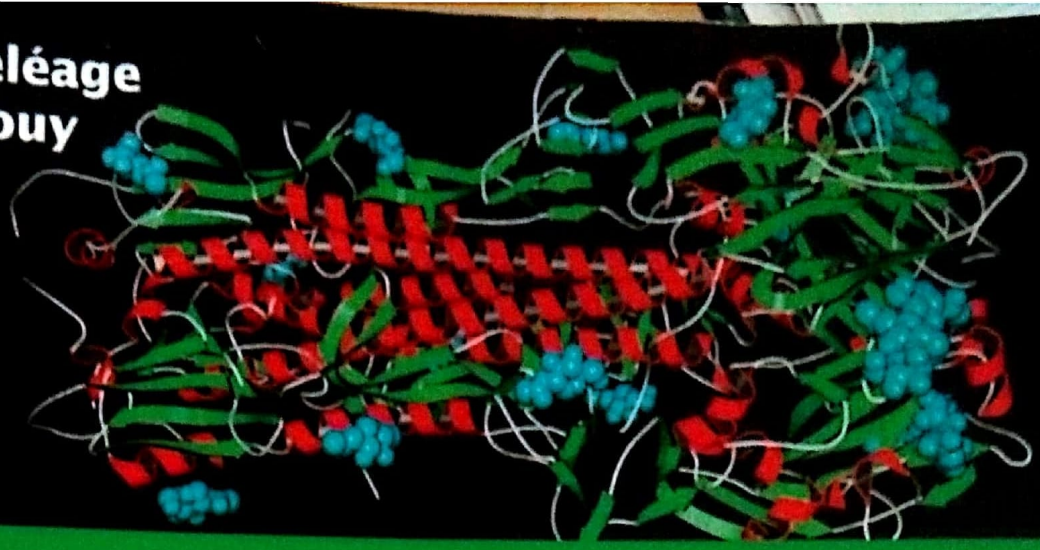
Gilbert Deléage
Manolo Gouy



Bioinformatique

Cours et cas pratique

Avec ce livre, des bonus sur le web

- Licence 3
- Master
- Écoles d'ingénieurs

DUNOD

TABLE DES MATIÈRES

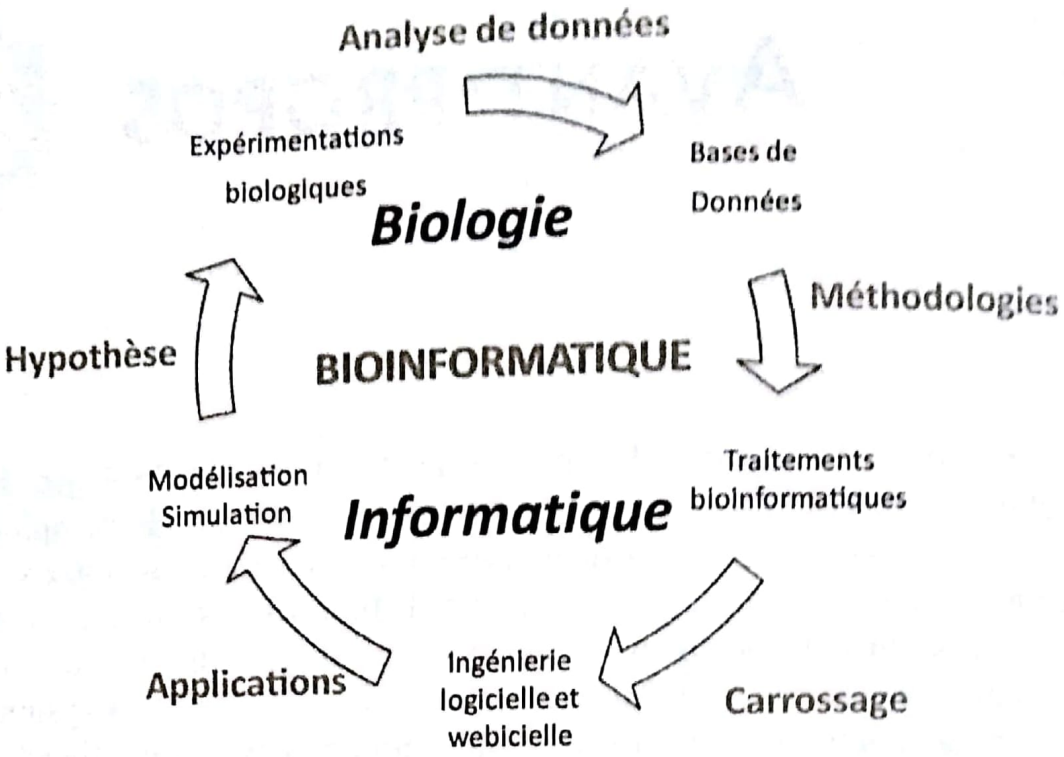
AVANT-PROPOS

La bioinformatique est une « interdiscipline » à la frontière de la biologie, de l'informatique et des mathématiques. Les systèmes biologiques sont très complexes et les techniques modernes d'investigation du monde biologique fournissent une vaste quantité de **données** expérimentales. Le but ultime de la bioinformatique est d'intégrer ces données d'origines très diverses pour modéliser les systèmes vivants afin de comprendre et prédire leurs comportements (biologie systémique ou biologie des systèmes) dans des conditions de fonctionnement normales ou pathologiques. Ainsi, à titre d'exemple, le séquençage à très haut débit offre la possibilité de connaître de manière personnalisée le **génome** de chacun. Pour tirer le bénéfice de cette connaissance, il faut développer et appliquer de nouvelles méthodes d'analyse bioinformatique qui permettent d'extraire l'information utile cachée dans la séquence du génome et, de manière plus générale, des données biologiques à grande échelle issues des progrès de l'expérimentation et des technologies de l'automatique. La bioinformatique est donc étroitement couplée à ses applications. Bon nombre de bioinformaticiens ne travaillent pas dans des laboratoires formellement estampillés « bioinformatique ». La bioinformatique et la modélisation procèdent selon un cercle vertueux (schématisé page suivante) dans lequel le point de départ est l'expérimentation biologique (un séquençage par exemple), les données produites sont ensuite organisées dans des dépôts de données (banques ou bases de données). Les méthodes d'analyse qui utilisent ces données sont développées par les bioinformaticiens souvent en association avec des informaticiens et mathématiciens. Pour que ces méthodes permettent le traitement ultérieur des données, il est nécessaire de « carrosser » ces méthodes (sous forme de logiciels ou serveurs Web) afin de permettre au biologiste de les utiliser pour émettre de nouvelles hypothèses qui seront testées et qui généreront de nouvelles données.

Aujourd'hui tout projet de biologie comporte une étape d'analyse bioinformatique des données. Par conséquent, un biologiste passe environ 20-30 % de son temps à utiliser des outils bioinformatiques.

Ce livre décrit de manière simple les tâches courantes de la bioinformatique qu'un biologiste/biochimiste doit savoir traiter par lui-même sans avoir recours au spécialiste afin de répondre à des questions usuelles comme :

- Comment extraire des informations pertinentes dans les banques de données biologiques ?
- Est-ce qu'une nouvelle séquence a déjà été complètement ou partiellement répertoriée ?
- Est-ce que ce gène appartient à une famille connue ?

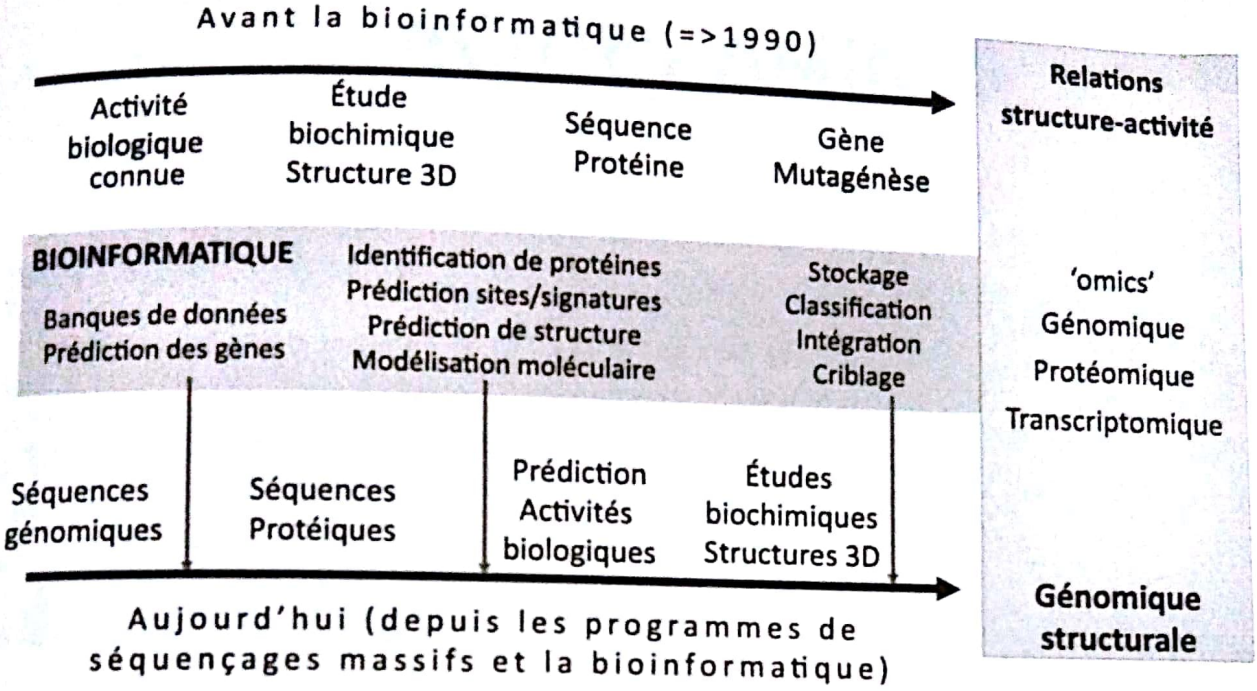


- Existe-t-il d'autres gènes homologues ?
- Est-ce que deux séquences correspondent à deux gènes homologues ?
- Existe-t-il des résidus essentiels à la fonction ?
- Alignement multiple, quel outil ? Pour quoi faire ? Établissement de consensus.
- Quelle peut être la fonction d'une protéine (prédit d'après sa séquence, sa structure...) ?
- Recherche de sous-motifs communs à un ensemble de séquences.
- Recherche de régions contenant des séquences répétées.
- Recherche d'hélices ou de brins dans les protéines.
- Comment construire un modèle tridimensionnel de protéine ?
- Optimisation et comparaison de structures 3D.
- Quelle est la charge globale d'une protéine à un pH donné ?

Ce livre n'a pas la prétention d'être exhaustif (il se limite d'une manière générale aux protéines, mais les **algorithmes** sont souvent très proches de ceux développés pour les acides nucléiques). Il a été rédigé afin de faciliter la compréhension des approches, méthodes, algorithmes et **implémentations** les plus courantes en bioinformatique moléculaire et structurale. À ce titre, il est parfois simplificateur et doit être considéré comme une introduction à la bioinformatique moléculaire et structurale. Il s'adresse donc aux étudiants de biologie/biochimie, de niveau licence, master ou classes préparatoires, ou bien aux biologistes qui souhaitent s'initier et comprendre les méthodes sous-jacentes aux programmes afin d'estimer la qualité de leurs analyses.

La logique suivie dans le livre est de partir des séquences de protéines pour aller vers leurs structures secondaires, leurs structures tridimensionnelles et finir par leurs fonctions. Elle suit la stratégie actuelle d'analyse d'une question biologique qui a été revisitée du fait de l'avènement de la bioinformatique et des séquençages massifs.

La bioinformatique moléculaire a pour première mission de « faire parler cette séquence » pour en tirer le maximum d'informations selon le schéma suivant :



Un exercice de mise en pratique de l'analyse de séquence est fourni avec son corrigé (chapitre 13).

La plupart des images des structures 3D présentées ont été générées à l'aide du logiciel AnTheProt pour Windows (http://antheprot-pbil.ibcp.fr).

Les vidéos fournies dans le complément numérique (www.dunod.com) ont été capturées à l'aide du logiciel CAMSTUDIO (http://camstudio.org/). Un quiz en ligne est disponible à l'adresse suivante : https://publi.ibcp.fr/scripts/bio_info.php.

Les auteurs remercient Christophe Combet et Céline Brochier pour leur relecture.

LA COMPOSITION EN ACIDES AMINÉS

1

PLAN

OBJECTIFS

- Savoir calculer la masse d'une protéine
- Savoir tracer une courbe théorique de titrage d'une protéine
- Prédire le pHi d'une protéine

1.1 ACIDES AMINÉS ET SÉQUENCE

Les protéines naturelles sont constituées d'**acides aminés** de série L de structure chimique générale donnée dans la figure 1.1.

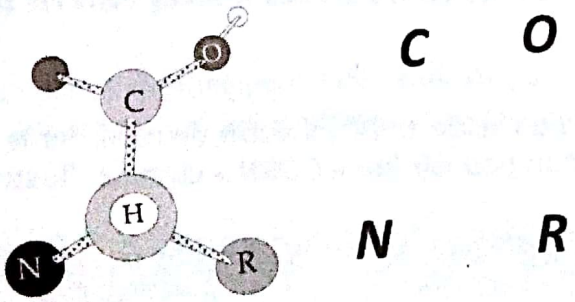


Figure 1.1 – Structure chimique d'un acide aminé de série L.

Il existe 20 acides aminés principaux dans les protéines naturelles. La correspondance entre les acides aminés, leur abbréviation et leur structure chimique est donnée dans la figure 1.2. Avec les ambiguïtés (Aspartate/Asparagine, Glutamate/Glutamine) et lorsque l'acide aminé est inconnu, ce sont au total 25 lettres qui sont utilisées (la lettre O désigne la pyrolysine et U la sélénocystéine).

A	Alanine	Ala
C	Cysteine	Cys
D	Aspartic Acid	Asp
E	Glutamic Acid	Glu
F	Phenylalanine	Phe
G	Glycine	Gly
H	Histidine	His
I	Isoleucine	Ile
K	Lysine	Lys
L	Leucine	Leu
M	Methionine	Met
N	Asparagine	Asn
O	Pyrrolysine	Pyl
P	Proline	Pro
Q	Glutamine	Gln
R	Arginine	Arg
S	Serine	Ser
T	Threonine	Thr
U	Sélénocystéine	Sec
V	Valine	Val
W	Tryptophane	Trp
Y	Tyrosine	Tyr
B		Asn/Asp
Z		Gln/Glu
X	Inconnu	

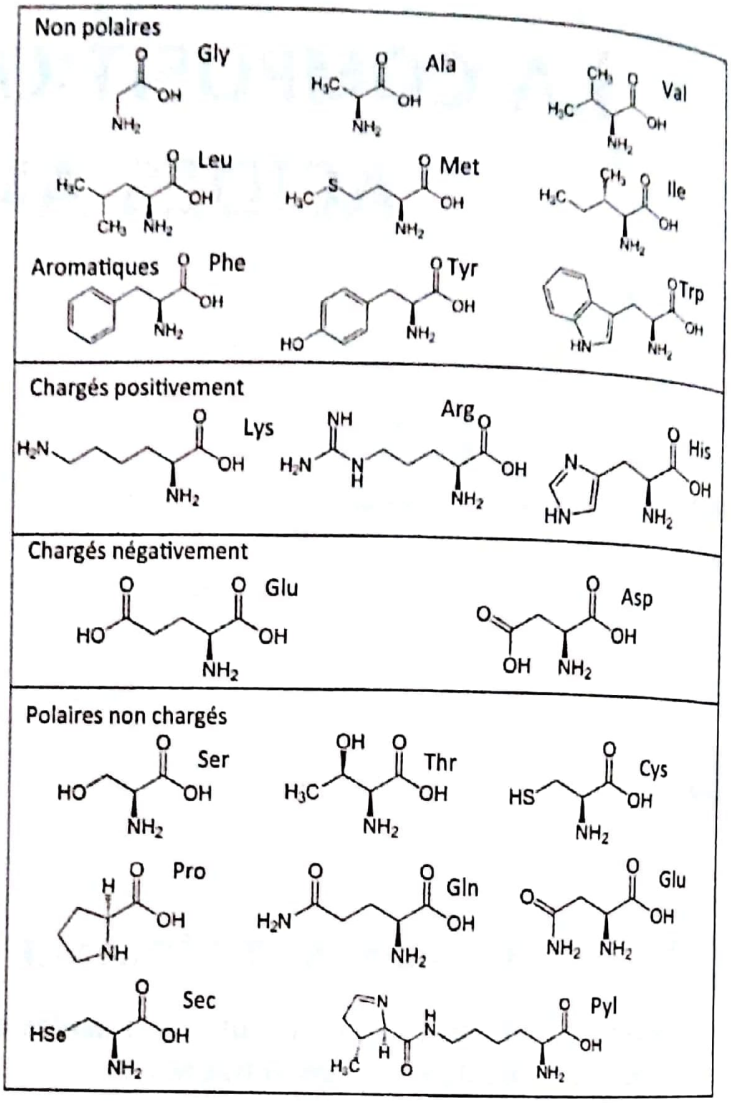


Figure 1.2 – Correspondance entre CODE 1 lettre, CODE 3 lettres et la structure chimique des acides aminés trouvés dans les protéines.

Pour identifier la série d'un acide aminé, il suffit de regarder le Cα avec le H devant les autres atomes. On doit pouvoir lire « CORN » comme illustré dans la figure 1.1.

Certains acides aminés partagent des propriétés physico-chimiques avec d'autres. Cela conduit à une distribution des groupes d'acides aminés selon le diagramme (non exclusif) de Venn schématisé figure 1.3.

Au niveau chimique, les protéines sont obtenues par condensation des acides aminés et élimination d'eau lors de la formation de la liaison peptidique (pour chaque acide aminé ajouté). La suite des lettres indiquant l'enchaînement des acides aminés constitue la **séquence** de la protéine (on parle aussi de **structure primaire**). Chaque séquence caractérise de manière unique une protéine. Une infime partie des séquences théoriquement possibles existe vraiment. Ce sont celles qui ont été sélectionnées par l'évolution et qui sont douées d'une activité biologique (structurelle et/ou fonctionnelle).

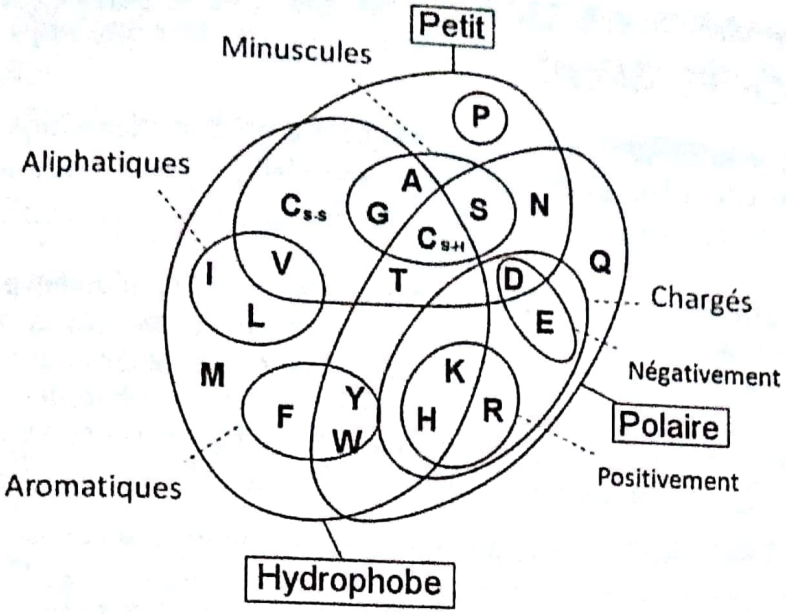


Figure 1.3 – Diagramme de Venn des propriétés des acides aminés.

Le génome humain comprend $3{,}4.10^9$ bases et coderait pour 20 563 séquences protéiques.

La bioinformatique s'est emparée très tôt de la comparaison des séquences. En effet, au sens informatique, il s'agit principalement de comparer des mots entre eux, rechercher des mots communs, trouver le plus grand mot commun, aligner les mots en autorisant des « jokers » à certaines positions.

Le nombre de séquences de longueur 100 réalisable à partir de 20 acides aminés différente (20^{100}) est supérieur au nombre d'atomes dans l'Univers ($\sim 10^{80}$).

ENCART Combien de séquences protéiques différentes peut-on générer en théorie ?

Le nombre de séquences différentes de longueur N qu'il est possible de générer en prenant les 20 acides aminés principaux est 20^N.

Exemples :

Peptide (5 acides aminés) : 20^5

Protéine de taille standard moyenne de 400 acides aminés : 20^{400}

Protéome humain (soit ~20 000 protéines de longueur moyenne 400) : $20^{8\,000\,000}$

1.2 Informations déduites de la composition en acides aminés

La première information dérivable d'une séquence est la composition en acides aminés. Cette composition (nombre et pourcentage de chacun des acides aminés) peut aussi être obtenue expérimentalement par des méthodes d'analyse biochimiques.

Si la composition en acide aminé d'une protéine X est biaisée par rapport à la composition moyenne de l'ensemble des protéines, on dit que la protéine X présente une **faible complexité**. Cette faible complexité peut aussi ne concerner qu'une partie de la séquence. Ainsi, dans certains récepteurs stéroïdiens, on observe jusqu'à 37 glutamines consécutives constituant un cas extrême de faible complexité.

Tableau 1.1 - **Les pKa des acides aminés ionisables.**

i	pKa i	j	pKa j
His	6,00	Ser	13,60
Arg	12,48	Tyr	10,10
Lys	10,53	Glu	4,20
N_{ter}	9,80	Thr	13,60
		Asp	3,86
		C_{ter}	2,10
		Cys	8,33

La composition permet au biochimiste de calculer la masse moléculaire théorique M de la protéine en utilisant la relation suivante :

$$M = \sum_{i=1}^{N} m(i) - 18 \times (N-1)$$

où $m(i)$ est la masse moléculaire de l'acide aminé i et N le nombre d'acides aminés. Connaissant la composition en acides aminés, le coefficient ε_{280} d'extinction molaire à 280 nm se calcule grâce à la relation suivante :

$$\varepsilon_{280} = [N_{Trp} \times 5\,500] + [N_{Tyr} \times 1\,490] + [N_{Cys} \times 125].$$

Il est alors possible de doser précisément par spectrophotométrie (densité optique) la concentration en protéine grâce à la relation de Beer-Lambert :

$$DO_{280} = \varepsilon_{280}\, L\, C$$

où L est la longueur du trajet optique, C la concentration en g/l.

Enfin, le pI (ou **point isoélectrique** d'une protéine) correspond à la valeur de pH telle que $NC = 0$ dans la relation suivante :

$$NC = \sum_i Ni\left(1 - \frac{10^{-pka(i)}}{10^{-pka(i)} + 10^{-pH}}\right) - \sum_j Nj\left(\frac{10^{-pka(j)}}{10^{-pka(j)} + 10^{-pH}}\right)$$

NC est le nombre de charges théoriques portées par la protéine.

i désigne un résidu qui peut être chargé positivement (Arg, Lys, His) ayant un pKa(i).
j désigne un résidu qui peut être chargé négativement Asp, Glu, Tyr, Cys, Ser, Thr ayant un pKa(j).

À partir de cette relation, il est possible de calculer la **courbe de titrage** théorique (*NC*) = f (pH) d'une protéine. Cette information même très approximative est très utile au biochimiste avant de se lancer dans une purification de protéine car la physico-chimie des solutions fait que solubilité d'une protéine est minimale quand le pH de la solution est égal au pHi. Par ailleurs, la connaissance du pHi d'une protéine permet de choisir une colonne de purification de type échangeuse d'ions qui soit adaptée aux conditions de pH utilisées pendant la purification.

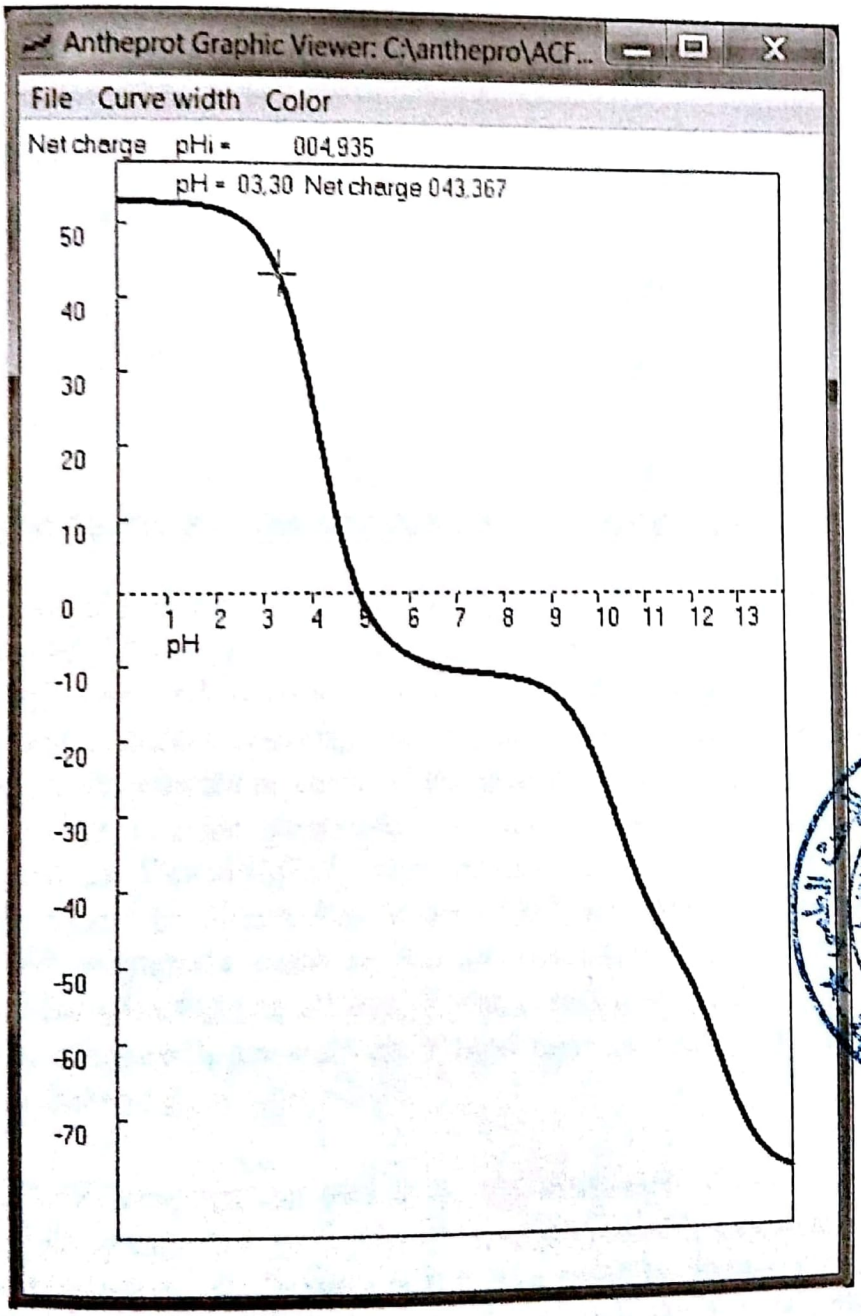


Figure 1.4 – Courbe de titrage théorique d'ATPA_TOBAC.
La courbe représente le nombre de charges théoriques portées par la protéine en fonction du pH. Le point isoélectrique est le pH pour lequel le nombre de charge est égal à 0 (ici 4,98).

BASES DE DONNÉES POUR DONNÉES DE BASES

2

PLAN

OBJECTIFS

- Comprendre l'intérêt des banques de données en biologie
- Connaître une entrée au format Swiss-Prot
- Savoir interroger les banques de données de séquences

2.1 LES BANQUES DE DONNÉES GÉNÉRALISTES

La problématique des données en biologie est très différente de celle d'autres disciplines. Les données biologiques présentent une forte hétérogénéité, ce qui pose la question de l'information à en tirer, de leur structuration et des systèmes de requêtes à développer pour pouvoir interroger de manière pertinente ces données. De plus, elles sont fortement corrélées entre elles (exemple des séquences nucléiques et protéiques à travers le code génétique). La qualité des données est très variable (erreur de séquences, d'**annotation**, redondance). Pour les protéines, il existe principalement trois manières différentes d'interroger les banques de séquences : par l'annotation des séquences dans la banque (commentaires, mots-clés associés) comme illustré dans les figures 2.1 et 2.5, par comparaisons directes des séquences décrites dans le chapitre 4, par numéro d'accession ou identifiant unique (exemple du champ AC décrit au paragraphe 2.2).

Exemple d'erreur dans les banques de données

Pour mettre en évidence la présence d'erreur dans les banques de séquences, l'utilisateur peut faire une requête sur le site de l'EBI (http://srs.ebi.ac.uk) avec comme mot-clé « psuedogene » au lieu de « pseudogene ». La requête suivante effectuée sur l'EMBL fournit plus de 80 entrées en 2012) !

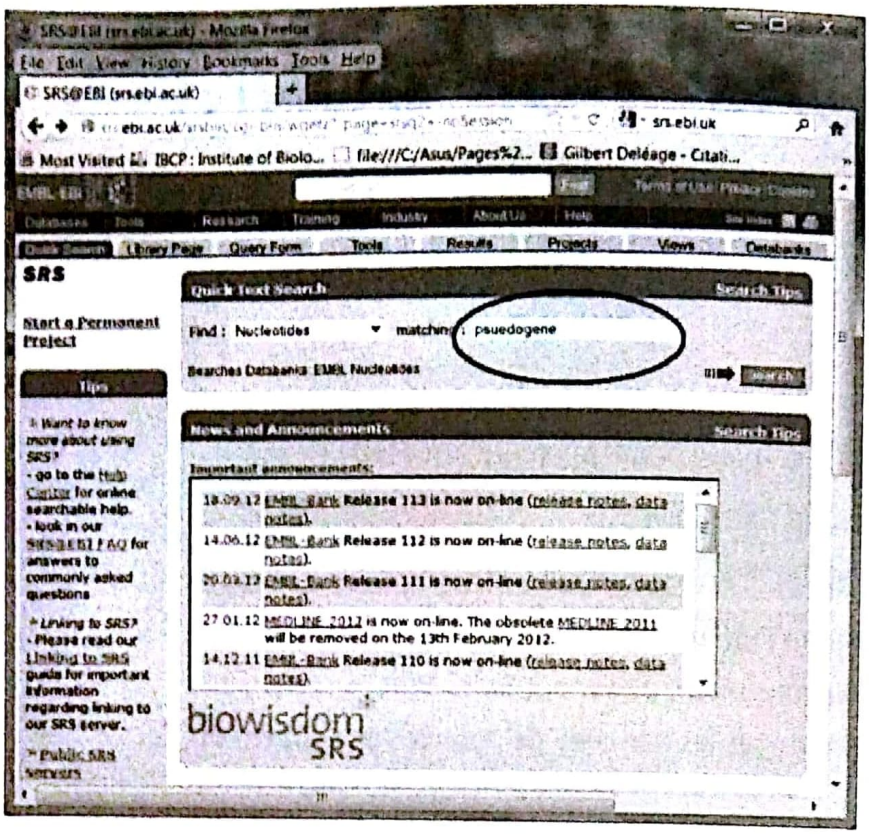


Figure 2.1 – Interrogation de banques nucléiques sur le serveur SRS de l'EBI avec le mot-clé « psuedogene ».

En biologie, de nouveaux types de données issus des progrès technologiques (puces, spectrométrie de masse, imagerie médicale) émergent constamment. Ces nouveaux types de données émergents sont fortement associés aux appareils (par exemple les puces Affymetrix ou les appareils de spectrométrie de masse) et aux auteurs qui les produisent, ce qui génère des formats de données différents et le plus souvent incompatibles car souvent liés à des constructeurs d'appareils.

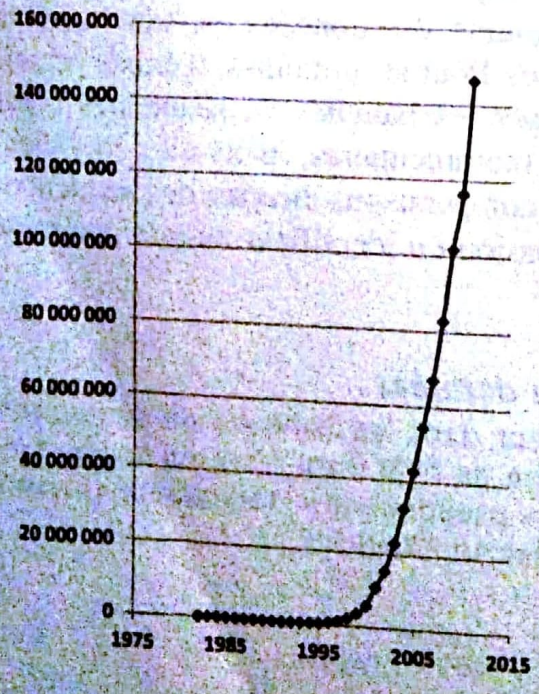
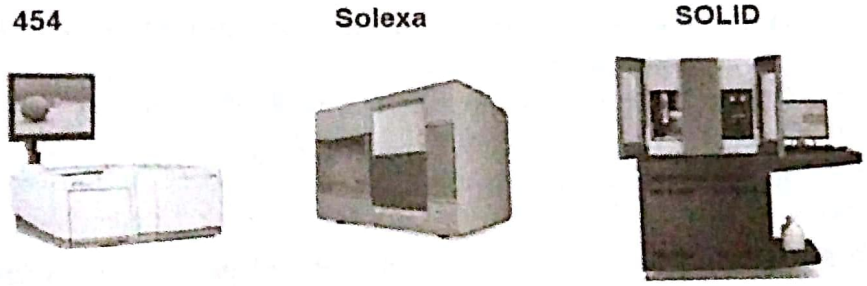


Figure 2.2 – Progression du nombre de séquences dans GENBANK.

De plus, le volume des données en biologie (en particulier les séquences) croît de manière exponentielle et double tous les 18 mois imposant au bioinformaticien de refaire périodiquement les analyses. Cette croissance pourtant déjà considérable est bien inférieure à celle liée aux séquençages massifs. Tout d'abord, les programmes de séquençage massif de génomes complets font exploser les volumes acquis. De plus, les capacités d'obtention de séquences par les nouvelles techniques de séquençage (NGS ou *Next Generation Sequencing*) sont telles que les coûts des séquençages ont été divisés par 10 000 depuis 2008 pour atteindre 1 $ pour 10 Mb séquencés. À titre de comparaison, en 2001, le coût de 1 Mb était de 8 000 $! Les nouvelles méthodes de séquençages illustrées dans la figure 2.3 présentent des caractéristiques de taille de séquence, de longueur de lecture (« read »), de temps d'obtention et de degré de parallélisation différents. Toutes ces méthodes permettent de générer un grand nombre de fragments de longueur variable selon la technologie qui seront assemblés par bioinformatique pour finalement donner la séquence (cas d'un petit génome) ou pour positionner la séquence sur un génome de référence.



	454	Solexa	SOLID
Technologie	: Pyroséquençage	Fluorescence	Solide
Parallélisation	: $4\ 10^5$	$3\ 10^7$	$5\ 10^7$
Nucléotides par lecture	: ~400	~50	35
Temps d'obtention	: 8 H	144 H	240 H
Longueur de séquence	: $\sim 5\ 10^8$	$\sim 4\ 10^9$	$\sim 2\ 10^{10}$

Figure 2.3 – *Next Generation Sequencing* (NGS).

Il faut souligner que ces technologies progressent tant sur le plan de la longueur des « read » que sur le degré de parallélisation. Les coûts ont aussi chuté au point que dans un avenir proche, la séquence du génome complet d'un humain coûtera environ 500 €, ce qui ouvre des perspectives de médecine personnalisée mais pose aussi des questions éthiques importantes. Par ailleurs, de nouvelles approches sont en cours de développement (Ion Proton ou GridION™) et permettent une miniaturisation encore plus grande du système (MinIOn USB) et une parallélisation par empilement des unités (comme pour les calculateurs). Dans cette course aux génomes, il ne faut pas perdre de vue que séquencer n'est pas déchiffrer.

Par ailleurs, la sémantique et la représentation d'un concept ou d'une notion varient selon la culture scientifique, ce qui est une difficulté pour une interdiscipline.

Ainsi, la définition même de ce qu'est une protéine est différente pour l'informaticien (qui voit souvent un mot), pour un biologiste (qui voit un intermédiaire dans une chaîne fonctionnelle), pour le biochimiste (qui y associera une activité enzymatique) et pour un chimiste (qui y associera un assemblage d'un grand nombre d'atomes).

Le programme 10 000 génomes humains

À titre d'illustration, le programme 10 000 génomes humains (http://www.uk10k.org/) lancé en 2010 par le Sanger Institute pour étudier la variabilité génétique humaine a généré en 6 mois un volume de données équivalent au contenu accumulé dans GENBANK pendant 20 ans ! D'autres projets de séquençage sont en cours comme le séquençage de 10 000 génomes de vertébrés.

La mouvance des données biologiques (quantité et qualité) oblige de refaire régulièrement les analyses bioinformatiques.

L'information biologique est :

- disséminée dans une multitude de banques de données ;
- stockée sous des formats syntaxiquement hétérogènes ;
- en général non disponible dans des systèmes de gestion de bases de données (SGDB) mais distribuée sous forme de fichiers plats ;
- modélisée dans ces différentes banques selon des sémantiques hétérogènes et difficiles à mettre en relation.

Au début de la biologie moderne, les séquences nucléiques et protéiques étaient déposées dans un grand livre édité par Margaret Dayhoff. Cet atlas des séquences a été remis à jour périodiquement jusqu'en 1978. Les premières banques informatisées de données de séquences biologiques ont été développées à Lyon par C. Gautier dans les années 1980 au Laboratoire de Biométrie et de Biologie Évolutive. Depuis, plusieurs initiatives européennes (EMBL, devenue aujourd'hui l'ENA), américaine (GenBank) ou japonaise (DDBJ) ont émergé de manière concurrente et parallèle pour collecter l'ensemble des séquences génomiques. Depuis 1995, ces trois organisations ont passé des accords d'échanges mutuels de données, ce qui a pour résultat que toute nouvelle séquence incluse dans une banque est automatiquement intégrée dans les deux autres. Aujourd'hui, les trois banques font partie du consortium International Nucleotide Sequence Databases Collaboration (INSDC). Ce consortium fait que les trois banques ayant un souci d'exhaustivité ont un contenu quantitatif et qualitatif assez comparable et qui a tendance à converger. Les deux plus grands centres de bioinformatique du monde sont l'Institut Européen de Bioinformatique (EBI) à Hinxton, au Royaume-Uni (http://ebi.ac.uk/), et le National Center for Biotechnology Information (NCBI), à Bethesda aux États-Unis (http://ncbi.nlm.nih.gov/), qui rassemblent la plupart des banques de données. Enfin, depuis 1986, il faut souligner l'initiative d'A. Bairoch de créer une banque de séquences de protéines Swiss-Prot (http://www.uniprot.org/) devenue **UniProtKB/Swiss-Prot** qui soit non redondante et de haute qualité car riche en **annotations fonctionnelles** et

structurale et intégrant les informations des autres banques de données. Du fait de sa faible redondance, cette banque est particulièrement utile pour établir des statistiques sur les protéines. Les premières banques de données (pas encore des bases de données) étaient généralistes.

Différence entre base de données et banque de données

Une banque de données est un ensemble de fichiers textes sans relation entre eux (on parle de fichier « plat »). Une base de données est un ensemble de relations entre des données gérées avec un système de gestion de base de données (SGBD) et interrogeable par SQL (*Structure Query Langage*).

Depuis 25 ans, une explosion des bases de données spécialisées est observée (1 380 répertoriées dans NAR).

La revue *Nucleic Acids Research* consacre un numéro spécial « database » chaque année (http://www.oxfordjournals.org/nar/). Avant de se lancer dans un nouveau projet, il convient de vérifier qu'il n'existe pas une banque spécialisée maintenue à jour.

Les bases de données spécialisées présentent l'avantage d'être maintenues par des experts du domaine qui gèrent les problèmes de numérotation, nomenclature, cohérence, annotation. On peut distinguer les bases de données thématiques biologiques (récepteurs couplés aux protéines G comme GPCR, ou immunologie IMGT), par organisme (dont le génome est en général complètement séquencé), par technologie (spectres RMN, cartes de spectrométrie de masse, gels d'électrophorèse bidimensionnelle) ou par type (séquence, structure, image, spectre, interaction). Le tableau 2.1 recense quelques ressources notoires en bioinformatique.

L'accès aux génomes se fait grâce à des outils dédiés appelés *genome browser*. Le serveur Ensembl (www.ensembl.org) répertorie les principaux génomes d'organismes modèles. Le serveur offre la possibilité de naviguer depuis le niveau caryotype (figure 2.4) jusqu'au niveau de la séquence nucléique et de sa traduction dans les différentes phases de lecture.

Il existe une seule banque de données des structures 3D des macromolécules biologiques appelée historiquement la PDB (*Protein Data Bank*). Cette banque (http://www.rcsb.org/) contient les coordonnées tridimensionnelles atomiques de protéines, d'acides nucléiques, de complexes nucléo-protéiques, de sucres. La croissance de la banque est constante depuis 5 ans et représente environ 7 500 structures/an en moyenne sur la période 2007-2011. En revanche, le nombre de structures présentant une architecture originale (**repliement** ou *fold*) est constant. Ainsi, le nombre de repliements différents connus est d'environ 1 500 et représente la redondance en structures 3D. La redondance en séquence fait qu'on peut distinguer environ 20 000 groupes de séquences qui partagent entre eux moins de 30 % d'identité. Ainsi, il existe des versions de PDB à 95 % (PDB95), 75 % (PDB75) et 25 % (PDB25).

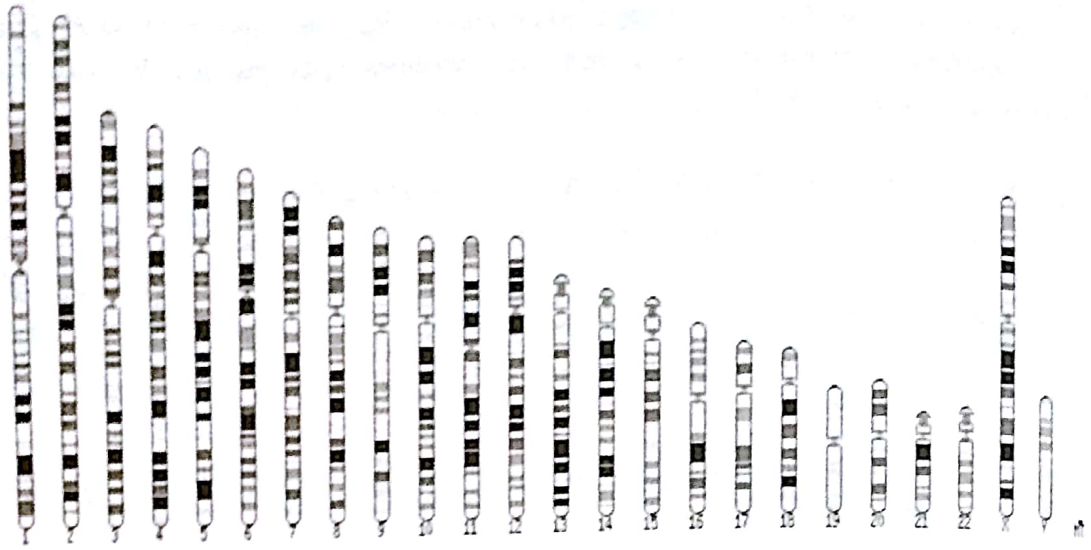


Figure 2.4 – **Caryotype humain sur Ensembl (23 chromosomes).**

Les données biologiques sont fortement biaisées. À titre d'illustration, même si plus de 11 000 espèces sont représentées dans **UniProtKB**, seulement 20 espèces couvrent 33 % des entrées (voir tableau 2.2).

Tableau 2.1 – **Quelques bases de données spécialisées.**

Acronyme	Description
IMGT	IG, récepteur de cellules T, Complexe Majeur d'Histocompatibilité
HIV	Base de séquences sur le SIDA à Los Alamos
GPCRDB	Récepteurs couplés aux protéines G
euHCVdb	Base de données de séquences du virus de l'hépatite C
OMIM	*Online Mendelian Inheritance in Man*
HGMD	*Human Gene Mutation Database*
KEGG	Kyoto Encyclopedia of Genes and Genomes
ENZYME	Nomenclature des enzymes
BRENDA	Base de connaissance sur les enzymes
NRSub	*Bacillus subtilis*
AceDB	*Caenorhabditis elegans*
FlyBase	*Drosophila melanogaster*
GOLD	Banque des génomes séquencés.
RCSB	Base de données des structures des macromolécules biologiques
IntAct	Base de données d'interactions protéiques
BIND	Base d'interactions
MiMI	Banque d'interactions moléculaires du Michigan
STRING	Banque d'interactions entre protéines
CATH	Banque de classification des structures de protéines
SCOP	Classification structurale des protéines
Ensembl	Explorateur de génomes complets d'organismes modèles.
NucleaRDB	Système d'information pour les récepteurs nucléaires

Tableau 2.2 - Le top 20 des séquences par espèce représentée dans UniProtKB.

N°	Nombre	Nom de l'espèce
1	20 331	*Homo sapiens* (Human)
2	16 072	*Mus musculus* (Mouse)
3	7 723	*Arabidopsis thaliana* (Mouse-ear cress)
4	7 269	*Rattus norvegicus* (Rat)
5	6 552	*Saccharomyces cerevisiae* (Baker's yeast)
6	5 559	*Bos taurus* (Bovine)
7	4 752	*Schizosaccharomyces pombe* (Fission yeast)
8	4 342	*Escherichia coli* (strain K12)
9	3 576	*Bacillus subtilis*
10	3 410	*Dictyostelium discoideum* (Slime mold)
11	3 212	*Caenorhabditis elegans*
12	2 921	*Xenopus laevis* (African clawed frog)
13	2 883	*Drosophila melanogaster* (Fruit fly)
14	2 374	*Danio rerio* (Zebrafish) (Brachydanio rerio)
15	2 184	*Pongo abelii* (Sumatran orangutan)
16	2 089	*Gallus gallus* (Chicken)
17	1 981	*Oryza sativa* subsp. japonica (Rice)
18	1 974	*Escherichia coli* O157:H7
19	1 782	*Methanocaldococcus jannaschii* (jannaschii)
20	1 773	*Haemophilus influenzae*

Tableau 2.3 - Nombre de structures 3D déposées dans la PDB.

Méthode	Protéines	Acides nucléiques	Complexes Prot/A.Nuc.	Autres	Total
Cristallographie rayons X	71 626	1 424	3 653	3	76 706
Résonance magnétique Nucléaire	8 529	1 014	191	7	9 741
Cryo-Microscopie électronique	337	29	123	0	489
Méthodes hybrides	45	3	2	1	51
Autres	144	4	5	13	166
Total	**80 681**	**2 474**	**3 974**	**24**	**87 153**

La diversité des banques et des logiciels de traitement de données à conduit à la création de plusieurs formats. Des formats sont adaptés pour les logiciels de traitement de séquences (exemple format Pearson-Fasta) car ils sont économiques en taille puisqu'ils ne contiennent que la séquence et une ligne de description, mais peu informatifs. D'autres formats dédiés aux banques sont très informatifs (beaucoup d'annotations) mais peu économiques en taille et donc peu utilisés par les logiciels d'analyse de séquence.

Tableau 2.4 - **Principaux formats des séquences.**

Banques de séquences	Phylogénie	Logiciels
Ig/Stanford	Phylip3.2	Fitch
Genbank/GB	Phylip	DNA strider
NBRF	Plain/Raw	AnTheProt
EMBL	PIR/CODATA	Olsen
GCG	MSF	Pretty
Pearson/Fasta	PAUP	Zuker

Le format standard et commun à tous les logiciels d'analyse en bioinformatique est Person/Fasta. Les banques de données sont aussi proposées dans ce format.

Format PEARSON-FASTA compatible avec tous les logiciels d'analyse

```
>sw|P02159|MYG_LYCPI Myoglobin.
GLSDGEWQIVLNIWGKVETDLAGHGQEVLIRLFKNHPETLDKFDKFKHLKTEDEMKGSED
LKKHGNTVLTALGGILKKKGHHEAELKPLAQSHATKHKIPVKYLEFISDAIIQVLQNKHS
GDFHADTEAAMKKALELFRNDIAAKYKELGFQG
>sw|Q9DEP1|MYG_PSEGE Myoglobin.
ADFDMVLKCWGLVEADYATYGSLVLTRLFTEHPETLKLFPKFAGIAHGDLAGDAGVSAHG
ATVLNKLGDLLKARGGHAALLKPLSSSHATKHKIPIINFKLIAEVIGKVMEEKAGLDAAG
QTALRNVMAVIIADMEADYKELGFTE
>sw|P02201|MYG_GRAGE Myoglobin.
GLSDDEWHHVLGIWAKVEPDLSAHGQEVIIRLFQVHPETQERFAKFKNLKTIDELRSSEE
VKKHGTTVLTALGRILKLKNNHEPELKPLAESHATKHKIPVKYLEFICEIIVKVIAEKHP
SDFGADSQAAMRKALELFRNDMASKYKEFGFQG
```

Un utilitaire de conversion de formats (READSEQ) est proposé par D. Gilbert (http://www.ebi.ac.uk/cgi-bin/readseq.cgi).

Le biologiste doit aussi faire attention aux caractères de fin et de saut de ligne qui sont différents selon les systèmes d'exploitation. Pour convertir un fichier issu d'un serveur Linux en un fichier MS Windows, il suffit de le charger dans Wordpad et de le sauver. Les caractères seront automatiquement substitués.

2.2 Une entrée SWISS-PROT

Dans une entrée Swiss-Prot (voir exemple pages suivantes), chaque fichier de séquence obéit à un format propre à base d'étiquette (deux lettres) qui renseigne la nature du champ d'information qui débute à la colonne 6 (http://web.expasy.org/docs/userman.html#linetypes). La première étiquette est **ID** (IDentifiant). Elle contient le nom de la protéine. Un nom Swiss-Prot est constitué d'un préfixe souvent évocateur du rôle ou de la fonction (ici **MYG** pour **MYOGLOBIN**), d'un séparateur, le « _ » (caractère underscore ou blanc souligné), et du nom (ou de son abréviation) **HUMAN** de l'espèce (en anglais). Attention, ce nom est susceptible de changer au cours des différentes versions de la banque. En effet, il se peut que la fonction ne soit pas connue avec précision à une date donnée et que celle-ci soit étudiée et finalement connue dans une version suivante. Le champ **AC** (numéro d'ACcès) est affecté de

manière définitive à une séquence et n'est pas susceptible de changer. En conséquence, il faut toujours associer le champ AC au champ ID dans toute communication. Les trois champs **DT** (**DaTe**) renseignent successivement les différentes dates concernant l'entrée (création, modification de séquence ou d'annotation). Le champ **DE** (**DEscripteur**) renseigne sur la nature de la protéine et est en général la ligne retournée par les programmes d'analyses (BLAST ou FASTA). Le champ **GN** (*Gene Name*), le champ **OS** contient le nom (latin et anglais) de l'espèce et de l'organisme (*Organism Specie*). Le champ **OG** (ici absent) désigne l'organite. Le champ **OC** correspond à la classification de l'organisme de la séquence. Le champ **OX** correspond à la taxonomie de l'organisme. Les différents champs **RN** (RP, RX, RT, RA, RL) concernent les références bibliographiques de séquences. Le champ **CC** est dédié aux commentaires (copyright ou annotations). La ligne **DR** fournit des liens croisés sur les autres banques de données. Le champ **KW** (**KeyWord** ou mot-clé). Le champ **FT** (**Feature Table**) est pour les informations et les annotations concernant la séquence. Si les informations sont non vérifiées expérimentalement, le mot « potential » ou « conflict » est ajouté. Enfin, le dernier champ est **SQ** pour **SéQuence**. Le terminateur d'entrée est « // ».

Plus récemment, le champ **OH** (*Organism Host*) a été ajouté pour les entrées virales et décrit l'hôte du virus avec la taxonomie du NCBI. Le champ **PE** (*Protein Evidence*) décrit le mode de mise en évidence de la protéine :

1 : par la présence de la protéine
2 : par la présence du transcrit
3 : déduite par homologie
4 : prédite
5 : incertaine

```
ID   MYG_HUMAN      STANDARD;      PRT:   153 AA.
AC   P02144;
DT   21-JUL-1986 (Rel. 01, Created)
DT   21-JUL-1986 (Rel. 01, Last sequence update)
DT   01-MAR-2002 (Rel. 41, Last annotation update)
DE   Myoglobin.
GN   MB.
OS   Homo sapiens (Human).
OC   Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi;
OC   Mammalia; Eutheria; Primates; Catarrhini; Hominidae; Homo.
OX   NCBI_TaxID=9606;
RN   [1]
RP   SEQUENCE.
RX   MEDLINE=71291923; PubMed=5285572;
RA   Romero-Herrera A.E., Lehmann H.;
RT   "Primary structure of human myoglobin.";
RL   Nature New Biol. 232:149-152(1971).
RN   [2]
RP   REVISIONS TO 19-22 AND 83.
RA   Romero-Herrera A.E., Lehmann H.;
RT   "The myoglobin of primates. I. Hylobates agilis (gibbon).";
RL   Biochim. Biophys. Acta 251:482-488(1971).
RN   [3]
..../...
```

```
CC   - !- FUNCTION: SERVES AS A RESERVE SUPPLY OF OXYGEN AND FACILITATES
CC       THE MOVEMENT OF OXYGEN WITHIN MUSCLES.
CC   - !- SIMILARITY: BELONGS TO THE GLOBIN FAMILY.
CC   ---------------------------------------------------------------------
CC   This SWISS-PROT entry is copyright. It is produced through a
      collaboration
CC   between  the Swiss Institute of Bioinformatics  and the  EMBL outstation
      -
CC   the European Bioinformatics Institute.  There are no  restrictions on
      its
CC   use  by  non-profit  institutions as long  as its content  is  in  no
      way
CC   modified and this statement is not removed.  Usage  by  and for
      commercial
CC   entities requires a license agreement (See http://www.isb-sib.ch/
      announce/
CC   or send an email to license@isb-sib.ch).
CC   ---------------------------------------------------------------------
DR   EMBL; M14603; AAA59595.1; -.
DR   EMBL; M10090; AAA59595.1; JOINED.
DR   EMBL; M14602; AAA59595.1; JOINED.
DR   EMBL; X00371; CAA25109.1; -.
DR   EMBL; X00372; CAA25109.1; JOINED.
DR   EMBL; X00373; CAA25109.1; JOINED.
DR   EMBL; AL049747; CAB41872.1; -.
DR   EMBL; AL022334; CAA18457.1; -.
DR   PIR; A02464; MYHU.
DR   PDB; 2MM1; 15-JAN-93.
DR   HSC-2DPAGE; P02144; HUMAN.
DR   MIM; 160000; -.
DR   InterPro; IPR000971; Globin.
DR   InterPro; IPR002335; Myoglobin.
DR   Pfam; PF00042; globin; 1.
DR   PRINTS; PR00613; MYOGLOBIN.
DR   PROSITE; PS01033; GLOBIN; 1.
KW   Heme; Oxygen transport; Transport; Muscle; Polymorphism;
KW   3D-structure.
FT   INIT_MET      0      0
FT   METAL        64     64       IRON (HEME DISTAL LIGAND).
FT   METAL        93     93       IRON (HEME PROXIMAL LIGAND).
FT   VARIANT      54     54       E -> K.
FT                                 /FTId=VAR_003180.
FT   VARIANT     133    133       K -> N.
FT                                 /FTId=VAR_003181.
FT   VARIANT     139    139       R -> Q.
FT                                 /FTId=VAR_003182.
FT   VARIANT     139    139       R -> W.
FT                                 /FTId=VAR_003183.
FT   CONFLICT    128    128       Q -> E (IN REF. 4).
FT   HELIX         4     17
FT   TURN         18     19
FT   HELIX        21     35
FT   HELIX        37     41
FT   TURN         42     42
FT   TURN         45     48
FT   HELIX        52     57
FT   HELIX        59     76
FT   TURN         77     80
FT   HELIX        83     95
FT   TURN         96     96
FT   TURN        101    101
FT   HELIX       102    118
```

```
FT   HELIX        120     122
FT   HELIX        125     148
FT   TURN         149     150
SQ   SEQUENCE   153 AA;  17053 MW;  5F84A2C481B8F0D5 CRC64;
     GLSDGEWQLV LNVWGKVEAD IPGHGQEVLI RLFKGHPETL EKFDKFKHLK SEDEMKASED
     LKKHGATVLT ALGGILKKKG HHEAEIKPLA QSHATKHKIP VKYLEFISEC IIQVLQSKHP
     GDFGADAQGA MNKALELFRK DMASNYKELG FQG
//
```

Lors d'échanges de listes de protéines, il est recommandé de fournir la liste des champs AC (avec éventuellement la liste des ID) afin de lever toute ambiguïté concernant les protéines concernées.

2.3 Les interrogations Entrez, ACNUC, SRS

Ces systèmes d'interrogations n'utilisent pas directement la séquence (comme traité dans le chapitre 4) mais les informations concernant les séquences (annotation) contenues dans les banques.

Les principaux systèmes d'interrogation sont EB-Eye (http://www.ebi.ac.uk/ebisearch), ACNUC (http://pbil.univ-lyon1.fr/databases/acnuc/acnuc.html), SRS en Europe et Entrez aux États-Unis (http://www.ncbi.nlm.nih.gov/sites/gquery).

Les systèmes EB-Eye, disponibles à l'EBI (European Bioinformatics Institute, situé au Royaume-Uni) et Entrez, disponible au NCBI (National Center for Biotechnology Information, aux États-Unis) permettent d'interroger depuis un navigateur Web toutes les banques de données de séquences à partir d'une ou plusieurs chaînes de caractères qui seront cherchées dans les annotations des séquences de toutes les banques. Ces systèmes, et particulièrement Entrez, permettent aussi d'interroger la banque de données MEDLINE réunissant tous les articles de la littérature scientifique médicale et une grande partie de la littérature biologique.

Le système Entrez permet d'interroger très rapidement des ressources très diverses et nombreuses comme par exemple PubMed pour la bibliographie en biologie et médecine, la banque des taxons, la banque Nucléotide GenBank, séquence et structure de protéine. L'avantage du système d'interrogation du NCBI est de proposer une interface unique, très simple (figure 2.5) qui permet de construire des requêtes complexes associant les différents champs (ou annotation) aux opérateurs logiques d'addition (AND), d'exclusion (OR) ou de négation (NOT). De plus, l'interface ainsi que la logique d'interrogation est conservée selon les domaines et la nature des ressources interrogées. De même, il est possible de combiner des requêtes pour croiser des listes de résultats.

Le système ACNUC permet une recherche multicritère sur une banque, qui peut être GenBank, ENA/EMBL ou UniProtKB/SwissProt. Cette recherche peut se faire à partir de listes de sélections successives sur lesquelles on peut appliquer des opérations logiques (AND, OR, NOT).

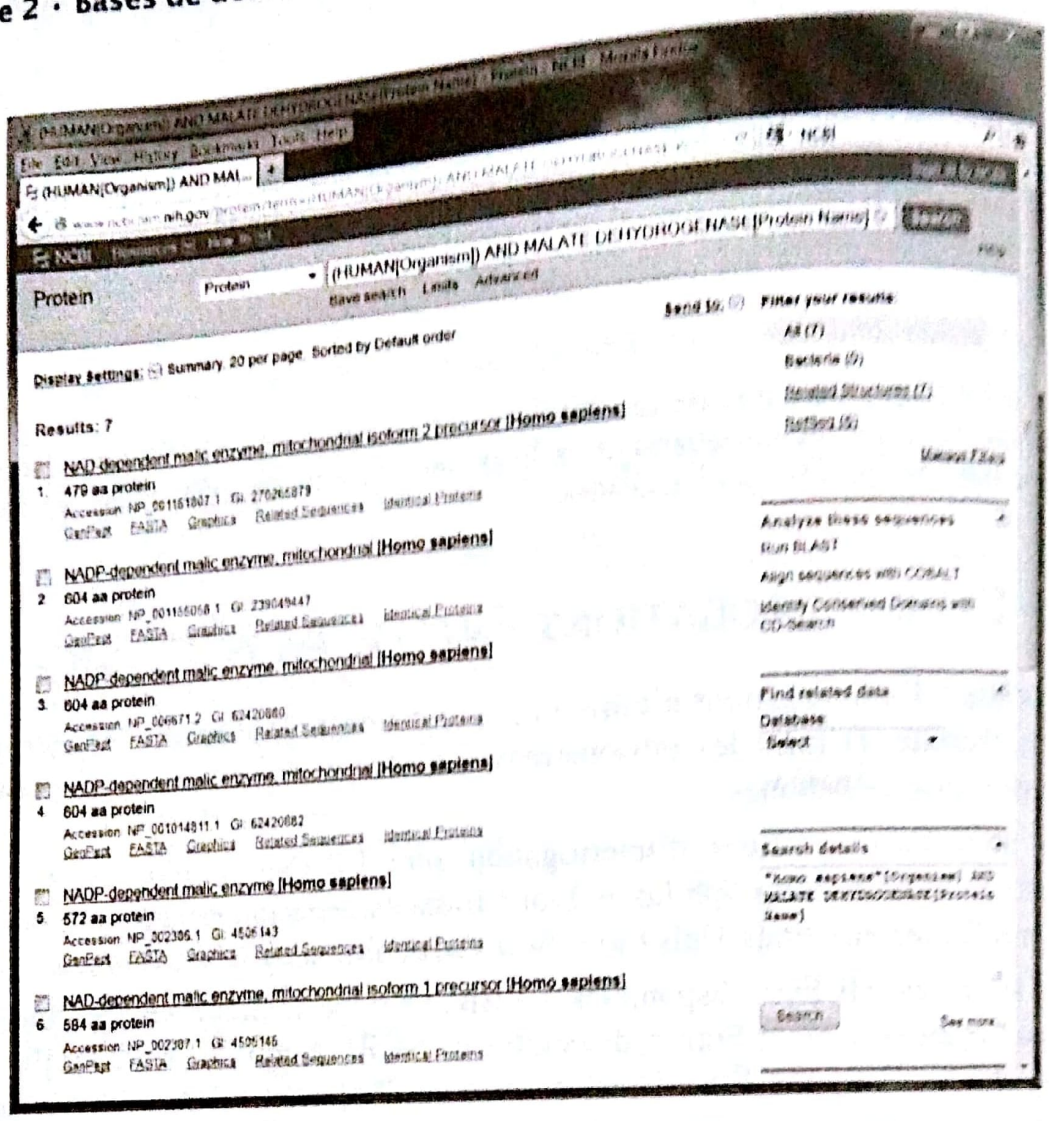


Figure 2.5 – Interrogation du NCBI par Entrez sur la base protéine.
Le champ Organism contient la valeur HUMAN et le champ Protein Name contient la valeur MALATE DEHYDROGENASE. L'opérateur de combinaison est le ET logique (AND).

Par exemple, la requête

```
sp=HIV2 and k=rev gene and t=CDS
```

permet de sélectionner toutes les parties codantes du gène rev dans les séquences de virus HIV2 dans la banque de séquences choisie. Le critère `sp=HIV2` sélectionne toutes les séquences de l'espèce virale HIV2. Le critère `k=mot-clé` sélectionne toutes les séquences dont l'annotation contient ce mot-clé. Le critère `t=CDS` conduit à extraire des séquences disponibles les seules parties codant une protéine, les éventuels introns étant aussi exclus. L'efficacité de la sélection de séquences par mot-clé est limitée par la variabilité des mots-clés utilisés entre séquences (dans l'exemple, le mot-clé « `rev protein` » est aussi utilisé). Le système ACNUC est accessible de trois façons :

- à travers une interface Web permettant de formuler des requêtes multicritères et d'extraire les séquences correspondantes ;

- à l'aide du programme client « query_win » qui permet d'interroger les banques à travers le réseau en utilisant une interface graphique ;
- à travers une interface programmable pour les langages C, C++, Python et R qui permet de requêter les banques en réseau.

Le système SRS (figure 2.1) a été développé initialement pour permettre l'interrogation simultanée de plusieurs banques. Ainsi, la recherche dans une banque implique que la requête soit effectuée sur la version stable de la banque et sur les mises à jour (*update*) permettant d'avoir une interrogation exhaustive. Un serveur SRS s'installe sur des sites hébergeant des banques et des outils. Le plus complet de ces serveurs SRS est celui de l'EBI (http://srs.ebi.ac.uk). ACNUC et SRS reposent sur un système d'indexation qui se caractérise par une bonne performance en lecture (typiquement à l'utilisation par le biologiste) mais aussi par une incapacité de modification dynamique du système d'index. Ces deux systèmes sont donc performants en mode lecture mais pas en mode écriture. À l'opposé, les Systèmes de Gestion de Bases de Données (SGBD) sont efficaces en écriture (et parfois un peu moins en lecture). La plupart des banques de données sont maintenant disponibles dans des SGBD (Oracle, Sybase, db2, MySQL, PostgreSQL). Cependant, les plus performants de ces SGBD sont commerciaux et tous nécessitent de savoir quels types d'interrogations devront être faits afin d'optimiser la structure de la base de données. De plus, le temps de chargement nécessaire en cas de changement d'architecture peut être très long (plusieurs jours pour EMBL). Les bases de données sont interrogeables à l'aide d'un langage standard de requête *Structured Query Language* (SQL) et la plupart des bases de données biologiques offrent une interface avec le Web *via* un formulaire le plus souvent écrit en python, PHP (PHP : *hypertext Preprocessor*), Perl ou Java. À noter qu'il existe aussi une tendance actuelle au NoSQL (*Not only SQL*) avec par exemple MongoDB (www.mongodb.org/) qui ne nécessite pas de schéma prédéfini des données et qui permet l'évolution des champs au cours du temps. Ce type d'outil est particulièrement utile pour des données ou des documents qui sont changeants et de volume important (*big data*) comme la biologie récente. Enfin, il faut mentionner le *cloud computing* ou nuage en plein développement pour affranchir l'utilisateur de lien direct avec une machine de calcul et qui stocke les données quelque part dans le « nuage » sur des systèmes informatiques distribués et inconnus de l'utilisateur.

LA COMPARAISON DE DEUX SÉQUENCES

3

PLAN

OBJECTIFS

- Savoir détecter des répétitions internes, duplications, transpositions, insertions
- Identifier les régions de faible complexité
- Connaître le principe d'une matrice de substitution
- Savoir choisir une matrice de substitution
- Savoir interpréter une matrice de points

3.1 MATRICE DE POINTS

La méthode de comparaison de deux séquences par matrice de points ou « dot-plot » consiste à écrire dans un tableau (ligne et colonne) une séquence selon un axe horizontal et l'autre (qui peut être la même) selon un axe vertical. Dans la méthode initiale décrite par R. Staden en 1982, on met un point à l'intersection d'une ligne et d'une colonne si et seulement si la lettre horizontale est identique à la lettre verticale (figure 3.1).

La matrice de points présentée dans la figure 3.1 montre que si la séquence est comparée face à elle-même, alors une diagonale parfaite sépare le plan en deux triangles symétriques. Dans ce cas, tout segment parallèle à cette diagonale représente une répétition interne ; ici le peptide **AEIGL** est présent deux fois (1-5) et (7-11). Ce type de matrice peut être aussi utilisé pour les acides nucléiques mais le diagramme est en général très « bruité ». Si les deux séquences sont différentes, on n'observe pas de diagonale et le diagramme n'est pas symétrique (cas des figures 3.2 à 3.6). Plusieurs phénomènes biologiques observés dans les protéines et les génomes peuvent ainsi être mis en évidence comme les répétitions internes (figures 3.2 et 3.7), l'**homologie** (figure 3.8), les **palindromes** (figure 3.4), les insertions (figure 3.3), les **transpositions** (figure 3.5 et 3.6) et la faible complexité (figure 3.9). Cette approche graphique a été progressivement supplantée par les alignements de séquences beaucoup plus faciles à interpréter par le biologiste. Néanmoins, ces matrices de points présentent l'avantage d'explorer sans *a priori* toutes les combinaisons

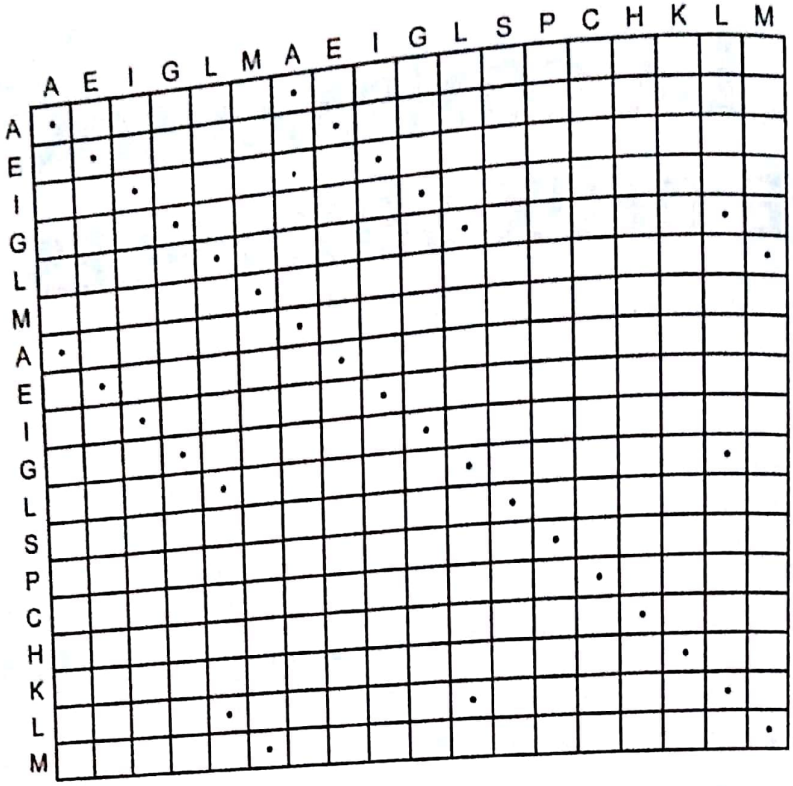


Figure 3.1 – Construction d'une matrice symétrique simple.

possibles de ressemblance entre deux séquences (et demandent une interprétation) alors que l'alignement de deux séquences est un des choix possibles parmi toutes les possibilités. Cependant, la capacité à réaliser des alignements multiples a offert la possibilité de gérer de gros jeux de données et a eu tendance à faire disparaître les matrices de points.

Homologie

Deux protéines sont homologues si et seulement si elles résultent de l'évolution à partir d'un ancêtre commun. En pratique, l'homologie entre deux protéines est inférée avec confiance lorsque le pourcentage d'identité entre leurs deux séquences est supérieur à 30 % et que l'alignement couvre 70 % des deux séquences. En revanche, des protéines partageant moins de 30 % d'identité peuvent être homologues (si le taux de mutation a été élevé au cours de l'évolution). L'homologie est donc une propriété intrinsèque et ne peut être qualifiée de forte ou de faible.

Palindrome

Un palindrome est un mot qui se lit dans les deux sens comme LAVAL ou ANA ou SIS. Ces palindromes sont fréquents dans les sites des acides nucléiques qui fixent des protéines. Le plus souvent, le palindrome en biologie est imparfait comme dans le cas du site de fixation du répresseur de l'opéron lactose.

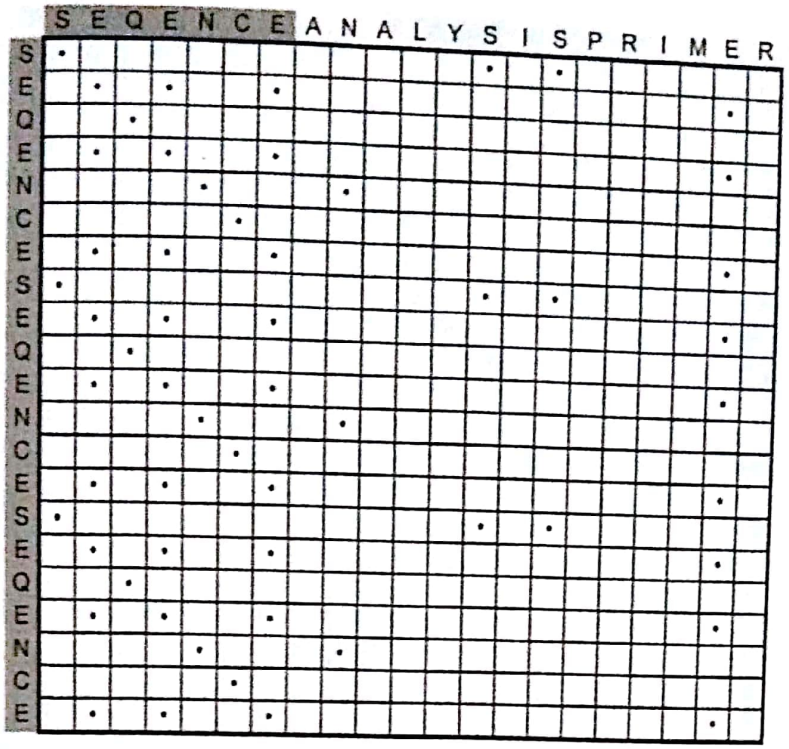


Figure 3.2 – Cas d'une répétition interne du segment « SEQENCE » dans deux séquences différentes.

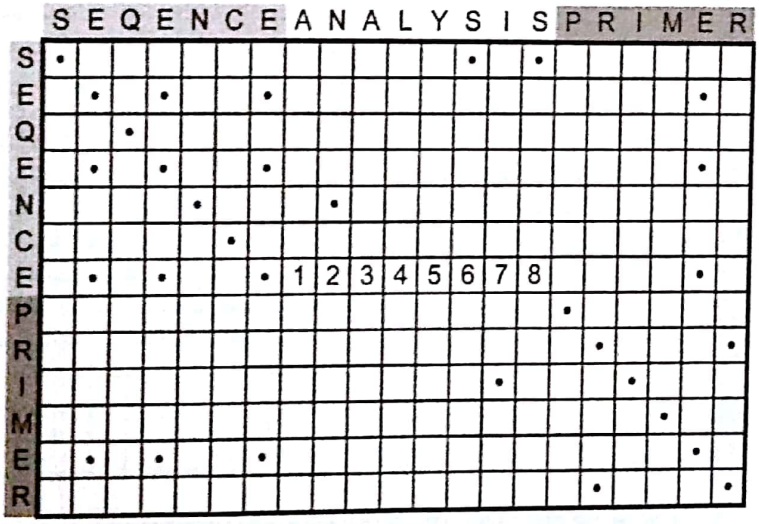


Figure 3.3 – Cas d'une insertion de huit acides aminés (numéros 1 à 8) dans la séquence horizontale. La séquence « ANALYSIS » a été insérée dans la séquence A ou perdue dans la séquence B.

Afin d'augmenter le rapport signal/bruit, on peut considérer qu'il faut un nombre suffisant de points dans un segment pour être indiqué. Ainsi, si dans la matrice de la figure 3.5, on met un point si un segment de longueur 3 contient deux identités, on obtient la matrice de la figure 3.6. Cette notion correspond au filtrage classiquement utilisé en analyse d'images. En bioinformatique, il s'agit de définir une fenêtre de calcul (le plus souvent glissante et chevauchante), ici de longueur égale à 3 sur

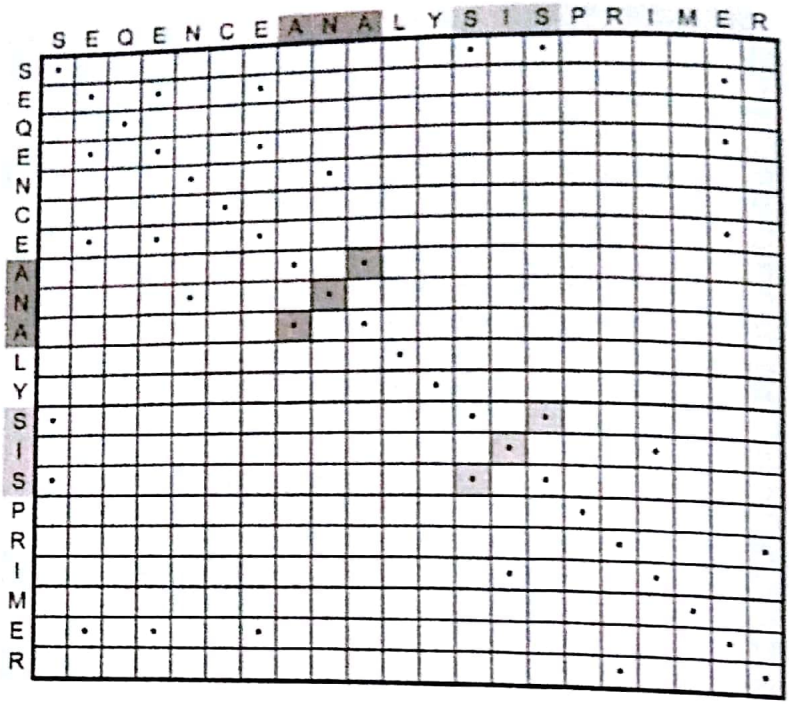


Figure 3.4 – Cas de palindromes (ANA et SIS).

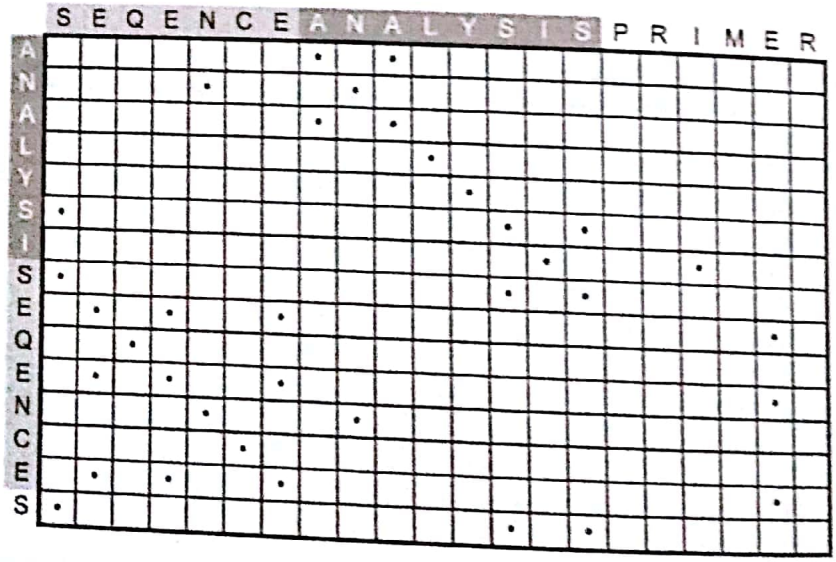


Figure 3.5 – Cas de transposition (exemple d'inversions des segments « SEQENCE » et « ANALYSIS » dans la séquence).

laquelle un score seuil est calculé (ici égal à 2). Cette notion de fenêtre est omniprésente en bioinformatique.

Cette technique de filtrage peut être ajustée en fonction des séquences. Ainsi pour des séquences nucléiques, il n'est pas rare de mettre un point par segment qui contient cinq identités dans un segment de sept bases. Ces matrices de points peuvent être utilisées pour comparer des génomes entiers afin de voir les zones codantes qui sont plus conservées que les régions non codantes.

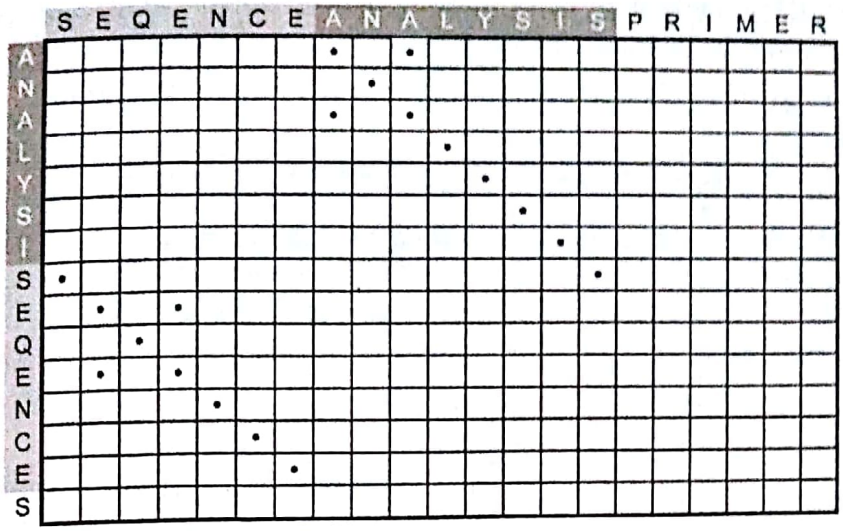


Figure 3.6 – Application d'un filtre sur la matrice n° 5 (deux identités sur trois résidus).

- *Exemples de matrices de points réelles obtenues avec des séquences issues des banques*

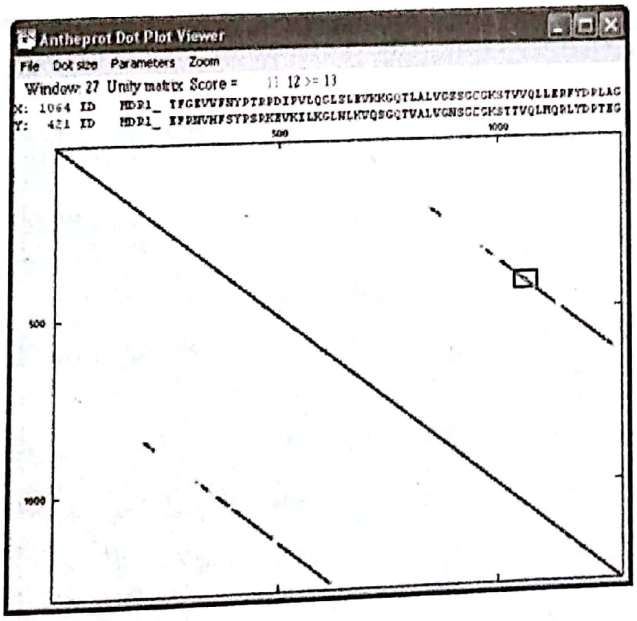


Figure 3.7 – Cas d'une protéine de résistance multiple aux drogues MDR_HUMAN qui est le résultat d'une duplication en tandem de gène (voir vidéo dot_plot.avi).

Vidéo dot_plot.avi. La vidéo montre comment obtenir le tracé d'une matrice de point avec le logiciel AnTheProt. La protéine MDR1_HUMAN.seq (Protéine de MultiDrug Resistance 1 humaine) comprenant 1 280 acides aminés est utilisée sur les deux axes du tracé. Le tracé est effectué avec une longueur de fenêtre de 25 et un seuil d'identité de 5. Ainsi chaque point à l'intersection des deux séquences indique deux segments de 25 acides aminés dans lesquels 5 identités sont présentes. On peut ainsi voir la diagonale complète et la symétrie par rapport à celle-ci du tracé. Le

segment parallèle à cette diagonale indique une répétition interne de longueur 580 (environ la moitié de la longueur de la séquence complète), ce qui est le résultat d'une duplication en tandem du gène codant pour la protéine initiale.

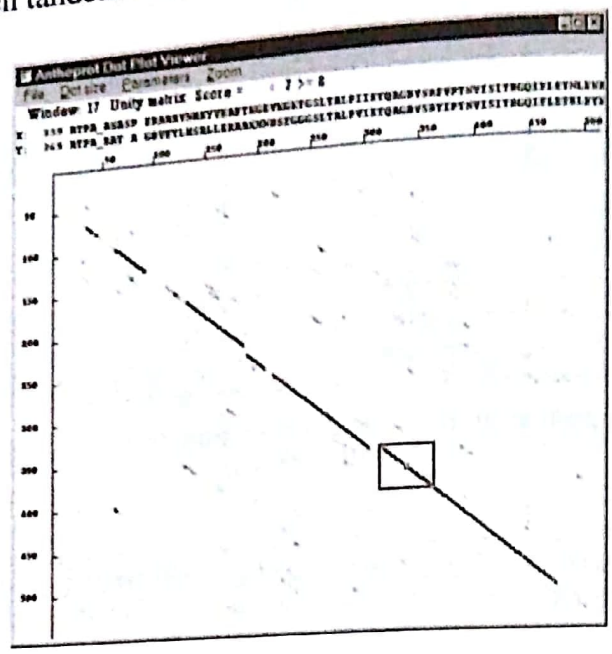


Figure 3.8 – Cas de protéines homologues d'ATP synthase mitochondriale de rat (ATPA_RAT) en Y et de cyanobactérie (ATPA_ANASP) en X.

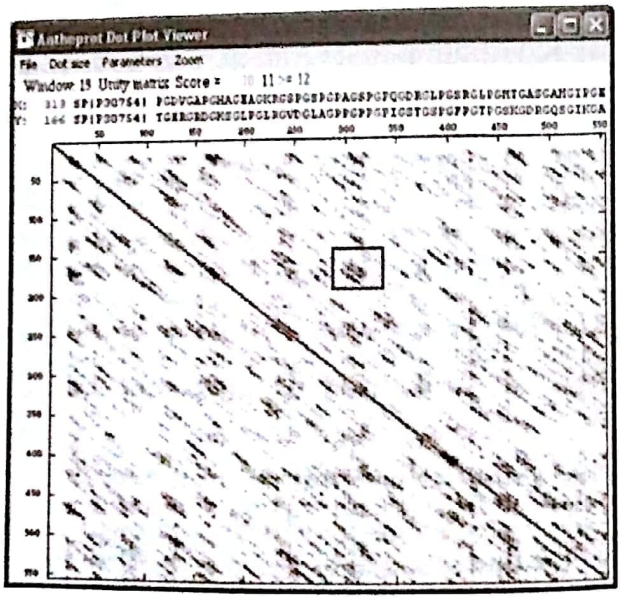


Figure 3.9 – Cas de faible complexité dans du collagène formant des fibrilles CAFF_RIFPA (X) vs CAFF_RIFPA (Y).

3.2 MATRICE DE SUBSTITUTION

Dans toutes les comparaisons précédentes, ne sont visibles que les résidus identiques (matrices 1-5) ou les segments identiques (matrice 6). La plupart du temps, le biologiste

souhaite comparer avec non pas des identités mais avec des similitudes. Ce concept amène la notion de **matrice de substitution**.

Le besoin de comparer des séquences différentes nécessite que l'on puisse quantifier la ressemblance entre des peptides ne présentant pas d'identités. En effet, les trois peptides suivants ne présentent pas d'identité entre eux. Pourtant n'importe quel biochimiste ou phylogénéticien est capable de dire que les peptides 1 et 2 se ressemblent plus entre eux que les peptides 1 et 3. En effet, pour chaque position des peptides 1 et 2, les propriétés physico-chimiques (voir figure 1.3, diagramme de Venn des acides aminés) correspondent mieux pour les deux premiers peptides que pour les peptides 1 et 3.

Comparaison de peptides différents

Peptide 1	A E I G L M A E I G L S E K I L
Peptide 2	L D V A A I G D L A I T Q R L M
Peptide 3	W R G I Y S H H D E T W D C P C

La quantification de la ressemblance entre les peptides implique que l'on soit capable de donner un score au remplacement d'un acide aminé par un autre. Cette quantification est à l'origine des tables de substitution (20 × 20) qui affecte une valeur à chaque paire (i,j). La première table ou **matrice de substitution** est due au travail pionnier de Margaret Dayhoff qui est basé sur le fait que les mutations accumulées au cours de l'évolution n'ont pas altéré la fonction des protéines. On parle alors de **mutations ponctuelles acceptées** (ou PAM). Le postulat est que les protéines évoluent *via* des successions de mutations ponctuelles indépendantes les unes des autres et acceptées dans la population. Pour faire son étude, M. Dayhoff a aligné manuellement un grand jeu de séquences très préservées par l'évolution : les cytochromes c. Ces séquences ont près de 85 % de conservation des acides aminés dans l'alignement.

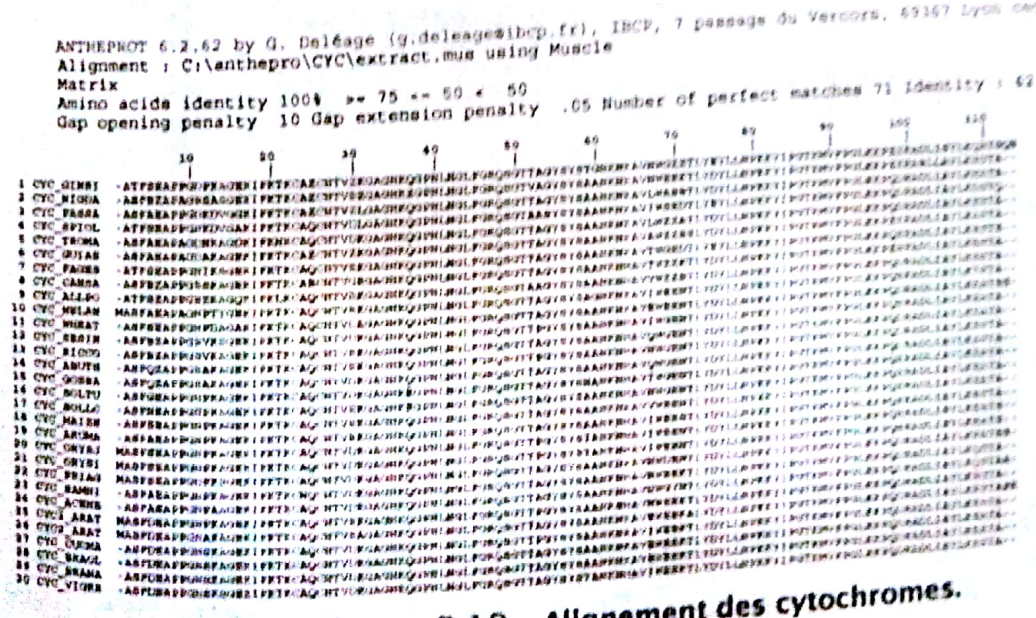


Figure 3.10 - Alignement des cytochromes.

À partir de l'alignement suivant, on peut calculer les changements A_{ij} et la mutabilité m_j.

```
I   D   N   F   K   N
I   D   D   W   K   N
```

	I	D	N	F	W	K
Changements	0	1	1	1	1	0
Occurences	2	3	3	1	1	2
Mutabilité	0	1/3	1/3	1	1	0

À partir des mutabilités calculées, on peut exprimer :

$$M_{ij}^{1} = m_j \frac{A_{ij}}{\sum_{i=1}^{20} A_{ij}} \quad (1)$$

qui peut être normalisée par les probabilités d'occurrences p_i :

$$R_{ij}^{1} = \frac{M_{ij}^{1}}{p_i} \quad (2)$$

On peut ensuite calculer la matrice log PAM 1 :

$$S_{ij}^{1} = \log R_{ij}^{1} \quad (3)$$

On peut déduire des matrices PAM k pour des distances évolutives = k PAM.

$$R_{ij}^{k} = (R_{ij})^{k} \quad (4)$$

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X	*
A	2	-2	0	0	-2	0	0	1	-1	-1	-2	-1	-1	-3	1	1	1	-6	-3	0	0	0	0	-8
R	-2	6	0	-1	-4	1	-1	-3	2	-2	-3	3	0	-4	0	0	-1	2	-4	-2	-1	0	-1	-8
N	0	0	2	2	-4	1	1	0	2	-2	-3	1	-2	-3	0	1	0	-4	-2	-2	2	1	0	-8
D	0	-1	2	4	-5	2	3	1	1	-2	-4	0	-3	-6	-1	0	0	-7	-4	-2	3	3	-1	-8
C	-2	-4	-4	-5	12	-5	-5	-3	-3	-2	-6	-5	-5	-4	-3	0	-2	-8	0	-2	-4	-5	-3	-8
Q	0	1	1	2	-5	4	2	-1	3	-2	-2	1	-1	-5	0	-1	-1	-5	-4	-2	1	3	-1	-8
E	0	-1	1	3	-5	2	4	0	1	-2	-3	0	-2	-5	-1	0	0	-7	-4	-2	3	3	-1	-8
G	1	-3	0	1	-3	-1	0	5	-2	-3	-4	-2	-3	-5	0	1	0	-7	-5	-1	0	0	-1	-8
H	-1	2	2	1	-3	3	1	-2	6	-2	-2	0	-2	-2	0	-1	-1	-3	0	-2	1	2	-1	-8
I	-1	-2	-2	-2	-2	-2	-2	-3	-2	5	2	-2	2	1	-2	-1	0	-5	-1	4	-2	-2	-1	-8
L	-2	-3	-3	-4	-6	-2	-3	-4	-2	2	6	-3	4	2	-3	-3	-2	-2	-1	2	-3	-3	-1	-8
K	-1	3	1	0	-5	1	0	-2	0	-2	-3	5	0	-5	-1	0	0	-3	-4	-2	1	0	-1	-8
M	-1	0	-2	-3	-5	-1	-2	-3	-2	2	4	0	6	0	-2	-2	-1	-4	-2	2	-2	-2	-1	-8
F	-3	-4	-3	-6	-4	-5	-5	-5	-2	1	2	-5	0	9	-5	-3	-3	0	7	-1	-4	-5	-2	-8
P	1	0	0	-1	-3	0	-1	0	0	-2	-3	-1	-2	-5	6	1	0	-6	-5	-1	-1	0	-1	-8
S	1	0	1	0	0	-1	0	1	-1	-1	-3	0	-2	-3	1	2	1	-2	-3	-1	0	0	0	-8
T	1	-1	0	0	-2	-1	0	0	-1	0	-2	0	-1	-3	0	1	3	-5	-3	0	0	-1	0	-8
W	-6	2	-4	-7	-8	-5	-7	-7	-3	-5	-2	-3	-4	0	-6	-2	-5	17	0	-6	-5	-6	-4	-8
Y	-3	-4	-2	-4	0	-4	-4	-5	0	-1	-1	-4	-2	7	-5	-3	-3	0	10	-2	-3	-4	-2	-8
V	0	-2	-2	-2	-2	-2	-2	-1	-2	4	2	-2	2	-1	-1	-1	0	-6	-2	4	-2	-2	-1	-8
B	0	-1	2	3	-4	1	3	0	1	-2	-3	1	-2	-4	-1	0	0	-5	-3	-2	3	2	-1	-8
Z	0	0	1	3	-5	3	3	0	2	-2	-3	1	-2	-4	-1	0	0	-5	-3	-2	3	2	-1	-8
X	0	-1	0	-1	-3	-1	-1	-1	-1	-1	-2	0	-2	-5	0	0	-1	-6	-4	-2	2	3	-1	-8
*	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	-8	1

Figure 3.11 – Matrice PAM 250 de M. Dayhoff calculée avec le logiciel 'pam' (http://www.bioinformatics.nl/cgi-bin/pam.csh).

Plus les valeurs sont élevées plus la substitution est observée et donc plus la substitution sera favorable. Le score de comparaison des peptides est la somme des substitutions individuelles lues dans la matrice PAM250. Ainsi les scores de transformation du peptide 1 dans les peptides 2 et 3 sont respectivement de 27 et de –32.

Peptide 1	A	E	I	G	L	M	A	E	I	G	L	S	E	K	I	L	
	-2	3	4	1	-2	2	1	3	2	1	2	1	2	3	2	4	27
Peptide 2	L	D	V	A	A	I	G	D	L	A	I	T	Q	R	L	M	
Peptide 3	W	R	G	I	Y	S	H	H	D	E	T	W	D	C	P	C	
	-6	-1	-3	-3	-1	-2	-1	1	-2	0	-2	-2	3	-5	-2	-6	-32
Peptide 1	A	E	I	G	L	M	A	E	I	G	L	S	E	K	I	L	

D'autres matrices existent comme la matrice unitaire (qui correspond à PAM0). Cette matrice consiste à ne considérer que les identités dans les comparaisons. Ce type de matrice peut être utilisé pour rechercher des signatures (voir chapitre 8, paragraphe 8.2).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V	B	Z	X
A	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
R	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
N	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
D	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
C	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Q	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
E	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
G	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
H	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
I	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1
L	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
K	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1
M	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1
F	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1
S	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1
W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1
V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
B	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Z	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
X	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 3.12 – Matrice unitaire pour les protéines (PAM0).

La matrice de structure secondaire a été déterminée empiriquement de manière à fournir la meilleure qualité de prédiction de structure secondaire sur un jeu de référence (voir méthode des plus proches voisins au chapitre 10).

	A	R	N	D	C	Q	E	G	H	I	L	K	M	F	P	S	T	W	Y	V
A	2	0	0	0	0	0	1	0	0	0	0	0	0	0	-1	1	0	-1	-1	0
R	0	2	0	0	0	0	0	0	0	-1	-1	1	-1	-1	0	0	0	-1	-1	-1
N	0	0	3	1	0	1	0	0	0	-1	-1	1	-1	-1	0	0	0	-1	-1	-1
D	0	0	1	2	0	0	1	0	0	-1	-1	0	-1	-1	0	0	0	-1	-1	-1
C	0	0	0	0	2	0	0	0	0	0	0	0	0	-1	0	0	0	-1	-1	0
Q	0	0	1	0	0	2	1	0	0	-1	-1	0	-1	-1	0	0	0	-1	-1	-1
E	1	0	0	1	0	1	2	0	0	-1	-1	0	-1	-1	-1	0	0	-1	-1	-1
G	0	0	0	0	0	0	0	2	0	-1	-1	0	-1	-1	1	0	0	-1	-1	-1
H	0	0	0	0	0	0	0	0	2	-1	-1	0	-1	-1	0	0	0	0	-1	-1
I	0	-1	-1	-1	0	-1	-1	-1	-1	2	0	-1	0	1	-1	-1	0	0	0	1
L	0	-1	-1	-1	0	-1	-1	-1	-1	0	2	-1	2	0	-1	-1	0	0	0	1
K	0	1	1	0	0	0	0	0	0	-1	-1	2	-1	-1	0	0	0	0	-1	-1
M	0	-1	-1	-1	0	-1	-1	-1	-1	0	2	-1	2	0	-1	-1	0	0	0	0
F	0	-1	-1	-1	-1	-1	-1	-1	-1	1	0	-1	0	2	-1	-1	0	0	1	0
P	-1	0	0	0	0	0	-1	1	0	-1	-1	0	-1	-1	3	0	0	-1	-1	-1
S	1	0	0	0	0	0	0	0	0	-1	-1	0	-1	-1	0	2	0	-1	-1	-1
T	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	-1	-1	0
W	-1	-1	-1	-1	-1	-1	-1	-1	0	0	0	0	0	0	-1	-1	-1	2	0	0
Y	-1	-1	-1	-1	-1	-1	-1	-1	-1	0	0	-1	0	1	-1	-1	-1	0	2	0
V	0	-1	-1	-1	0	-1	-1	-1	-1	1	1	-1	0	0	-1	-1	0	0	0	2

Figure 3.13 – **Matrice pour la prédiction de structure secondaire.**

Enfin, une matrice dite BLOSUM (*BLOck of amino acids SUbstitution Matrix*) est basée sur un découpage de zones conservées au-dessus d'un seuil d'identité d'alignements réels. Il n'y a donc pas d'extrapolation comme pour les matrices PAM. Ainsi, la matrice BLOSUM 62 est une des plus utilisées dans les programmes de comparaison de séquences.

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

Figure 3.14 – **Matrice BLOSUM 62.**

Les matrices sont disponibles sur le serveur ftp du NCBI (ftp://ftp.ncbi.nih.gov/blast/matrices/). Le choix de la matrice à utiliser dépend du taux de conservation des séquences. En fonction du jeu de séquences, l'utilisateur souhaite choisir la matrice la plus pertinente. Pour des matrices (comme par exemple PAM et BLOSUM) qui sont établies de manière différente, une équivalence est donc nécessaire.

La figure 3.15 montre quelles matrices utiliser en fonction de la plage d'identité couverte par les protéines. On peut remarquer que les nombres associés aux matrices PAM et BLOSUM évoluent en sens contraire. Les programmes les plus récents sont capables de déterminer automatiquement la matrice optimale à partir du jeu de séquences à comparer.

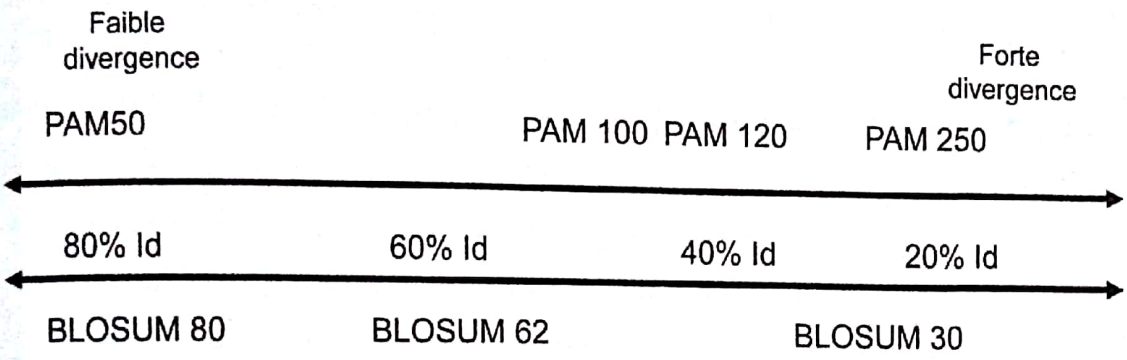


Figure 3.15 – Gamme d'utilisation des matrices PAM et BLOSUM.

Les matrices de substitution peuvent être utilisées pour construire les matrices de points afin de comparer des séquences au niveau de leur similitude. Dans ce cas, il suffit de calculer les scores sur des segments de longueurs impaires et de tracer un point à l'intersection des deux résidus au centre de chaque segment si et seulement si le score de substitution est supérieur à une valeur seuil. Cette méthode permet d'identifier des segments similaires entre deux protéines sans que des identités soient présentes. Il faut choisir le score seuil adapté à chaque matrice de substitution.

RECHERCHE DANS LES BANQUES

4

PLAN

OBJECTIFS

- ➤ Comprendre les scores de similitude entre séquences
- ➤ Choisir une méthode de recherche locale ou globale
- ➤ Connaître le principe de FASTA et ses heuristiques
- ➤ Connaître le principe de BLAST et ses limitations

La recherche dans les banques peut être considérée comme un mode d'interrogation de séquences alternatif au mode par annotation (traité au chapitre 2).

4.1 SCORE DE SIMILITUDE ENTRE SÉQUENCES

Afin de pouvoir comparer des alignements différents de séquences ou de comparer des scores entre séquences à l'issue d'une recherche de ressemblance dans une banque, il est important de savoir comment calculer un score de similitude et surtout de savoir si un score obtenu est significatif. Le pourcentage d'identité (%Id) est le nombre d'identité entre deux séquences après alignement des séquences rapporté à la longueur de l'alignement. Ainsi pour deux séquences données, le pourcentage d'identité dépend de l'alignement.

Exemples

```
S1    DDLSKQAVAYRQMSLLLRG
S2    DGKTAVATDTILLLQG
      *     *
Id%=(2/19)×100 = 11 %

S1    DDLSKQAVAYRQMSLLL-RG
S2    -D-GKTAVA-TDTILLLQG-
       *  * ***     ***
Id%=(8/20) × 100 = 40 %
```

Dans l'alignement 2, il s'agit d'un vrai alignement (avec possibilité d'insertions et de délétions appelées aussi *gaps*). Le pourcentage de similitude Si% est la même notion que l'identité mais en utilisant une matrice de substitution. Il ressort que le Si% ≥ Id%.

Comment savoir si le pourcentage obtenu est fort et significatif ou pas ? Pour répondre, il faut :

- définir la façon dont on calcule le score (pénalité sur les insertions-délétions par exemple) ;
- faire un choix sur l'alignement (ici nous considérons le deuxième alignement comme celui de référence, son score est Sr = 40) ;
- prendre en compte la longueur des séquences ;
- savoir si le score mesuré est significatif.

Une difficulté est que le score Sr obtenu n'est comparable que pour un système de score donné. Il est possible de normaliser le score Sr en S_{bit} par :

$$S_{bit} = \frac{\lambda Sr - \ln K}{\ln 2}$$

où λ et K sont des paramètres estimés en fonction des matrices/pénalités et composition en acides aminés des séquences pour un espace de recherche donné. On peut considérer Sbit comme un score normalisé qui permet de comparer des scores obtenus sur des séquences différentes.

Pour savoir si un score Sr (ou S_{bit}) est significatif, il suffit de brasser les séquences (*shuffling*) comme dans la figure 4.1. Dans cet exemple, seule la séquence 1 est générée aléatoirement tout en respectant la composition en acides aminés. Par simplification, le score est ici simplement le nombre d'identités. Après avoir refait les alignements, les scores sont calculés.

Le pourcentage maximal d'identité reflète la différence de composition en acides aminés. Pour s'en convaincre, il suffit de ranger ensemble toutes les lettres identiques. Il ressort que lors d'un tirage, le score obtenu par hasard peut être supérieur à celui obtenu pour les vraies séquences.

La moyenne des scores obtenus après brassage représente le bruit moyen Sm pour ces séquences.

La position des acides aminés dans la séquence 1 est tirée au sort. Ici quatre tirages successifs sont effectués. Les séquences aléatoires respectent la composition en acides aminés des séquences initiales.

On peut définir alors le Z-score :

$$Z = \frac{Sr - Sm}{\sigma}$$

(σ est l'écart-type de la distribution des scores).

Ce Z-score permet de comparer des paires de séquences de longueur et de compositions différentes.

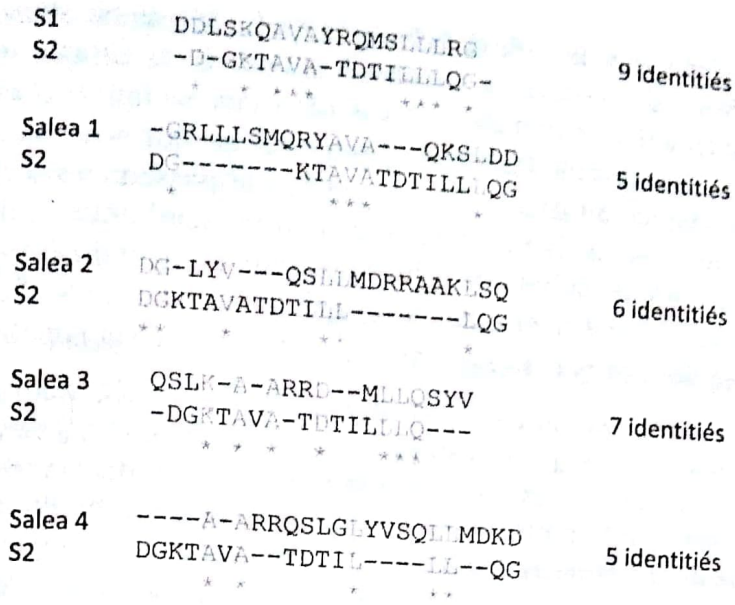


Figure 4.1 – Brassage des séquences et alignements obtenus.
Pour chaque tirage Salea, l'ordre des acides aminés de la séquence 1 est tiré au sort. Cela garantit de ne pas avoir de biais car la composition initiale en acides aminés est conservée.

Une question importante pour juger du score est : « Combien de fois le score Sr est attendu par hasard ? »

Pour deux séquences de longueurs n et m, on peut calculer la E-value notée E() comme le nombre de fois que deux séquences auront par chance un score supérieur ou égal à Sr (ou un score normalisé S_{bit}) :

$$E() = K.m.n.e^{-\lambda Sr}$$

où K et λ sont des constantes.

En utilisant le S_{bit} la relation devient :

$$E() = m.n.2^{-Sbit}$$

Avec l'exemple pris précédemment le Z score devient :

$$Z = (9 - 5{,}75)/0{,}95 = 3{,}4.$$

À partir de la valeur de E(), on peut calculer la probabilité p d'obtenir un score Sr en faisant une hypothèse sur la loi de distribution des scores. En prenant comme hypothèse une distribution des scores selon une loi de Poisson on obtient :

$$p\text{-value} = 1 - e^{-E()}$$

On peut remarquer que la longueur des séquences influe sur l'E-value. Il en est de même lorsqu'une séquence est comparée à une banque de données. Dans ce cas, on peut considérer la banque comme une séquence virtuelle de taille égale au nombre de résidus de la banque. Plus la banque est de taille importante, plus la E() value a tendance à augmenter. Dans le même temps, la probabilité d'avoir des séquences

homologues augmente avec la taille de la banque. Or les séquences homologues auront des E-value faibles. La E() est donc sensible à la taille de la banque selon deux sens opposés. Compte tenu de ces données, plus la E() value est faible, plus les scores ont des chances d'être significatifs. Par ailleurs, pour une banque de taille infinie, on trouvera toujours par chance au moins une séquence qui possède un score égal à celui obtenu. Il ressort de cela que la E() considérée comme seuil limite (minimal et maximal) dépend de la taille de la banque (plus la banque est de taille importante, plus la E() seuil de recherche doit être grande), alors que le score Sr (ou S_{bit}) est indépendant de la banque et ne dépend que du système de score (matrice et pénalités de gaps).

Dans le cas de faible complexité (voir chapitre 3), le Z-score peut être artificiellement élevé, faisant croire à tort à des séquences similaires. **La faible complexité biaise les statistiques** et est le « cauchemar » du bioinformaticien. C'est pourquoi elle doit être systématiquement recherchée (ou automatiquement éliminée par le choix de paramètres appropriés).

4.2 RECHERCHE GLOBALE OU LOCALE

La comparaison d'une séquence avec une banque de séquences est le traitement informatique le plus commun effectué par les biologistes. En général, l'objectif est d'identifier l'ensemble des séquences de protéines homologues ou d'identifier des segments de séquences communs à différentes protéines.

L'utilisateur doit donc choisir entre un algorithme local ou global car les résultats obtenus seront différents. En effet, beaucoup de protéines ont des organisations en **modules** (souvent improprement nommés **domaines**). Cette organisation influe sur le choix des algorithmes à utiliser.

Soit deux protéines qui ont les organisations schématisées figure 4.2.

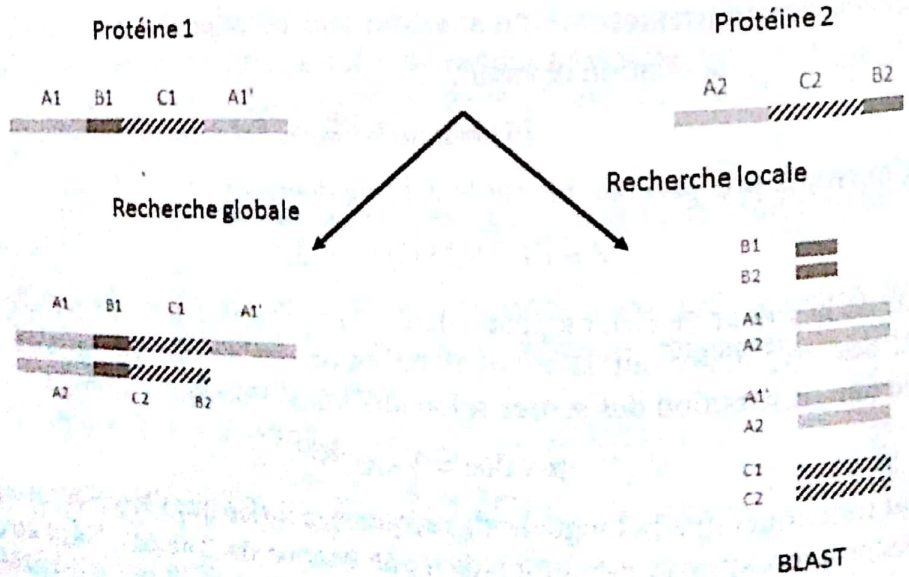


Figure 4.2 – Méthode globale ou méthode locale.
Les segments indiqués par des lettres identiques correspondent à des modules ayant des séquences proches.

Dans un algorithme global, ce qui est recherché c'est l'ensemble des séquences avec un score significatif sur une longueur proche de la longueur des deux séquences. Cela correspond typiquement à la recherche d'homologues. Dans l'algorithme local, ce qui est recherché ce sont des zones de similitudes dans des protéines quelconques (homologues ou pas).

4.3 FASTA

Dans FASTA, les *n* matrices de points entre la séquence d'intérêt (query) et chacune des séquences (Subject) de la banque qui contient *n* séquences sont calculées dans la mémoire de la machine. Une première **heuristique** est utilisée : deux séquences de protéines homologues présentent des régions identiques. Il s'agit d'une heuristique dans la mesure où deux séquences homologues peuvent partager jusqu'à 50 % d'identité sans segment de longueur > 1 identique (cas de l'alternance d'un acide aminé identique et d'un acide aminé différent). Cette heuristique est d'autant plus « risquée » que la longueur des segments identiques (mot de longueur k ou k-tuple) est grande. Mais un k-tuple élevé accélère la vitesse des calculs. Ainsi, pour les protéines, on utilise un k-tuple compris entre 1 et 4 avec, par défaut, la valeur 2 (et 7 à 11 pour les séquences nucléiques). Une deuxième heuristique est que ces k-tuples identiques doivent être proches de la diagonale d'une matrice de points et que pour des raisons d'économie de calculs, seules les identités dans une bande d'une distance *d* de la diagonale seront considérées. Les quatre étapes de FASTA sont résumées dans les quatre figures suivantes (figures 4.3 à 4.6).

1. Recherche des k-tuples identiques (calcul du score init1).

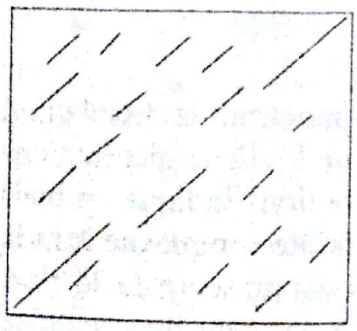


Figure 4.3 – Un score init1 élevé indique un long segment identique.

2. Chaînage des k-tuples qui sont sur la même diagonale (calcul du score initn). Un score initn élevé reflète la capacité à étendre sur une même diagonale des segments identiques.

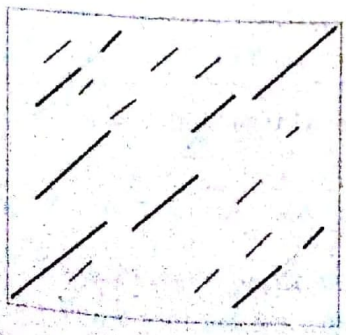


Figure 4.4 – Les diagonales représentées en gras ont pu être étendues.

3. Sélection d'une bande (symbolisée par les pointillés) de largeur *d* autour de la diagonale.

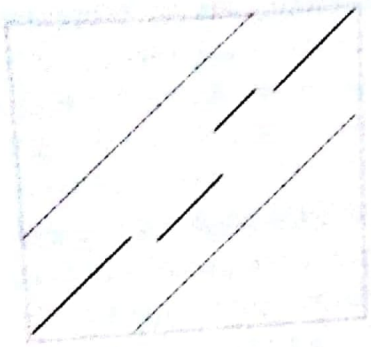


Figure 4.5 – Filtrage par une matrice de substitution et sélection des plus fortes densités en k-tuples proches d'une distance *d* de la diagonale.

4. Alignement des *x* meilleurs scores et génération de la statistique associée (score Opt et E-value).

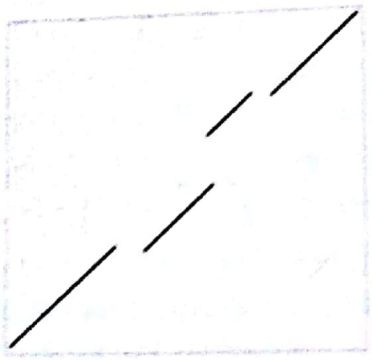


Figure 4.6 – Alignement optimal entre les deux séquences (voir chapitre 5).

La première partie du fichier généré est un histogramme montrant la distribution des scores **Opt** et de **E() value**. Le biologiste doit vérifier que les deux distributions sont globalement superposées. Un décalage dans les distributions indique un biais dans la représentation des séquences dans la banque ou une faible complexité dans la séquence soumise. La ligne 9 indique que 16 899 séquences ont un score de 36 alors que 15 792 sont attendues par chance avec un tel score. La dernière ligne indique que 2 431 séquences ont un score supérieur à 120 alors que 10 séquences sont attendues. Cela indique que ces 2 431 séquences ont des scores très probablement significativement différents de ceux obtenus par hasard.

Fichier de résultats fourni par FASTA

```
FASTA searches a protein or DNA sequence data bank version 3.4t24 April 23, 2004
Please cite:
 W.R. Pearson & D.J. Lipman PNAS (1988) 85:2444-2448

UNK_172120, 507 aa
vs /db/UniProt/sp.fas library
```

```
      opt      E()
< 20   612      0:=
  22     1      0:=
  24     1      0:=          one = represents 888 library sequences
  26     8     11:*
  28    46    121:*
  30   385    733:*
  32  1992   2835:===*
  34  7098   7689:========*
  36 16899  15792:================*==
  38 29625  26099:==========================*
  40 41476  36405:=====================================*====
  42 49791  44501:=============================================*=====
  44 53263  49089:================================================*======
  46 51122  49998:================================================*======
  48 47190  47867:============================================*=====
  50 42041  43679:=========================================*=
  52 35226  38401:=================================*
  54 29323  32801:============================ *
  56 24303  27399:========================  *
  58 20574  22494:====================  *
  60 16150  18222:================ *
  62 13506  14608:=============*
  64 10863  11618:==========*
  66  8341   9182:========*
  68  6861   7223:======*
  70  5304   5660:=====*
  72  4116   4423:====*
  74  3447   3448:===*
  76  2651   2684:===*
  78  2293   2086:==*
  80  1770   1620:=*
  82  1347   1239:=*
  84  1045    982:=*
  86   764    760:*
  88   627    588:*          inset = represents 49 library sequences
  90   464    455:*
  92   384    352:*         :=======*
  94   312    272:*         :======*=
  96   259    211:*         :=====*=
  98   150    163:*         :===*
 100   109    126:*         :==*
 102   124     98:*         :=*=
 104   118     75:*         :=*=
 106    84     58:*         :=*
 108    52     45:*         :*=
 110    46     35:*         :*
 112    36     27:*         :*
 114    34     21:*         :*
 116    22     16:*         :*
 118     9     13:*         :*
>120  2431     10:*==       :*=======================================
189667530 residues in 534695 sequences
 statistics sampled from 60000 to 532269 sequences
  Expectation_n fit: rho(ln(x))= 5.1283+/-0.000199; mu= 12.7887+/- 0.011
 mean_var=87.3052+/-18.250, 0's: 70 Z-trim: 344  B-trim: 3679 in 1/66
 Lambda= 0.137263
 Kolmogorov-Smirnov  statistic: 0.0345 (N=29) at  46
```

La deuxième partie est un classement des occurrences (*hits*) par ordre croissant de E() Value (seules les cinq premières sont montrées).

```
ASTA (3.47 Mar 2004) function [optimized, BL50 matrix (15:-5)] ktup: 2
 join: 37, opt: 1, gap-pen: -12/-2, width:  16
The best scores are:                                      opt bits E(534695)
gnl|sp|P00823 (ATPA_TOBAC) ATP synthase subunit a  ( 507) 3137 631.2 1.4e-179
gnl|sp|Q2MIB5 (ATPA_SOLLC) ATP synthase subunit a  ( 507) 3120 627.8 1.4e-178
gnl|sp|Q27S65 (ATPA_SOLTU) ATP synthase subunit a  ( 507) 3120 627.8 1.4e-178
gnl|sp|Q2MIK2 (ATPA_SOLBU) ATP synthase subunit a  ( 507) 3120 627.8 1.4e-178
gnl|sp|Q3C1H4 (ATPA_NICSY) ATP synthase subunit a  ( 507) 3119 627.6 1.6e-178
-
```

Dans cette partie, les 954 premières protéines sont des orthologues (voir chapitre 6) de la sous-unité α de l'ATP synthase et présentent une E() value < 9,1.10^{-28}. Les 1 356 suivantes (0,0052 > E-value > 2,8 10^{-25}) sont des ATP synthases ou des ATPases de flagelles ou de vacuoles. Enfin les 86 dernières sont listées ci-après.

```
gnl|sp|P52153 (RHO_DEIRA) Transcription terminati  ( 426)  173  44.1  0.006
gnl|sp|Q619C0 (VATB_CAEBR) Probable V-type proton  ( 491)  173  44.2  0.0066
gnl|sp|P31410 (VATB_HELVI) V-type proton ATPase s  ( 494)  172  44.0  0.0076
gnl|sp|P31401 (VATB_MANSE) V-type proton ATPase s  ( 494)  172  44.0  0.0076
gnl|sp|A6L8P2 (ATPB_PARD8) ATP synthase subunit b  ( 505)  172  44.0  0.0078
gnl|sp|Q38680 (VATB2_ACEAT) V-type proton ATPase   ( 492)  171  43.8  0.0087
gnl|sp|Q5R5V5 (VATB2_PONAB) V-type proton ATPase   ( 511)  168  43.2  0.014
gnl|sp|P62814 (VATB2_MOUSE) V-type proton ATPase   ( 511)  168  43.2  0.014
gnl|sp|P21281 (VATB2_HUMAN) V-type proton ATPase   ( 511)  168  43.2  0.014
gnl|sp|P31408 (VATB2_BOVIN) V-type proton ATPase   ( 511)  168  43.2  0.014
gnl|sp|P62815 (VATB2_RAT) V-type proton ATPase su  ( 511)  168  43.2  0.014
gnl|sp|Q9HNE4 (VATB_HALSA) V-type ATP synthase be  ( 471)  167  43.0  0.015
gnl|sp|B0R754 (VATB_HALS3) V-type ATP synthase be  ( 471)  167  43.0  0.015
gnl|sp|P15313 (VATB1_HUMAN) V-type proton ATPase   ( 513)  165  42.6  0.02
gnl|sp|Q76NU1 (VATB_DICDI) V-type proton ATPase s  ( 493)  163  42.2  0.026
gnl|sp|P48413 (VATB_CYACA) V-type proton ATPase s  ( 500)  163  42.2  0.026
gnl|sp|P22550 (VATB_CANTR) V-type proton ATPase s  ( 511)  162  42.0  0.031
gnl|sp|P49712 (VATB_CHICK) V-type proton ATPase s  ( 453)  161  41.8  0.032
gnl|sp|P45835 (RHO_MYCLE) Transcription terminati  ( 610)  162  42.1  0.035
gnl|sp|Q25691 (VATB_PLAFA) V-type proton ATPase s  ( 494)  158  41.2  0.052
gnl|sp|P66028 (RHO_MYCTU) Transcription terminati  ( 602)  159  41.5  0.052
gnl|sp|P66029 (RHO_MYCBO) Transcription terminati  ( 602)  159  41.5  0.052
gnl|sp|P31407 (VATB1_BOVIN) V-type proton ATPase   ( 513)  158  41.2  0.054
gnl|sp|O03073 (ATPB_LONHI) ATP synthase subunit b  ( 208)  153  39.9  0.056
gnl|sp|Q11Y90 (ATPB_CYTH3) ATP synthase subunit b  ( 501)  157  41.0  0.06
gnl|sp|Q40079 (VATB2_HORVU) V-type proton ATPase   ( 483)  156  40.8  0.068
gnl|sp|Q9SZN1 (VATB2_ARATH) V-type proton ATPase   ( 487)  156  40.8  0.068
gnl|sp|Q43432 (VATB1_GOSHI) V-type proton ATPase   ( 488)  156  40.8  0.068
gnl|sp|Q40078 (VATB1_HORVU) V-type proton ATPase   ( 488)  156  40.8  0.068
gnl|sp|P31411 (VATB_SCHPO) V-type proton ATPase s  ( 503)  155  40.6  0.08
gnl|sp|D1AWS1 (RHO_STRM9) Transcription terminati  ( 416)  154  40.4  0.08
gnl|sp|O67031 (RHO_AQUAE) Transcription terminati  ( 436)  153  40.2  0.095
gnl|sp|P11574 (VATB1_ARATH) V-type proton ATPase   ( 486)  153  40.2  0.1
gnl|sp|Q8W4E2 (VATB3_ARATH) V-type proton ATPase   ( 487)  153  40.2  0.1
gnl|sp|Q56404 (VATB_THET8) V-type ATP synthase be  ( 478)  152  40.0  0.12
gnl|sp|Q72J73 (VATB_THET2) V-type ATP synthase be  ( 478)  152  40.0  0.12
gnl|sp|P52154 (RHO_MICLU) Transcription terminati  ( 691)  152  40.2  0.15
gnl|sp|Q9HTV1 (RHO_PSEAE) Transcription terminati  ( 419)  149  39.4  0.16
gnl|sp|P38527 (RHO_THEMA) Transcription terminati  ( 427)  149  39.4  0.16
gnl|sp|C7LJY3 (RHO_SULMS) Transcription terminati  ( 379)  147  38.9  0.2
gnl|sp|O03077 (ATPB_OSMCI) ATP synthase subunit b  ( 220)  142  37.7  0.27
gnl|sp|Q7NXP1 (RHO_CHRVO) Transcription terminati  ( 418)  143  38.2  0.36
gnl|sp|Q8Y3A8 (FMT_RALSO) Methionyl-tRNA formyltr  ( 327)  141  37.7  0.4
gnl|sp|P52155 (RHO_PSEFC) Transcription terminati  ( 419)  142  38.0  0.42
gnl|sp|P52156 (RHO_RHOS4) Transcription terminati  ( 422)  142  38.0  0.42
```

```
gnl|sp|P52152 (RHO_ALLVD) Transcription terminati  ( 418)  141  37.8    0.48
gnl|sp|Q61891 (RHO_BUCAP) Transcription terminati  ( 419)  141  37.8    0.48
gnl|sp|Q28129 (RHO_STRRD) Transcription terminati  ( 383)  138  37.2    0.68
gnl|sp|P0CH92 (RHO1_EHRCR) Transcription terminat  ( 422)  138  37.2    0.73
gnl|sp|P0CH93 (RHO2_EHRCR) Transcription terminat  ( 422)  138  37.2    0.73
gnl|sp|P57652 (RHO_BUCAI) Transcription terminati  ( 419)  136  36.8    0.96
gnl|sp|P33661 (RHO_BORBU) Transcription terminati  ( 515)  137  37.1    0.96
gnl|sp|Q9PA21 (RHO_XYLFA) Transcription terminati  ( 411)  135  36.6     1.1
gnl|sp|Q06447 (RHO_NEIGO) Transcription terminati  ( 419)  133  36.2     1.4
gnl|sp|P52157 (RHO_STRLI) Transcription terminati  ( 707)  135  36.8     1.6
gnl|sp|P56466 (RHO_HELPY) Transcription terminati  ( 436)  132  36.0     1.7
gnl|sp|Q9ZLS9 (RHO_HELPJ) Transcription terminati  ( 436)  132  36.0     1.7
gnl|sp|O03070 (ATPB_HYPHQ) ATP synthase subunit b  ( 208)  128  34.9     1.7
gnl|sp|Q1RIJ6 (RHO_RICBR) Transcription terminati  ( 449)  131  35.8       2
gnl|sp|Q68WL0 (RHO_RICTY) Transcription terminati  ( 457)  131  35.8       2
gnl|sp|Q9ZDZ4 (RHO_RICPR) Transcription terminati  ( 457)  131  35.8       2
gnl|sp|Q92HL2 (RHO_RICCN) Transcription terminati  ( 458)  131  35.8       2
gnl|sp|Q4ULF7 (RHO_RICFE) Transcription terminati  ( 468)  131  35.8       2
gnl|sp|P0AG32 (RHO_ECO57) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|P0AG33 (RHO_SHIFL) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|P0AG30 (RHO_ECOLI) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|P0A296 (RHO_SALTI) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|P0AG31 (RHO_ECOL6) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|P0A295 (RHO_SALTY) Transcription terminati  ( 419)  130  35.6     2.2
gnl|sp|Q89A22 (RHO_BUCBP) Transcription terminati  ( 419)  129  35.4     2.5
gnl|sp|P44619 (RHO_HAEIN) Transcription terminati  ( 420)  129  35.4     2.5
gnl|sp|O83281 (RHO_TREPA) Transcription terminati  ( 519)  130  35.7     2.5
gnl|sp|Q66JY6 (CI100_MOUSE) Vav-like protein C9or  ( 344)  127  34.9     2.8
gnl|sp|C1A5H8 (RHO_GEMAT) Transcription terminati  ( 737)  130  35.9     3.2
gnl|sp|Q8RG42 (RHO_FUSNN) Transcription terminati  ( 413)  127  35.0     3.2
gnl|sp|Q2G493 (FMT_NOVAD) Methionyl-tRNA formyltr  ( 301)  124  34.3     3.9
gnl|sp|A6UWG4 (SRP54_META3) Signal recognition 54  ( 460)  126  34.8     3.9
gnl|sp|P39180 (AG43_ECOLI) Antigen 43 OS=Escheric (1039)  130  36.0     4.1
gnl|sp|Q6L1A8 (RL18_PICTO) 50S ribosomal protein   ( 158)  117  32.6     6.5
gnl|sp|A81275 (MUTS_AZOC5) DNA mismatch repair pr  ( 931)  126  35.2     6.6
gnl|sp|A5PKE4 (TEAN2_BOVIN) Transcription elongat  ( 208)  118  32.9     6.9
gnl|sp|O03222 (RHO_BACSU) Transcription terminati  ( 427)  121  33.8     7.6
gnl|sp|Q57565 (SRP54_METJA) Signal recognition 54  ( 451)  121  33.9     7.9
gnl|sp|B7KHD0 (FMT_CYAP7) Methionyl-tRNA formyltr  ( 334)  118  33.1     9.6
gnl|sp|P21212 (YLC7_YEREN) Uncharacterized protei  (  58)  109  30.6     9.6
gnl|sp|Q9Z661 (DCDA_ZYMMO) Diaminopimelate decarb  ( 421)  119  33.4     9.8
```

Il existe une zone floue dans laquelle des séquences intruses s'intercalent avec des séquences de la famille recherchée. C'est le cas des facteurs de terminaisons de la transcription qui ont de meilleurs scores que la séquence ATPB_PARDS. La troisième partie du fichier contient les alignements par paire.

Utiliser un algorithme de recherche du type de FASTA ou BLAST pour rechercher les séquences dans différentes espèces suppose que soit établi le fait que les protéines ont évolué par divergence.

4.4 BLAST

Le programme BLAST (*Basic Local Alignment Search Tool*) est un algorithme de recherche de similitudes locales. La première étape consiste à établir la liste de mots exacts (dans la version initiale) de longueur fixée (W = 3 protéines, W = 11 acides nucléiques).

Figure 4.7 - Liste des L-W+1 mots.

La deuxième étape consiste à établir la liste exhaustive des mots trouvés dans la banque.

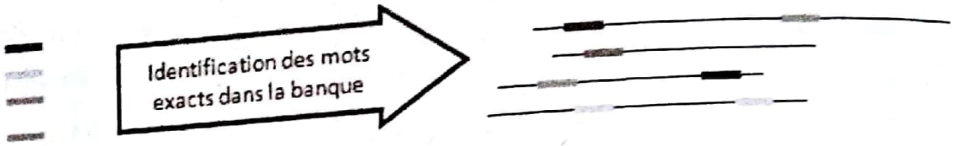


Figure 4.8 - Identification des mots dans les séquences de la banque.

Ensuite, pour chaque mot trouvé, l'algorithme étend progressivement de part et d'autre tant que le score sur le segment est supérieur à une valeur seuil.

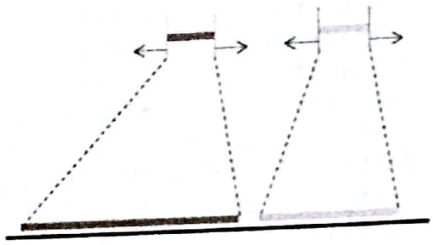


Figure 4.9 - Extension des mots exacts trouvés.

Plusieurs évolutions ont été apportées : tout d'abord des mots proches et non plus seulement exacts ont été utilisés ; pour un mot donné, les mots voisins sont ceux qui ont un score supérieur à une valeur seuil fixée (ici 13).

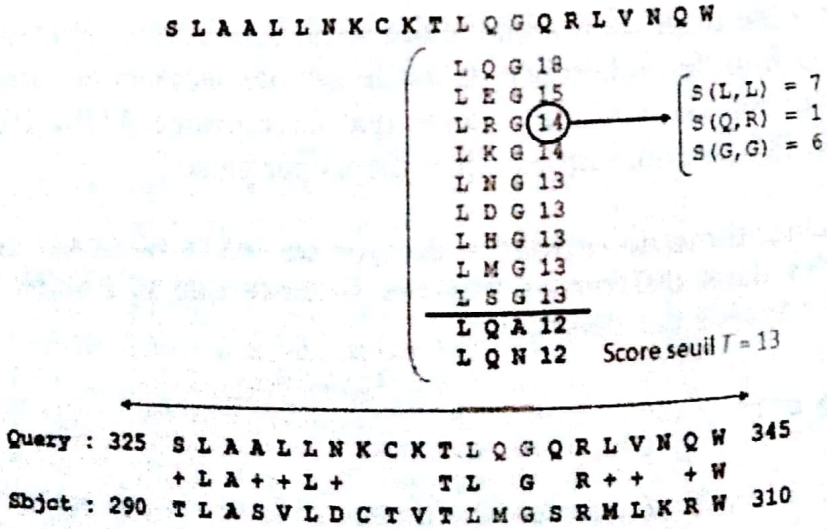


Figure 4.10 - Prise en compte des mots voisins des mots exacts.

Ensuite la possibilité d'inclure des insertions-délétions a été prise en compte dans Gapped-BLAST. Enfin, la version avec profil de BLAST appelée PSI-BLAST (*Position Specific Iterative BLAST*) permet de construire un profil (voir page 106) à partir des plus forts scores issus d'un premier parcours de la banque. Ensuite, ce profil est utilisé pour rechercher des séquences distantes de manière itérative jusqu'à ce que le jeu de séquence soit inchangé. Cependant, il existe un risque que l'algorithme dérive en incluant des séquences intruses et fasse dévier le profil.

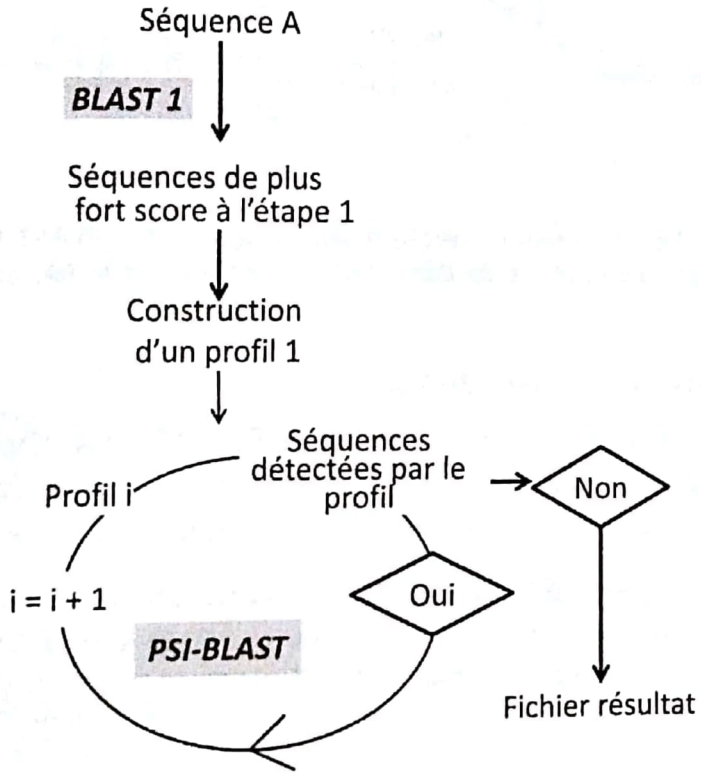


Figure 4.11 – Organigramme de PSI-BLAST.

À noter qu'il existe aussi d'autres variantes du programme BLAST comme PHI-BLAST, qui utilise la syntaxe de PROSITE comme critère additionel de détection, ou RPS-BLAST, DELTA-BLAST (*Domain Enhanced Lookup Time Accelerated BLAST*), efficaces pour identifier des homologues lointains car ces programmes utilisent une banque préconstruite de profils position spécifiques (PSSM).

La figure 4.12 indique les différentes versions de programmes à utiliser (à noter que les équivalents existent pour FASTA) en fonction des séquences et des banques.

Dans toute recherche dans les banques, la distribution des séquences de la famille chevauche partiellement la distribution des intrus, le biologiste doit donc faire un compromis entre récupérer le maximum de séquence de sa famille au risque d'avoir beaucoup intrus (bonne **sensibilité**) ou bien de récupérer le minimum d'intrus (bonne **spécificité**) au risque de ne pas retenir certaines protéines homologues de la famille.

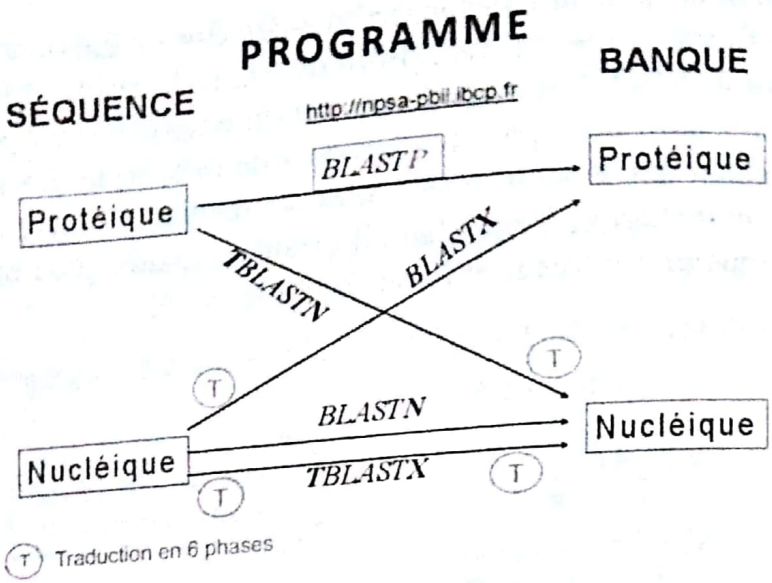


Figure 4.12 – Les différentes versions des programmes BLAST suivant la nature de la séquence de départ et du contenu de la banque.

Compromis sensibilité et spécificité

La sensibilité consiste à minimiser le nombre de faux négatifs. Une bonne sensibilité se caractérise par le maximum de « bons » trouvés dans le lot retenu ; elle est obtenue par un décalage du seuil vers les plus grandes valeurs de E().

La spécificité est la capacité à minimiser le nombre de faux positifs. La spécificité se caractérise par le fait de n'avoir que des « bons » dans le lot retenu. Pour cela, le seuil sera décalé vers les faibles valeurs de E().

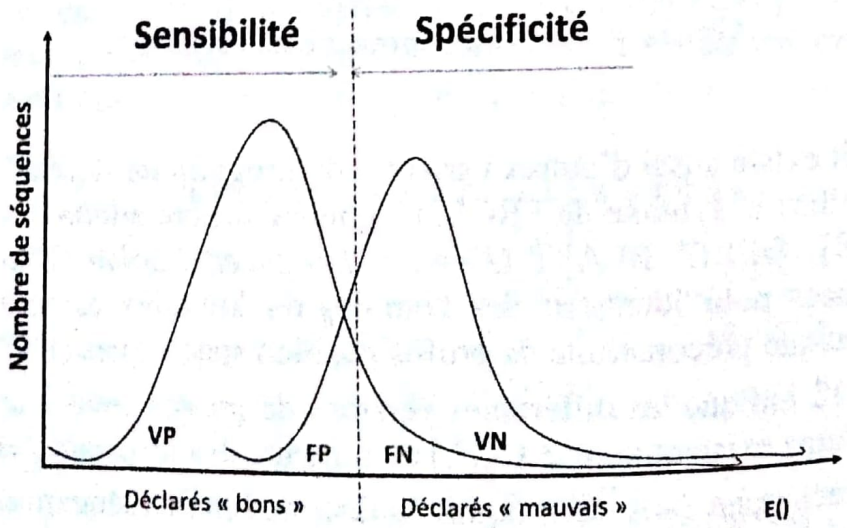


Figure 4.13 – Sensibilité *versus* spécificité.

$$\text{Sensibilité} = \frac{VP}{VP + FN} \qquad \text{Spécificité} = \frac{VN}{VN + FP}$$

Une méthode idéale présente une spécificité et une sensibilité égales à 1 (les deux distributions sont alors parfaitement séparées). Malheureusement, les deux variables varient en sens inverse et les méthodes bioinformatiques font un compromis entre les deux grandeurs. En général, on peut caractériser la qualité de la méthode grâce à la courbe de ROC Sensibilité = f(1 – spécificité), voir figure 4.14.

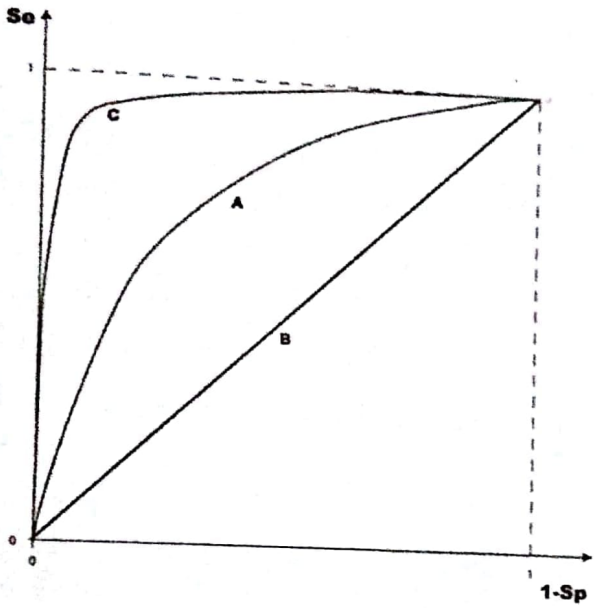


Figure 4.14 – Courbe de ROC.
La courbe B montre une méthode aléatoire. Une méthode meilleure que le hasard sera représentée par la courbe A (superposition importante des deux courbes). La courbe C est obtenue pour une meilleure séparation des deux courbes présentées dans la figure 4.13. Se : sensibilité ; Sp : spécificité.

L'option -F de BLAST permet de filtrer la faible complexité (voir chapitre 3) en remplaçant les régions par X. Cette option fixée par défaut dans le programme BLAST devrait toujours être laissée active.

ALIGNEMENT DE SÉQUENCES

5

PLAN

OBJECTIFS

- Comprendre les algorithmes de programmation dynamique
- Maîtriser les différences entre alignement local et global
- Alignements multiples
- Savoir mettre en évidence la conservation des résidus

5.1 INTRODUCTION

L'alignement de séquences concerne au minimum deux séquences. Un alignement est l'écriture de deux séquences (ou plus), l'une sous l'autre de façon à faire apparaître des identités (ou des similitudes de séquences). À chaque alignement correspond un score id% qui peut être calculé comme le pourcentage d'identité (nombre d'identités/longueur de l'alignement) lors de l'édition des séquences.

La distance d'édition de deux séquences est le nombre minimal d'opérations à effectuer pour transformer l'une dans l'autre.

Soit les deux séquences suivantes :

```
Sequence 1 A G V S I L N Y A
Sequence 2 V S I L Y A K R
```

L'écriture des deux séquences donne : Identité = 0 ; longueur = 9 ; id% = 0.

L'alignement de séquences nécessite une police de caractères non proportionnelle (courrier par exemple).

Si on admet que la séquence 2 a perdu les deux acides aminés N terminaux au cours de l'évolution, l'alignement devient :

```
Sequence 1  A G V S I L N Y A -
Sequence 2  - - V S I L Y A K R
              * * * *
```

L'alignement donne : Identité = 4 ; Longueur = 10 ; Id% = 40.

Cela revient à faire glisser la séquence 2 de deux acides aminés sur la droite. L'opération autorisée est un coulissement de séquence. On pourrait aussi envisager un coulissement de trois lettres vers la droite qui donnerait l'alignement suivant :

```
Sequence 1  A G V S I L N Y A - -
Sequence 2  - - - V S I L Y A K R
                      * *
```

Le nouvel alignement donne : Identité = 2 ; Longueur = 11 ; Id% = 18.

Trouver le meilleur alignement par glissement est trivial (informatiquement parlant).

En pratique, l'évolution a pu faire disparaître (délétion) ou apparaître (insertion) des acides aminés à l'intérieur des séquences. Une délétion dans une séquence correspond à une insertion dans l'autre (on parle alors d'« **indel** »). Cela revient à faire des « trous » (gaps) dans les séquences comme dans l'exemple suivant dans lequel une délétion dans la séquence 2 est apparue en position 5.

```
Sequence 1  A G V S I L N Y A - -
Sequence 2  - - V S I L - Y A K R
              * * * * *   * *
```

L'alignement donne : Identité = 6 ; Longueur = 11 ; Id% = 55.

À ce stade, la question qui se pose est : « Quel est le meilleur alignement ? » Le dernier possède le plus fort taux d'identité mais il a fallu pour cela créer un gap interne (faire l'hypothèse d'un événement évolutif moins probable qu'une simple substitution). Il est donc logique de pénaliser le score de l'alignement obtenu pour la création d'indel. On pourrait par exemple diminuer le score de six identités d'une valeur égale au nombre de gap interne (ici 1), ce qui donnerait 5/11, soit un score = 45 qui reste le meilleur score. À ce stade, on voit bien que le score de la pénalité va influencer l'alignement final. Si on avait choisi de pénaliser de deux unités chaque gap interne, le score aurait été 4/11, soit 36, et l'alignement à quatre identités serait devenu le meilleur. Ainsi, plusieurs systèmes de pénalités pour les gaps ont été imaginés. Tout d'abord, la suppression d'un segment peut correspondre à un seul événement évolutif qui concerne la totalité du segment. Ainsi, le système de pénalité doit prendre en compte la longueur de l'indel (un indel long doit être plus pénalisé qu'un indel court). Par ailleurs, il a pu se produire plusieurs événements évolutifs distincts dans une région (même si le nombre total de gaps reste faible) ; mais l'apparition d'un indel doit être plus pénalisante que le prolongement d'un indel déjà présent. D'où la pénalité souvent représentée par une fonction affine :

$$P = x + y\,L$$

où x est une pénalité fixe pour la création d'un indel (ouverture de gap) et $y\,L$ est la pénalité pour un gap de longueur L. Enfin, les événements évolutifs ont tendance à se produire dans les régions externes des protéines, les plus exposées au solvant, les

moins structurées. Ainsi, on peut imaginer des systèmes de pénalité qui favorisent les indels dans ces régions (voir le chapitre 9 sur les profils d'hydrophobie pour l'identification des régions externes). Le nombre d'alignements possibles entre deux séquences en autorisant des indels peut dépasser le nombre d'atomes dans l'Univers (en fonction des longueurs).

Tout comme la recherche dans les banques (décrite dans le chapitre précédent), l'alignement peut être global ou local.

Par exemple, l'alignement suivant présente sept identités :

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

Ainsi, le premier alignement est celui qui maximise les identités sur la totalité des deux séquences. On parle alors d'alignement global. L'application majeure de ce type d'alignement est l'alignement de séquences de protéines homologues en vue d'identifier des acides aminés conservés par l'évolution. Au cours de l'évolution, les séquences varient de façon à préserver (voire optimiser) la fonction biologique.

Un alignement local des mêmes séquences fournira aussi sept identités mais donnera un alignement différent :

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

Cet alignement local sera privilégié si le plus long chevauchement entre deux séquences est recherché comme dans le cas de la reconstruction à partir de données obtenues par séquençage. Le choix de l'alignement global ou local revient donc à l'utilisateur biologiste en fonction des objectifs poursuivis.

5.2 Comparaison de protéines homologues (algorithme global)

Il s'agit d'un algorithme de programmation dynamique pour l'alignement global optimal entre deux séquences. Les trois paramètres du programme sont les scores pour i) l'identité, ii) la substitution et iii) l'indel. Les deux séquences sont placées dans un tableau. Le principe consiste à calculer des scores de chaque case du tableau en partant de la case (0,0) jusqu'à la case (n,m) en remplissant ligne par ligne en simulant les trois types d'opérations possibles (insertion, délétion ou mise en correspondance). Dans le cas de la mise en correspondance, on peut avoir substitution de Ai par Bj ou identité Ai, Aj. Pour chaque cas, le score S(i,j) de la case i,j est calculé des trois façons symbolisant les trois déplacements possibles :

```
S(i,j)= S(i-1,j-1) + subst(i,j) (substitution ou identité)
S(i,j)=s(i-1,j) + Indel() car insertion à la position i-1 (ou délétion à la
position j)
S(i,j)=s(i,j-1) + Indel() car insertion à la position j-1 (ou délétion à la
position i)
```

	M	P	R	C	L	C
0	-2	-4	-6	-8	-10	-12
-2	-1	1	-1	-3	-5	-7
-4	-3	-1	0	-2	-4	-6
-6	-5	-3	2	0	-2	-4
-8	-7	-5	0	5	3	1
-10	-9	-7	-2	3	4	2
-12	-11	-9	-4	1	2	7
-14	-13	-11	-6	-1	0	5
-16	-15	-13	-8	-3	-2	3
-18	-17	-15	-10	-5	-4	1
-20	-19	-17	-12	-7	-6	-1
-22	-21	-19	-14	-9	-8	-3
-24	-23	-21	-16	-11	-10	-5

e tableau 5.2 correspondant aux déplace
haque case (table des chemins).

Tableau 5.2 - Déplacements effectués p

es flèches indiquent la transition effectuée d
btenir le score maximal de chaque case qui

Ensuite, on part de la case de plus fort score (ici la dernière case en bas à droite du tableau 5.1) et on suit les mouvements dans la table des chemins qui ont été suivis pour maximiser les scores pour remonter à l'origine (« backtracking »). Il suffit de suivre les cases grisées pour générer automatiquement l'alignement. Cependant, la case (N10, I9) présente une égalité de score selon que l'on procède d'abord à une insertion ou à une délétion. Cela indique que deux chemins sont possibles pour donner le score maximum de cette case. Cela signifie qu'il y a deux alignements optimaux équivalents :

- si à la case de la double flèche on choisit la flèche horizontale, l'alignement suivant est obtenu et le chemin suit les cases sur fond gris ;

```
MP-RCLCQR-INCYA
 | || | | | | |
-PYRCKC-RNI-CIA
```

- si à la case de la double flèche on choisit la flèche verticale, localement le chemin passe par les cases sur fond noir :

```
MP-RCLCQRIN-CYA
 | || | | | | |
-PYRCKC-R-NICIA
```

Les deux alignements présentent le même nombre d'identités (8), d'indels (5) et de substitutions (2). Le score de chaque alignement est égal $12 = (8 \times 3)+(5 \times -2)+(2 \times -1)$. Cet algorithme garantit qu'il n'y a pas de meilleur alignement que ceux proposés, mais dans cet exemple, il existe deux alignements équivalents alternatifs. Pour choisir entre des alignements équivalents alternatifs, le biologiste peut avoir recours à une séquence d'une autre espèce. La plupart des programmes d'alignement fournissent un seul alignement sans indiquer à l'utilisateur s'il y a des alignements équivalents.

```
tableaux : S1[N], S2[M], Matrice[N][M]
S1 <-- séquence 1                          /* N caractères d'indice i */
S2 <-- séquence 2                          /* M caractères d'indice j */

INDEL = -2                                 /*PARAMETRES */
SUBSTITUT = -1
IDENT = 3

pour j=0 jqa M faire
  {
    pour i=0 jqa N faire
     {
Matrice[i][0] = (i x INDEL)                /* initialization ligne 1  */
Matrice[0][j] = (j x INDEL)                /* initialization colonne 1 */
     }
}

pour j=1 jqa M faire
 {
    pour i=1 jqa N faire
     {
                      | Matrice[i][j-1] + INDEL
Matrice[i][j]<- MAX   | Matrice[i-1][j-1] + SUBS[S1[i]],[S2[j]]
                      /* SUBS[S1[i],S2[j]]=IDENT si S1[i]=S2[j] ou
```

```
                    SUBS[S1[i],S2[j]]=SUBSTITUT si S1[i]<>S2[j]*/
                  | Matrice[i-1][j] + INDEL
    }
}
```

Cet algorithme présente une complexité en $O(n^2)$.

5.3 Meilleur chevauchement entre séquences (algorithme local)

Au lieu de considérer chaque séquence globalement, cet algorithme compare les segments de toutes longueurs (suffixes de i et j dans les deux séquences) et retient celui qui maximise le score de similitude sur les segments. La solution du problème de l'alignement local consiste à trouver le score maximal des suffixes sur tous les indices i et j des deux séquences. La première ligne et la première colonne sont initialisées à 0 permettant ainsi à l'alignement de commencer n'importe où sur une des deux séquences. Un alignement local sera représenté par un chemin dans le tableau qui continue tant que le score est positif.

```
tableaux : S1[N], S2[M], Matrice[N][M]
S1 <-- séquence 1                          /* N caractères d'indice i */
S2 <-- séquence 2                          /* M caractères d'indice j */

                                           /* PARAMETRES */
INDEL = -1
SUBSTITUT = 0
IDENT = 2

pour j=0 jqa M faire
 {
   pour i=0 jqa N faire
    {
      Matrice[i][0] = 0)                   /* initialization ligne 1  */
      Matrice[0][j] = 0                    /* initialization colonne 1 */
    }
 }
pour j=1 jqa M faire
 {
   pour i=1 jqa N faire
    {
              | Matrice[i][j-1]   + INDEL
Matrice[i][j] <- MAX| Matrice[i-1][j-1] + SUBS[S1[i]],[S2[j]]
                  /*SUBS[S1[i],S2[j]]=IDENT si S1[i]=S2[j] ou
                    SUBS[S1[i],S2[j]]=SUBSTITUT si S1[i]<>S2[j]*/
                  | Matrice[i-1][j]   + INDEL
    }
 }
```

Soit les deux séquences : LIBRESEQENCE et SEQANCELIBRE.

Avec identité = 2, substitution = 0 et indel = –1, la table des scores sera :

Tableau 5.3 - Score d'alignement local.

		L	I	B	R	E	S	E	Q	E	N	C	E
	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	0	2	1	0	0	0	0	0
E	0	0	0	0	0	2	1	4	3	2	1	0	2
Q	0	0	0	0	0	1	2	3	6	5	4	3	1
A	0	0	0	0	0	0	1	2	5	6	5	4	3
N	0	0	0	0	0	0	0	1	4	5	8	7	6
C	0	0	0	0	0	0	0	0	3	4	7	10	9
E	0	0	0	0	0	2	1	2	2	5	6	9	12
L	0	2	1	0	0	1	2	1	2	4	5	8	11
I	0	1	4	3	2	1	1	2	1	3	4	7	10
B	0	0	3	6	5	4	3	2	2	2	3	6	9
R	0	0	2	5	8	7	6	5	4	3	2	5	8
E	0	0	1	4	7	10	9	8	7	6	5	4	7

La table des déplacements correspondante sera :

Tableau 5.4 - Déplacements correspondant à l'alignement local.

		L	I	B	R	E	S	E	Q	E	N	C	E
		-	-	-	-	-	-	-	-	-	-	-	-
S	-	↘	↘	↘	↘	↘	↘	→	↘	↘	↘	↘	↘
E	-	↘	↘	↘	↘	↘	↓	↘	→	↘	→	↘	↘
Q	-	↘	↘	↘	↘	↓	↘	↓	↘	→	→	→	↓
A	-	↘	↘	↘	↘	↘	↘	↘	↓	↘	↘	↘	↘
N	-	↘	↘	↘	↘	↘	↘	↘	↓	↘	↘	→	→
C	-	↘	↘	↘	↘	↘	↘	↘	↓	↘	↓	↘	→
E	-	↘	↘	↘	↘	↘	→	↘	↓	↘	↓	↓	↘
L	-	↘	→	↘	↘	↓	↘	↘	↘	↓	↘	↓	↓
I	-	↓	↘	→	→	→	↘	↘	↘	↓	↘	↓	↓
B	-	↘	↓	↘	→	→	→	→	↘	↓	↘	↓	↓
R	-	↘	↓	↓	↘	→	→	→	→	→	↘	↓	↓
E	-	↘	↓	↓	↓	↘	→	↘	→	↘	→	↓	↘

L'alignement généré sera pour la diagonale grisée du haut :

```
SEQ1 LIBRESEQENCE-----
SEQ2 -----SEQANCELIBRE
          *** ***
```

Noter que la valeur de 10 dans la table à la fin des segments indique que « LIBRE » dans la table est une 2e zone de similitude entre les deux séquences qui conduit à l'alignement suivant la diagonale en fond noir du bas :

```
SEQ1 -------LIBRESEQENCE
SEQ2 SEQANCELIBRE-------
          *****
```

Il faut cependant faire attention car l'alignement optimal (au sens programmation dynamique) n'est pas obligatoirement celui qui est pertinent au niveau biologique. Cette remarque est d'autant plus valable que les séquences ont présenté des taux importants de mutations et que l'alignement final est peu robuste (sensibilité aux changements de paramètres). Finalement, d'un point de vue biologique, l'alignement le plus pertinent est celui qui retrace le déroulement évolutif le plus probable. Dans ce contexte, il est évident que la pression évolutive s'exerce sur les séquences de façon à favoriser les mutations des bases tout en conservant les acides aminés, ce qui traduit que le code génétique introduit un biais important dans les probabilités de mutations des positions. Finalement en biologie, l'hypothèse d'équiprobabilité mutationnelle des positions n'est jamais satisfaite.

5.4 Alignements multiples

En général, le biologiste dispose d'un grand nombre de séquences (plusieurs centaines) et a besoin d'un alignement multiple des séquences appartenant à la même famille afin d'identifier les résidus essentiels qui ont été préservés au cours de l'évolution. Dans ce cas, il s'agit le plus souvent d'un alignement global qui est recherché. La méthode de programmation dynamique peut, au moins dans le principe, s'appliquer sur N séquences. Il s'agira de calculer les scores dans un hyper cube de dimension N. Cependant, elle est inexploitable en pratique (le nombre de chemins menant à chaque case est égal à 2^N) et la taille mémoire qui serait nécessaire pour stocker les matrices deviendrait prohibitive comme le montre le tableau 5.5.

Tableau 5.5 - Relation entre nombre (N), longueur (L) des séquences et mémoire requise par la programmation dynamique.

	Taille moyenne des séquences					
N	L=100 AA		L=500 AA		L=1 000 AA	
	Éléments	Mémoire	Éléments	Mémoire	Éléments	Mémoire
2	100^2	10 Ko	500^2	250 ko	$1\,000^2$	1 Mo
3	100^3	1 Mo	500^3	125 Mo	$1\,000^3$	1 Go
5	100^5	10 Go	500^5	30 Po	$1\,000^5$	1 000 Po
10	100^{10}	100 000 Po	500^{10}	10^{11} Po	$1\,000^{10}$	10^{15} Po

À la vue du tableau ci-dessus, il est évident que la programmation dynamique brute n'est plus envisageable lorsque le nombre de séquences croît. Dès lors, les alignements multiples utiliseront toujours des heuristiques conduisant à des alignements

questionnables par le biologiste. Il n'est pas rare d'optimiser « à la main » des alignements multiples en se basant sur la connaissance biologique apportée par l'expert du domaine biologique et de la famille de protéine concernée.

De nombreuses approches ont été envisagées pour les alignements multiples dont la méthode d'alignement progressif. En simplifiant, il s'agit de décomposer la question de l'alignement simultané de N séquences en $N \times (N-1)/2$ alignements de toutes les paires possibles puis de grouper les paires en partant tout d'abord des paires les plus proches pour ensuite diverger.

Soit les quatre séquences A, B, C et D. Les six alignements par paires sont effectués par programmation dynamique et les scores obtenus sont placés sur un arbre des scores. Plus la distance sur le dendrogramme entre deux lettres est faible, plus les séquences sont proches.

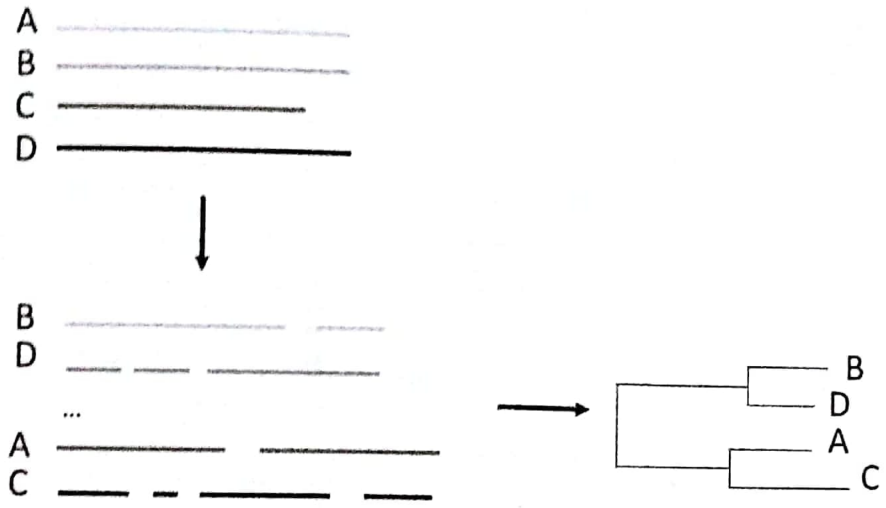


Figure 5.1 – Alignement binaire (par paires).

Ensuite les alignements sont progressivement ajoutés les uns aux autres en partant des paires ayant les plus fortes ressemblances.

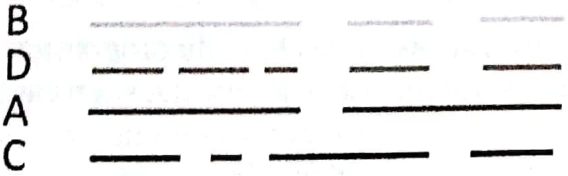


Figure 5.2 – Alignement multiple progressif.

Le point de départ est l'alignement B-D ensuite, l'alignement A-C est agrégé à l'alignement B-D.

Une des premières façons d'agréger les séquences a été d'écrire chacun des deux alignements B-D et A-C sous forme d'une séquence. Il suffit ensuite d'aligner les deux séquences résultantes par programmation dynamique. L'inconvénient de cette

méthodologie est que l'algorithme est de type glouton (« greedy ») car les alignements déjà effectués ne sont pas réévalués pendant l'agrégation. La conséquence de cette procédure est que la tendance générale est de dégrader progressivement l'alignement en incorporant de plus en plus d'indels. C'est d'ailleurs pour cela que l'incorporation de séquences divergentes (qui peuvent être des séquences intruses) est retardée au maximum dans le processus.

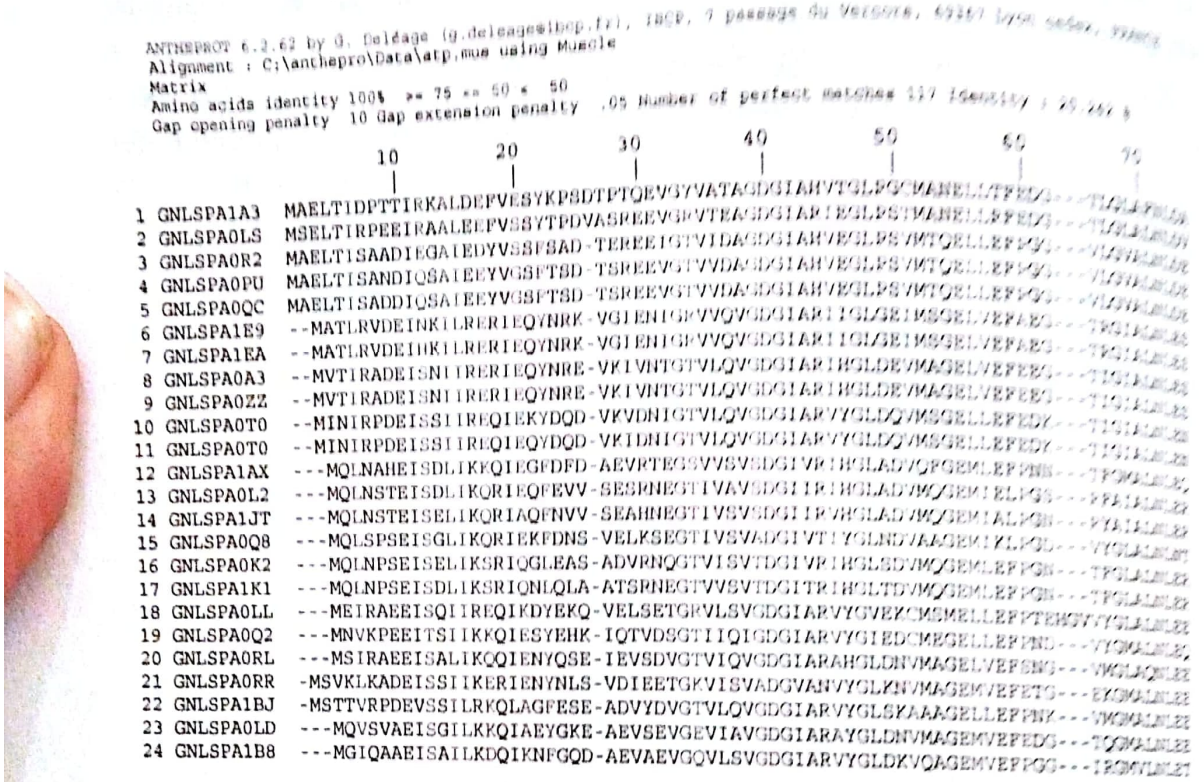


Figure 5.3 – Alignement multiple de séquences de sous-unités α d'ATP synthase.

Les méthodes les plus récentes utilisent un alignement itératif, dans lequel l'alignement est construit de manière itérative en optimisant progressivement le score global. Les itérations peuvent être obtenues par des techniques stochastiques de type tirage par Monte Carlo.

Le biologiste s'interroge parfois sur le choix du programme d'alignement multiple à utiliser. Il n'y a pas de réponse générique à cette question car aucun programme n'est optimal au sens informatique. Le meilleur programme est celui qui fourni un alignement pertinent dans un temps raisonnable. Un des critères est la rapidité d'exécution qui sera prédominante si le nombre de séquences est élevé (~400). À titre d'illustration, un alignement de 400 séquences de sous-unités d'ATP synthase de longueur moyenne 500 acides aminés prend 2 minutes à 3 minutes avec Muscle sur un ordinateur portable. Un autre critère est la proximité des séquences à aligner. Si les séquences sont très proches (>80% Id) et ne présentent pas de difficulté d'alignement (famille homogène), rien ne sert d'utiliser un programme très sophistiqué (par exemple T-COFFEE qui prend en compte les structures et qui sera donc plus lent). Le tableau 5.6 est une comparaison qualitative de différents programmes d'alignement multiple.

Tableau 5.6 - Quelques programmes d'alignement multiple.

	Rapidité	Séquences proches	Séquences éloignées	Qualité
Multalin	++	+++	+	++
Clustal W	+	++	++	+++
Muscle	+++	+++	+	+++
MAFFT	++	++	+	+++
T-Coffee	+	+	+++	+++
DIALIGN	+	+	+++	+

5.5 Représentation « logo »

À partir d'un alignement multiple, il est possible de générer une séquence virtuelle qui contient pour chaque position l'acide aminé le plus représenté (figure 5.4). Ainsi dans la représentation logo (calcul de l'**entropie** ci-après), la hauteur d'une colonne indique le degré de conservation de la position et la hauteur d'une lettre reflète la fréquence relative de la lettre à la position considérée.

La hauteur $H(i)$ de la lettre a à la position i est :

$$H(i) = I(i) \cdot f_{a,i}$$

$$I(i) = \log_2(20) - (E_i + e_n)$$

Avec :

$$E_i = -\sum f_{a,i} \log_2 f_{a,i}$$

où $f_{a,i}$ est la fréquence relative de l'acide aminé a à la position i et e_n est une variable d'ajustement telle que :

$$e_n = \frac{s-1}{2\ln(2)n}$$

avec s = 20 pour les protéines et n le nombre de séquences dans l'alignement.

Figure 5.4 - Le programme logo.

Par ailleurs, la fréquence de représentation des acides aminés peut être utilisée pour construire une **signature** caractéristique de la fonction étudiée.