République Algérienne Démocratique et Populaire
وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique
جامعة برج بوعريريج
Université de Bordj Bou Arréridj

Faculté de Mathématiques et d'informatique

Département Informatique

MEMOIRE
Présenté en vue de l'obtention du Diplôme de
Master Informatique
Spécialité : Technologies de l'Information et de la Communication

Thème :

Extraction des règles d'association pour l'amélioration de l'alignement des séquences biologiques

Présenté par :

- ***BOUABDALLAH AMIRA.***
- ***ZERGUINE IBTISSEM.***

Devant le jury composé :

Président	***Mr. Attia Abdelouahab.***	MCA à L'U.EI Bachir El Ibrahimi-bba.
Examinateur	***Mr. Maaza sofian.***	MCA à L'U.EI Bachir El Ibrahimi-bba.
Encadrant	***Mr. Zouache djaafar.***	MCA à L'U.EI Bachir El Ibrahimi-bba.

Promotion : 2019/2020

Remerciements

Tout d'abord nous remercions notre bon Dieu le tout puissant, pour son aide et pour nous avoir donné le courage et la patience afin d'accomplir notre travail.

Nous tenons aussi à remercier notre encadreur Mr Zouache Djaafar pour son suivi et ses précieux conseils durant l'évolution de ce travail.

Nos remerciements qui vont également :

A tous nos enseignants qui ont contribué à notre formation, ainsi que tous les personnels de Collège.

A toute notre promotion pour tous les bons moments qu'on a passés ensemble.

Dédicace

Nous voilà à la fin d'un parcours c'est long mais plein de bons événements, de meilleurs souvenirs et d'inoubliables amitiés tissées tout au long de nos années d'études ;

C'est avec une grande gratitude et des mots sincères, que nos dédie ce modeste travail à

Nos parents, vous êtes pour nous le symbole de la bonté par excellence et la source de la tendresse. Rien au monde ne vaut les efforts fournis jour et nuit pour nos éducation et nos biens être.

Que dieu le tout puissant, vous préserve et vous accorde santé, bonheur et longue vie.

Nos frères et nos sœurs et nos belles familles.

Aux deux personnes que nous aimons tant qui nous ont soutenus tout au long de ce projet, nos maris Hichem et Amine.

Merci à mes chères Houda et Rayan qui font partie de ma vie. Et a tous mes amis et collègues.

Table des matières

Table des matières

Chapitre III : Introduction au data mining, les techniques de regroupement hiérarchique et méthode de construction des arbres phylogénétique

Chapitre IV : l'algorithme CLUSTAL pour l'alignement multiple de séquence

Liste des figures

Liste des tableaux

Introduction générale

Introduction générale

La bio-informatique est une discipline qui vise le traitement automatique de l'information biologique. Bio-informatique implique maintenant la création et le développement de bases de données, des algorithmes, des techniques informatiques et statistiques et de la théorie pour résoudre les problèmes formels et pratiques découlant de la gestion et l'analyse des données biologiques. L'objectif de la bio-informatique est donc de réunir les compétences disponibles dans chaque discipline scientifique afin de contribuer à l'amélioration des méthodes de résolution. Pour cela la construction automatique des alignements est devenue aujourd'hui une tâche importante en bio-informatique. Par exemple pour le problème d'alignement multiple de séquences, des solutions informatiques et mathématiques ont été apportées.

L'alignement multiple de séquences MSA (Multiple Sequence Alignment) consiste à aligner plusieurs séquences dans leur intégralité afin de tirer les relations entre une famille de séquences. Le but principal de l'alignement multiple est de montrer les rapports essentiels et les caractéristiques communes entre un ensemble de séquences de protéines ou de nucléotides. Le MSA permet de caractériser les régions conservées et les régions variables au sein d'une famille de séquences. On peut étendre l'algorithme exact d'alignement de deux séquences pour l'appliquer à l'alignement multiple. On est alors garanti d'obtenir une solution optimale du point de vue de la fonction objectif. Cependant la complexité de l'algorithme qui en résulte est telle qu'on ne peut l'appliquer qu'à un petit nombre des séquences de faible longueur. Différents algorithmes existent pour l'alignement multiple de séquences. On peut les classer selon trois catégories : Les algorithmes exacts, Les algorithmes progressifs et Les algorithmes itératifs.

L'objectif de cette mémoire est pour la résolution du problème d'alignement multiple des séquences, par l'algorithme Clustal.

Ce travail comprend cinq chapitres qui nous permettent de présenter les différents aspects de notre travail.

Le premier chapitre sera introduit le domaine de la biologie moléculaire et la bio-informatique. Le deuxième chapitre fournit une description des différents alignements des séquences et pour déterminer les similitudes et liens on a l'arbre phylogénétique. Au troisième chapitre, nous discuterons les techniques de regroupement hiérarchique tel que la classification hiérarchique ascendante et les méthodes de construction de l'arbre phylogénétique. Dans le quatrième chapitre nous présenterons l'algorithme clustal pour l'alignement multiple de séquence. Le dernier chapitre nous effectuons l'environnement de développement et les résultats expérimentaux le mémoire s'achèvera par une conclusion.

CHAPITRE 1

Introduction à la Bio-informatique

1. Introduction

Le terme de "bio-informatique" date du début des années 80. Le concept sous-jacent de traitement de l'information biologique est bien plus vieux. Durant les années 60, la biologie moléculaire a eu besoin de modélisation formelle, ce qui a mené à la création des "biomathématique" [1].

L'apparition de la bio-informatique n'est donc pas une conséquence de la génomique (séquençage d'un génome et son interprétation), mais plutôt une de ses fondations [1].

La bio-informatique est l'étude de l'information biologique. Ce n'est pas simplement l'application à la biologie de l'informatique ; c'est une branche à part entière de la biologie. La bio-informatique actuelle se concentre surtout sur l'étude des séquences d'ADN et sur le repliement des protéines [1].

2. Quelques dates clés en biologie

1953 Découverte de la structure de l'ADN James D. Watson and Francis Crick.

1955 Détermination de la structure des protéines Frédérique Sanger (Prix Nobel 1958) Découverte de la première séquence protéique : l'insuline.

1965 Découverte des mécanismes de la régulation génétique. Jacques Monod, François Jacob et A. Wolf.

1967 Méthode de construction d'arbres phylogénétiques par Fitch et Margoliash.

1968 Atlas of protein sequence and structure. Premier ouvrage contenant 20 séquences de protéines. Margaret Dayhoff.

1970 Alignement global de séquences. Algorithme de Needleman et Wunsch.

1977 Méthode de séquençage de l'ADN. Séquençage du premier génome à ADN, le bactériophage phiX174 (5386pb) Frederik Sanger.

1980 Création de la banque EMBL : banque européenne généraliste de séquences nucléiques.

1982 Création de la banque Genbank : banque américaine généraliste de séquences nucléiques.

1987 Apparition de la technologie des puces à ADN.

1990 Programme BLAST (Altschul et al.) : recherche rapide d'alignements locaux dans une banque.

1996 Séquençage du 1er génome eucaryote, Saccharomyces cerevisiae (12 Mb, 6000 gènes, 16 chromosomes).

2003 Séquençage complet du génome humain.

3. Notion de base de biologie moléculaire

3.1. Cellule

La cellule est l'unité de base de tout organisme (une sorte de pièce de légo). Un corps humain est un immense assemblage, consistué de ces pièces. La première distinction que l'on peut faire parmi les organismes vivants sépare ceux dont les cellules ont un noyau (les eucaryotes, comme nous), et ceux qui n'ont pas de noyau, plus primitifs (les procaryotes, comme les bactéries). Parmi les êtres dont les cellules ont un noyau, certains n'ont qu'une seule cellule (les protistes, comme les amibes), d'autres sont constitués des très nombreuses cellules [2].

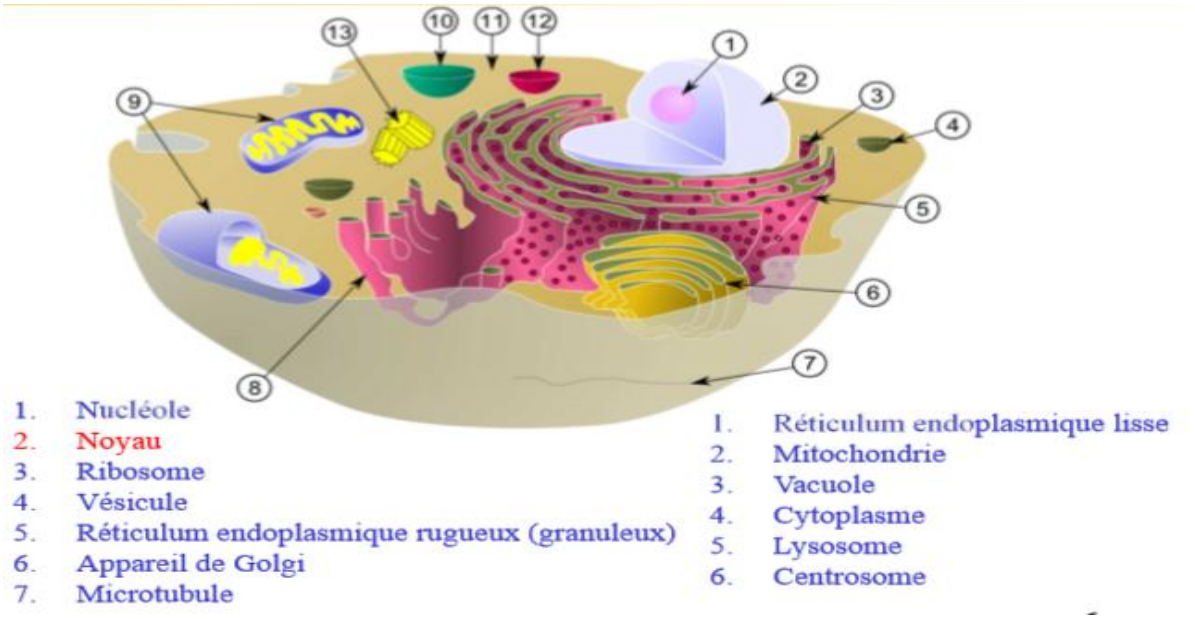


Figure 1.1 : Schéma d'une cellule animale

3.2. ADN (Acide Désoxyribonucléique)

L'ADN est un acide de la famille des acides nucléiques, composé d'une succession de nucléotides. On peut définir l'ADN d'un organisme comme l'ensemble des informations génétiques nécessaires à l'édification, au fonctionnement et à la reproduction de chaque organisme [7].

3.2.1. Nucléotides

Un nucléotide d'ADN (Figure 1.2) a 3 composants : un sucre (désoxyribose), un composant d'acide phosphorique (phosphate), et une base d'azote (un des quatre types : Adénine ou Adénosine (A), Guanine (G), Cytosine (C) et Thymine (T)).

L'ADN peut être en simple brin ou double brin. Un brin simple (aussi appelé poly nucléotide) est un Polymère linéaire (Figure 1.3).

On représente un poly nucléotide par une séquence orientée de lettres :

5’ -A-T-T-C-A-G-G-C-A-T-T-A-G-C- 3’

Les brins de nucléotides peuvent coller ensemble pour former une épine dorsale continue. Ceci donne une forme d’échelle (Figure 1.4). La forme d'échelle se torde sur elle-même pour donner une forme hélicoïdale (Figure 1.5). Cette structure est la célèbre " double hélice ", découverte par Crick et Watson en 1953. Les bases ou nucléotides (A, T, C, G) s’organisent en paires selon une complémentarité exclusive : A-T et G-C. C’est cet appariement qui permet un enroulement quasi-parfait en hélice droite des deux chaînes sucre -phosphate qui portent ces nucléotides [3].

La structure est stabilisée par l’interaction (liaisons d’hydrogène) entre les bases et l’empilement successif des paires de nucléotides (Figure 1.6).

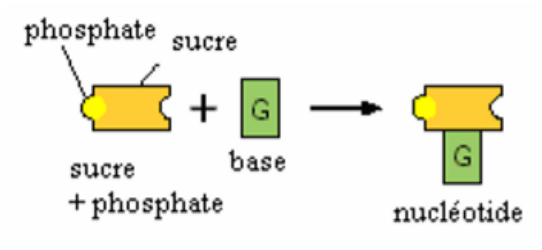


Figure 1.2 : Construction d’un nucléotide

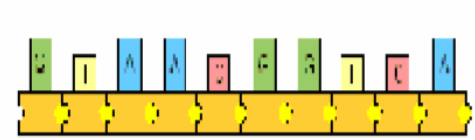


Figure 1.3 : Un brin d’ADN ou polynucléotide

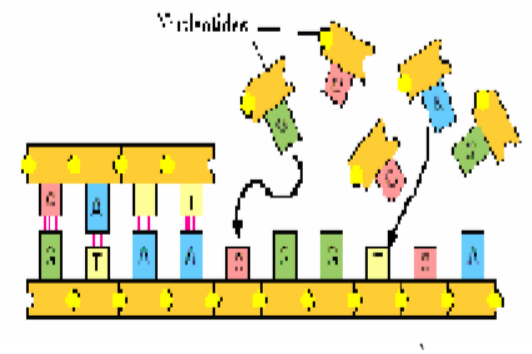


Figure 1.4 : Construction du 2ème brin d’ADN

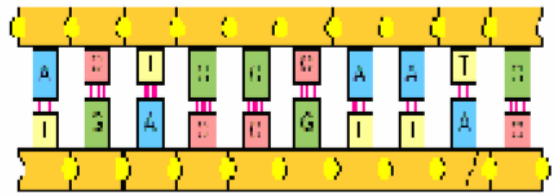


Figure 1.5 : Double brins d’ADN (Forme d’échelle)

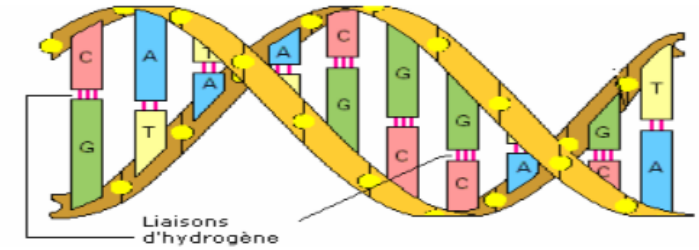


Figure 1.6 : Double brins d’ADN (Forme hélicoïdale)

Dans un brin d'ADN, il y a des segments dits codants (Exons) (figure 1.8) et des segments non codants (Introns). Le premier type qui est l'exon, va participer à la génération d'autres macromolécules (ARNs et par la suite des protéines) contrairement aux introns qui sont sans utilité apparente [3].

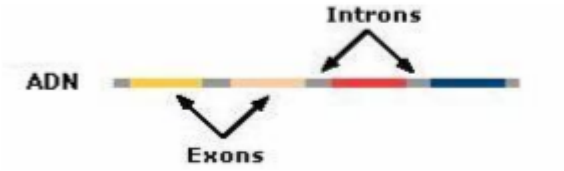


Figure 1.7: Exons et Introns dans un brin d'ADN

3.3. Les Chromosomes

Les chromosomes sont des éléments du noyau cellulaire en nombre constant, qui déterminent l'hérédité. Un chromosome est une structure en bâtonnet, constituée de longues chaînes d'ADN, auxquelles sont fixées des protéines. L'ADN de l'homme est divisée en 23 paires de chromosomes contenus dans le noyau de chacune de ces cellules, 22 paires sont communes aux deux sexes. Les deux chromosomes restants sont les chromosomes sexuels. Chez la femme, ils forment une paire. On les appelle les chromosomes X et l'autre, beaucoup plus court est appelé chromosome Y.

3.4. L'ARN (Acide Ribonucléique)

L'ARN (Acide Ribonucléique) ressemble énormément à l'ADN mais il y a des différences telles que :

- Le sucre de l'ADN (désoxyribose) et celui de l'ARN est le ribose.
- La Thymine (T) de l'ADN est remplacée par l ' uracile (U) dans l'ARN.
- L'ARN peut s'apparier avec un autre ARN complémentaire mais les ARNs sont généralement simple brin. Contrairement aux brins de l'ADN qui vont en couple.
- 3 types d'ARNs ont été identifiés : ARN messager (ARNm), ARN ribosomiques (ARNr) et ARN transfert (ARNt). Mais d'autres types ont été découverts ces dernières années.

3.4.1. Structures d'un brin ARN

Un brin d'ARN peut avoir plusieurs structures : primaire (Figure 1.8), secondaire (Figure 1.9) et tertiaire (Figure 1.10) [4]. Cette définition est valable même pour les protéines à qui on peut attribuer encore une structure quartenaire.

Figure 1.8 : La structure primaire d'une séquence d'ARNt de la phénylalanine

Figure 1.9 : La structure secondaire d'une séquence d'ARNt de la phénylalanine

Figure 1.10 : La structure tertiaire d'une séquence d'ARNt de la phénylalanine

3.5. Les Protéines

Les protéines sont les macromolécules les plus importantes. Elles sont responsables de presque de toutes les réactions biochimiques qui ont lieu à l'intérieur de la cellule. Les protéines sont de sortes différentes et avec une variété de fonctionnalités. Certaines d'entre elles incluent [3] :

- Protéines structurelles : elles sont les bases de construction des divers tissus.
- Enzymes : elles catalysent les réactions chimiques essentielles qui auraient pris beaucoup de temps pour se produire.
- Transporteuses : elles portent les éléments chimiques qui font partie de l'organisme à d'autres (par exemple les hémoglobines qui portent l'oxygène).

Les protéines se composent de chaîne des acides aminés. Chaque acide aminé a une structure constante. Il y a 20 acides aminés. Deux acides aminés peuvent se joindre, avec un " lien de peptide ", formant une chaîne : un " polypeptide ".

Une séquence protéique est une collection ordonnée de lettres choisis dans l'alphabet = {A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. Où chacune des lettres correspond à un acide aminé.

Exemple d'acide aminé : « Lysine » codé par la lettre « K ».

Exemple d'une protéine : l'insuline :

"F V N Q H L C G S H L V E A L Y L V C G E R G F F Y T P K A"

La synthèse de protéine se produit dans des structures appelées les ribosomes situés dans la cellule mais en dehors du noyau. Le modèle de la protéine est dans l'ADN, située dans le noyau. Donc il y a un besoin d'un " messager " pour transférer l'information à partir de l'ADN aux ribosomes. L'ARN est ce messager (ARNm). Il est synthétisé en utilisant l'ADN comme modèle. Ce processus s'appelle **la transcription.**

Comment interprète-on l'information diffusée par ARNm ? Ceci est une séquence de "triplets" de nucléotides, ou de codons. Chaque codon indique un acide aminé. Mais puisqu'il y a 43 = 64 de codons possibles, mais seulement 20 acides aminés, il y a une certaine redondance dans le code où des triplets différents codent le même acide aminé (voir la table 1.1). Cette fonction de codage, f : codon acide aminé est le code génétique elle est universel, pour tous les organismes.

N.B. : les trois codons spéciaux : stop codons ; ils ne codent pas un acide aminé ; mais ils indiquent la fin d'une région de codage de protéine sur une grande molécule d'ADN.

La traduction : est le processus par lequel une séquence des codons est traduite vers une séquence d'acides aminés. Une molécule appelée l'ARN de transfert (ARNt) permet le passage des codons aux acides aminés. ARNt contient un triplet appelé anticodon, celui-ci possède une extrémité à laquelle un acide aminé spécifique vient s'attacher.

ARNt est situé dans le cytoplasme, et porte les acides aminés vers les ribosomes. Les acides aminés rassemblés par les ARNts vont alors collés les uns aux autres pour former une chaîne de peptides appelée polypeptides. Une chaîne de polypeptides peut atteindre une taille de 50 à 30000 acides aminés, la moyenne étant 400 acides aminés.

Après transcription d'ADN et avant la synthèse de protéine, un processus enlève quelques segments de l'ARN (introns), laissant seulement les codons significatifs (exons) qui seront exprimés. Ce processus est appelé « l'épissage »[5].

La structure de la protéine : La protéine possède quatre structures : primaire, secondaire, tertiaire et quartenaire.

Parmi les paradigmes de la biologie moléculaire, celui de la relation entre structure et fonction. La structure secondaire ou tertiaire peut inférer la fonction d'une protéine. Donc connaître la structure va faciliter l'identification de sa fonction et par conséquent son importance pour tout l'organisme [4].

Structure ⟶ **Fonction**

1st base													
	U			C			A			G			
2nd base	U	UUU	Phenylalanine	U	UCU	Serine	U	UAU	Tyrosine	U	UGU	Cysteine	U	3rd base
		UUC	Phenylalanine	C	UCC	Serine	C	UAC	Tyrosine	C	UGC	Cysteine	C	
		UUA	Leucine	A	UCA	Serine	A	UAA	Stop	A	UGA	Stop	A	
		UUG	Leucine	G	UCG	Serine	G	UAG	Stop	G	UGG	Tryptophan	G	
	C	CUU	Leucine	U	CCU	Proline	U	CAU	Histidine	U	CGU	Arginine	U	
		CUC	Leucine	C	CCC	Proline	C	CAC	Histidine	C	CGC	Arginine	C	
		CUA	Leucine	A	CCA	Proline	A	CAA	Glutamine	A	CGA	Arginine	A	
		CUG	Leucine	G	CCG	Proline	G	CAG	Glutamine	G	CGG	Arginine	G	
	A	AUU	Isoleucine	U	ACU	Threonine	U	AAU	Asparagine	U	AGU	Serine	U	
		AUC	Isoleucine	C	ACC	Threonine	C	AAC	Asparagine	C	AGC	Serine	C	
		AUA	Isoleucine	A	ACA	Threonine	A	AAA	Lysine	A	AGA	Arginine	A	
		AUG	Methionine (Start)	G	ACG	Threonine	G	AAG	Lysine	G	AGG	Arginine	G	
	G	GUU	Valine	U	GCU	Alanine	U	GAU	Aspartic Acid	U	GGU	Glycine	U	
		GUC	Valine	C	GCC	Alanine	C	GAC	Aspartic Acid	C	GGC	Glycine	C	
		GUA	Valine	A	GCA	Alanine	A	GAA	Glutamic Acid	A	GGA	Glycine	A	
		GUG	Valine	G	GCG	Alanine	G	GAG	Glutamic Acid	G	GGG	Glycine	G	

Nonpolar, aliphatic Polar, uncharged Aromatic Positively charged Negatively charged

Tableau 1.1 : Le code génétique des acides aminés.

3.6. Le Gène

Le gène est l'élément d'un chromosome constitué d'ADN et conditionnant la transmission et l'expression d'un caractère héréditaire déterminé.

Les gènes sont les unités responsables de l' hérédité, qui contrôlent les caractères ou aptitudes propres à un organisme.

Les gènes sont situés sur les chromosomes, dans le noyau de la cellule. Ils sont constitués de segments de molécules d'ADN. Chaque gène est une séquence d'ADN d'environ un millier de paires de bases. La molécule d'ADN d'un seul chromosome humain comporte environ 175 000 gènes. Ces gènes agissent par paires pour transmettre différents caractères, allant de la couleur des yeux jusqu'aux constituants du système nerveux.

4. Définition et thèmes en bio-informatique

4.1. Alignement de séquences

L'alignement de séquences est une problématique importante de la bio-informatique. En effet aligner des séquences constitue un problème à part entière, mais il est également utilisé comme point de départ pour d'autres problèmes de bio-informatique [7]. Le problème de l'alignement de séquence sera détaillé dans le chapitre suivent :

4.2. Phylogénie

La phylogénie ou reconstruction phylogénétique peut être définie comme la reconstruction de l'histoire évolutionnaire d'un ensemble d'espèces. L'évolution entre les

espèces est représentée sous forme d'un arbre (phylogénétique) dont les branches indiquent le degré de proximité entre les espèces [8].

Il existe deux façons de caractériser les espèces :

- Soit en se basant sur leurs phénotypes, c'est à dire l'ensemble des caractéristiques qu'elles expriment (suivant leur apparence) [8].
- Soit en se basant sur leurs génotypes, c'est à dire l'ensemble des caractéristiques comprisent dans leurs génomes et qu'elles peuvent éventuellement exprimer [8].

Le premier genre d'approche basé sur les phénotypes est utilisée lorsqu'on ne dispose d'aucune information sur les génomes des espèces. Depuis le lancement des différents programmes de séquençage des génomes on a de plus en plus tendance à utiliser le second type d'approche [8].

Ici nous ne nous focaliserons que sur le second type de caractérisation. Nous utiliserons des séquences nucléotidiques (ou d'acides aminés) pour distinguer les espèces entre elles [8].

Il existe 3 types d'approches pour s'attaquer au problème de reconstruction phylogénétique

- Les approches basées sur les distances utilisent une mesure de distance ou de similarité afin de regrouper des séquences proches, en anglais on parle de clustering. Ces approches sont très efficaces car elles sont basées sur un algorithme de complexité polynomiale ce qui les rend particulièrement adaptées pour des études à grande échelle [8].

Les approches axées sur les caractères cherchent le meilleur arbre possible dans une topologie d'arbre étant donné un critère d'optimalité. Le critère le plus largement utilisé est le critère de maximum de parcimonie (Maximum Parsimony Criterion) pour lequel on considère que le meilleur arbre est celui qui requiert le minimum de changements. Le problème de maximum de parcimonie est un problème NP-dur dans le cas général, il est donc nécessaire de faire appel à des techniques heuristiques et d'optimisation afin de le traiter efficacement dans un temps raisonnable [8].

- Enfin, les approches statistiques principalement basées sur la méthode de maximum de vraisemblance (Maximum Likelihood) et l'approche bayésienne [8].

Chacune de ces méthodes possède des avantages et des inconvénients. Faisant partie d'une équipe de recherche travaillant en optimisation combinatoire, nous avons choisi de nous intéresser à la recherche du Maximum de Parcimonie [8].

4.3. Recherche de motifs

Un motif (ou Pattern) au sens bio-informatique du terme, représente une expression qui permet de caractériser un ensemble de séquences d'ADN, d'ARN ou de protéines. Le motif peut concerner les structures primaires, secondaires et tertiaires. Le motif trouve notamment

son intérêt dans la caractérisation des fonctions des protéines : si on était capable d'exhiber un motif pour chaque fonction alors on serait en mesure de prédire automatiquement la fonction associée à une protéine [7].

On distingue deux étapes dans la recherche de motif :

- La découverte qui, étant donné un ensemble de séquences, tente d'exhiber un motif commun à ces séquences. Il s'agit d'un problème complexe car on ne sait pas ce qui doit être trouvé. Dans le cas de séquences similaires, on peut utiliser un alignement multiple des séquences afin de trouver un motif simple [7].
- La recherche à proprement parler, qui concerne la détection d'un motif donné sur un ensemble de séquences. Ce problème est bien plus simple que le premier [7].

Les deux problèmes rencontrés dans la recherche de motifs concernent la définition du motif. Un motif généralement défini à partir d'un ensemble référence de séquences qui possèdent la même fonction :

- Si le motif n'est pas assez fin, on risque de le découvrir sur des séquences qui n'ont pas la fonction liée au groupe de séquences référence, ces séquences seront appelées faux positifs [8].
- Par contre, s'il est trop fin, certaines séquences qui possèdent la fonction liée au motif ne seront pas découvertes, on les qualifiera de vrais négatifs [8].

4.4. Prédiction de structures

Le nombre de structures primaires possibles pour les protéines est exponentiel en fonction de la longueur ℓ [7]. En revanche le nombre de combinaisons de structures tridimensionnelles est beaucoup plus réduit [7]. Ainsi, des séquences très différentes peuvent avoir des structures similaires. La structure tridimensionnelle d'une séquence est une information contenue dans sa structure primaire. Cependant, cette information est actuellement difficile à déterminer sans utiliser une analyse directe de la séquence. Connaitre la structure primaire d'une protéine ne permet pas actuellement d'en déduire sa structure tridimensionnelle [7].

La structure 3D d'une protéine peut être déterminée expérimentalement par cristallographie ou par résonance magnétique nucléaire. Ces méthodes sont toutefois assez lourdes à mettre en œuvre, et nécessitent un matériel spécialisé. La prédiction de structures est une branche de la bio-informatique qui consiste à essayer de déterminer la structure d'une protéine sans passer par la phase expérimentale [7].

La méthode qui donne les meilleurs résultats actuellement procède par homologie, c'est-à-dire en se basant sur des séquences ayant des structures primaires assez proches, et dont la structure tridimensionnelle est déjà connue. Il s'agit l`a d'une application directe du problème d'alignement de séquences. Il est nécessaire pour cela de trouver des séquences similaires [7].

5. Représentation informatique

Nous abordons dans cette section le point de vue informatique pour la représentation des données biologiques [7]. Les notions vues à la section précédente peuvent être formalisées. Il devient ainsi possible de les représenter aisément pour un traitement informatique [7].

5.1. Séquence

On appelle séquence S sur un alphabet ∑ une suite ordonnée d'éléments appartenant à

∑, S = (x1,2,...xn) [7].

5.2. Alphabet

On appelle alphabet tous ensemble fini ∑ des symboles distincts deux à deux, ainsi $l'ADN$ et $l'ARN$ sont représentés par un ensemble de quatre lettres ($A - T - C - G$ pour ADN) ($A - U - C - G$ pour ARN) et les protéines avec un ensemble de 20 $lettres$ [7].

5.3. Sous séquence

Soit S une séquence de longueur n On appelle sous séquence de S toute partie de S Composée d'un ensemble de caractères consécutifs de S [7].

Nous noterons S [i…j] avec 1 ≤ i ≤ j ≤ n la sous séquence S = (xi,...xi) [7].

5.4. Longueur

Longueur d'une séquence c'est le nombre d'éléments qui la composent $|S| = n$, [7].

6. Format de séquence

Il existe de nombreux formats des séquences : plus d'une trentaine sont répertories et utilise. Nous citons quelque format :

6.1. Fasta format

Format qui permet de représenter un ou plusieurs séquences (nucléiques ou protéiques) [11]. Une ligne qui commence par le symbole ' > ' caractérise le début d'une nouvelle séquence. Le symbole ' > ' est suivi d'un identifiant de séquence et des commentaires éventuels [11]. Les lignes suivantes constituent la séquence (jusqu'à ce qu'une nouvelle ligne commence par ' > ' ou la fin de fichier) [11].

- Exemple :

> $FBtr$0302953 $type$ = $mRNA$; loc = 2R ; $name$ = CG42703 –

RARAHHICITAWTGWCGAASASTSACTWRTYUIHWS

RAHHICITAWTGWCGAASASTSACTWRTYUIHWI

6.2. MSF

Un format entrelacé qui est conçu pour simplifier la comparaison des séquences avec des longueurs similaires [11].

- Exemple :

```
   MSF:   96  Type: P    Check:  7038   ..

 Name: 1aab_ oo  Len:   96  Check:  4681  Weight:  10.0
 Name: 1j46_A oo  Len:   96  Check:  1914  Weight:  10.0
 Name: 1k99_A oo  Len:   96  Check:  8221  Weight:  10.0
 Name: 2lef_A oo  Len:   96  Check:  2222  Weight:  10.0

//

1aab_        ...GKGDPKK PRGKMSSYAF FVQTSREEHK KKHPDASVNF SEFSKKCSER
1j46_A       ......MQDR VKRPMNAFIV WSRDQRRKMA LENP..RMRN SEISKQLGYQ
1k99_A       MKKLKKHPDF PKKPLTPYFR FFMEKRAKYA KLHP..EMSN LDLTKILSKK
2lef_A       ........MH IKKPLNAFML YMKEMRANVV AEST..LKES AAINQILGRR


1aab_        WKTMSAKEKG KFEDMAKADK ARYEREMKTY IPPKGE.... ......
1j46_A       WKMLTEAEKW PFFQEAQKLQ AMHREKYPNY KYRPRRKAKM LPK...
1k99_A       YKELPEKKKM KYIQDFQREK QEFERNLARF REDHPDLIQN AKK...
2lef_A       WHALSREEQA KYYELARKER QLHMQLYPGW SARDNYGKKK KRKREK
```

6.3. Clustal

Similaire à MSF, mais comprend une ligne supplémentaire signifiant la qualité de l'alignement entre un ensemble de séquences. Produit et lu par les programmes $clustalx$ et $clustalw$ [11].

7. Les Banques de Données Biologiques

Les premières banques de données biologiques sont apparues au début des années 80 sous l'initiative de quelques équipes de recherches. Leur principale mission est de rendre publiques les séquences qui ont été déterminées.

Les données biologiques stockées dans ces banques sont des séquences primaires d'ADN, d'ARN et de protéines. Les données peuvent être soumises et consultées par l'intermédiaire du Web. Les séquences stockées dans ces banques sont obtenues de plusieurs manières différentes. Il y a celles isolées à partir d'une cellule, déduites à partir de la séquence nucléique par simple traduction (cas des séquences d'ARN ou protéines) ou encore par génie génétique.

Les données stockées doivent être consultées d'une manière significative (Figure I.12), et souvent le contenu de plusieurs banques de données doit être consulté simultanément et en corrélation les uns avec les autres. Des langages spéciaux ont été développés pour faciliter

cette tâche (tels que le système de récupération de séquence « SRS » et le système « Entrez»). Certaines bases de données fournissent la fonctionnalité d'accès aux séquences mais encore des liens vers d'autres bases de données et les résultats d'analyse déjà obtenus. Par exemple, SWISSPROT contient des séquences de protéine ainsi que des annotations décrivant la fonction d'une protéine. Des structures 3D des protéines sont stockées dans des bases de données spécifiques. On peut trouver des banques spécialisées pour le stockage des motifs. En outre, des bases de données de la littérature scientifique (telles que PUBMED, MEDLINE) fournissent des fonctionnalités additionnelles, par exemple elles peuvent rechercher les articles scientifiques semblables basés sur l ' utilisation de la reconnaissance des mots. Ils ont développé des systèmes d'identification des textes qui extraient automatiquement l'information concernant un sujet tel que la fonction d'une protéine à partir des résumés des articles scientifiques [6].

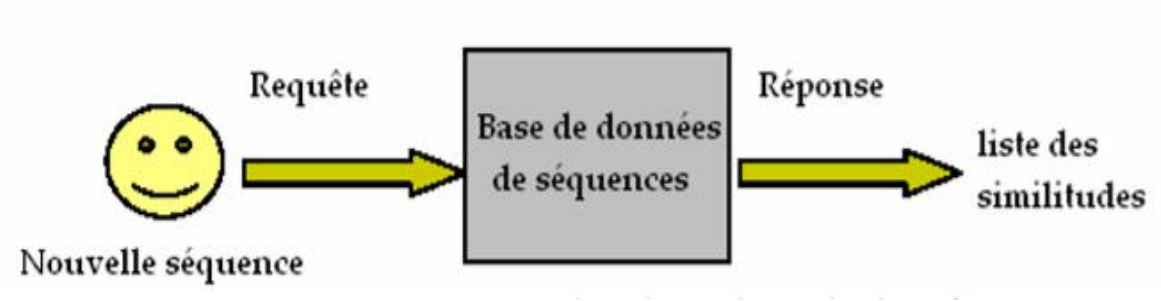


Figure 1.11 : Interrogation d'une base de données

7.1. Les Banques de Séquences Nucléiques

Nous citons les banques les plus populaires malgré que l'accès soit toujours contrôlé via des mots de passe :

- **EMBL :** banque européenne créée en 1980 et financée par l'EMBO (European Moleculary Biology Organization), elle est aujourd'hui diffusée par l'EBI (European Bioinformatics Institute, Cambridge, UK) [6].
- **GenBank :** créée en 1982 par la société IntelliGenetics et diffusée maintenant par le NCBI (National Center for Biotechnology Information, Los Alamos, US). Elle est soutenue par le NIH (National Institute of Health). Elle possède plus de 50 millions séquences stockées [6].
- **DDBJ (Dna Data Bank) :** créée en 1986 et diffusée par le NIG (National Institute of Genetics, Japon) [6].

La collaboration entre les deux premières banques a commencé relativement tôt. Elle s'est étendue en 1987 avec la participation de la DDBJ. Ils ont adopté un système de conventions communes : "The DDBJ/EMBL/GenBank Feature Table Definition" en 1990 qui a défini un format unique pour la description des caractéristiques biologiques qui accompagnent les séquences dans les banques de données nucléiques [6].

7.2. Les Banques de Séquences Protéiques :

PDB : La banque de données sur les protéines du " Research Collaboratory for Structural Bioinformatics", plus communément appelée Protein Data Bank ou PDB est une collection mondiale de données sur la structure tridimensionnelle (ou structure 3D) de macromolécules biologiques : protéines, essentiellement, et acides nucléiques. Ces structures sont essentiellement déterminées par cristallographie aux rayons X ou par spectroscopie RMN. Ces données expérimentales sont déposées dans la PDB par des biologistes et des biochimistes du monde entier et appartiennent au domaine public [9]. Leur consultation est gratuite et peut se faire directement depuis les sites web de la banque :

- Europe : PDBe ;
- Japon : PDBj ;
- États-Unis : RCSB PDB.

La PDB est la principale source de données de biologie structurale et permet en particulier d'accéder à des structures 3D de protéines d'intérêt pharmaceutique [9].

PIR-NBRF : créée en 1984 par la NBRF (National Biomedical Research Foundation). Elle est maintenant un ensemble de données issues du MIPS (Martinsried Institute for Protein Sequences, Munich, Allemagne) et de la banque japonaise JIPID (Japan International Protein Information Database) [6].

SwissProt : créée en 1986 à l'Université de Genève et maintenue depuis 1987 dans le cadre d'une collaboration, entre cette université (via ExPASy, Expert Protein Analysis System) et l'EBI. Celle-ci regroupe aussi des séquences annotées de la banque PIRNBRF ainsi que des séquences codantes, traduites de l'EMBL [6].

7.3. Les Banques de Motif

Prosite : La base de données dédiée aux stockages des motifs protéiques ayant une signification biologique peut être considérée comme un dictionnaire de motifs [6].

Les bases de ce type ont pour mission le recensement dans des catalogues les séquences des différents motifs pour lesquels une activité biologique a été identifiée [6].

8. Conclusion

Dans ce chapitre nous avons présenté une introduction sur la bio-informatique. Alors que l'informatique est devenue un apport fondamental à la biologie moléculaire. Les moyens informatiques sont naturellement utilisés pour le stockage ou la gestion des données mais également pour l'interprétation de ces données. Le traitement informatique des séquences peut par exemple déterminer la fonction biologique d'un gène [1]. Cet apport informatique concerne principalement quatre aspects :

• Le premier est l'organisation des données avec essentiellement la création de bases de données afin de réunir le plus d'information possible sur les séquences [1].

• Le deuxième aspect concerne les traitements que l'on peut effectuer sur les séquences afin de repérer un élément biologique intéressant. Ces programmes représentent les traitements couramment utilisés dans l'analyse des séquences comme la recherche des similitudes d'une séquence avec l'ensemble d'une base de données [1].

• Le troisième aspect est celui qui permet d'élaborer des stratégies pour apporter des connaissances biologiques supplémentaires que l'on pourra ensuite intégrer dans des traitements standards [1]. Par exemple la mise au point de nouvelles matrices de substitution des acides aminés, etc.…

• Enfin, le quatrième aspect est celui de l'évaluation des différentes approches citées précédemment dans le but de valider [1].

CHAPITRE 2 :

Alignement des séquences et arbre phylogénétique

1. Introduction

En bio-informatique, l'alignement de séquences (ou alignement séquentiel) est une manière de représenter deux ou plusieurs séquences de macromolécules biologiques (ADN, ARN ou protéines) les unes sous les autres, de manière à en faire ressortir les régions homologues ou similaires. L'objectif de l'alignement est de disposer les composants (nucléotides ou acides aminés) pour identifier les zones de concordance. Ces alignements sont réalisés par des programmes informatiques dont l'objectif est de maximiser le nombre de coïncidences entre nucléotides ou acides aminés dans les différentes séquences. Ceci nécessite en général l'introduction de « trous » à certaines positions dans les séquences, de manière à aligner les caractères communs sur des colonnes successives. Ces trous correspondent à des insertions ou des délétions (appelés indel) de nucléotides ou d'acides aminés dans les séquences biologiques. Le résultat final est traditionnellement représenté comme des lignes d'une matrice.

2. Alignement de séquences

Alignement de séquences d'ADN (ou d'acides aminés): opération de base en bio-informatique qui a pour but d'identifier des zones conservées entre séquences.

CAGCA-CTTGGATTCT-GG

CAGC- - -TTG- -TACTCGG

2.1. Utilité de l'alignement :

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

2.2. Type d'alignements :

On distingue 2 types d'alignements qui diffèrent suivant leur complexité :

- L'alignement par paires : consiste à aligner 2 séquences peut être réalisé grâce à un algorithme de complexité polynomiale. Il est possible de réaliser un alignement :
  - Global, c'est à dire entre les 2 séquences sur toutes leurs longueurs
  - Local entre une séquence et une partie de l'autre séquence
- L'alignement multiple, qui est un alignement global : consiste à aligner plus de 2 séquences et nécessite un temps de calcul et un espace de stockage exponentiel en fonction de la taille des données.

3. L'alignement par paires

3.1. Les définitions :

Définition 1:

L'alignement par paire de séquences de protéines est un outil fondamental de la bio-informatique. Il a pour but principal de faire ressortir les séquences apparentées, en mettant en évidence les régions communes. L'alignement par paire est principalement utilisé pour la comparaison d'une séquence avec un ensemble de séquences. Les algorithmes F$asta$ et Blast permettent de comparer une séquence à un ensemble de séquences contenues dans une base de données. Une séquence peut être définie comme une suite finie et ordonnée de lettres prises dans un alphabet Σ. Pour les protéines, cet alphabet est constitué de 20 lettres, appelées acides aminés ou résidus [10].

Définition 2 :

Soient $S_1 = (x_{11}, x_{12}, \dots x_{1|S_1|})$ et $S_2 = (x_{21}, x_{22}, \dots x_{2|S_2|})$ 2 séquences définies sur un alphabet Σ [10]. Un alignement A de S1 et S2 est une matrice de caractères de Σ ∪ {−} définie par [10] :

$$A = \begin{bmatrix} a_{11}, a_{12}, \dots, a_{1q} \\ a_{21}, a_{22}, \dots, a_{2q} \end{bmatrix}$$

Et vérifiant les propriétés [10] :

$$Max(|S_1|, |S_2|) \leq q \leq |S_1| + |S_2|,$$

$$a_{ui} = x_{uv}\ ou -, \forall u \in \{1,2\}, \forall v \in [1..|S_u|]$$

$$\nexists i\ tel\ que\ a_{1i} = a_{2i} = -.$$

Définition 3 :

Soient S1 et S2 deux séquences de longueurs respectives m et n définies sur un alphabet Σ

Soient *q* ***(****i, j****)*** le problème consistant à aligner les deux sous-séquences S1 *[1. . i] et* S2 *[1..j]*.

i ≤ *m et j* ≤ *n et D* (i, *j*), la distance d'édition associée [10].

Le problème consiste à déterminer *P* (*m*, *n*).

Le sous-problème *P* (i, *j*) consiste à calculer *D* (*i, j*).

L'initialisation sa fait avec *P* (*i,* 0) *et P* (0*, j*).

Les problèmes *P (i, 0) et* P (*0*, *J*) représentent le décalage d'une des séquences par rapport à l'autre, et leurs coûts sont simples à calculer [10].

-Exemple d'alignement de deux séquences :

Figure 2.1 : Alignement de deux séquences protéiques [6]

3.2. Évaluation d'un Alignement

Cependant, il est clair que pour deux séquences données quelconques il y a plusieurs alignements possibles. Il est devenu alors nécessaire de pouvoir déterminer quel est le meilleur alignement ou plutôt l'optimal si possible. Évaluer un alignement revient alors à mesurer sa qualité en déterminant la distance qui sépare les deux séquences. Le score d'un alignement est la somme des scores de toutes les positions de bases (résidus) prises deux à deux [6].

-Exemple d'évaluation :

On peut attribuer une valeur positive à des symboles alignés identiques et une pénalité (valeur négative) à une substitution ou à un gap [6].

Si l'on considère l'exemple précédent :

- Score (identité) =2
- Score (substitution) = -1
- Score (gap) = -2

Le score de cet alignement serait alors :

Figure 2.2 : Score d'un alignement [6]

Pour évaluer un alignement, le poids de chaque paire de résidus (identité ou substitution) dépend de la nature des résidus mis en correspondance [6]. Le calcul de score d'un alignement de deux séquences A et B de longueur équivalente L est alors :

$$Score\ (A, B) = \sum SC\ (A_i, B_j)$$

3.3. Distance et similarité entre deux séquences

3.3.1. Similarité et homologie

La similarité entre deux séquences peut être expliquée en prenant comme postulat de départ que toutes les espèces vivantes sont issues d'un même ancêtre. Selon cette théorie, des mutations interviennent au niveau de protéine, générant ainsi de nouvelles espèces. Ces mutations se produisent localement et peuvent être de différents types [7] :

- Suppression d'un ou plusieurs acides aminés,
- Insertion d'un ou plusieurs acides aminés,
- Mutation d'un acide aminé en un autre.

Au sens de la théorie de l'évolution, deux séquences peuvent donc être plus ou moins proches, selon le nombre de modifications ayant eu lieu [7].

- Définition :

On dit qu'il y a homologie entre deux séquences lorsque celles-ci possèdent une parenté du point de vue de l'évolution. On dit qu'il y a similarité de séquences, lorsqu'il y a de nombreuses identités entre les séquences. Pour les protéines, cela se caractérise par des paires de résidus composées de deux acides aminés appartenant à la même famille physicochimique [7].

3.3.2. La distance de *Hamming*

Soient *S* et *T* deux séquences de même longueur n sur un alphabet $\sum$. La distance de H*a*mming entre *S* et *T*, notée *dH(S, T),* représente le nombre de caractères *S[i]* et *T[i]* qui ne se correspondent pas [7]. La distance de H*a*ming g est définie par :

$$dH(S,T) = |i \in \frac{[1..n]}{S[i]} \neq T[i]|$$

Cette formule compte le nombre de positions ou les lettres ne se correspondent pas [7]

3.3.3. Le Système de Score

Un système de score est le coût à attribuer aux opérations élémentaires (identité, substitution, délétion et insertion) de comparaisons de séquences. En général, on a besoin donc [6] :

- Des systèmes de scores qui soient « biologiquement pertinent ».
- Des matrices de substitution et donc des scores individuels *SC* (A_i,B_j), dont le choix dépend de la relation recherchée entre les deux séquences :
  - Relation structurelle (propriétés physico chimiques).
  - Relation d'homologie (évolution moléculaire).

3.3.4. Les Matrices de Substitution

Le choix d'une matrice de substitution gouverne le système des scores et par conséquent influe sur les résultats obtenus. Il existe deux types de matrices de substitution à utiliser selon la nature des séquences nucléiques ou protéiques [12].

3.3.4.1. Matrices de Scores pour l'ADN

- **La matrice Identité :** Cette matrice consiste en l'attribution d'un score 1 en cas d'identité sinon un zéro.
- **La matrice de Transition/Transversion :** Dans cette matrice en prend en considération l'effet des actions des transitions (A à G, G à A, C à T, et T à C) et Transversion (les autres passages entre nucléotides) Identité=3Transition= 1, Transversion = 0.
- **La matrice BLAST :** La matrice identité Blast. C'est une matrice de même principe que la matrice Identité sauf que les valeurs attrib-uées en cas d'identité et substitution sont différentes de 1 et 0. On Remarque que la substitution ici est fortement pénalisée [12].

3.3.4.2. Matrices de score pour les protéines

Deux grandes familles de matrices.

- **Matrices PAM** : Les matrices PAM pour ≪ Percent Accepted Mutation / Accepted Point Mutation ≫, sont construites par étude de segments pris dans des séquences protéiques homologues (moins de 15% de différences) [13]. PAM x : x % de mutations acceptées entre les séquences qui ont servi à construire la matrice. Les fréquences de substitutions observées (ou probabilité conditionnelle : appelée "odd") sont transformées en logarithme de probabilité, normalisé en unité d'évolution. Le logarithme est utilisé pour que dans les programmes de recherche de ressemblance, la somme de ces éléments donne le logarithme de la probabilité pour la séquence entière (le modèle étant Markovien : indépendance de fréquences de substitution) [13].

  Les éléments diagonaux de la matrice indiquent un Évolution sans substitution. Pour PAM1, leur somme est telle qu'elle correspond à une probabilité de 99/100 (1 mutation pour 100 résidus : d'où le nom PAM : accepte point mutation) [13]. L'indépendance des fréquences et les éléments de la matrice étant des logarithmes de fréquences, on peut calculer PAM (N) en élevant PAM1 à la puissance N, par exemple : pour PAM120, il faut multiplier PAM1 par elle-même 120 fois [13].

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

Tableau 2.1 : Matrice PAM 250

Remarque : la valeur SC $(Xi,)$ sera :

- S = 0 : les probabilités observées et attendues sont identiques.
- S < 0 : les probabilités observées sont inférieures aux attendues.
- S > 0 : les probabilités observées sont supérieures aux attendues.

- **Matrice BLOSUM :** Ces matrices BLOSUM (Blocks Substitutions Matrices) sont construites par analyse de séquences de protéines. Les séquences sont découpées en blocs (2000résidus au total) par rapport au pourcentage d'acides aminés inchangés [14]. BLOSUM x : matrice obtenue à partir de séquences présentant au minimum x % d'identité (similitude) entre elles [14]. Une matrice "odds" est calculée à partir des blocs d'alignement pour chaque valeur de similitude, et ensuite chaque élément est transformé en unité d'information en prenant le logarithme du rapport de la valeur observée à la valeur qu'on obtiendrait au hasard. Cette matrice est ensuite normalisée [14]. Les correspondances entre BLOSUM et PAM, basées sur la théorie de l'information sont :
  - $PAM250 \longrightarrow BLOSUM45$
  - $PAM160 \longrightarrow BLOSUM\ 62$
  - $PAM120 \longrightarrow BLOSUM\ 80$

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

Tableau 2.2 : Matrice BLOSUM 62

3.3.5. Evaluation des brèches

L'évaluation des brèches dans un alignement est très importante. Selon les valeurs attribuées, le nombre des brèches peut varier. Si elles ne sont pas assez pénalisantes, la fonction de score risque de favoriser les alignements qui en contiennent beaucoup. En particulier, en morcellent les séquences il est possible d'augmenter le nombre de correspondances [15].

Inversement, si les brèches sont trop fortement évaluées, les alignements ne comportant pas assez des brèches sont favorises, empêchant ainsi d'avoir toutes les correspondances. Il existe principalement deux modélisations pour évaluer le cout engendré par l'insertion d'une brèche. Les valeurs associées à une brèche pour ces modelés peuvent être obtenues par des fonctions. Ces fonctions prennent en paramètre la longueur de la brèche et retourne le cout de celle-ci [15].

3.4. Principe de la programmation dynamique

La programmation dynamique est un principe souvent simple à mettre œuvre pour résoudre des problèmes complexes. Elle ne s'applique toutefois qu`a une certaine catégorie de problèmes, et il est nécessaire de vérifier certaines conditions pour qu'elle puisse être appliquée.

3.4.1. Problème et sous-problème

Soit *P(n)* un problème d'optimisation de taille *n*. On appelle sous-problème de *P(n)* tout problème *P(n)* avec *i* < *n*.

3.4.2. Principe d'optimalité

On dit qu'un problème *P(n)* satisfait au principe d'optimalité lorsqu'une solution optimale peut être exprimée en fonction de solutions optimales de sous-problèmes [7].

3.4.3. Programmation dynamique

Soit *P(n)* un problème satisfaisant au principe d'optimalité. On dit qu'un algorithme de résolution de *P(n)* est basé sur le principe de la programmation dynamique s'il utilise les deux étapes suivantes :

- *P(n)* est calculé récursivement en partant des problèmes de plus bas niveau.
- Une table est construite dynamiquement pour conserver tous les résultats intermédiaires obtenus [7].

L'utilisation de la programmation dynamique suppose donc que pour les valeurs les plus faibles de *n*, *P(n)* soit connu ou facile à calculer. En conservant tous les résultats intermédiaires dans une table on évite d'avoir à les recalculer de nombreuses fois. En effet, refaire ainsi les mêmes calculs à chaque itération est à l'origine de la complexité exponentielle des algorithmes "naïfs" [7].

3.5. Les Méthodes d'Alignement par paires

Il existe deux types d'alignements de séquences : global et local.

3.5.1. Alignement Global

Plusieurs méthodes ont été développées afin de réaliser un alignement global de deux séquences le plus correct que possible. Parmi ces méthodes et qui sont toujours utilisées on trouve des méthodes graphiques et autres qui utilisent la programmation dynamique [6].

- **Algorithme Needleman & Wunsch**

Basé sur la programmation dynamique (la récursivité), cet algorithme ne calcule pas la différence entre deux séquences mais la similarité. Considérons deux séquences *A (1, n) B (1, m)* [6].

Un tableau à deux dimensions est rempli ligne après ligne (en partant de la dernière) et pour chaque ligne, colonne après colonne (en partant de la dernière) en obéissant à la règle suivante [6] :

Le score *S (i, j)* est le nombre maximum de correspondance entre les deux parties de séquences *A (i, n)* et *B (j, m)* (en prenant tous les chemins possibles à partir de *(i, j)* et en appliquant une fonction de score [6] :

- Score pour une identité =1.
- Score pour une substitution, une insertion ou délétion =0.

La formule de récurrence est [6] :

$$S(i,j) = \max \begin{cases} \text{Si } ai = bj+1 \quad S(i, j+1) -1 + s(ai, bj), \quad \text{sinon } S(i, j+1) + s(ai, bj) \\ \text{Si } ai+1 = bj+1 \quad S(i+1,j+1) -1 + s(ai, bj), \quad \text{sinon } S(i+1,j+1) + s(ai, bj) \\ \text{Si } ai+1 = bj \quad S(i+1,j) -1 + s(ai, bj), \quad \text{sinon } S(i+1, j) + s(ai, bj) \\ \text{Avec évidemment} \quad S(n+1, j) = S(i, m+1) = 0 \end{cases}$$

La similarité entre les deux séquences est égal à la valeur de *S*(1,1) et l'alignement est un graphe qui a pour origine *S*(1,1) et parcourt la matrice pour des i et j croissant en recherchant l'élément maximal voisin [6].

3.5.2. Alignement Local

Ce type d'alignement est favorable aux séquences divergentes car un alignement global serait non significatif. Pour l'alignement Local, Smith et Waterman une méthode exacte qui permet d'aligner deux séquences en essayant d'aligner des segments communs ou motifs [6].

- **L'algorithme Smith &Waterman**

L'algorithme de Smith et Waterman est décrit pour l'alignement local de deux séquences. Il identifie les sous séquences maximales de deux séquences par programmation dynamique [6].

La différence essentielle de cet algorithme avec l'algorithme de **Needleman et Wunsch** est que n'importe quelle case de la matrice de comparaison (initiale) peut être considérée comme point de départ pour le calcul des scores sommes et que tout score somme qui devient inférieur à zéro stoppe la progression du calcul des scores sommes et il sera réinitialisé par la valeur 0. La case concernée peut être considérée comme nouveau point de départ. Cela implique que le système de score choisi possède des scores négatifs pour les mauvaises associations qui peuvent exister entre les éléments des séquences [6].

L'équation utilisée pour le calcul de chaque score somme pendant la transformation de la matrice initiale prend alors l'expression suivante [6] :

$$S(i,j) = \max \begin{cases} Sc(i,j) + S(i+1, j+1) \\ Sc(i,j) + \max S(x, j+1) - P \\ Sc(i,j) + \max S(i+1, y) - P \\ 0 \qquad \text{avec } i+2 < x < m \text{ et } j+2 < y < n \end{cases}$$

Où *S (i, j)* est le score somme de la case d'indice *i* et *j*, Sc le score élémentaire de la case d'indice *i* et *j* de la matrice initiale issu d'une matrice de substitution et *P* la pénalité donnée pour une insertion [6].

3.6. Comparaison avec les Banques de Séquences

Lorsque l'alignement d'une séquence est réalisé contre une banque, le problème du temps de calcul devient prédominant. Les algorithmes précédents sont trop gourmands en ressource.

La façon de voir la ressemblance a été posée d'une manière différente pour contourner cet obstacle et des heuristiques sont été proposées [6].

Les méthodes d'alignement par paires sont utilisées pour comparer des séquences deux à deux. Elles sont utilisées pour rechercher une homologie entre une séquence test et une séquence de référence, souvent extraite d'une base de données. Elles sont les plus simples à mettre en œuvre, et ce sont les seules pour lesquelles il existe des solutions algorithmiques optimales, basées sur la programmation dynamique. Il existe également des méthodes heuristiques rapides, qui permettent d'effectuer des recherches systématiques dans les banques de séquence. Dans ce cas, on compare une séquence inconnue à toutes les séquences de la base, en les testant successivement une par une. Les méthodes les plus connues sont :

L'algorithme de Needleman-Wunsch qui réalise l'alignement global optimal entre deux séquences,

- **L'algorithme de Smith-Waterman** qui réalise un alignement local optimal.

4. Alignement multiple des séquences

Le problème d'alignement multiple correspond à une généralisation de l'alignement par paire avec *K > 2* séquences. Il ne s'agit plus en revanche ici de détecter une simple similitude entre séquences. L'alignement multiple de séquences est utilisé pour différentes opérations.

Il permet de déterminer des sous-groupes de séquences en fonction du degré de similarité. Ce qui constitue le point de départ pour la reconstruction de phylogénie [10].

L'alignement multiple permet également de mettre en évidence les zones conservées dans un ensemble de séquences. En partant du principe que des motifs similaires induisent des fonctions identiques, l'alignement multiple permet de prédire la fonction de protéines inconnues en les alignant avec des protéines connues [10].

4.1. Définition

Soit S = {S_1, . . ., S_K} un ensemble de séquences définies sur un alphabet Σ. Un alignement multiple d'une matrice d'éléments de $\Sigma \cup \{-\}$ définie par [10] :

$$A=\begin{bmatrix} a_{11}, a_{12} & \cdots & a_{1q} \\ \vdots & \ddots & \vdots \\ a_{k1}, a_{k2} & \cdots & a_{kq} \end{bmatrix}$$

Et vérifiant les propriétés :

- $\mathrm{Max}_i(|S_i|) \leq q \leq X_i|S_i|$,
- $a_{ui} = x_{uv}\, ou -, \forall u \in \{1..k\}, \forall v \in [1..|S_u|]$,
- $\nexists j\ tel\ que\ \forall i, a_{ij} = -$.

4.2. Les utilisations en bio-informatique

L'alignement multiple de séquences permet de mettre en évidence les similarités entre plusieurs séquences. Il est donc possible de comparer simultanément la proximité de toutes ces séquences. Les informations apportées par ces comparaisons permettent d'obtenir des renseignements importants sur les séquences comme les distances d'une séquence par rapport aux autres ou encore la mise en évidence de zones identiques entre plusieurs ou toutes les séquences [7].

Or ces opérations sont très employées en bio-informatique pour la résolution de plusieurs problèmes. L'alignement multiple de séquence est donc principalement utilisé comme opération préalable pour ces déférents problèmes. Citons par exemple la construction de phylogénie, la prédiction de structure 3D, la détermination de fonction des protéines [7].

4.3. Évaluation d'un alignement multiple de séquence

Le critère d'évaluation utilisé est souvent une fonction de score. Soit f une fonction permettant de définir la qualité d'un alignement multiple. *f* est une fonction définie sur Σ'² 2et à valeurs dans *R*. Nous donnons dans la section suivante quelques exemples des fonctions les plus utilisées [7].

Le nombre d'alignements multiples possibles pour l'ensemble des séquences de *S* est très important. Il est donc nécessaire de pouvoir évaluer la qualité de chaque solution. Quelques fonctions ont été définies afin de permettre une évaluation des alignements, mais elles peuvent également être utilisées par des algorithmes lors de la construction de ces alignements [7].

4.4. Les approches d'Alignements Multiple de Séquences

4.4.1. L'Approche Exacte

L'approche exacte n'est autre qu'une généralisation des méthodes de programmation dynamique de La méthode de programmation dynamique utilisée pour aligner deux séquences, a été appliquée à l'alignement de plusieurs séquences (N dimensions) tels que MSA [6] et DCA [41].

Ce type de méthodes représente de gros problèmes : Le temps de calcul et l'espace mémoire [19].

- Dans la pratique, un alignement devient délicat pour un nombre de séquence $N > 3$, et même impossible pour $N = 10$.
- Pour N séquences de longueur L, l'alignement optimal (au sens mathématique) nécessite :
  - Un temps de calcul proportionnel à $2^n L^n$
  - Un espace mémoire proportionnel à L^n

4.4.2. L'Approche Itérative

L'approche itérative a été employée plusieurs fois comme méthode d'optimisation pour produire des alignements multiples. Parfois elle est utilisée seule ou en combinaison avec

d'autres méthodes. L'itération a un grand avantage parce qu'elle est souvent très simple soit en termes de code Des algorithmes soit en termes de complexité temporelle et spatiale [19].

4.4.3. L'Approche Progressive

L'alignement progressif est l'heuristique la plus répandue pour aligner un grand nombre de séquences. L'alignement multiple est construit progressivement en alignant des paires de séquences suivies des paires d'alignements/profils. Un arbre guide détermine l'ordre dans lequel les séquences vont être alignées, les plus proches d'abord. Cette technique est employée dans différents packages d'alignement multiple tels que, ClustalW, et T-Coffee …etc. Un alignement multiple progressif suit les étapes suivantes [20] :

Un alignement multiple progressif suit les étapes suivantes [20] :

- Alignement deux à deux de toutes les séquences.
- Construction d'une matrice de distances entre toutes les séquences.
- Détermination de l'ordre selon lequel les séquences seront alignées en utilisant la notion de clustering :
  - Alignement de deux séquences.
  - Alignement d'une séquence et d'un profil.
  - Alignement de deux profils.

Problèmes majeurs des alignements multiples progressifs [20] :

- Les alignements entre sous-groupes sont gelés. Si une erreur est produite au début, aucune modification ou correction ultérieure n'est possible.
- Les erreurs dans les alignements des sous-groupes initiaux se propagent dans tous l'alignement.

4.5. Classification des Méthodes

Les algorithmes d'alignement multiples de séquences sont très nombreux, ils peuvent prendre des formes très différentes et être basés sur des principes très différents. Il est toutefois possible de les regrouper selon trois classes [36] en fonction de critères assez simples. Il existe en effet quelques algorithmes exacts permettant de réaliser des alignements d'un petit nombre de séquences. Les algorithmes approchés sont bien sur beaucoup plus nombreux, et ils peuvent être également subdivisés en deux catégories : les algorithmes progressifs et les algorithmes itératifs. Les algorithmes progressifs ont tous un point commun lié au processus d'alignement. Les séquences sont alignées progressivement en suivant un ordre défini, seule la façon dont vont être alignés les groupes de séquences va différer. Pour les algorithmes itératifs toutes les séquences sont alignées simultanément et les méthodes sont en revanche très différentes les unes des autres. Il est possible de décomposer à nouveau, suivant qu'il s'agit ou non d'algorithmes stochastiques [7].

4.5.1. L'approche exacte

Le problème d'alignement multiple de séquences est possible de généraliser la méthode basée sur la programmation dynamique. Sa forte complexité spatiale la rend inutilisable en

pratique pour la plupart des problèmes d'alignement. Comme nous le montrerons plus loin, à partir de 4 séquences, la quantité de mémoire nécessaire pour réaliser les calculs est bien souvent supérieure à ce qui est disponible dans la plupart des ordinateurs actuels. Aligner 5 séquences est considéré comme étant un petit problème, il est pourtant totalement Exclu de chercher à le résoudre de façon exacte par la méthode la programmation Dynamique que nous avons exposée. Nous présentons donc également ici deux algorithmes Basés sur la programmation dynamique et qui utilisent chacun une heuristique. Le premier Peut aligner une dizaine de séquences, et le second une vingtaine [7].

4.5.1.1. Méthode basée sur la programmation dynamique

Nous avons vu comment aligner deux séquences en utilisant programmation dynamique[7]. Nous allons montrer maintenant que cet algorithme peut être étendu à plus de deux séquences.

Cas de 3 séquences :

Le cas de 3 séquences est le plus simple pour expliquer comment réaliser un alignement multiple en utilisant le principe de la programmation dynamique. L'explication donnée ici et que nous allons principalement détailler concerne le cas le plus simple, à savoir l'alignement avec brèches à cout constant. Le problème consiste ici à aligner 3 séquences *S*, *T* et *U* de longueurs respectives *p, q* et *r*. En faisant le parallèle avec le cas de 2 séquences, nous appellerons *P (i, j, k)* le problème consistant à aligner les 3 sous séquences *S* [1 . . . *i*], *T* [1. . . *J*] et *U* [1. . . *K*], et la valeur associée à ce problème sera notée *V (i, j, K)*. Aligner *S, T* et *U* consiste donc à déterminer *P (p, q, r)* et la valeur de cet alignement est *V (p, q, r).* Le principe de l'algorithme est le même que pour due séquences il suffit de chercher à déterminer la dernière position de l'alignement. Plusieurs cas sont alors possibles, selon qu'il y a une, deux ou trois lettres [7].

- La dernière position de l'alignement constituée d'une unique lettre peut se produire de trois façons différentes, selon la séquence où se trouve la lettre [7].
- La dernière position de l'alignement constituée de deux lettres peut se produire de trois façons différentes, selon la séquence où se trouve la brèche [7].
- La dernière position de l'alignement constituée de trois lettres correspond à un cas unique. Il y a donc au total 7 possibilités pour la dernière position de l'alignement. Ces 7 possibilités correspondent aux 7 sous-problèmes directs de *P (p, q, r)*. La valeur de *V (p, q, r)* Comme pour le cas de l'alignement de deux séquences, le principe de la programmation dynamique peut être utilisé pour résoudre le problème. Toutefois dans ce cas il n'est plus possible d'utiliser une matrice en deux dimensions puisqu'il faut conserver toutes les valeurs *V (i, j, K).* Il est donc nécessaire d'utiliser un cube de taille (p + 1) × (q+ 1) × (r + 1). Les cas de base correspondent aux problèmes où une au moins des valeurs *i*, *j* ou k est égale à 0 [7]. Le calcul des cas de base peut se ramener aux trois cas dégénérés suivants :
    - *i* = 0, ce qui correspond à réaliser l'alignement des deux séquences *T* et U,

- j = 0, ce qui correspond à réaliser l'alignement des deux séquences *S* et *U*,
- k= 0, ce qui correspond à réaliser l'alignement des deux séquences S et T,

4.5.1.2. La méthode MSA

Nous venons de voir que la programmation dynamique pouvait théoriquement être utilisée pour aligner un nombre quelconque de séquences. Toutefois en pratique l'algorithme ne peut pas être utilisé en raison de la mémoire qu'il requiert. Cas de 2 séquences L'algorithme *MSA* propose une heuristique basée sur l'algorithme complet de Needleman-Wunsch. Le principe est simple, et il est facile à comprendre sur un alignement de deux séquences. À savoir que le chemin permettant la construction de l'alignement des deux séquences se situe à proximité de la diagonale. En fonction de la similarité entre les séquences à aligner, il est possible de déterminer avant d'utiliser l'algorithme de programmation dans quelle partie se trouvera l'alignement. Plus la similarité est forte, et plus la zone centrale peut être réduite. Les zones hachurées marquées 1 et 2 correspondent aux valeurs qui n'ont pas besoin d'être calculées.

La méthode exposée pour 2 séquences est généralisable pour un nombre quelconque *n* de séquences. La zone de calculs est toujours une partie de la matrice centrée sur la diagonale issue de l'origine. La représentation graphique est toutefois difficile à réaliser, même pour *n*=3.Cette méthode basée sur une méthode exacte est une heuristique. Même s'il est fortement probable qu'elle donne une solution optimale, cela n'est aucunement garanti. Elle offre l'avantage de permettre d'augmenter la limite du nombre de séquences pouvant être alignées. Il est ainsi possible d'aligner jusqu'à une dizaine de séquences similaires [7].

4.5.1.3. La méthode DCA

L'algorithme *MSA* est basé sur une restriction de la zone de calculs autour de la diagonale de la matrice. Il est à la base d'un autre algorithme de type "divide-and-conquer" appelé *DCA* (Divide and Conquer mulitple sequence Alignment). L'intérêt d'un tel algorithme peut se comprendre facilement.

- **Principe de l'algorithme** :

Comme nous venons de le voir il est possible de gagner un facteur 1.000 sur la quantité de mémoire en divisant les séquences en 6 sous-séquences. Et il est bien entendu possible de diviser encore la longueur de chaque séquence pour obtenir autant de problèmes de plus petite taille. Toutefois l'algorithme *DCA* ne réalise pas uniquement un découpage des séquences, et ce pour une raison très simple. Un découpage aléatoire des séquences pose des problèmes d'alignement puisque l'on impose le début et la fin pour chacun d'eux. Le découpage ne doit donc pas être fait de cette façon. En se basant sur les alignements par paires, l'algorithme *DCA* détermine les points de découpe de chaque séquence pour former des groupes de sous-séquences à aligner. Une fois tous les alignements de ces groupes de séquences effectués, l'algorithme les assemble pour former le résultat. En réalisant

l'alignement des groupes de sous-séquences avec l'algorithme *MSA*, il est possible de réaliser des alignements de plus de 20 séquences. Nous constatons donc qu'avec deux heuristiques, il est possible d'utiliser la programmation dynamique pour obtenir des alignements comportant nettement plus de séquences. Rien ne garantit toutefois que l'on puisse calculer l'optimum[7].

4.5.2. L'approche progressive

L'Approche progressif [23] est l'heuristique la plus répandue pour aligner un grand nombre de séquences. L'alignement multiple est construit progressivement en alignant des paires de séquences suivies des paires d'alignements/profils. Un arbre guide détermine l'ordre dans lequel les séquences vont être alignées, les plus proches d'abord. Cette technique est employée dans différents packages d'alignement multiple tels que MULTALIGN [24], ClustalW [25], et T-Coffee [26] …etc.

Un alignement multiple progressif suit les étapes suivantes [20] :

- Alignement deux à deux de toutes les séquences.
- Construction d'une matrice de distances entre toutes les séquences.

Détermination de l'ordre selon lequel les séquences seront alignées en utilisant la notion de clustering [20] :

- Alignement de deux séquences.
- Alignement d'une séquence et d'un profil.
- Alignement de deux profils.

Problèmes majeurs des alignements multiples progressifs [20] :

- Les alignements entre sous-groupes sont gelés. Si une erreur est produite au début, aucune modification ou correction ultérieure n'est possible.
- Les erreurs dans les alignements des sous-groupes initiaux se propagent dans tous l'alignement.

Les algorithmes d'alignement multiple progressifs sont basés sur les informations obtenues d'un arbre guide construit au préalable. Cet arbre définit un certain rapprochement entre les séquences (homologie ou similitude). Puis on construit progressivement l'alignement multiple en respectant l'ordre défini par l'arbre. Vu le nombre de méthodes développées, cette approche paraît la plus utilisée malgré ses inconvénients [20].

4.5.2.1. Clustal

ClustalW est basé sur le principe de l'algorithme de Feng et Doolittle. Le principe général est le suivant [27] :

1. Calculer tous les alignements par paires des séquences, et en déduire une matrice des distances.

2. Construire un guide-tree à partir de la matrice des distances en utilisant la méthode du NeighbourJoining.
3. Utiliser le guide-tree afin de déterminer l'ordre dans lequel les séquences doivent être, Alignées :

   a) Choisir les séquences ou profils à aligner en suivant le guide-tree.

   b) Les aligner en utilisant une méthode basée sur la programmation dynamique.

   c) Créer un profil à partir du résultat de l'alignement.

   d) Si on n'est pas arrivé à la racine de l'arbre reprendre en 3.a.

4. Retourner l'alignement obtenu.

L'alignement de deux séquences et/ou profils dans ClustalW fait intervenir des paramètres supplémentaires par rapport à la méthode d'alignement par paire

- Ainsi, pour éviter le morcellement des séquences que peuvent créer plusieurs brèches trop rapprochées un paramètre est ajouté. Il s'agit d'une distance minimale devant séparer deux brèches, par défaut cette valeur est égale à 5. IE l reste bien entendu possible d'insérer deux brèches plus proches car cela peut être nécessaire, mais dans ce cas une pénalité est ajoutée au coût de la deuxième brèche.
- ClustalW prend également en compte les propriétés physico chimiques propres aux différents acides aminés. Ainsi si deux acides aminés sont normalement plus difficiles à séparer, un sur coût sera attribué pour l'insertion d'une brèche.
- ClustalW détermine les paramètres d'alignement à utiliser (coût d'ouverture et d'extension de brèches ainsi que matrice de substitution) en fonction des séquences à aligner. ClustalW permet d'aligner plus d'une centaine de séquences, et il donne de bons résultats dans la plupart des cas. De par sa construction, le guide-tree associé à n séquences nécessite de réaliser $n - 1$ alignements pour obtenir l'alignement de toutes les séquences. Le temps de calcul de ClustalW est donc pratiquement linéaire en fonction du nombre de séquences à aligner. Pendant près de vingt ans, ClustalW a été l'algorithme de référence pour l'alignement multiple, et il reste encore aujourd'hui très utilisé.

4.5.2.2. MUSCLE

La méthode *MUSCLE* emploie deux mesures de distance pour une paire des séquences : une distance de k-mer de (pour une paire non alignée) et le Kimura distance (pour une paire alignée). Un k-mer est une subséquence contiguë de longueur k également connu sous le nom de mot ou k-tuplet. Les séquences homogènes possèdent plus de k-mers en commun que prévu par hasard. Cette mesure n'exige pas un alignement, elle donne un avantage significatif de vitesse contrairement à Kimura [16].

La méthode *MUSCLE* peut être décrite en trois étapes essentielles [16] :

1. Le but de la première étape est de produire rapidement un alignement multiple avec plus d'exactitude possible. Ceci est basé sur la détermination d'une matrice D1 de distances partir de la distance de $K-mers$ entre toutes les paires de séquences.
2. La matrice obtenue est alors clustérisée par *UPGMA*, pour produire un arbre binaire *TREE1*. Un alignement progressif *MSA1* est construit alors en suivant l'ordre dicté par l'arbre. La source d'erreur principale à l'étape progressive est la mesure approximative de distances $K-mer$, qui a comme conséquence un arbre sous optimal. *MUSCLE* re-estime donc l'arbre en utilisant la distance de Kimura, qui est plus précise mais exige l'utilisation un alignement dans ce cas c'est *MSA1* donnant ma matrice D2. D2 va subir le même procédé de clustérisation afin de produire un arbre binaire *TREE2* et progressivement construire l'alignement *MSA2*.
3. C'est une étape d'amélioration. *TREE2* est divisé en deux sous arbres en supprimant la branche qui les relient. Celle-ci est choisie en parcourant l'arbre à partir de la racine. Le profil de l'alignement multiple dans chaque sous arbre est alors calculé. Un nouvel alignement multiple est produit en réalignant les deux profils. Si le score de *SP* est amélioré, le nouvel alignement est gardé, autrement il est rejeté et l'étape 3 est alors répétée jusqu' à la convergence ou jusqu'à ce qu'une limite définie soit atteinte.

Considérée la plus rapide et plus exacte, la méthode *MUSCLE* est la plus répandue actuellement avec ClustalW [16].

4.5.3. L'approche itérative

4.5.3.1. SAGA

SAGA (Sequence Alignment by Genetic Algorithm) [28] est base sur un algorithme de type génétique. Une population *G* d'alignements est initialement générée, et le programme permet de contrôler son évolution. La population initiale est créée en générant cent individus par insertion aléatoire de brèches. Le schéma général suivi par l'algorithme est le suivant [29] :

- Sélection de la partie de la population devant être remplacée ; par défaut cette valeur est de 50%.
- Utilisation de l'un des nombreux operateurs les individus sélectionnés,
- Mise à jour de la population en ne conservant que les cent meilleurs individus. Ces trois opérations permettent de passer de la génération G_n à la génération G_{n+1}. L'algorithme s'arrête lorsque la population se stabilise.

Il existe plusieurs méthodes et nous parlerons de l'algorithme " Clustal ".

5. La Phylogénie

La similitude entre des mécanismes moléculaires des organismes qui ont été fortement étudiées, suggère que tous les organismes sur terre ont eu un ancêtre commun. Ainsi toutes les espèces ont des liens de parentés et cette relation s'appelle une phylogénie. Habituellement le rapport peut être représenté par un arbre phylogénétique. Le rôle du phylogénétique est de construire cet arbre à partir des observations sur les organismes existants.

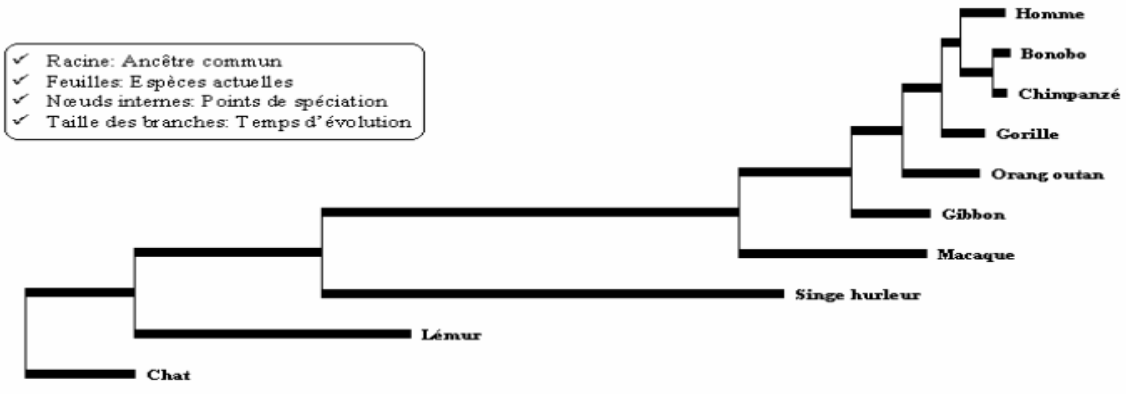


Figure2.3 : Arbre phylogénétique des espèces

Les recherches ont montré que les séquences moléculaires fournissent des ensembles de caractères communs qui peuvent porter une grande quantité d'information. Si l'on possède un ensemble de séquences d'espèces différentes, on est capable de les employer pour fonder une phylogénie probable de l'espèce en question. Ceci assume que les séquences sont descendues d'un certain gène ancêtre commun d'une espèce ancêtre commune.

Pour quoi faire ?

- Retracer l'histoire évolutive d'une famille de gènes.
- Reconstruire les relations évolutives entre espèces.
  - ex : arbre du vivant
- Classer une nouvelle espèce
  - ex : souche virale

Comment ?

- Aligner correctement les séquences nucléiques ou protéiques.
- Appliquer une méthode de génération d'arbres.
- Évaluer statistiquement la robustesse des arbres.

5.1. Méthodes de Reconstruction d'Arbres

Afin de déterminer les similitudes et liens entre éléments d'un arbre (en général des séquences), plusieurs méthodes ont été suivies telles que :

- La méthode de parcimonie essaye de trouver l'arbre le plus parcimonieux, c.à.d., celui qui explique le lien entre deux séquences avec le moins de mutations (substitutions/ insertions/délétions) possibles. Cette méthode est valable pour les séquences très poches.
- Les méthodes de distances commencent par calculer la distance entre les séquences et essaye de trouver l'arbre qui approche le mieux cette distance. Le calcul des distances peut tout simplement compter le nombre de mésappariements entre les deux séquences dans l'alignement ou utiliser un modèle stochastique tel que le modèle de Kimura, où la probabilité d'un changement dépend des bases (A<->G et C<->T sont plus fréquentes), on a deux probabilités. Donc les transitions et les Trans versions ont une probabilité différente.
- La méthode du Maximum de vraisemblance est basée sur un modèle probabiliste évolutif et elle cherche l'évolution la plus probable. Elle cherche à trouver le scénario pour lequel la probabilité d'obtenir actuellement les données observées est le plus grand possible. Cette probabilité s'appelle "Vraisemblance". En d'autres termes, on cherche la valeur qui maximise la probabilité d'observer les résultats effectivement observés. La proximité des séquences n'est pas importante.

Un véritable arbre phylogénique possède une racine, ou l'ancêtre terminal de tout es les séquences. Quelques algorithmes fournissent des informations au sujet de l'endroit de la racine. D'autres, basé parcimonie, sont non informatifs au sujet de sa position.

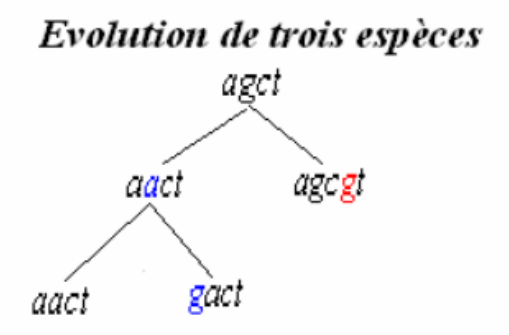


Figure2.4 : Accumulation des substitution/insertion

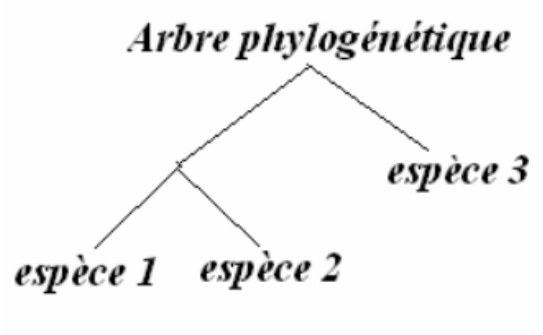


Figure2.5 : Apparition des espèces

Notions de bases (arbres)

Un arbre phylogénétique est caractérisé par :

- Sa topologie
- La longueur de ses branches (éventuellement)

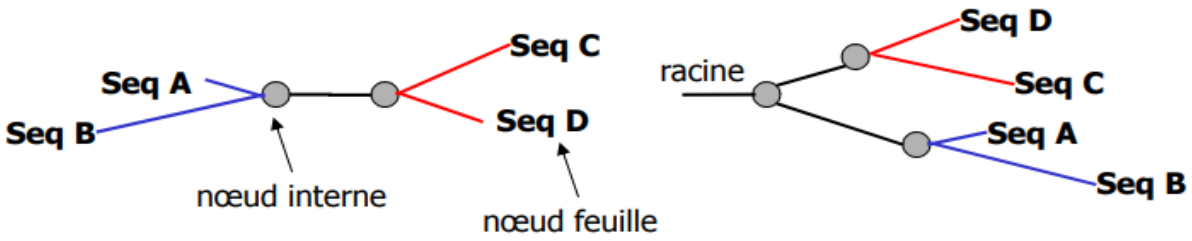


Figure2.6 : Accumulation des nœuds

- **Nœud :** estimation de l'ancêtre commun des éléments appartenant à ce nœud.
- **Racine (root) :** ancêtre commun de tous les éléments de l'arbre.

Un arbre peut avoir ou non une racine.

Notation de Newick

Pour stocker un arbre dans un fichier texte, on peut utiliser la notation :

((A, B), C)

On peut aussi ajouter la longueur de chaque branche :

((A:1, B:1):2, C:4)

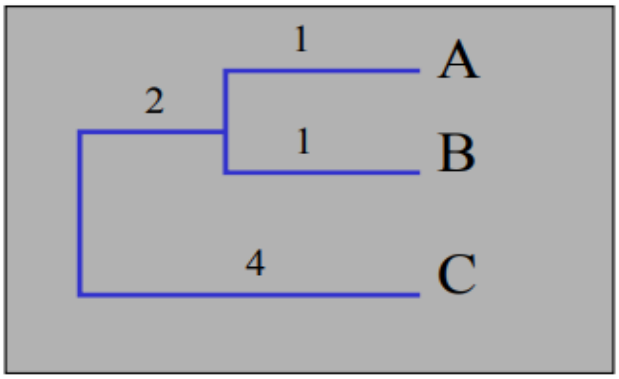


Figure2.7 : Notation de stockage d'un arbre

5.2. Unweighted Pair Group Method with Arithmetic Mean (UPGMA)

Cette méthode est utilisée pour reconstruire des arbres phylogénétiques si les séquences ne sont pas trop divergentes.

UPGMA emploie un algorithme de clustering qui utilise des moyennes arithmétiques. On procède par une clustérisation des séquences, à chaque fois que l'on fusionne deux clusters les plus proches, on crée un nouveau nœud sur l'arbre.

L'arbre peut être imaginé comme étant dirigé vers le haut, où chaque nœud est ajouté au-dessus des autres, et les longueurs des arcs sont déterminées par la différence dans les tailles des nœuds au-dessus et au bas d'un arc. UPGMA fournit un arbre sans racine.

D'abord on définit la distance *dij* entre deux clusters *Ci* et *Cj* c'est la moyenne distance entre les paires de séquences de chaque cluster :

$$\boldsymbol{d_{ij}} = \frac{1}{|C_i||C_j|} \sum_{p \in C_i, q \in C_j} \boldsymbol{d_{pq}}$$

Où $|C_i|$ et $|C_j|$ dénotent le nombre de séquences dans les clusters i et j respectivement.

Si $C_k = C_i \cup C_j$ et si C_l est n'importe quel autre cluster, alors :

$$\boldsymbol{d_{il}} = \frac{\boldsymbol{d_{il}}|C_i| + d_{jl}|C_j|}{|C_i| + |C_j|}$$

Exemple : Si on considère la matrice de distances associées à un groupe de 6 éléments et que l'on veuille obtenir l'arbre associé :

	A	B	C	D	E
B	2				
C	4	4			
D	6	6	6		
E	6	6	6	4	
F	8	8	8	8	8

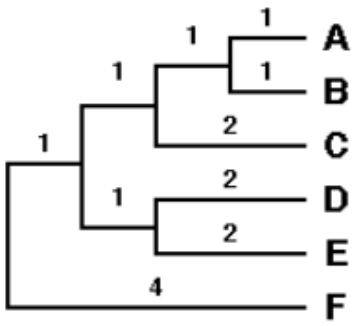


Figure 2.8 : Un arbre phylogénétique construit par la méthode UPGMA

5.3. Neighbor-Joining (NJ)

Elle est basée sur la recherche d'une paire d'OTU (operational taxonomic units : feuille de l'arbre) qui minimise la longueur totale des branches de l'arbre et ceci à chaque étape de regroupement.

Cette méthode développée par Saitou et Nei.[17] Elle tente de corriger la méthode UPGMA afin d'autoriser un taux de mutation différent sur les branches. Elle construit un arbre phylogénétique sans racine à partir d'un indice d'écart (par exemple distance ou dissimilitude entre séquences).

Les données initiales permettent de construire une matrice qui donne un arbre en étoile. Cette matrice de distances est ensuite corrigée afin de prendre en compte la divergence moyenne de chacune des séquences avec les autres. L'arbre est alors reconstruit en reliant les séquences les plus proches dans cette nouvelle matrice. Lorsque deux séquences sont liées, le nœud représentant leur ancêtre commun est ajouté à l'arbre tandis que les deux feuilles sont enlevées.

Ce processus convertit l'ancêtre commun en un nœud terminal dans un arbre de taille réduite.

Exemple :

	A	B	C	D	E
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

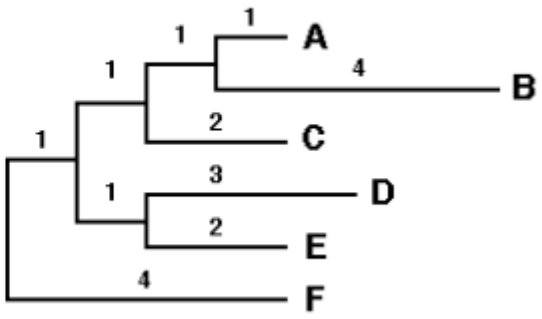


Figure 2.9 : Un arbre phylogénétique construit par la méthode NJ

6. Conclusion

Les méthodes d'alignement par paires dans ce chapitre sont utilisées pour comparer des séquences deux à deux. Elles sont utilisées pour rechercher une homologie entre une séquence test et une séquence de référence, souvent extraite d'une base de données. Elles sont les plus simples à mettre en œuvre, et ce sont les seules pour lesquelles il existe des solutions algorithmiques optimales, basées sur la programmation dynamique. Ensuite on a abordé les déférant approches de résolution de problème de l'alignement multiple des séquences ainsi que les méthodes de chaque approche enfin nous avons en particulier présenté la phylogénie qu'on va détailler dans le prochain chapitre.

Chapitre III :

Méthodes hiérarchique et phylogénétique

1. Introduction

Les données brutes, malgré leur quantité qui augmente d'une façon exponentielle, n'ont presque aucune valeur, ce qui est le plus important en fait c'est les connaissances pour lesquelles nous sommes tous assoiffés et qui sont obtenus par la compréhension de ces données, mais plus on a de données plus ce processus devient difficile. De nos jours, les changements de notre environnement sont dénotés par des capteurs qui sont devenus de plus en plus nombreux. Par conséquent, la compréhension de ces données est très importante.

2. Data Mining

2.1. Définition et historique

Selon le Groupe Gartner, le Data Mining appelé aussi fouille de données est le processus de découverte de nouvelles corrélations, modèles et tendances en analysant une grande quantité de données, en utilisant les techniques de reconnaissance des formes ainsi que d'autres techniques statistiques et mathématiques. Il est apparu au milieu des années 1990 aux Etats-Unis comme une nouvelle discipline à l'interface de la statistique et des technologies de l'information : bases de données, intelligence artificielle, apprentissage automatique [25].

Il existe d'autres définitions, nous citons quelques-unes :

- Le data mining est l'analyse de grands ensembles de données observationnelles pour découvrir des nouvelles relations entre elles et de les reformuler afin de les rendre plus utilisables de la part de leurs propriétaires [22]
- Le data mining est un domaine interdisciplinaire utilisant en même temps des techniques d'apprentissage automatique, de la reconnaissance des formes, des statistiques, des bases de données et de visualisation pour déterminer les manières d'extraction des informations de très grandes bases de données [24].
- Le data mining est l'extraction de connaissances à partir de grandes quantités de données. C'est un domaine relativement récent qui se situe à l'intersection des statistiques, de l'apprentissage automatique et des bases de données.
- Le Data Mining est un processus inductif, itératif et interactif dont l'objectif est la découverte de modèles de données valides, nouveaux, utiles et compréhensibles dans de larges Bases de Données

3. Data mining sur quels types de données ?

Le Data Mining n'est pas spécifique à un type de médias ou de données. Il est applicable à n'importe quel type d'information. Le Data Mining est utilisé et étudié pour les Bases de Données incluant les Bases de Données relationnelles et les Bases de Données Orientées-Objets, les data warehouses, les Bases de Données transactionnelles, les supports de données non structurés et semi-structurés comme le World Wide Web, les Bases de Données avancés comme les Bases de Données spatiales, les Bases de Données multimédia, les Bases de données de séries temporelles et les Bases de Données textuelles et même fichiers plats.

4. Les taches du data mining

Beaucoup de problèmes intellectuels, économiques ou même commerciaux peuvent être exprimés en termes des six tâches suivantes :

- La classification
- L'estimation
- La prédiction
- Le groupement par similitude
- L'analyse des clusters
- La description [31]

4.1. La classification

La classification est la tâche la plus commune du Data Mining et qui semble être une obligation humaine. Afin de comprendre notre vie quotidienne, nous sommes constamment classifiés, catégorisés et évalués [31].

La classification consiste à étudier les caractéristiques d'un nouvel objet pour lui attribuer une classe prédéfinie. Les objets à classifiés sont généralement des enregistrements d'une base de données, la classification consiste à mettre à jour chaque enregistrement en déterminant un champ de classe. La tâche de classification est caractérisée par une définition de classes bien précise et un ensemble d'exemples classés auparavant. L'objectif est de créer un modèle qui peut être appliqué aux données non classifiées dans le but de les classifiées [32].

4.2. L'estimation

L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorique. En fonction des autres champs de l'enregistrement l'estimation consiste à compléter une valeur manquante dans un champ particulier. Par exemple on cherche à estimer La lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang. La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas [31].

4.3. La prédiction

La prédiction est la même que la classification et l'estimation, à part que dans la prédiction les enregistrements sont classés suivant des critères (ou des valeurs) prédites (estimées). La principale raison qui différencie la prédiction de la classification et l'estimation est que dans la création du modèle prédictif on prend en charge la relation temporelle entre les variables d'entrée et les variables de sortie [31].

4.4. Le groupement par similitude

Le groupement par similitude consiste à déterminer quels attributs "vont ensemble". La tâche la plus répandue dans le monde du business, où elle est appelée l'analyse d'affinité ou l'analyse du panier du marché, est l'association des recherches pour mesurer la relation entre

deux et plusieurs attributs. Les règles d'associations sont de la forme "Si antécédent, alors conséquent".

4.5. L'analyse des clusters

Le clustering (ou la segmentation) est le regroupement d'enregistrements ou des observations en classes d'objets similaires ; un cluster est une collection d'enregistrements similaires l'un à l'autre, et différents à ceux existants sur les autres clusters. La différence entre le clustering et la classification est que dans le clustering il n'y a pas de variables sortantes. La tâche de clustering ne classifie pas, n'estime pas, ne prévoit pas la valeur d'une variable sortante. Au lieu de cela, les algorithmes de clustering visent à segmenter la totalité de données en dessous groupes relativement homogènes. Ils maximisent l'homogénéité à l'intérieur de chaque groupe et la minimisent entre ces derniers [33].

4.6. La description

Parfois le but du Data Mining est simplement de décrire se qui se passe sur une Base de Données compliquée en expliquant les relations existantes dans les données pour en premier lieu comprendre le mieux possible les individus, les produit et les processus présents sue cette base. Une bonne description d'un comportement implique souvent une bonne explication de celui-ci.

5. Les étapes du processus de data mining

- **Collecte des données :** la combinaison de plusieurs sources de données, souvent hétérogènes, dans une base de données [34] [35].
- **Nettoyage des données :** la normalisation des données : l'élimination du bruit (les attributs ayant des valeurs invalides et les attributs sans valeurs) [34] [35].
- **Sélection des données :** Sélectionner de la base de données les attributs utiles pour une tâche particulière du data mining [37].
- **Transformation des données :** le processus de transformation des structures des attributs pour être adéquates à la procédure d'extraction des informations[38].
- **Extraction des informations (Data mining) :** l'application de quelques algorithmes du Data Mining sur les données produites par l'étape précédente (Knowledge Discovery in Databases, ou KDD) [35][37].
- **Visualisation des données :** l'utilisation des techniques de visualisation (histogramme, camembert, arbre, visualisation 3D) pour exploration interactive de données (la découverte des modèles de données) [35][38].
- **Evaluation des modèles :** l'identification des modèles strictement intéressants en se basant sur des mesures données [34].

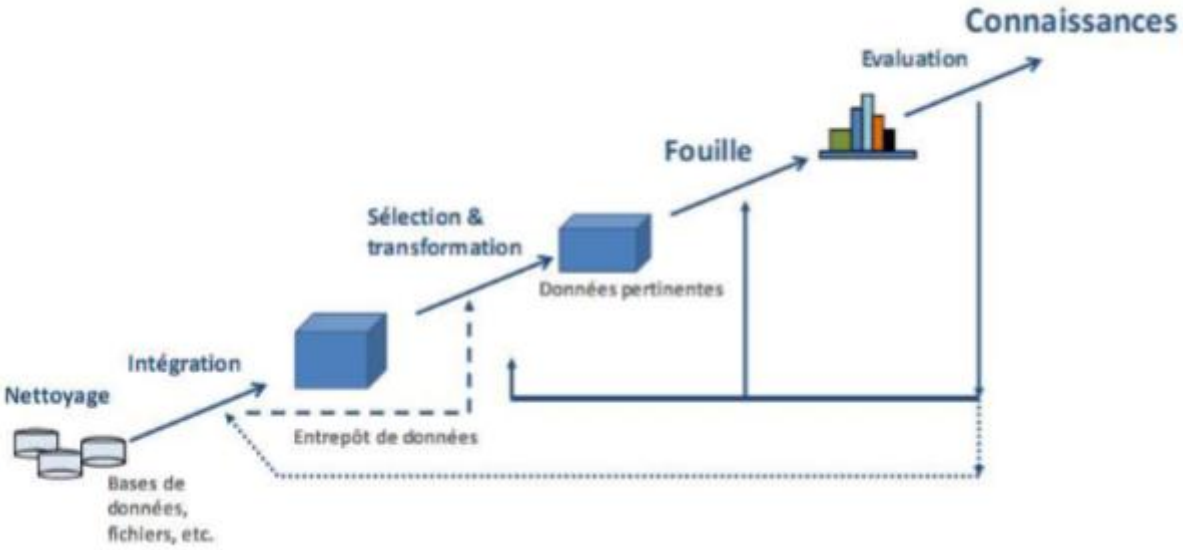


Figure 3.1 : Les étapes du processus de data mining

6. Techniques du data mining

Pour effectuer les taches du Data Mining il existe plusieurs techniques issues de disciplines scientifiques diverses (statistiques, intelligence artificielle, base de données) afin de faire apparaître des corrélations cachées dans des gisements de données pour construire des modèles à partir de ces données. Les techniques du data mining les plus connues :

- Les réseaux de neurones
- Les arbres de décision
- Les algorithmes génétiques
- Les règles associatives
- L'algorithme des k-Plus proches voisins
- L'algorithme des k-moyennes (K-Means)

7. Catégorisation des systèmes du data mining

Les systèmes de data mining peuvent être catégorisés selon plusieurs critères. Parmi les catégorisations existantes nous citons :

- **Classification selon le type de données à explorés :** dans cette classification les systèmes de data mining sont regroupés selon le type des données qu'ils manipulent tel que les données spatiales, les données de séries temporelles, les données textuelles et le Word Wide Web, etc.
- **Classification selon les modèles de données avancés :** cette classification catégorise les systèmes de data mining en se basant sur les modèles de données avancés tel que les bases de données relationnelles, les bases de données orientées objets, les data warehouses, les bases de données transactionnelles, etc.
- **Classification selon le type de connaissance à découvrir :** cette classification catégorise les systèmes de data mining en s'appuyant sur le type de connaissance à

découvrir ou les tâches de data mining tel que la classification, l'estimation, la prédiction, etc.

- **Classification selon les techniques d'exploration utilisées :** cette classification catégorise les systèmes de data mining suivant l'approche d'analyse de données utilisés la reconnaissance des formes, les réseaux neurones, les algorithmes génétiques, les statistiques, la visualisation, orienté-base de données ou orienté-data warehouse, etc [34].

8. Le data mining dans la bio-informatique et la biotechnologie

La Bio-informatique est un domaine de recherche en développement rapide, qui a des racines aussi bien dans la biologie que dans la technologie d'informations.

Quelques applications du data mining dans ce domaine sont :

- La prédiction les structures de différentes protéines.
- La détermination de la complexité des structures de plusieurs médicaments.

9. Les classifications

9.1. Un peu d'histoire

En 1813 Augustin Pyramus de Candolle a utilisé pour désigner la science des lois de la classification des formes vivantes selon les critères de regroupement : taille, forme des feuilles, racines, etc. sous le nom Taxinomie (grec : ordre, arrangement et loi).

Linné (en science naturelles), et Koppen (classification des climats) en 1911.

Et pour la première fois en 1939 l'utilisation de terme classification avec ces différents algorithmes par Tryon. Robert R. Sokal et Peter H.A. Sneath présentent en 1963 des méthodes quantitatives appliquées à la taxinomie [39].

9.2. Définition

Classifier c'est regrouper entre eux des objets similaires selon certain critères par les diverses techniques de classification visent toutes à répartir n individus caractérisés par p variables$X1$,2, …, Xp en un certain nombre m de sous-groupes aussi homogènes que possible. Selon la méthode de classification qui peut être directe en un nombre fixé de classes ou sous la forme d'une hiérarchie à plusieurs niveaux d'agrégation. Le modèle général s'appuie sur la distance entre un individu et un autre. Plus cette distance est réduite, plus les deux entités sont proches et la classification se fait sur cette base quel que soit la méthode utilisée, ce critère de regroupement ou la nature de distance utilisée.

9.3. Formalisation mathématique de problème de classification

En terme mathématique, un problème de classification comporte les ingrédients suivants :

- Une population de N individu $I\ i$ (i variant de 1 à N)
- P variables descriptives $Xd\ i$ qui permettent de décrire les individus ; elles sont aussi appelées plus simplement descripteurs (d variant de 1 à P)
- C classes Ck dans lesquelles on cherche à ranger les individus (k variant de1 à C)

Résoudre un problème de classification, c'est trouver une application de l'ensemble des objets à classer, décrits par les variables descriptives choisies, dans l'ensemble des classes [40].

L'algorithme ou la procédure qui réalise cette application est appelé classifieur.

9.4. Préparation des données en vue d'une classification

Des variables sans rapport avec le problème posé peuvent entrainer une classification futile car elles ne peuvent qu'affecter négativement les mesures de proximité et éliminer la tendance à la structuration en classes.

Un analyse exploratoire préliminaire est donc essentielle pour éliminer ces variables inappropriées, réduire la cardinalité des variables catégorielle, mettre en évidence la présence d'oublier et homogénéiser, si nécessaire, les variables hétérogènes à prendre en compte simultanément dans une méthode de classification.

D'autre part, une standardisation de variables numériques retenues pour la classification permet de donner le même poids à toutes ces variables dans l'analyse [40].

9.5. L'objectif de la classification

La classification a pris aujourd'hui une place importante en analyse des données exploratoire et décisionnelle, l'objectif exploratoire vise à découvrir une partition hypothétique dans un ensemble d'objets. Dans l'analyse décisionnelle, on cherche généralement à affecter tout nouvel objet à des groupes préalablement définis.

La classification a pour but plus simple est répartir l'échantillon en groupes d'observation homogènes, chaque groupe étant bien différencié des autres.

On veut en général obtenir des sections à l'intérieur des groupes principaux, puis des subdivisions plus petites de ces sections, et ainsi de suite. En bref, on désire avoir une hiérarchie de plus en plus fine, sur l'ensemble d'observations initial.

9.6. Les termes désignant la classification

Plusieurs termes sont utilisés dans la littérature pour désigner une technique de classification parmi lesquels : classification automatique, apprentissage non supervisé (dans le domaine de la reconnaissance des formes), analyse typologique, taxinomie ou taxonomie numérique (en biologie et zoologie), nosologie (en médecine), partition dans la théorie des graphes.

Le terme anglais pour désigner une technique de classification est clustering (non superviser) classification [40].

9.7. Les méthodes de classification

Il existe un grand nombre de méthodes et surtout beaucoup de variantes. Il est d'objectif de les différentier grossièrement soit par leur structure de classification, soit par le type de représentation des classes.

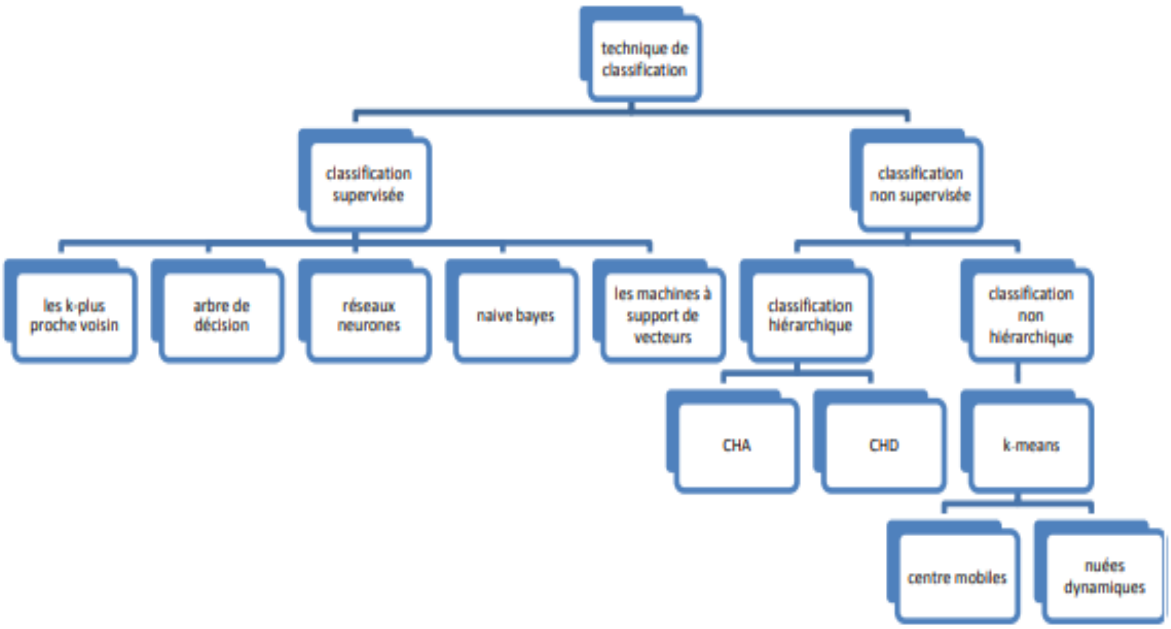


Figure 3.2 : Les méthodes de classification.

Ainsi, nous discuterons du deuxième type :

9.7.1. La classification non supervisée

Cette classification est aussi appelée "classification automatique", "clustering" ou encore "regroupement". Dans ce type de classification on est amené à identifier les populations d'un ensemble de données. On suppose qu'on dispose d'un ensemble d'objets que l'on note par X = {x1,2,……,}caractérisé par un ensemble de descripteurs D, l'objectif du clustering est de trouver les groupes auxquels appartiennent chaque objet x qu'on note par C = {C1 , C2 , … … . . } Ce qui revient à déterminer une fonction notée Ys qui associe à chaque élément de X un ou plusieurs éléments de C. Il faut pouvoir affecter une nouvelle observation à une classe. Les disponibles ne sont pas initialement identifiées comme appartenant à telle ou telle population. L'absence d'étiquette de classe est un lourd handicap qui n'est que très partiellement surmontable. Seule l'analyse de la répartition spatiale des observations peut permettre de "deviner" où sont les véritables classes. Parmi les méthodes non-supervisées les plus utilisées, citons deux types d'approches : les centres mobiles (k-means) et la classification hiérarchique [42].

9.7.1.1. Classification non hiérarchique

- Toute classe soit non vide ;

- Deux classes distinctes sont disjointes ;

- Tout individu appartient à une classe.

Cet algorithme porte le nom de "agrégation autour de centres variables". Une version légèrement différente, connue sous le nom de "nuées dynamiques" consiste à représenter chaque groupe non pas par son centre, mais par un ensemble de points (noyau) choisis

aléatoirement à l'intérieur de chaque groupe. On calcule alors une distance "moyenne" entre chaque observation et ces noyaux et l'on procède à l'affectation [43].

9.7.1.1.1. Méthode de k-means

C'est une méthode dont le but est de diviser des observations en k partitions dans lesquelles chaque observation appartient à la partition avec la moyenne la plus proche.

Nous citons deux méthodes connues sur le principe de k-means sont :

→ Méthodes de centres mobiles ;

→ Méthodes des nuées dynamiques.

- **Méthode de entres mobiles**
  Cette méthode consiste à construire une partition en k classes en sélectionnant k individus commence, des classes tirées au hasard de l'ensemble d'individus. Après cette sélection, on affecte chaque individu au centre le plus proche en créant k classes, les centres des classes seront remplacés par les centres de gravité et nouvelles classes seront créés par le même principe. Généralement la partition obtenue est localement optimale car elle dépend du choix initial des centres. Pour cela les résultats entre deux exécutions de l'algorithme sont significativement variés.
- **Méthode de nuées dynamiques**
  Dans ce cas, le problème posé est la recherche d'une partition en k (k fixé) classes d'un ensemble de n individus. C'est un algorithme itératif.
  Soit *I* une population d'individus, cette population est représentable sur R et forme un nuage de n points.
  On cherche à constituer une partition en k classes sur i. chaque classe est représentée par son centre, également appelé noyau, constitué du petit sous-ensemble de la classe qui minimise le critère de dissemblance.

9.7.1.2. Classification hiérarchique

La classification hiérarchique : pour un niveau de précision donné, deux individus peuvent être confondus dans un même groupe, alors qu'à un niveau de précision plus élevé, ils seront distingués et appartiendront à deux sous-groupes différents. Le résultat d'une classification hiérarchique n'est pas une partition de l'ensemble des individus. C'est une hiérarchie de classes telle que :

- Toute classe est non vide.

- Tout individu appartient à une (et même plusieurs) classes.

- Deux classes distinctes sont disjointes, ou vérifient une relation d'inclusion (l'une d'elle est incluse dans l'autre)

- Toute classe est la réunion des classes qui sont incluse dans elle.

L'avantage de cette méthode est qu'elle n'est soumise à aucune initialisation particulière de paramètre(s) ce qui la rend déterministe, et en outre, que le nombre de classe n'a pas à

être fixé a priori. Cependant, ce type de méthode impose le calcul de la matrice des distances de tous les points d'observation avec tous les autres, et cette masse de calculs est beaucoup trop importante compte tenu du temps que nous voulons consacrer à cette étape [43].

Parmi les méthodes non-supervisées les plus utilisées, citons deux types d'approches :

9.8. Classification hiérarchique ascendante

La CAH permet de construire une hiérarchie entière des objets sous la forme d'un "arbre" dans un ordre ascendant. On commence en considérant chaque individu comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe.

Le processus est itéré jusqu'à ce que tous les individus se trouvent dans une même classe. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre des classes plus ou moins grand.

Différentes mesures de la distance interclasses peuvent être utilisées : la distance euclidienne, la distance inferieur (qui favorise la création de classes de faible inertie) ou la distance supérieure (qui favorise la création de classes d'inertie plus importante) etc.

Le cas de la classification ascendante hiérarchique, à partir des éléments, on forme des petites classes ne comprenant que des individus très semblables, puis à partir de celle-ci, on construit des classes de moins en moins homogènes, jusqu'à obtenir la classe tout entière[43].

9.8.1. Distance définie sur un ensemble E

C'est une application du produit cartésien E×E dans $\mathbb{R}$+ satisfaisant aux axiomes suivants :

Symétrie d(Xi ,Xi ') =d(Xi ' , Xi), ∀Xi ∈ E, ∀Xi ' ∈ E.

Positivité stricte d(Xi ,Xi ') >0 si $Xi \neq Xi$ ' et d(Xi ,Xi ') =0 ⟺siXi = Xi ' , ∀Xi ∈ E, ∀Xi ' ∈ E.

Inégalité triangulaire d(Xi ,Xi ') ≤ d Xi , Xi '' + d(Xi '' , Xi '), ∀Xi ∈ E, ∀Xi ' ∈ E et, ∀Xi '' ∈ E.

9.8.2. Les mesures de distance

La classification ascendante hiérarchique CAH utilise des mesures de dissemblance ou de distance entre les objets pour former des classes. Ces distances peuvent être basées sur une ou plusieurs dimensions. La méthode la plus directe pour calculer des distances entre objets dans un espace multidimensionnel consiste à calculer les distances euclidiennes. Si nous avons un espace à deux ou trois dimensions, cette mesure est celle des distances géométriques normales entre les objets dans l'espace.

La classification permet de calculer de nombreux types de mesures de distances, afin de l'utiliser directement dans la procédure.

- **Distance Euclidienne**

C'est probablement le type de distance le plus couramment utilisé. Il s'agit simplement d'une distance géométrique dans un espace multidimensionnel. Elle se calcule ainsi :

Distance (X, Y) = (∑i (Xi -Yi) 2) ½

- **Distance Euclidienne au carré**
  Vous pouvez élever la distance Euclidienne standard au carré afin de surpondérer les objets atypiques (éloignés). Cette distance se calcule ainsi :
  Distance (X, Y) = ∑i (Xi -Yi) 2
- **Distance du city-block (Manhattan)**
  Cette distance est simplement la somme des différences entre les dimensions. Dans la plupart des cas, cette mesure de distance produit des résultats proches de ceux obtenus par la distance euclidienne simple. En revanche, notez qu'avec cette mesure, l'effet des différences simples importantes (points atypiques) est atténué (puisque ces distances ne sont pas élevées au carré). Cette distance se calcule ainsi :
  Distance (X, Y) = ∑i |Xi -Yi |
- **Distance de Tchebychev**
  Cette mesure de distance est adaptée lorsque nous considérons deux objets comme étant différents à partir du moment où ils sont différents sur l'une des dimensions. La distance de Tchebychev se calcul ainsi :
  Distance (X, Y) = Maximum |Xi -Yi |
- **Distance de puissance**
  Nous pouvons parfois souhaiter augmenter ou diminuer la pondération progressive associée à des dimensions pour lesquelles les objets respectifs sont très différents. Cette opération est rendue possible par la distance à la puissance. La distance à la puissance se calcule ainsi :
  Distance (X, Y) = (∑i |Xi -Yi | P) 1 r
  Où r et p sont des paramètres définis par l'utilisateur. Le paramètre p contrôle la pondération progressive affectée aux différences entre les dimensions individuelles, tandis que le paramètre contrôle la pondération progressive affectée aux grandes différences entre les objets. Si r et p sont égaux à 2, cette distance équivaut à la distance euclidienne.
- **Percent désagrègement**
  Cette mesure est particulièrement utile si les données des dimensions utilisées dans l'analyse sont de nature catégorielle. Cette distance se calcule ainsi :
  Distance (X, Y) = (Nombre de$Xi \neq Yi$) /i

9.8.3. La dissimilarité définie sur un ensemble E

Elle est définie par à partir de l'indice de similarité : dis (Xi, Xi ') = 1- S (Xi ', Xi)

Les critères d'agrégation

De nombreux critères d'agrégation ont été proposés les plus connus sont :

- **Le critère du saut minimal**

La distance entre 2 classes $C1$ et $C2$ est définie par la plus courte distance séparant un individu de $C1$ et un individu de$C2$.

D ($C1$, $C2$) = min ({d (x, y)}, x∈ $C1$, y∈ $C2$)

- **Le critère du saut maximal**

  La distance entre 2 classes $C1et\ C2$ est définie par la plus grande distance séparant un individu de $C1$ et un individu de $C2$

  D ($C1$, $C2$) =max ({d (x, y)}, x ∈ $C1$, y∈ $C2$)

- **Le critère de la moyenne**

  Ce critère consiste à calculer la distance moyenne entre tous élément de $C1$ et tous les éléments de$C2$.

  D ($C1$, $C2$) = 1 $nC1\ n\ C2 \sum \sum d\ x, y\ x \in C1\ y \in C2$

  Avec :

  $nC1$: Le cardinal de$C1$

  $nC2$: Le cardinal de $C2$

- **Le critère de Ward**

  Ce critère ne s'applique que si on est muni 'un espace euclidien. La dissimilarité entre 2 individus doit être égale à la moitié du carré de la distance euclidienne d, le critère de Ward consiste à choisir à chaque étape le regroupement de classes tel que l'augmentation de l'inertie intraclasse soit minimal.

  D ($C1$, $C2$) = $nC1nC2\ nC1+nC2\ d\ 2\ (gC1, gC2)$

  Avec :

  $gC1$: Centre de gravité de $C1$

  $gC2$: Centre de gravité de $C2$

- **Le critère de centre de gravité**

  La distance entre 2 classes $C1$et $C2$ est définie par la distance entre leurs centres de gravité.

  D ($C1$, $C2$) = d ($gC1$, $gC2$)

  La difficulté du choix du critère d'agrégation réside dans le fait que ces critères peuvent déboucher sur des résultats différents.

9.8.4. L'algorithme CAH

L'algorithme de la classification ascendante hiérarchique est très simple. Il est dû à Lance et William (1967).

Initialisation construction du tableau des distances, peu importe la formule utilisée pour le construire car l'algorithme de CAH est indépendant de la métrique utilisée. Ainsi, entre chaque couple de point (x, y) de M, nous disposons d'une valeur d (x, y). La partition initiale est la plus fine ρ0 de M.

Regroupement parcourir le tableau de distance pour déterminer le couple d'élément ($x*$, $y*$) les plus proches :

d ($x*$, $*$) ≤ $min\ x,y \in M$ {d(x, y)}

On réunit les deux éléments dans une même classe A=$x*$ ∪ $y*$ les autres classes restent inchangées. Nous obtenons une nouvelle partition ρi moins fie que la précédente.

Tableau des distances la classe A sera vue comme un seul point. Il faut donc calculer les distances qu'il y a entre le point A qui est un ensemble de cardinal supérieur à un, et tous les autres points qui ne sont pas dans A et peuvent être des singletons. Par souci de généralité, nous les notons B.

d (A, B) ; B ⊄ A

Pour cela, on peut utiliser l'un des cinq critères proposés plus haut. Nous disposons alors d'un nouveau tableau des distances ayant une ligne et une colonne de moins que le précédent dont il ne diffère que par ligne et colonne qui correspond au point A.

Condition d'arrêt si nous avons atteint la partition du niveau souhaité, généralement c'est la partition grossière, celle qui ne comporte qu'une seule classe réunissant la totalité des points, alors, c'est terminé. Dans le cas contraire, nous repartons de l'étape regroupement à partir du tableau des distances calculé à la suite du précédent regroupement.

9.9. Classification hiérarchique descendante

Dans la CDH, en considérant tous les individus comme une seule classe au début, on divise successivement les classes en classes plus raffinées. Le processus marche jusqu'à ce que chaque classe contienne un seul point ou bien si l'on atteint un nombre de classes désiré [43].

10. Construction des arbres phylogénétiques

10.1. Structure des arbres

Les relations évolutives entre les objets étudiés sont représentées par des arbres phylogénétiques.

Les arbres sont des graphes composés de :

- Nœuds et de branches
- Nœuds = unités taxonomiques
  - Feuilles ou OTU = Unités Taxonomique Opérationnelles ou (A, B, C, D, E)
  - Nœuds internes ou HTU = Unités taxonomique Hypothétiques (F, G, H, I)
- Branches = relations de parentés (ancêtre /descendants) entre les unités taxonomiques
  - Branches internes
  - Branches externes
  - L'ensemble des branchements de l'arbre =topologie de l'arbre.

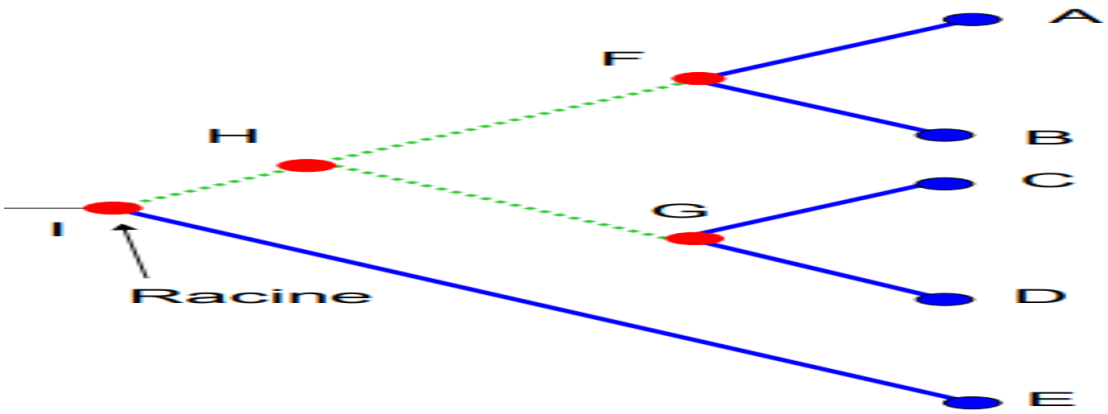


Figure 3.3 : Topologie d'un arbre

- **Arbres enracinés vs Arbres non enracinés :**

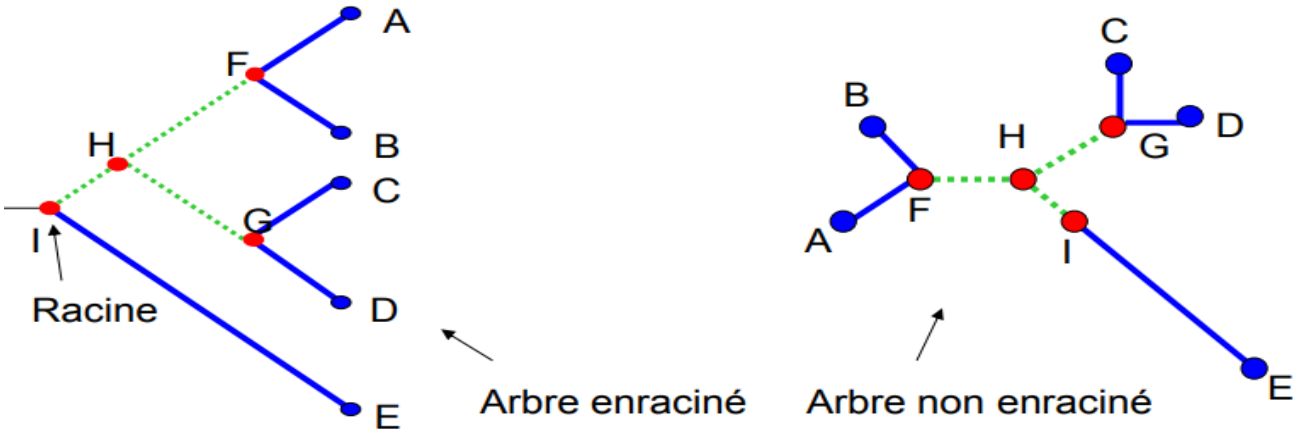


Figure 3.4 : Arbres enracinés vs Arbres non enracinés

- La racine symbolise le dernier ancêtre commun (i.e. le plus récent) de toutes les OTU.
- Les arbres non racinés ne sont pas réellement des arbres phylogénétiques car ils n'ont pas de dimension temporelle => N'indiquent pas les relations de parenté entre les OTU.

10.2. Méthodes de construction des arbres phylogénétiques :

Il existe deux grands types de méthodes permettant la reconstruction d'arbres phylogénétiques :

- les méthodes basées sur les mesures de distances entre séquences prises deux à deux, c'est à dire le nombre de substitutions de nucléotides ou d'acides aminés entre ces deux séquences.
- les méthodes basées sur les caractères qui s'intéressent au nombre de mutations (substitutions /insertions /délétions) qui affectent chacun des sites (positions) de la séquence.

10.2.1. Méthodes fondées sur les distances

Ce sont des méthodes de reconstruction d'arbre phylogénétique sans racine basée sur la recherche d'OTU (operationnal taxonomic unit), le plus souvent équivalent à une séquence) les plus proches et ceci à chaque étape de regroupement.

Ces méthodes sont rapides et donnent de bons résultats pour des séquences ayant une forte similarité.

Programmes DNADIST et PROTDIST de Phylip.

10.2.1.1. UPGMA (Unweight Pair Group Method with Arithmetic mean)

Cette méthode est utilisée pour reconstruire des arbres phylogénétiques si les séquences ne sont pas trop divergentes.

UPGMA utilise un algorithme de clusterisation séquentiel dans lequel les relations sont identifiées dans l'ordre de leur similarité et la reconstruction de l'arbre se fait pas à pas grâce à cet ordre.

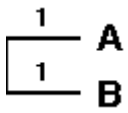
Il y a d'abord identification des deux séquences les plus proches et ce groupe est ensuite traité comme un tout, puis on recherche la séquence la plus proche et ainsi de suite jusqu'à ce qu'il n'y ait plus que deux groupes.

-Exemple

On considère la matrice de distances associé à un groupe de 6 OTUs

	A	**B**	**C**	**D**	**E**
B	**2**				
C	**4**	**4**			
D	**6**	**6**	**6**		
E	**6**	**6**	**6**	**4**	
F	**8**	**8**	**8**	**8**	**8**

On clustérise tout d'abord les deux OTUs avec la distance la plus faible (A et B). Le point de branchement est positionné à la distance 2/2=1.
On peut alors construire le sous arbre suivant :



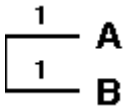
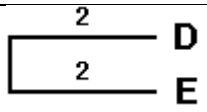
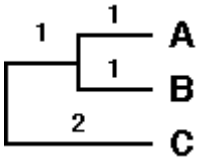
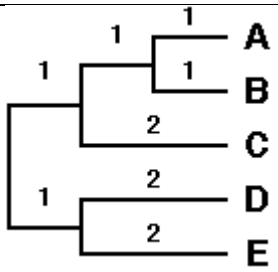
Dans la suite, le cluster (A,B) est considéré comme un tout et on peut calculer une nouvelle matrice de distance :

dist(A,B),C = (distAC + distBC) / 2 = 4

dist(A,B),D = (distAD + distBD) / 2 = 6

dist(A,B),E = (distAE + distBE) / 2 = 6

dist(A,B),F = (distAF + distBF) / 2 = 8

Cycle 1	**A** **B** **C** **D** **E** **B** 2 **C** 4 4 **D** 6 6 6 **E** 6 6 6 4 **F** 8 8 8 8 8	
Cycle 2	**A,B** **C** **D** **E** **C** 4 **D** 6 6 **E** 6 6 4 **F** 8 8 8 8	
Cycle 3	**A,B** **C** **D,E** **C** 4 **D,E** 6 6 **F** 8 8 8	
Cycle 4	**AB,C** **D,E** **D,E** 6 **F** 8 8	

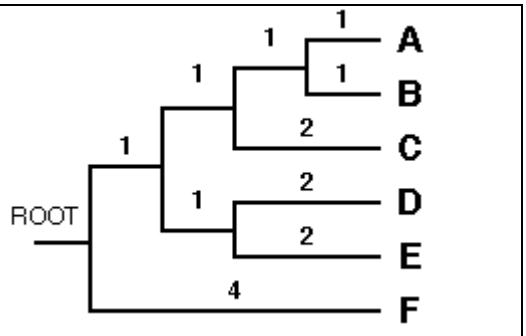
Cycle 5	**ABC,DE** **F** 8	

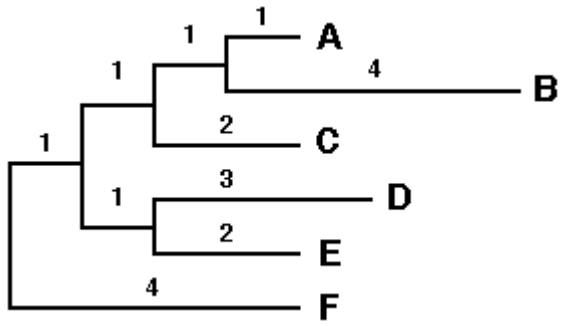
Tableau 3.1 : matrice de distance

Cette méthode conduit essentiellement à un arbre non enraciné. Si on veut enraciner l'arbre, on peut appliquer la méthode du "mid-point rooting" : la racine de l'arbre est à équidistance de tous les OTUs soit (ABCDE), F / 2 = 4

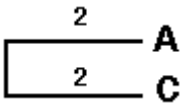
10.2.1.2. Les inconvénients de la méthode UPGMA

L'inconvénient majeur est la sensibilité de la méthode à des taux de mutations différents sur les différentes branches

Supposons que l'on veuille reconstruire l'arbre suivant à partir de la matrice de distances associée aux séquences :



Depuis que A et B ont divergé, B a accumulé beaucoup plus de mutations que A

	Matrice	**Arbre**
Cycle 1	**A** **B** 5 **C** 4 7 **D** 7 10 7 **E** 6 9 6 5 **F** 8 11 8 9 8	

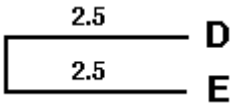
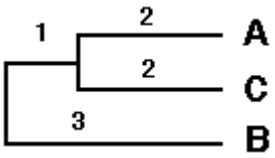
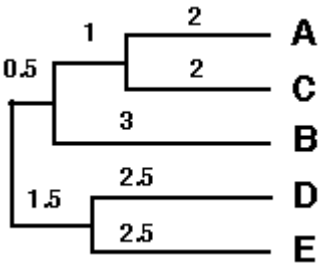
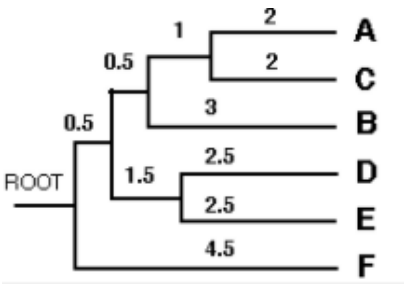
Cycle 2	**A,C** **B** **D** **E** **B** 4 **D** 7 10 **E** 6 9 5 **F** 8 11 8 9	
Cycle 3	**A,C** **B** **D,E** **B** 6 **D,E** 6.5 9.5 **F** 8 11 8.5	
Cycle 4	**AC,B** **D,E** **D,E** 8 **F** 9.5 9.5	
Cycle 5	**ABC,DE** **F** 9	

Tableau 3.2 : matrice de distance pour UPGMA

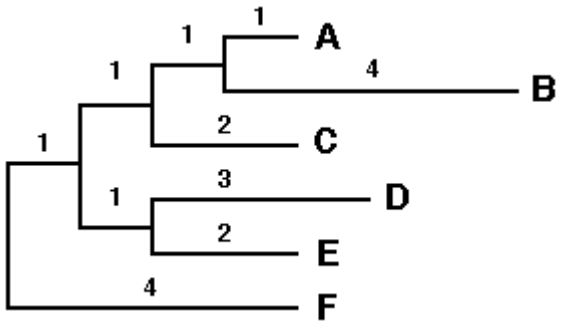
10.2.1.3. NJ (Neighbor-Joining)

Cette méthode développée par Saitou et Nei (1987) tente de corriger la méthode UPGMA afin d'autoriser un taux de mutation différent sur les branches.

Les données initiales permettent de construire une matrice qui donne un arbre en étoile. Cette matrice de distances est ensuite corrigée afin de prendre en compte la divergence moyenne de chacune des séquences avec les autres.

L'arbre est alors reconstruit en reliant les séquences les plus proches dans cette nouvelle matrice. Lorsque deux séquences sont liées, le nœud représentant leur ancêtre commun est ajouté à l'arbre tandis que les deux feuilles sont enlevées. Ce processus convertit l'ancêtre commun en un nœud terminal dans un arbre de taille réduite. Programme NEIGHBOR de Phylip.

-Exemple :



La matrice de distance associée à cet arbre est la suivante :

	A	**B**	**C**	**D**	**E**
B	5				
C	4	7			
D	7	10	7		
E	6	9	6	5	
F	8	11	8	9	8

Etape 1 : calcul de la divergence de chacun des N OTUs par rapport aux autres (N= 6)

r (A) = 5+4+7+6+8 = 30
r(B) = 42
r(C) = 32
r(D) = 38
r (E) =34
r(F) = 44

Etape 2 : calcul de la nouvelle matrice en utilisant la formule

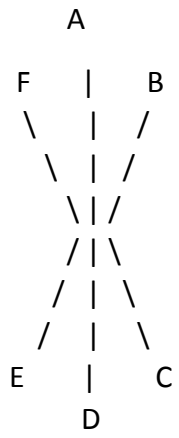
M (i,j) = d(ij) -[r(i) + r(j)] / (N-2)

ce qui donne pour la paire AB : M(AB) = 5 - [30 + 42] / 4 = -13

A

	A	**B**	**C**	**D**	**E**
B	-13				
C	-11.5	-11.5			
D	-10	-10	-10.5		
E	-10	-10	-10.5	-13	
F	-10.5	-10.5	-11	-11.5	-11.5

Ceci permet de construire l'arbre en étoile suivant :



Etape 3 : Choix des plus proches voisins, c'est à dire des deux OTUs ayant le M(i,j) le plus petit, donc soit A et B soit D et E.

On prend A et B et on forme un nouveau noeud U et on calcule la longueur de la branche entre U et A ainsi qu'entre U et B :

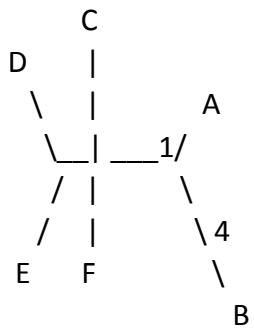
S (AU) = d (AB) / 2 + [r(A) - r(B)] / 2 (N-2) = 5/2 + [30-42] /2(6-4) = 1
S(BU) = d (AB) - S(AU) = 5 - 1 = 4

Etape 4 : on définit les nouvelles distances entre U et les autres OTUs
d (CU) = d(AC) + d (BC) - d(AB) / 2 = 3
d (DU) = d(AD) + d(BD) -d(AB) /2 = 6
d (EU) = d(AE) + d (BE) - d(AB) / 2 = 5
d (DU) = d(AF) + d(BF) -d(AB) /2 = 7

Création d'une nouvelle matrice :

	U	**C**	**D**	**E**
C	**3**			
D	**6**	**7**		
E	**5**	**6**	**5**	
F	**7**	**8**	**9**	**8**

Et d'un arbre en étoile :



La procédure complète repart de l'étape 1 avec N = N-1 = 5.

10.2.1.4. Méthodes dérivées des méthodes basées sur les distances

Dans ces méthodes, l'arbre sélectionné est celui présentant la somme des carrés des écarts entre les distances observées et les distances calculées la plus faible.

Il existe, dans Phylip, deux programmes utilisant ces méthodes :

- FITCH : la longueur des branches n'est pas proportionnelle au temps écoulé et la racine n'est pas localisée dans l'arbre.
- KITSCH : on fait l'hypothèse que toutes les espèces sont contemporaines et qu'il y a une horloge moléculaire. Cela signifie que les branches de l'arbre ne peuvent pas avoir une longueur arbitraire mais sont proportionnelles au temps écoulé. Les arbres générés par cette méthode présentent une racine.

Les corrections

Si le temps de divergence entre deux séquences augmente, la probabilité d'avoir une seconde mutation à un site augmente également. Ceci fait que le simple comptage des différences entre deux séquences n'est pas le reflet exact de la réalité mais sous-estime le nombre d'évènements mutationnels. On tente de corriger ce biais en faisant des hypothèses sur la façon dont les bases ou acides aminés se sont subsitués à un locus donné. Les premiers à avoir proposés une solution à ce problème sont Jukes et Cantor en 1969.

- Types de substitutions

On distingue différents types de substitution suivant les bases impliquées.

Transitions :

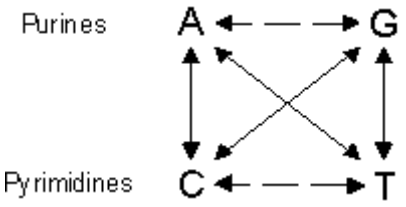
A <-> G, C <-> T

Trans versions :

A <-> C, A <-> T,

G <-> C, G <-> T

Lorsque l'on compare deux séquences, on différencie aussi les substitutions selon leur ordre et leurs conséquences.



	Séquence 1	**Séquence 2**	**Nb de substitutions observées**	**Nb réel de substitutions**
Substitution unique	C	C -> A	1	1
Substitutions multiples	A	A -> C -> T	1	2
Substitutions coïncidentes au même site	C -> A	C-> G	1	2
Substitutions parallèles	T -> A	T -> A	0	1
Substitutions convergentes	C -> T -> A	C -> A	0	3
Substitutions reverses	C -> T -> C	C	0	1

Tableau 3.3 : correction pour les substitutions multiples

Exemples de corrections pour les substitutions multiples

- Correction de Jukes et Cantor (1969) : On fait l'hypothèse que tous les sites sont équivalents (tous les changements ont une probabilité égale mais elle varie au cours du temps), qu'il n'y a pas de biais dans la direction du changement et qu'il n'y a eu ni insertions ni délétions. C'est l'hypothèse la plus simple, mais pas forcément la plus correcte.
- Correction de Kimura ou 2 paramètres (1980) : ce modèle est similaire au modèle de Jukes-Cantor mais on fait l'hypothèse que le taux de transition est différent du taux de Trans version. Ce modèle a été développé suite à l'observation que les transitions étaient souvent beaucoup plus fréquentes que les Trans versions.

Si **P** est la fréquence des transitions et **Q** la fréquence des Trans versions :

10.2.2. Méthodes fondées sur les caractères :

Ces méthodes sont très lentes mais elles sont précises.

10.2.2.1. Parcimonie

La parcimonie consiste à minimiser le nombre de "pas" (mutations / substitutions) nécessaires pour passer d'une séquence à une autre dans une topologie de l'arbre.
Pour cela, cette méthode s'appuie sur les hypothèses suivantes :
- les sites évoluent indépendamment les uns des autres (la séquence peut être considérée comme une suite de caractères non ordonnés)
- la vitesse d'évolution est lente et constante au cours du temps.
Cette méthode, quand elle est appliquée à des séquences protéiques, utilise le code génétique pour comptabiliser le nombre de substitutions nécessaires (changements de bases) pour passer d'un site à l'autre d'une séquence à l'autre.
La méthode de maximum de parcimonie recherche toutes les topologies possibles afin de trouver l'arbre optimal (minimum) et le temps nécessaire pour cette exploration croit rapidement avec le nombre de séquences :
le nombre d'arbres enracinés possibles pour n OTUs : Nr = (2n - 3)! / (2exp(n-2))(n-2)!
le nombre d'arbres non enracinés possibles pour n OTUs : Nu = (2n -5)! / (2exp(n-3))(n-3)!
Programme DNAPARS et PROTPARS de Phylip.

Nombre d'OTUs	**Nb d'arbres non enracinés**	**Nb d'arbres enracinés possibles**
2	1	1

3	1	3
4	3	15
5	15	105
6	105	945
7	954	10 395
8	10 395	135 135
9	135 135	34 459 425
10	34 459 425	2.13 E15
15	2.13 E15	8E21

Tableau 3.4 : nombre des arbres enracines et non enracines

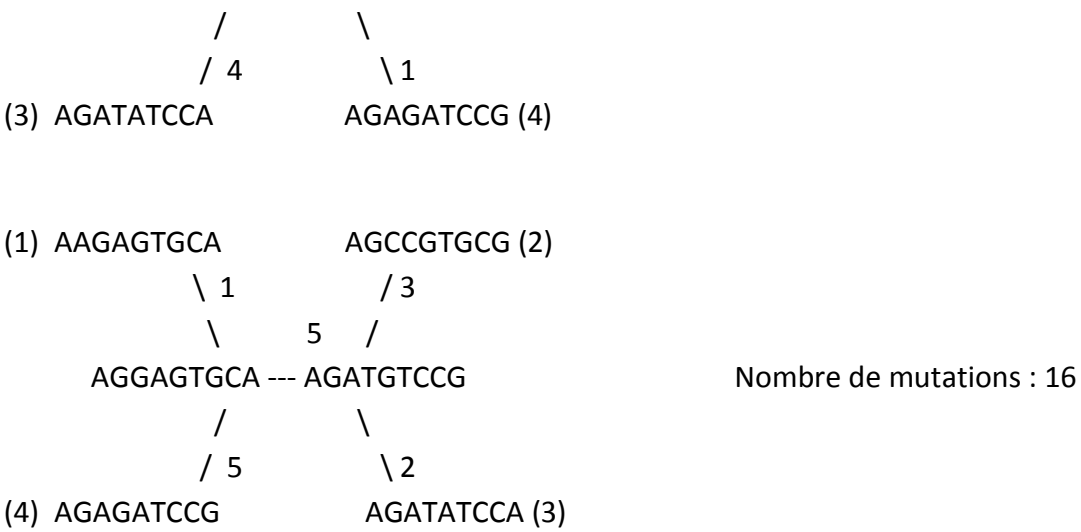
Exemple

Séquence	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	A	A	G	A	G	T	G	C	A
2	A	G	C	C	G	T	G	C	G
3	A	G	A	T	A	T	C	C	A
4	A	G	A	G	A	T	C	C	G

Pour 4 séquences, il y a 3 arbres non enracinés possibles. Ces trois arbres sont analysés (recherche de la séquence ancestrale et comptage du nombre de mutations)

(1) AAGAGTGCA AGATATCCA (3)
\ 4 / 2
\ 4 /
AGCCGTGCG --- AGAGATCCG Nombre de mutations : 10
/ \
/ 0 \ 0
(2) AGCCGTGCG AGAGATCCG (4)

(1) AAGAGTGCA AGCCGTGCG (2)
\ 1 / 3
\ 5 /
AGGAGTGCA --- AGAGGTCCG Nombre de mutations : 14



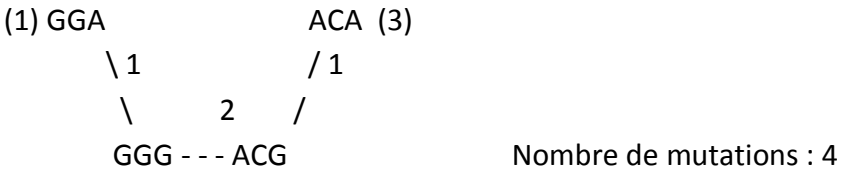
Nombre de mutations : 16

L'arbre I est celui nécessitant le moins de mutations, c'est donc le plus parcimonieux. Cette analyse prend en compte tous les sites des séquences mais l'analyse peut également se faire uniquement sur les sites informatifs, c'est à dire quand à cette position il y a au moins 2 nucléotides différents, représentés chacun dans au moins deux séquences.

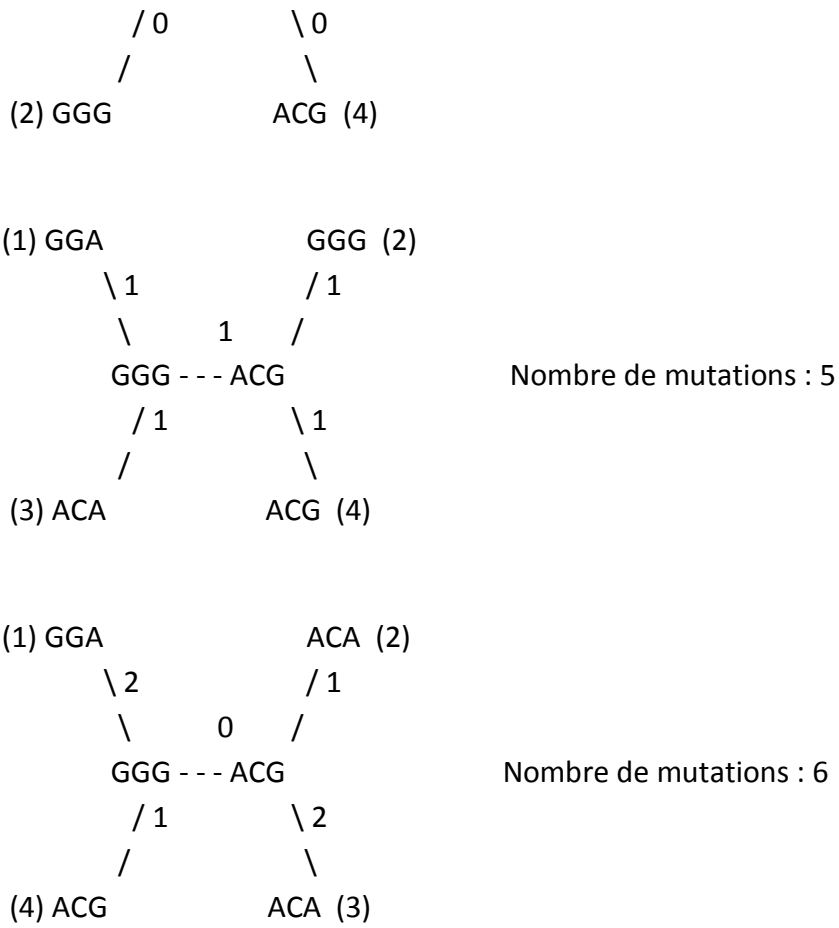
Séquence	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	A	A	G	A	G	T	G	C	A
2	A	G	C	C	G	T	G	C	G
3	A	G	A	T	A	T	C	C	A
4	A	G	A	G	A	T	C	C	G
					*		*		*

On peut donc "réduire" les séquences aux seuls sites informatifs :

1	G	G	A
2	G	G	G
3	A	C	A
4	A	C	G
	*	*	*



Nombre de mutations : 4



Dans le cas de 4 séquences, un site informatif favorise seulement un arbre : le site 5 favorise l'arbre I plus que les arbres II et III (il supporte l'arbre I). L'arbre le plus parcimonieux est celui qui est supporté par le plus grand nombre de sites informatifs.
Le maximum de parcimonie recherche l'arbre optimal et dans ce processus, il est possible de trouver plusieurs arbres optimaux (= arbres ex-aequo = configuration comptabilisant le même nombre minimal de substitutions nécessaires pour passer d'une séquence à l'autre dans l'ensemble de l'arbre).
Afin de garantir de trouver l'arbre le meilleur possible, il faut faire une évaluation de toutes les topologies possibles mais cela devient impossible lorsque l'on a plus de 12 séquences.

10.2.2.2. Branch and Bound

Cette méthode est dérivée du maximum de parcimonie, elle garantit de trouver le meilleur arbre mais sans évaluer tous les arbres possibles. Elle permet de traiter un plus grand nombre de séquences mais reste limitée.

10.2.2.3. Recherche heuristique

Il y a un réarrangement des branches à chaque étape, cette méthode ne garantit pas de trouver l'arbre optimal.

10.2.2.4. Arbre consensus

Comme la méthode du maximum de parcimonie peut conduire à trouver plusieurs arbres équivalents, on peut créer un arbre consensus (avec utilisation du bootstraping). Cet arbre consensus est construit à partir des noeuds les plus fréquemment rencontrés sur l'ensemble des arbres possibles.

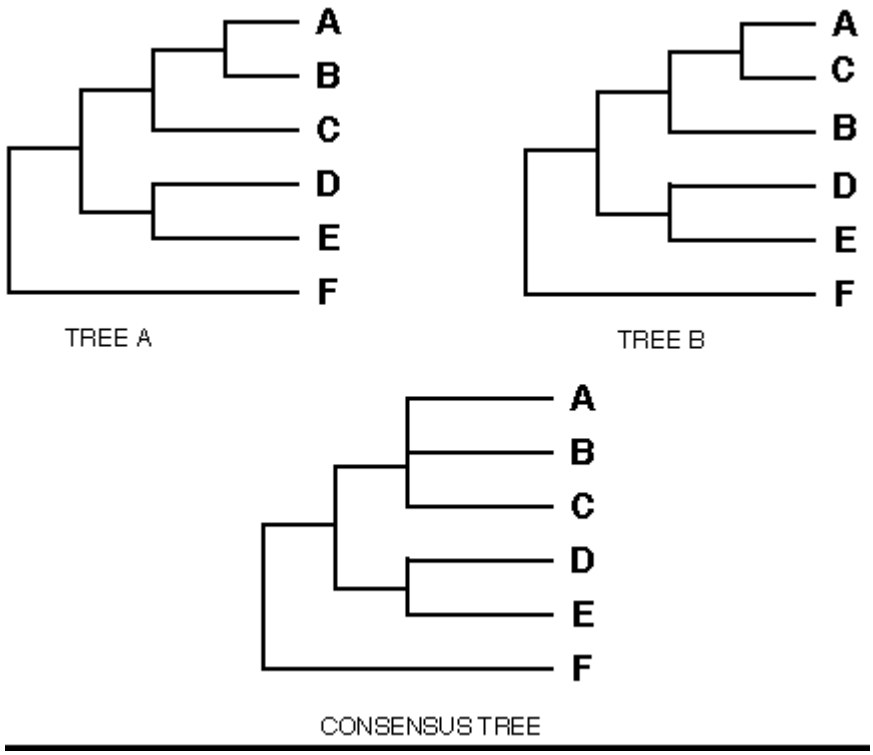


Figure3.5 : arbre consensus

10.2.2.5. Avantages et inconvénients de la parcimonie

Avantages :

- Méthode basée sur les caractères : méthode cladistique plutôt que phénétique.
- Méthode ne réduisant pas la séquence à un simple nombre.
- Méthode essayant de donner une information sur les séquences ancestrales.
- Méthode évaluant différents arbres.

Inconvénients :

- Méthode très lente par rapport aux méthodes basées sur les distances.
- Méthode n'utilisant pas toute l'information disponible (seuls les sites informatifs sont pris en compte)
- Méthode ne faisant pas de corrections pour les substitutions multiples

- Méthode ne donnant aucune information sur la longueur des branches
- Méthode connue pour être très sensible au biais des codons

10.2.2.6. Maximum de vraisemblance

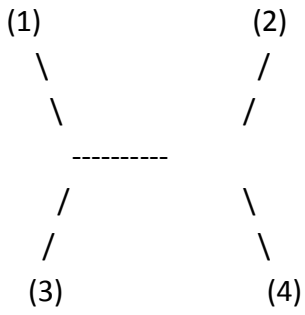
Cette méthode de reconstruction phylogénétique évalue, en termes de probabilités, l'ordre des branchements et la longueur des branches d'un arbre sous un modèle évolutif donné.

Programme DNAML de Phylip.

Exemple :

1					j
C	G	A	G	A	**C**
A	G	C	G	A	**C**
A	G	A	T	T	**A**
G	G	A	T	A	**G**

A partir des 4 séquences ci-dessus, on veut estimer la probabilité que l'arbre A soit le bon, sous le modéle choisi.



Arbre A

La vraisemblance de l'arbre est en général indépendante de la position de la racine, on peut donc l'enraciner de manière arbitraire :

1 2 3 4

La vraisemblance au site j :

La vraisemblance pour un site j est la somme des probabilités de toutes les possibilités de reconstruction de l'état ancestral sous le modèle choisi.

La vraisemblance de l'arbre A est en général évaluée en sommant les logs des vraisemblances pour chaque site (la somme des probabilités est trop faible).

L'arbre du maximum de vraisemblance est celui avec la vraisemblance la plus élevée.

10.3. Les modèles évolutifs

Les probabilités obtenues à chaque site dépendent du modèle choisi et dans le modèle le plus simple

- on suppose que la probabilité de chaque changement est indépendante des changements précédents (Modèle de Markov).
- on suppose que les probabilités de substitution ne changent pas au cours du temps (le long de l'arbre).
- on suppose les changements réversibles : P (A -> T) = P(T -> A).

On peut introduire d'autres paramètres dans le modèle afin d'accroître son réalisme :

- des taux de substitutions différents pour chaque remplacement (matrice 4*4 pour l'ADN ou matrice de substitution)
- une correction pour le nombre de sites susceptibles de muter et des taux de substitutions variables pour ces sites.
- un taux de variation différents pour chaque site : on peut par exemple utiliser une distribution statistique (distribution gamma)

Il faut savoir que plus on introduit de paramètres, plus le calcul sera long et plus il y aura une accumulation de petites erreurs : il vaut mieux utiliser un modèle simple.

Le maximum de vraisemblance est une bonne méthode de reconstruction phylogénétique mais il faut que le modèle de départ corresponde bien aux données. Pour estimer les paramètres, on peut utiliser une méthode plus rapide et utiliser l'arbre obtenu pour fixer les paramètres de départ.

Cette méthode n'est utilisable que si on a un petit nombre de séquences.

11.Conclusion

Nous avons dans ce chapitre présenté l'analyse des données qui est une famille de méthode statistiques dont les principales caractéristiques sont d'être multidimensionnelles et descriptives la notion de regroupement hiérarchique recouvre différentes méthodes de clustering est-à-dire de classification par algorithme de classification. Et Afin de déterminer les similitudes et liens entre éléments d'un arbre (en général des séquences), plusieurs méthodes ont été suivies citer dans ce chapitre.

CHAPITRE IV :

L'algorithme CLUSTAL pour l'alignement multiple de séquence

1. Introduction

Dans les chapitres précédents, nous avons parlé du problème de l'alignement multiple des séquences et nous avons vu que le problème est NP-Complet.

Il existe plusieurs algorithmes pour résoudre le problème et sont classés en trois catégories :

- Les algorithmes exacts : ils ne peuvent être utilisés que pour un nombre très limité de séquences à cause de la quantité de mémoire nécessaires.
- Les algorithmes itératifs : ils donnent en général des résultats acceptables dans un temps de calculs très importants.
- Les algorithmes progressifs : ils donnent en général des résultats acceptables dans un temps limité.

Nous avons implémenté dans ce mémoire une approche progressive est appelé l'algorithme CLUSTAL pour résolution de problème alignement multiple des séquences.

2. Abstrait

A été testé CLUSTAL W dans une grande variété de situations, et il est capable de gérer certains problèmes très difficiles d'alignement des protéines. Si l'ensemble des données est constitué de séquences suffisamment proches les unes des autres pour que les premiers alignements soient précis, alors CLUSTAL W trouvera généralement un alignement très proche de l'idéal.

3. La Méthode ClustalW

Clustalw est, de facto, le programme de référence pour calculer un alignement multiple progressif global. Bien qu'étant le standard, Clustalw présente un système de score compliqué. Les auteurs se sont inspirés des observations de [40] pour fixer les pénalités d'insertion et de délétion. Celles-ci d´dépendent de nombreux paramètres tels que la matrice de substitution, la similarité des s´séquences, la longueur des s´séquences, la différence de longueurs des s´séquences, les positions des brèches, la présence de brèches, la proximité d'une brèche, la présence de régions hydrophiles... Le choix de ces paramètres empiriques est le point critique de ce programme. Le coût d'insertion d'un gap est souvent trop important et le programme a tendance à mésapparier au d´dépens des insertions.

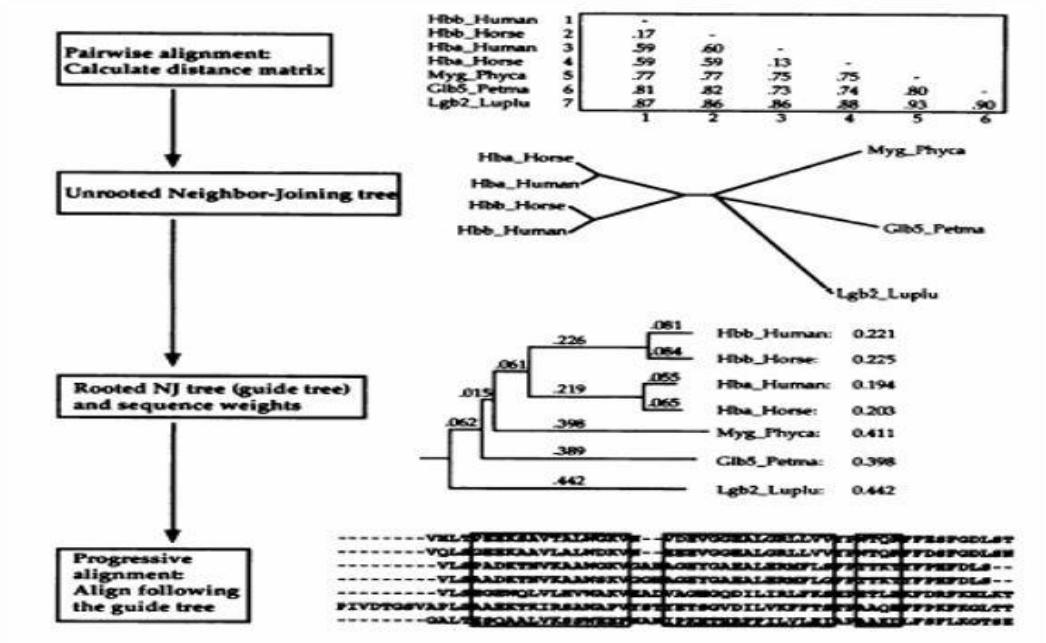


Figure 4.1 : Le déroulement de l'algorithme de ClustalW

Depuis [Feng and Doolittle, 1987] et [Thompson et al., 1994], aucune amélioration n'avait été proposé au niveau du temps de calcul ; hormis [Li, 2003] qui a proposé une parallélisation de ClustalW. Depuis les années 2000, de nouvelles m´méthodes ont vu le jour et ont apporté de nombreuses solutions : rapidité, système de score plus simple... Dans la section suivante, nous d´écrirons brièvement ces m´méthodes en mettant en avant leurs particularités.

Etape	option	paramètre
Alignement par paire	-PWMATRIX -PWDNAMATRIX	Choix de la matrice de substitution (protéines ou ADN)
	-PWGAPOPEN -PWGAPEXT	Choix des pénalités d'ouverture et d'extension de gaps
Alignement multiple	-PWMATRIX -PWDNAMATRIX	Choix de la matrice de substitution (protéines ou ADN)
	-PWGAPOPEN -PWGAPEXT	Choix des pénalités d'ouverture et d'extension de gaps
	-MAXDIV	pourcentage d'identité entre les séquences

Tableau 4.1: Tableau regroupant différents paramètres externes du programme de référence ClustalW

4. Les étapes de ClustalW

La première étape de ClustalW consiste à aligner les paires de séquences afin de déterminer la matrice des distances. ClustalW utilise des matrices de substitutions différentes pour la programmation dynamique à des moments différents de l'alignement. Les matrices changent selon la divergence ou la convergence des deux séquences à aligner. L'avantage est que les séquences divergentes sont plus ou moins bien alignées.

Dans la deuxième étape, ClustalW utilise la méthode N.J [35] pour construire un arbre guide et calculer les poids des séquences.

Pendant la troisième étape : alignement progressif proprement dit, ClustalW n'affecte pas la même valeur de pénalité d'un gap quel que soit sa position dans la séquence mais essayent de distinguer entre les gaps du début, du milieu et de la fin de la séquence.

Dans ClustalW, il y a une grande étude et des nouvelles propositions sur la manière de faire changer les valeurs affectées à un gap selon sa position dans une séquence ou dans un alignement de séquences.

Une particularité de ClustalW est qu'il possède une interface graphique conviviale contrairement aux autres méthodes.

5. Exemple du ClustalW

-Exemple 1 :

Alignement progressif avec CLUSTALW des 5 séquences suivantes :

>Seq1 ATCTCGAGA

>Seq2 ATCCGAGA

>Seq3 ATGTCGACGA

>Seq4 ATGTCGACAGA

>Seq5 ATTCAACGA

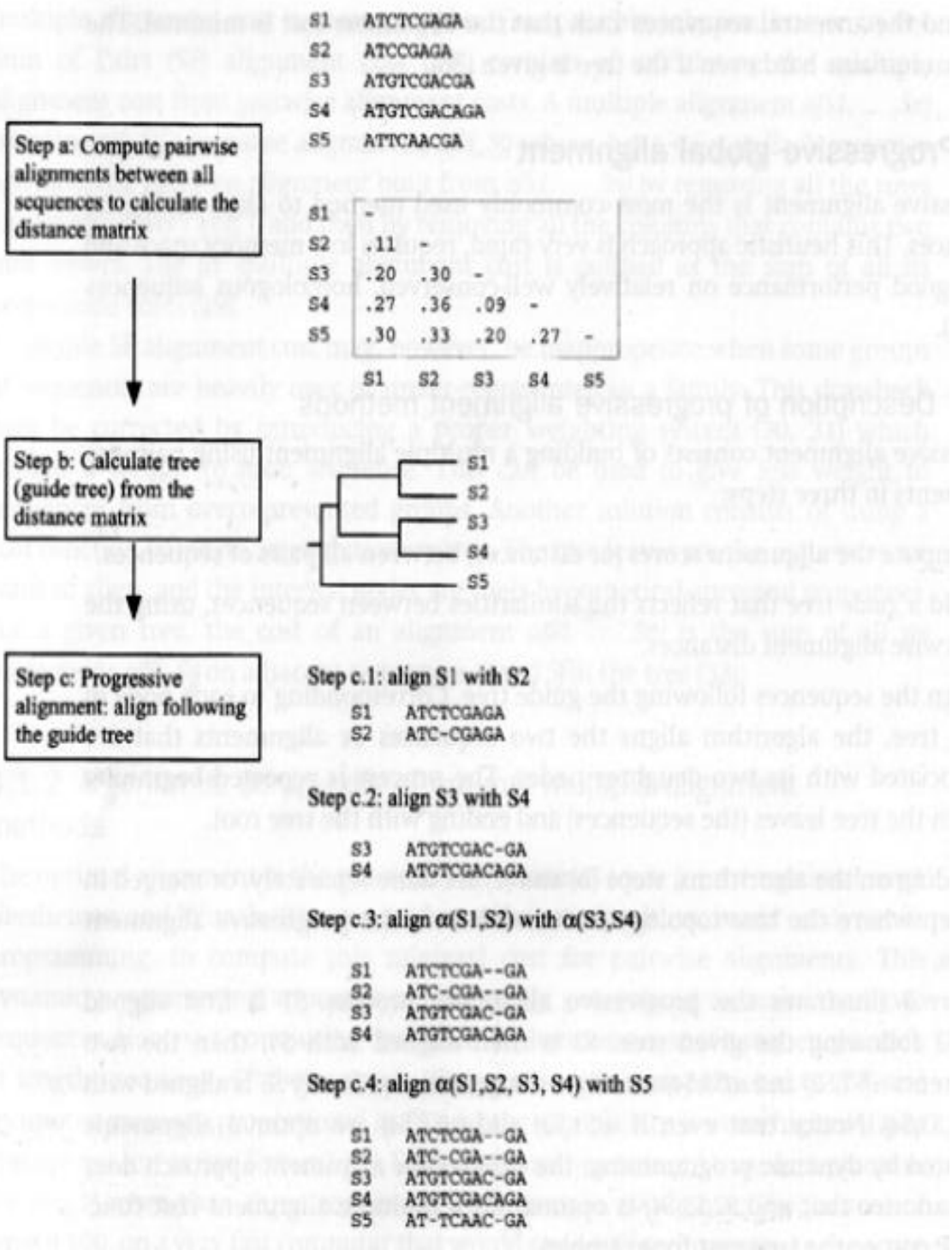


Figure 4.2 : processus d'alignement progressif

-Exemple 2 :

Étape 1 : alignement par paire de toutes les séquences

```
Hbb_human    1 VHLTPEEKSAVTALWGKVNVDEVGGEALGRLLVVYPWTQRFFESFGDLST ...
                 | |. |||.|| ||| ||| :|||||||||||||||||||||:||||||
Hbb_horse    2 VQLSGEEKAAVLALWDKVNEEEVGGEALGRLLVVYPWTQRFFDSFGDLSN ...

Hbb_human    1 LTPEEKSAVTALWGKV..NVDEVGGEALGRLLVVYPWTQRFFESFGDLST ...
                 |.| :|. | | ||||  .  | | ||| |: . :| |. :|  | |||
Hba_human    3 LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF.DLS. ...

Hba_human    3 LSPADKTNVKAAWGKVGAHAGEYGAEALERMFLSFPTTKTYFPHF.DLSH ...
                 ||  :|  | | | ||     | | ||| |: . :| |. :|  | |||.
Hbb_horse    2 LSGEEKAAVLALWDKVNEE..EVGGEALGRLLVVYPWTQRFFDSFGDLSN ...
```

Figure 4.3 : alignement par paire de toutes les séquences.

Les alignements peuvent être obtenus :

- Par des méthodes globales ou locales
- Par programmation dynamiques ou des méthodes heuristiques (non optimales)

Étape 2 : construction de la matrice de distance

Dans Clustalw:

Distance entre deux séquences = 1 - $\frac{\text{Nb de résidus identiques}}{\text{Nb de résidus comparés}}$

Ex : Hbb_human vs Hbb_horse = 83% identité = distance de 17%

Hbb_human	1	-					
Hbb_horse	2	.17	-				
Hba_human	3	.59	.60	-			
Hba_horse	4	.59	.59	.13	-		
Myg_phyca	5	.77	.77	.75	.75	-	
Glb5_petma	6	.81	.82	.73	.74	.80	-
Lgb2_lupla	7	.87	.86	.86	.88	.93	.90	-
		1	2	3	4	5	6	7

Figure 4.4 : la matrice de distance

Étape 3 : construction de l'arbre guide

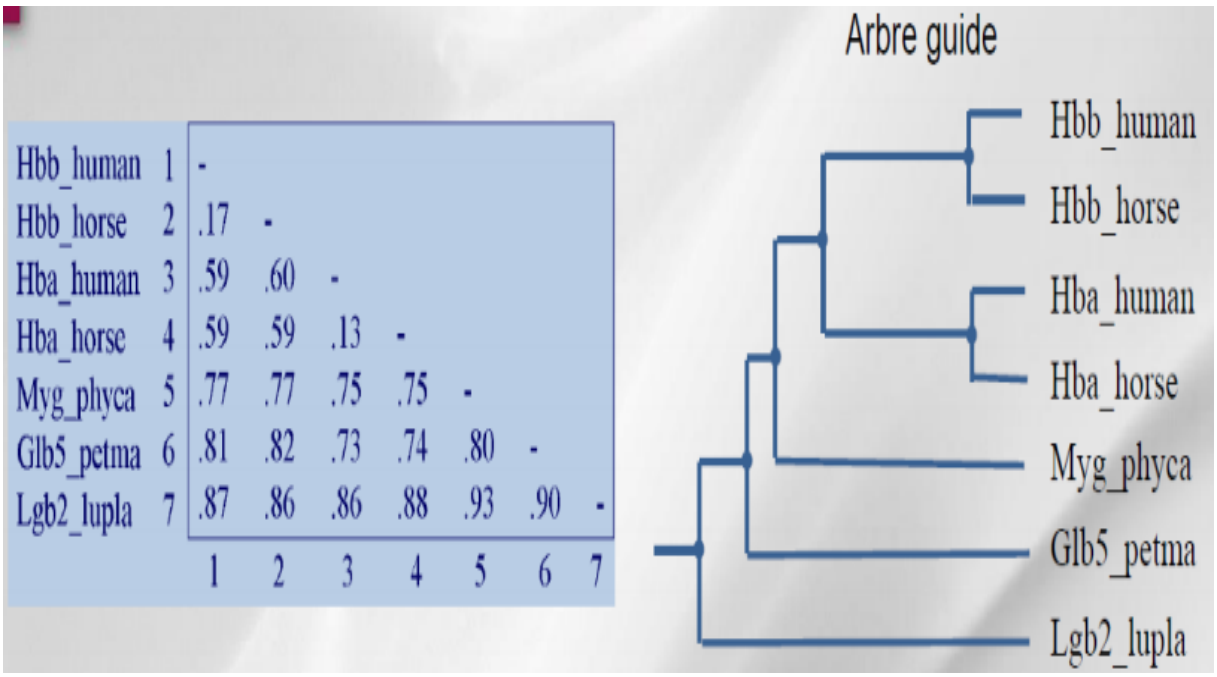


Figure 4.5 : l'arbre guide

1. Joint les deux séquences les plus proches
2. Calcul à nouveau les distances et joint les deux séquences les plus proches ou les noue
3. Répétition de l'étape 2 jusqu'à ce que toutes les séquences soient jointes

Étape 4 : Alignement progressif selon l'ordre des branches de l'arbre guide

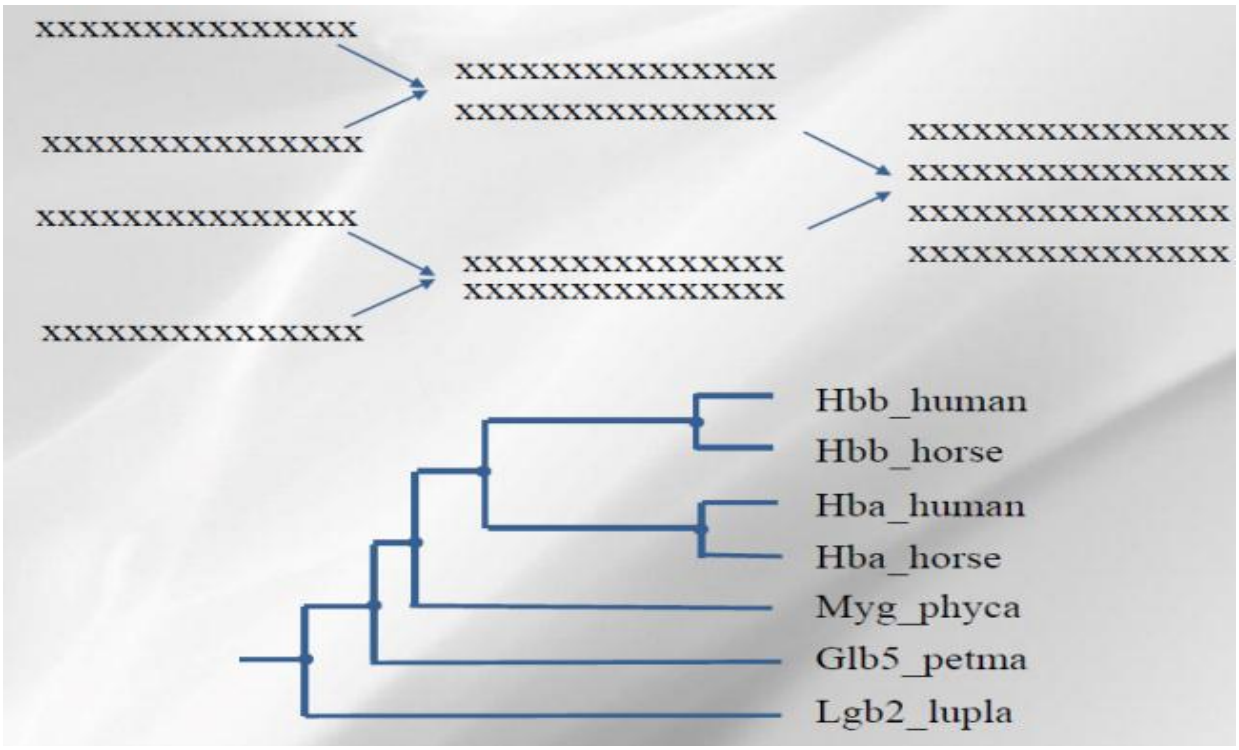


Figure 4.6 : Alignement progressif selon l'ordre des branches de l'arbre guide

6. Conclusion

Dans ce chapitre, nous avons présenté l'approche progressive d'alignement multiple des séquences ClustalW, ses principes et ses étapes. Ensuite nous avons donnée en détail des exemples de cet algorithme. Le chapitre suivant sera consacré à la réalisation et les résultats expérimentaux de notre étude en particulier pour l'algorithme développés CLUSTAL.

Chapitre 5 :

Implémentation et bilan

1. Introduction

Nous avons introduit dans les chapitres précédents, l'importance de l'alignement multiple de séquences (Multiple Sequence Alignment : MSA). Pendant la dernière décennie, plusieurs méthodes ont été décrites dans ce domaine. Nous présentons la méthode progressif Clustal.

Dans ce chapitre, nous commençons d'abord par une description des outils et langage de programmation utilisés dans ce travail. Ensuite nous présentons l'application réalisée et la façon de sa manipulation.

2. Les outils et langage de programmation

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les technologies HTML et CSS, JavaScript est parfois considéré comme l'une des technologies cœur du World Wide Web. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript dédié pour l'interprète.

C'est un langage orienté objet à prototype, c'est-à-dire que les bases du langage et ses principales interfaces sont fournies par des objets qui ne sont pas des instances de classes, mais qui sont chacun équipés de constructeurs permettant de créer leurs propriétés, et notamment une propriété de prototypage qui permet de créer des objets héritiers personnalisés. En outre, les fonctions sont des objets de première classe. Le langage supporte

le paradigme objet, impératif et fonctionnel. JavaScript est le langage possédant le plus large écosystème grâce à son gestionnaire de dépendances npm, avec environ 500 000 paquets en août 2017ter [32].

Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement la page, de mettre en forme le contenu, de créer des formulaires de saisie, d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS). HTML est inspiré du Standard Generalized Markup Language (SGML). Il s'agit d'un format ouvert [30].

Les **CSS**, Cascading Style Sheets (feuilles de styles en cascade), servent à mettre en forme des documents web, type page HTML ou XML. Par l'intermédiaire de propriétés d'apparence (couleurs, bordures, polices, etc.) et de placement (largeur, hauteur, côte à côte, dessus-dessous, etc.), le rendu d'une page web peut être intégralement modifié sans aucun

code supplémentaire dans la page web. Les feuilles de styles ont d'ailleurs pour objectif principal de dissocier le contenu de la page de son apparence visuelle [30]. Ceci permet :

- De ne pas répéter dans chaque page le même code de mise en forme.
- D'utiliser des styles génériques, avec des noms explicites (par exemple un style encadré pour du texte ou des images).
- De pouvoir changer l'apparence d'un site web complet en ne modifiant qu'un seul fichier.
- De faciliter la lecture du code de la page.

La puissance et de l'intérêt des CSS peut être démontrée en modifiant radicalement l'apparence d'une page, sans changer son code HTML d'un iota... Bref les CSS permettent de gagner en productivité et en maintenabilité des sites web, tout en offrant des possibilités graphiques incontestables.

3. Interface

3.1. Menu

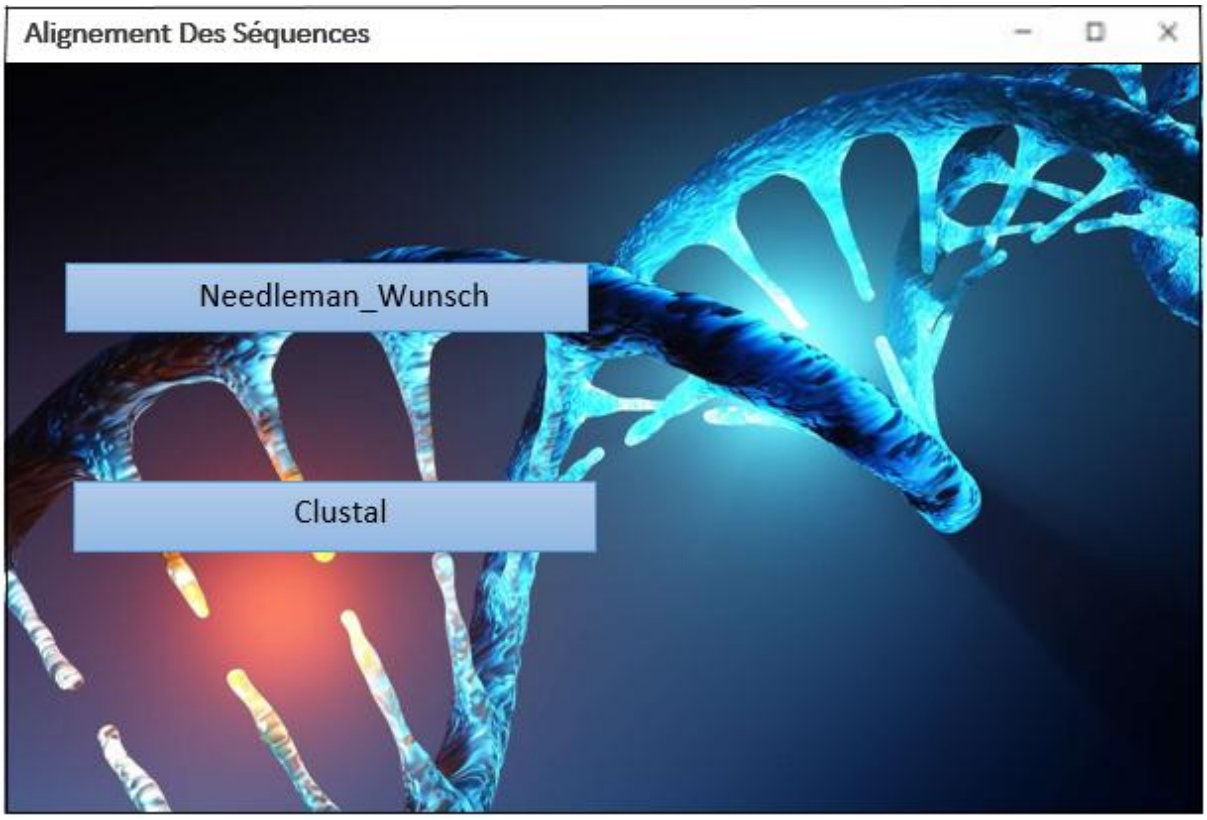


Figure 5.1 : menu principale

Cette figure montre le menu principal « la première fenêtre » de l'application. Elle se compose de deux boutons : le premier « Needleman_wunsch » donne accès à la fenêtre qui donne l'alignement global et le deuxième « Clustal » à l'alignement multiple des séquences.

3.2. Neesleman_Wunsch

La Figure (5.2) est la première fenêtre qui s'affiche en cliquant sur le bouton « Needleman_Wunsch ». Elle se compose de deux champs pour remplir les séquences, et un

champ pour les informations d'alignement global, et un bouton « Démarrer » pour lancer l'algorithme. Le résultat de l'algorithme s'affiche sur l'écran comme table finale indique à la « figure 5.3 ».

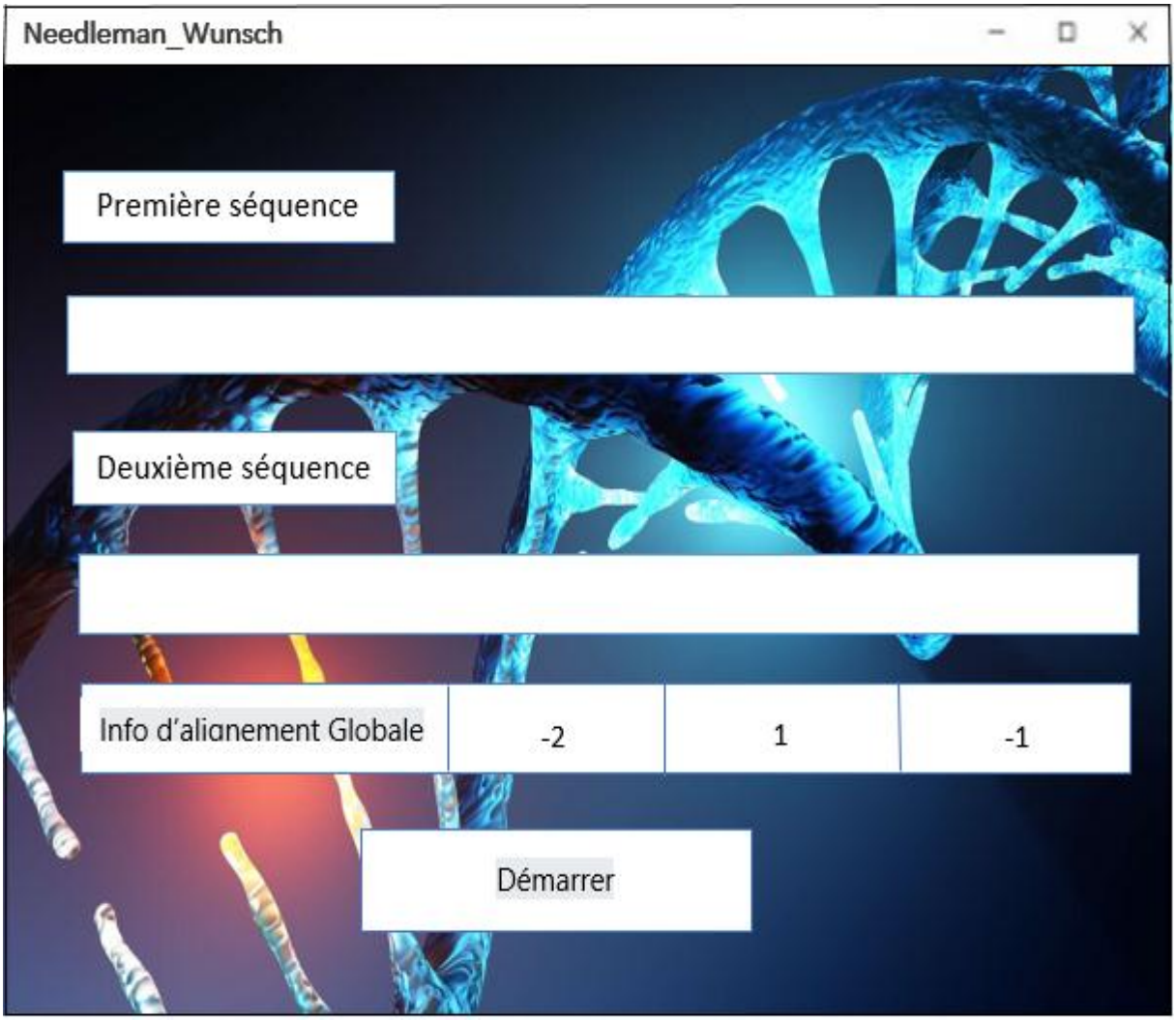


Figure 5.2 : Alignement global « Needleman_Wunsch »

Table finale

#	-	C	G	A	C	C	A	T	T	G	T	A	G	C	T	A	C	C	T	G
-	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26	-28	-30	-32	-34	-36	-38
C	-2	1	-1	-3	-5	-7	-9	-11	-13	-15	-17	-19	-21	-23	-25	-27	-29	-31	-33	-35
G	-4	-1	2	0	-2	-4	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26	-28	-30	-32
A	-6	-3	0	3	1	-1	-3	-5	-7	-9	-11	-13	-15	-17	-19	-21	-23	-25	-27	-29
G	-8	-5	-2	1	2	0	-2	-4	-6	-6	-8	-10	-12	-14	-16	-18	-20	-22	-24	-26
C	-10	-7	-4	-1	2	3	1	-1	-3	-5	-7	-9	-11	-11	-13	-15	-17	-19	-21	-23
C	-12	-9	-6	-3	0	3	2	0	-2	-4	-6	-8	-10	-10	-12	-14	-14	-16	-18	-20
A	-14	-11	-8	-5	-2	1	4	2	0	-2	-4	-5	-7	-9	-11	-11	-13	-15	-17	-19
T	-16	-13	-10	-7	-4	-1	2	5	3	1	-1	-3	-5	-7	-8	-10	-12	-14	-14	-16
T	-18	-15	-12	-9	-6	-3	0	3	6	4	2	0	-2	-4	-6	-8	-10	-12	-13	-15
G	-20	-17	-14	-11	-8	-5	-2	1	4	7	5	3	1	-1	-3	-5	-7	-9	-11	-12
T	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	6	4	2	0	-2	-4	-6	-8	-10
A	-24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	7	5	3	1	-1	-3	-5	-7
G	-26	-23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	8	6	4	2	0	-2	-4
C	-28	-25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	9	7	5	3	1	-1
T	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	10	8	6	4	2
A	-32	-29	-26	-23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	13	11	9	7	5
C	-34	-31	-28	-25	-22	-19	-16	-13	-10	-7	-4	-1	2	5	8	11	14	12	10	8
T	-36	-33	-30	-27	-24	-21	-18	-15	-12	-9	-6	-3	0	3	6	9	12	13	13	11
G	-38	-35	-32	-29	-26	-23	-20	-17	-14	-11	-8	-5	-2	1	4	7	10	11	12	14

Tableau 5.1 : La table finale de l'algorithme needlemen_wunsch

La couleur bleue pour le chemin critique.

CGA-CCATTGTAGCTACCTG CGA-CCATTGTAGCTACCTG

CGAGCCATTGTAGCTA-CTG CGAGCCATTGTAGCTAC-TG

3.3. Clustal

L'accès à cette première fenêtre est effectué à partir du menu principal en cliquant sur le bouton « Clustal ».

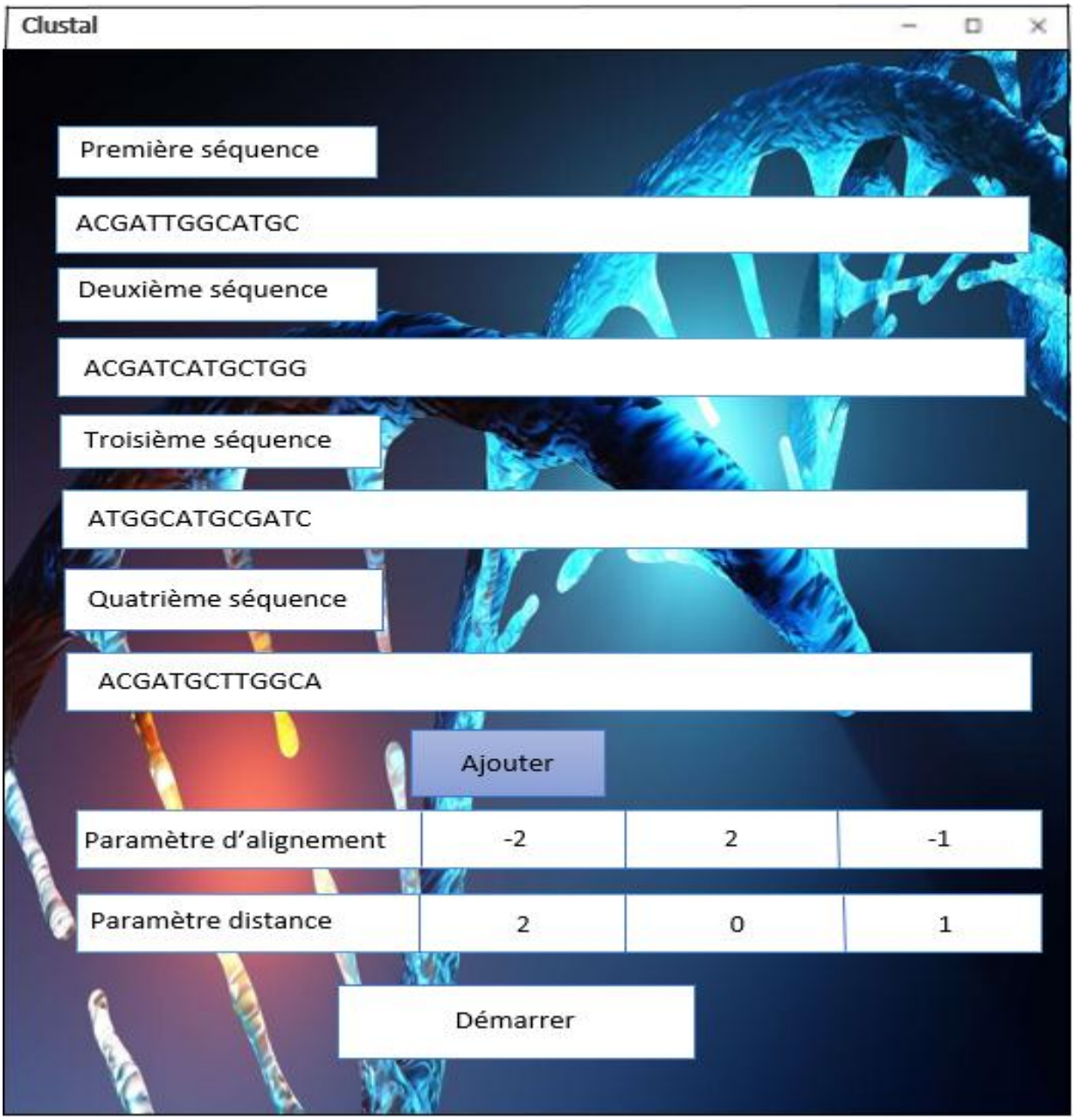


Figure 5.3 : Alignement multiple « Clustal »

La Figure (5.4) est la deuxième fenêtre qui s'affiche en cliquant sur le bouton « Clustal ». Elle se compose de des champs pour remplir les séquences, et un bouton « Ajouter » pour ajouter autre champ de séquence et deux champs pour les paramètres d'alignement et de distance, et un autre bouton « Démarrer » pour lancer l'algorithme.

Le résultat de l'algorithme s'affiche sur l'écran comme une arbre phylogénie indique à la « figure 5.5 ».

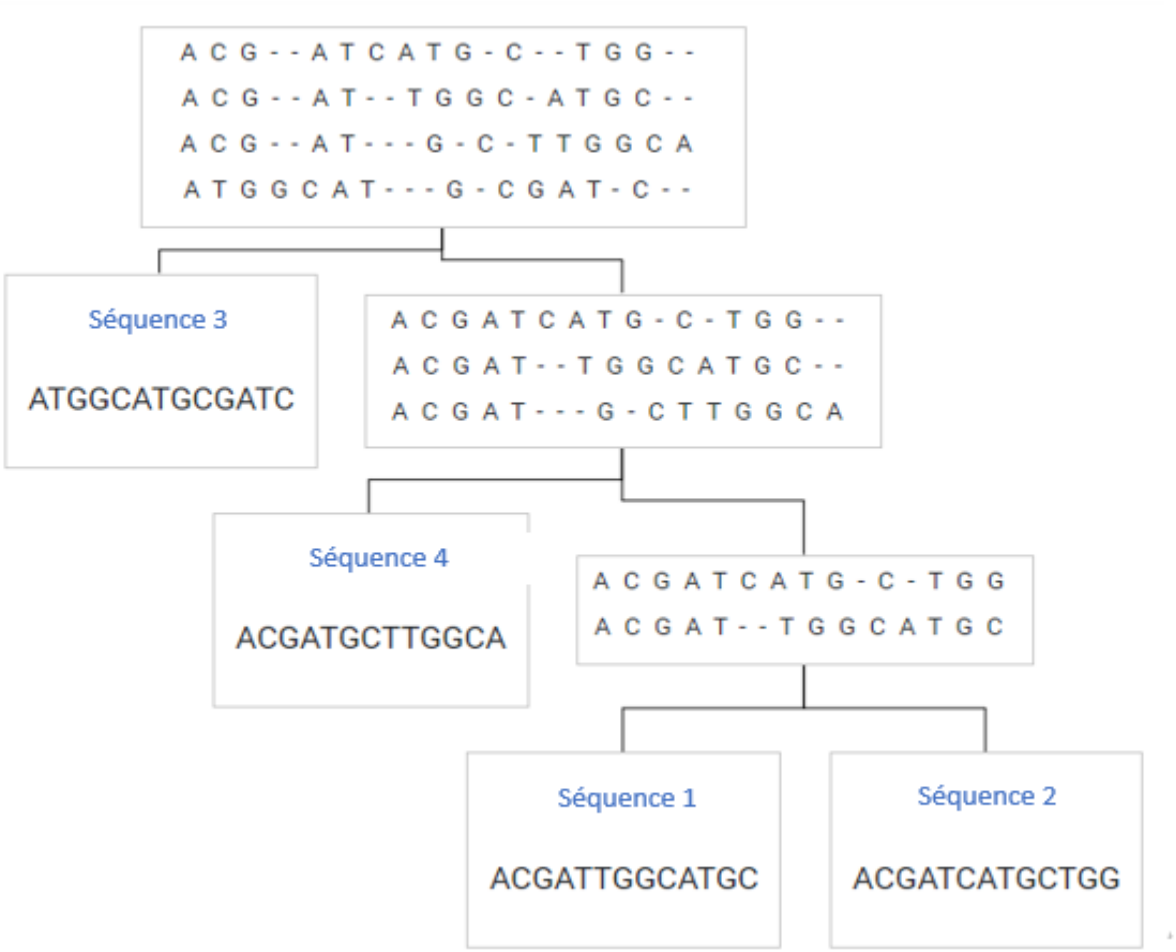


Figure 5.4 : L'algorithme Clustal avec arbre phylogénétique

4. Conclusion

Pour la résolution du problème d'alignement multiple des séquences, l'algorithme Clustal est la méthode plus utilisée pour l'alignement.

Dans ce chapitre nous présenter l'application réalisée et la façon de sa manipulation de cet algorithme.

Conclusion générale

Conclusion générale

Nous avons abordé dans ce mémoire une problématique très important en bio-informatique celui de les règles d'associations utilisées pour l'amélioration de l'alignement des séquences biologique

Ce problème a toujours été défini comme un problème mono-objectif, où les différentes méthodes développées cherchent à utiliser une seule fonction objectif.

Nous avons démontré l'incapacité de ces méthodes à identifier et évaluer un alignement multiple des séquences dans un temps de calcule minimal par l'algorithme CLUSTAL . Ainsi que l'algorithme CLUSTAL c'est une approche itérative.

Nous avons présenté dans ce mémoire, que cette Algorithme utilisant des règles simples pour diminuer l'espace de recherche des solutions ainsi que les méthodes utilisé

- Méthodes hiérarchiques qui basé sur le calcul des distances
  - Calcul des distances entre groupes d'individus
  - La méthode de saut minimal
  - La méthode de diamètre
  - La distance moyenne

Cette méthode n'est pas adaptée pour un grand nombre d'individus comme on ne définit pas a priori le nombre de classes.

- Méthodes phylogénétiques : Il existe plusieurs techniques de construction des arbres phylogénétiques, plus ou moins rapides et plus ou moins fiables, On peut être amené à chercher à optimiser plusieurs critères dans l'arbre : la distance, la parcimonie, ou la vraisemblance.
- Cette méthode est basée sur trois étapes :
- rechercher tous les arbres phylogénétiques possibles pour les différents taxons étudiés,
  - rechercher tous les arbres phylogénétiques possibles pour les différents taxons étudiés,
  - mesurer la longueur totale de chaque arbre,
  - sélectionner celui ou ceux qui présentent la longueur la plus petite.

Les arbres fournis par cette méthode sont non polarisés, cependant l'utilisation d'*out groups* (espèces externes aux groupes étudiés) permet dans un deuxième temps de polariser l'arbre.

C'est une méthode très lente si l'on génère tous les arbres possibles pour en calculer la parcimonie.

Résumé :

Les alignements multiples de séquences sont considérés comme une partie fondamentale des processus d'une multitude d'applications dans le domaine bio-informatique, qui sert à traiter automatiquement l'information biologique.

Dans cette étude destinée à la mémoire de fin d'étude, nous avons implémenté un Algorithme CLUSTAL pour trouver une solution au problème des alignements multiple de séquences, de sorte que tous les règlements de l'algorithme ont été appliqués sur un ensemble d'alignements. Ainsi, on a obtenu des résultats améliorés après maints nombre des répétitions.

Mot clé : Bio-informatique, Alignements Multiple de Séquence, CLUSTAL

Abstract:

The multiple sequence alignment are considered a fundamental part of the Processes of a multitude of applications in the bioinformatics field whose function is the automatic processing of biological information.

In this study, dedicated to the fulfillment of the dissertation, we have implemented an algorithm CLUSTAL in order to solve the problem of the multiple sequence alignment, in a way that all algorithm regulations were applied on a set of alignments. Thus, we have obtained improved results after many number of repetition.

Keywords: Bioinformatics, Multiple Sequence Alignment, Algorithm CLUSTAL.

Bibliographie

[1] Damiens IMBS et MOHAMED SAYED HASSAN, Bioinformatique travail d'étude, Univ de nice sophia antipolis.

[2] Jean-Claude Callen, Avec la collaboration de Roland Perasso," biologie cellulaire : des molécules aux organismes", Edition,2. Publisher, Dunod, 2005. ISBN, 2100492365, 9782100492367.

[3] Alberts et autres : B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, ”Molecular Biology of the Cell”, Garland Science, 4ème édition 2002 à travers le site de NCBI : http://www.ncbi.nlm.nih.gov/books

[4] Batzoglou : S. Batzoglou, “Sequence Alignment’ I : CS262 Winter 2004 : Lecture II, 2004

[5] Vert : J. P. Vert : « Introduction à la biologie moléculaire et à la bioinformatique » cours de Master Recherche M2, 2004/2005

[6] Nadira Benlahrache, " Optimisation Multi-Objectif Pour l’Alignement Multiple de Séquences ".

[7] Vincent Derrien," Heuristiques pour la résolution du problème d’alignement multiple ", Thése de doctorat, N˚d’ordre 885, 2008.

[8] Guillaume Lecointre, MNHN , Relations de parenté entre les êtres vivants , http://acces.ens-lyon.fr/biotic/evolut/parente/html/arbphyl.htm

[9] Bernstein FC, Koetzle TF, Williams GJ, Meyer Jr EF, Brice MD, Rodgers JR, Kennard O, Shimanouchi T, Tasumi M. « The Protein Data Bank: a computer-based archival file for macromolecular structures » J Mol Biol. 1977;112:535-542.

[10] Vincent Derrien, Jean-Michel Richer,Jin-Kao Hao." Plasma, un nouvel algorithme progressif pour l’alignement multiple de séquences ". LERIA - Université d’Angers, 2 Bd Lavoisier, 49045 Angers, France.

[11] NCBI , http://www.ncbi.nlm.nih.gov, consulté le 17/03/ 2018.

[12] V.I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and Reversals". Soviet Physics Doklady, 10:707–710,1966.

[13] S.MESHOUL, "Approche quantique évolutionnaire pour l'alignement multiple de séquences en bioinformatique" , Magistère, Université Mentouri de Constantine, 2005.

[14] M O Dayhoff,R.M.Schwartz and B C Orcutt, "A model of evolutionary change in proteins ", Atlas Protein Seq,Struct,vol.6 pp.345-362.1978.

[15] S Henikoff and J Henikoff."Amino acid substitution matrices from protein blocks",Proceedings of the National Academy of Sciences,No.89,pp.915-919, 1992.

[16] Batzoglou, 04. S.Batzoglou , “Sequence Alignment’ I: CS262 Winter 2004: Lecture II, 2004.

[17] Saitou et Nei, 87 : N. Saitou, and M. Nei, “The neighbor-joining method: a new method for reconstructing phylogenetic trees”. Mol. Biol. Evol., Vol. 4, pp. 406-425. 1987.

[19] Stoye et autres, J. Stoye, V, Moulton, and A. W. Dress, "DCA, an efficient implementation of the divide and conquer approach to simultaneous multiple sequence alignment", Comput. Appl. Biosc., Vol. 13, No. 6, pp. 625-631, 1997.

[20] D.F. Feng and R.F Doolittle. "Progressive sequence alignment as a prerequisite to correct phylogenetic trees". J. Mol. Evol., Vol. 25, pp.351-360, 1987.

[21] G. PIATETSKY-SHAPIRO, Data mining and knowledge discovery 1996 to 2005: overcoming the hype and moving from «university» to «business» and «analytics», Data mining and Knowledge Discovery, 15(1), 99-105.

[22] D. HAND, H. MANNILA et P. SMYTH, Principles of Data Mining,MIT Press, Cambridge, MA, 2001.

[23] W.R.Taylor,"Multiple sequence alignment by a pairwise algorithm". Comput Appl Biosci, Vol. 3, pp. 81-7. 1987.

[24] P. CABENA, P. HADJINIAN, R. STADLER, J. VERHEES et A. ZANASI, Discovering Data Mining: From Concept to Implementation, Prentice Hall, Upper Saddle River, NJ, 1998.

[25] The Gartner Group, www.gartner.com.

[26] J. HAN, M. KAMBER, Data Mining: Concepts and Techniques, Simon Fraser University, 2000.

[27] R. Chenna H Sugawara T Koike R Lopez TJ Gibson DG. Higgins JD Thompson," Multiple sequence alignment with the clustal series of programs", Nucleic Acids Research,2003.

[28] C Notredame and D.G. Higgins."Saga: Sequence alignment by genetic algorithm". Nucleic Acid Research, Vol.24,1996.

[29] J H Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.Lecture II, 2004.

[30] J. HAN, M. KAMBER, Data Mining: Concepts and Techniques, Second Edition, University of Illinois at Urbana-Champaign, 2006.

[31] M. J. BERRY, G. S. LINOFF, Data Mining Techniques For Marketing, Sales, and Customer Relationship, Management, Second Edition, 2004.

[32] M. J. BERRY, G. S. LINOFF, Mastering Data Mining : The Art and Science of Customer Relationship Management, 2000.

[33] D.T. LAROSE, Discovering Knowledge In Data : An Introduction to Data Mining, Central Connecticut State University, 2005.

[34] 13=O. R. ZAÏANE, Principles of Knowledge Discovery in Databases, CMPUT690, University of Alberta, 1999.

[35]19=Ph. PREUX, Fouille de données : Notes de cours, Université de Lille 3, 9 octobre 2008.

[36] Notredame,C, "Recent progress in multiple sequence alignment: a survey. Pharmacogenomics", 2002.

[37] G. DONG, J. PEI, Sequence Data Mining, Springer Edition, 2007.

[38] S. PRABHU, N. VENKATESAN, Data Mining and Warehousing, New Age International (P) Ltd., Publishers, New Delhi, 2007.

[39] 12 [Psychol.] *Classification des caractères* Classification.

[40]3 Jean-Pierre NAKACHE approche pragmatique de la classification.

[41] J. Stoye, V, Moulton, and A. W. Dress, « DCA, an efficient implementation of the divide and conquer approach to simultaneous multiple sequence alignment", Comput. Appl. Biosc., Vol. 13, No. 6, pp. 625-631, 1997.

[42]11Fatma Karem, Mounir Dhibi, Arnaud Martin Combinaison de classification supervisée et non-supervisée par la théorie des fonctions de croyance.

[43]23 PSY83A - Analyses multidimensionnelles et applications informatiques.