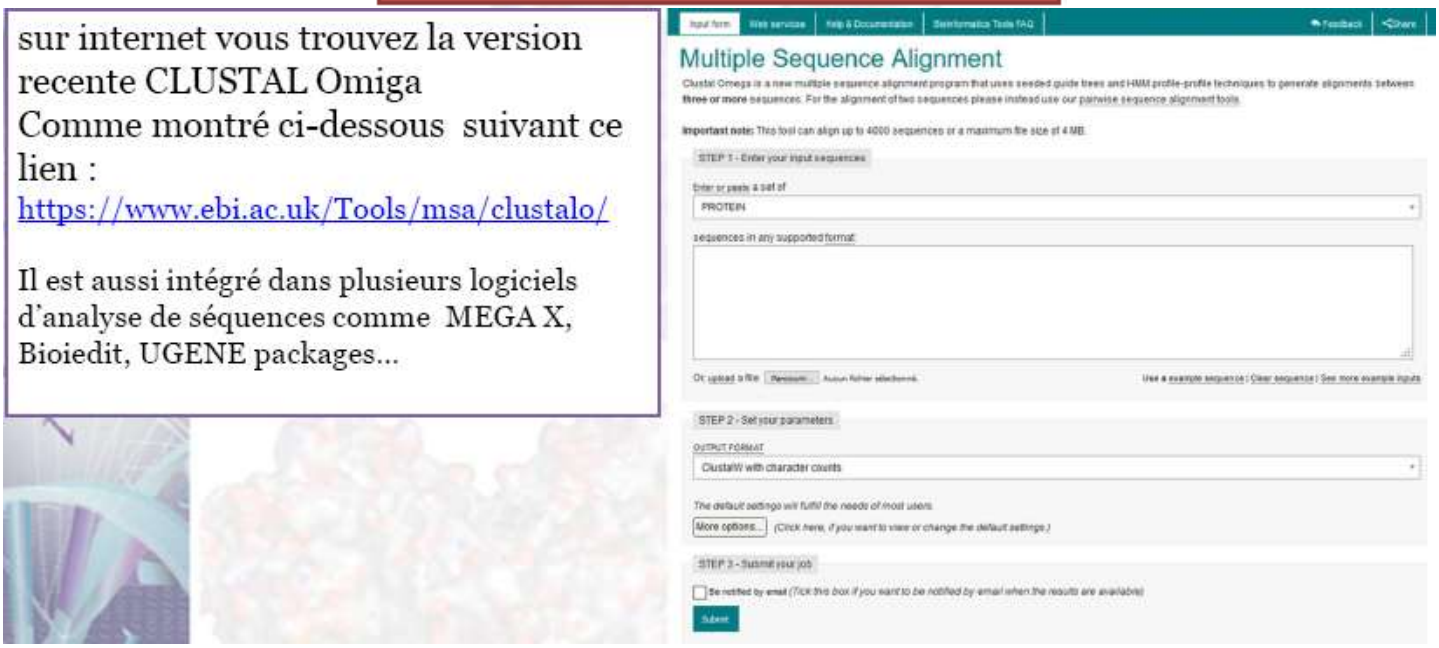
	S1	S2	S3	S4
S1	-			
S2	61	-		
S3	30	42	-	
S4	61	92	28	-

Étape schématique de l'alignement multiple avec CLUSTALW avec 4 séquences d'acides aminés :
1-Alignement de toutes les séquences 2 à 2 et détermination des scores des alignements
2-Construction d'une matrice de score (BLOSUM62) pour l'ensemble des séquences
3-construction d'un arbre guide à partir de la matrice traduisant les relations globales entre les séquences
4-Alignement progressif à partir de l'alignement des 2 séquences les plus proches. les séquences voisines sont alignées de proche en proche jusqu'à l'alignement multiple final.
Légende : « * » : résidus conservés « . » substitution conservatives

CLUSTAL sur internet

sur internet vous trouvez la version recente CLUSTAL Omiga
Comme montré ci-dessous suivant ce lien :
https://www.ebi.ac.uk/Tools/msa/clustalo/

Il est aussi intégré dans plusieurs logiciels d'analyse de séquences comme MEGA X, Bioiedit, UGENE packages...



T-COFFEE (Noterdame, Higgins, Hering, 2000)

T-COFFEE est fondé également sur un alignement progressif. En plus de réaliser un alignement global entre chacune des paires de séquences, il procède à un alignement local afin d'optimiser l'alignement entre les séquences très divergentes.

6. Algorithmes dédies à l'alignement

Il existe 3 grandes classes d'algorithmes de comparaison de séquences :

- **Méthode de programmation dynamique**
- **Méthode heuristique**
- **Méthode d'apprentissage machine**

6. 1. Algorithme de Needleman & Wunsch et algorithme de Smith & Waterman

Tous deux sont des algorithmes de programmation dynamique utilisés pour obtenir l'alignement global ou local (respectivement) optimal de deux séquences protéiques ou d'acides nucléiques.

La programmation dynamique est une méthode développée par R. Bellman (1955) qui permet de résoudre de nombreux problèmes dont la solution directe n'est pas possible puisque de complexité exponentielle.

Exemple : calcul de la distance d'édition entre deux chaînes de caractères (séquences protéiques ou d'acides nucléiques).

La programmation dynamique une méthode de résolution ascendante qui détermine une solution optimale du problème à partir des solutions de tous les sous-problèmes.

L'algorithme de Needleman & Wunsch et l'algorithme de Smith & Waterman se déroulent globalement en deux étapes :

- La construction, ou descente, qui permet de calculer le meilleur score, c'est à dire le coût de la transformation de la première séquence en la seconde (étape de programmation dynamique)
- La construction de l'alignement lui-même, ou remontée

Ces algorithmes n'utilisent pas d'heuristique : ils sont donc sensibles mais longs.

Algorithme de Needleman & Wunsch alignement global optimal de 2 séquences	Algorithme de Smith & Waterman alignement local optimal de 2 séquences
$F(i,j)=\max\begin{cases}F(i-1,j-1)+s(x_i,y_j),\\F(i-1,j)+d,\\F(i,j-1)+d.\end{cases}$	$F(i,j)=\max\begin{cases}0,\\F(i-1,j-1)+s(x_i,y_j),\\F(i-1,j)+d,\\F(i,j-1)+d.\end{cases}$
La ligne i = 0 et la colonne j = 0 sont initialisées aux valeurs de pénalité des gaps. La fonction de récurrence ne réinitialise pas la valeur à 0 si aucune valeur positive n'est présente.	La ligne i = 0 et la colonne j = 0 sont initialisées à 0. N'importe quelle case de la matrice de comparaison peut être un point de départ pour le cacul des scores finaux. Si ce score devient inférieur à zéro, la fonction de récurrence réinitialise la valeur à 0 et la case peut être utilisée comme un nouveau point de départ.

F(i,j) : valeur à la position (i,j) de la matrice.
s(xi,yj) : valeur obtenue à partir de la matrice de substitution pour les nucléotides ou les acides aminés (xi,yj) correspondant à la position (i,j) de la matrice. C'est donc le score correspondant à l'alignement des lettres xi et yj.
Ce score prend, par exemple, les valeurs suivantes :
identité : +3
non identité : -1
s(xi,-) et s(-,yj) est la fonction simple de pénalité de l'alignement d'un résidu avec un gap : -5
Remarque : si on opte pour d'autres valeurs, on obtient d'autres alignements optimaux, d'où le choix crucial de la meilleure matrice de substitution lors des alignements.

Application : alignement de la séquence 1 = ACGCT avec la séquence 2 = ACT
On remplit la 1ère ligne et la 1ère colonne de la matrice qui correspondent à un gap à la 1ère position :
l'alignement du A de la séquence 2 avec l'insertion d'un gap dans la séquence 1 coûte : -5
celui du C de la séquence 2 avec l'insertion d'un second gap de la séquence 1 coûte : -5 + -5 = -10
et ainsi de suite ...

F(1,1) aura pour valeur la valeur maximale de l'une des possibilités suivantes :

- F(0,0) + s(A,A) = 0 + 3 = 3
- F(0,1) + s(A,-) = -5 + -5 = -10
- F(1,0) + s(-,A) = -5 + -5 = -10

F(2,1) aura pour valeur la valeur maximale de l'une des possibilités suivantes :

- F(1,0) + s(C,A) = -5 + -1 = -6
- F(1,1) + s(C,-) = 3 + -5 = -2
- F(2,0) + s(-,A) = -10 + -5 = -15

	j	0	1	2	3
i		- (gap)	A	C	T
0	- (gap)	0	-5	-10	-15
1	A	-5	3	-2	-7
2	C	-10	-2	6	1
3	G	-15	-7	1	5
4	C	-20	-12	-4	0
5	T	-25	-17	-9	-1

Et ainsi de suite.

Pour reconstituer l'alignement, on démarre de la dernière case (5,3) et on détermine la case à partir de laquelle cette case a été atteinte :

a. la valeur -1 de la case (5,3) ne peut-être obtenue qu'en ajoutant +3 (soit une identité) à la valeur -4 [(case (4,2)]. Celà correspond à l'alignement du "T" de la séquence 1 avec le "T" de la séquence 2.

Seq1	A	C	G	C	T
Seq2	A	-	-	C	T

b. la valeur -4 de la case (4,2) peut être obtenue de 2 manières :

- en ajoutant +3 (soit une identité) à la valeur -7 [(case (3,1)]. Celà correspond à l'alignement du "C" de la séquence 1 avec le "C" de la séquence 2.
- en ajoutant -5 (soit un gap) à la valeur 1 [(case (3,2)]. Celà correspond à l'alignement du "C" de la séquence 1 avec un gap dans la séquence 2.

Seq1	A	C	G	C	T
Seq2	A	C	-	-	T

c. Et ainsi de suite.

Dès lors, on obtient 2 alignements optimaux qui ont le même score de +1.

Annexe

Alignement

- Mise en correspondance de deux séquences (ADN ou protéines) pour faire apparaître les similarités, i.e., segments communs

AAAATTTTTTGGCCTTTAA et AAAAGCCCAA

AAAATTTTTTGGCCTTTAA
AAAAGCCCAA

gaps

AAAATTTTTTGGCCTTTAA
AAAA GCCC AA

Score d'un alignement

- Il y a un nombre exponentiel d'alignements possibles
- On doit ordonner les qualité avec un Score
  - On somme 3 événements élémentaires le long de l'alignement
    - Même lettre : match C / C
    - Lettre différente : mismatch C / G
    - Insertion/Deletion (indel) C / - ou - / C

- Exemple sur **ACGGCTAT ACTGTAT** avec le score :
  - Correspondance/Match : +2
  - Substitution/Mismatch : -1
  - Indel : -2

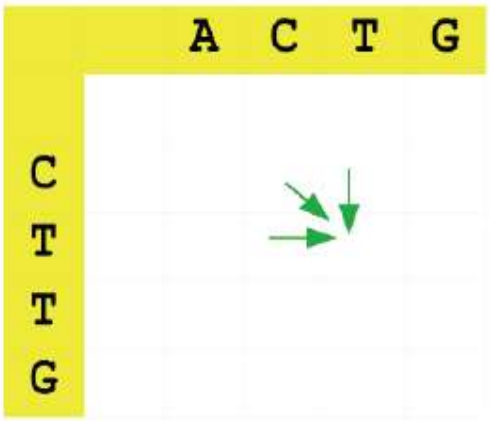
```
ACGGCTAT
||  |
ACTGTAT-
```

Score = 2+2-1+2-1-1-1-2 = 0

```
ACGGCTAT
||  | |||
ACTG-TAT
```

Score = 2+2-1+2-2+2+2+2 = 9

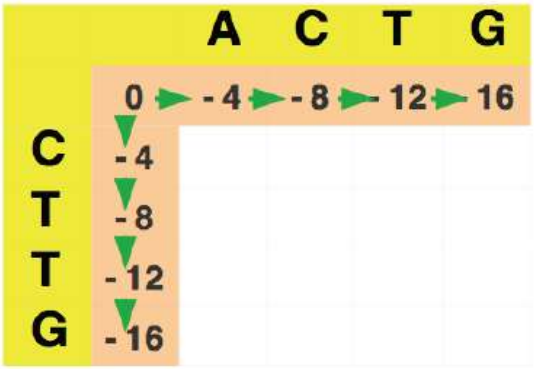
Alignement Global: Programmation dynamique



- **Règle 1:**
  chaque case va contenir un score; le score de l'alignement sera celui de la case en bas à droite
- **Règle 2:**
  le score d'une case se déduit à partir de celui des cases au-dessus, à gauche ou en diagonale
- **Règle 3:**
  un pas horizontal/vertical coûte 1 gap
  un pas diagonal coûte 1 position alignée (match ou mismatch)

Etape 1 :

Needleman & Wunsch - 1970



Gap penality -4

Etape 2:

on remplit toutes les cases <u>en gardant en mémoire le mouvement qui donne le meilleur score</u>

Score:
gap: -4 mismatch: -4

		A	C	T	G
	0	-4	-8	-12	-16
C	-4				
T	-8				
T	-12				
G	-16				

alignement AC → score = 0-4 = -4

insertion de gap → score = -4-4 = -8

insertion de gap → score = -4-4 = -8

on remplit toutes les cases <u>en gardant en mémoire le mouvement qui donne le meilleur score</u>

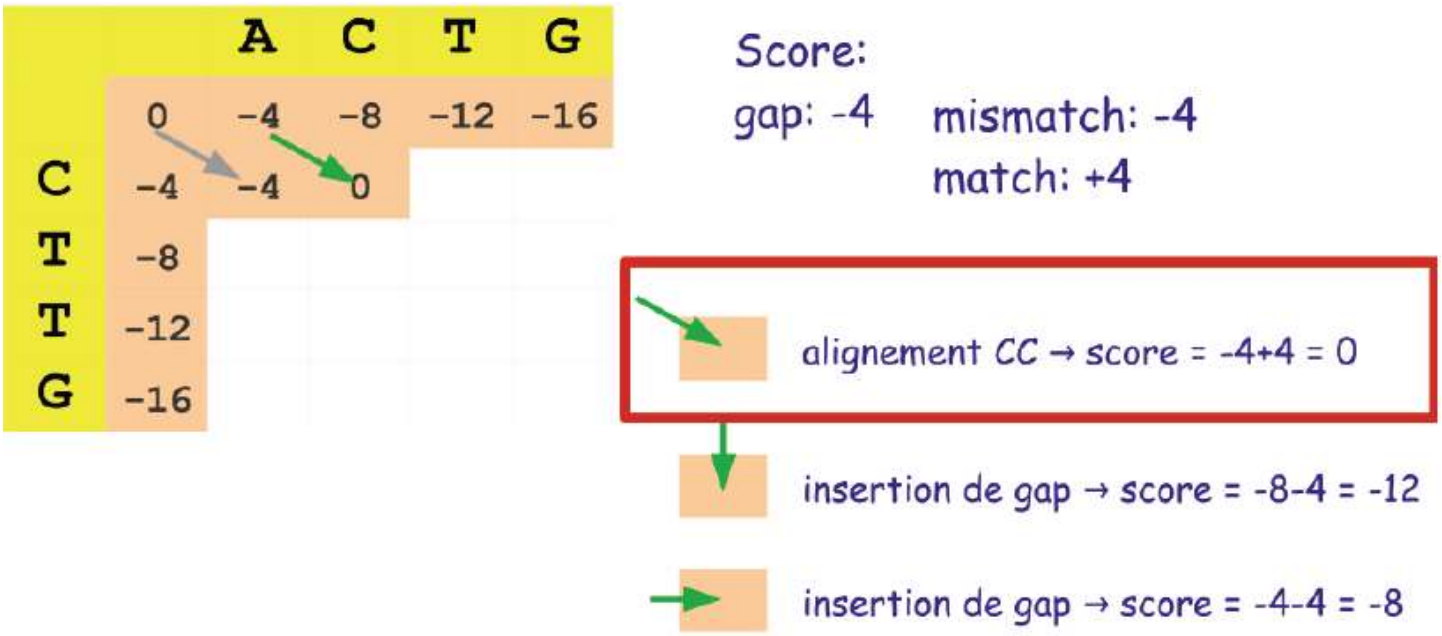
Score:
gap: -4 mismatch: -4

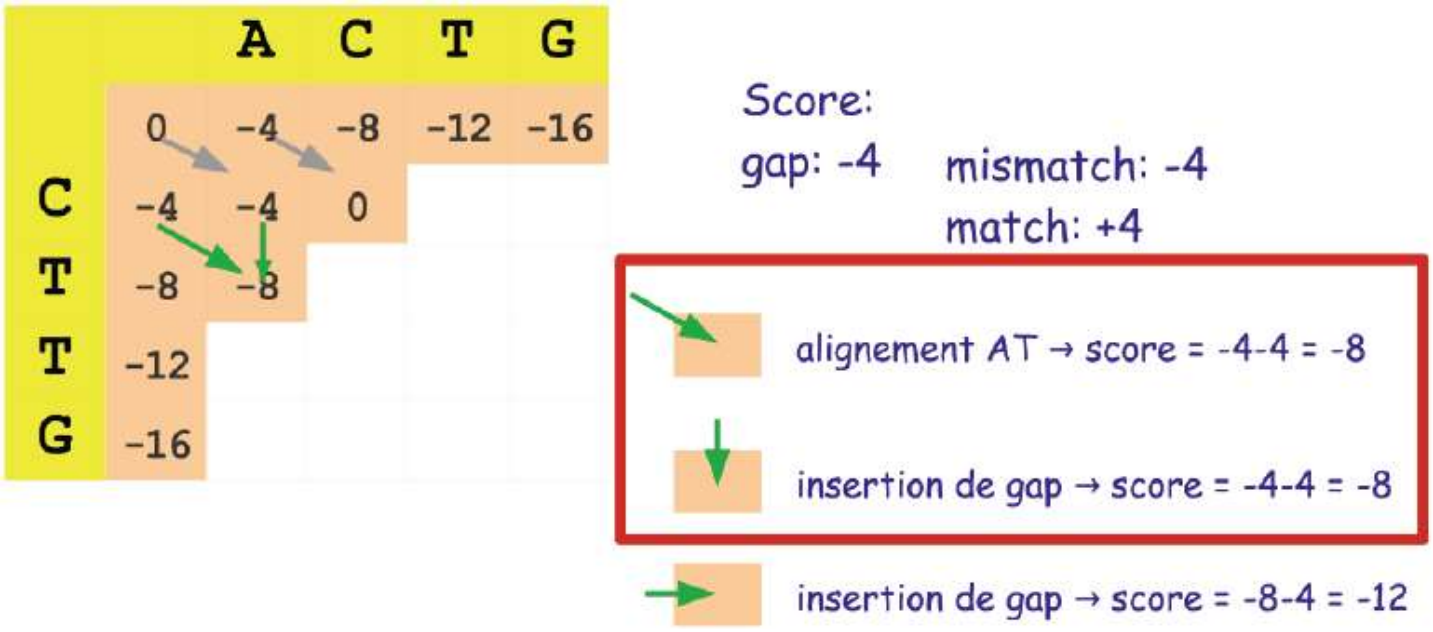
		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4			
T	-8				
T	-12				
G	-16				

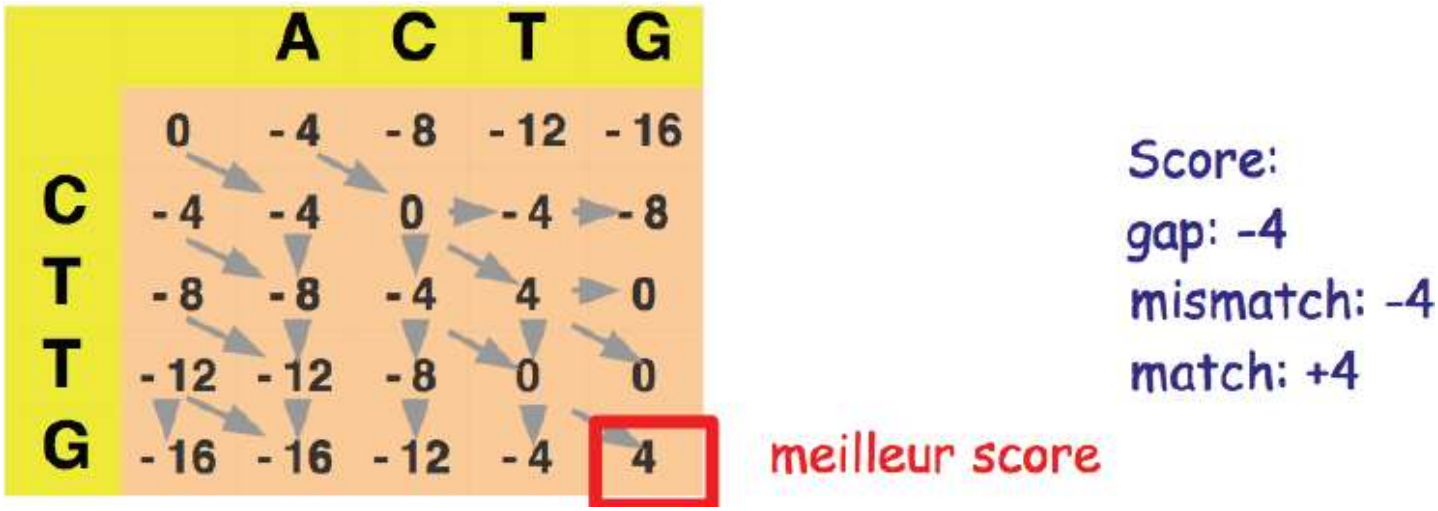
alignement AC → score = 0-4 = -4

insertion de gap → score = -4-4 = -8

insertion de gap → score = -4-4 = -8


A C T G
0 -4 -8 -12 -16
C -4 -4 0
T -8
T -12
G -16
Score:
gap: -4 mismatch: -4
match: +4
alignement CC → score = -4+4 = 0
insertion de gap → score = -8-4 = -12
insertion de gap → score = -4-4 = -8


A C T G
0 -4 -8 -12 -16
C -4 -4 0
T -8 -8
T -12
G -16
Score:
gap: -4 mismatch: -4
match: +4
alignement AT → score = -4-4 = -8
insertion de gap → score = -4-4 = -8
insertion de gap → score = -8-4 = -12


A C T G
0 -4 -8 -12 -16
C -4 -4 0 -4 -8
T -8 -8 -4 4 0
T -12 -12 -8 0 0
G -16 -16 -12 -4 4
meilleur score
Score:
gap: -4
mismatch: -4
match: +4

Etape 3:

On part du score en bas à droite, et on remonte le cours des flèches pour trouver l'alignement (« **backtracking** »)

		A	C	T	G
	0	-4	-8	-12	-16
C	-4	-4	0	-4	-8
T	-8	-8	-4	4	0
T	-12	-12	-8	0	0
G	-16	-16	-12	-4	4

2 chemins =
2 alignements **optimaux**:

AC-TG
-CTTG

ACT-G
-CTTG

score: +4

Bilan:
- 24 scores calculés
- 3^{4+4} = 6561 chemins possibles

Alignement **global**: on aligne les 2 séquences du début à la fin

- 41

- 2 séquences A = $(a_1,...,a_n)$ et B=$(b_1,...,b_m)$
- $S_{i,j}$ = score maximum entre 2 séquences alignées du début jusqu'aux résidus a_i et b_j.
- Initialisation : $S_{i,0} = i * g$

  $S_{0,j} = j * g$
- Récurrence : $$S_{i,j} = \max \begin{cases} S_{i-1,j-1} & +\sigma(a_i, b_j) \\ S_{i-1,j} & +g \\ S_{i,j-1} & +g \end{cases}$$

$$\sigma(a_i, b_j) = \begin{cases} \text{score de match si } a_i = b_j \\ \text{score de mismatch sinon} \end{cases}$$

g = score de gap