

# Table des matières :

Introduction générale :	1
Chapitre 1 : introduction à la bioinformatique	3
1.1 Introduction:	4
1.2 Notions de base de biologie :	4
1.2.1 ADN : acide désoxyribonucléique :	4
1.2.2 L'ARN : Acide Ribonucléique :	6
1.2.3 Les protéines:	6
1.2.3.1 La structure primaire :	7
1.2.3.2 Les structures secondaires :	9
1.2.3.3 La structure tertiaire :	11
1.2.3.4 La structure quaternaire :	12
1.3 Problèmes issus de la bio-informatique :	12
1.3.1 Prédiction de structures :	12
1.3.2 Alignement de séquences :	13
1.3.3 Phylogénie :	13
1.3.4 Recherche de motifs :	14
1.4 Les banques de données :	14
1.4.1 La PDB (Protein Data Bank) :	14
1.4.2 GenBank:	15
1.5 Représentation informatique, format de séquence :	16
1.5.1 Le format FASTA (ou format Pearson) :	16
1.5.2 Pdb format :	17
1.6 Conclusion :	18
Chapitre 2 : prédiction de structure secondaire	19
2.1 Introduction :	20
2.2 Les méthodes statistiques:	20
2.3 Les méthodes tenant compte des propriétés physico-chimiques des acides aminés :	23
2.4 La méthode du plus proche voisin :	23
2.5 Les chaînes de Markov cachées :	25

2.5.1	Principe :	25
2.5.2	Prédiction des structures secondaires par les HMMs :	26
2.6	Méthode d'apprentissage par réseaux de neurones :	26
2.7	Programmes de prédiction de structure secondaire et état de l'art :	27
2.7.1	Méthodologie :	27
2.7.2	État de l'art :	27
2.8	Conclusion :	28
Chapitre 3 : Notions de base des réseaux de Neurones artificiels.....		29
3.1	Introduction :	30
3.2	Le neurone: .....	30
3.2.1	Le modèle biologique: .....	30
3.2.2	Vers une simulation du neurone biologique :	31
3.2.3	Le modèle forme1 :	32
3.3	Les réseaux de neurones artificiels: .....	33
3.3.1	Différentes architectures de réseaux de neurones :.....	33
3.3.1.1	Réseau multicouche (feed-forward) :	33
3.3.1.2	Réseau à connexion locale :	34
3.3.1.3	Réseau à connexion complète: .....	34
3.3.2	L'information dans les réseaux de neurones: .....	35
3.3.3	L'apprentissage: .....	35
3.3.3.1	Apprentissage supervisé: .....	36
3.3.3.2	Apprentissage non supervisé: .....	36
3.4	Les réseaux de neurones à apprentissage supervise: .....	36
3.4.1	Le cas d'un neurone seul :	36
3.4.2	Le Perceptron Multicouches :	37
3.4.2.1	Structure du Perceptron Multicouches :	37
3.4.2.2	Apprentissage d'un PMC .....	37
3.5	L'apprentissage non supervisé des réseaux de neurones:.....	39
3.6	Conclusion :	39
Chapitre 4 : adaptation et implementation .....		40
4.1	Introduction :	41
4.2	Présentation des outils d'implémentation :.....	41

4.2.1	Langage Java :	41
4.2.2	Netbeans :	41
4.3	Réseaux de neurones pour la prédiction des structures secondaires :	42
4.4	Description la d'algorithme :	43
4.4.1	Codage :	43
4.4.2	Réseaux de neurone et l'apprentissage :	44
4.3.3	Décodage :	44
4.5	Expérimentation des résultats :	45
4.5.1	Guide pour utiliser l'application :	45
4.5.2	Evaluation des résultats:	46
4.5.3	Présentation des résultats en fonction de quelques contraintes :	46
4.6	Conclusion :	48
	Conclusion générale :	50
	BIBLIOGRAPHIE :	51

## Liste de figure :

Figure 1.1 : Représentation de la double hélice d'ADN .....	5
Figure 1.2 : Formule développée d'une protéine de n acides aminés. ....	6
Figure 1.3 : Séquence primaire d'une protéine. ....	7
Figure 1.4 : Formulation d'une liaison peptidique. ....	9
Figure 1.5 : Le diagramme de Ramachandran .....	9
Figure 1.6 : hélice .....	10
Figure 1.7: Feuillet $\beta$ .....	11
Figure 1.8 : Différents types de coudes .....	11
Figure 1.9 : Evolution du nombre d'entrée dans la banque de structures 3D, la PDB .....	15
Figure 1.10 : un exemple FASTA format .....	17
Figure 2.1 : Classification des acides aminés selon les propriétés Physicochimiques .....	23
Figure 2.2 : Prédiction de la structure protéique par un l'algorithme des plus proches voisin .....	24
Figure 2.3 : Modèle contraint complet à 21 états cachés .....	25
Figure 2.4 : Chaîne de Markov caché pour la méthode Zheng .....	26
Figure 3.1 : Un neurone biologique et ses principaux composants .....	31
Figure 3.2 : schéma d'un neurone formel .....	32
Figure 3.3 : quelques fonctions d'activation.....	33
Figure 3.4 : Réseau multicouche.....	34
Figure 3.5 : Réseau à connexion locale .....	34
Figure 3.6 : Réseau à connexion complète .....	35

Figure 3.7 : Architecture d'un Perceptron Multicouches à une seule couche cachée .....	37
Figure 4.1 : Netbeans .....	41
Figure 4.2 : Modèle en couche de réseaux de neurones .....	42
Figure 4.3 : schéma représenté l'algorithme de prédiction. ....	43
Figure 4.4 : saisir la séquence de Protéine .....	45
Figure 4.5 : saisir le degré d'apprentissage et la taille de fenêtre .....	45
Figure 4.6 : la troisième partie .....	46
Figure 4.7 : la relation entre le degré d'apprentissage et la qualité des résultats.....	47
Figure 4.8 : la relation entre la taille de la fenêtre et la qualité des résultats .....	48

**s**

## **Introduction générale :**

Depuis bien longtemps les biologistes ont eu le besoin de faire des calculs sur les données à fin de pouvoir établir des nouvelles lois biologiques .Cependant, avec le développement des ordinateurs puissants et la grande disponibilité des données biologiques (des séquences d'ADN, d'ARN ou de protéines), une nouvelle discipline est apparue appelée la bio-informatique.

La bio-informatique est une discipline récente, qui fait appel aux compétences de la plupart des disciplines scientifiques pour lesquelles il existe déjà des méthodes permettant de résoudre des problèmes analogues. Il y a principalement les mathématiques et l'informatique, mais également dans certains cas la physique ou la chimie. La bio-informatique regroupe donc une partie de chacun de ces domaines, ainsi que la biologie elle-même. La bio-informatique traite différents problèmes parmi lesquelles nous citons : la phylogénie, la recherche de motifs, l'alignement de séquences et la prédiction des structures.

Le problème de la prédiction de structure secondaire d'une protéine à partir de sa seule séquence en acides aminés est un problème très difficile, il existe déjà des méthodes permettant de traiter ce problème parmi lesquelles nous citons : les méthodes statistiques, la méthode des plus proches voisins, la méthode de Chaînes de Markov, la méthode d'apprentissage par réseaux de neurones

Dans ce mémoire Nous avons essayé d'adapté et implémenté la méthode d'apprentissage par réseaux de neurones pour résoudre ce problème.

Notre mémoire est organisé en quatre chapitres. Le premier contient une introduction à la bio-informatique, pour cela nous présentons quelques notions de biologie moléculaire et les différents thèmes de la bio-informatique. Dans Le deuxième chapitre nous allons présenter les principales méthodes pour traiter le

problème de la prédiction de structure secondaire. Le troisième contient une introduction aux réseaux de neurones qui constituent actuellement un des outils les plus efficaces pour le traitement des problèmes de classification. Dans Le dernier chapitre nous allons présenter les outils d'implémentation que nous avons exploitée pour l'algorithme et voir comment adapter le réseau de neurone pour la prédiction des structures secondaires de protéine.



## Chapitre 1

---

# INTRODUCTION A LA BIOINFORMATIQUE

---



### 1.1 Introduction:

La Bio-informatique est un domaine de recherche qui analyse et interprète des données biologiques, au moyen de méthodes informatiques

La Bio-informatique aussi appelé en anglais « computational biology » ou « in silico biology » regroupe sous le même mot, 2 approches. La première approche correspond à la sensibilité plus informatique du domaine et consiste en l'élaboration d'algorithmes et le développement de programmes pour extraire l'information biologique. La seconde approche correspond à la bio-analyse dont le but est centré sur l'analyse de ces données et leur signification dans un contexte biologique [1].

La Bio-informatique doit également son développement à la compréhension des objets biologiques utilisant les techniques de l'informatique et de la mathématique pour servir l'être vivant. Par exemple analyser les séquences d'ADN et des protéines, détecter les codes circulaires dans les séquences, ... [1].

Dans ce chapitre nous commençons par une brève introduction à la biologie moléculaire en citant les notions biologiques de base nécessaires pour un bio informaticien. Ensuite, nous donnerons une définition à la bio-informatique ainsi que quelques champs d'applications de cette dernière. Le reste du chapitre sera consacré à la banque de données biologiques populaires et usuelles, et en termine par représentation informatique des données biologique.

### 1.2 Notions de base de biologie :

Afin de pouvoir donner des solutions informatiques efficaces aux problèmes biologiques, il faut bien avoir une certaine connaissance biologique. C'est pourquoi nous commençons par définir quelques notions de base de biologie les plus utilisées en bio-informatique.

#### 1.2.1 ADN : acide désoxyribonucléique :

C'est une molécule responsable de la transmission de l'information génétique héréditaire de génération en génération, présenté dans toutes les cellules vivantes de l'organisme, représenté par un long fil constitué par une séquence précise d'unité

élémentaires (nucléotides). Chaque nucléotide est composé d'un désoxyribose (sucre), d'un phosphate, et d'une base azotée, à partir de laquelle on peut distinguer entre les nucléotides [2].

Il existe 4 bases azotées différentes : Adénine (notée A), thymine (notée T), la Cytosine (notée C), et la guanine (notée G). La structure originelle de l'ADN a été caractérisée en 1953 par Watson et Crick, formée de deux brins complémentaires enroulés en hélice nommée la structure en "double hélices" [3].

La séquence de base dans l'ADN détermine la séquence des acides aminés dans les protéines, cependant l'ADN se transcrit en ARN (Acide Ribo Nucléique), un ARN particulier nommé ARNm (ARN Messenger) est ensuite traduit en protéine.

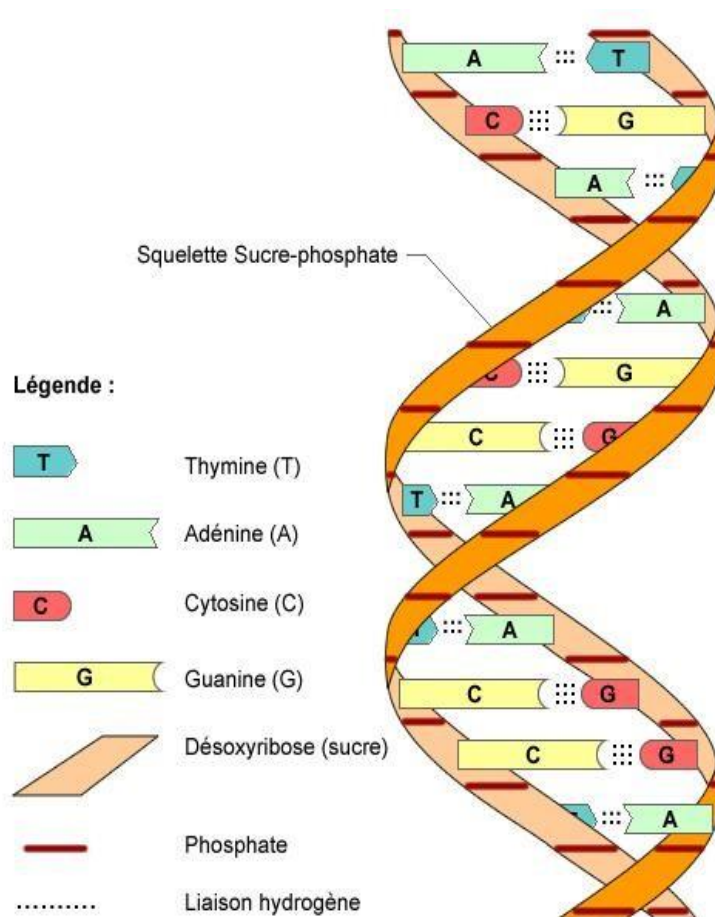


Figure 1.1 : Représentation de la double hélice d'ADN [4].

### 1.2.2 L'ARN : Acide Ribonucléique :

L'ARN est obtenu à partir d'un des deux brins d'ADN. A chaque nucléotide de la séquence d'ADN va correspondre le nucléotide complémentaire, sauf pour l'Adénine. En effet, pour l'ARN, le complémentaire de l'Adénine n'est pas la Thymine mais l'Uracile [5].

La transcription est le mécanisme qui permet de dupliquer un brin d'ADN sous forme d'ARN [5].

L'ARN est subdivisé en trois catégories, chacune ayant un rôle différent [5] :

- L'ARN de transfert (ARNt) est utilisé dans la phase de traduction de l'ARN en protéines.
- L'ARN ribosomique (ARNr) forme une trame sur les ribosomes, afin de permettre aux protéines synthétisées par les ribosomes de se fixer.
- L'ARN messenger (ARNm) est utilisé chez les eucaryotes pour véhiculer l'information génétique du noyau vers le cytoplasme (substance de la cellule qui entoure le noyau).

### 1.2.3 Les protéines:

Une protéine est un polymère dont les unités monomériques (appelés aussi résidus) sont les acides aminés unis par des liaisons peptidiques (figure 1.2). La conformation (c'est-à-dire le repliement) qu'adopte une protéine au sein de la cellule est appelée conformation native. C'est cette conformation unique qui lui assure ses propriétés spécifiques : fonctions enzymatiques et mécaniques, stabilité thermique... [6].

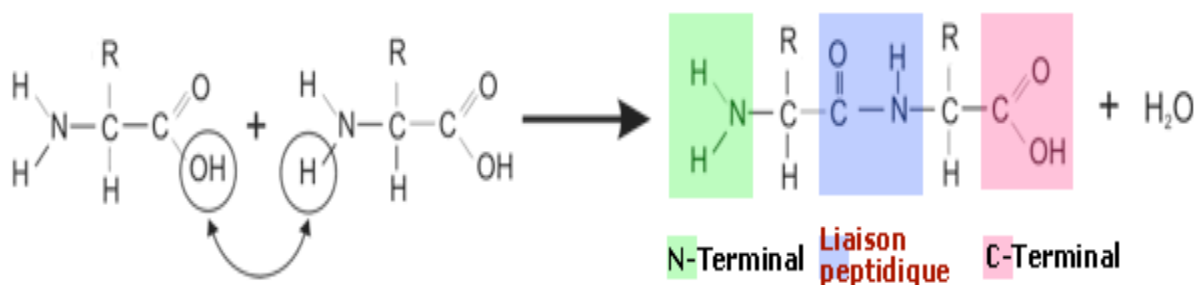


Figure 1.2 : Formule développée d'une protéine de n acides aminés. Les Ri désignent les Différentes chaînes latérales des résidus [7].

La traduction est le mécanisme qui permet de synthétiser sous forme de protéine l'ARNm obtenu par transcription.



La protéine possède quatre structures : primaire, secondaire, tertiaire et quaternaire :

### 1.2.3.1 La structure primaire :

La structure primaire est l'ordre d'enchaînement des acides aminés de la chaîne protéique. On nomme la liste des résidus en commençant par la terminaison amine (ou ammonium) et en terminant par le résidu portant la fonction acide carboxylique (ou carboxylate). Le premier résidu est alors nommé N-terminal et le dernier C-terminal. Le code à une lettre des acides aminés est alors très pratique pour décrire les protéines [6].

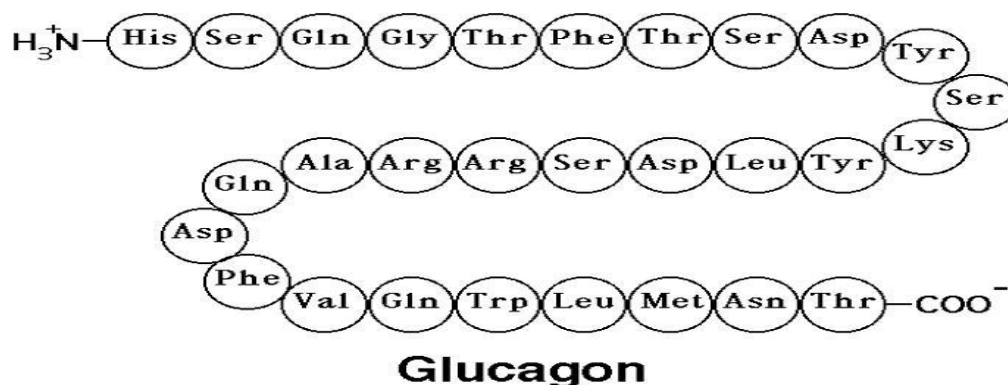


Figure 1.3 : Séquence primaire d'une protéine.

#### a. Les Acides Aminés :

##### a.1 Le carbone chiral :

Un acide aminé est un composé organique contenant un groupement amine et un groupement acide carboxylique. Le type ( $\alpha$ ,  $\beta$ ,  $\delta$ ,...) d'acide aminé est relié à la position de l'amine sur la chaîne carbonée. Les acides aminés qui composent les protéines sont les acides  $\alpha$ -aminés. En effet, la fonction amine est en position  $\alpha$  de la fonction acide. Le carbone où se rattache la fonction amine est appelé carbone  $\alpha$  et sera noté par la suite  $C\alpha$ . Comme ce carbone est relié à quatre groupes différents ( $\text{COOH}$ ,  $\text{NH}_2$ ,  $\text{H}$  et  $\text{R}$ ), il est chiral (sauf pour la glycine où  $\text{R}$  est un hydrogène) [6].

## a.2 Classification suivant la nature des chaînes latérales :

Il existe 20 acides aminés naturels (20 chaînes latérales R différentes) qui composent les protéines. Un code de trois lettres et un code d'une lettre (Table 1.1).

On peut les répertorier en trois groupes selon leur réactivité chimique :

- Les acides aminés polaires.
- Les acides aminés chargés.
- Les acides aminés hydrophobes.

Acide aminé	Abréviation à trois lettres	Abréviation à une lettre
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Acide aspartique	Asp	D
Cystéine	Cys	C
Glutamine	Gln	Q
Acide glutamique	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Méthionine	Met	M
Phénylalanine	Phe	F
Proline	Pro	P
Sérine	Ser	S
Thréonine	Thr	T
Tryptophane	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Table 1.1 Les 20 Acides aminés.

## b. la liaison peptidique :

Une liaison peptidique est une liaison entre le groupe carboxyles (COOH) d'un acide aminé et le groupe amine (NH<sub>2</sub>) de l'acide aminé suivant [8]. On appelle un peptide l'ensemble de deux ou plusieurs acides aminés enchainés les uns aux autres par des liaisons peptidiques. Les acides aminés sont alors appelés des résidus.

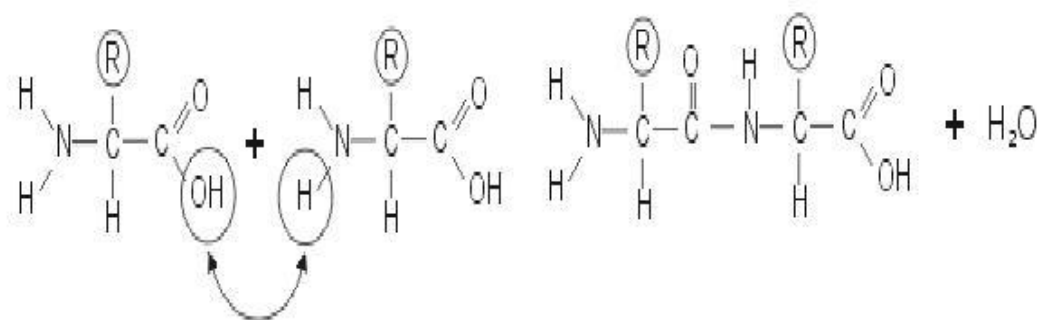


Figure 1.4 : Formulation d'une liaison peptidique.

### 1.2.3.2 Les structures secondaires :

La structure secondaire des protéines résulte de la possibilité de formation du repliement local à partir d'un ensemble de liaisons d'hydrogène entre l'oxygène O et l'hydrogène H des groupements C=O et N-H. Parmi ces repliements il y a deux catégories des structures secondaires fréquentes : les hélices et les feuillets. [8]. Ces structures sont caractérisées par une répétition des valeurs des angles  $\phi$  et  $\psi$ .

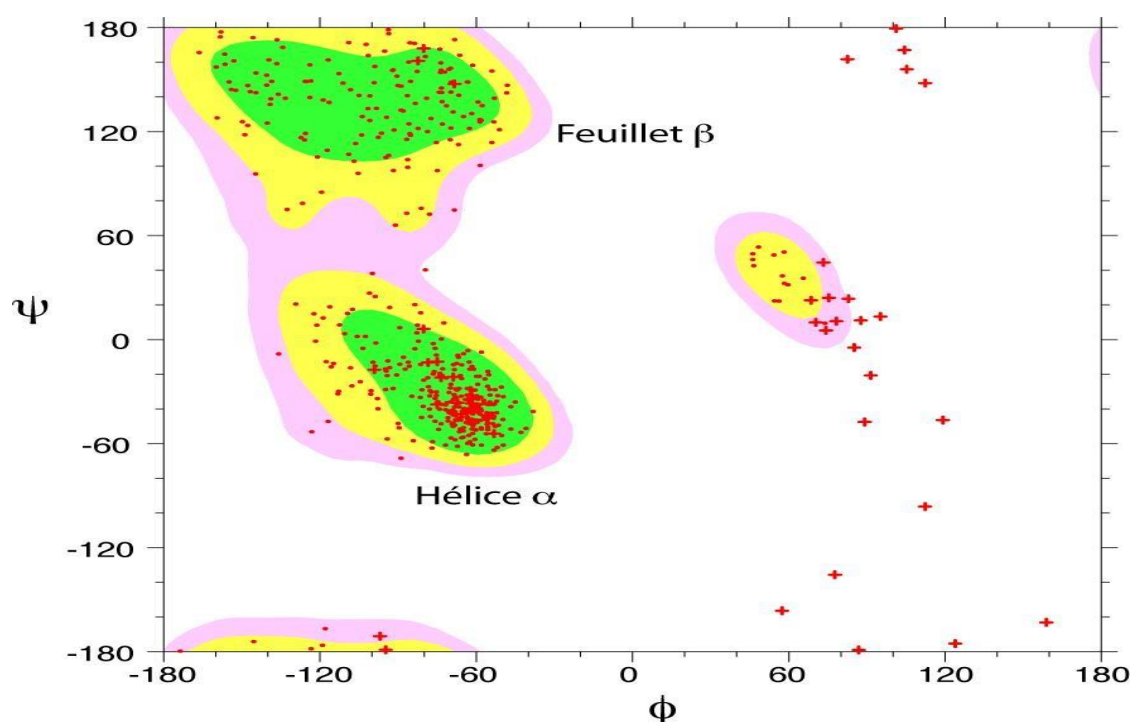


Figure 1.5 : Le diagramme de Ramachandran [2].

#### a. L'hélice $\alpha$ :

L'hélice  $\alpha$  est une structure secondaire caractérisée par des liaisons hydrogène entre le résidu (i) et le résidu (i+4), la chaîne principale tourne par rapport à elle-même d'un

tour tous les 4 résidus. Un tour d'hélice comporte 3,6 résidus, et les radicaux (R) de chaque résidu se projettent vers l'extérieur de l'hélice. Les angles dièdres pour les acides aminés de l'hélice ont des valeurs ( $\phi = -57$ ) et ( $\psi = -47$ ) en moyenne [8]. La longueur des hélices  $\alpha$  peut varier considérablement, de 4 ou 5 résidus à plus d'une quarantaine, avec une moyenne de 10 résidus. La terminologie hélice " $\alpha$ " n'est basée que sur une classification ancienne, antérieure à la Détermination de la structure [6].

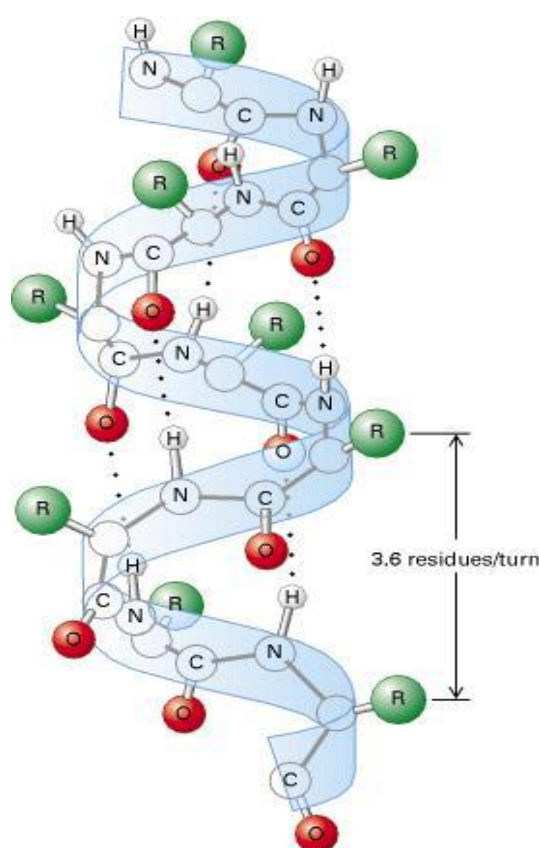


Figure 1.6 : hélice [9].

### b. Le feuillet $\beta$ :

Dans le feuillet  $\beta$ , les liaisons hydrogène intermoléculaires stabilisent l'alignement ordonné des chaînes peptidiques. Les chaînes polypeptidiques (ou brins) voisines sont alors dites parallèles si leurs bouts N-terminaux sont tous du même côté et antiparallèles dans le cas (Figure 1.7) [6].



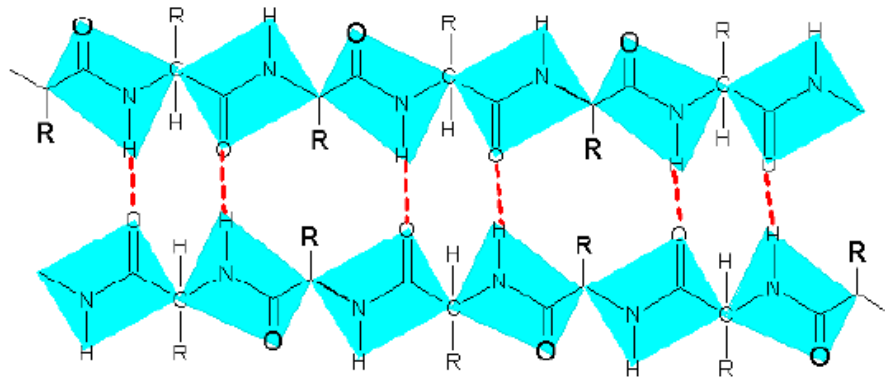


Figure 1.7: Feuille  $\beta$ .

### c. Coudes et boucles :

Les coudes  $\beta$  sont des segments polypeptidiques qui relient deux structures secondaires répétitives (hélices ou feuillets). Ils se trouvent presque toujours à la surface des protéines. On parle souvent d'épingles à cheveux  $\beta$  ( $\beta$  hairpin) car les deux extrémités sont parallèles entre elles (voir figure 1.8) [6].

Les boucles  $\Omega$  peuvent contenir plusieurs coudes  $\beta$  et ont la forme de la lettre grecque majuscule. Elles sont compactes car leurs chaînes latérales ont tendance à remplir l'intérieur de leurs cavités [6].

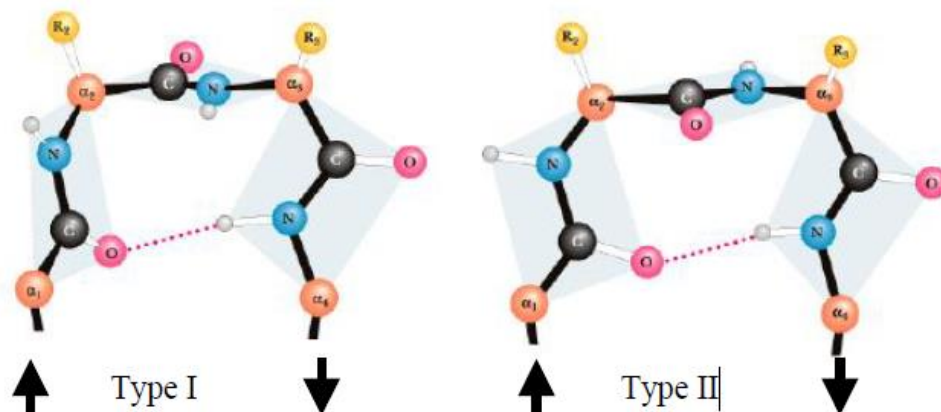


Figure 1.8 : Différents types de coudes  $\beta$ . À gauche : type I ( $\phi_2 = -60^\circ$ ,  $\psi_2 = -30^\circ$ ,  $\phi_3 = -90^\circ$ ,  $\psi_3 = 0^\circ$ ); à droite: type II ( $\phi_2 = -60^\circ$ ,  $\psi_2 = 120^\circ$ ,  $\phi_3 = +90^\circ$ ,  $\psi_3 = 0^\circ$ ) [6].

### 1.2.3.3 La structure tertiaire :

C'est la conformation tridimensionnelle thermodynamiquement stable (due à un ensemble de liaisons non covalentes, comme les liaisons hydrogènes ou les ponts salins, ainsi qu'à des ponts disulfures qui sont des liaisons covalentes) qu'on domptent les différents éléments de la



structure secondaire entre eux pour former la protéine ou une des sous-unités d'une protéine plus complexe. La conformation native d'une protéine dépend à la fois de sa séquence et du milieu dans lequel elle est solubilisée. [10]

Le repliement 3D ("fold" en anglais) représente le meilleur compromis entre l'enfouissement des résidus d'acides aminés hydrophobes (alanine, leucine, isoleucine, proline et valine), puisque la plupart des milieux organiques sont aqueux, et les possibilités de rotation autour des liaisons chimiques [10].

### **1.2.3.4 La structure quaternaire :**

Certaines protéines, complexes, sont constituées de plusieurs sous-unités : les monomères. La structure 4D est l'arrangement spatial de ces différentes unités ; leur rassemblement est un oligomère. Il existe bien sûr des méthodes physiques expérimentales pour déterminer la structure mais elles sont lourdes et coûteuses, et ne peuvent s'appliquer à toutes les protéines (inutilisables pour les protéines non solubles, comme les protéines membranaires, d'où l'importance, là encore, de la prédiction in silico) [10].

## **1.3 Problèmes issus de la bio-informatique :**

Nous présentons ici quelques-uns des principaux domaines de la bio-informatique :

### **1.3.1 Prédiction de structures :**

En plus de la génomique structurale, la prédiction de la structure des protéines vise à développer des moyens permettant d'élaborer efficacement des modèles plausibles décrivant la structure de protéines qui n'ont pu être résolues expérimentalement. Le mode de prédiction de structure de plus efficace, appelé modélisation par homologie, se fonde sur l'existence de structures modèles connues dont la séquence présente des similitudes avec celle de la protéine étudiée. Le but de la génomique structurale est de fournir suffisamment de données sur les structures résolues afin de permettre l'élucidation de celles qui restent à résoudre. Bien qu'il demeure malaisé de modéliser précisément des structures lorsqu'il n'existe que des modèles structuraux éloignés auxquels se référer, on pense que le nœud du problème se trouve au niveau de l'alignement des séquences car des modèles très exacts peuvent être établis dès lors qu'un alignement de séquences très exact est connu. De nombreuses prédictions de structures ont été utiles au domaine émergent du génie protéique, qui a notamment élaboré de nouveaux modes de repliement. Un problème plus complexe à résoudre par le calcul est la prédiction

des interactions intermoléculaires, comme la prédiction de l'ancrage des molécules et des interactions protéine [12]. Le problème de **prédiction de structures** sera détaillé dans le chapitre 2.

### 1.3.2 Alignement de séquences :

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 [13].

### 1.3.3 Phylogénie :

Les arbres phylogénétiques fournissent une méthode simple pour déterminer les relations existantes entre plusieurs espèces. Pour cela le point de vue utilisé est celui de l'évolution. En effet l'objectif est de créer un arbre montrant la proximité de ces espèces. On suppose qu'à l'origine elles ont toutes un ancêtre commun. Celui-ci est représenté par la racine d'un arbre dont les feuilles représentent les espèces observées [5].

Construire un arbre revient à donner un scénario possible pour l'évolution depuis l'ancêtre commun jusqu'aux espèces actuelles. Le nombre d'arbres possibles augmente de façon exponentielle en fonction du nombre de feuilles. Il convient donc d'avoir un critère pertinent pour déterminer le meilleur arbre possible, c'est à dire correspondant au scénario le plus probable. Dans la mesure du possible, pour que cela ait un sens, il faut que les séquences aient un lien de parenté. Autrement dit il est préférable d'avoir des séquences orthologues [5].

### 1.3.4 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 [5].

### 1.4 Les banques de données :

Il est difficile de déterminer la structure des protéines expérimentalement. Les banques des séquences ne cessent de croître à l'instar des banques de structures protéiques : actuellement, plus de dix millions de gènes sont présents dans GenBank alors que seulement vingt milles structures protéiques sont répertoriées dans PDB (Protein Data Bank).

#### 1.4.1 La PDB (Protein Data Bank) :

La PDB est la principale banque internationale de structures tridimensionnelles (Kouranov et al. 2006). Cette banque a été fondée en 1971 par le BNL (Brookhaven National Laboratory) et contenait 7 structures. Depuis 1998, elle est sous la tutelle du RSCB (Research Collaboratory for Structural Bioinformatics). Les structures de protéines constituent l'essentiel des entrées de la PDB avec 100000 entrées en août 2015, mais on y trouve également des structures de molécules d'ARN (511) et d'ADN (1068), de complexes protéines-acides nucléiques (1519). Ces structures sont déterminées expérimentalement par cristallographie aux rayons X, par RMN ou encore microscopie électronique. On peut noter que plus de 90% des structures déposées dans la PDB sont résolues par la technique de cristallographie aux rayons X. Les entrées de la banque comprennent des informations sur les structures primaires et secondaires des molécules considérées, les coordonnées atomiques, souvent des détails des expériences (conditions de cristallisation, empilement cristallin, statistiques d'affinement, etc.) ainsi que des références bibliographiques. Bien que le nombre de structures de macromolécules biologiques connues à l'heure actuelle soit très inférieur à celui des séquences (100000 structures dans la banque PDB contre 222289 protéines dans la seule banque Swiss-Prot en août 2015), celui-ci croît actuellement à une vitesse comparable à celle observée pour les séquences protéiques il y a quelques années (Figure 1.8). Cependant, il faut noter une redondance importante dans la PDB

car plusieurs structures 3D peuvent correspondre à la même séquence selon les conditions d'obtention de la structure ou la finesse de sa résolution [1].

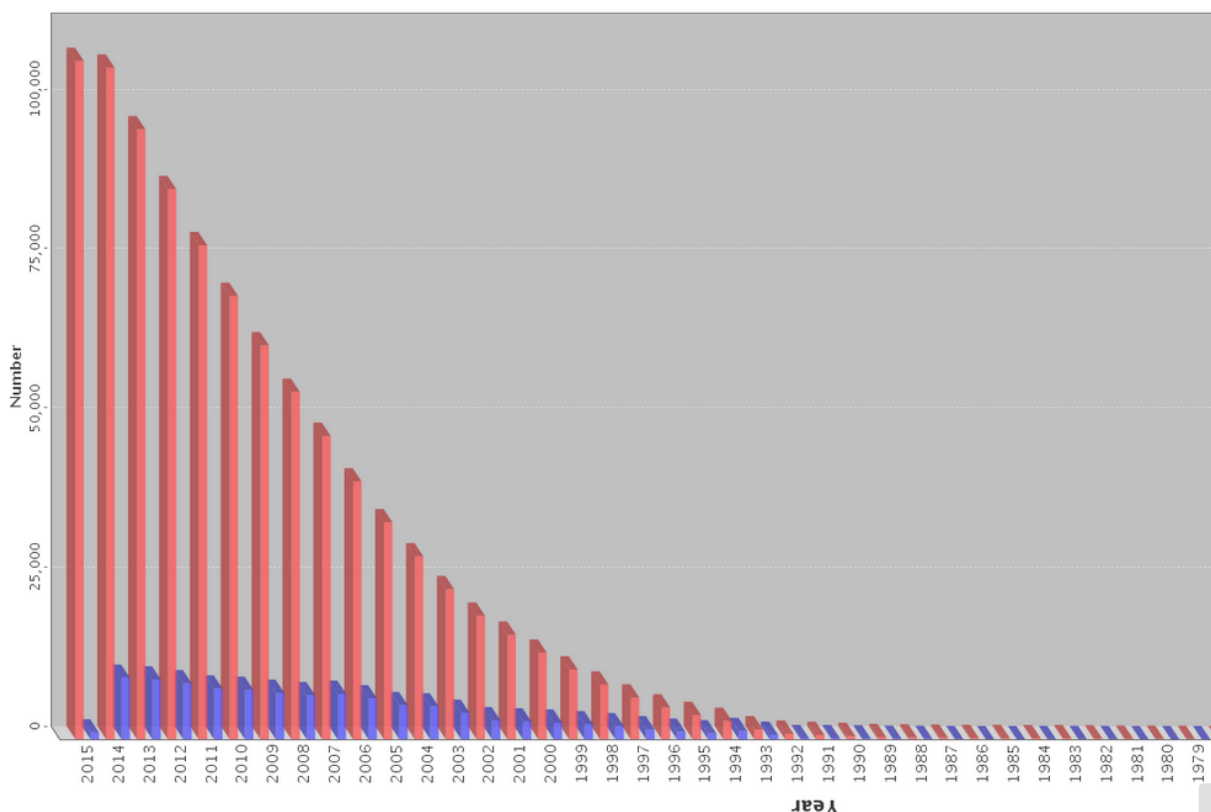


Figure 1.9 : Evolution du nombre d'entrée dans la banque de structures 3D, la PDB.

### 1.4.2 GenBank:

La GenBank est une banque de séquences d'ADN, comprenant toutes les séquences de nucléotides publiquement disponibles et leur traduction en protéines. Cette base de données américaine « Nucleotide », en libre accès, a été créée au Centre national pour l'information biotechnologique (NCBI) dans le cadre de la collaboration internationale sur le séquençage des nucléotides (INSDC selon le sigle anglais). La GenBank et ses collaborateurs reçoivent des séquences produites dans des laboratoires du monde entier à partir de plus de 100 000 organismes différents. La GenBank continue de grossir à un taux exponentiel positif, doublant de taille tous les dix mois. La version 155, datée d'août 2006, contenait plus de 65 milliards de bases de nucléotides dans plus de 61 millions de séquences. La GenBank se construit soit par des dépôts directs en provenance de laboratoires, soit des dépôts en masse des centres de séquençage à grande échelle [16].

Les dépôts directs à la GenBank se font par l'intermédiaire de Bank It, qui est un formulaire Internet, ou par le programme de dépôt autonome, Sequin. À la réception du dépôt d'une séquence, l'équipe de la GenBank attribue un numéro d'ordre à la séquence et réalise les contrôles d'assurance qualité. Les dépôts sont ensuite inscrits dans la base de données publique, dont on peut consulter les entrées par Entrez ou les télécharger par FTP. Les dépôts en masse de données de exprimées, Séquence (STS), Genome Survey Séquence (GSS) et High-Throughput Genome Sequence (HTGS) sont généralement transmis par les centres de séquençage à grande échelle. Les dépôts de groupe directs à la GenBank traitent également des séquences complètes de génomes microbiens [16].

### 1.5 Représentation informatique, format de séquence :

Représenter un alphabet en informatique est une chose aisée, d'autant que les séquences qui sont manipulées peuvent être codées avec des caractères ASCII. Une première représentation intuitive consiste donc à considérer une séquence comme une simple chaîne de caractères.

Chaque caractère appartenant à l'alphabet sur lequel cette séquence est définie. La manipulation des chaînes de caractères est souvent assez simple en informatique, car faisant partie des types de base de tous les langages. Le stockage peut également être réalisé très simplement dans des fichiers au format texte.

#### 1.5.1 Le format FASTA (ou format Pearson) :

Un format de fichier texte utilisé pour stocker des séquences biologiques de nature nucléique ou protéique. Ces séquences sont représentées par une suite de lettres codant pour des acides nucléiques ou des acides aminés selon la nomenclature IUPAC. Chaque séquence peut être précédée par un nom et des commentaires. Ce format est originellement issu de la suite de programmes FASTA mais, de par son utilisation très répandue, est devenu un standard *de facto* en bio-informatique [14].

Un fichier FASTA est composé au minimum de deux lignes. La ligne 1 décrit la séquence en commençant par le signe ">" suivi immédiatement de l'identifiant de la séquence et d'un commentaire séparé de l'identifiant par un espace [15]. Le signe ">" est obligatoire mais identifiant et commentaire sont optionnels, même si pour des questions de bonnes pratiques bio-informatiques il est fortement recommandé d'ajouter au moins un identifiant à

la séquence. Identifiant et commentaire peuvent contenir tout type de caractères excepté les caractères de contrôle autres que ceux codant une fin de ligne [15].

La ligne 2 est constituée des lettres représentant les acides nucléiques ou les acides aminés de la séquence. Cette ligne possède cependant une longueur maximale de 120 résidus: toute séquence de longueur supérieure doit être découpée en plusieurs lignes. Pour des raisons historiques liées aux premiers affichages sur écran DEC-VT, le découpage généralement rencontré est de 80 caractères, correspondant au mode 80 caractères par ligne permis à l'époque (en parallèle du mode 132 qui était plus difficile à lire). Des découpages de 60 ou 70 caractères sont aussi largement répandus mais ce découpage peut en réalité être réalisé avec n'importe quelle longueur de chaîne de caractères inférieure ou égale à 120 caractères [15].

```
>gi|5524211|gb|AAD44166.1| cytochrome b
LCLYTHIGRNIYYGSYLYSETWNTGIMLLLITMATAAFMGYVLPWGQMSFWGATVITNLFSaipYIG
TNLVEWIWGGFSVDKATLNRFFAFHFILPFTMVALAGVHLTFLHETGSNNPLGLTSDSDKIPFHPY
YTIKDFLGLLILLLLLLLLALLSPDMLGDPDNHMPADPLNTPLHIKPEWYFLFAYAILRSVPNKLGG
VLALFLSIVILGLMPFLHTSKHRSMMLRPLSQALFWTLTMDLLTLTWIGSQPVEYPYTIGQMASIL
YFSIILAFLPIAGXIENY
```

Figure 1.10 un exemple FASTA format.

### 1.5.2 Pdb format :

Le format pdb est le format original de la banque. La guide de ce format a été révisée à plusieurs reprises ; la version actuelle (nov. 2012) est la version 3.30. Il est fortement conseillé de lire ce guide avant d'examiner les données brutes des fichiers pdb [17].

Les archives contiennent les coordonnées cartésiennes des atomes, la bibliographie, les informations structurales, les facteurs de la structure cristallographique et les données expérimentales de la RMN. À l'origine, le format pdb a été dicté par l'utilisation et la largeur de cartes perforées pour ordinateur. En conséquence, chaque ligne contient exactement 80 caractères [17].

Un fichier au format pdb est un fichier texte où chaque colonne possède sa signification : chaque paramètre est positionné de façon immuable. Ainsi, les 6 premières colonnes, c'est-à-dire les 6 premiers caractères pour une ligne donnée,

déterminent le champ du fichier. On retrouve par exemple les champs « TITLE\_ » (c'est-à-dire le titre de la macromolécule étudiée), « KEYWDS » (les mots-clés de l'entrée), « EXPDTA » qui donne des informations sur la méthode expérimentale employée, « SEQRES » (la séquence de la protéine étudiée), « ATOM\_\_ » ou « HETATM », champs comprenant toutes les informations liées à un atome particulier. Dernier exemple, dans ces derniers champs, le nom de l'atome est décrit par les colonnes 13 à 16 (soit du treizième au seizième caractère de la ligne) [17].

Les lignes « ATOM\_\_ » concernent les acides aminés ou les acides nucléiques, et les lignes « HETATM » sont dédiées aux autres molécules (solvant, substrat, ion, détergent...). Il y a autant de lignes « ATOM\_\_ » et « HETATM » que d'atomes observés par l'expérimentateur, pour une macromolécule ou un complexe donné [17].

La longue histoire du format pdb a abouti sur des données non uniformes. Ce format laisse également la place à de nombreuses erreurs, qui ne sont pas systématiquement éliminées lors des contrôles accompagnant le dépôt des structures. Il peut s'agir de désaccords entre la séquence et les résidus représentés, ou de problèmes liés à la nomenclature des atomes des acides aminés ou des ligands [17].

### 1.6 Conclusion :

Dans ce chapitre nous avons commencé par donner quelques notions de base sur la biologie moléculaires nécessaire pour un bio informaticien. Ensuite nous avons donné une définition de quelques problèmes issue comme prédiction de structures qui sera détaillé dans le prochain chapitre. Enfin nous avons défini quelques banques populaires de ces données et les techniques de représentation informatique des données biologiques.



---

## **Chapitre 2 :**

---

# **PREDICTION DE STRUCTURE SECONDAIRE :**

---



### 2.1 Introduction :

La prédiction de la structure tertiaire d'une protéine à partir de sa seule séquence en acides aminés est un problème très difficile. En revanche, la simplification permise par les définitions de structures secondaires plus restreintes ci-dessus a permis de rendre la question plus accessible et la prédiction de structure secondaire des protéines a été l'objet de recherches actives depuis de nombreuses années [25].

La prédiction des structures secondaires d'après les séquences constitue un vaste domaine d'étude depuis les années 70. Le nombre impressionnant d'articles publiés sur le sujet interdit de faire une liste exhaustive de toutes les méthodes de prédiction. Seront présentées ici les principales avancées du domaine, ainsi que quelques méthodes choisies [18].

Les méthodes développées dans les années 80 et jusqu'au début des années 90 tiennent compte, dans le processus de prédiction, de l'environnement local des résidus dans la séquence. Pour cela, la prédiction de la conformation d'un résidu est réalisée à partir d'une fenêtre glissante dans la séquence. La taille de cette fenêtre est variable selon les méthodes [18].

A partir de la séquence d'acides aminés, on cherche à déterminer la structure secondaire qui est codifiée en une succession d'états grâce à un alphabet de trois lettres : portions en hélice alpha (H), feuillet beta (E) et en coude (C).

Pour cela, les programmes de prédiction ont recourt à plusieurs méthodes :

- Les méthodes statistiques
- méthodes tenant compte des propriétés physico-chimiques des acides aminés
- méthode des plus proches voisins
- Chaînes de Markov
- méthode d'apprentissage par réseaux de neurones

Dans ce chapitre nous allons décrire les principales méthodes de prédictions.

### 2.2 Les méthodes statistiques:

Les premières datent de 1974. A partir de la connaissance des structures tertiaires d'un échantillon de protéines modèles, on établit une table d'occurrences comptabilisant les proportions observées de chacun des vingt acides aminés dans un état structural donné. La prédiction est établie à partir de table 2.1 [10].

Les méthodes statistiques prédisent les structures secondaires d'une protéine à l'aide de tableaux de valeurs expérimentales calculées à partir de structures cristallines connues.

### - **Chou-Fasman :**

Cette méthode, connue en 1974, se base sur les propriétés physico-chimiques définissant la stabilité de la protéine, telles que l'hydrophobicité. Les auteurs ont donc calculé les valeurs des paramètres de conformation d'un aminoacide de se trouver dans une structure d'hélice  $\alpha$ , de feuillet  $\beta$  ou de coude à partir de la structure cristalline de 29 protéines déterminée par cristallographie rayon X. Le tableau 2.1 regroupent les probabilités d'un aminoacide  $i$  de se trouver dans une structure d'hélice  $\alpha$  ( $P\alpha(i)$ ), de feuillet  $\beta$  ( $P\beta(i)$ ) ou de coude ( $Pt(i)$ ) ainsi que les fréquences  $f_i$  de courbures des quatre aminoacides consécutives participant à la structure de coude [19].

### - **Principe :**

La séquence pour laquelle on veut prédire les structures secondaires est parcourue par une fenêtre glissante de quatre aminoacides [19].

Le score  $Sc_s(i)$  pour la structure  $s$  de la première aminoacide  $i$  de cette fenêtre est calculé comme suit, en tenant compte des trois acides aminés suivants ( $i + 1$ ,  $i + 2$  et  $i + 3$ ) [19] :

$$Sc_s(i) = \sum_{j=i}^{i+3} P_s(j) \quad (2.1)$$

De même, la probabilité de courbure au niveau du résidu  $i$  positionné au début de la fenêtre de quatre aminoacides est :

$$pt(i) = \prod_{j=i}^{i+3} f(j) \quad (2.2)$$

L'ensemble des règles définissant l'algorithme de Chou-Fasman permet ensuite de prédire la structure secondaire de chaque aminoacide de la séquence :

– **Règle 1** : Un ensemble de quatre acides aminés d'affinité  $H\alpha$  ou  $h\alpha$  ( $Sc_\alpha > Sc_\beta$  et  $Sc_{coude}$ ) sur six consécutifs initie une hélice. Le segment est étendu dans les deux sens jusqu'à la rencontre d'acides aminés empêchant la formation d'hélice  $\alpha$ , c'est à dire si  $Sc_\alpha < 1.00$ . Les deux conditions suivantes confirment la structure en hélice  $\alpha$  de ce segment étendu :

	$P_{\alpha}$		$P_{\beta}$		$P_{coude}$		$f_i$		$f_{i+1}$		$f_{i+2}$		$f_{i+3}$
Glu	1.51	Val	1.70	Asn	1.56	Asn	0.161	Pro	0.301	Asn	0.191	Trp	0.167
Met	1.43	Ile	1.69	Gly	1.56	Cys	0.149	Ser	0.139	Gly	0.190	Gly	0.152
Ala	1.42	Tyr	1.47	Pro	1.52	Asp	0.147	Lys	0.115	Asp	0.179	Cys	0.128
Leu	1.21	Phe	1.38	Asp	1.46	His	0.140	Asp	0.110	Ser	0.125	Tyr	0.125
Lys	1.16	Trp	1.37	Ser	1.43	Ser	0.120	Thr	0.108	Cys	0.117	Ser	0.106
Phe	1.13	Leu	1.30	Cys	1.19	Pro	0.102	Arg	0.106	Tyr	0.114	Gln	0.098
Gln	1.11	Cys	1.19	Tyr	1.14	Gly	0.102	Gln	0.098	Arg	0.099	Lys	0.095
Trp	1.08	Thr	1.19	Lys	1.01	Thr	0.086	Gly	0.085	His	0.093	Asn	0.091
Ile	1.08	Gln	1.10	Gln	0.98	Tyr	0.082	Asn	0.083	Glu	0.077	Arg	0.085
Val	1.06	Met	0.05	Thr	0.96	Trp	0.077	Met	0.082	Lys	0.072	Asp	0.081
Asp	1.01	Arg	0.93	Trp	0.96	Gln	0.074	Ala	0.076	Thr	0.065	Thr	0.079
His	1.00	Asn	0.89	Arg	0.95	Arg	0.070	Tyr	0.065	Phe	0.065	Leu	0.070
Arg	0.98	His	0.87	His	0.95	Met	0.068	Glu	0.060	Trp	0.064	Pro	0.068
Thr	0.83	Ala	0.83	Glu	0.74	Val	0.062	Cys	0.053	Gln	0.037	Phe	0.065
Ser	0.77	Ser	0.75	Ala	0.66	Leu	0.061	Val	0.048	Leu	0.036	Glu	0.064
Cys	0.70	Gly	0.75	Met	0.60	Ala	0.060	His	0.047	Ala	0.035	Ala	0.058
Tyr	0.69	Lys	0.74	Phe	0.60	Phe	0.059	Phe	0.041	Pro	0.034	Ile	0.056
Asn	0.67	Pro	0.55	Leu	0.59	Glu	0.056	Ile	0.034	Val	0.028	Met	0.055
Pro	0.57	Asp	0.54	Val	0.50	Lys	0.055	Leu	0.025	Met	0.014	His	0.054
Gly	0.57	Glu	0.37	Ile	0.47	Ile	0.043	Trp	0.013	Ile	0.013	Val	0.053

Table 2.1 – Tableau des paramètres de conformation de la méthode de Chou-Fasman [10].

– La proline ne peut être ni à l'intérieure de l'hélice, ni du côté C-terminal de l'hélice; elle peut cependant apparaître pour l'un des trois résidus du côté N-terminal [10].

– La longueur du segment étendu est au moins de six aminoacides avec  $Sc_{\alpha} > 1.03$  et  $Sc_{\alpha} > Sc_{\beta}$  ( $Sc$  est la moyenne arithmétique des scores sur les six aminoacides) [10].

– **Règle 2** : Un ensemble de trois acides aminés d'affinité H $\beta$  ou h $\beta$  ( $Sc_{\alpha} > Sc_{\beta}$  et  $Sc_{coude}$ ) sur cinq consécutifs initie un feuillet  $\beta$ . Le segment est étendu dans les deux sens jusqu'à la rencontre d'acides aminés empêchant la formation de feuillet  $\beta$ , c'est à dire si  $Sc_{\beta} < 1.00$ . Si  $Sc_{\beta} > 1.05$  et  $Sc_{\beta} > Sc_{\alpha}$  pour le segment étendu, alors le segment représente une structure de feuillet  $\beta$  [10].

– **Règle 3** : Si, pour le résidu  $i$  :

$$\left\{ \begin{array}{l} p_t(i) > 0.75 \times 10^{-4} \\ \hat{Sc}_t > 1.00 \\ \hat{Sc}_t > \hat{Sc}_{\alpha} \\ \hat{Sc}_t > \hat{Sc}_{\beta} \end{array} \right. \quad (2.3)$$

(avec  $Sc$  : moyenne arithmétique des scores sur les quatre aminoacides de la fenêtre initiée par  $i$ ), alors le segment de quatre aminoacides représente une structure de coude [10].

– **Règle 4** : Tout segment recouvrant les régions  $\alpha$  et  $\beta$  est une hélice  $\alpha$  si  $P\alpha > Sc\beta$  ou un feuillet  $\beta$  si  $Sc\beta > Sc\alpha$ . La méthode de Chou-Fasman présente une efficacité de l'ordre de 50 à 60 % [10].

### 2.3 Les méthodes tenant compte des propriétés physico-chimiques des acides aminés :

Ces méthodes ont pour objectif de prédire la structure secondaire de la protéine à partir de différentes propriétés physico-chimiques des acides aminés, telles que la charge, l'hydrophobicité et l'hydrophile. Celles-ci influent directement sur le repliement du résidu à l'intérieur de la séquence. Parmi les méthodes de prédiction basées sur les propriétés physico-chimiques : TMHMM, PHDhtm, DAS et TopPred2, PHDacc, ASC [20].

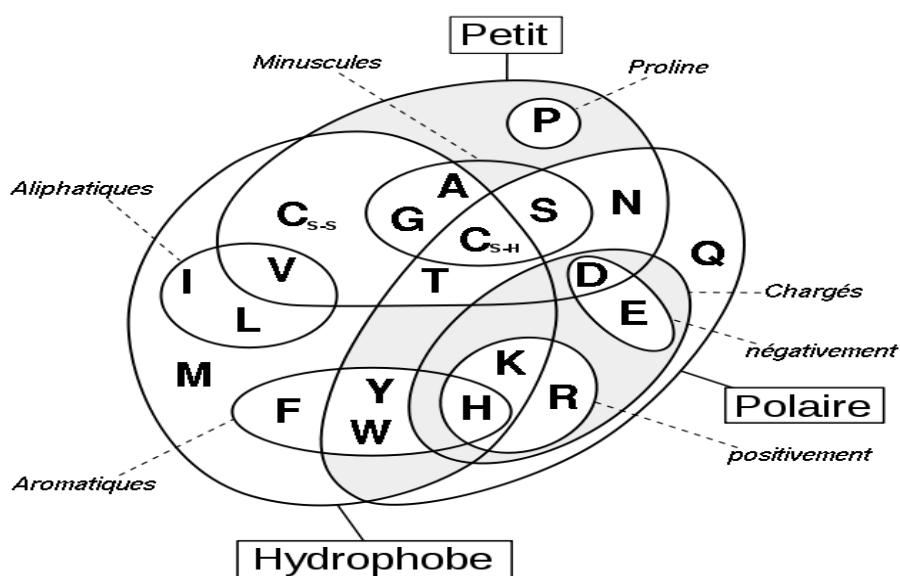


Figure 2.1 : Classification des acides aminés selon les propriétés Physicochimiques.

### 2.4 La méthode du plus proche voisin :

Un nouveau type de prédiction de structure secondaire basé sur des méthodes du plus proche voisin ont vu le jour suite à la découverte d'un grand nombre de structure tertiaire durant les années 1980 [10].

### Principe :

Les séquences protéiques de ces structures tertiaires sont identifiées à la séquence dont on veut prédire la structure secondaire. Procédé général :

1. Une liste de fragments de taille  $n$  (en général,  $n=16$ ) est constituée depuis 100 à 400 séquences de structure connue (appelées également séquences d'entraînement).
2. Une fenêtre de la même taille est extraite de la séquence en entrée pour être comparée à chacun des fragments de la liste. Les 50 fragments les plus similaires sont identifiés.
3. Les fréquences de structure secondaire de l'acide aminé situé au milieu des 50 fragments retenus ( $f_{\alpha}$ ,  $f_{\beta}$  et  $f_{\text{coudé}}$ ) sont utilisés afin de prédire la structure secondaire de l'acide aminé situé au milieu de la fenêtre de la séquence en entrée.
4. La fenêtre courante glisse d'une position pour prédire la conformation d'un nouvel acide aminé; les étapes 2 et 3 sont répétées et le procédé est réitéré jusqu'à ce que tous les résidus-milieu de la séquence aient leur structure secondaire prédite.

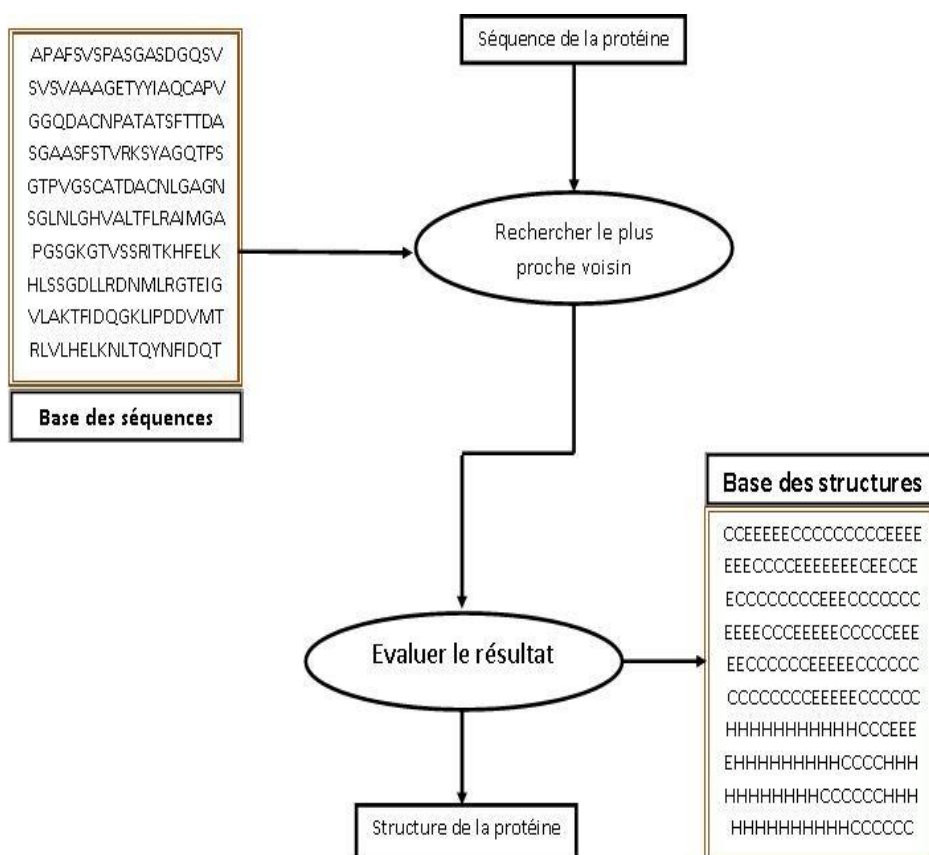


Figure 2.2– Prédiction de la structure protéique par un l'algorithme des plus proches voisins.

## 2.5 Les chaînes de Markov cachées :

La prédiction des structures secondaires des protéines par les HMM (hidden Markov model) est basée sur l'homologie des protéines, chaque type de structure secondaire modélisé par une chaîne de Markov caché. La modélisation HMM d'une séquence est caractérisée par un double processus:

- un processus caché : modélisant les différents états d'un résidu dans une structure secondaire (hélice, feuillet, coud)
- le processus observé : modélisant la séquence principale de la protéine

La difficulté des chaînes de Markov cachées réside dans l'identification des paramètres de ces modèles. Une fois les paramètres déterminés, un score est associé à chaque chaîne de Markov cachée pour une séquence donnée. Le meilleur score d'une portion de séquence détermine sa structure secondaire [21].

### 2.5.1 Principe :

La prédiction des structures secondaires des protéines basées sur les chaînes de Markov caché suit les étapes suivantes :

1. Alignement multiple des fragments similaires de séquences protéiques dont la structure est connue.
2. Génération de modèles de familles structurales (HMM-profil) sous forme de chaînes cachées de Markov.
3. Prédiction de la structure secondaire de séquences à partir des modèles.

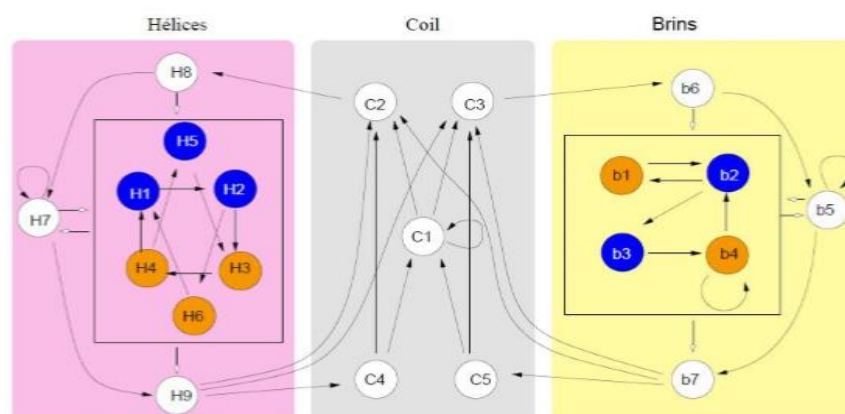


Figure 2.3 – Modèle contraint complet à 21 états cachés [20].

### 2.5.2 Prédiction des structures secondaires par les HMMs :

Déférentes approches ont été proposées pour prédire la structure secondaire avec les HMMs [23]:

- a) une modélisation dite automatique par Asai et al en 1993.
- b) une modélisation experte des protéines par Stultz et White en 1993.
- c) plus récemment, une approche basée sur une collection de fragments : HMMSTR par Bystroff et al en 2000.
- d) la prise en compte d'une fenêtre glissante de structure secondaire par Crookes et Brenner, ainsi que Zheng, en 2004.

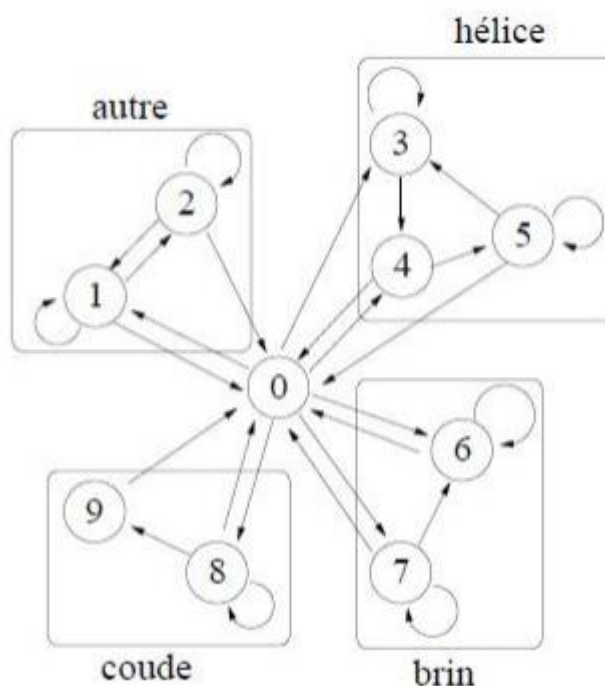


Figure 2.4 – Chaîne de Markov caché pour la méthode Zheng [22].

### 2.6 Méthode d'apprentissage par réseaux de neurones :

La méthode de prédiction des structures secondaires des protéines est une méthode de classification supervisée. Elle modélise les structures secondaires existantes par entraînement sur la séquence des acides aminés. Un modèle en couche a été proposé pour prédire la structure secondaire d'une séquence donnée [24]. La méthode d'apprentissage par réseaux de neurones sera détaillée dans les chapitres 3 et 4.

## 2.7 Programmes de prédiction de structure secondaire et état de l'art :

### 2.7.1 Méthodologie :

Il n'existe pas d'outils permettant de prédire les structures secondaires d'une séquence de manière exacte. Par conséquent, une stratégie à adopter pour augmenter les chances de prévision est la suivante :

1. Obtenir des alignements de la séquence avec autant d'homologues que possibles
2. Utiliser un maximum de méthodes de prédiction de structures secondaires pour améliorer la prédiction
3. Vérifier les motifs conservés des résidus

### 2.7.2 État de l'art :

La plupart des algorithmes actuels qui sont utilisés dans la prédiction secondaire de la structure des protéines dépendent fortement des structures en trois dimensions connues. En utilisant ces algorithmes certains paramètres sont alors imposés à des séquences inconnues. Ces méthodes reposent sur les données disponibles pour leurs prédictions [32]. Algorithmes antérieurs pour la prédiction de structures de protéines secondaires ont signalé un succès élevé, mais ces études étaient basées sur de petites quantités de données qui ont été principalement tirées au cours des sessions d'entraînement. Par exemple, Lim [33] a obtenu 70% de succès de prédiction de 25 protéines de Q3 (détaillé dans le chapitre 4), Garnier [34] a réalisé 63% de réussite pour les 26 protéines, tandis que Qian et Sejnowski [35] ont rapporté 64,3% de succès de prédiction en 19 protéines, Rost et Sander [36] ont testé une méthode dans la prédiction des protéines dans lesquelles ils n'utilisent pas les mêmes protéines pour l'essai de leur algorithme. Ils ont obtenu le succès de prédiction mieux que 70%.

PSIPRED	Réseau de neurones à deux stages	79.34%
PHD Expert	Réseau de neurones	78.01 %
SSPRO	BRNNs	76.30 %
SAM	HMM	80.14 %

Table 2.2 – Les meilleurs serveurs de prédiction des structures secondaires des Protéines.



### 2.8 Conclusion :

Nous avons présenté dans ce chapitre les méthodes de prédiction et **État de l'art** sur les méthodes de prédiction des structures secondaires des protéines.

Enfin nous avons présenté Les meilleurs serveurs de prédiction des structures secondaires des Protéines.



---

## Chapitre 3

# Notions de base des réseaux de Neurones artificiels

### 3.1 Introduction :

Le terme « réseaux de neurones artificiels » regroupe un certain nombre de modèles dans l'intention d'imiter certaines fonctions du cerveau humain reproduisant quelques unes de ses structures de base [26].

Par ailleurs, les réseaux de neurones sont adaptés comme outil d'aide aux opérations de reconnaissance et de classification, entre autre, celles liées à la résolution des problèmes de diagnostic et prédiction [26].

En informatique, on appelle réseau de neurones un ensemble d'entités (les neurones) interconnectées.

dans ce chapitre nous allons présenter les réseaux de neurones qui constituent actuellement un des outils les plus efficaces pour le traitement des problèmes de classification et nous donnons une description de l'architecture générale d'un réseau de neurones et de son mode de fonctionnement, et les types de perceptions. Nous terminons cette partie par perception multicouche et ces étapes.

### 3.2 Le neurone:

Comme les réseaux de neurones mis au point par les informaticiens sont largement inspirés de ce que la biologie nous apprend sur ceux que l'on trouve chez les êtres vivants, il convient d'abord de décrire brièvement le modèle biologique.

#### 3.2.1 Le modèle biologique:

Chez les êtres vivants, les neurones sont les cellules nerveuses. Un neurone est doté de ramifications que l'on nomme les dendrites par lesquelles transite l'information (sous la forme de courants électriques) venue de l'extérieur vers le corps cellulaire. Le neurone traite cette information et renvoie le résultat au travers de son axone. Ce signal émis par le neurone peut ensuite être transmis, au travers d'une synapse, à un autre neurone, ou encore à un muscle ou à une glande (Figure 3.1) [27].

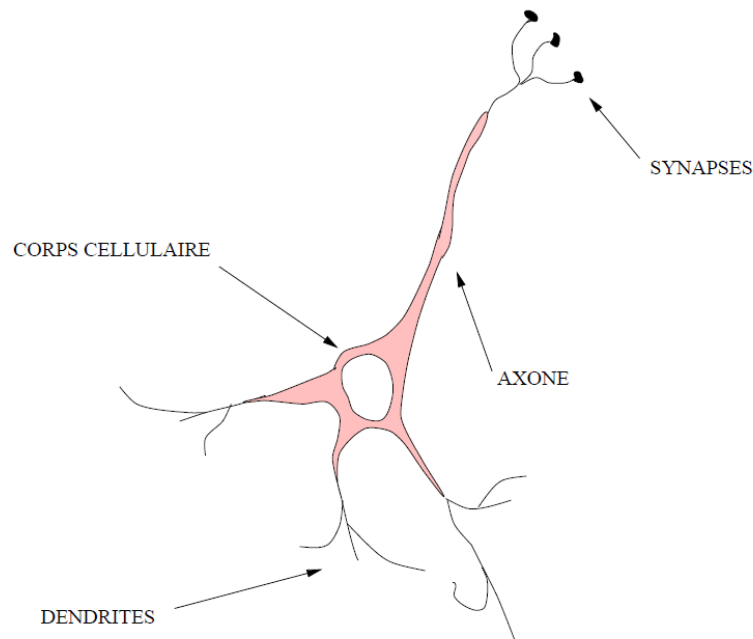


Figure 3.1 : Un neurone biologique et ses principaux composants.

### 3.2.2 Vers une simulation du neurone biologique :

Voici quelques résultats d'expériences sur les cellules nerveuses. Nous nous intéressons ici au comportement des neurones biologiques, comportement dont sont inspirées les caractéristiques des neurones simulés ou formels. Les quelques observations qui suivent vont nous aider à faire le parallèle entre les neurones biologiques et les neurones simulés [27].

- Lorsqu'on stimule un neurone (par un courant électrique par exemple), sa membrane cellulaire se dépolarise. Lorsque la stimulation est suffisamment intense, la dépolarisation est telle qu'une inversion de polarité brutale apparaît et se propage le long de l'axone de manière unidirectionnelle, c'est le potentiel d'action.
- Les synapses peuvent être inhibitrices : elles induisent alors une hyperpolarisation qui a tendance à empêcher la formation d'un potentiel d'action sur le neurone en aval. On comprend aisément leur rôle dans le contrôle de muscles antagonistes où, par exemple, les muscles extenseurs d'un membre ne doivent pas travailler en même temps que les muscles fléchisseurs.
- Les synapses peuvent être excitatrices : elles induisent une dépolarisation qui tend à générer un potentiel d'action sur le neurone en aval. Voyons maintenant comment ces caractéristiques comportementales sont implémentées dans les neurones formels.

### 3.2.3 Le modèle formel :

Les neurones formels (Figure 3.2) sont dotés de caractéristiques inspirées de celles des Neurones biologiques que nous avons passées en revue dans la section précédente [27]:

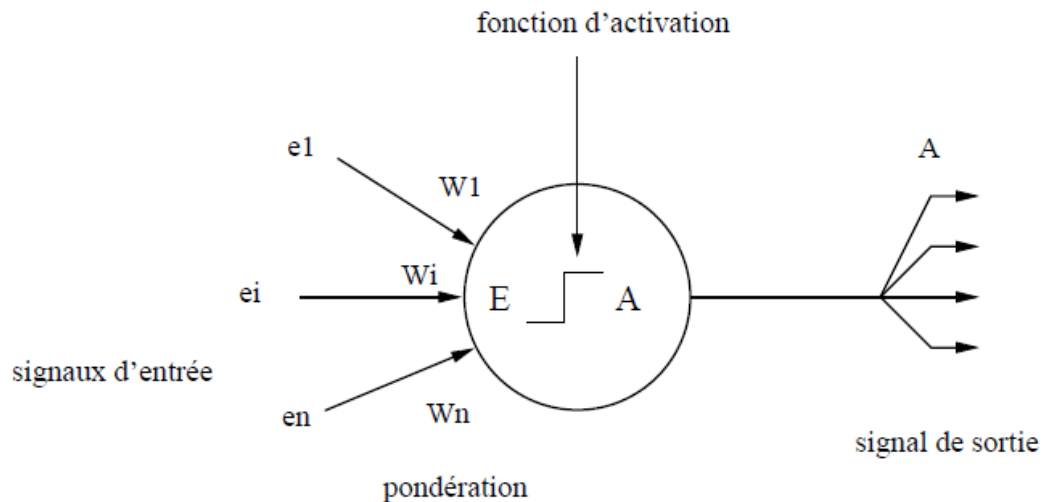


Figure 3.2 : schéma d'un neurone formel.

- **Le potentiel d'action des cellules nerveuses :** il s'agit ici (pour le neurone formel) d'une valeur numérique, qui peut être transmise à des neurones en aval. Un neurone formel, ne peut transmettre qu'une valeur unique qui correspond à son état d'activation.
- **Les dendrites :** des neurones biologiques leur permettent de recevoir différents signaux de l'extérieur. De la même manière, un neurone formel peut recevoir des signaux  $e_i$  de plusieurs neurones. Ces signaux sont combinés en un signal d'entrée unique  $E$  :

$$E = \sum w_i \cdot e_i \quad (3.1).$$

Où les  $w_i$  sont les poids affectés aux signaux extérieurs.

- **Les synapses :** les nombres  $w_i$  pondèrent les signaux émis par les différents neurones situés en amont. On retrouve ici l'analogie des synapses qui, rappelons-le, peuvent être inhibitrices ( $w_i < 0$ ), ou excitatrices ( $w_i > 0$ ).
- **Le seuil d'activation :** une fonction d'activation  $A$  gère l'état du neurone formel. Généralement, si  $A \approx 1$  le neurone est excité et il est au repos si  $A \approx -1$  ou  $A \approx 0$  selon les cas. L'allure de cette fonction est généralement telle qu'il existe un seuil d'activation du neurone (Figure 3.3) : le neurone n'est excité que s'il reçoit un signal d'entrée  $E$  supérieur à ce seuil  $s$ .

Il existe deux motivations distinctes pour l'étude des réseaux de neurones artificiels : celle des neurobiologistes qui désirent mettre à l'épreuve leurs modèles, et celle des informaticiens qui voient là une méthode supplémentaire pour résoudre les problèmes qui leur sont posés. C'est bien entendu la seconde motivation qui nous anime, ainsi n'insisterons nous pas davantage sur les neurones biologiques [28].

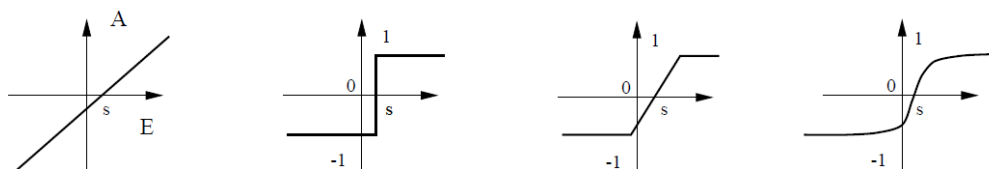


Figure 3.3 : quelques fonctions d'activation.

### 3.3 Les réseaux de neurones artificiels:

Les recherches actuelles faites sur les réseaux de neurones artificiels n'ont pas permis de donner une définition exacte ou universelle à ce concept. En effet, la définition admise, utilise la notion de réseaux d'automates ou de système connexionniste, Un réseau de neurones artificiels est composé d'automates connectés en réseau et fonctionnant en parallèle et dans lequel les connexions contiennent la connaissance d'un domaine particulier [29].

#### 3.3.1 Différentes architectures de réseaux de neurones :

Selon la topologie de connexion des neurones, on peut classifier plusieurs modèles de réseaux de neurones.

##### 3.3.1.1 Réseau multicouche (feed-forward) :

Dans les réseaux multicouche (feed-forward) les neurones sont arrangés par couche, il n'y a pas de connexion entre neurone d'une même couche et la connexion ne se fait qu'avec les neurones des couches en aval. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement, nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et de définir les concepts de neurone d'entrée et de neurone de sortie. Par extension on appelle couche d'entrée l'ensemble des neurones d'entrées et la couche de sortie l'ensemble des neurones de sorties. Les couches intermédiaires n'ayant aucun contact avec l'extérieur sont appelées couches cachées, tel que représenté sur la figure 3.4 [30].

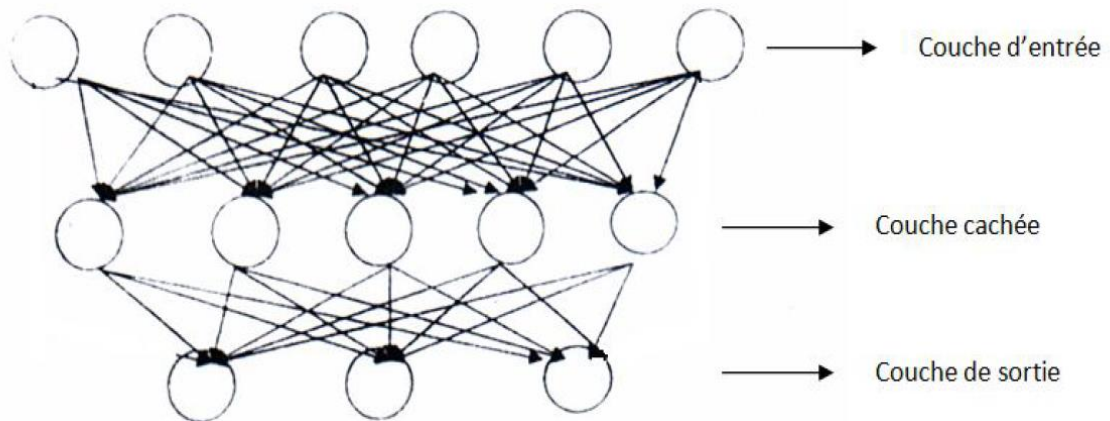


Figure 3.4 : Réseau multicouche.

### 3.3.1.2 Réseau à connexion locale :

Il s'agit d'une structure multicouche, mais qui a l'image de la rétine, conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurone de couche avale. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique. Tel que le montre la figure 3.5.

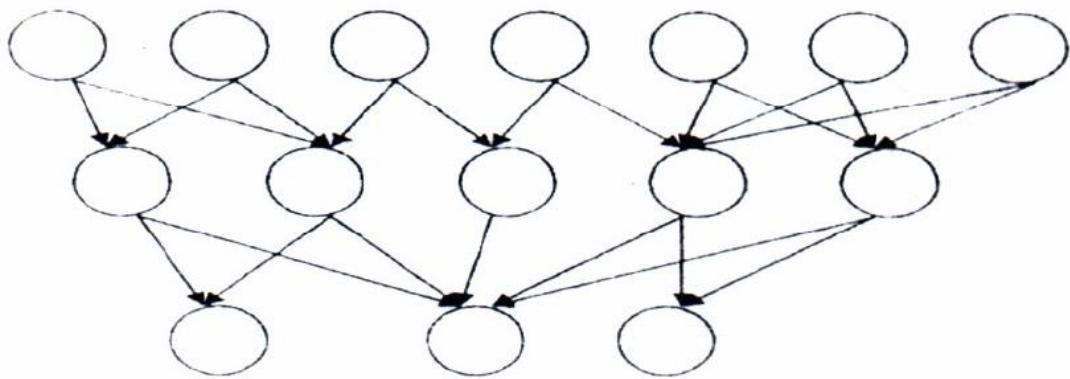


Figure 3.5 Réseau à connexion locale.

### 3.3.1.3 Réseau à connexion complète:

C'est la structure d'interconnexion la plus générale. Chaque neurone est connecté à tous les neurones du réseau et à lui-même. Tel que le montre la figure 3.6.

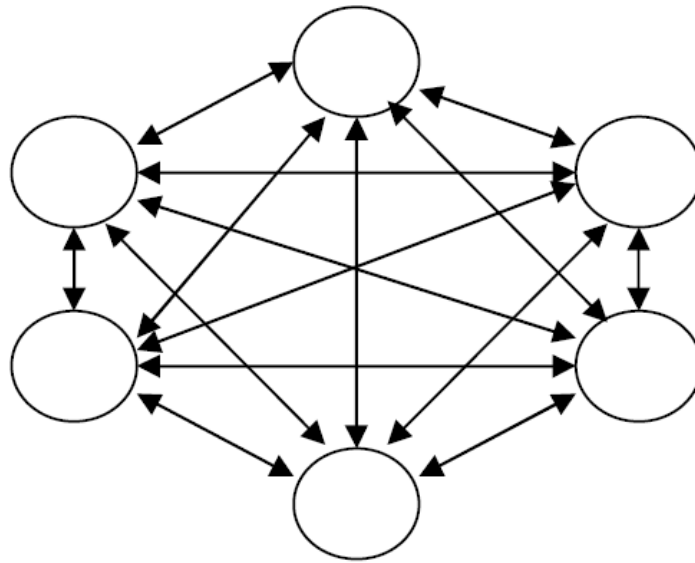


Figure 3.6 Réseau à connexion complète.

### 3.3.2 L'information dans les réseaux de neurones:

Nous venons de voir le fonctionnement individuel des neurones formels, ainsi que la manière dont ils peuvent être connectés en réseau. Nous pouvons maintenant insister sur le fait que toute l'information contenue dans un réseau de neurones réside dans [27] :

- la configuration du réseau (le nombre de neurones, le nombre de liens entre les neurones, le nombre de couches différentes etc...) ;
- la force (définie par les nombres  $w_i$ ) des connexions qui lient les neurones entre eux, en leur permettant de transmettre des signaux avec une plus ou moins grande importance, et avec une action inhibitrice ou excitatrice ;
- la loi qui détermine les réactions des neurones en fonction des informations qu'ils reçoivent (la fonction d'activation).

Une fois l'architecture du réseau et la fonction d'activation des neurones déterminés, il reste à fixer les valeurs des pondérations des liens qui les relient. C'est le but de ce que l'on appelle l'apprentissage.

### 3.3.3 L'apprentissage:

L'apprentissage d'un réseau de neurones formels consiste à déterminer les poids  $w_i$  optimaux (de la force optimale des connexions) suivant le problème à résoudre. Rappelons que



l'architecture du réseau est déjà déterminée à ce stade. On distingue deux types de réseaux de neurones en fonction du type d'apprentissage auquel ils sont soumis :

### 3.3.3.1 Apprentissage supervisé:

Dans ce type d'apprentissage, l'algorithme détermine les poids synaptiques à partir d'exemples étiquetés de formes auquel un professeur (teacher) a associé des réponses ou des cibles également étiquetées. Il existe plusieurs algorithmes, parmi lesquels on distingue l'algorithme de rétro-propagation qui est destiné aux réseaux multicouches [27].

### 3.3.3.2 Apprentissage non supervisé:

Ce mode d'apprentissage est moins intuitif, il correspond au cas où l'on ne dispose pas des bases d'apprentissage, par exemple lorsqu'on ne sait pas à priori déterminer ponctuellement si une sortie est ou non valable. L'apprentissage repose alors sur un « critère interne » de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations [27].

## 3.4 Les réseaux de neurones à apprentissage supervise:

L'apprentissage supervise d'un réseau de neurones consiste à en modifier les poids tant que la réponse correspondant à chaque entrée n'est pas assez proche de la réponse souhaitée.

### 3.4.1 Le cas d'un neurone seul :

La base d'apprentissage comprend  $e$  entrées de dimension  $d$ , et les  $n$  réponses souhaitées  $R = \pm 1$  correspondantes. Le neurone (Figure 3.2) dispose de  $d$  entrées pondérées, chacune pondérée par un nombre  $w$ , et d'une fonction d'activation  $F$  qui détermine sa valeur de sortie. Voici l'algorithme de l'apprentissage [31] :

1. Initialiser les poids  $w_i$  avec des valeurs aléatoires.

$$A = F \left( \sum_{i=1}^d w_i \cdot e_i \right) \quad (3.2).$$

2. Présenter une entrée  $e$  de la base d'apprentissage.
3. Calculer la valeur d'activation du neurone.
4. Calculer l'erreur sur la sortie :  $\Delta = R - A$
5. Modifier les poids selon la relation :

$$w_i(t+1) = w_i + k \cdot e_i \cdot \Delta \quad (3.3).$$

où  $k$  est le pas de l'apprentissage ( $k > 0$ ) poids ne sont pas modifiés si l'activation  $A$  la réponse désirée  $R$  ( $\Delta = 0$ )

6. Retourner à l'étape 2 jusqu'à ce que  $\Delta = 0$  de la base d'apprentissage.

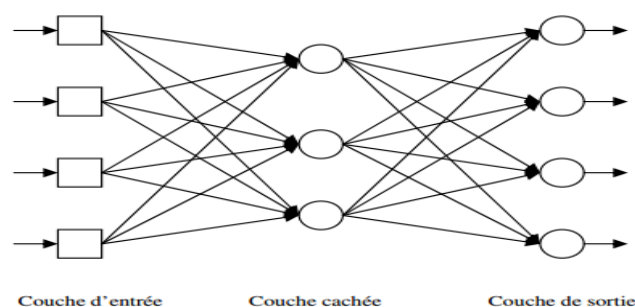
### 3.4.2 Le Perceptron Multicouches :

Le Perceptron Multicouches (PMC), en anglais Multi Layer Perceptron (MLP). Ce type de réseau se situe dans la famille générale des réseaux à propagation vers l'avant, c'est-à-dire qu'en mode normal d'utilisation, l'information se propage dans un sens unique, des entrées vers les sorties sans aucune rétroaction. Son apprentissage est de type supervisé, par correction des erreurs [32]. Dans ce cas uniquement, le signal d'erreur est rétro propagé vers les entrées pour mettre à jour les poids des neurones. Le Perceptron Multicouches est un des réseaux de neurones les plus utilisés pour des problèmes d'approximation, de classification et de prédiction.

#### 3.4.2.1 Structure du Perceptron Multicouches :

Les Perceptrons Multicouches (PMC) sont les réseaux de neurones les plus courants et les plus simples. Ils sont très largement utilisés en classification et en reconnaissance de formes, notamment pour leurs bonnes performances et leur simplicité.

Le PMC est une extension du Perceptron simple, avec une ou plusieurs couches cachées entre l'entrée et la sortie, donc un PMC possède trois types de couches : une couche d'entrée, une ou plusieurs couches cachées et une couche de sortie. Chaque neurone d'une couche est connecté à tous les neurones de la couche qui le précède, ce qui donne un réseau complètement connecté. Le schéma donné dans la figure 3.7 représente un PMC à trois couches. La couche d'entrée comporte quatre neurones. La couche cachée contient trois neurones et enfin la couche de sortie possède quatre neurones [32].



**Figure 3.7 :** Architecture d'un Perceptron Multicouches à une seule couche cachée.

#### 3.4.2.2 Apprentissage d'un PMC

L'apprentissage d'un réseau PMC est de type supervisé c'est-à-dire que l'on présente au réseau, en même temps, une forme et son modèle, ce qui consiste à appliquer des couples

(entrées, sorties désirées) à l'entrée du réseau. L'algorithme le plus utilisé est celui de rétro propagation des erreurs

Algorithme de rétro-propagation du gradient L'algorithme de rétro-propagation (Backpropagation / BP) est le plus utilisé dans l'apprentissage des réseaux MLP [30] :

Entrée : un exemple, sous la forme (vecteur  $x$ , vecteur  $y$ );

Epsilon le taux d'apprentissage

Un Perceptron Multicouches avec  $q-1$  couches cachées  $C_1, \dots, C_{q-1}$ ,

Une couche de sortie  $C_q$ .

$S_i$  : la sortie du neurone  $i$  de la couche de sortie.

$Y_i$  : la sortie attendue pour ce même neurone.

Répéter

Prendre un exemple (vecteur  $x$ , vecteur  $y$ ) et calculer  $g(\text{vecteur } x)$

Pour toute cellule de sortie  $i$   $d_i \leftarrow (1 - S_i)(Y_i - S_i)$  fin Pour

Pour chaque couche de  $q-1$  à 1

Pour chaque cellule  $i$  de la couche courante

$d_i = O_i(1 - O_i) * \text{Somme [pour } k \text{ appartenant aux indices}$

des neurones prenant en entrée la sortie du neurone  $i]$  de  $d_k * w_{ki}$

fin Pour

fin Pour

Pour tout poids  $w_{ij} \leftarrow w_{ij} + \text{epsilon} * d_i * x_{ij}$  fin Pour

fin Répéter

En plus, il est possible d'arrêter l'apprentissage en fixant une limite au nombre d'itérations. Noter que l'ordre de présentation des exemples doit être aléatoire. Généralement le pas d'apprentissage et le moment  $U_m$  doivent être adaptés quand le nombre d'itération augmente [33].

### 3.5 L'apprentissage non supervisé des réseaux de neurones:

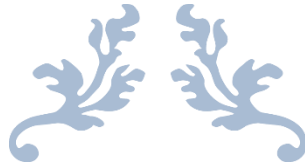
L'apprentissage non supervisé des réseaux de neurones consiste, comme dans le cas des apprentissages supervisés, à modifier les poids des connexions des neurones. Dans ce cas, les exemples de la base d'apprentissage sont des données seules : il n'est pas possible de modifier les poids du réseau en fonction d'une erreur sur les réponses souhaitées, puisqu'aucune réponse n'est connue a priori, car les poids du réseau sont modifiés en fonction de leurs composantes. L'information utile se trouve donc uniquement dans les données, en particulier dans les redondances qui peuvent exister dans l'ensemble des données d'apprentissage [34].

On peut classer les réseaux à apprentissage non supervisé en deux catégories Réseaux du type "winner takes all", Réseaux de type "winner takes most" [34].

### 3.6 Conclusion :

Dans ce chapitre, nous avons commencé par Le neurone et les réseaux de neurones artificiels on y présenté les différentes architectures des RNA (Réseaux de Neurones Artificiels), et leurs méthodes d'apprentissage.

Enfin nous avons présenté les différents algorithmes pour L'apprentissage des réseaux de neurones.



---

## Chapitre 4 :

# ADAPTATION ET IMPLEMENTATION

### 4.1 Introduction :

Nous allons voir dans ce chapitre les outils d'implémentation que nous avons exploité pour l'algorithme et voir comment adapter le réseau de neurone pour la prédiction des structures secondaires de protéine ainsi que présenter les résultats obtenus en les accompagnants avec quelques contraintes.

### 4.2 Présentation des outils d'implémentation :

Pour implémenter cet algorithme nous avons exploité le langage Java en utilisant le NetBeans comme un éditeur.

#### 4.2.1 Langage Java :

Java est un langage de programmation et une plate-forme informatique créée par Sun Microsystems en 1995. Il s'agit de la technologie sous-jacente qui permet l'exécution de programmes modernes et performants, notamment dans la construction des utilitaires, des jeux et des applications professionnelles. Java est utilisée sur plus de 850 millions d'ordinateurs de bureau et plus d'un milliard de périphériques dans le monde, dont des périphériques mobiles et des systèmes de diffusion télévisuelle [35].

#### 4.2.2 Netbeans :

NetBeans est un environnement de développement intégré (EDI), placé en *open source* par Sun en juin 2000 sous licence CDDL (Common Development and Distribution License) et GPLv2. En plus de Java, NetBeans permet également de supporter différents autres langages, comme C, C++, JavaScript, XML, Groovy, PHP et HTML de façon native ainsi que bien d'autres (comme Python ou Ruby) par l'ajout de greffons. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web) [36].



Figure 4.1 : Netbeans.

### 4.3 Réseaux de neurones pour la prédiction des structures secondaires :

La méthode de prédiction des structures secondaires des protéines est une méthode de classification supervisée. Elle modélise les structures secondaires existantes par entraînement sur la séquence des acides aminés. Un modèle en couche à été proposé pour prédire la structure secondaire d'une séquence donnée [24].

Un réseau de neurones artificiel est un modèle qui permet d'inspirer certaines fonctions du cerveau (la mémorisation, l'apprentissage par l'exemple, parallélisme), en fonction des variables d'entrées les réseaux de neurones font un apprentissage pour aboutir aux sorties (résultats).l'utilisation des méthodes neuronales pose certaines difficultés, la principale étant dans l'inférence (l'optimisation de la phase d'apprentissage) [24].

L'application des réseaux de neurones dans la bio-informatique a montré une grande efficacité. Une analogie est faite pour prédire les structures secondaires ; la donnée ici est une fenêtre de n acide aminés de la séquence, le signal (fenêtre) est analysé à travers le réseau est produit en résultat la structure secondaire de l'acide amines de la fenêtre [22].

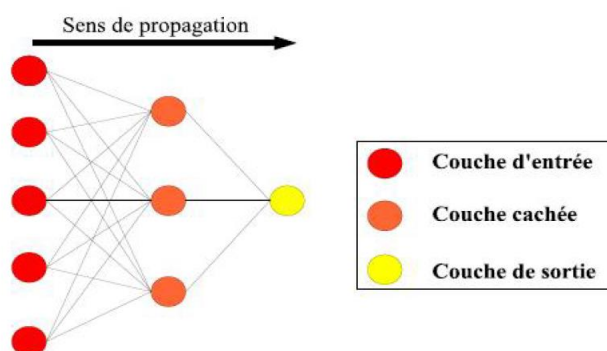


Figure 4.2 : Modèle en couche de réseaux de neurones [24].

Une fois l'ensemble des modèles de réseaux de neurones entraînés, la structure secondaire d'une séquence sera déterminée comme suit [24]:

- Choix d'une fenêtre de n acides aminés de la séquence en entrée
- Prédiction de la structure secondaire de la fenêtre en utilisant un réseau de neurone
- Glissement de la fenêtre sur la séquence puis prédiction de la structure de l'acide aminé de cette nouvelle fenêtre (réitération des étapes 2) jusqu'à ce que la totalité de la séquence soit traitée.

#### 4.4 Description la d'algorithme :

Le concept d'algorithme pour prédire la structure secondaire de la protéine est représenté sur la Figure 4.3.

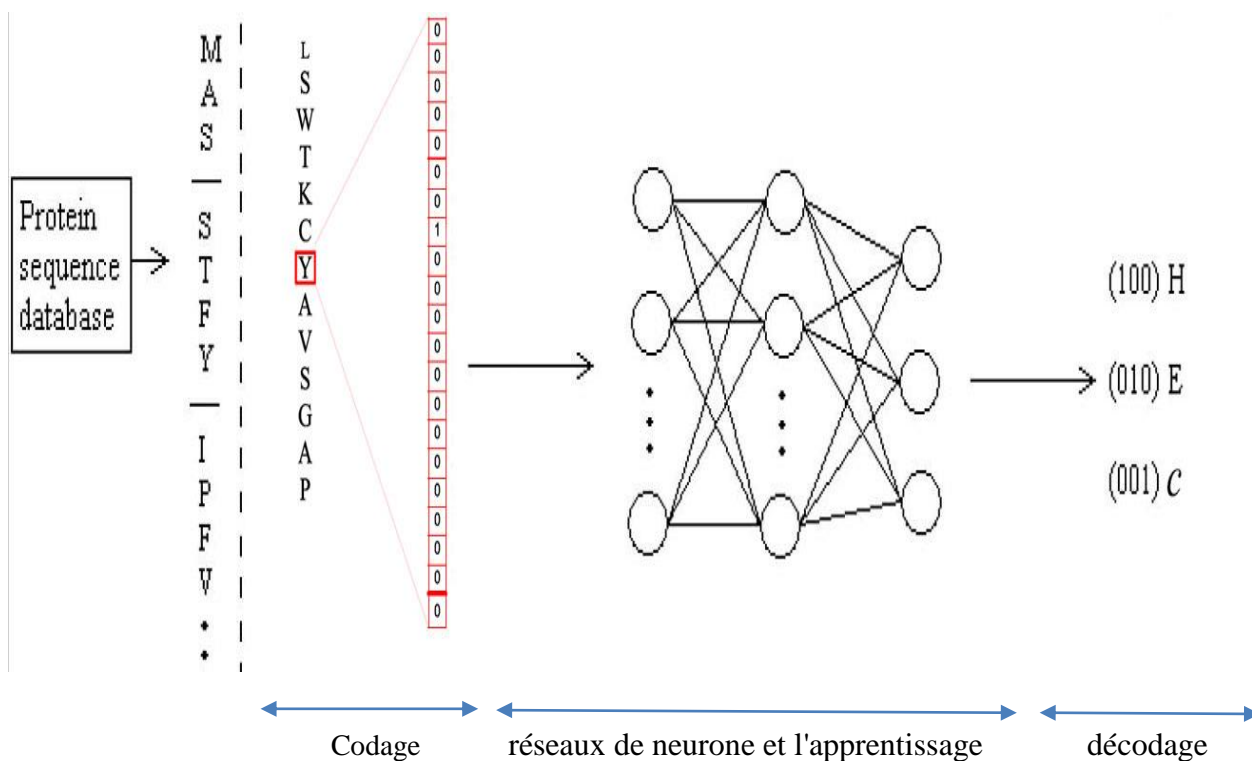


Figure 4.3 : schéma représenté l'algorithme de prédiction.

##### 4.4.1 Codage :

Choix d'une fenêtre de n aminoacides de la séquence en entrée et codés Les acides aminés sous la forme de vecteurs de 0 et 1 de longueur 21 voir la table :

Acides aminés	1-letter symbol	Codage
Alanine	A	1,0
Cysteine	C	0,1,0
Aspartate	D	0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Glutamate	E	0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Phenylalanine	F	0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Glycine	G	0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Histidine	H	0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Isolecine	I	0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0
Lysine	K	0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0
Leucine	L	0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0
Methionine	M	0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0
Asparagine	N	0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0
Proline	P	0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0
Glutamine	Q	0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0
Arginine	R	0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0
Serine	S	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0



Threonine	T	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0
Valine	V	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0
Tryptophan	W	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0
Threonine	Y	0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0

Tableau 4.1 le codage de chaque **Acide aminé**.

## 4.4.2 Réseaux de neurone et l'apprentissage :

Un réseau de neurones est constitué de trois couches de traitement par lesquelles le signal transite (voir figure 4.2) [24] :

1. La couche d'entrée : couche stockant l'information. elle contient le résultat de Codage d'une fenêtre de la séquence
2. la couche cachée : Par la suite, chaque unité  $j$  de la couche cachée somme les signaux en entrée ( $s_{j,in}$ ) qui, à leur tour, transmette un signal vers les unités de sortie  $s_j = 1/(1+\exp(-k.s_{j,in}))$  avec  $k$  constante et  $0 < s_j < 1$ ).
3. La couche de sortie : couche de réponse. Constituée de trois unités, chacune reçoit le signal pondéré  $s_{j,out}$  de chaque unité  $j$  de la couche cachée (poids  $w_i$  entre l'unité  $j$  de la couche cachée et de l'unité  $i$  de la couche en sortie  $s_{j,out} = s_j * w_i$ ). Chaque unité  $i$  de la couche.

Dans l'étape d'apprentissage de réseau de neurone, nous utilisons l'algorithme d'apprentissage supervisé feed-forward et d'erreur de rétro propagation. Nous avons implémenté deux fonctions :

- 1- Une fonction calcule le pourcentage d'erreur.
- 2- Une autre fonction qui modifie les poids des inputs selon le pourcentage d'erreur et la rapidité d'apprentissage.

## 4.3.3 Décodage :

Le résultat de réseau de neurone d'une fenêtre de la séquence est :

L'hélice $\alpha$	H	1 0 0
Le feuillet $\beta$	E	0 1 0
Coudes et boucles :	C	0 0 1

## 4.5 Expérimentation des résultats :

### 4.5.1 Guide pour utiliser l'application :

La première partie sert à saisir la séquence de Protéine soit d'une manière manuelle, soit le lire via un fichier de « Fasta format ».

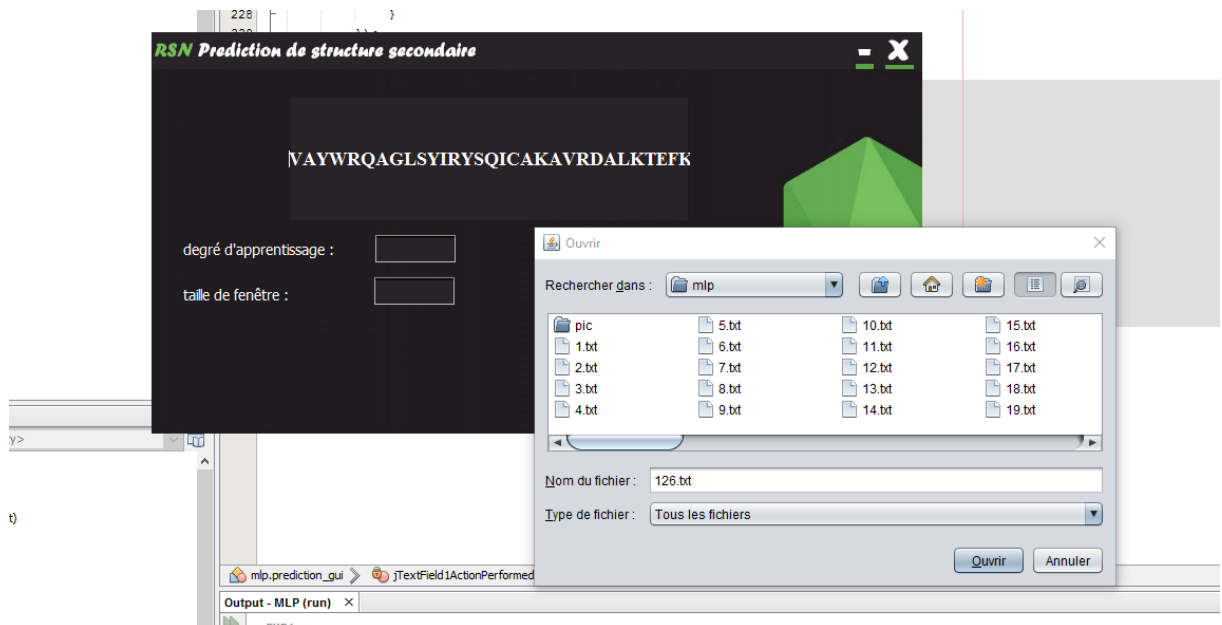


Figure 4.4 : saisir la séquence de Protéine.

En suite l'utilisateur doit saisir le degré d'apprentissage et la taille de fenêtre de protéine

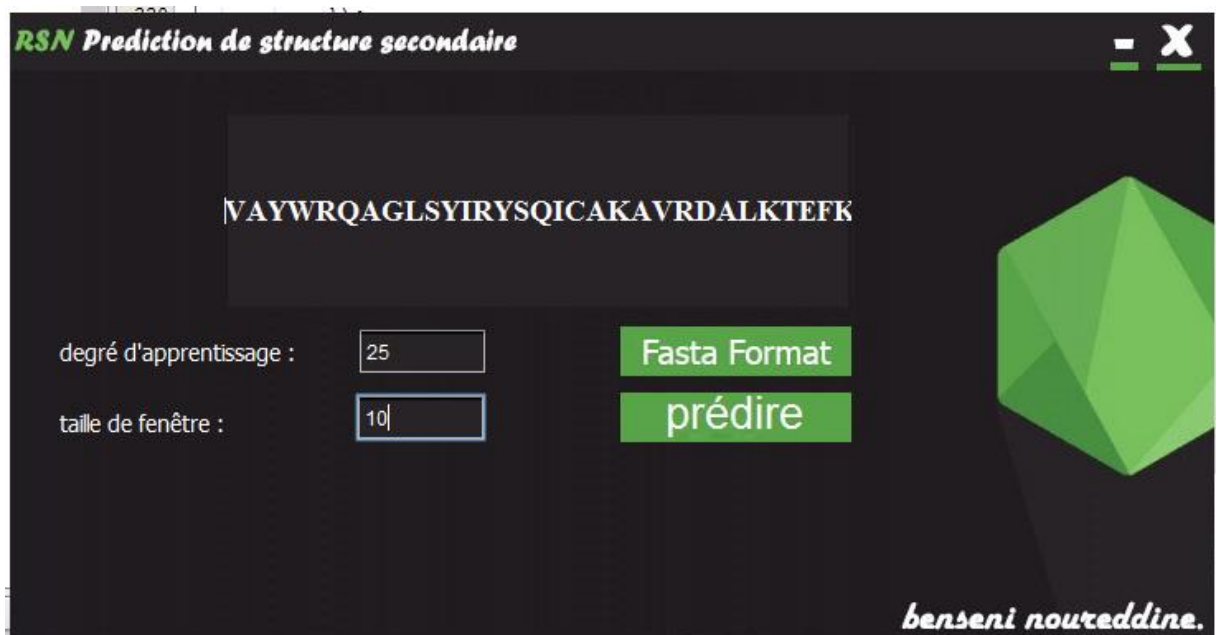


Figure 4.5 : saisir le degré d'apprentissage et la taille de fenêtre.

En suite cliquer sur le bouton « prédire » en attendant les résultats.

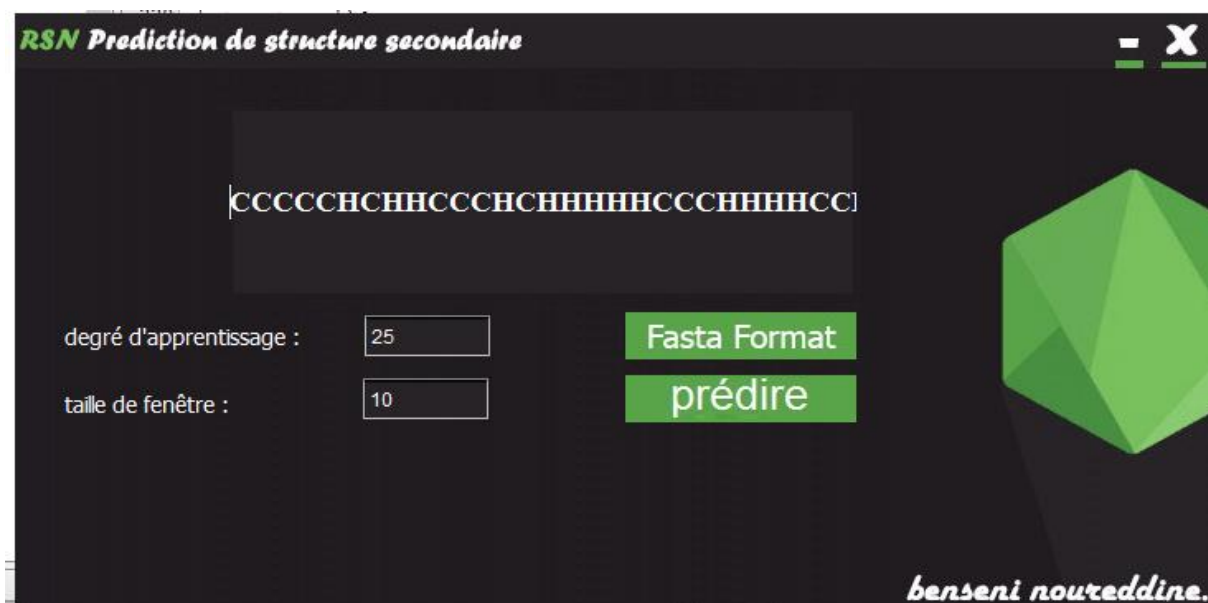


Figure 4.6 : la troisième partie.

### 4.5.2 Evaluation des résultats:

Une fois la prédiction réalisée, une question qui se pose est : est-il possible, et si oui comment, évaluer la qualité des prédictions proposées pour une séquence de structure inconnue. Pour répondre à cette question, il existe des indices de confiance permettant de calculer le degré d'efficacité d'un algorithme de prédiction, l'un des principaux critères est :

#### - Q3 :

L'indice le plus simple et le plus couramment utilisé pour déterminer la véracité d'une prédiction est le taux de reconnaissance Q3 par résidu. Le Q3 représente le pourcentage des résidus correctement prédit [24].

Pour une structure secondaire particulière :

$Q_i = \text{Nombre des résidus correctement prédits de structure } i * 100 / \text{nombre des résidus observés de structure } i$

### 4.5.3 Présentation des résultats en fonction de quelques contraintes :

Nous testons notre algorithme sur 8 instances différentes de protéine. Nous allons présenter dans 2 tableaux la relation de qualité des résultats avec quelques contraintes.

Premièrement nous constituons un tableau qui représente les résultats de chaque instance :

le degré d'apprentissage :	3	6	9	13	16	30
Protéine 1	90 %	82 %	82 %	82 %	80 %	82%
Protéine 2	44%	48%	48%	98%	97%	93%
Protéine 3	35%	44%	46%	46%	56%	62%
Protéine 4	36%	41%	44%	45%	45%	55%
Protéine 5	45%	46%	46%	44%	45%	46%
Protéine 6	44%	47%	47%	47%	48%	49%
Protéine 7	44%	42%	42%	42%	45%	49%
Protéine 8	43%	47%	48%	47%	47%	48%
Q total:	48%	49.6%	50.12 %	56.35%	57.8%	60.5%

Tableau 4.2 représentation des résultats.

Protéine 1 et 2 sont parmi les protéines sur lesquelles nous avons apprendre le réseau de neurones.

Nous constituons un Figure qui montre la relation entre le degré d'apprentissage et la qualité des résultats :

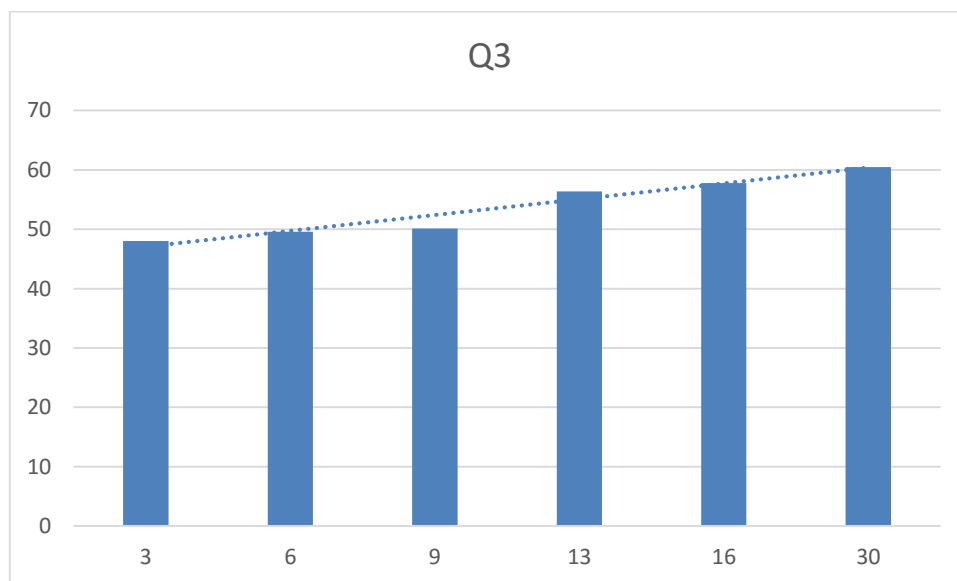


Figure 4.7 : la relation entre le degré d'apprentissage et la qualité des résultats.

Après un ensemble de tests Nous avons remarqué que lorsque nous augmentons le degré d'apprentissage, ça touche positivement la qualité des résultats.

En suite un autre tableau pour la relation de la taille de la fenêtre avec la qualité des résultats avec un degré d'apprentissage 30 :

la taille de la fenêtre :	3	10	13
Protéine 3	55%	50%	62%
Protéine 4	46%	63%	55%
Protéine 5	41%	47%	46%
Protéine 6	56%	49%	49%
Protéine 7	43%	47%	49%
Protéine 8	47%	51%	48%
Q total:	48%	51.1 %	51.5%

Tableau 4.3 : la relation entre la taille de la fenêtre et la qualité des résultats.

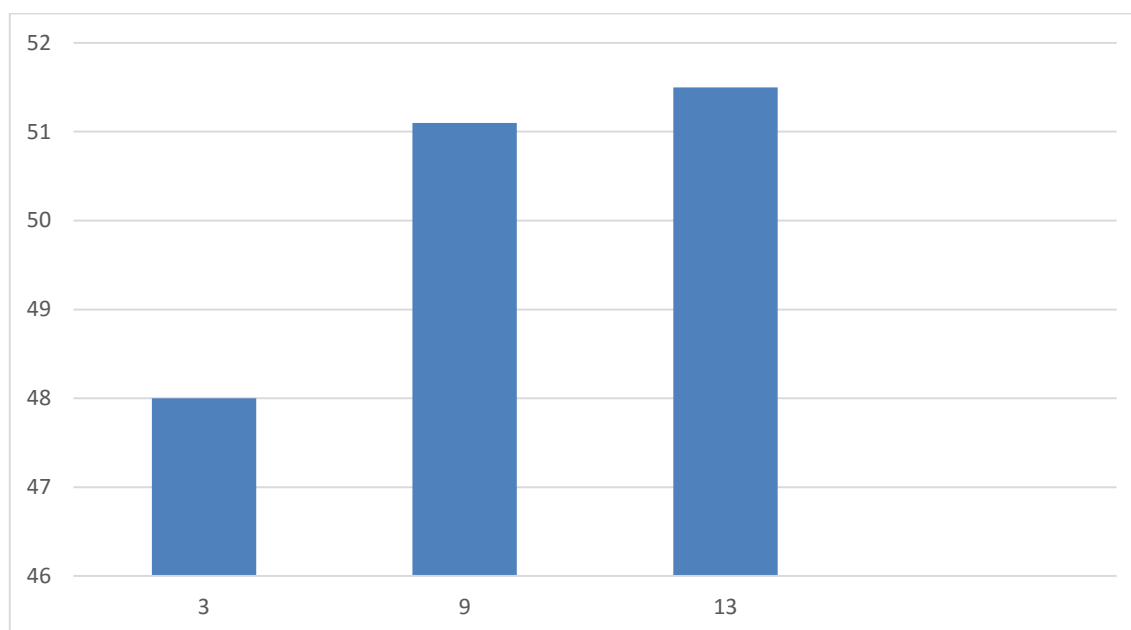


Figure 4.8 : la relation entre la taille de la fenêtre et la qualité des résultats.

Après un ensemble de tests Nous avons remarqué que lorsque nous augmentons le degré d'apprentissage et la taille de la fenêtre, ça touche positivement la qualité des résultats, dans notre implémentation le meilleur résultat de prédiction 63% avec une taille de fenêtre 10 et degré d'apprentissage 30.

### 4.6 Conclusion :

Nous avons présenté dans ce chapitre les outils d'implémentation qui nous ont permis à la réalisation de cette approche, nous avons choisi le langage Java spécialement en considérant sa richesse énorme. Aussi nous avons profondément expliqué la méthode de réseau de neurone qui sert à prédire les structures secondaires des protéines

Enfin nous avons présenté les résultats de cette approche et les analyser en les combinant avec quelques contraintes.

## Conclusion générale :

Au cours de ce travail de master, Nous avons essayé d'adapter et implémenter la méthode d'apprentissage par réseaux de neurones pour résoudre Le problème de la prédiction de structure secondaire d'une protéine. Plusieurs méthodes ont été proposées pour la résolution de ce problème comme les méthodes statistiques, la méthode des plus proches voisins, la méthode de Chaînes de Markov mais aucune ne peut le résoudre efficacement dans tous les cas.

Notre implémentation combine trois étapes, Le premier contient la partie de codage des acides aminés sous la forme de vecteurs de 0 et 1. Dans la deuxième étape nous avons adapté le réseau de neurones multicouches du type feedforward pour prédire la structure secondaire. Dans La dernière étape nous avons décodé le résultat des réseaux de neurones sous la forme de lettres H pour l'hélice  $\alpha$  E pour le feuillet  $\beta$  C pour le coudes.

Nous avons testé notre implémentation sur différents structures de protéines. En fournissant un ensemble de tests de prédiction prend en compte la relation entre le degré d'apprentissage et la taille de la fenêtre et la qualité des résultats.

Nous avons remarqué que lorsque nous augmentons le degré d'apprentissage et la taille de la fenêtre, ça touche positivement la qualité des résultats, dans notre implémentation le meilleur résultat de prédiction 63% avec une taille de fenêtre 10 et degré d'apprentissage 30.

Comme perspective, nous intentons d'hybrider cette approche avec la méthode de plus proche voisin ainsi qu'utiliser la méthode de classification de protéine afin d'améliorer les résultats.

## **BIBLIOGRAPHIE :**

- [1] J.MULLER "Analyse du cytosquelette par des approches bio-informatiques à haut débit de génomique comparative et de transcriptomique.". Thèse de doctorat l'Université Louis Pasteur Strasbourg 1,2006.
- [2] Ramachandran ET Sasisekharan."Conformation of polypeptides and proteins 1968".
- [3] Watson ET Crick. "A structure for deoxyribose nucleic acid." F.H.C, 1953.
- [4] Alain Raisonnier. "Structures Biologiques." Université Paris-VI, 2010.
- [5] V. Derrien, "Heuristiques pour la résolution du problème d'alignement multiple", Thèse de doctorat, Université d'Angers, 2008.
- [6] Isabelle SOURY-LAVERGNE NAVIZET, "MODÉLISATION ET ANALYSE DES PROPRIÉTÉS MÉCANIQUES DES PROTÉINES", Thèse de doctorat l'UNIVERSITÉ PARIS 6, 2004.
- [7] Azaquar, [http://www.azaquar.com/iaa/chimie/ca\\_images/ca\\_proteine3.gif](http://www.azaquar.com/iaa/chimie/ca_images/ca_proteine3.gif), consulté le 15/2/2016.
- [8] D Robert et B Vian. "Element de biologie cellulaire." 2008.
- [9] I.Ruczinski. "Protein Structure Prediction: Secondary Structure. Department of Biostatistics," Johns Hopkins University, 2008.
- [10] G.Chakroun "Prédiction de la structure d'une protéine "2004.
- [11] C. Gibas and P. Jambeck."Introduction à la bioinformatique. O'Reilly", 2002.
- [12]Wikipedia,[https://fr.wikipedia.org/wiki/Prot%C3%A9ine#Pr.C3.A9diction\\_de\\_structure\\_et\\_simulation](https://fr.wikipedia.org/wiki/Prot%C3%A9ine#Pr.C3.A9diction_de_structure_et_simulation) consulté le 18/3/2016.
- [13]Wikipedia,[https://fr.wikipedia.org/wiki/Alignement\\_de\\_s%C3%A9quences](https://fr.wikipedia.org/wiki/Alignement_de_s%C3%A9quences). Consulté le 18/3/2016.
- [14] Cock PJ., Fields CJ., Goto N., Heuer ML. & Rice PM., "The Sanger FASTA file format for sequences with quality scores, and the Solexa/Illumina FASTA variants. ", Nucleic Acids



Research, vol.38, no6, 2010, p.176771 (ISSN 13624962, PMID 20015970, DOI 10.1093/nar/gkp1137).

[15] William R. Pearson, "Documentation des versions 3.x de la suite de programmes FASTA", sur Center for Biological Sequence analysis.

[16] Wikipédia, <https://fr.wikipedia.org/wiki/GenBank>, consulté le 22/4/2016.

[17] Wikipédia, [https://fr.wikipedia.org/wiki/Protein\\_Data\\_Bank#Le\\_format\\_PDB](https://fr.wikipedia.org/wiki/Protein_Data_Bank#Le_format_PDB), consulté le 22/4/2016.

[18] B. Rost. Prediction in id: secondary structure, membrane helices, and accessibility. Methods Biochem Anal, 44:559–87, 2003.

[19] Chou ET Fasman. "Prediction of protein conformation. Biochemistry". Springer, 1974.

[20] M. Zaki ET C. Bystroff. "Protein Structure Prediction." Oxford, 2008.

[21] Blaise Gassend, Charles W. O'Donnell, William Thies, Andrew Lee, Marten van Dijk et Srinivas Devadas. "Secondary Structure Prediction of All-Helical Proteins Using Hidden Markov Support Vector Machines. Computer Science and Artificial Intelligence Laboratory (CSAIL)" 2005.

[22] Juliette Martin. "Prédiction de la structure locale des protéines par des modèles de chaîne de Markov Caché. Université" Paris VII - Denis Diderot, 2005.

[23] B. Bergeron. "Bioinformatics Computing. Prentice Hall PTR", 2002.

[24] B. Messabih et Hafida Bouziane, Belhadri Messabih et Abdellah Chouarfia. Prédiction de la Structure des Protéines par Apprentissage Automatique. SETIT 2009, 2009.

[25] Wikipédia, [https://fr.wikipedia.org/wiki/Structure\\_secondaire](https://fr.wikipedia.org/wiki/Structure_secondaire), le 22/4/2016. le 25/4/2016.

[26] Wikipedia, [https://fr.wikipedia.org/wiki/R%C3%A9seau\\_de\\_neurones\\_artificiels](https://fr.wikipedia.org/wiki/R%C3%A9seau_de_neurones_artificiels), Consulte le 12/05/2016.

[27] Philippe POINCOT "Classification et recherche d'information bibliographique par l'utilisation des cartes auto-organisatrices, applications en astronomie" Thèse de Doctorat université louis pasteur 1999.

- [28] L.Personnaz, I.Rivals, "Réseaux de neurones formels pour la modélisation, la commande, et la classification", CNRS éditions, collection Sciences et Techniques de l'Ingénieur, 2003.
- [29] G.Z winngelsten, " Diagnostic des défaillances : théorie et pratique pour les systèmes industriels ", Ed. Hennes Pane, 1995.
- [30]developpez,<http://alp.developpez.com/tutoriels/intelligence-artificielle/reseaux-de-neurones/>. Consulte le 12/05/2016.
- [31]T.A.Freman, D.M.Skapura,"Neural networks: algorithm, applications and programming techniques ", CNS, Computation and neural systems series, 1992.
- [32] Marc Parizeau,"*Réseaux de Neurones* (Le perceptron multicouche et son algorithme de retropropagation des erreurs) ", Université Laval, Laval, 2004.
- [33]UniversitéLaval3, <http://www.grappa.univ-lille3.fr/polys/apprentissage/sortie005.html>. Consulte le 12/05/2016.
- [34] Wikipedia, [https://en.wikipedia.org/wiki/Unsupervised\\_learning](https://en.wikipedia.org/wiki/Unsupervised_learning). Consulte le 12/05/2016.
- [35] java, [http://www.java.com/fr/download/faq/whatis\\_java.xml](http://www.java.com/fr/download/faq/whatis_java.xml). Consulte le 15/05/2016.
- [36] Wikipédia, <https://fr.wikipedia.org/wiki/NetBeans>, Consulte le 15/05/2016.