



HAL
open science

Classification des séquences d'ADN par les réseaux d'ondelettes

Abdesselem Dakhli

► **To cite this version:**

Abdesselem Dakhli. Classification des séquences d'ADN par les réseaux d'ondelettes . Informatique [cs]. ENIS-SFAX, 2017. Français. NNT: . tel-01771864

HAL Id: tel-01771864

<https://theses.hal.science/tel-01771864v1>

Submitted on 19 Apr 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



THESE

Présentée à

L'École Nationale d'Ingénieurs de Sfax

En vue de l'obtention du

DOCTORAT

Dans la discipline Ingénierie des Systèmes Informatiques

Par

Abdesselem DAKHLI

(Mastère en Informatique)

**Classification des séquences d'ADN par
les réseaux d'ondelettes**

Soutenu le 29 Juin 2017, devant le jury composé de :

M. Mounir BEN AYED (Maître de conférences)	Président
M. Walid MAHDI (Maître de conférences)	Rapporteur
M. Dorra SELLAMI (Professeur)	Rapporteur
M. Abdennaceur KACHOURI (Professeur)	Examineur
M. Chokri BEN AMAR (Professeur)	Directeur de Thèse

DEDICACES

Cette thèse de doctorat est dédiée

A ma très chère mère, pour m'avoir entouré et soutenu durant tout mon parcours éducatif, favorisant ainsi à mon esprit et mes compétences tout développement et toute ampleur, matérialisés en ce mémoire.

A l'âme de mon père Chedly que Dieu l'accueille dans son vaste paradis.

A mes frères et mes sœurs pour leurs sacrifices et leur amour.

A ma femme Wided BAKARI, pour toute l'attention et la patience qu'elle m'a accordées.

A tous mes amis, et surtout Massoud J., Taher.k, Mourad M., wadie N., Fouead k., Malek W., Slah B., Abdelbasset S., Rafik J., Zouhair H., Ali B., Nabil F., Mohamed R., Mabrouk A., sofiène A., Mohamed A.,Radhoine B.,Taher L.,Arafet H., Ali B., Houcine T, Houcine S .,Adel B.,Anes E.,Lazher L., Mabrouk k., Samir T, Samir K.,Jemaa L., pour leurs encouragements et leur aide.

Abdesselem

REMERCIEMENTS

Je tiens à présenter mes remerciements, les plus vifs et les plus sincères à tous ceux qui ont contribué à la réalisation de ce travail.

J'exprime toute ma reconnaissance et ma haute considération aux membres de mon jury de thèse qui ont bien voulu me faire l'honneur de juger mon travail :

- Monsieur **Mounir BEN AYED**, maître de conférences à la FSS et président de ce jury.
- Monsieur **Walid MAHDI**, maître de conférences à l'ISIMS et rapporteur de cette thèse.
- Madame **Dorra SELLAMI**, professeur à l'ENIS et rapporteur de cette thèse.
- Monsieur **Abdennaceur KACHOURI**, professeur à l'ENIS et examinateur de cette thèse.

Je remercie plus particulièrement Monsieur **Chokri BEN AMAR**, professeur à l'ENIS, pour avoir bien voulu m'encadrer et pour ses précieux conseils qu'il m'a fournis durant la réalisation de cette thèse.

Je remercie également Monsieur **Adel ALIMI**, Professeur à l'ENIS et responsable du laboratoire REGIM, pour avoir bien voulu m'accepter membre de son laboratoire de recherche.

Bien évidemment, je remercie Monsieur **Wajdi BELLIL** d'avoir accepté de co-encadrer ce travail.

Je tiens à remercier aussi Monsieur **Mourad Zaied** pour ses précieux conseils et pour ses remarques pertinentes.

Je remercie également tous mes collègues à l'ISG de Gabès, à l'ENIG et au laboratoire REGIM pour leur gentillesse et leur amabilité.

Table des matières

Table des figures	vii
Liste des tableaux	xi
liste d'abréviations et de symboles	xiii
Introduction générale	1
1 Etat de l'art sur la classification des séquences d'ADN	4
1.1 Introduction	6
1.2 Contexte de la thèse	6
1.2.1 La bioinformatique	6
1.2.2 La Biologie Moléculaire pour un Bio-informaticien	7
1.2.3 Mutation d'ADN	17
1.2.4 Réplication ou duplication d'ADN	18
1.2.5 L'alphabet Biologie	20
1.3 Définition et but de classification	21
1.3.1 Homologie et similitude des gènes	22
1.3.2 Principes de base de l'alignement de séquences à classifier	23
1.4 Le problème de classification des données	24
1.4.1 Définition	25
1.4.2 Types de classification	25
1.5 La classification des séquences d'ADN	31
1.5.1 Description des problèmes de classification des séquences d'ADN	31
1.5.2 Motivations de classification des séquences d'ADN	32
1.5.3 Problématique de classification des séquences d'ADN	33
1.5.4 Objectifs principaux	34
1.5.5 Etat de l'art de classification des séquences d'ADN	36

1.5.6	Les limites des méthodes existantes	48
1.6	Conclusion	49
2	Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction	51
2.1	Introduction	53
2.2	Les Réseaux de neurones	54
2.2.1	Définition	54
2.2.2	Historique	54
2.2.3	Application	56
2.2.4	Fondement biologique des neurones	57
2.2.5	Principe de fonctionnement des neurones	58
2.2.6	Les réseaux de neurones artificiels	59
2.2.7	Exemples d'architecture des réseaux de neurones	60
2.3	Les réseaux d'ondelettes	63
2.3.1	Définition	63
2.3.2	Les ondelettes	64
2.3.3	Architecture des réseaux d'ondelettes	75
2.3.4	De la transformée inverse aux réseaux d'ondelettes	77
2.3.5	Comparaison des réseaux d'ondelettes aux réseaux de neurones	78
2.4	Apprentissage des réseaux d'ondelettes	78
2.4.1	L'apprentissage supervisé	79
2.4.2	L'apprentissage non supervisé	80
2.4.3	Apprentissage d'un réseau d'ondelettes par l'analyse multirésolution	81
2.4.4	Apprentissage spécifique aux réseaux d'ondelettes	84
2.5	Téchniques de construction des réseaux d'ondelettes	85

2.5.1	Les méthodes incrémentales utilisées pour la construction des réseaux d'ondelettes	85
2.5.2	Construction de réseaux d'ondelettes par l'algorithme pyramidal	92
2.5.3	Construction de réseaux d'ondelettes par des techniques basées sur la transformée discrète	92
2.5.4	Construction d'un réseau d'ondelettes basée sur l'analyse fréquentielle	93
2.5.5	Construction d'un réseau d'ondelettes en utilisant la théorie des ondelettes orthogonales	94
2.5.6	Construction d'un réseau d'ondelettes pour un système adaptatif	94
2.5.7	Construction d'un réseau d'ondelettes basée sur la construction des frames	95
2.5.8	Calcul direct des pondérations (poids) de connexion	99
2.6	Les méthodes de sélection d'ondelettes	100
2.6.1	La technique de choix par orthogonalisation	100
2.6.2	La méthode des moindres carrées orthogonales (OLS)	102
2.7	Conclusion	107

3 Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes **108**

3.1	Introduction	109
3.2	Proposition d'une procédure de construction et d'apprentissage d'un réseau d'ondelettes pour classifier les séquences d'ADN	110
3.2.1	Conversion d'une séquence d'ADN	113
3.2.2	L'optimisation des réseaux d'ondelettes	117
3.2.3	Le principe de l'algorithme à suivre pour construire notre réseau d'ondelettes	120

3.2.4	Construction de la bibliothèque d'ondelettes	121
3.2.5	La méthode de sélection des ondelettes pour notre ap- proche	122
3.2.6	Calcul des poids en utilisant la méthode Lasso	124
3.2.7	La procédure d'apprentissage	129
3.2.8	La phase de reconnaissance	134
3.3	Conclusion	138
4	Etude Expérimentale de l'approche proposée	139
4.1	Introduction	140
4.2	Les critères d'évaluation	141
4.2.1	Précision	141
4.2.2	Rappel	142
4.2.3	F-mesure	142
4.2.4	Matrice de confusion	142
4.3	Evaluation et interprétation des résultats obtenus	144
4.3.1	Les bases de données des séquences d'ADN utilisées . .	144
4.3.2	Interprétation des résultats obtenus	146
4.4	Conclusion	173
	Conclusion et perspectives	174
	Publications Scientifiques	176
	Bibliographie	178

Table des figures

1.1	Acteurs de la bioinformatique.	7
1.2	Schéma représentant les données biologiques.	8
1.3	Acide phosphorique.	8
1.4	Schéma d'une base d'azote.	9
1.5	Construction d'un nucléotide.	9
1.6	Un brin d'ADN ou polynucléotide.	9
1.7	Construction du 2ème brin.	10
1.8	Double brins d'ADN(Forme d'échelle).	10
1.9	Double brins d'ADN(Forme hélicoïdale).	10
1.10	Un brin d'ARN.	11
1.11	Le mécanisme de transcription.	12
1.12	La complémentarité des bases azotées d'un brin d'ADN.	12
1.13	Transcription d'un brin d'ADN.	13
1.14	Association de l'Adénine(A) au Thymine(T).	13
1.15	Association de l'Uracile(U) à l'Adénine (A).	13
1.16	L'ajout des nucléotides d'un brin d'ARN.	14
1.17	Construction d'un brin d'ARN.	14
1.18	Construction d'un brin d'ARN et reconstruction d'un brin d'ADN.	14
1.19	Synthèse d'un brin d'ARN portant l'information génétique.	15
1.20	Migration d'un brin d'ARN de noyau vers le cytoplasme pour produire une protéine.	15
1.21	Production d'une protéine à partir d'un brin d'ARN.	16
1.22	Le code génétique présentant les codons et les acides aminés correspondants.	16
1.23	Processus général de la synthèse des protéines.	16
1.24	Duplication d'une molécule d'ADN.	19
1.25	La croissance du nouveau brin d'ADN.	20

1.26	Processus général d'une classification	25
1.27	Les couches et les connexions dans la carte auto-organisatrice .	30
1.28	Sélection du neurone vainqueur dans la carte auto-organisatrice	31
1.29	Les stratégies de classification utilisées.	48
2.1	Structure d'un neurone	57
2.2	La synapse d'un neurone	58
2.3	Perceptron multicouche à une couche cachée	61
2.4	Un réseau à fonction radiale de base	62
2.5	Prémière dérivée de la fonction Bêta(en trait bleu) et celle di- latée translatée(en trait vert)	68
2.6	Différentes formes d'ondelettes des dérivées de la fonction Bêta	73
2.7	Translation et dilatation des dérivées de la fonction Bêta . . .	74
2.8	Dérivée première dilatée et translatée de la fonction Bêta 2D .	74
2.9	Type 1 de réseaux d'ondelettes	76
2.10	Type 2 de réseaux d'ondelettes	77
2.11	Décomposition d'un signal g en appliquant l'analyse multirésolu- tion	84
2.12	Reconstruction d'un signal g	84
2.13	L'algorithme de l'apprentissage supervisé.	86
2.14	L'état initial d'un réseau utilisant l'algorithme en cascade (sans couche cachée)	89
2.15	L'ajout d'une première couche cachée au réseau utilisant l'al- gorithme en cascade	90
2.16	L'ajout d'une deuxième couche cachée au réseau utilisant l'al- gorithme en cascade	90
2.17	Construction d'un réseau de neurones par l'algorithme en cascade	91
2.18	Base orthogonal	98
2.19	Base biorthogonal	98

2.20	Frame orthogonal	98
2.21	Interprétation géométrique du choix (sélection) des ondelettes par orthogonalisation	102
3.1	Processus général de l'approche proposée	112
3.2	Séquence d'ADN $X[n]$	114
3.3	Le signal d'une séquence d'ADN en utilisant la densité spectral de puissance (Power Spectrum)	116
3.4	Les sept premières ondelettes de la bibliothèque et un signal à analyser [Zaied 2008]	118
3.5	Topologie du réseau d'ondelettes à concevoir pour classer les séquences d'ADN	121
3.6	Phase d'apprentissage pour notre réseau d'ondelettes multi- entrées multi-sorties	130
3.7	Phase d'apprentissage	134
3.8	Phase de reconnaissance	135
3.9	L'approche proposée	137
4.1	Codes-barres d'ADN	145
4.2	Répresentation des classes obtenues à partir de la base de don- nées bacillus-subtilis	149
4.3	Comparaison des séquences d'ADN de la classe 1 pour la base de données bacillus-subtilis	150
4.4	Evolution de la précision pour chaque base de données d'ADN.	151
4.5	Evolution du temps d'apprentissage de notre approche.. . . .	152
4.6	Temps d'apprentissage de l'approximation des signaux de sé- quences d'ADN	154
4.7	Arbre phylogénique	156
4.8	Temps d'exécution en fonction de la taille de séquence d'ADN	160

4.9	Comaparaision des taux de classification de l'approche proposée avec d'autres classifieurs	163
4.10	Les résultats de classification d'une méthode basée sur les ondelettes utilise les vecteurs de caractéristiques (OVC)des différents ensembles de données ayant des tailles différentes.	165
4.11	Les résultats de classification de notre approche (WNN-Lasso) qui utilise les vecteurs de caractéristiques des différents ensembles de données ayant des tailles différentes.	166
4.12	F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG100.	167
4.13	F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG200.	168
4.14	F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG300.	168

Liste des tableaux

1.1	Les autres méthodes de classification des séquences d'ADN . . .	46
2.1	Les différentes formes et constatations de la fonction Bêta	71
3.1	Codification binaire(4-bits) des nucléotides	114
3.2	Codification des nucléotides en utilisant EIIP	114
4.1	Matrice de confusion	143
4.2	Distribution des séquences d'ADN d'une base de données des bactéries	147
4.3	Précision de la classification des bases de données d'ADN en utilisant RNA,SVM et RO(notre approche)	147
4.4	Le Temps d'apprentissage de notre approche (RO) et les méthodes RNA et SVM	151
4.5	Distribution des bases de données en deux échantillons (Apprentissage et test)[A.Dakhli 2014b]	153
4.6	Matrice de confusion de classification des séquences d'ADN en utilisant notre réseau d'ondelettes	155
4.7	Taux de classification des séquences d'ADN en utilisant le réseau de neurones probabiliste(RNP) et le réseau d'ondelettes RO (notre approche)	155
4.8	MSE de l'approximation d'un signal d'une séquence d'ADN en utilisant le réseau d'ondelettes[A.Dakhli 2015][A.Dakhli 2016] .	159
4.9	Matrice de confusion de la classification de barcodes de séquences d'ADN	161
4.10	Taux de classification qui concerne notre approche et des autres méthodes (SVM,Jrip,J48,Naive Bayes)	162

4.11 Bases de données des sequences d'ADN des espèces microbiennes (HOG100, HOG200 et HOG300)	164
4.12 Distribution des données d'apprentissage et de test Données .	164
4.13 Les résultats de classification de l'approche proposée (WNN- Lasso) et les autres méthodes basées sur le principe de l'aligne- ment pour les différents ensembles de données	169
4.14 Temps d'exécution des méthodes pour les différents ensembles de données	171
4.15 Erreur quadratique moyenne(EQM) d'approximation des sé- quences d'ADN en utilisant l'approche proposée.	172

Liste d'abrégations et de symboles

E : Erreur.

μ_b : Pas d'apprentissage du paramètre de translation b .

μ_a : Pas d'apprentissage du paramètre de dilatation a .

μ_k : Pas de gradient.

μ_w : Pas d'apprentissage du paramètre de coefficient w .

Ψ : La dérivée de l'ondelette Bêta.

θ : Seuil d'activation du neurone.

w_i : Coefficients associés aux fonctions d'ondelettes.

w_{ji} : Poids des entrées.

x_i : Signal d'entrée du neurone artificiel i .

$y(t)$: La sortie réelle obtenue par le réseau.

y_d : La sortie désirée.

y_i : La sortie du neurone artificiel i .

a : Paramètre de dilatation.

ADN : Acide DésoxyriboNucléique.

A : Adénine.

T : Thymine.

C : Cytosine.

G : Guanine.

AMR : L'analyse multirésolution.

ARN : Acide Ribonucléique.

b : Paramètre de translation.

CAH : Classification ascendante hiérarchique.

DSP : La Densité Spectrale de Puissance(Power Spectral Density).

DVS : Décomposition des valeurs singulières.

$EIIP$: Electron-ion interaction pseudopotential.

EQM : L'Erreur Quadratique Moyenne.

Lasso : Least Absolute Shrinkage and Selection Operator (Rétrécissement moins absolue et l'opérateur de sélection).

LTS : Least Trimmed Squares (les moindres carrés tronqués).

MCP : La méthode des moindres carrés itératifs pondérés (iteratively reweighted least squares (IRLS)).

MSE : Le carré moyen des erreurs ou erreur quadratique moyenne (Mean square Error).

NSRMSE : L'erreur quadratique moyenne à racine normalisée (Normalized Root Mean Square Error).

OLS : La méthode des moindres carrés orthogonales.

OVC : Les ondelettes qui utilisent les vecteurs de caractéristiques (wavelet based feature vector (WFV)).

PMC : Perceptron multicouche.

RBF : Réseau à Fonction Radiale de Base.

RNA : Le réseau de neurones artificiel.

RNP : Le réseau de neurones probabiliste.

RO : Le réseau d'ondelettes.

SE : Les scores élémentaires.

SOM : Self Organizing Maps ou carte auto-organisatrices.

SP : Les scores des pénalités.

SVM : Machines à support de vecteurs.

TACOA : Classificateur d'analyse de la composition taxonomique (The Taxonomic Composition Analysis classifier).

TETRA : Classificateur qui utilise les fréquences tétranucléotides.

TF : Le Transformée de Fourier.

WNN : Wavelet Neural Networks.

Introduction générale

La bioinformatique est le traitement automatique de l'information biologique. Ce traitement permet d'analyser et d'interpréter les connaissances obtenues d'une façon claire et significative. La clarté des résultats aide le biologiste à prendre des décisions. L'application de l'informatique dans le domaine biologique traite plusieurs concepts qui concernent surtout le niveau moléculaire des cellules de l'organisme ou aussi les traitements manipulant les séquences d'ADN. Dans le cadre des travaux de cette thèse, nous avons essayé de construire une approche qui permet de classifier les séquences d'ADN, comme étant un support de l'information génétique de l'organisme. La classification des séquences d'ADN se rapporte à la comparaison de deux séquences d'ADN ou plus. Cette notion est particulièrement importante dans de nombreux domaines biologiques. En effet, elle permet par exemple de détecter les différences ou les similarités entre les génomes, d'assurer l'analyse génomique et par la suite la détection des maladies et des corps inconnus. La classification des séquences d'ADN nous permet aussi de faire une analyse ou un diagnostic génétique pour dépister une éventuelle maladie génétique et analyser les risques de maladie de cancers, de diabète, d'épilepsie, etc. Les travaux de recherche présentés dans cette thèse concernent le développement d'une méthode de classification automatique qui permettra de classifier les séquences d'ADN. Cette classification va être utilisée pour extraire des connaissances et des informations biologiques qui vont être analysées et interprétées pour la prise de décision. Nous formulons ainsi les questions traitées dans le cadre de cette thèse comme suit :

- Comment doit-on utiliser les réseaux d'ondelettes pour construire un classificateur ?
- Comment la classification des séquences d'ADN, peut-elle être appli-

quée au domaine biologique ?

- Comment peut-on trouver l'ensemble des groupes (séquences) dont les membres sont très similaires, mais distants des autres membres sur la base de leur profil d'expression ?
- Comment peut-on ressortir des groupes de séquences qui ont la même fonction biologique ou de même structure ?
- Etc.

Les réponses à ces questions seront à la base des contributions de notre thèse. Notre manuscrit est structuré en quatre chapitres qui sont définis comme suit : Nous allons présenter, dans le premier chapitre, la notion de bioinformatique comme étant une discipline qui vise le traitement automatique de l'information biologique. Dans ce chapitre, nous allons, définir d'une façon détaillée le contexte de cette thèse qui touche les concepts et les notions biologiques. Ce chapitre s'intéresse à étudier et à connaître la partie moléculaire d'une cellule biologique, spécifiquement les séquences d'ADN comme étant un support de l'information génétique qui pilote la synthèse des protéines au niveau des cellules. Ces protéines permettent d'assurer les fonctions biologiques dans les cellules.

De même, nous dresserons l'état de l'art concernant la classification des séquences d'ADN. Ce chapitre présente le problème de classification des séquences d'ADN et les méthodes qui sont utilisées pour résoudre ce problème. L'objectif de cette partie est d'étudier et comprendre les limites des méthodes appliquées pour classifier les brins d'ADN ; nous présenterons le problème de classification des séquences d'ADN et les motivations à la fois applicatives et théoriques qui ont poussé la communauté scientifique à se pencher sur ce problème. Nous y trouverons en particulier les principaux résultats de classification des brins d'ADN.

Nous ferons un tour d'horizon sur les des différents problèmes de classification visant à montrer dans quels cas les difficultés sont bien résolues en termes de

taille et de nombre des instances traitées et dans quels cas ils ne le sont pas. Le deuxième chapitre, sera consacré à la partie théorique des réseaux de neurones et des réseaux d'ondelette et leurs applications. Nous aborderons les principales architectures de réseaux de neurones que nous retrouvons dans la littérature et nous présenterons les relations qui existent entre les deux types de réseaux ainsi que et les différentes architectures des réseaux d'ondelettes. Dans le troisième chapitre nous présenterons l'approche retenue dans le cadre de cette thèse. Cette approche se base sur un classificateur performant pour assigner un échantillon contenant plusieurs séquences d'ADN. Enfin, le quatrième chapitre présentera la partie expérimentale de l'approche retenue pour la classification des séquences d'ADN. Nous allons présenter puis analyser les différents résultats obtenus en utilisant plusieurs critères et métriques qui permettent de mesurer la performance de notre approche du point de vue complexité et pertinences comme à travers des métriques comme : Précision, Rappel, Matrice de confusion. Notre manuscrit sera clôturé par une conclusion générale et quelques perspectives d'amélioration.

Etat de l'art sur la classification des séquences d'ADN



Sommaire

1.1	Introduction	6
1.2	Contexte de la thèse	6
1.2.1	La bioinformatique	6
1.2.2	La Biologie Moléculaire pour un Bio-informaticien	7
1.2.3	Mutation d'ADN	17
1.2.4	Réplication ou duplication d'ADN	18
1.2.5	L'alphabet Biologie	20
1.3	Définition et but de classification	21
1.3.1	Homologie et similitude des gènes	22
1.3.2	Principes de base de l'alignement de séquences à classifier	23
1.4	Le problème de classification des données	24
1.4.1	Définition	25
1.4.2	Types de classification	25
1.5	La classification des séquences d'ADN	31
1.5.1	Description des problèmes de classification des séquences d'ADN	31
1.5.2	Motivations de classification des séquences d'ADN	32
1.5.3	Problématique de classification des séquences d'ADN	33
1.5.4	Objectifs principaux	34
1.5.5	Etat de l'art de classification des séquences d'ADN	36
1.5.6	Les limites des méthodes existantes	48
1.6	Conclusion	49

1.1 Introduction

Ce chapitre dresse le contexte et l'état de l'art de la classification des séquences d'ADN. Il expose les motivations à la fois applicatives et théoriques qui ont poussé la communauté scientifique à se pencher sur ce problème. Nous y trouverons en particulier les principaux résultats de classification des brins d'ADN. Ce tour d'horizon de différents problèmes de classification vise à montrer dans quels cas les problèmes sont bien résolus en termes de taille et du nombre des instances traitées et dans quels cas ils ne le sont pas.

1.2 Contexte de la thèse

1.2.1 La bioinformatique

La bioinformatique est un domaine de recherche qui analyse et interprète des données biologiques, au moyen de méthodes informatiques, afin de créer de nouvelles connaissances en biologie.[Burge 2002]. Elle permet d'étudier 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 bioinformatique actuelle se concentre surtout sur l'étude des séquences d'ADN et sur le repliement des protéines, donc travaille surtout au niveau moléculaire. De nombreux bioinformaticiens travaillent également à l'élaboration d'outils biologiques permettant de résoudre des problèmes de l'informatique classique.

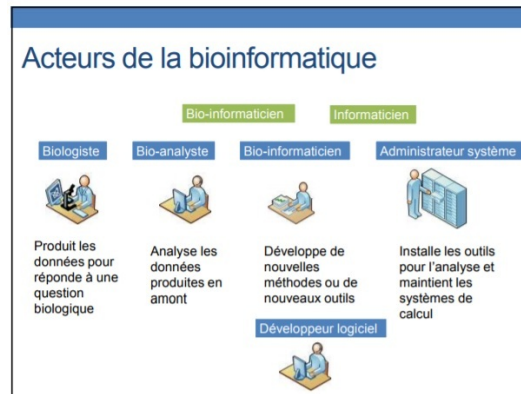


FIGURE 1.1 – Acteurs de la bioinformatique.

La classification des séquences d'ADN constitue une tâche fondamentale pour plusieurs applications en bioinformatique. Afin d'évaluer une classification donnée, plusieurs méthodes de validation de regroupements sont utilisées[Cohen 2004].

1.2.2 La Biologie Moléculaire pour un Bio-informaticien

1.2.2.1 L'ADN (Acide Désoxyribonucléique)

L'ADN est un acide nucléique. Il est un support de l'information génétique et de sa transmission au cours des générations (hérédité). Il est un constituant principal des chromosomes. La Figure 1.1 montre un schéma abstrait d'une cellule. Il y a un noyau contenant l'ADN. Les protéines sont à l'intérieur de la cellule, mais en dehors du noyau. Les acides nucléiques y compris l'ADN et l'ARN, forment le matériel génétique de tout l'organisme. Ce sont toutes les informations dont a besoin un organisme pour fonctionner ainsi que toutes les caractéristiques héréditaires. Ce sont des molécules structurées en chaîne, composées des nucléotides [Watson 1953]. Les molécules d'ADN sont les plus grosses molécules du monde vivant et présentant dans tous les organismes vivants. L'ADN se présente sous forme d'une double hélice composée de deux brins enroulés l'un autour de l'autre ; nous disons que cette molécule est bicaaténaire (contrairement à l'ARN, qui est monocaténaire). Chacun de ces brins

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

est formé par une série de bases dites puriques (Guanine(G) ; Adénine(A)) et pyrimidiques (Cytosine(C) ; Thymine(T)).

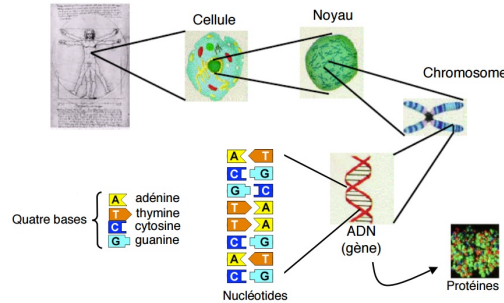


FIGURE 1.2 – Schéma représentant les données biologiques.

Un nucléotide d'ADN (Figure 1.4) a trois composants : un sucre (désoxyribose), un composant d'acide phosphorique (phosphate) (Figure 1.2), et une base d'azote (un des quatre types : Adénine ou Adénosine (A), Guanine (G), Cytosine (C) et Thymine (T)). La molécule d'ADN peut être en simple brin ou double brin. Un brin simple (aussi appelé polynucléotide) est un Polymère linéaire (Figure 1.5).

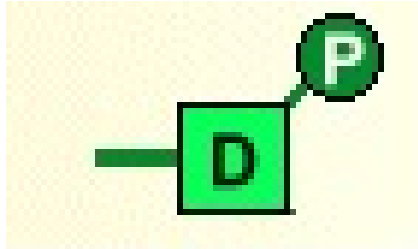


FIGURE 1.3 – Acide phosphorique.

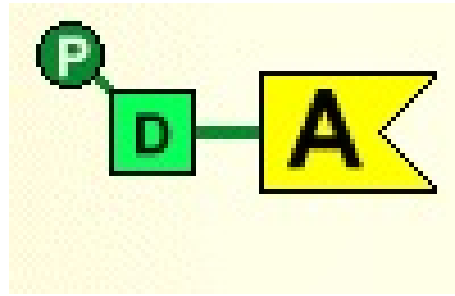


FIGURE 1.4 – Schéma d'une base d'azote.

Nous représentons une molécule d'ADN 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 se coller ensemble pour former une épine dorsale continue. Ceci donne une forme d'échelle (Figure 1.7). La forme d'échelle se torde sur elle même pour donner une forme hélicoïdale (Figure 1.8). Cette forme est la célèbre " double hélice ", explorée par Crick et Watson en 1953[Watson 1953]. Les bases ou nucléotides (A, T, C, G) s'organisent en paires selon une complémentarité exclusive : A-T et G-C. Cette complémentarité assure un enroulement quasi-parfait en hélice droite de deux chaînes sucre-phosphate qui portent ces nucléotides [Alberts 2014]. La structure est stabilisée par l'interaction (liaisons d'hydrogène) entre les bases et l'empilement successif des paires de nucléotides (Figures 1.6, 1.7).

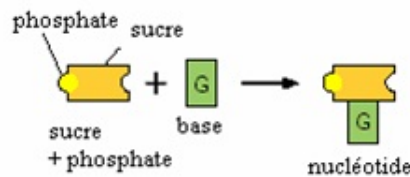


FIGURE 1.5 – Construction d'un nucléotide.

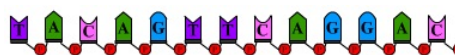


FIGURE 1.6 – Un brin d'ADN ou polynucléotide.

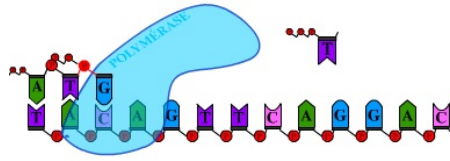


FIGURE 1.7 – Construction du 2ème brin.

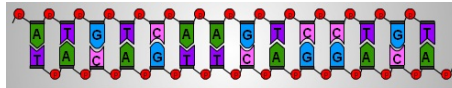


FIGURE 1.8 – Double brins d'ADN(Forme d'échelle).

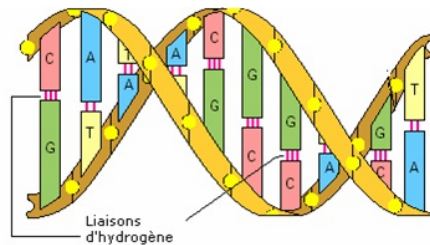


FIGURE 1.9 – Double brins d'ADN(Forme hélicoïdale).

1.2.2.2 Les Chromosomes

Les chromosomes sont des éléments du noyau cellulaire en nombre constant, qui détermine 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. Nous les appelons les chromosomes X et l'autre, beaucoup plus court est appelé chromosome Y.

1.2.2.3 L'ARN

L'ARN (Acide Ribonucléique) ressemble énormément à l'ADN (figure 1.9), mais il y a des différences telles que [Alberts 2014] :

- Le désoxyribose est sucre de l'ADN alors que le ribose est celui de l'ARN ;
- La Thyminine (T) de l'ADN est remplacée par l'uracile (U) dans l'ARN ;
- L'ARN peut s'apparier avec un autre ARN complémentaire vu que 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.



FIGURE 1.10 – Un brin d'ARN.

La transcription (de l'ADN à l'ARN) :

La transcription se fait dans le noyau de la cellule (Figure 1.10).

Mécanisme :

- Copie d'un brin d'ADN sur la portion d'un gène en une séquence nucléotidique complémentaire formant le brin d'ARNm (m comme messenger).
- L'ADN se « déroule » au niveau d'un gène codant pour une protéine donnée grâce à un enzyme appelé : l'ARN polymérase.
- Une des deux séquences de l'ADN à savoir le brin informatif (ou transcrit ou 3'-5') sert de modèle à la synthèse de l'ARNm (Figure 1.11, 1.12).
- Chaque nucléotide du brin d'ADN « attire » un nucléotide complémentaire à l'exception de l'Uracile qui substitue la Thyminine sur l'ARNm

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

(Figure 1.13, 1.14, 1.15).

- L'ordre de nucléotides de l'ARNm est imposé par l'ordre de ceux de l'ADN (Figure 1.16).
- L'ARNm se détache et migre hors du noyau cellulaire dans le cytoplasme en sortant par les pores nucléaires (Figure 1.17).
- Réassociation des brins d'ADN lorsque l'ARN polymérase se détache. Le brin d'ARNm est similaire à celui de l'ADN non transcrit sauf que l'Uracile remplace la Thymines dans la séquence (Figure 1.14).

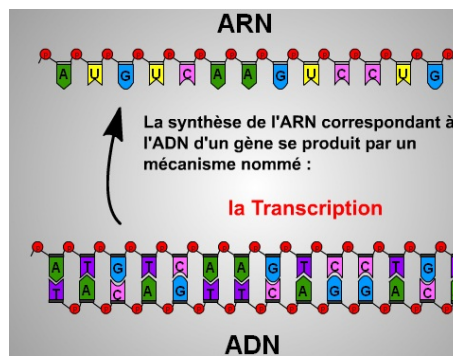


FIGURE 1.11 – Le mécanisme de transcription.

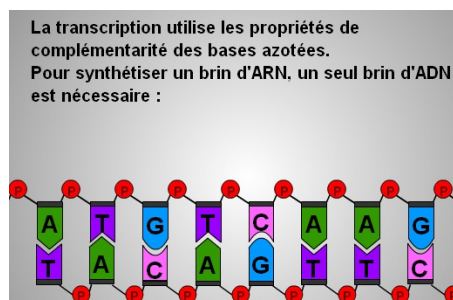


FIGURE 1.12 – La complémentarité des bases azotées d'un brin d'ADN.

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

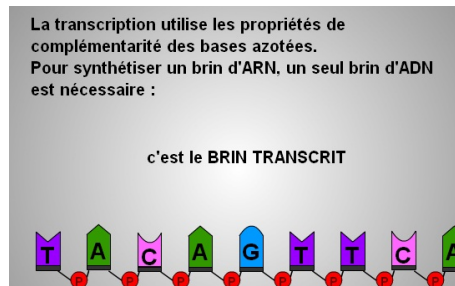


FIGURE 1.13 – Transcription d'un brin d'ADN.

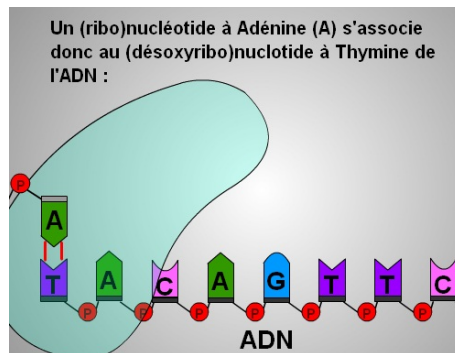


FIGURE 1.14 – Association de l'Adénine(A) au Thymines(T).

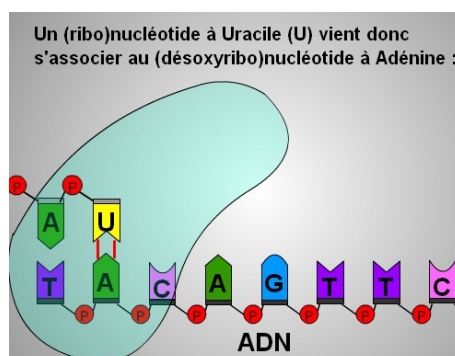


FIGURE 1.15 – Association de l'Uracile(U) à l'Adénine (A).

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

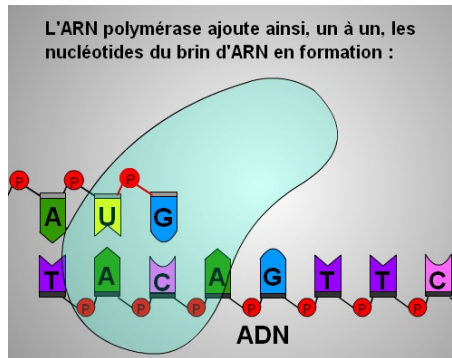


FIGURE 1.16 – L'ajout des nucléotides d'un brin d'ARN.

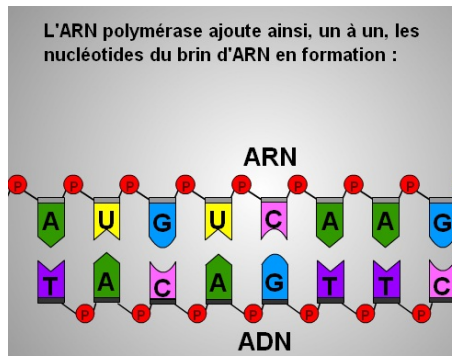


FIGURE 1.17 – Construction d'un brin d'ARN.

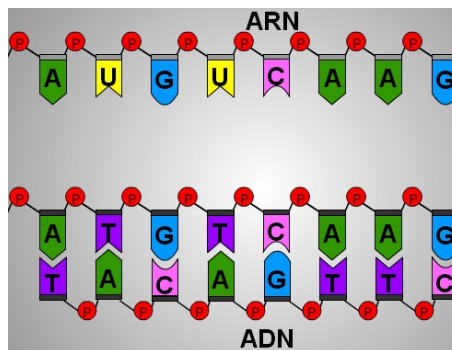


FIGURE 1.18 – Construction d'un brin d'ARN et reconstruction d'un brin d'ADN.

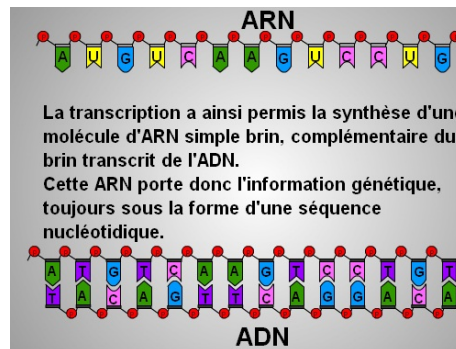


FIGURE 1.19 – Synthèse d'un brin d'ARN portant l'information génétique.

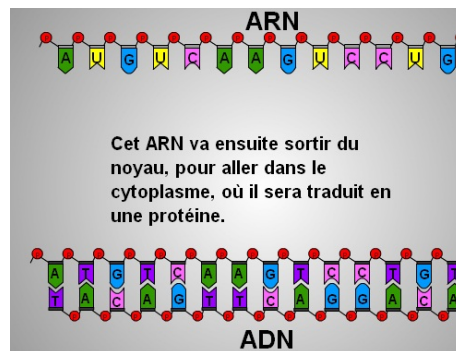


FIGURE 1.20 – Migration d'un brin d'ARN de noyau vers le cytoplasme pour produire une protéine.

1.2.2.4 Protéine

La synthèse des protéines se fait en deux étapes. Premièrement, il y a la transcription qui consiste à fabriquer une molécule d'ARNm à partir d'un modèle d'ADN. Puis, c'est la traduction, l'étape où la cellule fabrique la protéine à partir du message contenu dans l'ARNm. Examinons tout d'abord la transcription (Figure 1.20).

Le code génétique est la conformité, au niveau de l'ARN messager, entre les triplets de nucléotides et les acides aminés. C'est sa lecture, lors de la traduction, qui assure la production des protéines à partir de l'information génétique. Ce code permet le passage du gène à la protéine (Figure 1.21).

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

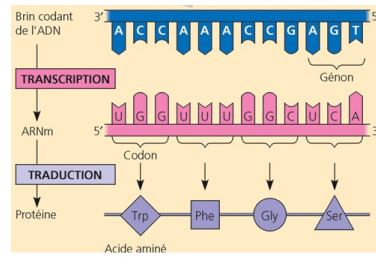


FIGURE 1.21 – Production d'une protéine à partir d'un brin d'ARN.

Dans lequel un acide aminé correspond à une succession de 3 nucléotides : TRIPLET ou CODON. Ce code est répétitif car plusieurs triplets peuvent avoir la même signification, c'est-à-dire coder pour le même acide aminé. Il est universel car un même triplet correspond à un même acide aminé, qu'il soit un vivant (l'homme, l'animal, le végétal ou la bactérie). Les codons « Stop » ou « Non Sens » sont les codons qui ne correspondent à aucun acide aminé.

		Deuxième lettre												
		U			C			A			G			
U	UUU	Phénil-	UCU	UUAU	tyrosine	UGU	cystéine	U						
	UUC	alanine	UCC	UAC	stop	UGC	tryptophane	C						
	UUA	leucine	UCA	UAA	codons	UGA	codon stop	A						
	UUG	leucine	UCG	UAG	stop	UGG	tryptophane	G						
C	CUU	leucine	CCU	CAU	histidine	CGU	arginine	U						
	CUC	leucine	CCC	CAC	histidine	CGC	arginine	C						
	CUA	leucine	CCA	CAA	glutamine	CGA	arginine	A						
	CUG	leucine	CCG	CAG	glutamine	CGG	arginine	G						
A	AUU	isoleucine	ACU	AAU	asparagine	AGU	sérine	U						
	AUC	isoleucine	ACC	AAC	asparagine	AGC	sérine	A						
	AUA	isoleucine	ACA	AAA	lysine	AGA	arginine	C						
	AUG	méthionine	ACG	AAG	lysine	AGG	arginine	G						
G	GUU	valine	GGU	GAU	acide	GGU	glycine	U						
	GUC	valine	GCC	GAC	aspartique	GGC	glycine	C						
	GUA	valine	GCA	GAA	acide	GGA	glycine	A						
	GUG	valine	GCG	GAG	glutamique	GGG	glycine	G						

FIGURE 1.22 – Le code génétique présentant les codons et les acides aminés correspondants.

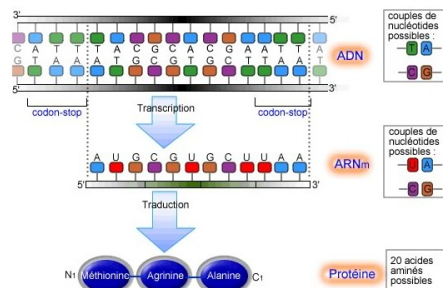


FIGURE 1.23 – Processus général de la synthèse des protéines.

1.2.3 Mutation d'ADN

1.2.3.1 Définition

La mutation est un terme désignant un changement brusque de la séquence d'ADN en nucléotides d'un gène, dû soit à un ajout ou à une perte d'un ou plusieurs nucléotides d'une séquence d'ADN, soit à une substitution d'un nucléotide par un autre. Ce changement de l'information génétique peut provoquer, lors de la traduction de l'ADN, que la synthèse d'une protéine ne fonctionne pas correctement. Ces modifications génétiques peuvent être dues à la fois à des facteurs internes et externes :

- Les facteurs internes provoquent des mutations dites spontanées se produisant au moment de la réplication cellulaire. En général, elles sont rares, et donc elles constituent la principale source de diversité génétique, moteur de l'évolution. Les causes des modifications spontanées sont inconnues.
- Les facteurs externes possibles sont des composés physiques ou chimiques, ils peuvent accroître considérablement le taux de mutations dans certaines circonstances. Ils sont appelés agents mutagènes, tels que les ondes électromagnétiques (rayons gamma, rayons X, les rayons ultraviolets), l'alcool, la cigarette ou encore les substances chimiques comme les pesticides, les dérivés de benzène, la colchicine, les solvants (ils assurent une altération du nombre de chromosomes).

1.2.3.2 Les types de mutations

Une mutation peut être de forte amplitude (quelque fois visible au niveau du chromosome) ou très ponctuelle[Zhang 2003]. C'est ce dernier cas qui est envisagé ici.

Les substitutions de paires de nucléotides :

La substitution, c'est à dire remplacement d'un nucléotide par un autre. Elles

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

sont de deux types :

- Les transitions : purine substituée par une purine (A \leftrightarrow G) ou pyrimidine par une pyrimidine (T \leftrightarrow C).
- Les transversions : pyrimidine substituée par une purine ou l'inverse (A \leftrightarrow C).

Elles peuvent être silencieuses c'est à dire n'engendrent pas de modification d'acide aminé dans la protéine (souvent pour les changements touchant la troisième bases du triplet). Elles peuvent aussi provoquer la substitution d'un acide aminé par un autre (exemple : CTC \rightarrow Glu ; CAC \rightarrow Val dans le cas de la drépanocytose) ou produire un codon "stop" écourtant prématurément la protéine : c'est une mutation efficace. Des modifications peuvent agir dans le promoteur ou la zone régulatrice du gène ou encore dans un intron provoquant la transcription et la traduction de l'ARNm (cas de nombreuses hémoglobinopathies).

Les changements du cadre de lecture :

Ce sont les insertions ou les délétions.

Exemple : la séquence d'ARNm : AUG CAG AUA AAC GCU GCA UAA.

Nous obtenons la protéine suivante : met gln ile asn ala ala stop.

Une délétion du A initial produit : UGC AGA UAA ACG CUG CAU ...

Nous obtenons la protéine suivante : cys arg stop.

Les protéines produites sont donc écourtées et souvent non fonctionnelles.

1.2.4 Réplication ou duplication d'ADN

Les conserves de l'information génétique dans une cellule et sa transmission sont établies par l'aptitude qu'a une cellule de produire deux molécules d'ADN ayant la même séquence, à partir d'une seule. Ce phénomène biochimique, appelé réplication, est basé sur la propriété de complémentarité des bases G-C et A-T. Il fait agir un brin d'ADN chromosomique, des

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

bases T, C, A et G libres, et plusieurs enzymes catalysant cette réaction, les ADN polymérase[Masai 2010]. Les bases azotées des couples Adénine-Thymine et Cytosine-Guanine ne sont liées entres-elles que par des liaisons faibles, qui vont être brisées par une enzyme polymérase pour écarter la molécule d'ADN[Koren 2014]. Cette enzyme va alors fixer des nucléotides libres, disponibles dans la cellule, sur les bases complémentaires de la séquence ainsi ouverte. Deux nouvelles molécules d'ADN vont ainsi être produites formées chacune d'un brin de l'ancienne molécule et d'un brin nouvellement produit. Nous disons que la réplication se fait suivant un mode semi-conservatif (Figure1.23).

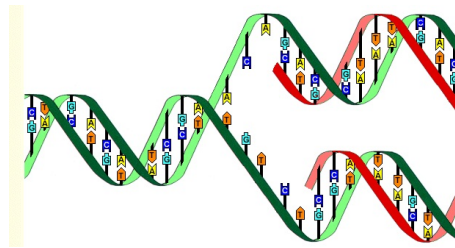


FIGURE 1.24 – Duplication d'une molécule d'ADN.

La duplication permet, de construire, à partir d'une cellule " mère", deux cellules filles contenant les mêmes chromosomes (dans lesquels un brin construit à partir de la cellule mère, et l'autre est nouvellement synthétisé), par ailleurs similaires à ceux de la cellule mère. C'est ce mécanisme, qui illustre comment l'information génétique est conservée dans toutes les cellules de l'espèce, lesquelles vont permettre la transmission de cette information à la descendance (c'est l'hérédité). La duplication est, de la même manière, à l'origine de la permanence des caractéristiques globales de chaque espèce (Figure1.24).

Remarque

Une mutation spontanée est produite d'un processus naturel, par contre les mutations induites résultent d'une interaction entre la séquence d'ADN et un agent extérieur ou mutagène. Cependant, dans les deux cas, la plupart des

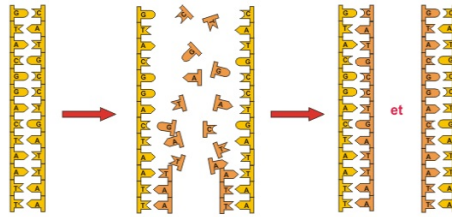


FIGURE 1.25 – La croissance du nouveau brin d'ADN.

mécanismes sont similaires. Toutefois les modifications qui résultent d'erreurs de duplication sont vraiment spontanées. Une erreur de duplication avec mise en place d'un nucléotide incorrect provoque une mutation lors de la duplication suivante. La fréquence des erreurs de l'enzyme polymérase agit sur la fréquence des modifications spontanées[Mazouzi 2014]. En effet, pendant la duplication on peut avoir une mutation c'est-à-dire une insertion ou une délétion ou une substitution de paires de nucléotides et par la suite il y a un changement brusque de l'information génétique.

1.2.5 L'alphabet Biologie

Un alphabet $X = a_0, a_1, \dots, a_n$ est un ensemble fini de symboles distincts deux à deux. En particulier, le symbole a_0 est appelé brèche ou gap (en anglais) et il est représenté par le caractère -.

1.2.5.1 Définition (Alphabet de l'ADN)

L'alphabet des molécules d'ADN est composé de 5 symboles. $X_{ADN} = \{-, A, C, G, T\}$ qui représentent respectivement un gap, l'Adénine, la Cytosine, la Guanine et la Thymine.

1.2.5.2 Définition (Alphabet de l'ARN)

L'alphabet des molécules d'ARN est composé de 5 symboles. $X_{ARN} = \{-, A, C, G, U\}$ qui représentent respectivement un gap, l'Adénine, la Cytosine, la Guanine et l'Uracile.

sine, la Guanine et l'Uracile.

1.2.5.3 Définition (Alphabet des Protéines)

L'alphabet des protéines est composé de 21 symboles :

$X_{AA} = \{-, A, C, D, E, F, G, H, I, K, L, M, N, P, Q, S, T, V, W, Y\}$ qui constituent les différents acides aminés.

1.3 Définition et but de classification

La notion de classification des séquences d'ADN se rapporte à la comparaison de deux séquences d'ADN(ou plus). Cette notion est particulièrement importante dans de nombreux domaines biologiques. En effet, elle permet par exemple de détecter les différences ou les similarités entre les génomes, elle permet aussi l'analyse génomique et par la suite la détection des maladies. Cette détection peut être couplée à une notion de score qui permet de mesurer une distance entre les génomes et ainsi d'estimer une date de divergence entre deux espèces(ou plus). Une grande partie des travaux effectués par les généticiens et biologistes se base sur des comparaisons de séquences génétiques et utilise ainsi l'alignement de séquences. Le but de la comparaison de séquences d'ADN à classifier est de découvrir des similitudes « biologiques » (structurelles ou fonctionnelles) parmi les séquences. Des séquences biologiquement similaires peuvent ne pas montrer une forte similitude de séquences d'ADN et le nous aimerons reconnaître la ressemblance structurelle/ fonctionnelle, même lorsque les séquences d'ADN sont très différentes. Si la similitude est faible, l'alignement par paires peut ne pas identifier des séquences d'ADN apparentées biologiquement, car de faibles similitudes au niveau des paires peuvent faire échouer les tests statistiques. L'alignement simultané de plusieurs séquences d'ADN permet souvent de montrer des similitudes invisibles dans la comparaison de séquences par paires[Ruffalo 2011]. Trouver des diffé-

rences (distance d'édition) entre des séquences équivaut souvent à trouver des similitudes entre celles-ci. Par exemple, si les opérations d'édition se limitent aux insertions et aux suppressions (pas de substitution), le problème de la distance d'édition est équivalent au problème du plus Long Sous-mot Commun (LSC). Les mathématiciens ont commencé à s'intéresser au problème LSC bien avant la découverte de l'algorithme de programmation dynamique pour la comparaison de séquences. Bien que la plupart des aspects algorithmiques de la comparaison de séquences soient captés par le problème LSC, les biologistes préfèrent utiliser les alignements pour la classification de séquences d'ADN et de protéines.

1.3.1 Homologie et similitude des gènes

Le paradigme central de la bioinformatique est : « la déduction par homologie ». Terminologie :

Identité : proportion des paires de bases (résidus) identiques entre deux séquences exprimée généralement en pourcentage.

Similitude : mesure de la similarité entre deux séquences d'ADN. Le degré de similitude est quantifié par un pourcentage de substitutions conservatives des séquences d'ADN.

Homologie : deux séquences d'ADN sont homologues si elles ont un ancêtre commun. Il n'y a pas de degré d'homologie. Nous ne disons pas : très homologues ou faiblement homologue. Deux gènes sont homologues ou ils ne le sont pas. Toutes les opérations de mutations de gènes, permettent :

- La spéciation : c'est la séparation d'une espèce en deux, chaque population évolue et forme une nouvelle espèce. Cette modification est le fruit d'une insertion, délétion, substitution ou mutation au niveau d'un gène.
- Les nouveaux organismes (espèces) héritent les mêmes gènes, mais mo-

difiés.

- La divergence : leurs gènes accumulent des mutations et génèrent d'autres espèces.

1.3.2 Principes de base de l'alignement de séquences à classifier

Pour faire une classification il faut aligner (comparer) les séquences entre elles. L'une des utilisations de la bioinformatique consiste à aligner les séquences d'ADN (les d'acides nucléiques)[Li 2010]. Les buts de l'alignement des séquences d'ADN sont :

- Identifier au sein d'une banque une séquence d'ADN obtenue en laboratoire de biologie.
- Localiser une séquence d'acide nucléique au sein du génome d'une espèce.
- Reconnaître et identifier un rôle à une molécule séquencée par comparaison avec des molécules de fonctions identiques déjà répertoriées.
- Construire une étude phylogénétique.
- Prédire la structure secondaire (tertiaire) d'une protéine.

Principe de l'alignement (comparaison)

- **Comparer des séquences d'ADN** : Rechercher le maximum d'appariements entre les résidus des séquences d'ADN comparées. La comparaison est d'autant plus parfaite qu'il n'y a pas de mésappariements et de brèches.
- **Mesure du degré de similitude** : La plupart des approches d'alignement de séquences d'ADN, et en particulier les méthodes d'alignement de séquence d'ADN cherchent à optimiser un score de comparaison. Ce score est relié au taux de ressemblance entre les deux séquences d'ADN

alignés.

$$score = \sum (SE) - \sum (SP). \quad (1.1)$$

avec SE désignant les scores élémentaires et SP indiquant les scores des pénalités.

1.4 Le problème de classification des données

La classification est un problème fréquemment rencontré pour la prise de décision d'activité humaine. Ce problème se produit quand un objet a besoin d'être assigné dans un groupe prédéfini ou bien à une classe basée sur un nombre d'attributs observés en rapport avec cet objet. Plusieurs problèmes dans les affaires, les sciences, les industries, et la médecine peuvent être traités comme des problèmes de classification, par exemples la prédiction de la faillite, le crédit scoring, les analyses et le diagnostic médical, le contrôle de la qualité, la reconnaissance du caractère et la reconnaissance de la parole. Dans le cas du diagnostic médical, il s'agira par exemple de décider, en fonction de données notamment physiologiques, si un individu présente un risque d'accident cardiaque ou non [Breiman 1984].

Autrement, la classification est un problème couramment rencontré dans des domaines allant de l'analyse de données à l'intelligence artificielle. Il consiste à trouver des groupements et des appartenances à ces groupements parmi les éléments d'un jeu de données. Les outils permettant de modéliser ce problème sont appelés des "classifieurs", il en existe différents types comme nous les verrons par la suite. Un classifieur s'utilise en deux étapes : il y a d'abord la phase d'apprentissage où nous entraînons la machine à reconnaître les éléments puis vient la phase de reconnaissance où ce qui a été appris est utilisé avec de nouvelles données.

1.4.1 Définition

La classification peut être définie comme étant un processus qui permet de séparer un ensemble d'objets en plusieurs sous-ensembles (classes) sur la base de leur ressemblance [Campedel 2008]. Le but est de définir des groupes en minimisant généralement la distance entre les membres dans la même classe et en maximisant la distance entre les ensembles de groupes, c'est-à-dire trouver des groupes qui ont des membres qui sont semblables l'un à l'autre, mais distant à des membres d'autres groupes. Le problème de classification est traité par un algorithme appelé classificateur. Il faut donc à priori définir le mode de représentation de l'objet à classer ou à reconnaître. En effet, la première étape de classification consiste à échantillonner puis à numériser le signal d'entrée. Ensuite, il y a extraction d'un certain nombre de caractéristiques, qui permet de faciliter le rôle du classifieur. Ces caractéristiques jouent un rôle très important dans la complexité du classifieur (Figure 1.25).

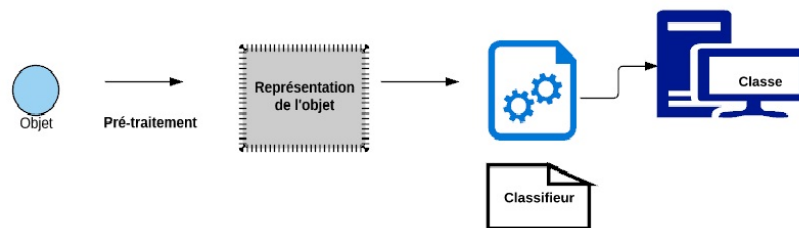


FIGURE 1.26 – Processus général d'une classification

1.4.2 Types de classification

Il existe deux types de classification : supervisée et non supervisée.

1.4.2.1 Classification supervisée

Une classification supervisée est un regroupement dont les classes sont connues à l'avance donc le but de classifieur est d'essayer de prédire ou d'af-

fecter tout nouvel individu à une classe parmi les classes disponibles. De manière plus formelle, nous considérons un ensemble E (ensemble des individus) muni d'une partition à priori $C = C_1, C_2, \dots, C_m$ en m classes. Chaque individu X est caractérisé par un ensemble de p variables descriptives ou prédicteurs $X = x_1, x_2, \dots, x_p$ chaque prédicteur peut être une variable qualitative ou quantitative.

Parmi les méthodes supervisées nous trouvons : les k -plus proches voisins, les réseaux de neurones, les arbres de décision, les machines à support de vecteurs(SVM), les classificateurs de Bayes, le markov caché

La classification supervisée confronte plusieurs problèmes par exemples le manque de données pour réaliser l'apprentissage. De même, des données imprécises empêchent d'élaborer une construction d'un modèle correcte [Denoeux 1995].

L'arbre de décision est une méthode rapide et compréhensible comme modèle, lorsque nous voulons prédire un nouvel exemple ; c'est-à-dire identification de la classe convenable pour cet exemple. Cette rapidité s'explique par le type de parcours suivi par ce classifieur. En effet, il utilise le parcours d'un chemin d'arbre pour classifier le nouvel exemple. Le problème majeur de cette technique c'est le sur-apprentissage lorsque cet arbre de décision est trop profond. Le réseau de neurone est performant en pratique. Il est un approximateur universel pour les fonctions, mais cette technique reste incompréhensible comme modèle ; ce réseau est considéré comme étant une boîte noire. De même le nombre de paramètre à optimiser est très important(le nombre de neurones dans la couche cachée, le choix des fonctions calculées par les neurones) et aussi les problèmes de sur-apprentissage et la convergence.

k -plus proches voisins est une méthode qui n'exige pas un modèle à construire. Il est simple à mettre en œuvre. Il est efficace pour les groupes réparties d'une façon irrégulière. Cette technique exige de nombreuses données de référence de même le volume de calcul ainsi que l'espace mémoire sont souvent prohibitifs [Bhatia 2010], [Waghodekar 2015],[Abbasifard 2014].

Le support vector machines(SVM) est une technique performante en pratique, mais elle exige un calcul important pour effectuer la phase d'apprentissage [Biswas 2015],[P.K.Srimani 2013].

Le Markov caché est une méthode rapide pendant la phase d'apprentissage et la phase de reconnaissance. Elle est stable comme étant un modèle, mais ce classifieur exige un nombre important de paramètres et d'hypothèses à définir [Satish 1993],[AlKhateeb 2011].

1.4.2.2 Classification non supervisée

Une classification non supervisée est un regroupement dont les classes ne sont pas connues à l'avance. Les classes se construisent au fur et à mesure durant la procédure de classification.

De manière plus formelle, nous disposons d'un ensemble des objets $X = x_1, x_2, \dots, x_n$ caractérisé par un ensemble de descripteurs D , l'objectif de regroupement est d'essayer de trouver les classes auxquelles appartient chaque objet x que nous notons par $C = c_1, c_2, \dots, c_n$.

La classification non-supervisée rencontre plusieurs problèmes : la séparation des classes n'est pas toujours franche et reconnaissable. Ces regroupements dépendent des paramètres initiaux comme le nombre de classes ce qui peut poser des problèmes.

Parmi les méthodes non supervisées nous trouvons : les k-means (centres mobiles), la classification hiérarchique, les cartes auto organisatrices,...

La classification non supervisée exige des différents choix qui sont laissés à l'initiative de l'utilisateur par exemple le nombre de groupe, la mesure d'éloignement (distance, dissemblance ou dissimilarité) entre les objets et aussi le critère d'homogénéité des groupes à optimiser.

L'algorithme k-means c'est un algorithme appliqué sur les grands jeux de données en raison de sa rapidité [Kanungo 2002]. Pendant l'application de cet

algorithme nous supposons qu'il existe k groupes différents. Au début, il y a désignation de k centres de classes (c_1, c_2, \dots, c_k) parmi les individus. Ces centres peuvent être soit choisis aléatoirement soit choisis par l'utilisateur. Ensuite, il y a réalisation itérative des deux étapes suivantes :

Étape 1 : pour chaque objet qui n'est pas un centre de groupe, nous cherchons quel est le centre de groupe le plus proche. Nous construisons ainsi k groupes g_1, \dots, g_k où g_i =ensemble des points les plus proches du centre c_i .

Étape 2 : Dans chaque nouveau groupe g_i , nous définissons le nouveau centre de groupe c_i comme étant le barycentre des points de g_i . L'algorithme s'arrête suivant un critère d'arrêt fixé par l'utilisateur ; le nombre limite d'itérations est atteint, soit l'algorithme a convergé, c'est-à-dire entre deux itérations les groupes construits restent les mêmes, soit l'algorithme à « presque » convergé, c'est-à-dire que l'inertie intra-groupe ne s'améliore quasiment plus entre deux itérations.

Les points initialement choisis comme centres de groupes (classes) permettent d'obtenir des partitions qui peuvent être très distinctes. Cette instabilité dans la composition des groupes en fonctions de l'initialisation est l'inconvénient majeur de la méthode qui ne permet donc pas de d'obtenir la partition optimale, mais converge plutôt vers une partition localement optimale (minima locaux). Pour résoudre ce problème nous pouvons faire tourner la méthode plusieurs fois avec différentes initialisations, et choisir la meilleure des partitions obtenues au sens de l'inertie intra-groupe (classe).

Algorithme K-means

Données : $D = x_1, x_2, \dots, x_n$

K =Nombre de classes

Paramètre appris = c_1, c_2, \dots, c_k = centres des classes

1. Initialiser les c en choisissant $c_i = x_j$ au hasard dans D
2. Calculer $s_i = \operatorname{argmin}_j \|x_i - c_j\|^2$.

- Pour chaque objet x_i qui n'est pas un centre de groupe, nous cherchons quel est le centre de groupe le plus proche. Nous construisons ainsi k groupes g_1, \dots, g_k où g_i = ensemble des points les plus proches du centre c_i .
- Dans chaque nouveau groupe g_i , nous définissons le nouveau centre de groupe c_i comme étant le barycentre des points de g_i .

3. Itérer jusqu'à ce que l'erreur de reconstruction ne baisse plus.

La classification ascendante hiérarchique (CAH) est un algorithme qui permet de construire une suite de partitions emboîtées des données en n groupes (classes), $n-1$ groupe, ..., 1 groupe [F. Sousa 2005].

Cet algorithme se fait selon les étapes suivantes :

Etape 1 : Les n objets sont considérés dans une seule classe.

Etape 2 : Nous calculons les distances deux à deux entre objets, et les deux objets les plus proches sont réunis en un groupe (classe).

Etape 3 : La distance entre ce nouveau groupe et les $n-2$ objets restants est ensuite calculée, et à nouveau les deux éléments (classes ou objet) les plus proches sont réunis.

Ce processus est réitéré jusqu'à ce qu'il ne reste plus qu'une seule classe constituée de tous les objets.

Les cartes auto organisatrices qui sont appelées aussi carte auto-organisatrices (Self Organizing Maps : SOM), sont une classe de réseau de neurones. Dans ce type de réseau, les neurones sont liés entre eux selon le concept de voisinage et non selon la notion de couche. Chacun des neurones j est connecté à des neurones d'entrée i , chaque connexion possédant un poids particulier w_{ij} [Lampinen 1992],[Vesanto 2000].

Ce réseau est composé par une couche d'entrée et une couche de sortie. Tous les objets à classer sont représentés par un vecteur multidimensionnel et à chaque objet est affecté un neurone qui représente le centre de la classe. Dans

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

la couche de sortie les neurones de cette couche entrent en compétitions et les meilleurs neurones qui vont être gagnés.

L'apprentissage de ce réseau consiste à adapter, de manière itérative, les poids des connexions afin de spécialiser les neurones en fonction des types de signaux présentés en entrée du réseau. Nous définissons autour de chaque neurone un voisinage qui évoluera dans le temps en se rétrécissant. Après avoir initialiser les connexions aléatoirement, nous calculons la distance qui s'étend entre l'entrée et chaque neurone de sortie, puis nous choisissons le neurone de distance minimale. Il ne reste alors qu'à modifier les poids de connexions des neurones de son voisinage. Ce modèle de réseau a donné des résultats impressionnants dans le domaine de la reconnaissance de parole. L'analyse de cette méthode permet de représenter les données en conservant la topologie et aussi les données proches ont des représentations proches dans l'espace de sortie. Ces données vont être regroupés dans un même groupe ou dans des groupes voisins.

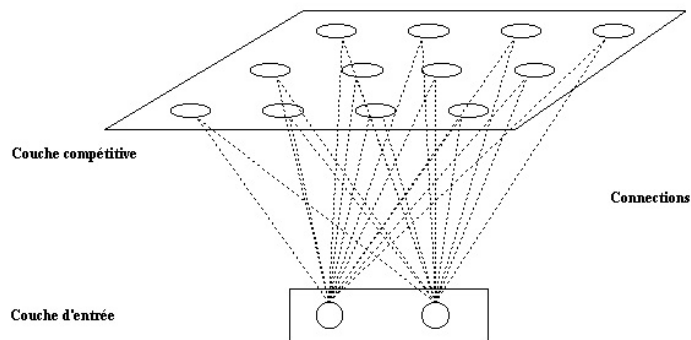


FIGURE 1.27 – Les couches et les connexions dans la carte auto-organisatrice

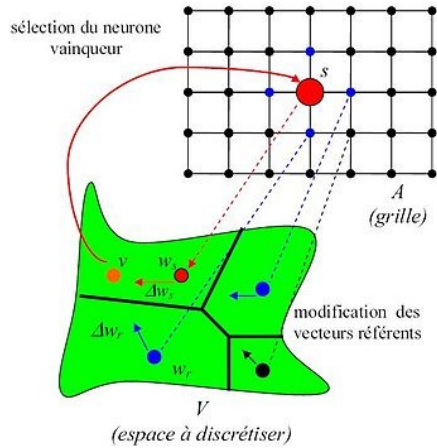


FIGURE 1.28 – Sélection du neurone vainqueur dans la carte auto-organisatrice

1.5 La classification des séquences d'ADN

1.5.1 Description des problèmes de classification des séquences d'ADN

La notion de classification des séquences d'ADN se rapporte à la comparaison de deux séquences d'ADN ou plus. Cette notion est particulièrement importante dans de nombreux domaines biologiques. En effet, elle permet par exemple de détecter les différences ou les similarités entre les génomes. Elle permet aussi l'analyse génomique et par la suite la détection des maladies et des corps inconnues. Cette détection peut être couplée à une notion de score qui permet de mesurer une distance entre les génomes et ainsi d'estimer une date de divergence entre deux espèces ou plus. Une grande partie des travaux effectués par les généticiens et biologistes se base sur des comparaisons de séquences génétiques et utilise ainsi l'alignement de séquences [Aniba 2010], [Kato 2010], [Marucci 2014].

Le but de la comparaison de séquences d'ADN à classifier est de découvrir des similitudes « biologiques » (structurelles ou fonctionnelles) parmi les séquences. Des séquences d'ADN biologiquement identiques peuvent ne pas ex-

hiber une forte similitude de séquences et le nous aimerons découvrir la ressemblance fonctionnelle ou structurelle, même lorsque les séquences d'ADN sont très différentes. Si la similitude de ces séquences est faible, la comparaison par paires peut ne pas identifier des séquences apparentées biologiquement, car de faibles identités (similitudes) au niveau des paires peuvent provoquer l'échec des tests statistiques. L'alignement simultané de plusieurs séquences d'ADN permet souvent de trouver des similitudes invisibles dans l'alignement de séquences par paires[Gavarraju 2016], [Daugelaite 2013].

1.5.2 Motivations de classification des séquences d'ADN

Les biologistes traitent l'alignement de séquences d'ADN en utilisant le traitement automatique pour reconnaître dans quelle mesure deux séquences d'ADN se ressemblent. En effet, la classification des séquences d'ADN permet de déduire des connaissances sur une séquence à partir des connaissances attachées à une autre. Ainsi, si dans une classe les séquences d'ADN sont très similaires, et si l'une est connue pour être codante, l'hypothèse que les autres le soient aussi peut être avancée. De même, si des séquences protéiques sont identiques, il est souvent fait l'hypothèse que les protéines correspondantes assurent des fonctions similaires ; si la fonction d'une séquence est connue, la fonction des autres peut ainsi s'en déduire. Ce principe d'inférence se justifie par des considérations sur le processus d'évolution.

Cette classification nous permet de faire une analyse génétique pour discerner une éventuelle maladie génétique ou pour examiner un risque, de maladie par exemple, pour les diabètes, plusieurs cancers et l'épilepsie,

De même, elle permet de faire une analyse génétique pour établir l'identification des cadavres (guerres ou catastrophes naturelles), l'identification de l'auteur d'un acte criminel (en médecine légale), le test de paternité,etc.

Elle permet d'inférer la fonction de gènes inconnus par similitude avec le pro-

fil d'expression de gènes connus[Volfovsky 2001]. L'alignement d'ADN permet d'identifier des sous-groupes de maladies pour mieux comprendre les mécanismes de régulation, en identifiant des coexpressions, et de distinguer les individus sains de ceux atteints par une maladie. Dans le domaine biologique il existe plusieurs bases de données qui contiennent l'ensemble des séquences nucléiques publiques avec leurs indications (par exemple GenBank), ou l'ensemble des séquences protéiques expertisées (SwissProt). Le premier réflexe d'un biologiste qui dispose d'une séquence nouvelle d'ADN est de parcourir ces bases, afin d'y trouver les séquences similaires et de faire succéder (hériter) à la nouvelle séquence les connaissances qui leur sont associées. De même l'alignement des séquences d'ADN d'espèces actuelles permet de reconstruire et de tracer des arbres phylogénétiques qui expliquent l'histoire évolutive des organismes.

La création manuelle des bibliothèques des familles des séquences d'ADN, pour les génomes sont largement traités, tels que la souris et l'humain. Enfin, toutes les méthodes existantes pour l'identification des familles des séquences d'ADN se basent sur la comparaison locale (recherche de similarités, paires par paires), qui exige un temps de calcul et un espace de stockage exponentiels en fonction de la taille des données.

1.5.3 Problématique de classification des séquences d'ADN

Etant données un échantillon composé de n séquences s_1, s_2, \dots, s_n , trouver la meilleure classification pour ces séquences.

La classification consiste à trouver des groupements et des appartenances à ces groupements parmi les éléments d'un jeu de données. La classification des séquences d'ADN est un problème fréquemment étudié et traité par les biologistes pour interpréter et analyser automatiquement les connaissances obtenues. Elle est un problème NP-complet. En effet, l'alignement est au-delà

de deux séquences, le problème devient rapidement très complexe car l'espace des alignements (comparaisons) possibles explose. (Rappel, pour la programmation dynamique, la complexité croît comme le produit des longueurs des séquences)[Eddy 2002].

Les avancées récentes dans les technologies de séquençage amènent aujourd'hui à disposer d'un nombre conséquent de séquences d'ADN. Nous pouvons se trouver confronté à analyser quelques millions de séquences (comme c'est le cas pour les données de méta-génome) et une première étape pour cette analyse est de déterminer s'il existe une structure des données en groupes homogènes selon un critère à déterminer. C'est aussi l'objectif en génomique comparative où nous cherchons à comparer des séquences, par exemple identifier parmi ces séquences celles qui se ressemblent. L'abondance de ces séquences n'en permet pas une analyse simple et nécessite donc le recours à des méthodes automatiques.

1.5.4 Objectifs principaux

Notre travail consiste à modéliser une nouvelle approche de classification inspirée des réseaux d'ondelettes beta pour classifier et analyser les séquences d'ADN. La performance de cette approche devra être mesurée par plusieurs critères d'évaluation par exemple : précision, Rappel, Matrice de confusion, de même les résultats devront être validés par des experts en biologie pour construire le système le plus fiable possible. L'idée étant de comparer les brins des séquences d'ADN. Le but sera donc d'appliquer une méthode développée de classification non supervisée et de voir si les groupes obtenus sont cohérents, c'est-à-dire nous étudierons aussi les groupes de gènes obtenus pour voir s'ils partagent des fonctions biologiques ou s'ils sont impliqués dans les mêmes voies métaboliques. Cette collaboration aura pour but de valider la méthode qui sera développée au cours de cette thèse, mais aussi de propo-

ser de nouvelles hypothèses de travail aux biologistes. Donc nous cherchons à classer des séquences se comportant de la même manière, en optimisant les deux critères suivants : homogénéité (pour les séquences regroupées) et séparation (entre les différents regroupements).

L'objectif général de notre travail est de développer une approche automatique de complexité polynomiale, qui permet de classer un échantillon de n séquences d'ADN en utilisant une approche conçue à partir des réseaux d'ondelettes Bêta. Cette approche doit être indifférente à l'ordre des séquences en entrée. Elle permet de générer des résultats ayant les propriétés de l'interopérabilité. Elle doit découvrir des classes avec des formes variables et biologiquement significatives selon l'échantillon de séquences en entrées, c'est-à-dire une classification qui minimise la variabilité intra-groupe tout en maximisant les distances intra-groupes. Plus précisément, nous visons à trouver l'ensemble des groupes (séquences d'ADN) dont les éléments sont très semblables, mais éloignés des autres éléments sur la base de leurs nucléotides durant la phase d'apprentissage. Cette approche va nous permettre de créer des groupes de séquences de même fonction biologique ou de même structure.

Cette méthode doit avoir la capacité à gérer différents types de variables (attributs) et des grandes bases de données, de même elle doit être capable de traiter des contraintes incorporées par l'utilisateur.

Ce travail va mener à une étude de validation de l'approche proposée. La validation des regroupements est une étape importante. Cette étude permet d'évaluer la qualité, la stabilité et la précision des groupes. Pour cela, nous devons mettre au point une importante étude de simulations à des données décrivant la structuration du génome des espèces ou à des données de méta-génomique. D'une part, pour étudier les performances de cette méthode en classification et d'autre part pour effectuer une étude de comparaisons de cette méthode avec les méthodes de classification de séquences d'ADN qui ont été précédemment proposées.

Pour évaluer les performances des ces types de réseau nous devons réaliser plusieurs essais comparatifs et les résultats vont être exprimés en taux de bonne classification avec des configurations de réseaux d'ondelettes ayant des paramètres bien optimisés. Nous devons essayer aussi d'appliquer une méthode pour valider la classification des séquences d'ADN.

1.5.5 Etat de l'art de classification des séquences d'ADN

Les algorithmes de classification sont définis comme des méthodes de répartition d'un ensemble d'objets (points ou vecteurs) en plusieurs sous-ensembles, sur la base de leurs similarités ou dissimilarités. Le but est de construire des groupes qui minimisent la variabilité intra-groupe tout en maximisant les distances inter-groupes. Plus précisément, ils visent à trouver l'ensemble des groupes (gènes ou échantillons) dont les membres sont très similaires, mais distants des autres membres sur la base de leur profil d'expression.

Il existe plusieurs méthodes pour mesurer les similarités et les dissimilarités entre les gènes à regrouper par exemple la distance euclidienne, la distance de Jaccard qui mesure la dissimilarité entre les ensembles [Gilbert 2000],[Rani 2012]. Les algorithmes de classification se regroupent en deux grandes catégories : les approches supervisées et non supervisées. Les méthodes non supervisées regroupent les objets sans à priori. Ces techniques sont dites exploratoires et sont essentiellement employées pour la découverte de classes. A l'inverse, les méthodes supervisées utilisent la connaissance à priori [Cavalieri 2005]. Elles établissent des règles et un modèle de classification à partir d'un jeu de données annotées, ou jeu d'apprentissage (training set), pour prédire ensuite la classification (Class prediction) de nouveaux cas appartenant à un jeu de données test.

Le problème de classification d'ADN est étudié par plusieurs méthodes. En effet, il existe trois méthodes de classification ;

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

La première catégorie de classification est basée sur la caractéristique (fonction), qui transforme une séquence en un vecteur de caractéristique (fonction) et applique ensuite les méthodes de classification conventionnelle. La sélection de caractéristique (fonction) joue un rôle important dans cette sorte de méthodes ; c'est à dire identifier un sous-ensemble de séquences caractéristiques pour cette classification.

La deuxième catégorie de classification est basée sur la distance de séquence. La fonction de distance mesure la similitude entre les séquences et détermine une qualité significative de classification. La troisième catégorie est basée sur les modèles, comme l'utilisation du modèle de Markov caché (HMM) et d'autres modèles statistiques pour classer les séquences d'ADN.

Cependant, il existe d'autres approches qui permettent de classer les séquences d'ADN. Ces approches combinent certaines méthodes et techniques utilisées par les catégories de classification qui sont déjà citées précédemment.

En effet, Vrinda et al. ont appliqué les réseaux de neurones artificiels (RNA) en utilisant la représentation et le principe du jeu de Chaos pour classer les séquences d'ADN des individus inconnus. Cette classification permet de classer les organismes des êtres vivants en utilisant les séquences d'ADN et préciser les caractéristiques d'évolution entre les êtres vivants en présentant la relation mutuelle entre les organismes et plusieurs autres aspects dans l'étude d'êtres vivants [Nair 2010]. L'approche appliquée présente une nouvelle méthode de classification des organismes basée sur la combinaison entre le principe du jeu de Chaos (PJC) et l'utilisation des réseaux de neurones artificiels comme classificateur.

L'apprentissage et le test de RNA sont appliqués sur une base de données qui contient plusieurs séquences d'ADN des organismes d'eucaryotes (organismes vivants possédant un noyau par exemples les plantes, les animaux, les champignons,...).

Vrinda et al. ont étudié plusieurs types des réseaux de neurones artificiels

(RNA) qui ont donné des résultats performants, avec un pourcentage de 92,3% de degré de classification. Dans cette méthode de classification les séquences d'ADN sont représentées par le principe du jeu de Chaos en utilisant la fréquence des nucléotides dans chaque séquence d'ADN.

Agnieszka et al. ont utilisé les ondelettes et les cartes auto-organisatrices comme classificateur pour regrouper les séquences d'ADN. Ils ont combiné les deux techniques pour modéliser un classificateur [Jach 2010]. Au début, ils ont utilisé les ondelettes pour extraire la variation des nucléotides dans la séquence d'ADN dans le but d'élaborer le vecteur de caractéristique pour chaque brin de taille différente, ensuite ils ont développé un classificateur en utilisant la technique des cartes auto-organisatrices. Enfin, ils ont constaté que les ondelettes sont performantes pour des séquences de petites tailles. Le taux de classification maximal obtenu est de 73.13% et la complexité de ce réseau dépend du nombre de neurones dans la couche cachée et de la longueur des séquences d'ADN à classer. Dans cette approche l'algorithme de back propagation est utilisé pour faire l'apprentissage du réseau.

Gamal. F. Elhadi et al, ont appliqué les réseaux de neurones artificiels pour classer les séquences d'ADN [Elhadi 2012]. Au début, ils ont analysé biologiquement les brins d'ADN. Ensuite, ils ont développé un classificateur inspiré des réseaux de neurones pour regrouper ces séquences en plusieurs classes d'une façon performante. Les séquences d'ADN classifiées sont de différentes tailles. Enfin, ils ont classé ces séquences sous la forme d'une représentation hiérarchique. Ce classificateur représente la similarité entre les séquences d'une façon claire et significative surtout pour les séquences de petites tailles, contrairement aux brins de grande taille. Ils ont montré que cette méthode est performante et efficace pour présenter les similarités entre les couples d'ADN. Donc le résultat de l'approche est efficace pour les séquences d'ADN de tailles réduites contrairement pour les données de grandes tailles, de même la représentation des classes par le dendrogramme reste flou pour les séquences

de grandes tailles. Nous constatons donc que la complexité de cette approche dépend de la taille des données à classifier.

Amiya Kumar Patel et al, ont proposé un classificateur inspiré des réseaux des neurones artificiels qui utilise la propagation comme un algorithme d'apprentissage [Patel 2009]. Ils ont prouvé que ce classificateur permet de regrouper les séquences d'ADN d'une façon un peu performante ; il a une prédiction de 72.99%, un pourcentage de prédiction correcte qui vaut 73.952%, une sensibilité égale à 81.53% et une spécificité de 72.54%. Ces critères sont utilisés pour mesurer la performance de ce classificateur.

Cependant, David L. et al, ont appliqué une méthode de compression basée sur l'énumération pour classifier les séquences d'ADN[Loewenstern 1995]. Cette méthode utilise la technique de compression du texte pour résoudre le problème de classification d'ADN. Au début, cette méthode organise les séquences d'ADN en sous-séquences c'est-à-dire sous-chaîne pour chaque séquence. Ensuite, ces sous-chaînes seront codifiées. Enfin, il y a application d'un algorithme appelé algorithme Zdiff pour classifier les séquences d'ADN. Ils ont montré que cette méthode est performante surtout pour les brins d'ADN de petites tailles.

M. Mohamed. et al., ont utilisé la technique des cartes auto-organisatrices et le principe du réseau de neurones pour classifier les séquences d'ADN. Ils ont appliqué cette approche sur des bases de données se trouvant dans le site web NCBI (The National Center for Biotechnology Information)[Mohamed 2013]. Les résultats élaborés montrent bien que cette technique est performante en la comparant avec les autres classificateurs et dépend de la structure et des représentations des données (séquences d'ADN).

Les cartes auto-organisatrices ont été appliquées aussi par Cheng-Chang Jeng et al. Ils ont utilisé ces cartes comme étant un classificateur pour regrouper les séquences d'ADN[Jeng 2006]. Ils ont utilisé le principe de Power Spectrum pour former les vecteurs de caractéristiques pour chaque séquence d'ADN.

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

L'application du Power Spectrum comme technique pour représenter les séquences d'ADN montre bien qu'il est puissant et efficace pour distinguer les distributions des nucléotides dans chaque séquence d'ADN des bactéries, qui ont des tailles importantes. Les cartes auto-organisatrices (SOM) sont utilisées pour enquêter sur les différences et le regroupement des bactéries. Ce classificateur a donné un résultat fiable et efficace, mais ce résultat ne permet pas de donner des interprétations et des analyses biologiques qui sont nécessaires pour la prise de décision.

Shakir Mohamed et al, ont appliqué deux théories pour classifier les brins d'ADN[Mohamed 2006],[Mohamed 2007]. Ils ont utilisé la théorie de la Résonance Adaptative(ART) [Carpenter 1992], et les cartes auto-organisatrices pour classifier les séquences d'ADN. Au début, les séquences d'ADN sont transformées en vecteurs numériques en utilisant la technique de reconnaissance. Ensuite, ils ont combiné l'ART, les cartes et l'algorithme génétique comme étant un algorithme d'apprentissage pour classifier les séquences d'ADN. Enfin, ils ont estimé que ce système est capable de classifier les séquences d'ADN avec une exactitude de 93%. Cette exactitude est comparée avec d'autres méthodes et démontre que l'approche proposée est convenable vue sa haute exactitude et sa rapidité d'apprentissage.

L'ARTMAP(Résonance Adaptative(ART), Self organizing MAP) flou basé sur la théorie de la résonance adaptative a été introduit par G. A. Carpenter et al. [Carpenter 1992]. Cette approche est comparée avec cinq autres classifieurs(le Modèle Linéaire Généralisé, le k-Nearest Classifieur Neighbour, le Multi-Layer Perceptron, le Réseau Neural et la Fonction de la Base Radiale Réseau Neural). Cette comparaison montre bien que ce classificateur est performant pour classifier des séquences d'ADN de tailles différentes, mais l'analyse et l'interprétation biologique reste toujours difficile à atteindre.

H.-M. Muller et al. ont utilisé la technique d'analyse en composante principale (ACP) pour regrouper les séquences d'ADN [Muller 2003]. Au début, les

séquences sont traduites dans des vecteurs du document qui représentent leur contenu du mot ; Ensuite, l'ACP définit les classes de la séquence, donc la classification utilise le contenu du mot et sa variation d'usage pour distinguer les séquences. Enfin, ils ont testé leur approche avec plusieurs données de génome d'ADN. Ce test a montré que ce classifieur est capable de classer les introns et les exons avec une exactitude allant jusqu'à 96%.

Cathy Wu et al., ont appliqué les réseaux des neurones pour essayer de résoudre le problème de classification d'ADN [Wu 1993]. A l'aide de cette approche, ils ont classifié des grandes bases de données moléculaires d'une façon rapide et efficace. Les réseaux neuraux utilisés sont des réseaux à trois couches et l'algorithme de propagation comme étant un outil d'apprentissage du réseau modéliser. Les séquences moléculaires sont codées sous forme des vecteurs d'entrée pour le réseau des neurones. Ce codage a été fait par l'utilisation de la méthode de hachage ou par la méthode de décomposition des valeurs singulières (DVS). Au début, l'apprentissage du réseau a été fait à l'aide des séquences d'ADN qui sont connues. Ensuite, le système neural devient une mémoire associative capable de classer les séquences inconnues basées sur l'information de classe enfoncée dans les interconnexions neurales. Le taux de classification obtenu à l'aide de cette approche est égale à 82% pour la classification d'un échantillon contenant des séquences d'ARN et 100% pour une base de données contenant des séquences d'ADN. Le système a été utilisé pour réduire le temps de recherche dans la base de données des séquences moléculaires.

D'autres travaux ont prouvé que la méthode SVM génère de bons résultats concernant le problème de classification des séquences d'ADN [Seo 2010]. Le principe de base de l'application de la méthode SVM sur les données des séquences d'ADN consiste à dresser une carte d'une séquence dans un espace des caractéristiques et de trouver l'hyperplan de marge maximale qui sépare deux classes. Soient deux séquences (x,y) , plusieurs fonctions kernel $k(x,y)$

doivent être utilisées pour présenter la similarité entre ces deux séquences d'ADN. Les défis d'appliquer SVM pour la classification incluent comment définir les espaces du trait (vecteur caractéristique) ou la fonction noyau, et comment accélérer le calcul de matrices des fonctions kernel ou noyau. De même la méthode SVM ne donne pas des résultats escomptés pour classier des données de très petites tailles par exemple. Donc la performance de cette approche exige les meilleurs choix du paramètre du noyau et aussi le choix de la fonction noyau.

Il existe une autre méthode de classification d'ADN basée sur la température de fusion des séquences d'ADN. Cette méthode est appliquée sur des séquences de plusieurs espèces mammifères. La comparaison de profil de température de fusion avec les arbres phylogénétiques moléculaires construits à l'aide des séquences, montre que l'approche basée sur la température de fusion est capable de reproduire la plupart des principales caractéristiques de l'arbre évolutif basé sur les séquences. Le principe de profil de température de fusion prend en considération la structure inhérente et la dynamique de molécule d'ADN. Cette méthode ne nécessite pas d'alignement de séquences avant la construction des arbres, et fournit un moyen qui permet de vérifier les résultats expérimentalement. Par conséquent, les résultats montrent que la classification basée sur la température de fusion des séquences d'ADN pourrait être un outil utile pour l'analyse des séquences d'ADN [Reese 2010]. L'avantage majeur de cette méthode est qu'elle fournit une façon de vérifier les phylogénétiques des séquences ADN et le processus évolutif moléculaire expérimentalement.

Kevin Crosby and al., ont utilisé l'algorithme SPRINT pour résoudre le problème de classification de types d'ADN (Intron et exon). Cette méthode permet d'aider les biologistes dans les laboratoires à regrouper les introns dans une classe et les exons dans une autre, d'une façon très rapide et performante. Cette technique permet d'utiliser l'arbre de classification en se basant sur

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

l'algorithme SPRINT [Crosby 2004]. Les génomes de l'elegans ont été utilisés pour l'apprentissage et le test. Un grand taux d'erreur de l'échantillon de l'épreuve de 15% a été montré pour le melanogaster Drosophile, alors que l'elegans Caenorhabditis était seulement 1.6% .

Jing Zhao et al, ont utilisé l'ondelette paquet pour classifier deux types d'ADN se trouvant dans 20 échantillons d'ADN artificiel. A l'aide de cette méthode ils sont arrivés à regrouper des séquences d'ADN naturel [Zhao 2001].

L'ondelette paquet est utilisée pour analyser et extraire les caractéristiques des séquences ADN. Pour étudier la séquence ADN avec la décomposition du paquet d'ondelette, ils ont codé chaque séquence par une représentation numérique. L'analyse du paquet d'ondelette est une généralisation de décomposition d'ondelette qui offre une gamme plus riche de possibilités pour analyser le signal. Dans cette analyse, ils ont utilisé la fonction d'échelle pour les approximations et la fonction d'ondelette pour les détails. Ils ont utilisé l'ondelette de type Daubechies3 pour décomposer et analyser les séquences d'ADN.

Selon les approches existantes, nous pouvons classifier les séquences d'ADN par trois méthodes ;

La première méthode est appelée méthode supervisée. Elle nécessite à priori des connaissances et des références pour les séquences d'ADN. Ces données vont être utilisées durant la phase d'apprentissage du modèle à construire et aussi pour faire la comparaison entre les séquences d'ADN. La deuxième méthode est nommée approche non supervisée c'est le contexte de notre travail. Cette méthode effectue une classification fondée entièrement sur les caractéristiques intrinsèques aux données du test. Durant la phase d'apprentissage, elle essaie de construire des modèles ou des classes pour les séquences d'ADN se trouvant dans la base d'apprentissage.

La troisième approche est appelée méthode semi-supervisée. Elle partage les caractéristiques des deux méthodes précédemment indiquées (supervisée et

non supervisée).

D'après ces méthodes d'apprentissages, il existe d'autres approches de classification des séquences d'ADN :

- **PhyloPythia** : Cette méthode permet de regrouper des fragments d'ADN qui ont des tailles réduites et elle utilise les informations pertinentes se trouvant dans les phylogénétiques pour classifier ces fragments [McHardy 2007]. Durant la phase d'apprentissage, une base de données test est utilisée pour mesurer la performance de ce système. Elle constitue 340 séquences d'ADN (bactéries). La méthode est appliquée pour regrouper des fragments de tailles variées. Les résultats montrent bien que ce système n'est performant que pour la classification des séquences de taille réduites. Le taux de classification de la prédiction d'une séquence d'ADN est influencé par la différence qui existe entre la taille de la séquence à prédire et la taille de séquences qui sont utilisées pour construire le model SVM. Le taux est optimal lorsque les deux tailles sont presque égales. La performance est diminuée lorsque la taille des séquences d'ADN d'apprentissage est plus importante que la taille de séquence d'ADN à prédire.
- **Naïve Bayesian** : Cette approche utilise la statistique bayésienne pour classifier les séquences d'ADN [Sandberg 2001],[Rosen 2008]. Elle utilise les concepts suivants : les fréquences oligonucleotides (courts segments d'acides nucléiques) et le théorème de bayes afin de classifier les séquences d'ADN. Cette méthode est robuste durant la classification des fragments d'ADN. Les résultats ont montré que la précision de la classification croit avec l'augmentation de taille des sequences.
- **TACOA** : Le classificateur TACOA exploite l'algorithme k voisin le plus proche (k-NN) afin de regrouper les fragments d'ADN en fonction de leurs profils de fréquences d'oligonucléotides sous-jacents. La taille d'un fragment d'ADN de requête influence directement la précision de

cette approche[Diaz 2009].

- **TETRA** :TETRA est une méthode de classification non supervisée[Teeling 2004]. Elle est appliquée pour regrouper des séquences d'ADN inconnues. Le regroupement se fait à l'aide de profils de fréquences tétranucléotides. Cette approche est incapable de classifier des fragments d'ADN dans les arbres phylogénétiques existants. Elle a la capacité de calculer le degré de parenté entre deux paires de fragments d'un ensemble de séquences d'ADN donné. Cette performance exige que les deux paires appartiennent au même génome donc TETRA reste incapable de calculer le bon degré de parenté entre deux paires de génome différents. Les résultats obtenus par cette approche montrent sa performance limitée pour la classification des séquences d'ADN qui appartiennent au même génome et aussi nous constatons que la taille des fragments d'ADN a une influence sur la précision et le taux de classification.
- **SOM** : SOM est une architecture d'un réseau de neurones artificiels. Elle est considérée comme étant une machine d'apprentissage qui permet d'approximer des données dont les tailles sont variées. Cette méthode utilise les compositions des séquences d'ADN pour faire la classification. Elle utilise le principe de fréquence tétranucléotides pour calculer la similarité entre les séquences afin de les classifier[Abe 2006]. La performance est très importante pour la classification des séquences de grandes tailles contrairement pour les fragments de tailles réduites ; 74.6% des séquences de grandes tailles sont bien classées par contre 40.6% de fragments d'ADN de tailles réduites sont regroupés. Le résultat montre que la taille des séquences a une influence sur la performance de cette méthode comme l'approche PhyloPythia qui a de bons résultats pour les séquences de tailles réduites.
- Etc.

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

Tableau 1.1 – Les autres méthodes de classification des séquences d'ADN

Classifieur	Catégorie d'apprentissage	Stratégie de classification	Méthodologie	Références
Phylo-Pythia	Supervisée	Composition	SVM +Les profils de fréquences tétranucléotides.	[McHardy 2007]
Naïve Bayesian	Supervisée	Composition	Bayésien+Les profils de fréquences n-mer (sous chaîne de taille n)	[Sandberg 2001]
TACOA	Supervisée	Composition	K plus proches voisins+ Les profils de fréquences oligonucléotides	[Diaz 2009]
Chi-squared (FAMeS)	Supervisée	Composition	Méthode Chi-squared+ Les profils de fréquences de nucléotides)	[Mavrommatis 2007]
TETRA	Non Supervisée	Composition	Z-scores+Les fréquences tétranucléotides	[Teeling 2004]
SOM	Non Supervisée	Composition	SOM+ Les fréquences tétranucléotides	[Abe 2006]
S-GSOM	Semi-Supervisée	Composition	GSOM+flanquants des séquences	[Chan 2008]
Compost-Bin	Semi-Supervisée	Composition	phylogénétiques+ Les fréquences hexanucléotides	[Chatterji 2008]
BLAS distr(FAMeS)	Supervisée	Homologue	Prédiction + taxonomique	[Chatterji 2008]
CARMA	Supervisée	Homologue	Les taxonomiques	[Krause 2008]

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

Les méthodes de classification des séquences d'ADN utilisent deux stratégies de classification. La première utilise le principe d'alignement pour comparer les séquences d'ADN. Ce principe consiste à comparer les séquences d'ADN deux à deux. L'application de Cette méthode prouve que la classification des séquences d'ADN est un problème NP-complet. En effet, pour l'alignement au-delà de deux séquences, le problème devient rapidement très complexe car l'espace des alignements possibles explose de même la complexité croît comme le produit des longueurs des séquences. Donc l'alignement nécessite un temps de calcul et un espace de stockage exponentiels en fonction de la taille des données. L'évaluation de l'alignement se fait à l'aide d'une métrique qui s'appelle score d'alignement qui permet de mesurer le degré de ressemblance entre les séquences d'ADN à classifier.

Lorsque deux séquences d'ADN ont une grande similarité au niveau de leur structure primaire (un pourcentage élevé de correspondance dans la composition des lettres), il y a une forte probabilité que ces deux séquences d'ADN aient la même fonction. Un simple alignement sert pour regrouper les séquences d'ADN. Toutefois deux séquences peuvent être très différentes au niveau de la structure primaire, tout en aient une structure tertiaire similaire. Dans ce cas un alignement des séquences ne peut aider à les classer. L'alignement de séquences comporte donc des lacunes pour certaines classifications.

La deuxième utilise le principe des fréquences k-nucléotidiques (Oligonucléotides, Dinucléotide, Tétranucléotides,). Cette méthode explore des motifs souvent récurrents dans les séquences nucléotidiques pour faire la classification. Il est difficile à identifier la meilleure taille des n-grammes(k-nucléotidiques). Pour résoudre ce problème, il faut effectuer plusieurs expérimentations.

Pour ces cas, une alternative existe : la classification de séquences biologiques par apprentissage machine. Les outils d'apprentissage machine sont intéressants car ils peuvent cibler des motifs cachés ou bruités qui échappent aux algorithmes d'alignement par similarité.

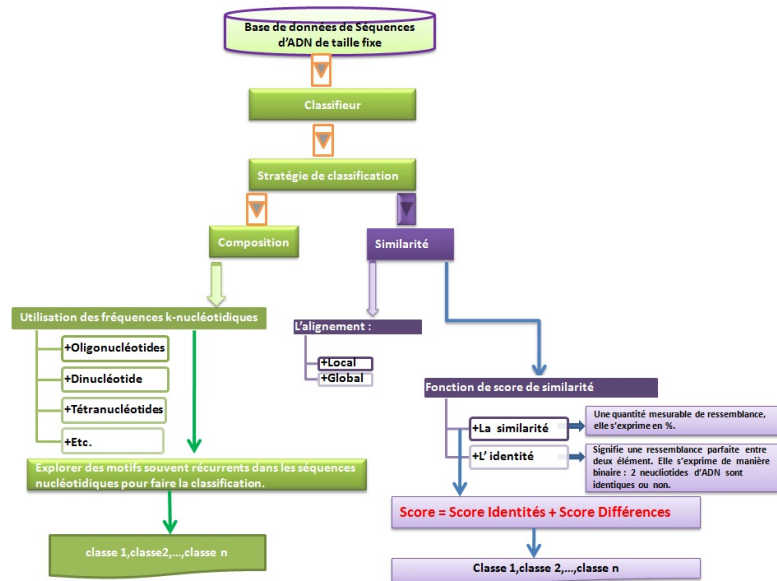


FIGURE 1.29 – Les stratégies de classification utilisées.

1.5.6 Les limites des méthodes existantes

La variété des méthodes de classification expose des résultats prometteurs qui peuvent montrer leurs capacités dans la classification des séquences d'ADN. La performance de ces méthodes est souvent fortement dépendante de plusieurs facteurs qui ne sont pas faciles à contrôler, telles que la longueur des fragments d'ADN dans l'échantillon, la complexité de la communauté métagénomique donnée, les similarités de composition entre les membres de la communauté et l'existence de séquences d'ADN étroitement apparentées dans les différentes bases de données de référence. Par exemple les approches fondées sur la composition des nucléotides au niveau d'une séquence (PhyloPythia, Tacoa) et les approches basées sur l'homologie (BLAST distr, CARMA) fonctionnent mieux lorsqu'elles sont appliquées à des fragments de longueur modérée à partir d'une métagénome de faible complexité ; c'est-à-dire une communauté qui contient un nombre limité de séquences d'ADN, pour laquelle les membres prédominants ont des parents proches dans les bases de

données de séquences ou dans l' ensemble de données d'apprentissage.

Toutes ces méthodes souffrent d'une diminution drastique des performances lors de la tentative de classer les séquences les plus courtes, des fragments à partir d'une communauté métagénomiques complexes, ou des séquences pour lesquelles un parent proche n'est pas disponible à la fin de comparaison. Les méthodes non supervisées telles que les différentes approches de la méthode SOM ne dépendent pas explicitement des bases de données de référence des séquences connues, mais elles n'ont tendance à réussir que pour classer des fragments d'ADN plus longs. Toutes les méthodes existantes pour la classification d'ADN montrent une tendance à la baisse de précision de la classification en proportion à un niveau croissant de spécificité du rang taxonomique à laquelle les séquences sont comparées. Donc ces limites peuvent provoquer une perturbation dans l'analyse et l'interprétation biologique des résultats et par suite une difficulté pour la prise de décision.

1.6 Conclusion

Ce chapitre nous a permis de présenter l'état de l'art concernant la classification des séquences d'ADN. Nous avons exposé les motivations à la fois applicatives et théoriques qui ont poussé la communauté scientifique à se pencher sur ce problème. Au debut, nous avons présenté la définition de la bio-informatique comme étant une discipline qui vise le traitement automatique de l'information biologique. Nous avons défini la séquence d'ADN d'une façon détaillée et nous avons précisé les problèmes fondamentaux qui peuvent provoquer et altérer les séquences d'ADN comme étant un support de l'information génétique pour chaque organisme. Ensuite, nous avons cité la définition et le but de classification des séquences d'ADN ; nous avons défini les concepts d'homologie et de similitude des gènes et nous avons montré aussi les principes de base de l'alignement de séquences des ADN à classer.

Chapitre 1 : Etat de l'art sur la classification des séquences d'ADN

Egalement, nous avons présenté le problème de classification des données comme étant un problème fréquemment rencontré pour la prise de décision d'activité humaine. Aussi nous avons cité que les différents types de classification (supervisée et non supervisée) et dressé quelques exemples des méthodes de classification des séquences d'ADN comme étant le contexte de notre thèse. Dans cette partie nous avons décrit les problèmes et les motivations de classification des brins d'ADN. De même, nous avons présenté l'état de l'art de classification des séquences d'ADN. Nous avons dressé les différents méthodes et algorithmes qui sont utilisés pour classifier les brins d'ADN et les principaux résultats de classification.

Ce tour d'horizon de différents problèmes de classification vise à montrer dans quels cas les problèmes sont bien résolus en termes de taille et de nombre des instances traitées et dans quels cas ils ne le sont pas.

Dans le chapitre suivant nous essayons de présenter la théorie et la construction des réseaux de neurones et des réseaux d'ondelettes.

CHAPITRE 2

Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction



Sommaire

2.1	Introduction	53
2.2	Les Réseaux de neurones	54
2.2.1	Définition	54
2.2.2	Historique	54
2.2.3	Application	56
2.2.4	Fondement biologique des neurones	57
2.2.5	Principe de fonctionnement des neurones	58
2.2.6	Les réseaux de neurones artificiels	59
2.2.7	Exemples d'architecture des réseaux de neurones	60
2.3	Les réseaux d'ondelettes	63
2.3.1	Définition	63
2.3.2	Les ondelettes	64
2.3.3	Architecture des réseaux d'ondelettes	75
2.3.4	De la transformée inverse aux réseaux d'ondelettes	77
2.3.5	Comparaison des réseaux d'ondelettes aux réseaux de neurones	78
2.4	Apprentissage des réseaux d'ondelettes	78
2.4.1	L'apprentissage supervisé	79
2.4.2	L'apprentissage non supervisé	80
2.4.3	Apprentissage d'un réseau d'ondelettes par l'analyse multirésolution	81
2.4.4	Apprentissage spécifique aux réseaux d'ondelettes	84
2.5	Téchniques de construction des réseaux d'ondelettes	85
2.5.1	Les méthodes incrémentales utilisées pour la construction des réseaux d'ondelettes	85
2.5.2	Construction de réseaux d'ondelettes par l'algorithme pyramidal	92

**Chapitre 2 : Des Réseaux de neurones vers les Réseaux
d'ondelettes : Théorie et construction**

2.5.3	Construction de réseaux d'ondelettes par des techniques basées sur la transformée discrète	92
2.5.4	Construction d'un réseau d'ondelettes basée sur l'ana- lyse fréquentielle	93
2.5.5	Construction d'un réseau d'ondelettes en utilisant la théorie des ondelettes orthogonales	94
2.5.6	Construction d'un réseau d'ondelettes pour un système adaptatif	94
2.5.7	Construction d'un réseau d'ondelettes basée sur la construc- tion des frames	95
2.5.8	Calcul direct des pondérations (poids) de connexion	99
2.6	Les méthodes de sélection d'ondelettes	100
2.6.1	La technique de choix par orthogonalisation	100
2.6.2	La méthode des moindres carrées orthogonales (OLS)	102
2.7	Conclusion	107

2.1 Introduction

Ce chapitre présente la partie théorique des réseaux de neurones et les réseaux d'ondelette et leurs applications. Nous aborderons les principales architectures de réseaux de neurones citées dans la littérature. Il ne s'agit pas de les traiter toutes, car elles sont trop nombreuses, mais plutôt d'en comprendre les principaux mécanismes internes et de savoir comment et quand les utiliser et les appliquer. En ce sens, nous mettrons autant l'emphase sur l'analyse mathématique de ces réseaux que sur la façon de les utiliser dans la pratique pour résoudre des problèmes concrets. Enfin, nous présenterons la relation qui existe entre les deux types de réseaux et les différentes architectures des réseaux d'ondelettes.

2.2 Les Réseaux de neurones

2.2.1 Définition

Les réseaux de neurones, sont composés par des structures cellulaires artificielles, constituant une approche permettant de résoudre plusieurs problèmes concernant la perception, le mémoire, l'apprentissage et le raisonnement. Ils s'avèrent aussi des choix très prometteurs pour dépasser certaines limitations des ordinateurs classiques. Grâce à leur manipulation parallèle de l'information et à leurs mécanismes inspirés des cellules nerveuses (neurones), ils infèrent des propriétés émergentes permettant de résoudre des problèmes jadis qualifiés de complexes. Un neurone est une cellule cérébrale dont la fonction fondamentale consiste à collecter, traiter et transmettre des signaux électriques. Nous pensons que la capacité du cerveau à traiter les informations marque essentiellement la mise en réseaux de neurones. C'est pourquoi, quelques premières recherches en intelligence artificielle (IA) ont eu pour objectif d'élaborer des réseaux de neurones artificiels[Russell 2003].

2.2.2 Historique

John Anderson et Edward Rosenfeld ont marqué dans le livre intitulé « Neurocomputing : Foundations of Research », l'aspect historique des recherches concernant les réseaux de neurones[Anderson 1989]. L'histoire des réseaux de neurones est donc tissée à travers des recherches conceptuelles et des développements technologiques survenus à diverses époques. Les premières recherches concernant les réseaux de neurones s'est déclenchée à la fin du 19e et au début du 20e siècle. Ces recherches ont traité le neurone d'une façon générale. Elles consistent en des travaux multidisciplinaires en physique, en psychologie et en neurophysiologie par des scientifiques tels Ernst Mach, Ivan Pavlov et Hermann von Helmholtz [Pavlov 1927]. A cette époque, il s'agissait

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

de théories plutôt générales sans modèle mathématique précis d'un neurone. On s'entend pour dire que l'apparition du domaine des réseaux de neurones artificiels revient aux années 1940 avec les travaux de Warren McCulloch et Walter Pitts qui ont signalé qu'avec de tels réseaux [McCulloch 1943], nous pouvons, en principe, traiter et calculer n'importe quelle fonction arithmétique ou logique. Vers la fin des années 1940, Donald Hebb a ensuite proposé une théorie fondamentale pour l'apprentissage [Hebb 1949].

Le premier travail concret des réseaux de neurones artificiels est survenu vers la fin des années 1950 avec l'invention du réseau dit «perceptron» par un dénommé Frank Rosenblatt [Rosenblatt 1958],[Rosenblatt 1962]. Rosenblatt et ses collègues ont construit un réseau et démontré ses habilités à reconnaître des formes. Malheureusement, il a été démontré par la suite que ce perceptron simple ne pouvait résoudre qu'une classe limitée de problème. Environ au même moment, Bernard Widrow et Ted Hoff ont proposé un nouvel algorithme d'apprentissage pour entraîner un réseau adaptatif de neurones linéaires, dont la structure et les capacités sont similaires au perceptron.

Vers la fin des années 1960, un livre publié par Seymour Papert et Marvin Minsky est venu signaler beaucoup d'ombre sur le domaine des réseaux de neurones [Minsky 1969]. Ces deux auteurs ont prouvé les limites des réseaux élaborés par Rosenblatt et Widrow-Hoff. Beaucoup de gens ont été influencés par cette démonstration qu'ils ont généralement mal interprétée. Ils ont conclu à tort que le domaine des réseaux de neurones était un cul-de-sac et qu'il fallait cesser de s'y intéresser (et de financer la recherche dans ce domaine), d'autant plus qu'on ne disposait pas à l'époque d'ordinateurs suffisamment puissants pour effectuer des calculs complexes. Heureusement, certains chercheurs ont persévéré en développant de nouvelles architectures et de nouveaux algorithmes plus puissants. En 1972, James Anderson et Teuvo Kohonen ont élaboré indépendamment et simultanément de nouveaux réseaux pouvant servir de mémoires associatives [Anderson 1989]. Egalement, Stephen Grossberg

a investigué ce qu'on appelle les réseaux auto-organisés.

Dans les années 1980, une pierre d'achoppement a été levée par l'invention de l'algorithme de rétropropagation des erreurs. Cet algorithme est la réponse aux critiques de Minsky et Papert formulées à la fin des années 1960. Cette nouvelle élaboration, généralement attribuée à James McClelland et David Rumelhart, mais aussi découverte plus ou moins en même temps par Yann LeCun et par Paul Werbos, ont littéralement ressuscité le domaine des réseaux de neurones. Depuis ce temps, c'est un axe où bouillonne constamment de nouveaux travaux, de nouvelles structures et de nouvelles méthodes. Dans cette partie, nous allons tenter d'en survoler les fondamentaux.

2.2.3 Application

Les réseaux de neurones servent aujourd'hui à toutes sortes d'applications dans divers axes. Par exemple, à l'aide de cette méthode, il y a eu le développement d'un auto-pilote qui permet de conduire un avion d'une façon automatique, ou encore un système de guidage pour automobile, un système qui permet de contrôler les centres nucléaires, Il y a eu l'élaboration des applications de lecture automatique d'adresses postales et des chèques bancaires, des systèmes de traitement du signal pour différentes utilisations militaires, un système pour le traitement de la parole, des réseaux sont aussi appliqués pour bâtir des systèmes de vision par machine, pour faire des prévisions sur les domaines monétaires, pour calculer et évaluer le risque financier ou en assurance, pour différents processus manufacturiers, pour l'exploration gazière ou pétrolière, en télécommunication, en robotique, pour le diagnostic médical, Les réseaux de neurones ont aujourd'hui un impact important et, il y a fort à parier, que leur importance ira grandissant dans le futur.

2.2.4 Fondement biologique des neurones

Le neurone est une cellule formée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites (Figure 2.1). Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le corps du neurone (soma). L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. En fait, il existe un espace intercellulaire de quelques dizaines d'Angstroms (10^{-9} m) entre l'axone du neurone afférent et les dendrites du neurone efférent. La jonction entre deux neurones est appelée la synapse (Figure 2.2).

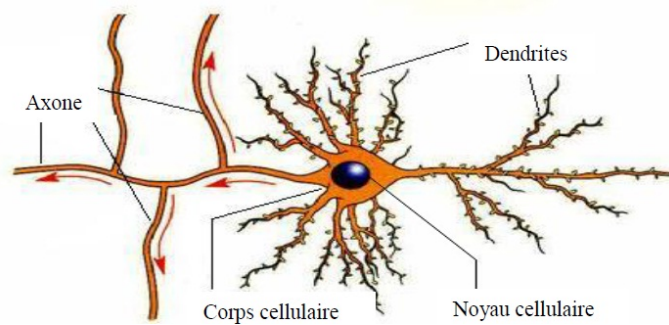


FIGURE 2.1 – Structure d'un neurone

Les neurones sont des cellules qui constituent l'élément de base du système nerveux (Figure 2.1). Un neurone est formé de :

- **Le corps cellulaire** : ce corps contient le noyau du neurone et traite les transformations biochimiques nécessaires à la synthèse des enzymes et d'autres molécules pour conserver la vie du neurone.
- **Les dendrites** : Ce sont des extensions tubulaires permettant de détecter les signaux arrivant au neurone, et les transférer vers son corps.

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

- **L'axone** : c'est une fibre nerveuse permettant le passage des signaux émis par un neurone vers d'autres. Il se distingue des dendrites par les propriétés de sa membrane externe et par sa structure.

Un réseau de neurones est formé de plusieurs neurones connectés par des synapses (Figure 2.2).

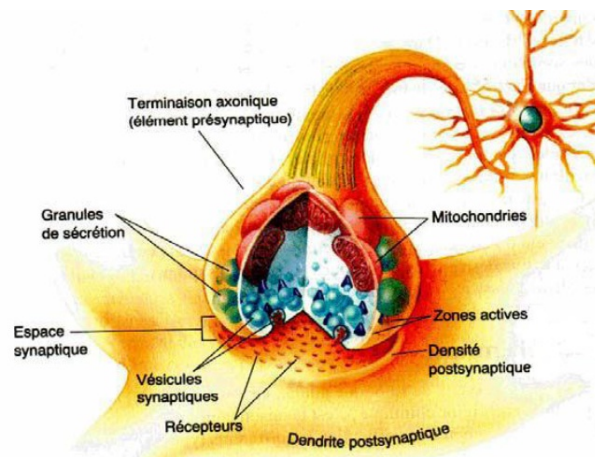


FIGURE 2.2 – La synapse d'un neurone

2.2.5 Principe de fonctionnement des neurones

Le fonctionnement d'un neurone dépend principalement des propriétés de sa membrane externe. Lorsque le neurone est stimulé, un potentiel électrique, désigné potentiel d'action, naît dans le corps cellulaire de neurones et s'étend tout au long de l'axone. Une fois arrivé à la frontière de l'axone, le potentiel d'action provoque la libération d'un médiateur chimique, nommé neurotransmetteur, au niveau de la synapse où le signal électrique de l'impulsion nerveuse est transformé en un signal biochimique. Le courant synaptique s'étend le long des dendrites jusqu'au corps cellulaire du neurone cible. A ce niveau, le corps cellulaire manipule l'ensemble des courants synaptiques qui lui parviennent en calculant une somme algébrique des courants synaptiques inhibiteurs et exci-

tateurs. Si le potentiel obtenu dépasse le seuil critique d'excitation du neurone (-10 mV), alors le neurone est stimulé et provoque à son tour un potentiel d'action qui s'étend le long de son axone. Dans le cas contraire, le neurone reste inactif.

2.2.6 Les réseaux de neurones artificiels

Un neurone artificiel fait une sommation pondérée des potentiels d'actions qui lui parviennent (chacun de ces potentiels est une valeur numérique qui montre l'état du neurone qui l'a émis), puis s'active selon la valeur de cette addition (sommation) pondérée. Si cette addition dépasse un certain seuil, le neurone est activé et propage une réponse dont la valeur est celle de son stimulation (activation), sinon le neurone reste inactif et ne propage rien.

Chaque neurone formel accepte un nombre variable d'entrées. A chacune de ces entrées est lié un poids w représentatif de la force de la connexion. Chaque neurone est accordé d'une sortie unique, qui permet d'alimenter un nombre variable de neurones aval. Le neurone traite la somme pondérée de ses entrées, puis il calcule sa sortie par une transformation non linéaire de cette addition. Les pondérations ou les poids représentent l'intensité synaptique de ce neurone.

Le fonctionnement d'un neurone formel est exprimé par les expressions suivantes :

$$E_j = \sum_{j=1}^n (x_j w_{ji}). \quad (2.1)$$

$$y_i = f(E_i - \theta). \quad (2.2)$$

- x_i : Signaux d'entrée du neurone artificiel i ;
- w_{ji} : Poids(coefficients) des entrées ;
- y_i : La sortie du neurone artificiel i ;

- E_i : Entrée globale ;
- θ : Seuil d'activation du neurone ;

2.2.7 Exemples d'architecture des réseaux de neurones

2.2.7.1 Le perceptron multicouche (PMC)

La forme la plus simple de réseau de neurones artificiels est appelé perceptron qui permet de grouper correctement des individus appartenant à deux groupes (classes) linéairement séparables. Il consiste en un seul neurone formel qui possède un seuil, ainsi qu'un vecteur de poids synaptiques ajustable. La mise en cascade de perceptrons produit ce que nous nommons les perceptrons multicouches (Figure 2.3). Lorsque le vecteur de caractéristiques d'un individu est présenté à l'entrée du réseau, il est communiqué à tous les neurones de la première couche. Les sorties des neurones artificiels de cette couche sont alors communiquées aux neurones de la couche suivante, et ainsi de suite. La couche de sortie c'est la dernière couche du réseau, les autres étant désignées sous le terme de couches dites cachées car les valeurs de sortie de leurs neurones artificiels ne sont pas accessibles de l'extérieur. La théorie d'approximation montre qu'un perceptron multicouche, à une seule couche cachée, est en théorie toujours suffisant. Toutefois, il ne prédit en aucun cas le nombre de couche cachées qui est nécessaire pour atteindre une qualité d'approximation suffisante et performante. Ce nombre pouvant parfois être gigantesque, l'application d'un perceptron multicouche à deux (ou plus) couches cachées ne consistant chacune qu'à un nombre limité de neurones artificiels, peut parfois s'avérer être plus utile.

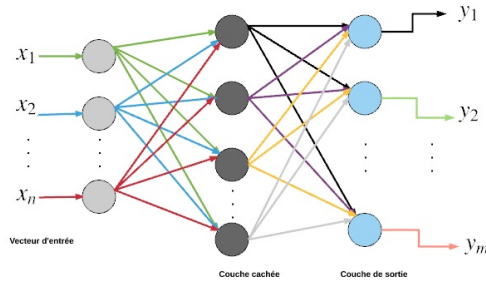


FIGURE 2.3 – Perceptron multicouche à une couche cachée

Le perceptron multicouche permet d'élaborer des fonctions discriminantes non linéaires grâce à l'utilisation des fonctions d'activation non linéaires. La démarche d'apprentissage supervisé du perceptron multicouche, distingué sous le nom d'algorithme de rétropropagation, exige toutefois que les fonctions d'activation des neurones formels soient continues et dérivables. Les fonctions qui sont le plus souvent appliquées sont probablement de type sigmoïdal (On-delettes par exemple). La donnée d'entrée est, à plusieurs reprises, élaborée au réseau de neurones formels en utilisant le principe de rétropropagation. Chaque élaboration, la sortie du réseau est confrontée à la sortie désirée et une erreur est calculée. Cette erreur est alors réinjectée dans le réseau de neurones formels et appliquée pour ajuster les poids de façon qu'elle diminue à chaque itération et que le réseau de neurones s'approche davantage de la reproduction de la sortie désirée. Ce processus s'appelle la formation.

2.2.7.2 Le réseau à Fonction Radiale de Base (RBF)

Le réseau à fonction radiale de base possède deux couches de neurones formels (Figure 2.4). Les neurones de sortie réalisent une combinaison linéaire de fonctions de base non linéaires, élaborées par les neurones de la couche cachée. Ces fonctions de base élaborent une réponse différente de zéro seulement lorsque l'entrée se localise dans une petite région bien située de l'espace des variables. Bien que plusieurs modèles de fonctions de base existent, le plus

fréquent est de type Gaussien.

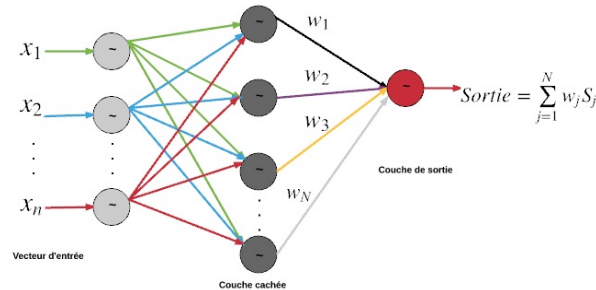


FIGURE 2.4 – Un réseau à fonction radiale de base

L'apprentissage du réseau des neurones formels à fonction radiale de base est souvent scindé en deux parties : Dans un premier temps, les poids des neurones formels de la couche cachée sont présentés par une quantification vectorielle. Il existe plusieurs méthodes de réaliser cette dernière. Lorsque les poids des cellules cachées sont figés, les paramètres de normalisation sont achevés en calculant la dispersion des données d'apprentissage liées à chaque centroïde. La seconde couche du réseau des neurones peut alors être entraînée. L'apprentissage est cette fois supervisé (les valeurs de sortie désirées sont fournies), et se réalise typiquement à l'aide d'un algorithme basé sur un critère des Moindres Carrés de l'Erreur.

L'entraînement de la seconde couche du réseau est très rapide, car, d'une part, les sorties de la couche cachée contenant des neurones, peuvent être évaluées une seule fois pour tous les exemples d'apprentissage, et d'autre part, les sorties des neurones formels de la seconde couche sont linéaires. Les algorithmes d'apprentissage, tel que la méthode des moindres carrés du perceptron, peuvent être utilisés. Le fait que l'apprentissage de la couche cachée comportant des neurones formels, soit non supervisé, est toutefois un inconvénient de ce modèle de réseau des neurones vis-à-vis d'un perceptron multicouche. Pour pallier à ce choix, des méthodes d'apprentissage supervisé

appliquées sur les réseaux à fonction radiale de base ont également été réalisées [Musavi 1992].

L'avantage du réseau des neurones à fonction radiale de base est que sa phase d'apprentissage est souvent plus rapide que celle du MPC (mise en cascade de perceptrons). Mais la non-linéarité, présente dans la couche de sortie du perceptron multicouche (mise en cascade de perceptrons), est inexistante dans le réseau des neurones formels à fonction radiale de base, ce qui provoque un désavantage de ce dernier vis-à-vis du premier. L'efficacité (taille du réseau / Erreur) d'un réseau des neurones à fonction radiale de base et d'un perceptron multicouche dépend du problème traité. La fonction de base la plus utilisée est la gaussienne. Elle s'exprime, sous sa forme la plus générale, par l'équation suivante :

$$\phi_j(x) = \exp\left(-\frac{\|x - c_j\|^2}{2\sigma_j^2}\right). \quad (2.3)$$

Où σ_j^2 et c_j désignent respectivement la variance et le centre associés à la cellule cachée.

La décroissance de la gaussienne est la même pour toutes les directions de l'espace. Un nombre restreint de fonction de base participe au calcul de la sortie pour une entrée donnée soit :

$$y(x) = \sum_{j=1}^k w_j \phi_j(x). \quad (2.4)$$

2.3 Les réseaux d'ondelettes

2.3.1 Définition

Les réseaux d'ondelettes (RO) est une combinaison de deux techniques d'analyse de signaux : La transformée en ondelettes et les réseaux de neurones artificiels. Les réseaux d'ondelettes appliquent les fonctions ondelettes

au lieu de la fonction sigmoïde traditionnelle comme sa fonction de transfert dans chaque neurone. Deux architectures différentes ont été proposées pour différentes applications ; la première a été proposée pour des buts généraux tels que la classification, la prédiction quantitative, et la reconnaissance de formes et la deuxième pour traiter la compression des signaux. Le RO a été présenté par Beneniste et Zhang en 1992 comme étant un réseau qui combine le réseau de neurone et la théorie d'ondelettes. Ce réseau est utilisé pour approximer les fonctions non linéaires[Zhang 1992]. Ce réseau a été étudié par plusieurs auteurs. Les premiers travaux ont été assurés par Krishnaprasad et Pati [Pati 1993]. Ils ont appliqué la superposition de fonctions sigmoïdes pour élaborer l'ondelette à fin de rapprocher la décomposition en ondelettes et assurer un développement sous forme de réseaux de neurones. L'algorithme de descente en gradient est appliqué pour calculer les coefficients entre la couche cachée et la couche de sortie pour minimiser la fonction d'erreur. La bibliothèque WaveNet a été utilisée par Bakshi et Stephanopoulos pour construire leur réseau. Cette bibliothèque constitue des familles d'ondelettes orthonormales [Bakshi 1993]. Ces familles sont élaborés en utilisant la théorie de l'analyse multirésolution et les résultats développés par Mallat [Mallat 1989].

2.3.2 Les ondelettes

2.3.2.1 Définition

L'ondelette c'est une fonction qui oscille durant un temps donné ou sur un intervalle spatiale de longueur finie. En dehors, la fonction décroît très rapidement vers zéro. Haar a élaboré les premières ondelettes dans les années 30. Ces ondelettes sont composées par une base des fonctions orthogonales. Ces fonctions sont appelées des ondelettes Haar qui ne sont pas dérivables. Dans les années 80, Meyer a élaboré de nouvelles fonctions ondelettes qui constituent une base de fonctions orthogonales. Ces ondelettes sont des fonctions

dérivables [Meyer 1985]. Ces fonctions sont utilisées dans l'analyse multirésolution de signaux par Mallat en 1989 [Mallat 1989]. Elles sont peu adaptées pour l'approximation de fonctions car elles ne peuvent pas s'exprimer sous forme analytique simple. Les ondelettes à structures obliques (frames) ont été développées par J.Morlet pour élaborer des bases de fonctions qui ne sont pas nécessairement orthogonales à fin de représenter des signaux. De même les frames sont développés par I.Daubechies en élaborant un support théorique aux résultats de J.Morlet. Ces structures sont définies par des expressions analytiques simples, et toute fonction de carré sommable peut être approchée, avec la précision voulue, par une addition finie des fonctions d'ondelettes issues d'une ondelette à structure oblique (frame) [Daubechies 1990].

2.3.2.2 L'analyse de Fourier

L'existence d'une technique pour transformer l'information initiale en une représentation claire est très importante. Cette technique joue un rôle très important pour faire apparaître clairement des informations qui ont été cachées dans la représentation initiale. Baron Jean Baptiste Joseph Fourier a élaboré une analyse qui s'appelle l'analyse de Fourier. Cette analyse permet de décomposer toutes les fonctions comme sommes de fonctions élémentaires par exemple somme de sinusoides. Il s'agit des fonctions périodiques, comme des fonctions sinus et cosinus.

Etant donnée une fonction $f(t)$,supposée périodique pour simplifier, c'est-à-dire tel que $f(t + T)=f(t)$, nous écrivons :

$$f(t) = \frac{1}{2}a_0 + a_1 \cos\left(\frac{2\Pi t}{T}\right) + b_1 \sin\left(\frac{2\Pi t}{T}\right) + a_2 \cos\left(\frac{4\Pi t}{T}\right) + b_2 \sin\left(\frac{4\Pi t}{T}\right).... \quad (2.5)$$

Cette expression constitue une infinité de paramètres. Les termes a_0, a_1, b_1, \dots représentent le poids de chacune des sinusoides dans la fonction $f(t)$, et sont nommés les coefficients de Fourier de $f(t)$.Ces coefficients se calculent par les expressions

suivantes :

$$a_k = \frac{1}{T} \int f(t) \cos\left(\frac{2k\Pi}{T}t\right) dt. \quad (2.6)$$

$$b_k = \frac{1}{T} \int f(t) \sin\left(\frac{2k\Pi}{T}t\right) dt. \quad (2.7)$$

L'intégral de Fourier (somme continue) sert à résoudre les fonctions non périodiques. Cette technique permet de représenter le signal par une superposition d'ondes sinusoïdales de toutes les fréquences possibles. Ces fréquences génèrent des amplitudes comme pour les séries de Fourier. Ces amplitudes élaborent alors une fonction de la fréquence nommée « spectre contenu des fréquences du signal » : c'est la transformée de Fourier du signal qui est évaluée à l'aide de l'intégrale de Fourier :

$$F(f) = \int f(t) \exp^{-2i\Pi ft} dt. \quad (2.8)$$

Le signal $f(t)$ est reconstruit par la transformée inverse calculée par l'expression suivante :

$$f(t) = \int_{-\infty}^{+\infty} F(f) \exp^{-2i\Pi ft} df. \quad (2.9)$$

En général, l'application de la transformée de Fourier exige que le signal doit être de carrée sommable c'est-à-dire d'énergie finie. Cette condition est souvent remplie car dans le cas des signaux réels la mesure est faite sur un temps fini. L'analyse de Fourier est très importante pour analyser les fonctions mais cette technique a plusieurs inconvénients, en particulier son manque évident de localisation temporelle. En effet, cette technique permet d'élaborer les différentes fréquences pour exciter dans un signal donné, c'est-à-dire son spectre, mais ne permet pas de connaître à quels instants ces fréquences ont été émises. Elle élabore une information globale et non locale, car les fonctions d'analyse

appliquées sont des sinusoides qui oscillent indéfiniment sans s'amortir. Cette perte de localité devient un problème pour l'étude de signaux non stationnaires. Et par la suite cette analyse ne permet pas l'étude de signaux dont la fréquence varie dans le temps. Pour résoudre ce problème il y a apparition des plusieurs travaux par exemple l'utilisation des ondelettes pour de composer et analyser des fonctions qui exigent les localisations temporelle.

2.3.2.3 L'analyse par ondelettes

L'analyse par ondelettes c'est l'application d'une famille de fonctions élaborée à partir d'une fonction ψ de $L^2(\mathbb{R})$, à valeurs éventuellement complexes, nommée ondelette mère, ou ondelette analysante :

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}}\psi\left(\frac{x-b}{a}\right). \quad (2.10)$$

Les variables a et b sont respectivement les paramètres de dilatation et de translation. Ces paramètres permettent d'influencer sur l'allure de l'ondelette analysante $\psi_{a,b}(x)$ (figure 2.5) qui dépend de ces deux paramètres suivant l'analyse du signal f . Les paramètres peuvent être appliqués de façon discrète ou continue. Donc nous pouvons grouper les transformées en ondelettes selon la famille de fonction d'ondelettes à laquelle appartiennent les fonctions analysantes sélectionnées. Les transformées obtenues sont suivant les cas discrète ou continues, redondantes ou non. La transformée discrète d'ondelette est utilisée dans la complémentarité des filtres, passe-haut et passe-bas pour extraire des informations qui caractérisent les transitions rapides du signal. Par contre la transformée continue d'ondelette exige une continuité des valeurs de ces paramètres (a, b) . Cette transformation est appliquée dans l'approximation d'un signal.

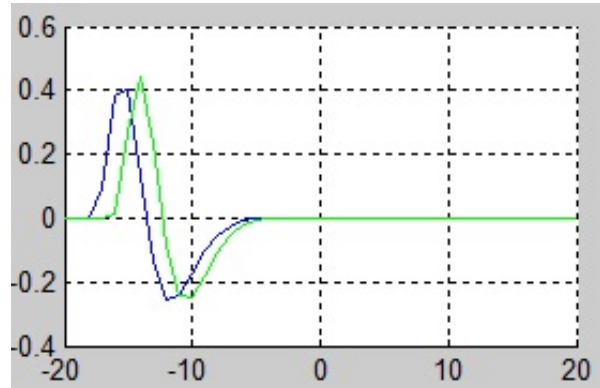


FIGURE 2.5 – Première dérivée de la fonction Bêta(en trait bleu) et celle dilatée tradatée(en trait vert)

La transformée continue en ondelettes consiste à élaborer des coefficients d'ondelettes $W(a, b)$ définis par :

$$W(a, b) = \langle f, \Psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(x) \Psi_{a,b}(x) dx = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} f(x) \Psi_{a,b}\left(\frac{x-b}{a}\right) dx. \quad (2.11)$$

Ces coefficients peuvent être utilisés pour reconstruire la fonction d'origine f en appliquant la formule de reconstruction dont l'existence est conditionnée par l'existence de la quantité C_Ψ définie par [Daubechies 1990] :

$$C_\Psi = \int_{-\infty}^{+\infty} \frac{|\hat{\Psi}(w)|}{|w|} dw < +\infty. \quad (2.12)$$

Où $\hat{\Psi}$ est la transformation de Fourier de Ψ .

La transformation en ondelettes est inversible et la fonction peut être reconstruite après analyse suivant l'équation [Daubechies 1990] :

$$f(x) = \frac{1}{C_\Psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(a, b) \frac{1}{\sqrt{a}} \Psi\left(\frac{x-b}{a}\right) \frac{dadb}{a^2}. \quad (2.13)$$

Le concept d'une fonction d'ondelette exige la condition d'admissibilité qui vérifie l'existence de la quantité C_Ψ qui sert à restreindre l'ensemble des fonctions

admissibles comme fonctions analysantes qui vérifient la condition suivante :

$$\hat{\Psi}(0) = \int_{-\infty}^{+\infty} \Psi(x) dx = 0. \quad (2.14)$$

Ces concepts concernent la transformée continue en ondelettes. Dans le cadre de notre travail nous souhaitons développer une topologie d'un réseau d'ondelettes en utilisant une approche d'approximation qui utilisera la transformée discrète en ondelettes.

2.3.2.4 L'analyse discrète en ondelettes

La discrétisation de la transformée en ondelettes doit être élaborée si nous voulons obtenir une transformation non redondante. Pour faire cette décomposition discrete nous devons prendre en considération les valeurs prises par les deux paramètres de l'ondelettes (a, b) . Il faut que ces paramètres prennent des valeurs dans un sous-ensemble discret de \mathfrak{R} . Cette discrétisation utilise souvent les ensembles de paramètres a et b définis par $a = a_0^m$ et $b = nb_0 a_0^m$, avec $(m, n) \in \mathbb{Z}^2$, l'ensemble des entiers $a_0 > 1$, $b_0 > 0$, [Daubechies 1992][Antonini 1992][Payan 2006].

La famille des fonctions analysantes $\Psi_{m,n}$ est alors donnée par :

$$\Psi_{m,n} = a_0^{\frac{-m}{2}} \Psi(a_0^{-m} x - nb_0). \quad (2.15)$$

Ainsi, pour un signal constituant a_0^j points nous calculons alors uniquement les coefficients :

$$W(a, b) = a_0^{\frac{-m}{2}} \sum_{i=1}^{a_0^j} f(x) \Psi(a_0^{-m} x - nb_0). \quad (2.16)$$

avec $m = 1, \dots, j$ et $n = 1, \dots, a_0^{j-m}$.

En fait, les propriétés de l'approximation obtenue sont élaborées par la sélection

tion des paramètres (a, b) . Les fonctions analysantes élaborent des informations redondantes lorsque a est proche de 1 et b est proche de 0. Si on choisit $a_0 = 2$ et $b_0 = 1$, nous parlons alors de transformée dyadique.

Dans le cadre de notre travail nous souhaitons implémenter une architecture d'un réseau d'ondelettes qui utilisera l'ondelette bêta comme étant une fonction de transfert dans la couche cachée.

2.3.2.5 Les ondelettes Bêta

2.3.2.5.1 Définition

L'ondelette Bêta est une fonction paramétrable qui est définie par $\beta(x) = \beta_{x_0, x_1, p, q}(x)$ [Alimi 2003], avec x_0, x_1, p et q des paramètres réels vérifiant : $x_0 < x_1$ et $x_c = \frac{px_1 + qx_0}{p+q}$. Nous devons nous limiter au seul cas où $p > 0$ et $q > 0$.


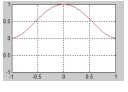
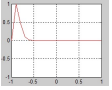
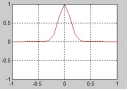
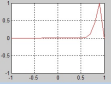
La fonction Bêta est définie comme suit :

$$\text{Beta}(x) = \begin{cases} \left(\frac{x-x_0}{x_c-x_0}\right)^p \left(\frac{x_1-x}{x_1-x_c}\right)^q & \text{si } x \in [x_0, x_1] \\ 0 & \text{sinon} \end{cases} \quad (2.17)$$

Les différentes formes de la fonction Bêta prises dans l'intervalle $[-1, 1]$, soit $x_0 = -1$ et $x_1 = 1$, pour distincts valeurs des paramètres p et q présentés dans le tableau 2.1 suivant :

**Chapitre 2 : Des Réseaux de neurones vers les Réseaux
d'ondelettes : Théorie et construction**

Tableau 2.1 – Les différentes formes et constatations de la fonction Bêta

p	q	Forme de la fonction Bêta Prise dans l'intervalle [-1,1], x0= -1, x1=1	Constations
2	10		La fonction Bêta est légèrement translatée vers la gauche
2	2		La fonction Bêta est symétrique
2	40		La fonction Bêta est fortement translatée vers la gauche
30	30		La fonction Bêta est symétrique
40	2		La fonction Bêta est fortement translatée vers la droite

2.3.2.5.2 Les ondelettes Bêta 1D

D'après les différentes formes de la fonction bêta nous constatons encore que cette fonction ne s'annule qu'en x_0 et x_1 donc elle ne vérifie pas la propriété d'oscillation, mais il a été prouvé dans les travaux[Amar 2005] [Amar 2006] que toutes les dérivées de la fonction bêta sont des ondelettes admissibles.

Les modifications des paramètres fonctionnels de la fonction bêta x_0, x_1, q et p nous permet d'obtenir différentes ondelettes. La dérivée n d'une ondelette bêta unidimensionnelle est élaborée par l'expression suivante[Bellil 2004] :

$$\Psi_n(x) = \frac{d^n \beta(x)}{dx^n} = \left[(-1)^n \frac{n!p}{(x-x_0)^{n+1}} + \frac{n!q}{(x_1-x)^{n+1}} \right] \beta(x) + P_n(x)P_1(x)\beta(x) + \sum_{i=1}^n C_n^i \left[(-1)^n \frac{(n-i)!p}{(x-x_0)^{n+1-i}} + \frac{(n-i)!q}{(x_1-x)^{n+1-i}} \right] \times P_1(x)\beta(x). \quad (2.18)$$

avec :

$$P_1(x) = \frac{p}{x-x_0} - \frac{q}{x_1-x}. \quad (2.19)$$

$$P_n(x) = (-1)^n \frac{n!p}{(x-x_0)^{n+1}} + \frac{n!q}{(x_1-x)^{n+1}}. \quad (2.20)$$

Pour $n \in N, 0 < n < p$ et $p = q$, la fonction d'ondelette $\Psi_n(x) = \frac{d^n \beta(x)}{dx^n}$ nous donne les différentes formes d'ondelettes qui sont les trois premières dérivées de la fonction Bêta et qui sont présentées respectivement :

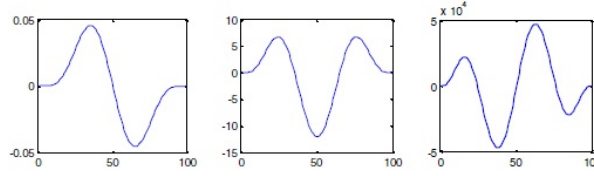


FIGURE 2.6 – Différentes formes d'ondelettes des dérivées de la fonction Bêta

Ces différentes formes d'ondelettes des dérivées de la fonction Bêta ont les propriétés suivantes :

Admissibilité : pour tout $n \in \mathbb{N}$ et $\forall n > 0$ les fonctions $\Psi_n(x) = \frac{d^n \beta(x)}{dx^n}$ vérifient la condition d'admissibilité :

$$\int_{-\infty}^{+\infty} \frac{|TF\Psi_n(w)|^2}{|w|} dw. \quad (2.21)$$

Energie finie : une ondelette a une énergie finie si :

$$\int_{-\infty}^{+\infty} \Psi(x) dx = 0. \quad (2.22)$$

Toutes les dérivées de la fonction bêta ont été prouvées qu'elles satisfont la condition d'énergie finie pour tout $n \in \mathbb{N}$ et $\forall n > 0$.

$$\int_{-\infty}^{+\infty} \frac{d^n \beta(x)}{dx^n} dx = \int P_n(x) \beta(x) dx = 0. \quad (2.23)$$

Moment nuls : il a été démontré que la dérivée n de la fonction Bêta a n moments nuls :

$$\int_{x_0}^{x_1} x^n P_n(x) \beta(x) dx = 0. \quad (2.24)$$

Support compact : toutes les ondelettes dérivées de la fonction Bêta ont un support $]x_0, x_1[$.

Translation et dilatation : les dérivées de la fonction Bêta vérifient les propriétés de translation et de dilatation.

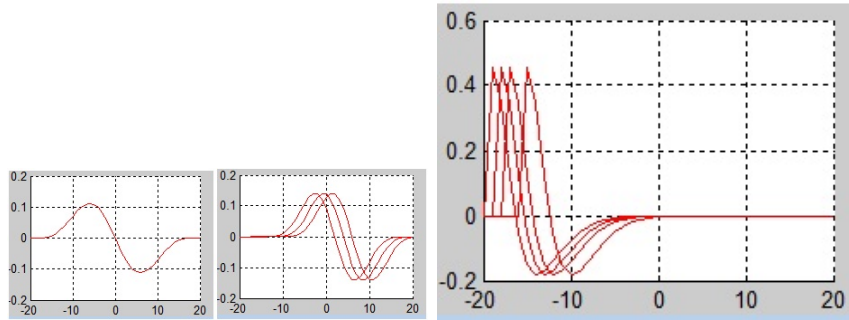


FIGURE 2.7 – Translation et dilatation des dérivées de la fonction Bêta

2.3.2.5.3 Les ondelettes Bêta 2D

L'ondelette Bêta 2D est le produit de deux ondelettes monodimensionnelles comme toute ondelette séparable [Bellil 2008].

$$\beta(x, y) = \beta(x) * \beta(y). \quad (2.25)$$

La dérivée première de la fonction Bêta concernant l'ondelette bêta bidimensionnelle.

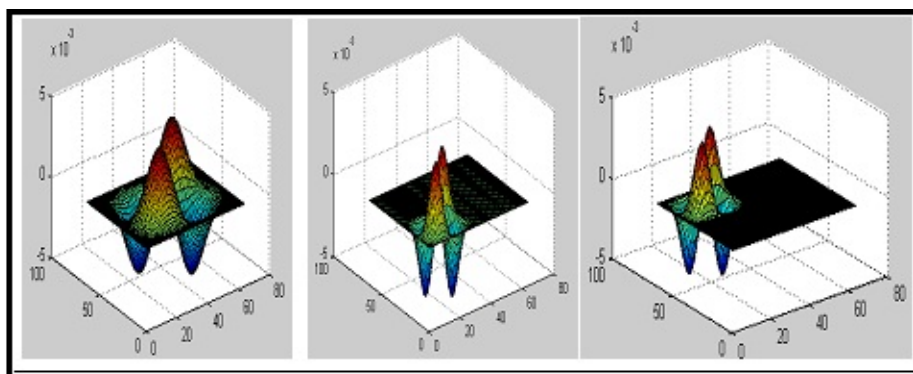


FIGURE 2.8 – Dérivée première dilatée et translatée de la fonction Bêta 2D

2.3.3 Architecture des réseaux d'ondelettes

2.3.3.1 Type 1

La topologie dans ce modèle, est proche de celle d'un réseau à fonction radiale (RBF) [Park 1991]. Le réseau considéré comporte deux couches : une première couche avec N_i entrées et une couche cachée constituée de N_w ondelettes, et un sommateur de sortie accueillant les sorties pondérées des ondelettes. Les cellules d'une couche sont liées à toutes les cellules de la couche du réseau suivant uniquement. La propagation des valeurs se fait des cellules d'entrées vers les cellules de sortie se trouvant au niveau de ce type de réseau. Ce modèle est donc tout à fait comparable aux réseaux de neurones formels utilisant des fonctions sigmoïdales. Elle illustre également une ressemblance avec le modèle des réseaux RBF (Réseau à Fonction Radiale) mais la fonction de transfert est substituée par une fonction ondelette $\Psi_{a,b}(t)$.

L'algorithme d'apprentissage est hérité aussi de celui des réseaux de neurone à fonction radiale(RBF). Il vise à diminuer l'erreur commise entre l'entrée du réseau et sa sortie en corrigeant et en rectifiant les paramètres directs ou indirects de ce réseau. Ces paramètres vont être ajustés pour diminuer cette erreur. La fonction de coût quadratique est appliquée pour évaluer cette erreur. L'apprentissage vise, ainsi, à minimiser et réduire le coût empirique donné par la quantité E :

$$E = \frac{1}{2} \sum_{t=1}^T (y_d - y(t))^2. \quad (2.26)$$

Où $y(t)$ est la sortie réelle élaborée par le réseau et y_d la sortie désirée. L'expression de la sortie du réseau est :

$$y(t) = \sum_{k=1}^N W_k \Psi_k\left(\frac{t - b_k}{a_k}\right). \quad (2.27)$$

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

L'algorithme de descente en gradient est utilisé à chaque itération de cet algorithme. Un exemple est présenté au réseau (paire entrée/sortie), nous propagons le calcul d'une couche à une autre jusqu'à la couche de sortie du réseau. L'algorithme d'apprentissage consiste à modifier les paramètres dans la direction opposée au gradient de la fonction d'erreur.

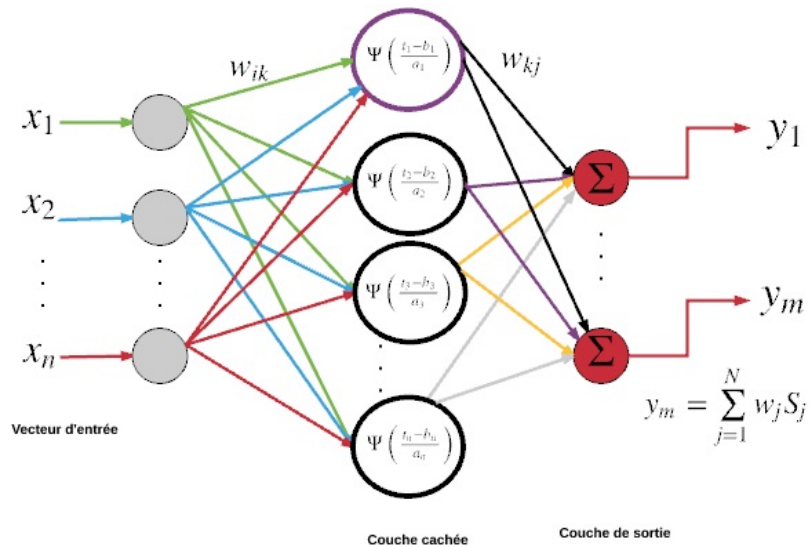


FIGURE 2.9 – Type 1 de réseaux d'ondelettes

2.3.3.2 Type 2

Dans ce type de modèle, l'entrée est un ensemble de paramètres t_i qui représentent les positions ordonnées du signal à traiter. Ces entrées ne sont pas des données proprement dites, mais uniquement des valeurs indiquant des positions bien précises du signal. La couche cachée du réseau contient un ensemble de neurones, dans chaque neurone une ondelette translatée et dilatée. La couche de sortie constitue un seul neurone qui additionne les sorties de la couche cachée du réseau pondérées par les poids de connexion w_i . L'algorithme de la descente de gradient est utilisé pour faire l'apprentissage de ce réseau. Ce modèle, introduit pour la première fois par Zhang et Benveniste, est un

cas particulier de l'architecture du modèle des réseaux d'ondelettes présentés au dessus [Zhang 1992] :

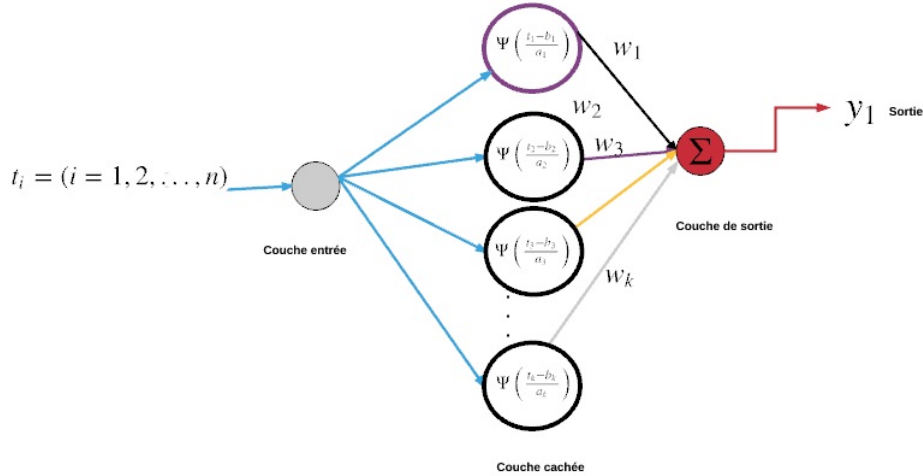


FIGURE 2.10 – Type 2 de réseaux d'ondelettes

2.3.4 De la transformée inverse aux réseaux d'ondelettes

Soit N_w un nombre fini d'ondelettes Ψ_i élaborées à partir de l'ondelette mère Ψ selon l'expression (2.13) qui représente la fonction f comme étant une fonction de carré sommable sous la forme d'une intégrale sur toutes les dilatations et toutes les translations possibles de l'ondelette mère, nous pouvons élaborer la relation suivant :

$$f(x) = \sum_{i=1}^{N_w} c_i \Psi_i(x). \quad (2.28)$$

Cette relation (2.28) qui représente une somme finie, est considérée comme étant une approximation d'une transformée inverse ou une décomposition d'une fonction en une somme pondérée d'ondelettes, où chaque c_i est proportionnel à $W(a_i, b_i)$ donc la réalisation d'une approximation d'une fonction définie sur un domaine fini, la transformée en ondelette de cette fonction existe, et sa reconstruction est possible.

2.3.5 Comparaison des réseaux d'ondelettes aux réseaux de neurones

La principale similitude entre ces deux réseaux réside au fait que les deux réseaux évaluent et calculent une combinaison linéaire, de fonctions non linéaires dont la forme et la structure dépendent de paramètres modifiables et ajustables (translations et dilatations) de cette combinaison. Mais la différence majeure est la nature des fonctions de transfert utilisées par les cellules cachées du réseau utilisé. Nous dressons dans ce qui suit quelques différences :

- Les ondelettes se trouvant au niveau des couches de réseau sont des fonctions qui décroissent rapidement, et tendent vers zéro dans toutes les directions de l'espace. Elles sont donc locales si (dilatation) a_i est petit.
- La forme de chaque ondelette unidimensionnelle est élaborée par deux paramètres ajustables et modifiables (translation et dilatation) qui sont des paramètres structurels de l'ondelette utilisée.
- Chaque ondelette unidimensionnelle possède deux paramètres structurels (translation et dilatation). D'où, pour chaque ondelette multidimensionnelle le nombre de paramètres ajustables et modifiables est le double du nombre de variables utilisés.
- etc.

2.4 Apprentissage des réseaux d'ondelettes

L'entraînement d'un réseau de neurones, par conséquent un réseau d'ondelettes à l'aide de l'algorithme rétropropagation est un axe principal de recherche. Cet axe se concentre sur la limite de cet algorithme pour entraîner ce réseau. Cette méthode d'apprentissage est conçue pour déterminer les poids et les paramètres convenable pour un réseau dont la topologie est

fixée à priori. Dans ce contexte, les spécialistes doivent définir l'architecture de réseau en déterminant le nombre de couche cachée, les neurones (les ondelettes), le taux et l'algorithme d'apprentissage. La conception de topologie réseau n'est pas une méthode définitive car aucun spécialiste ne peut garantir à l'avance l'optimalité de son choix car l'ajout d'une ondelette ou neurone peut augmenter le risque de surajustement (surapprentissage) des données qui peut compromettre la précision du modèle. Pour remédier cette pénurie, la plupart des chercheurs adoptent la stratégie de tests et d'erreurs. Cependant, le test de nombreuses options ne garantit pas une solution optimale. Pour cette raison il y a recours aux algorithmes constructifs et aux techniques qui utilisent dans certains temps des procédures de sélection. Dans ce chapitre nous présenterons les différentes méthodes qui permettent de construire les réseaux d'ondelettes et nous essayerons de citer les différents algorithmes d'apprentissage. Pour chacune de ces méthodes, nous préciserons les avantages et les inconvénients, dans la perspective de la mise au point d'une approche simple à appliquer et à mettre en œuvre et peu coûteuse en temps de calcul.

L'apprentissage est une étape de construction d'un réseau d'ondelettes. Au cours de cette phase le comportement et la réponse du réseau sont modifiés et ajustés pour obtenir la sortie désirée et souhaitée. Il existe deux grandes classes d'algorithmes d'apprentissage : les algorithmes supervisés et les algorithmes non supervisés.

2.4.1 L'apprentissage supervisé

Les algorithmes supervisés sont utilisés pour des réseaux de neurones dont le résultat attendu est connu au préalable. Cette classe n'étant pas utile dans notre étude, ces algorithmes seront présentés brièvement dans ce chapitre.

2.4.2 L'apprentissage non supervisé

Le problème de l'apprentissage non supervisé est d'essayer de trouver une structure pour les données non étiquetées. Cet apprentissage est étroitement lié au problème d'estimation de la densité dans les statistiques. Dans cette méthode il n'y a pas une sortie à priori ; en effet au début de processus il y a traitement de l'ensemble des données (entrées). Ensuite, il y a traitement automatique de ces données comme des variables aléatoires et enfin il y a construction d'un modèle de densités jointes pour les données à étudier. Cette méthode qui est appelée apprentissage par exploration où l'algorithme d'apprentissage ajuste les poids des connexions entre neurones de façon à augmenter la qualité de classification des entrées. Au cours de ce type d'apprentissage, les réseaux d'ondelettes utilisent la probabilité. En effet, ces réseaux vont se modifier en fonction des traitements statistiques de l'entrée pour élaborer des catégories en attribuant et en optimisant une valeur de qualité, aux catégories reconnues.

Nous pouvons regrouper les réseaux à apprentissage non supervisée en deux groupes, suivant la façon dont se déroule l'apprentissage : plus précisément, suivant que chaque exemple de la base d'apprentissage entraîne la modification des poids d'un ou plusieurs neurones à chaque itération, ou pas. Nous allons détailler ces types de réseaux ultérieurement. L'apprentissage non supervisé articule le contenu de ce chapitre. Cette classe d'apprentissage sera présentée d'une façon détaillée dans ce chapitre. Les algorithmes d'apprentissage non supervisé consistent à entraîner un réseau de neurones, par conséquent un réseau d'ondelettes. L'objectif fondamental de cet apprentissage est de construire un réseau performant qui peut par exemple classifier des données d'une façon claire et significative ce qui nous permet d'analyser et interpréter les résultats pour fournir des connaissances utiles. Plusieurs domaines d'applications des réseaux d'ondelettes utilisant un apprentissage non supervisé sont l'interpo-

lation de fonctions, la reconnaissance de formes et la classification lorsque les classes auxquelles doivent appartenir les données ne sont pas connues à priori. Il existe plusieurs types d'apprentissage non supervisé.

L'architecture utilisant l'apprentissage non supervisé

L'apprentissage non supervisé est appliquée à un réseau de neurones, par conséquent un réseau d'ondelettes est constitué de trois couches :

- Une couche d'entrée est formée par le même nombre de neurone que le nombre de données qui ont des composantes. Pour chaque neurone, par conséquent une ondelette adopte une activation égale à la composante correspondante de la donnée proposée à l'entrée du réseau ;
- Une couche cachée ;
- Une couche de sortie : dans cette couche chaque neurone i est connecté avec chaque neurone s de la couche cachée avec la pondération w_{si} et nous pouvons associer à chacun des neurones de sortie un vecteur de même dimension que la dimension des données. Nous nommons aussi ces vecteurs les vecteurs de références ;

2.4.3 Apprentissage d'un réseau d'ondelettes par l'analyse multirésolution

L'analyse multirésolution (AMR) est une approche qui permet de préciser et de définir l'espace d'approximation. Dans cette section nous essayerons de préciser les propriétés fondamentales des analyses multirésolutions que nous appliquerons par la suite au niveau d'apprentissage d'un réseau d'ondelettes. Le principe d'analyse multirésolution a été proposé par S.Malat en 1989 [Mallat 1989] et il est précisé de façon approfondie par Y. Meyer [Meyer 1990]. L'espace d'approximation défini par l'analyse multirésolution constitue les fonctions d'échelles et l'espace des détails contenant les fonctions d'ondelettes.

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

Ces fonctions vont être utilisées pour décomposer un signal selon les échelles caractéristiques élaborées par le principe de l'analyse multirésolution. Cette analyse de $L^2(\mathfrak{R})$ est une suite de sous-espaces fermés $(V_j)_{j \in \mathbb{Z}}$ de telle que :

$$V_j \subset V_{j+1}. \quad (2.29)$$

$$\bigcap_{j \in \mathbb{Z}} V_j = \{0\}. \quad (2.30)$$

$$\bigcup_{j \in \mathbb{Z}} V_j. \quad (2.31)$$

Cette relation est dense dans $L^2(\mathfrak{R})$.

Soit une fonction d'échelle ϕ qui génère une base orthonormée par translation sur un espace V_j et considérons un signal g . L'approximation de ce signal sur V_j est :

$$A_j = \sum_n a_n^j \phi_{j,n}. \quad (2.32)$$

Avec a_n^j les coefficients qui se calculent comme suit :

$$a_n^j = \langle g, \phi_{j,n} \rangle. \quad (2.33)$$

Soit W_{j+1} un espace complémentaire orthogonal de l'espace V_{j+1} . W_{j+1} existe et justifie la relation suivante :

$$V_{j+1} \subset V_j \Rightarrow V_j = V_{j+1} \oplus W_{j+1}. \quad (2.34)$$

Une base orthogonale de l'espace W_{j+1} constituant un ensemble de fonctions d'ondelettes dyadiques générée par une seule ondelette mère dilatée et traduite. Le détail (D) du signal g dans cet espace (W_{j+1}) est élaboré par la relation suivante :

$$D_j = \sum_n d_n^j \Psi_{j,n}. \quad (2.35)$$

Avec d_n^j les coefficients qui se calculent comme suit :

$$d_n^j = \langle g, \Psi_{j,n} \rangle. \quad (2.36)$$

Les deux relations (2.32) et (2.35) sont utilisées ensemble pour construire le signal g comme suit :

$$g = \sum_n a_n^j \phi_{j,n} + \sum_n d_n^j \Psi_{j,n} = A_j + D_j. \quad (2.37)$$

V_j sont des espaces d'approximation emboîtés et par suite à l'aide des échelles multirésolution le signal de rapprochement (approximation) A_j peut être analysé successivement donc à une échelle donnée j le signal g peut être exprimé par :

$$g = \sum_k a_k^i \phi_{i,k} + \sum_j \sum_k d_k^j \Psi_{j,k}. \quad (2.38)$$

Pour cela le signal g peut être exprimé par la relation (2.39) lorsque l'analyse est répétée jusqu'à la dernière échelle :

$$g = A_{n-1} + D_{n-1} + \dots + D_1 + D_0. \quad (2.39)$$

Le calcul des coefficients a_n^i se fait lorsque les fonctions d'échelle $\phi_{j,n}$ forment une base orthogonale qui nous permet de calculer la base duale composée par les fonctions $\tilde{\phi}_{j,n}$ de l'ensemble des fonctions primales $\phi_{j,n}$. Une base duale d'ensemble de fonctions peut être calculée par la relation (2.40) selon [Krug01, Somm01] :

$$\tilde{\phi}_i = \sum_{j=1}^N (\Phi_{i,j})^{-1} \phi_j. \quad (2.40)$$

Avec $\Phi_{i,j} = \langle \phi_i, \phi_j \rangle$

Le rapprochement du signal g à une échelle j et pour une position n , nous

pouvons calculer les coefficients a_n^i par la relation suivante :

$$a_n^j = \langle g, \tilde{\phi}_{j,n} \rangle. \quad (2.41)$$

De la même manière nous pouvons calculer la base duale des ondelettes et les coefficients de détail :

$$d_n^j = \langle g, \tilde{\Psi}_{j,n} \rangle. \quad (2.42)$$

L'analyse multirésolution et la reconstitution d'un signal g sont présentées dans les deux figures suivantes. La figure 2.11 montre la procédure de décomposition du signal g en utilisant les ondelettes et les fonctions d'échelle primales (ϕ_i, Ψ_i) et la figure 2.12 représente les étapes de reconstruction du signal g en utilisant les ondelettes et les fonctions d'échelle duales :

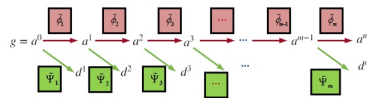


FIGURE 2.11 – Décomposition d'un signal g en appliquant l'analyse multirésolution

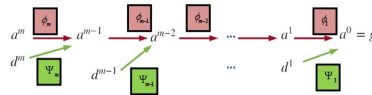


FIGURE 2.12 – Reconstruction d'un signal g

2.4.4 Apprentissage spécifique aux réseaux d'ondelettes

La construction d'une bibliothèque d'ondelettes est une étape très importante ; en effet, cette procédure participe d'une façon fondamentale à l'amélioration de performance de réseau de point de vue complexité et décision. Pour cette raison la construction de cette bibliothèque d'ondelettes s'incorpore dans la phase d'apprentissage. Cette structure contient les ondelettes candidates qui

vont être utilisées dans la couche cachée de réseau et par suite la construction de topologie de réseau. Dans cette partie nous préciserons la manière de construction de cette bibliothèque, puis nous expliquerons l'optimisation de ce réseau construit et en fin nous présenterons les méthodes de sélection appliquées pour sélectionner les ondelettes convenable afin de construire le réseau d'ondelettes.

2.5 Techniques de construction des réseaux d'ondelettes

2.5.1 Les méthodes incrémentales utilisées pour la construction des réseaux d'ondelettes

Les algorithmes qui utilisent une construction incrémentale [Dunkin 1997] [Polikar 2001] [B.Fritzke 1994][Fahlman 1990] , permettent de construire un réseau avec un nombre minimal de neurones sur la couche cachée. Au début ce nombre est égal à un seul neurone. Ensuite l'occurrence de neurone augmente progressivement lorsque le niveau d'erreur stagne durant la phase d'apprentissage. Le processus se poursuit jusqu'à une construction d'un réseau avec un nombre de neurones limité et optimal et par suite le temps d'apprentissage diminue. La réduction de ce temps permet d'avoir une complexité acceptable du système.

L'algorithme de rétropropagation du gradient

L'apprentissage supervisé par l'algorithme de rétropropagation du gradient s'applique sur l'architecture de réseaux multicouches pour assurer la pondération de ce type de réseaux. Cet apprentissage consiste à entraîner le réseau de neurones et par conséquent le réseau d'ondelettes. Nous alimentons le réseau par les entrées et nous lui demandons d'ajuster sa pondération pour

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

retrouver les sorties désirées. Pour assurer cet objectif : au début, l'algorithme de rétropropagation propage les entrées en avant jusqu'à une entrée calculée par le réseau. Ensuite, il y a comparaison de sortie calculée à la sortie réelle connue. Enfin, il y a modification des poids du réseau afin de minimiser l'erreur commise entre la sortie calculée et la sortie désirée. Pendant cette phase la rétropropagation d'erreur commise se fait vers l'arrière jusqu'à la couche cachée. Le processus s'arrête lorsqu'une erreur de sortie soit négligeable. Cet apprentissage est une méthode de rétropropagation du gradient ou encore "backpropagation". Cet algorithme se déroule en deux étapes :

- Une étape de propagation : au cours de cette phase il y a présentation d'une configuration d'entrée au réseau puis propagation de cette entrée de proche en proche de la couche d'entrée à la couche de sortie en traversant les couches cachées ;
- Une étape de rétropropagation qui se déroule après la phase de propagation. Pendant cette phase il y a minimisation de l'erreur commise sur l'ensemble des entrées présentées. Cette erreur est présentée comme étant une fonction des poids synaptique ; elle est égale à la somme des carrés des différences entre les réponses calculées et celle désirées pour toutes les entrées dans la phase d'apprentissage ;

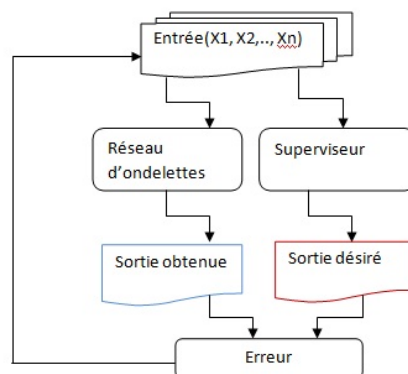


FIGURE 2.13 – L'algorithme de l'apprentissage supervisé.

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

Un réseau d'ondelettes est composé par des ondelettes interconnectées par des liaisons pondérées w . L'algorithme de rétropropagation est appliqué pendant la phase d'apprentissage afin de minimiser l'erreur commise par le réseau d'ondelettes sur les éléments de la base d'apprentissage en corrigeant les poids et les paramètres (dilatation(a) et translation(b)) pour chaque ondelette constituant la couche cachée de réseau. Pour évaluer cette erreur une fonction de coût quadratique est appliquée donc l'apprentissage vise à minimiser cette fonction qui est défini par la relation suivante :

$$E = \frac{1}{2} \sum_{t=1}^{n_t} (y_d(t) - y(t))^2. \quad (2.43)$$

Où $y_d(t)$ la sortie désirée et $y(t)$ est la sortie réelle obtenue par le réseau.

$$y(t) = \sum_{i=1}^N w_i \Psi_i\left(\frac{t - b_i}{a_i}\right). \quad (2.44)$$

Pendant la phase d'apprentissage il ya propagation de calcul d'une couche à une autre jusqu'à la couche de sortie et après chaque itération il y a modification des paramètres dans la direction opposée au gradient de la fonction d'erreur. Ces changements des paramètres se font comme suit[Zhang 1992] [Iyengar 2002] :

$$w(t + 1) = w(t) + \mu_w \Delta w. \quad (2.45)$$

$$v(t + 1) = v(t) - \varepsilon(t) \frac{\partial E}{\partial v}. \quad (2.46)$$

$$a(t + 1) = a(t) + \mu_a \Delta a. \quad (2.47)$$

$$b(t + 1) = b(t) + \mu_b \Delta b. \quad (2.48)$$

Où :

$$\Delta w = -\frac{\partial E}{\partial w} = \sum_{t=1}^{N_i} e(t) \Psi\left(\frac{t - b_i}{a_i}\right). \quad (2.49)$$

$$\Delta a = -\frac{\partial E}{\partial a} = \sum_{t=1}^{N_i} e(t) w_{ij} \frac{\partial \Psi\left(\frac{t - b_i}{a_i}\right)}{\partial a_i}. \quad (2.50)$$

$$\Delta b = -\frac{\partial E}{\partial b} = \sum_{t=1}^{N_i} e(t) w_{ij} \frac{\partial \Psi\left(\frac{t - b_i}{a_i}\right)}{\partial b_i}. \quad (2.51)$$

Avec μ_a , μ_b et μ_w sont les pas d'apprentissage de trois paramètres (a, b, w) du réseau et l'erreur est calculée par la relation suivante :

$$e(t) = y_d(t) - y(t). \quad (2.52)$$

L'ajustement ou le réglage du pas de gradient μ_k est nécessaire : en effet, une petite valeur de ce paramètre ralentit la progression de l'algorithme ; en revanche une grande valeur aboutit généralement à un phénomène d'oscillation autour de la solution. L'inconvénient majeur de cet algorithme est que cette méthode d'apprentissage ne permet pas de mieux connaître le nombre optimal de neurones à utiliser sur la couche cachée. Pour résoudre ce problème il y a naissance de plusieurs algorithmes d'apprentissage qui permettent de construire un réseau de neurones avec un nombre optimal de neurones et par suite amélioration des complexités. Ces algorithmes sont appelés des algorithmes de construction incrémentale.

L'algorithme de construction en cascade

Cet algorithme a été conçu par Scott Fahlman et Christina Lebiere[Fahlman 1990]. L'objectif principal de cet algorithme était de trouver pourquoi l'algorithme d'apprentissage de rétropropagation était si lent et ensuite proposer une alternative plus rapide. Les auteurs ont suggéré que les problèmes de vitesse de l'algorithme de rétropropagation ont été

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

principalement causés par deux facteurs : le premier est qu'il est difficile de déterminer la taille des étapes de mise à jour, lors de l'apprentissage. Les petites étapes impliquent un apprentissage plus lent alors que les grandes étapes peuvent dépasser la meilleure solution. Le deuxième facteur se concentre sur l'évolution du neurone : en effet, chaque neurone (ondelette) caché tente d'évoluer dans le cadre du classifieur, mais ses neurones connectés évoluent aussi. Cette mutation constante et parallèle rend plus difficile l'apprentissage de neurones (ondelettes) cachés. Ce type d'apprentissage utilise deux notions pour construire un réseau des neurones ; en effet, le premier concept se concentre sur la mise en œuvre de l'architecture du réseau à construire ; l'ajout de neurone se fait d'une façon progressive pendant la phase d'apprentissage. Le neurone ajouté se connecte directement avec les neurones de la couche cachée récemment ajoutée dans la même couche. De même ce neurone se relie avec les neurones de la couche d'entrée. Le deuxième principe de cet algorithme est de maximiser le rapport entre la valeur de sortie de neurone additionné et l'erreur résiduelle du réseau. L'architecture particulière de ce réseau est présentée dans les figures 2.14, 2.15 et 2.16.

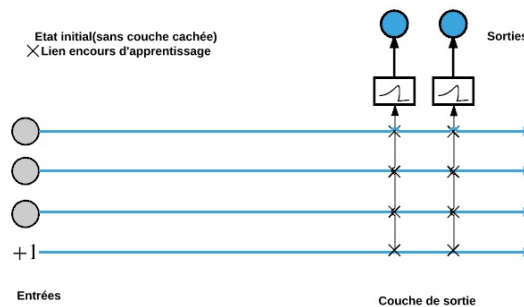


FIGURE 2.14 – L'état initial d'un réseau utilisant l'algorithme en cascade (sans couche cachée)

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

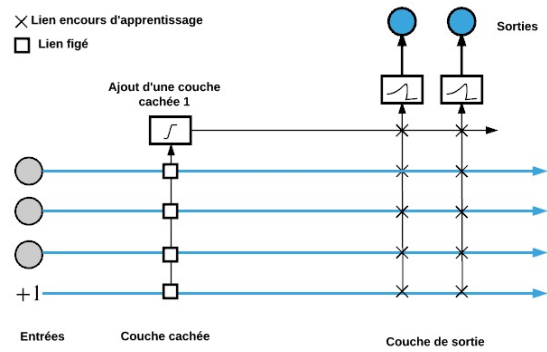


FIGURE 2.15 – L'ajout d'une première couche cachée au réseau utilisant l'algorithme en cascade

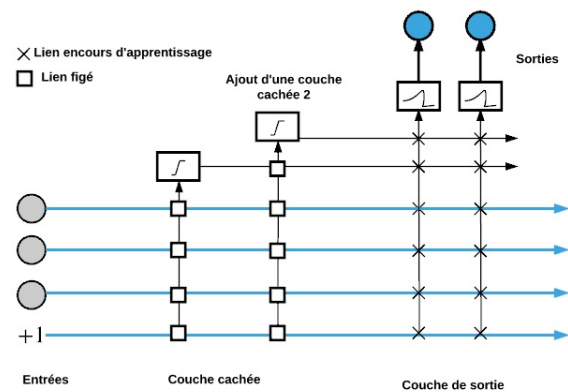


FIGURE 2.16 – L'ajout d'une deuxième couche cachée au réseau utilisant l'algorithme en cascade

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

Les avantages des réseaux de neurones en cascade sont bien connus. Premièrement, aucune structure de réseaux n'est prédéfinie. Le réseau est automatiquement construit à partir des données d'apprentissage. Le réseau détermine sa propre taille et topologies. En second lieu, le réseau en cascade apprend vite, car chacun de ses neurones est entraîné indépendamment des autres. Il apprend au moins 10 fois plus rapide que la norme Algorithmes Back-propagation. De même ce réseau est utile pour l'apprentissage progressif dans lequel les nouvelles informations sont ajoutées au réseau déjà formé. Toutefois, l'inconvénient est que les réseaux en cascade peuvent augmenter le risque de surajustement (surapprentissage) des données qui peut compromettre la précision du modèle.

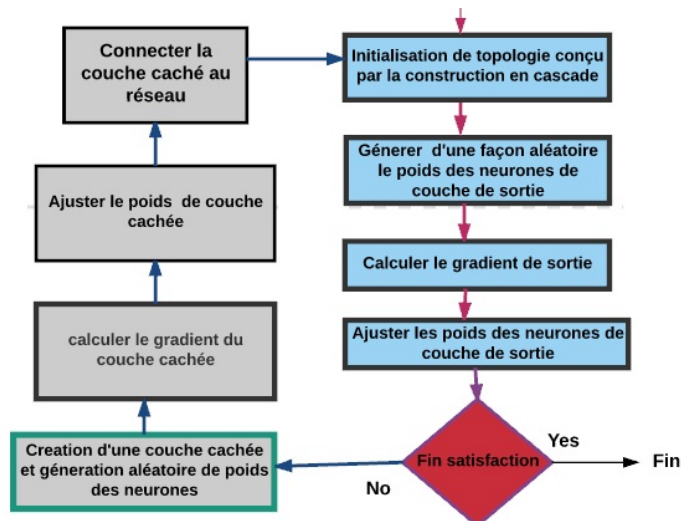


FIGURE 2.17 – Construction d'un réseau de neurones par l'algorithme en cascade

2.5.2 Construction de réseaux d'ondelettes par l'algorithme pyramidal

L'algorithme pyramidal a été proposé par Stephen Gallant [Gallant 1986]. Cet algorithme a été utilisé pour construire des réseaux d'ondelettes. Il permet d'améliorer la complexité d'un réseau ; cette méthode permet d'accélérer la phase d'apprentissage. La construction d'un réseau d'ondelettes se fait par l'ajout d'une fonction d'ondelette au niveau de la couche cachée du réseau. L'algorithme pyramidal est rapide et important car il utilise un schéma itératif ou incrémental donc la construction d'un réseau d'ondelettes se fait d'une façon incrémentale. Cet algorithme élabore les coefficients de fonction d'ondelettes en approximant le signal à différents échelles de résolution.

2.5.3 Construction de réseaux d'ondelettes par des techniques basées sur la transformée discrète

L'architecture des réseaux d'ondelettes fondées sur la transformée discrète a été construite par plusieurs techniques. La performance d'un réseau d'ondelettes exige le choix et l'évaluation de plusieurs paramètres. En effet, les paramètres à déterminer pour la construction du réseau d'ondelettes sont :

- Le nombre d'ondelettes suffisant pour atteindre une performance voulue ;
- L'évaluation des différents paramètres de réseau : les paramètres structurels, les pondérations (poids) des ondelettes et les termes directs.

Le problème de modélisation par des fonctions paramétrées réside dans l'évaluation et la détermination des paramètres (dilatation et translation) du réseau d'ondelettes. Ces paramètres peuvent prendre des valeurs discrètes. La réduction du coût appliquant le gradient n'est pas envisageable. En revanche, les valeurs discrètes prises par les paramètres,

peuvent être utilisées pour concevoir des techniques et des méthodes à fin de sélectionner et choisir des ondelettes dans un ensemble d'ondelettes discrètes(bibliothèque). Donc la performance du modèle dépend du choix initial des ondelettes de la bibliothèque, et d'un choix judicieux dans cette bibliothèques. Pour résoudre ce problème il y a apparition de plusieurs méthodes ; par exemple : la technique fondée sur l'analyse fréquentielle, la méthode fondée sur la théorie des ondelettes orthogonales, la technique basée sur la construction des frames(structures obliques étroites)

2.5.4 Construction d'un réseau d'ondelettes basée sur l'analyse fréquentielle

Cette méthode se concentre sur l'estimation du spectre d'énergie de la fonction à approximer [Pati 1993], le spectre d'énergie se trouvant dans un domaine de fréquence. Ce domaine est obtenu en calculant la transformée de fourrier de la fonction à approximer et le domaine des amplitudes des variables d'entrées définit par l'ensemble d'exemples. Par suite, nous déterminons les ondelettes correspondant à ce domaine amplitude-fréquence afin de construire le réseau d'ondelettes. Cette méthode basée sur l'analyse fréquentielle montre l'avantage de tirer parti des principes de localité des ondelettes dans les domaines fréquentiel et spatial. En revanche, elle présente un inconvénient majeur, notamment pour les modèles qui possèdent plusieurs variables (multivariables) : le volume de calcul nécessaire à l'estimation du spectre de fréquence est très important.

2.5.5 Construction d'un réseau d'ondelettes en utilisant la théorie des ondelettes orthogonales

Cette technique a été proposée par J.Zhang en 95 [Zhang 1995]. C'est une approche qui applique des bases d'ondelettes orthogonales. Etant donné les amplitudes des entrées de l'ensemble d'apprentissage, se trouvant dans un domaine bien déterminé, Nous sélectionnons les fonctions des ondelettes possédant leur centre à l'intérieur de ce domaine. Le nombre de dilatations différentes à utiliser dépend de la performance désirée. Cette approche présente l'avantage de mettre à profit la propriété d'orthogonalité des ondelettes, mais cette méthode reste un peu difficile à mettre en œuvre, par exemple pour l'ondelette Haar, nous ne savons pas l'expression analytique simple pour les ondelettes mères qui élaborent des familles de fonctions orthogonales. Cet inconvénient rend cette approche peu efficace pour résoudre certains problèmes.

2.5.6 Construction d'un réseau d'ondelettes pour un système adaptatif

Cette méthode a été conçue pour construire des réseaux d'ondelettes qui vont être utilisés dans un système adaptatif de commande. Cette technique a été proposée par Cannon [a 1995]. Dans cette approche, il y a construction d'une bibliothèque d'ondelettes en considérant le domaine des valeurs des variables d'état du modèle. L'échelle des dilatations est évaluée en appliquant le spectre d'énergie de la fonction à approximer, donc la construction de réseau d'ondelettes se fait à l'aide des ondelettes sélectionnées à partir de la bibliothèque. Ces ondelettes doivent être pondérées périodiquement. Ces pondérations sont comparées à un seuil. Une ondelette est gardée ou exclue du réseau suivant sa pondération ; si elle est supérieure ou inférieure à ce seuil. Cette approche

de construction de réseaux d'ondelettes peut être appliquée indifféremment pour la construction de modèles dynamiques ou statiques. Elle présente l'inconvénient de nécessiter l'estimation du spectre d'énergie de la fonction à approximer.

2.5.7 Construction d'un réseau d'ondelettes basée sur la construction des frames

Les limites théoriques peuvent être un obstacle pour construire les réseaux d'ondelettes orthogonales (l'absence d'expression analytique simple pour des ondelettes mères engendrant des bases orthonormales). La résolution de ce problème se fait par l'application et l'utilisation des structures obliques (frames). La question qui se pose alors est la sélection des paramètres a (dilatation) et b (translation). La construction d'une bibliothèque des ondelettes à l'aide d'une structure oblique est étroite pour éviter le calcul du spectre d'énergie de la fonction à approximer. Les paramètres a et b sont alors respectivement égaux à 2 et à 1 c'est-à-dire la bibliothèque d'ondelettes est construite en utilisant l'échantillonnage dyadique : ($a_0 = 2$ et $b_0 = 1$) [Juditsky 1994] [Zhang 1997]. Quatre ou cinq dilatations différentes sont utilisées pour construire la bibliothèque d'ondelettes. L'ondelette la plus large est celle dont le support a la taille du domaine des exemples. Les ondelettes sélectionnées sont celles dont les centres sont à l'intérieur de ce domaine. Une méthode ou technique de sélection constructive ou destructive est ensuite utilisée aux ondelettes se trouvant dans la bibliothèque pour sélectionner celles qui sont les plus significatives pour modéliser le processus étudié. La taille d'une bibliothèque d'ondelettes peut provoquer un problème concernant la performance du réseau à construire et par la suite la performance de décision du modèle. Pour minimiser la taille de cette bibliothèque plusieurs travaux ont été effectués par exemple Zhang [Zhang 1993], [Zhang 1997] a appliqué une première minimisation de la biblio-

thèque en éliminant les ondelettes comportant peu ou pas d'exemples sur leurs supports. Ces situations sont particulièrement fréquentes pour des modèles à plusieurs entrées où les exemples ne sont pas répartis de manière uniforme. Cette approche présente l'avantage de procéder à une construction de la bibliothèque d'ondelettes de manière simple, avec peu de calculs et par la suite cette construction peut améliorer la performance de modèle de point de vue complexité et décision.

La théorie des frames utilisée pour optimiser l'apprentissage de réseau d'ondelettes. La transformée inverse en ondelettes discrète n'est valable que si la famille d'ondelette forme un frame. Cette transformée peut être considérée comme étant la sortie d'un réseau d'ondelettes. Soit une ondelette définie par $\Psi \in L^2(\mathbb{R})$, S un échantillonnage sur une grille dyadique et $B_\Psi = \{\Psi_{a,b} \mid (a,b) \in S\}$, une famille discrète d'ondelettes. Nous disons que B_Ψ forme un frame, s'il existe $A > 0$ et $B < \infty$ tel que pour tout $f \in L^2(\mathbb{R})$ une famille discrète d'ondelettes. Nous disons que B_Ψ forme un frame, s'il existe $A > 0$ et $B < \infty$ tel que pour tout $f \in L^2(\mathbb{R})$:

$$A \int_{-\infty}^{+\infty} |f(x)|^2 dx \leq \sum_{(a,b) \in S} |\langle \Psi_{a,b}, f \rangle|^2 \leq B \int_{-\infty}^{+\infty} |f(x)|^2 dx. \quad (2.53)$$

Où A et B sont les limites du frame.

Une famille d'ondelettes discrètes fournit une représentation complète et sans perte de toute fonction $f \in L^2(\mathbb{R})$ lorsque cette famille forme un frame. B_Ψ est dite orthogonale si pour toutes $\Psi_i, \Psi_j \in B_\Psi$:

$$\langle \Psi_i, \Psi_j \rangle = \delta_{i,j} = \begin{cases} 1 & , \text{ si } i = j \\ 0 & , \text{ si } i \neq j \end{cases}. \quad (2.54)$$

Un frame est dit base si pour toute $f \in L^2(\mathbb{R})$ la combinaison linéaire $f(t) = \sum_i w_i \Psi_i(t)$ soit unique. Une famille d'ondelettes qui est à la fois orthogonale

et base, se nomme base orthogonale. En général, un frame n'est pas une base orthogonale sauf si $A = B = 1$. Aussi, il élabore une représentation redondante de la fonction f . Le rapport (A/B) est appelé facteur de redondance et pour les autres valeurs de A et B , cette représentation reste valable, elle n'est plus une base orthogonale mais une base dite biorthogonale. De même si la représentation de combinaison linéaire d'ondelettes n'est plus unique, la famille B_Ψ est un frame. Dans ces derniers cas nous sommes menés à écrire f en fonction de frame duale :

$$\tilde{B}_\Psi = \left\{ \tilde{\Psi}_{a,b} \mid (a,b) \in S \right\}. \quad (2.55)$$

$$f(t) = \sum_{(a,b) \in S} \langle \tilde{\Psi}_{a,b}, f \rangle \Psi_{a,b}(t) = \sum_{(a,b) \in S} \langle \Psi_{a,b}, f \rangle \tilde{\Psi}_{a,b}(t). \quad (2.56)$$

Lorsque la fonction Ψ est l'ondelette analysante, les coefficients d'ondelettes sont élaborés par le calcul du produit scalaire de cette ondelette translatée et dilatée et la fonction signal à analyser.

L'ondelette duale est utilisée pour la reconstruction. Une ondelette est égale à sa duale lorsque cette ondelette appartient à une famille d'ondelettes orthogonales de même à l'aide de projection orthogonale du signal f à analyser sur la base orthogonale des ondelettes analysantes nous pouvons calculer les coefficients d'ondelettes w_i . La famille duale des ondelettes Bêta peut être calculée par l'expression suivante :

$$\tilde{\Psi}_i = \sum_{j=1}^N (\Psi_{i,j})^{-1} \Psi_j. \quad (2.57)$$

Où $\Psi_{i,j} = \langle \Psi_i, \Psi_j \rangle$ Nous pouvons présenter les trois bases possibles qui peuvent être reconstruites avec une famille d'ondelettes représentées par des vecteurs.

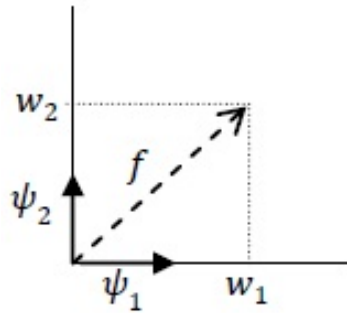


FIGURE 2.18 – Base orthogonal

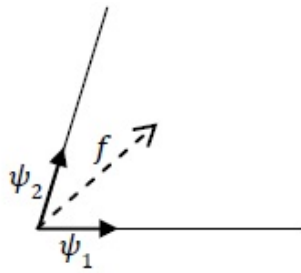


FIGURE 2.19 – Base biorthogonal

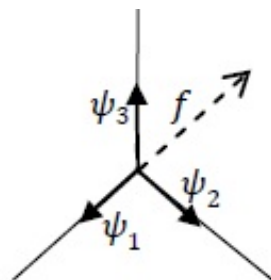


FIGURE 2.20 – Frame orthogonal

2.5.8 Calcul direct des pondérations (poids) de connexion

Le calcul des poids de connexion à chaque étape peut être assuré par projection du signal à analyser sur la même famille d'ondelettes. Ce calcul est possible pour les ondelettes qui sont orthogonales (figure 2.6). Mais pour les autres types de familles d'ondelettes il est impossible de calculer les poids de connexion par projection directe de la fonction f sur la même famille. Pour résoudre ce problème il y a recours à des ondelettes duales, donc pour calculer directement les poids de connexion, la technique utilisée se base sur la famille duale des fonctions du réseau. Cette approche présente des avantages du point de vue implémentation informatique et temps de calcul. Si pour toutes i et j nous avons : $\langle \Psi_i, \tilde{\Psi}_j \rangle = \delta_{ij}$, nous pouvons dire que les deux familles Ψ_i et $\tilde{\Psi}_i$ sont biorthogonales. L'ondelette $\tilde{\Psi}$ est dite duale alors que la fonction d'ondelette Ψ est dite primale. La famille Ψ forme une base orthogonale lorsque $\Psi = \tilde{\Psi}$.

Le calcul direct des pondérations (des poids) de connexion du réseau d'ondelettes se fait à l'aide de l'application des familles des ondelettes biorthogonales. Etant donnée une fonction f pour un signal S , Ψ_i une famille d'ondelettes qui constitue un frame ou structure oblique et $\tilde{\Psi}_i$ la famille d'ondelettes duales alors il existe des pondérations (poids) w_i vérifiant l'expression $f(x) = \sum_i w_i \Psi_i$ et par suite l'exploitation de l'ondelette duale permet de calculer le poids de connexion du réseau d'ondelettes :

$$w_i = \langle f, \tilde{\Psi}_i \rangle. \quad (2.58)$$

2.6 Les méthodes de sélection d'ondelettes

La technique de sélection des ondelettes à partir d'une bibliothèque déjà construite, joue un rôle très important. Elle améliore la performance du réseau d'ondelettes du point de vue complexité et décision surtout dans le domaine de classification. En effet, la méthode de sélection permet de choisir les ondelettes les plus significatives pour construire le réseau d'ondelettes c'est-à-dire les ondelettes qui améliorent plus la performance et l'objectif du réseau. Le recours aux techniques de sélection est dû au nombre important des ondelettes se trouvant dans la bibliothèque donc la méthode de choix doit sélectionner un nombre (N_w) déterminé des ondelettes à partir d'un nombre d'élément (M_w) dans la bibliothèque. Il existe plusieurs techniques de sélection qui permettent de gérer le choix des ondelettes par exemple : la technique de de sélection par orthogonalisation.

2.6.1 La technique de choix par orthogonalisation

La procédure d'orthogonalisation permet de classer les ondelettes de la bibliothèque. Cette technique est utilisée dans plusieurs travaux [Urbani 1995] [Stoppiglia 1997][Chen 1989] [Zhang 1993].

Etant donnée une base d'apprentissage contenant N exemples. On considère B une bibliothèque d'ondelettes contenant M_Ψ ondelettes candidates ($|B| = M_\Psi$).

A chaque candidate Ψ_i est associé un vecteur dont les composantes sont les valeurs de l'ondelette suivant les N exemples se trouvant dans la base d'apprentissage. Nous construisons ainsi une structure de données S comme étant

une matrice dont l'expression est :

$$S = \begin{bmatrix} \Psi_1(x_1) & \Psi_2(x_1) & \dots & \Psi_{M_\Psi}(x_1) \\ \Psi_1(x_2) & \Psi_2(x_2) & \dots & \Psi_{M_\Psi}(x_2) \\ \Psi_1(x_3) & \Psi_2(x_3) & \dots & \Psi_{M_\Psi}(x_3) \\ \Psi_1(x_4) & \Psi_2(x_4) & \dots & \Psi_{M_\Psi}(x_4) \\ \dots & \dots & \dots & \dots \\ \Psi_1(x_N) & \Psi_2(x_N) & \dots & \Psi_{M_\Psi}(x_N) \end{bmatrix}. \quad (2.59)$$

Nous pouvons écrire S autrement :

$$S = (s_1 s_2 \dots s_{M_\Psi}). \quad (2.60)$$

Où :

$$s_i = (\Psi_i(x_1) \Psi_i(x_2) \Psi_i(x_3) \dots \Psi_i(x_N))^T. \quad (2.61)$$

Avec $i = 1 \dots M_\Psi$

Nous avons $N \gg M_\Psi$ donc s_i sont des vecteurs généralement linéairement indépendants et non orthogonaux et par la suite nous pouvons associer aux vecteurs s_i un sous-espace vectoriel de dimension (taille) M_Ψ et nous estimons que ces vecteurs sont suffisants pour justifier la sortie du réseau à modéliser. En d'autres termes, la projection du vecteur des sorties dans cette espace correspond à une modélisation satisfaisante. La technique de choix par orthogonalisation permet de classer les entrées par ordre de significativité décroissante. Donc l'ondelette qui possède la plus grande projection sur la partie du vecteur des sorties et qui n'est pas expliquée par les entrées précédemment classées, sera sélectionnée. Nous pouvons assimiler cette procédure à une interprétation géométrique (figure 2.21).

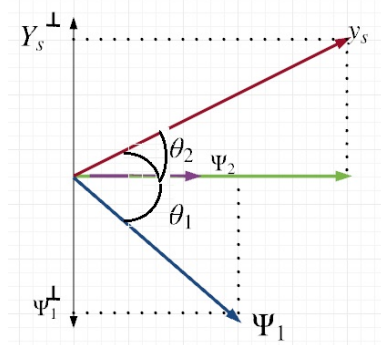


FIGURE 2.21 – Interprétation géométrique du choix (sélection) des ondelettes par orthogonalisation

Selon cette interprétation géométrique nous constatons que l'ondelette Ψ_2 est plus pertinente et significative que la fonction d'ondelette Ψ_1 ; en effet, l'ondelette Ψ_2 explique et justifie mieux la sortie Y_s que Ψ_1 selon la figure 2.21 nous constatons que $\theta_1 > \theta_2$. L'ondelette est donc ordonnée en premier rang selon le principe de cette méthode de sélection. L'élimination ou la suppression de la partie de Y_s expliquée par Ψ_2 , se déclenche par la projection Y_s (sortie du réseau) et les vecteurs correspondants aux fonctions d'ondelettes non ordonnées ; par exemple Ψ_1 dans l'espace orthogonal au vecteur que le nous venons de classer par exemple Ψ_2 . Ces projections sont représentées par : Ψ_1^\perp et Ψ_s^\perp .

2.6.2 La méthode des moindres carrées orthogonales (OLS)

La méthode des moindres carrées orthogonales(OLS) est une technique qui a été proposée par Chen et al[Chen 1991]. Cette approche est considérée comme étant un algorithme incrémental. Au cours de l'utilisation de cette technique un centre unique est désigné pendant chaque étape. De même les paramètres de couche cachée sont supposés fixes. La méthode des moindres carrées orthogonales sélectionne le centre afin de maximiser un pas vers la sortie désirée. La couche cachée convertit l'espace d'entrée sous forme d'un autre nouvel

Chapitre 2 : Des Réseaux de neurones vers les Réseaux d'ondelettes : Théorie et construction

espace de cardinalité N qui indique l'effectif des vecteurs de la base d'entraînement (apprentissage). Le nombre N indique aussi le nombre maximal de nuds cachés, donc l'algorithme a pour objectif de rechercher un sous-espace convenable de la fonction à approcher en appliquant la technique des moindres carrés orthogonaux. L'idée fondamentale de cette méthode est la détermination de l'apport de chaque vecteur se trouvant dans la base d'apprentissage et il faut que ce vecteur ne soit pas recruté précédemment à l'énergie de la structure en cours.

La méthode des moindres carrés orthogonales est utilisée pour sélectionner des ondelettes significatives afin de construire un réseau d'ondelettes performant du point de vue complexité et décision.

Soit un réseau d'ondelettes formé par M ondelettes donc nous pouvons établir la relation suivante :

$$\hat{y} = \sum_{j=1}^M W_{ej} \Psi_{ej}(x). \quad (2.62)$$

Où $\{s_1, s_2, \dots, s_M\}$ est un sous ensemble de $\{1, 2, \dots, N\}$ avec N est le nombre total d'ondelettes candidates.

Etant donnée une base d'apprentissage A contenant les entrées et sorties souhaitées, comme suit :

$$A = \{(x_i, y_i)\}. \quad (2.63)$$

Avec $x_i \in \mathfrak{R}^n, y_i \in \mathfrak{R}, i = 1, 2, \dots, I$

Ce modèle peut être mis sous la forme d'une équation matricielle comme suit :

$$y = \Psi w + e. \quad (2.64)$$

Où :

$$\Psi = \begin{bmatrix} \Psi_{s1}(x_1) & \Psi_{s2}(x_1) & \dots & \Psi_{sM}(x_1) \\ \Psi_{s1}(x_2) & \Psi_{s2}(x_2) & \dots & \Psi_{sM}(x_2) \\ \Psi_{s1}(x_3) & \Psi_{s2}(x_3) & \dots & \Psi_{sM}(x_3) \\ \Psi_{s1}(x_4) & \Psi_{s2}(x_4) & \dots & \Psi_{sM}(x_4) \\ \dots & \dots & \dots & \dots \\ \Psi_{s1}(x_I) & \Psi_{s2}(x_I) & \dots & \Psi_{sM}(x_I) \end{bmatrix}. \quad (2.65)$$

$$w = (w_{e1}, w_{e2}, \dots, w_{eM})^T. \quad (2.66)$$

beginequation $y=(y_1, y_2, \dots, y_k)^T$.

$$e = (e_1, e_2, \dots, e_k)^T. \quad (2.67)$$

L'objectif principal de l'approximation pour cette procédure est de réduire (minimiser) les erreurs (résidus) e_j durant la rapprochement de l'ensemble des données se trouvant dans la base de données d'apprentissage A . La minimisation se fait selon la relation suivante :

$$E = \min\left(\sum_{j=1}^k e_j^2\right). \quad (2.68)$$

La relation (2.68) est considérée comme étant la fonction de coût des moindres carrés, donc la méthode des moindres carrés orthogonales est utilisée pour sélectionner des ondelettes significatives afin de construire un réseau d'ondelettes. Ces dernières sélectionnées à partir de la bibliothèque doivent minimiser cette fonction qui calcule la somme des résidus au cours de l'approximation de données d'apprentissage. Cette minimisation exige une décomposition de la matrice Ψ , qui peut être résolue par plusieurs méthodes par exemple la méthode QR-décomposition, la méthode de Gram-Schmidt, la méthode de Householder, la methode de Given,...

Dans ce contexte nous devons essayer d'appliquer le principe de la méthode QR-décomposition afin de décomposer la matrice Ψ . Cette technique est utilisée pour résoudre le problème d'orthogonalité qui concerne les vecteurs constituant la matrice d'ondelettes Ψ . Pour simplifier les notations nous citons :

$$v_j = \{\Psi_j(x_1)\Psi_j(x_2), \dots, \Psi_j(x_I)\}. \quad (2.69)$$

Avec $j = 1, \dots, N$ Et par la suite nous pouvons écrire $\Psi = [v_{s1}v_{s2}\dots v_{sM}]$, ces vecteurs ne sont pas orthogonaux, pour résoudre ce problème nous utilisons la procédure QR-décomposition qui permet de décomposer Ψ . Cette méthode est appelée aussi factorisation QR ou décomposition QU. Elle est appliquée pour décomposer une matrice à une décomposition de la forme $A = QR$ avec Q est une matrice orthogonale c'est-à-dire $Q^T Q = I$ et R est une matrice triangulaire supérieure.

La décomposition QR :

Théorème : Soit A une matrice quelconque de $M_{n,m}(\mathbb{C})$.

- $\exists Q \in M_{m,m}$ une matrice orthogonale.
- $\exists R \in M_{n,m}(\mathbb{C})$ une matrice triangulaire supérieure telles que $A = QR$

Selon ce principe nous devons décomposer Ψ comme suit :

$$\Psi = QR. \quad (2.70)$$

Où Q est une matrice orthogonale de taille $I \times M$. Cette matrice est orthogonale et par la suite contenant des vecteurs colonnes orthogonaux et la structure de donnée R est une matrice triangulaire supérieure de taille $M \times M$. D'après le principe de cette méthode nous pouvons récrire la relation (2.64)

comme suit :

$$y = Qz + e. \quad (2.71)$$

Avec $z = Rw$

D'après cette expression nous appliquons la méthode des moindres carrés orthogonales pour déterminer le vecteur z , donc ce vecteur est évalué comme suit :

$$z = (Q^T Q)^{-1} Q^T y = ((q_1 \dots q_M)^T (q_1 \dots q_M))^{-1} (q_1 \dots q_M)^T y. \quad (2.72)$$

Où q_i vecteur qui appartient à la matrice Q qui constitue des vecteurs orthogonaux. Cette solution est considérée comme étant la meilleure solution pour faire une approximation qui minimise les erreurs.

D'après la relation (2.72) nous pouvons écrire :

$$y^T y = \sum_{j=1}^M z_j^2 q_j^T q_j + e^T e. \quad (2.73)$$

Les deux relations (2.71) et (2.73) indiquant que chaque ondelette sélectionnée à partir de la bibliothèque des ondelettes candidate, devrait maximiser le terme $t = z_j^2 q_j^T q_j$ et minimiser $e^T e$ afin de construire un réseau d'ondelettes contenant M ondelettes sélectionnées.

La procédure QR-décomposition est une méthode très utilisée surtout dans la résolution du système surdéterminé. Elle permet aussi de calculer les valeurs propres et des moindres carrés, mais cette procédure a une convergence lente, chaque itération demande une décomposition QR qui n'étant pas unique et aussi le coût de traitement des matrices est relativement élevé qui dépend de la dimension de ces matrices, il peut influencer la complexité du réseau d'ondelettes élaborée.

2.7 Conclusion

Ce chapitre nous a permis de présenter la théorie et la construction des réseaux de neurones et des réseaux d'ondelettes.

Au début, nous avons abordé dans cette section les principales architectures de réseaux de neurones que l'on retrouve dans la littérature. Dans cette partie nous avons défini les réseaux de neurones et leurs applications. Ensuite, on a dressé le principe de fonctionnement des neurones. Nous avons présenté également quelques exemples d'architecture des réseaux des neurones. Et aussi nous avons défini les réseaux d'ondelettes comme étant un réseau neurone qui utilise les ondelettes comme des fonctions de transferts dans chaque neurone et nous avons cité aussi les analyses qui concernent le Fourier et l'ondelette. Puis, nous avons précisé les différentes architectures de ce type de réseau.

Enfin, nous avons cité la différence entre les réseaux d'ondelettes et les réseaux de neurones. Dans le chapitre suivant nous présenterons l'approche proposée qui permettra de classifier les séquences d'ADN comme étant un support de l'information génétique.

Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes



Sommaire

3.1	Introduction	109
3.2	Proposition d'une procédure de construction et d'apprentissage d'un réseau d'ondelettes pour classifier les séquences d'ADN	110
3.2.1	Conversion d'une séquence d'ADN	113
3.2.2	L'optimisation des réseaux d'ondelettes	117
3.2.3	Le principe de l'algorithme à suivre pour construire notre réseau d'ondelettes	120
3.2.4	Construction de la bibliothèque d'ondelettes	121
3.2.5	La méthode de sélection des ondelettes pour notre approche	122
3.2.6	Calcul des poids en utilisant la méthode Lasso	124
3.2.7	La procédure d'apprentissage	129
3.2.8	La phase de reconnaissance	134
3.3	Conclusion	138

3.1 Introduction

Ce chapitre présente l'approche proposée pour regrouper un échantillon contenant des séquences d'ADN. Au début, nous présenterons la méthode à appliquer pour codifier et traiter une séquence d'ADN. Ensuite, nous citerons une approche pour convertir la séquence codifiée en signal.

Enfin, nous présenterons l'architecture et la construction de réseaux d'ondelettes à appliquer pour classifier les brins d'ADN.

3.2 Proposition d'une procédure de construction et d'apprentissage d'un réseau d'ondelettes pour classifier les séquences d'ADN

Dans cette partie, nous proposons une approche qui permet de classifier les séquences d'ADN. Il s'agit d'une approche de classification non supervisée inspirée d'un réseau d'ondelettes bêta pour grouper et analyser les séquences d'ADN d'une manière automatique.

Dans le cadre de notre travail, nous allons essayer de présenter deux sections afin de développer notre approche. Au début, la première section précisera notamment des traitements qui concernent les séquences d'ADN à classifier. Dans cette partie nous présenterons la méthode à appliquer pour convertir une séquence d'ADN contenant une suite de lettre qui précise les acides nucléotides (A, T, C, G). Ensuite, cette séquence codifiée devra être transformée en signal en utilisant la Transformée de Fourier (TF) et enfin, nous appliquerons le principe de densité spectrale de puissance(DSP) pour obtenir un signal convenable qui devra être utilisé pour alimenter notre réseau d'ondelettes comme étant des entrées pour le modèle à développer.

Le développement de notre méthode se base sur les structures primaires des séquences d'ADN (une suite de caractères de longueur variable prise dans un alphabet de 4 caractères qui représentent des acides nucléiques) et sur les réseaux d'ondelettes.

Au début, chaque séquence d'ADN est convertie en utilisant une codification binaire qui permet de représenter chaque nucléotide par une suite (séquence) d'indicateur binaire. Ensuite, la transformée de Fourier discrète est appliquée sur chaque suite pour obtenir des spectres de puissance respectifs. Ces spectres sont utilisés pour la construction des moments mathématiques qui

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

sont utilisés pour construire des vecteurs multidimensionnels de nombre réel. La nouveauté de ce procédé est que des séquences de longueurs différentes peuvent être comparées facilement grâce à l'utilisation des spectres de puissance et les moments périodiques.

Enfin, la deuxième section présente la procédure à suivre pour développer et modéliser notre réseau d'ondelettes. Dans cette partie, nous proposons une procédure de construction des réseaux d'ondelettes dont les paramètres (translation et dilatation) sont à valeurs discrètes. En appliquant alors la transformée en ondelettes discrètes.

Au cours de cette section, nous précisons donc la stratégie à appliquer pour construire l'architecture de ce réseau. Cette stratégie a pour objectif de respecter et conserver des optimisations précédentes qui consistent à réduire le nombre d'ondelettes sur la couche cachée du réseau. Nous essayerons d'améliorer la performance de notre réseau d'ondelettes du point de vue complexité et décision qui consiste à classifier un échantillon contenant les séquences d'ADN.

Pour construire la topologie du réseau d'ondelettes nous appliquerons une approche ascendante (incrémentale). En effet, la construction de notre réseau à l'aide de cette approche est obtenue par ajout d'ondelettes et connexion au sein de la couche cachée. L'ajout se fait lorsque le réseau utilisé produit des erreurs. Dans cette approche, la structure initiale comporte un nombre assez réduit de neurones, généralement une couche d'entrée et une couche de sortie. Donc notre problème se concentre sur la construction de la couche cachée à l'aide des ondelettes convenables qui peuvent améliorer la performance de notre réseau. Pour résoudre ce problème, au début, nous avons construit une bibliothèque contenant un nombre fini d'ondelettes qui sont soumises à la procédure de sélection. Ensuite, nous avons utilisé un estimateur pour sélectionner à chaque étape la fonction d'ondelette pertinente à rejoindre le réseau. Cet estimateur est appelé les moindres carrés tronqués (Least

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

Trimmed Squares : LTS).

Pour évaluer les coefficients (poids) de notre réseau nous avons appliqué une approche connue sous le nom "lasso" (least absolute shrinkage and selection operator) qui consiste à maximiser la vraisemblance pénalisée par la norme L1 des coefficients de régression [Tibshirani 1996]. Donc nous avons utilisé cette méthode pour évaluer les coefficients du réseau durant la phase d'apprentissage afin d'obtenir une topologie performante et optimale.

Dans ce qui suit, nous exposerons successivement la conversion de séquence d'ADN, le principe de l'algorithme à suivre pour construire notre réseau d'ondelettes, la construction de la bibliothèque d'ondelettes, la méthode de sélection des ondelettes dans cette bibliothèque, et les procédures d'apprentissage, de reconnaissance et de test.

La figure 3.1 résume le processus général de l'approche proposée.

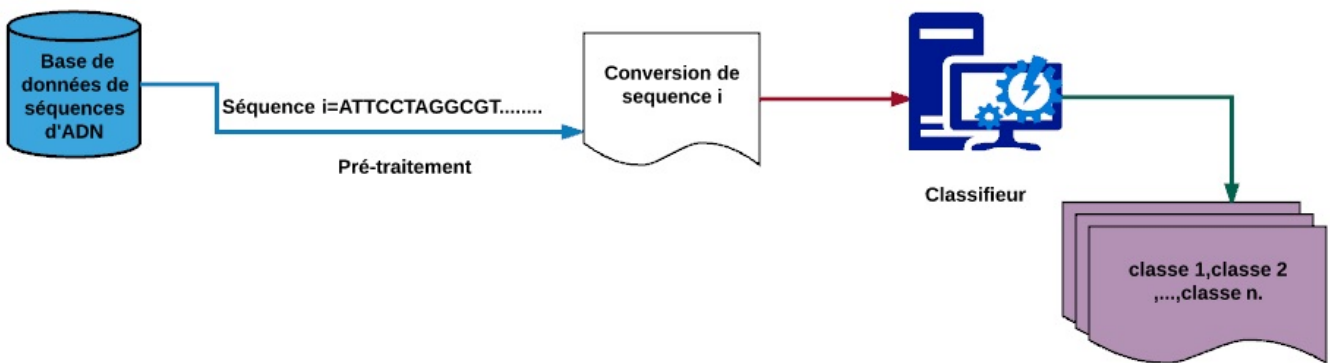


FIGURE 3.1 – Processus général de l'approche proposée

3.2.1 Conversion d'une séquence d'ADN

3.2.1.1 L'alphabet Biologie

Un alphabet $X = \{a_0, a_1, \dots, a_n\}$ est un ensemble fini de symboles distincts deux à deux. En particulier, le symbole a_0 est appelé brèche ou gap (en anglais) et est représenté par le caractère -.

3.2.1.2 L'Alphabet de l'ADN

Dans le cadre de notre travail, nous devons classifier des espèces selon leurs molécules d'ADN. Chaque espèce est identifiée par son séquence d'ADN [Shi 2012][Vinga 2003]. L'alphabet de ces molécules d'ADN est composé de 5 symboles.

$X_{ADN} = \{A, C, G, T, -\}$ qui représentent respectivement l'Adénine, la Cytosine, la Guanine, la Thymine et le gap.

La représentation d'une séquence d'ADN peut être faite par plusieurs types par exemple [b. Arniker 2012] : codification binaire (tableau 3.1), codification en utilisant EIIP (Electron-ion interaction pseudopotential) (tableau 3.2)...

L'indicateur de notre méthode translate l'information vers un format digital qui peut être utilisé pour la densité spectrale de puissance d'une séquence d'ADN qui indique 1 ou 0 pour l'existence ou l'inexistence d'un nucléotide spécifique à l'échelle d'une séquence d'ADN.

— **Codification en utilisant le nombre binaire (4-bits binaire)**

Tableau 3.1 – Codification binaire(4-bits) des nucléotides

Nucléotides	Codification par 4-bits binaire
A	1000
C	0100
G	0010
T	0001

Soit $X[n]$ la séquence d'ADN suivante :



FIGURE 3.2 – Séquence d'ADN $X[n]$

Selon la codification binaire indiquée dans le tableau 3.1, nous obtenons $Xe[n] = [100010001000000010010000101001000\dots]$.

— **Codification en utilisant l'Electron-ion interaction pseudopotential(EIIP)**

Cette méthode indique l'énergie localisée au niveau de chaque nucléotide.

Tableau 3.2 – Codification des nucléotides en utilisant EIIP

Nucléotides	EIIP
A	0.1260
C	0.1314
G	0.0806
T	0.1335

La séquence d'ADN $X[n]$ du figure 3.2 peut être codifiée en utilisant la conversion Electron-ion interaction pseudopotential(EIIP)(Tableau 3.2,) nous obtenons :

$Xe[n] = [0.12600.12600.12600.13350.08060.13350.13140.1260\dots]$.

Dans notre travail, nous devons utiliser la codification binaire pour convertir les séquences d'ADN.

Après cette codification, ce format digital (binaire) peut être manipulé à l'aide des méthodes mathématique. Par exemple nous appliquons la transformée de Fourier pour la représentation binaire obtenue. Donc nous pouvons représenter la séquence $X_e(n)$ de taille N comme suit :

$$f(x) = \sum_{n=0}^{N-1} X_e(n) e^{-\frac{j\pi n x}{N}}. \quad (3.1)$$

Avec $k = 0, 1, 2, \dots, N - 1$

Pour avoir un signal convenable nous appliquons la méthode de densité spectrale de puissance (Power Spectrum) pour les fréquences $k = 0, 1, 2, \dots, N - 1$. Nous obtenons la relation suivante :

$$PS(k) = |f(x)|^2. \quad (3.2)$$

Nous utilisons la densité spectrale de puissance (Power Spectrum) pour distinguer la composition des nucléotides au niveau d'une séquence d'ADN. La représentation graphique de $PS()$ est définie comme suit (figure 3.3) :

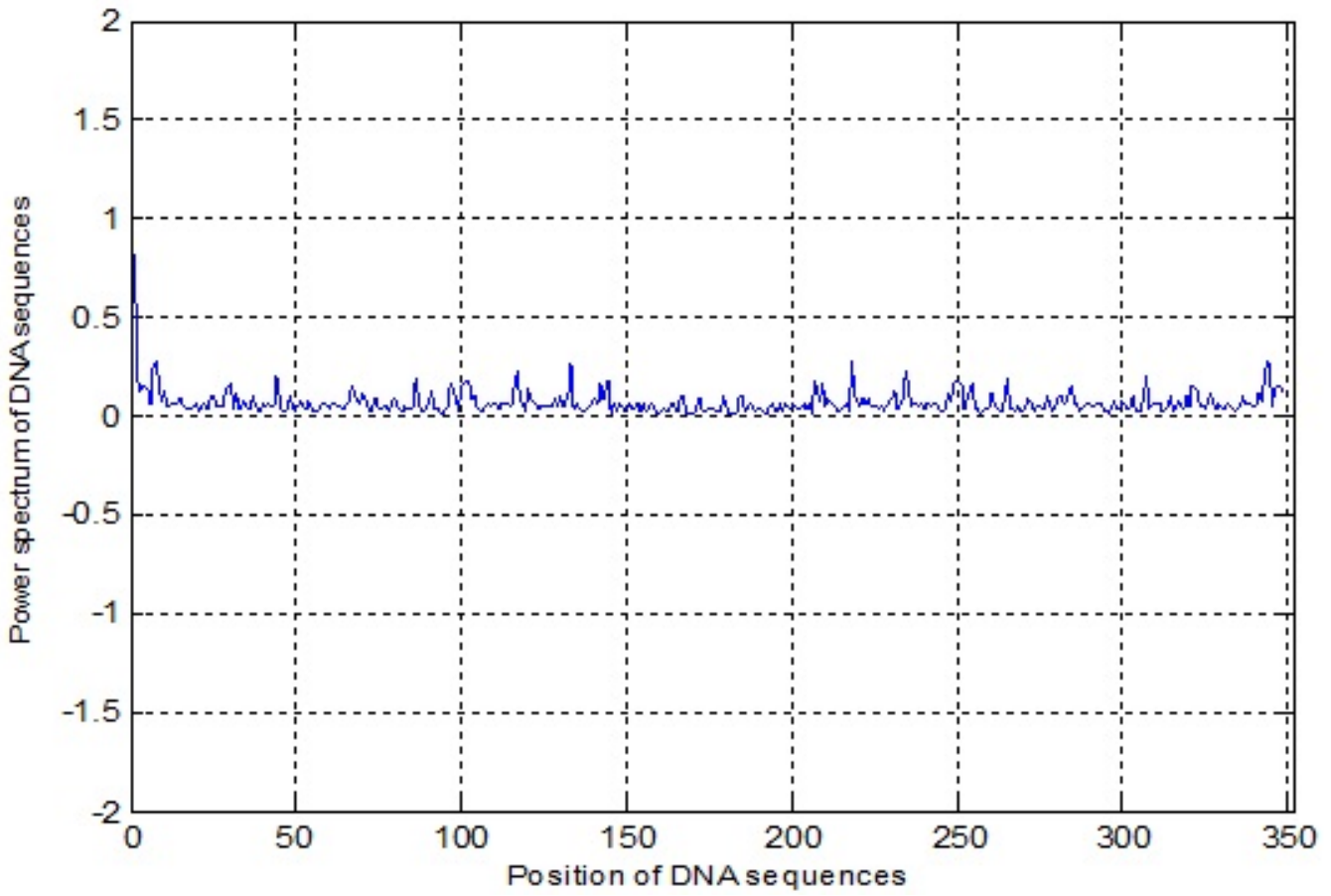


FIGURE 3.3 – Le signal d'une séquence d'ADN en utilisant la densité spectral de puissance (Power Spectrum)

3.2.2 L'optimisation des réseaux d'ondelettes

3.2.2.1 Technique d'optimisation d'un réseau d'ondelettes

Soit la fonction suivante $E = y - \hat{y}$ avec y la fonction à approcher (approximer) et \hat{y} la sortie du réseau d'ondelettes. Ces deux paramètres (y et \hat{y}) sont initialisés au début de la phase d'optimisation comme suit : $E = y$ et $\hat{y} = 0$. L'unique fonction d'ondelette analysante se trouvant dans la première échelle dyadique, la pondération (poids) de la première connexion ($i = 1$) du réseau d'ondelette se calcule de la manière suivante : $w_1 = \langle \tilde{\Psi}_1, y \rangle$ et $\hat{y} = w_1 \Psi_1$ comme étant la sortie du réseau d'ondelettes. Les fonctions d'ondelettes s'ajoutent à la couche cachée. Cette procédure s'arrête lorsque la condition d'arrêt est atteinte. Les ondelettes se trouvant dans la bibliothèque doivent vérifier la condition de l'indépendance linéaire ; les ondelettes sont linéairement indépendantes lorsqu'elles appartiennent à une famille d'ondelettes biorthogonales ou orthogonales. La notion de l'indépendance linéaire n'est pas vérifiée dans le cas d'un frame ou la structure oblique. Dans ce cas les ondelettes forment un frame avec les ondelettes du réseau qui sont déjà ajoutées. Ces ondelettes ont un rôle très important pour améliorer les pondérations (poids) du réseau d'ondelettes. La sortie du réseau est alors évaluée selon les fonctions d'ondelettes qui sont souvent linéairement indépendantes par projection du signal d'entrée sur la famille duale de toutes ces ondelettes.

3.2.2.2 Construction d'une bibliothèque pour le réseau des ondelettes

Les fonctions d'ondelettes résultant de l'échantillonnage temps-fréquence vont être utilisées pour construire la bibliothèque des ondelettes candidates qui vont être sélectionnées pour former le réseau d'ondelettes. L'échantillonnage dyadique sera appliqué pour sa simplicité. Cette technique va produire une ondelette qui a la décroissance la moins rapide au niveau de première échelle.

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

L'effectif d'ondelettes sera divisé par deux chaque fois que nous passons à l'échelle suivante. Chaque échelle contient des ondelettes qui se différencient seulement par leurs positions (paramètre de translation) et elles sont réparties sur l'axe de temps afin de couvrir la quasi-totalité du signal à analyser.

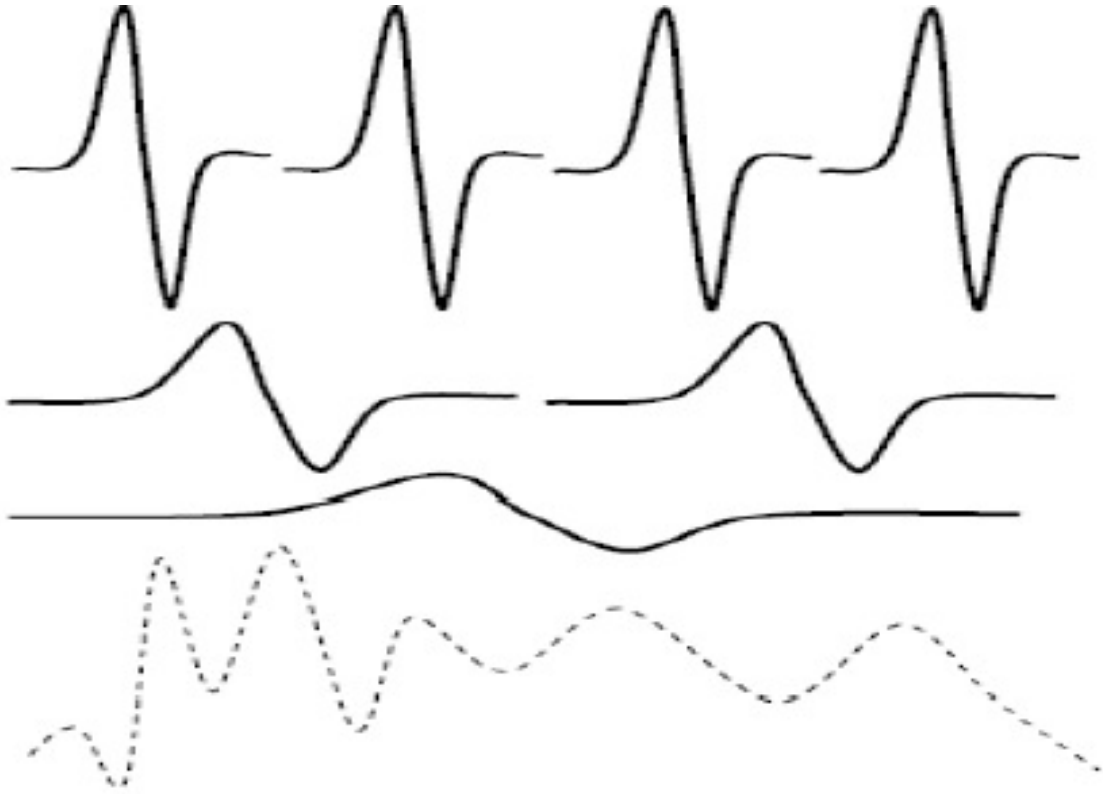


FIGURE 3.4 – Les sept premières ondelettes de la bibliothèque et un signal à analyser [Zaied 2008]

Nous appliquons la transformée inverse en ondelettes discrètes qui peut être interprétée comme la sortie d'un réseau d'ondelettes et par la suite nous pouvons construire la bibliothèque d'ondelettes. Les coefficients de dilatation (a) et de translation (b) sont évalués par les expressions suivantes :

$$a = a_0^m. \quad (3.3)$$

$$b = nb_0 a_0^m. \quad (3.4)$$

avec $a > 1$ et $b > 0$.

L'échantillonnage dyadique se fait lorsque $a_0 = 2$ et $b_0 = 1$.

L'optimisation du réseau d'ondelettes construit

L'échantillonnage dyadique sera appliqué pour sa simplicité. Cette notion permet d'optimiser le réseau d'ondelettes. Nous avons vu que l'utilisation de l'échantillonnage dyadique élabore un nombre important d'ondelettes qui se différencient uniquement par leurs positions (translations) et elles sont réparties sur l'axe de temps pour couvrir la quasi-totalité du signal à analyser ; Le nombre d'ondelettes sera multiplié par deux à chaque fois que nous passons à l'échelle suivante. Ces ondelettes jouent un rôle important pour approximer un signal ; en effet, l'approximation est acceptable avec l'utilisation des ondelettes de basses fréquences et les autres ondelettes de hautes fréquences sont nombreuses et sont utilisées pour affiner l'approximation obtenue. Le calcul du nombre d'ondelettes sur l'échelle dyadique de l'espace temps-fréquence sert à calculer la taille de la bibliothèque c'est-à-dire le nombre d'ondelettes introduites. Puisque les échelles sont prises sur des puissances de 2, nous avons besoin de $j = \log_2(N)$ échelles pour couvrir tout le signal, avec N la taille du signal à approximer. Le nombre d'ondelettes à chaque échelle m donnée est égal à 2^{j-m} ondelettes translitées.

Le nombre de fonction d'ondelette total sera évalué par l'expression suivant :

$$1 + 2 + 2^2 + 2^3 + \dots + 2^{j-1} + 1 = \frac{(1-2^j)}{(1-2)} + 1 = (2^j - 1) + 1 = (N - 1) + 1 = N$$

fonctions.

Bien que ce nombre de fonction d'ondelettes paraisse assez important, l'effectif d'ondelettes appliquées sera beaucoup plus réduit, puisque une seule fonction d'ondelette sera suffisante pour intercaler plus qu'un échantillon du signal à analyser.

3.2.3 Le principe de l'algorithme à suivre pour construire notre réseau d'ondelettes

L'objectif général de cet algorithme est de minimiser le nombre de neurones dans le réseau durant la phase d'apprentissage. Au début, la construction démarre par un seul neurone sur la couche cachée durant le processus d'entraînement. Lorsque le processus d'apprentissage se stabilise c'est-à-dire l'amélioration de l'erreur par rapport au pas précédent est inférieure à un seuil donné, il s'ajoute un nouvel neurone à la couche cachée du réseau figé et les poids du dernier neurone ajouté sont corrigés durant le recommencement de la phase d'apprentissage.

Ensuite, lorsque les améliorations du réseau ne sont pas significatives, le processus enchaîne les ajouts de neurone sur la couche cachée. Enfin, lorsque l'erreur globale du réseau a atteint le seuil désiré ou l'ajout d'un nouveau neurone n'améliore pas l'erreur la phase d'apprentissage s'achève. L'ajout d'ondelette dans la couche se fait à partir d'une bibliothèque d'ondelettes en utilisant une méthode de sélection intitulé les moindres carrés tronqués (Least Trimmed Squares : LTS). Cet estimateur devra choisir les fonctions l'ondelette Bêta convenables pour construire notre Réseau d'Ondelettes Bêta (ROBLTS). Cette topologie devra être utilisée pour classifier des séquences d'ADN.

La figure 3.5 représente la topologie du réseau d'ondelettes à concevoir.

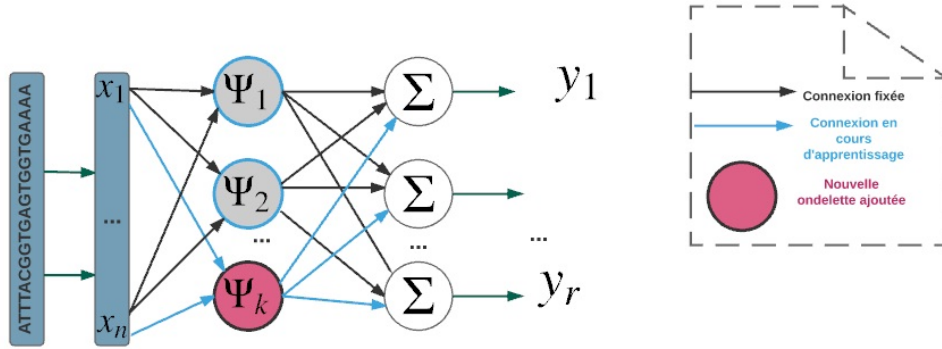


FIGURE 3.5 – Topologie du réseau d'ondelettes à concevoir pour classifier les séquences d'ADN

3.2.4 Construction de la bibliothèque d'ondelettes

Dans cette partie, nous proposons de construire une bibliothèque d'ondelettes candidates. Cette bibliothèque sert à construire notre réseau d'ondelettes (RO-BLTS). Elle comporte des versions dilatées et translatées des fonctions d'ondelettes et des fonctions d'échelle élaborées par échantillonnage sur une grille dyadique de la transformée en ondelette discrète. La construction de la bibliothèque d'ondelettes se fait à l'aide d'une famille de fonctions suivante :

$$\beta_{\Psi} = \{ \Psi(a_0^{-m}x - nb_0), \phi(a_0^{-j}x - nb_0) \}. \quad (3.5)$$

Les coefficients de dilatation (a) et de translation (b) sont évalués par les expressions suivantes selon les relations (2.58) et (2.59). Cette bibliothèque contient un nombre important d'ondelettes. Ce nombre peut influencer la performance de notre réseau d'ondelettes du point de vue complexité et décision. Dans notre approche, pour résoudre ce problème, nous suivons une stratégie mathématique afin de réduire le nombre d'ondelettes dans la bibliothèque ; en effet, une fonction g est périodique si et seulement si, il existe un réel $T > 0$ tel que : $\forall x \in D_g$, nous avons $x + T \in D_g$ et $g(x+T)=g(x)$.

Une période de la fonction g est le plus petit réel vérifiant la propriété ci-

dessus.

Si g est une fonction de période T , alors nous avons $\forall x \in D_g$ et $g(x+kT)=g(x)$ où $k \in \mathbb{R}$.

Et par suite l'intervalle d'étude d'une fonction périodique peut se réduire à un intervalle couvrant une seule période.

Sur l'échelle m de notre bibliothèque nous avons 2^{j-m} ondelettes ; les ondelettes sont adjacentes et de mêmes tailles, nous supposons qu'elles forment une seule fonction alors cette dernière est périodique et la taille de l'ondelette qui vaut $T = \frac{N}{2^{j-m}}$ où $j = \log_2(N)$ est le nombre d'échelles pour couvrir tout le signal et N est la taille du signal et nous avons $\forall 1 \leq i \leq j \leq 2^{j-m}$ pour $x_2 - x_1 = (j - i)T$ alors $\Psi_i(x_1) = \Psi_j(x_2)$ donc dans chaque échelle nous ne pouvons utiliser qu'une seule ondelette au lieu de 2^{j-m} et par la suite nous arrivons à diminuer le nombre d'ondelettes dans la bibliothèque en utilisant l'ondelette comme étant une fonction périodique.

3.2.5 La méthode de sélection des ondelettes pour notre approche

Après la construction et la réduction du nombre d'ondelette dans notre bibliothèque, nous utilisons un estimateur pour sélectionner à chaque étape la fonction d'ondelette pertinente à rejoindre le réseau. Cet estimateur est appelé les moindres carrés tronqués (Least Trimmed Squares : LTS)[Rousseeuw 1987]. Soit une régression linéaire pour p variables indépendants définie par la relation suivante :

$$y = X\beta + u. \tag{3.6}$$

Avec y un vecteur de contenant $n \times 1$ variables de réponses $y = (y_1, y_2, \dots, y_n)^T$, X une matrice de taille $n \times p$, contenant les entrés à régresser $\{x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,p}), i = 1, 2, \dots, n\}$, β un vecteur contenant des paramètres inconnus $\beta = (\beta_1, \beta_2, \dots, \beta_p)^T$ et u un vecteur constitué des erreurs

**Chapitre 3 : Approche proposée pour classifier les séquences
d'ADN par les réseaux d'ondelettes**

$u = (u_1, u_2, \dots, u_n)^T$. La construction d'un estimateur pour les paramètres inconnues β se fait à l'aide des moindres carrés (Least Squares : LS) en minimisant les erreurs u comme suit :

$$u = \min_{\beta} \sum_{i=1}^n u_i^2. \quad (3.7)$$

LTS est un estimateur qui permet de sélectionner les k meilleures observations en éliminant les $n - k$ observations restantes. Il est défini comme suit :

$$u = \min_{\beta} \sum_{i=1}^n u_i^2 \quad (3.8)$$

s/c. : $u_{(1)}^2 < u_{(2)}^2 < u_{(3)}^2 < \dots < u_{(k)}^2$.

Soit un réseau d'ondelette formé par M ondelettes donc nous pouvons établir la relation suivante :

$$\hat{y} = \sum_{j=1}^M W_{ej} \Psi_{ej}(x). \quad (3.9)$$

Où $\{s_1, s_2, \dots, s_M\}$ est un sous ensemble de $\{1, 2, \dots, N\}$ avec N est le nombre total d'ondelettes candidates.

Soit la base d'apprentissage $A = \{(x_i, y_j)\}$ avec $x_i \in \mathfrak{R}^n, y_i \in \mathfrak{R}, i = 1, 2, \dots, I$, x_i et y_i sont respectivement les entrées et les sorties élaborées par le réseau d'ondelettes. Selon la relation (3.48) nous pouvons définir l'expression (3.52) comme suit :

$$y = \Psi w + e. \quad (3.10)$$

Ce model peut être mis sous la forme d'une équation matricielle (paragraphe 2.6.2 chapitre 2).

L'objectif principal de l'approximation pour cette procédure est de réduire les erreurs (résidus) e durant le rapprochement de l'ensemble des données se trouvant dans la base de données d'apprentissage A . Nous devons utiliser l'estimateur LTS pour résoudre la minimisation en appliquant la relation

suivante :

$$E = \min\left(\sum_{j=1}^k \sum_{i=1}^I e_{ik}\right). \quad (3.11)$$

L'erreur ou le résidu au niveau de neurone de sortie k concernant l'entrée I est définie par :

$$e_{ik} = y_i - \hat{y}_{ik}. \quad (3.12)$$

Soit la base d'apprentissage A , l'approche des moindres carrés tronqués sélectionne les ondelettes qui minimisent le total des erreurs (les résidus) en utilisant la relation suivante :

$$E = \min\left(\sum_{j=1}^k \sum_{i=1}^I e_{ik}\right) \quad (3.13)$$

$$s/c. : e_{i1}^2 < e_{i2}^2 < e_{i3}^2 < \dots < e_{ik}^2.$$

Cette minimisation exige une décomposition de la matrice Ψ . Dans ce contexte nous devons essayer d'appliquer le principe de la méthode QR-décomposition afin de décomposer la matrice Ψ . Cette technique est utilisée pour résoudre le problème d'orthogonalité qui concerne les vecteurs constituant la matrice d'ondelettes Ψ (paragraphe 2.6.2 chapitre 2).

3.2.6 Calcul des poids en utilisant la méthode Lasso

Le calcul des poids de connexion à chaque étape se fait par l'application de la méthode de régression Lasso (Least Absolute Shrinkage and Selection Operator) [Tibshirani 1996]. En statistiques, la méthode lasso est une approche de contraction des coefficients de la régression développée par Robert Tibshirani [Tibshirani 1996].

La méthode s'applique dans le cas du problème où le nombre d'objets est inférieur au nombre de variables ($n < p$), si toutefois un faible nombre de ces variables a une influence sur les observations. Cette propriété n'est pas

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

vraie dans le cas de la régression linéaire classique avec un risque associé qui augmente comme la dimension de l'espace des variables même si l'hypothèse de parcimonie est vérifiée et aussi Lasso permet de choisir un sous-ensemble restreint de variables(dépendant du paramètre λ). Cette sélection restreinte permet aussi une meilleure interprétation de même elle permet une sélection consistante.

Le problème de construction de la structure du réseau d'ondelettes permet d'influencer la performance des résultats obtenus. Pour résoudre ce problème nous avons appliqué la méthode Lasso.

Au début, nous avons utilisé l'analyse temps-fréquence pour localiser les ondelettes à utiliser pour construire notre réseau. Ensuite, nous avons formulé le problème sous forme d'un problème d'optimisation sans contraintes en utilisant le principe Lasso afin de résoudre la structure et l'apprentissage pour notre réseau.

Le problème d'optimisation obtenu est résolu en utilisant la méthode des moindres carrés itératifs pondérés(MCP)(IRLS).

L'avantage de la méthode réside dans la propriété oracle du Lasso qui peut garantir la structure optimale du réseau d'ondelettes.

La construction du réseau d'ondelettes en utilisant la méthode Lasso :

Soit la base de donnée d'apprentissage $TN = \{(x^k, f(x^k))\}_{k=1}^N$, est utilisée pour ajuster les poids de connexion du réseau d'ondelettes et la sortie du

réseau peut être exprimée par la relation suivante :

$$Y = [y^{(1)}, y^{(2)}, \dots, y^{(N)}]^T = \begin{bmatrix} \Psi_1^{(1)} & \Psi_2^{(1)} & \dots & \Psi_r^{(1)} \\ \Psi_1^{(2)} & \Psi_2^{(2)} & \dots & \Psi_r^{(2)} \\ \dots & \dots & \dots & \dots \\ \Psi_1^{(N)} & \Psi_2^{(N)} & \dots & \Psi_r^{(N)} \end{bmatrix} \begin{bmatrix} c_1 \\ c_2 \\ \dots \\ c_r \end{bmatrix} = \Psi C. \quad (3.14)$$

Avec $y^k, \Psi_j^{(k)}, 1 \leq k \leq N, 1 \leq j \leq r$ sont respectivement la sortie du réseau d'ondelettes et la j^{eme} ondelette candidate utilisée pendant l'apprentissage de donnée $x(k)$.

Le problème d'architecture du réseau d'ondelettes est de sélectionner un sous-ensemble d'ondelettes à partir de r ondelettes candidates pour construire la couche cachée du réseau. Ce problème est résolu par une méthode de sélection. L'application du système d'identification du réseau d'ondelettes est de représenter le système dynamique non linéaire à travers le moins de neurones possible. Plusieurs travaux indiquent que le réseau de neurones d'ondelettes avec quelques ondelettes a une meilleure performance de point de vue complexité et décision.

Dans notre travail nous avons proposé une nouvelle méthode d'estimation basée sur le principe des moindres carrés itératifs pondérés (MCP) (IRLS). Cette méthode permet de réduire la somme résiduelle des carrés soumis à la somme de la valeur absolue des coefficients qui sont inférieurs à une constante. En plus de la nature de cette contrainte, il a tendance à créer des coefficients qui sont exactement nuls et donne donc des procédés interprétables. Nos études de simulation suggèrent que les moindres carrés itératifs pondérés bénéficient d'une partie des propriétés favorables à la fois de la sélection de sous-ensemble et de la crête de régression [Tibshirani 1996].

Sur la base de cette méthode, le problème d'architecture du réseau d'ondelettes peut être transféré sous forme d'un problème d'optimisation sans contrainte en appliquant le principe Lasso comme étant une technique de régularisation pour l'estimation simultanée et de sélection de variables.

Les estimations Lasso sont définies comme suite :

$$\hat{c}(Lasso) = \text{Min} \lambda \|C\|_0 + \frac{1}{2} \|Y - \Psi C\|_2^2. \quad (3.15)$$

Où λ est une variable de régularisation non négatif.

Le premier terme de la relation (3.15) est nommé «pénalité L_0 ». La régression régularisée L_0 permet de punir le nombre de variables non nulles dans le modèle direct.

Le deuxième terme de la relation (3.15) représente la mesure de la précision du modèle. La pénalité L_0 est utilisée pour la sélection de variables, car il punit directement le nombre de poids non nuls. Cependant, l'optimisation en cause est non convexe et discontinue, et il est donc très difficile à mettre en œuvre et ce problème d'optimisation est NP-durs, à savoir, le temps de calcul pour la résolution de cette optimisation est non polynomial. En outre, pour résoudre ce problème nous allons tenter de remplacer le problème d'optimisation (3.15) par la relaxation convexe (problème Lasso) suivante :

$$\hat{c}(Lasso) = \text{Min} \lambda \|C\|_1 + \frac{1}{2} \|Y - \Psi C\|_2^2. \quad (3.16)$$

La solution du problème d'optimisation (3.16) est égale à celle du problème d'optimisation (3,15). De plus, puisque l'objectif de (3.16) est une fonction convexe sans contrainte, nous pouvons appliquer les approches standards

pour construire un minimiseur.

L'architecture et l'apprentissage du réseau d'ondelettes sont considérés comme une recherche de la position et les valeurs des entrées non nulles à partir des entrées de r ondelettes candidates dans le vecteur C .

Supposons que Ω est l'ensemble des positions optimales des entrées non nulles et C_Ω désignent les valeurs. En même temps, nous supposons $\Gamma = \{i : \hat{c}_i \neq 0\}$ est la position des entrées non nulles de la solution du problème d'optimisation (3.16) et \hat{C}_Γ désigne les valeurs dans les positions Γ . La méthode Lasso a les propriétés d'oracle suivantes [Tibshirani 1996] :

- Détermine le sous-ensemble convenable pour le modèle : $\Gamma = \{i : \hat{c}_i \neq 0\} = \Gamma$.
- Le taux d'estimation optimal : $\sqrt{N} (\hat{C}_\Gamma - C_\Omega) \rightarrow N(0, \Sigma^*)$. Avec Σ^* est la matrice de covariance pour le modèle des sous-ensemble.

Le principe des moindres carrés itératifs pondérés (MCP)(IRLS) est utilisé pour minimiser la relation(3.16) en calculant les poids optimaux du réseau d'ondelettes. Donc l'évaluation des paramètres $C = (c_1, \dots, c_k)^T$ permet de minimiser la norme L1 concernant le problème Lasso de régression linéaire (3.16). Et par suite nous pouvons réécrire la relation (3.16) par l'expression suivante :

$$\text{Min} \lambda \|C\|_1 + \frac{1}{2} \|Y - \Psi C\|_2^2 = \text{Min} \lambda \sum_{i=1}^n |c_i| + \frac{1}{2} \sum_{i=1}^n |y_i - \Psi_i C|^2. \quad (3.17)$$

A l'étape $t+1$ l'algorithme des moindres carrés itératifs pondérés (MCP)(IRLS) affecte la pondération linéaire des moindres carrés de problème par la

relation suivante :

$$\text{Min} \lambda \|C\|_1 + \frac{1}{2} C^{(t+1)} = \text{Min} \lambda \sum_{i=1}^n |c_i| + \frac{1}{2} \sum_{i=1}^n w_i^{(t)} |y_i - \Psi c_i|^2 = (\Psi^T W^t \Psi)^{-1} \Psi^T W^t y. \quad (3.18)$$

Où W^t est la matrice diagonale des poids dont tous les composants sont initialement fixés à 1 : $w^0 = 1$ et durant la mise à jour les poids sont calculés par la relation suivante :

$$w_i^{(t)} = \frac{1}{\max(\delta, |y_i - \Psi_i C^{(t)}|)}. \quad (3.19)$$

Où δ prend une faible valeur, comme 0,0001.

L'algorithme (MCP)(IRLS)

- Etape 0 : $w^0 = 1, \varepsilon^0 = 0, t = 0$
- **Tant que**($\varepsilon^t \neq 0$) **faire**
- Etape 1 : $t = t + 1$
- Etape 2 : Calculer $C^{(t)}$ via (3.16).
- Etape 3 : $\varepsilon^{t+1} = \min(\varepsilon^{t+1}, C^t)$
- Etape 4 : Calculer $w_i^{(t)}$ via (3.19)
- **Fin Tant que**

3.2.7 La procédure d'apprentissage

La technique de classification à développer dans notre travail est basée sur les réseaux d'ondelettes multi-entrées multi-sorties (figure 3.6). Cette architecture devra être utilisée pour classifier les séquences d'ADN d'une façon non supervisée.

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

Lors de la phase d'apprentissage, notre modèle se prépare pour distinguer les différents groupes (classes) à l'aide des exemples de séquences d'ADN se trouvant dans la base d'apprentissage; en effet, notre système conçoit un modèle pour chaque séquence d'apprentissage donc chaque séquence sera connue par les poids de connexion et les ondelettes de la couche cachée.

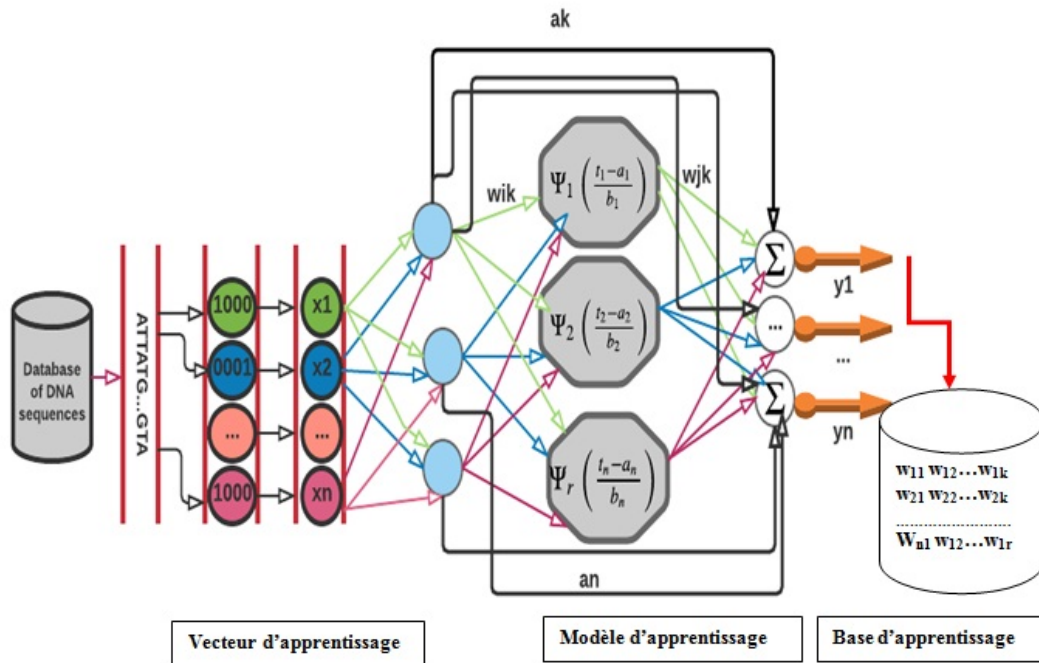


FIGURE 3.6 – Phase d'apprentissage pour notre réseau d'ondelettes multi-entrées multi-sorties

L'évaluation de la performance d'approximation de notre réseau se fait par la mesure d'Erreur Quadratique Moyenne(EQM) définie par la relation suivante :

$$EQM = \frac{1}{n * m} \sum_{i=1}^n \sum_{j=1}^m (C_1(i, j) - C_2(i, j))^2. \quad (3.20)$$

Dans cette relation C_1 et C_2 représentent respectivement les coefficients d'entrée du réseau et ceux de sortie et n et m leurs dimensions.

L'algorithme d'apprentissage pour notre approche en utilisant Lasso-LTS (figure 3.7) :

Entrées : Exemples des séquences d'ADN d'apprentissage.

Sortie : Base d'apprentissage(les poids et les connexions(les ondelettes)).

Etape 1 : Conversion des séquences d'ADN en utilisant la codification binaire et la méthode de densité spectrale de puissance(Power Spectrum).

Etape 2 : construire la bibliothèque qui contient les fonctions d'ondelettes qui devront être sélectionnées pour construire notre réseau.

1. Sélectionner l'ondelette mère qui couvre tout le support du signal à approximer.
2. Appliquer l'échantillonnage dyadique pour construire la bibliothèque.
3. Construction de la bibliothèque $\Psi = \{\Psi_i\}, i = 1, 2, 3, \dots, m$ avec $m = \|\Psi\|$ qui indique le nombre des fonctions d'ondelettes dans la bibliothèque.
4. Réduire le nombre d'ondelettes dans la bibliothèque en utilisant la notion mathématique de fonction périodique.
5. Fixer le critère d'arrêt par exemple l'erreur (e_{min}) entre le signal à approximer et la sortie du réseau ou le nombre d'ondelette maximal à sélectionner pour construire la couche cachée.

Etape 3 : Utiliser l'estimateur LTS et la méthode QR-décomposition pour sélectionner les fonctions d'ondelettes optimales afin de construire la couche cachée.

Etape 4 : Ajouter un nouvel neurone à la couche cachée du réseau fixé et les

poinds du dernier neurone ajouté qui sont corrigés durant le recommence de phase d'apprentissage en utilisant l'algorithme (MCP)(IRLS) .

Etape 5 : Calculer la sortie \hat{y}_i du réseau en utilisant la relation (3.9) et les poinds.

Etape 6 : Calculer l'erreur via(3.12)

Etape 7 : Evaluer le critère d'arrêt; si les conditions sont vérifiées alors Arrêt sinon passer pour l'étape suivante.

Etape 8 : Utiliser l'agorithme (MCP)(IRLS) afin de mettre à jour les poinds de connexion pour calculer w_{ij}^{opt} et appliquer l'algorithme de gradient pour ajuster la translation b_i et la dilatation a_i en utilisant les relations suivantes (2.47) et (2.48) afin de calculer a_{ij}^{opt} et b_{ij}^{opt}

Etape 9 : Evaluer l'Erreur Quadratique Moyenne(EQM) et si le critère d'arrêt est vérifié :

Etape 9.1 : Construire la matrice qui contient $w_{ij}^{opt}, a_{ij}^{opt} et b_{ij}^{opt}$ qui présentent les signatures des séquences d'ADN à classer ($signature_{ADN}$) et passer pour l'étape 10 .

Etape 9.2 : sinon retourner à l'étape 4.

Etape 10 : Calculer l'ensemble $V = \{v_1, v_2, , v_c\}$ qui représente les centres des signatures des séquences d'ADN se trouvant dans la matrice $signature_{ADN} = s_i = \{w_{ij}^{opt}, a_{ij}^{opt}, b_{ij}^{opt}\}$.

Etape 11 : Choisir 'c' comme étant un centre pour la classe.

Etape 12 : Evaluer les distances entre chaque signature d'ADN et le centre de classe('c').

Etape 13 : Attribuer la signature au centre du groupe dont la distance de centre de classe est minimale.

Etape 14 :Recalculer le nouveau center de classe en utilisant la relation suivante :

$$v_i = \left(\frac{1}{c_i}\right) \sum_{j=1}^{c_i} s_i. \quad (3.21)$$

Avec c_i représente le nombre des signatures dans l' i^{eme} classe.

Etape 15 :Recalculer la distance entre chaque signature de données et de nouveaux centres de groupe obtenus.

Etape 16 :Si aucune signature d'ADN n'a été réaffectée alors arrête, sinon répéter de l'étape 13.

Le figure 3.6 ,illustre l'organigramme de l'algorithme d'apprentissage.

L'organigramme de l'algorithme d'apprentissage :

L'organigramme de l'algorithme d'apprentissage présente les modules suivants :

- Construction de la bibliothèque contenant les ondelettes à sélectionner pour construire notre réseau d'ondelettes.
- Réduction du nombre d'ondelettes dans la bibliothèque.
- Sélection d'ondelettes par l'estimateur LTS et la méthode QR-décomposition pour choisir les fonctions d'ondelettes optimales afin de construire la couche cachée.
- Construction du réseau d'ondelettes.
- Calcul de différence entre les signaux de séquences d'ADN.
- Construction du réseau d'ondelettes optimal.

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

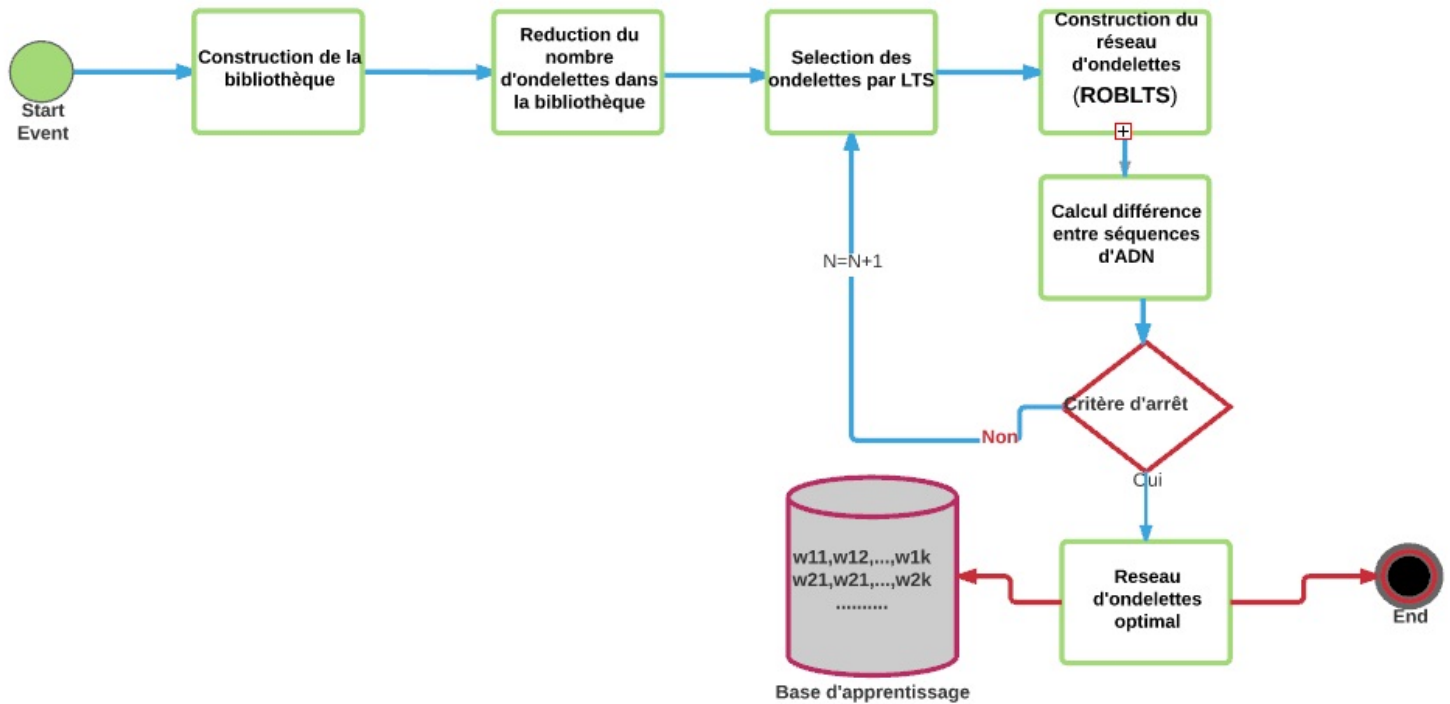


FIGURE 3.7 – Phase d'apprentissage

3.2.8 La phase de reconnaissance

Lors de la phase de reconnaissance, chaque élément de la base du test sera identifié à l'aide d'un vecteur construit par notre système.

Ce système modifiera les poids du réseau d'apprentissage jusqu'à approximer au maximum le vecteur test d'entrée (séquence d'ADN). A la fin de la procédure de reconnaissance nous enregistrerons les poids issus de chaque réseau après avoir approximer le vecteur test (figure 3.8).

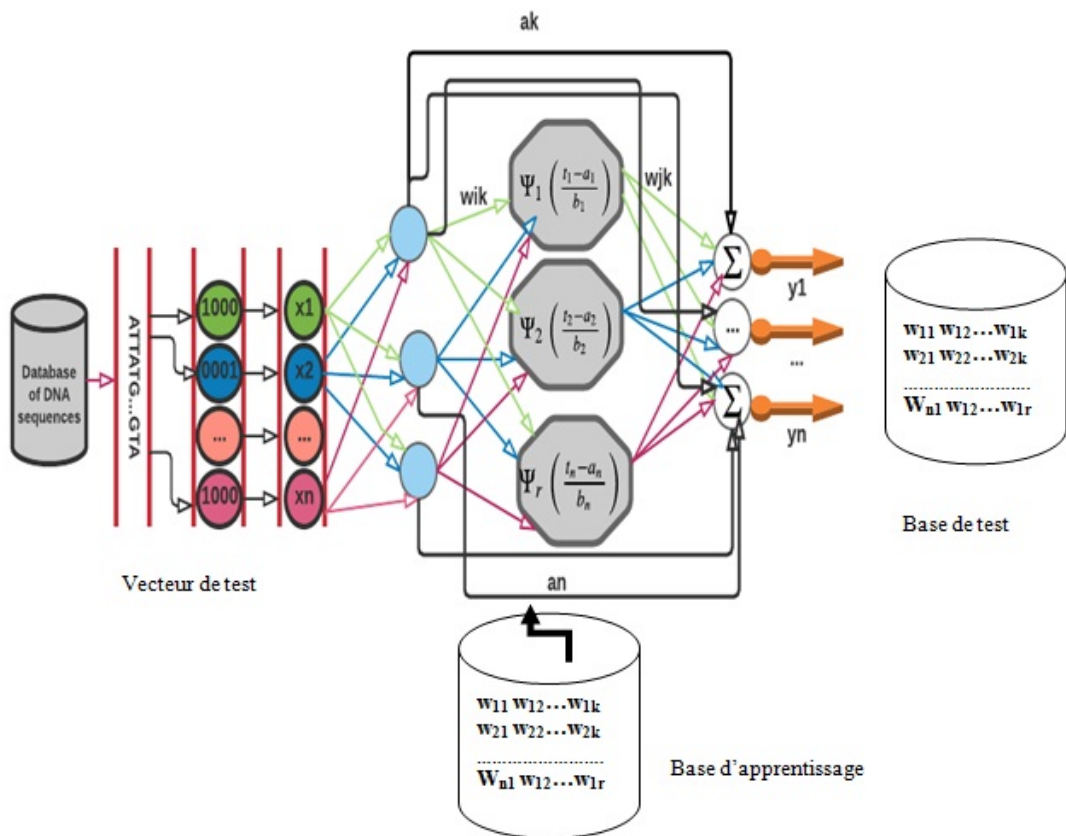


FIGURE 3.8 – Phase de reconnaissance

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

Le classement d'un vecteur test se fait à l'aide d'une mesure de distance ; en effet, nous avons calculé la distance entre les réseaux d'ondelettes d'apprentissage et les réseaux d'ondelettes de test en utilisant la matrice de poids de chaque réseau d'ondelettes.

Les différentes étapes de l'approche proposée sont présentées par la figure 3.9. La figure présente les modules à développer dans notre approche.

Elle présente les différentes contributions qui constituent les étapes suivantes :

- Codification des séquences d'ADN, qui consiste à coder les séquences d'ADN en utilisant la codification binaire
- Conversion des séquences d'ADN.
- Application de la méthode de densité spectrale de puissance (Power Spectrum) dans la classification des brins d'ADN.

Cette méthode est appliquée pour distinguer la composition des nucléotides au niveau d'une séquence d'ADN et résoudre la variété de la taille des séquences

- Application d'une approche incrémentale pour la construction du classifieur qui a été construit à partir d'un algorithme incrémental et une bibliothèque d'ondelettes

Chapitre 3 : Approche proposée pour classifier les séquences d'ADN par les réseaux d'ondelettes

L'approche proposée

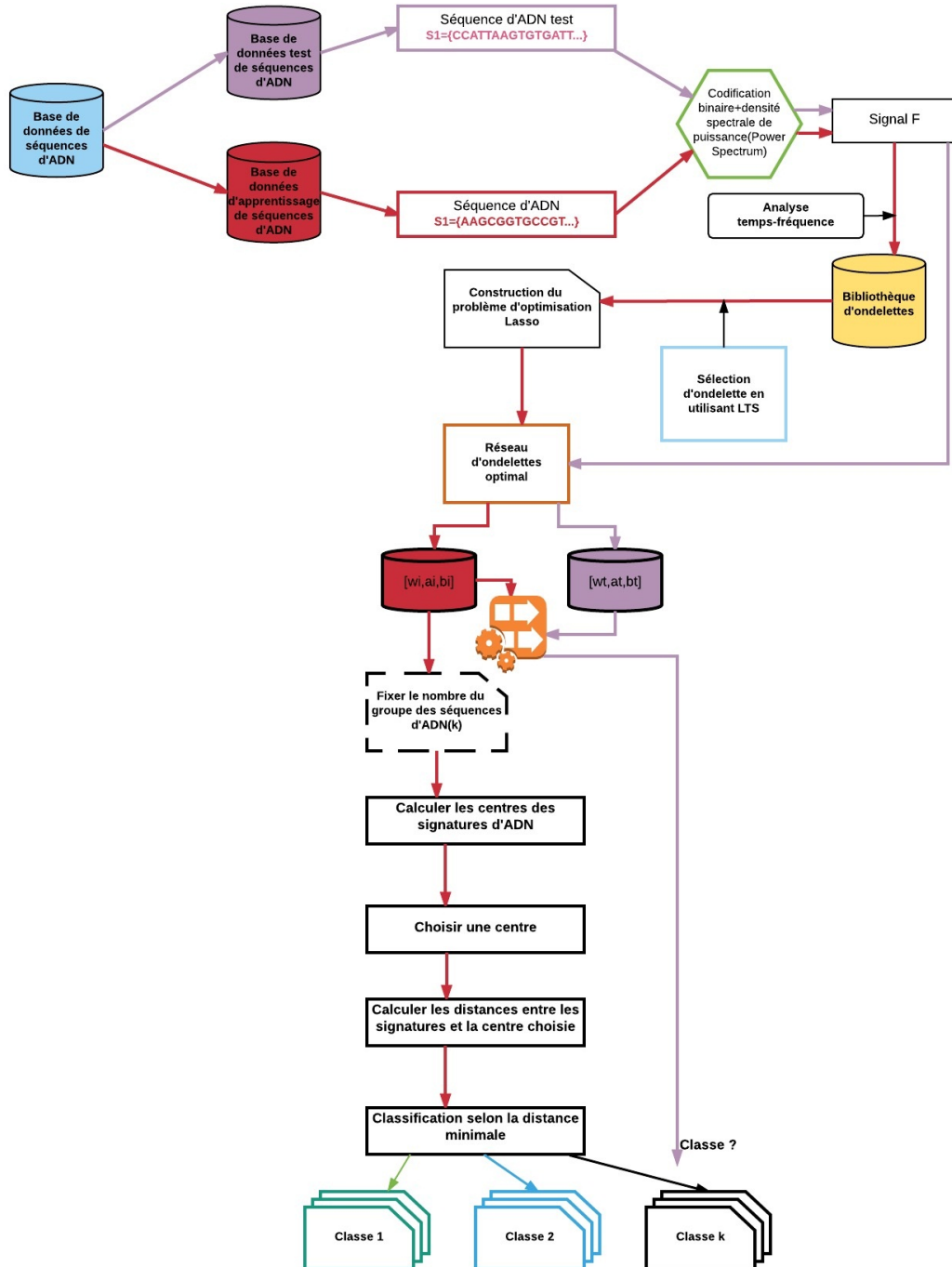


FIGURE 3.9 – L'approche proposée

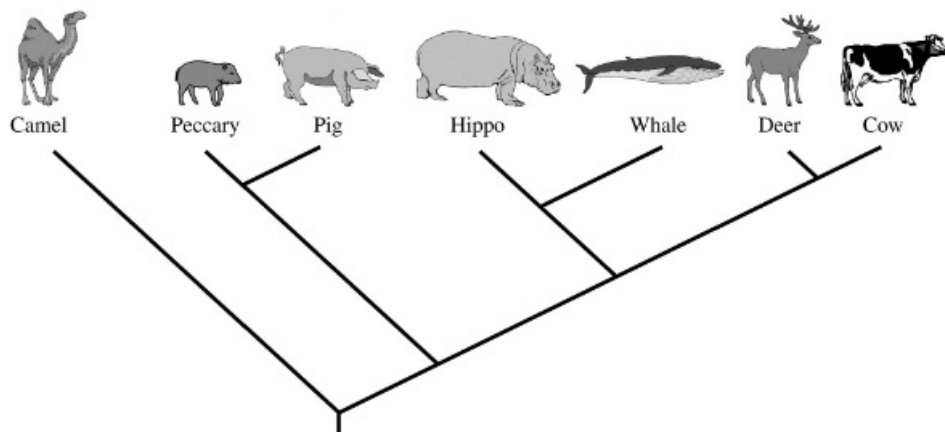
3.3 Conclusion

Au début de ce chapitre, nous avons présenté le traitement nécessaire pour codifier les séquences d'ADN et nous avons dressé la stratégie à utiliser pour construire les réseaux d'ondelettes en optimisant la performance de ces réseaux du point de vue complexité et décision.

Ensuite, nous avons présenté une nouvelle approche de classification de séquences d'ADN en utilisant un réseau d'ondelettes. Cette approche peut améliorer la structure et l'apprentissage du réseau d'ondelettes et par suite l'amélioration des performances des résultats. La construction de ce réseau a été faite à l'aide d'une nouvelle approche ascendante c'est-à-dire un algorithme incrémental, en effet ; cette construction se fait en utilisant une méthode de sélection pour choisir la fonction d'ondelettes pertinente à rejoindre le réseau afin d'améliorer l'approximation et la performance de notre réseau. De même nous avons appliqué une nouvelle méthode d'estimation basée sur le principe des moindres carrés itératifs pondérés (MCP) (IRLS) pour calculer les poids de connexion du réseau d'ondelettes.

Pour valider l'approche proposée, nous présenterons dans le chapitre suivant les différents résultats de classification de séquences d'ADN, en terme de performances biologique.

Etude Expérimentale de l'approche proposée



Sommaire

4.1	Introduction	140
4.2	Les critères d'évaluation	141
4.2.1	Précision	141
4.2.2	Rappel	142
4.2.3	F-mesure	142
4.2.4	Matrice de confusion	142
4.3	Evaluation et interprétation des résultats obtenus	144
4.3.1	Les bases de données des séquences d'ADN utilisées	144
4.3.2	Interprétation des résultats obtenus	146
4.4	Conclusion	173
	Conclusion et perspectives	174
	Publications Scientifiques	176

4.1 Introduction

Ce chapitre dresse les différents résultats qui sont utilisés pour mettre en œuvre et évaluer la performance de notre approche de classification des séquences d'ADN du point de vue complexité et décision. La classification d'ADN a été faite par plusieurs approches afin d'extraire des connaissances et des interprétations biologiques. Ces approches possèdent évidemment des particularités spécifiques ainsi que des qualités et défauts. Ainsi, plusieurs bases des séquences d'ADN ont été utilisées pour évaluer la performance de notre approche proposée. Ce dernier chapitre, présentera au début, les critères d'évaluation d'une méthode de classification. Ensuite, il dressera l'évaluation et l'interprétation des résultats expérimentaux des classifications des séquences d'ADN basés sur

le réseau d'ondelettes bêta. Enfin, ces résultats sont comparés avec d'autres élaborés par plusieurs approches.

4.2 Les critères d'évaluation

Dans cette section nous présenterons les différents critères et métriques d'évaluation. Ces techniques sont utilisées pour évaluer et mesurer la performance des classifications. Donc à l'aide de ces critères nous pouvons faire une étude comparative avec les autres approches. Parmi les critères les plus connus, nous pouvons citer : la précision, le rappel, la spécificité, la matrice de confusion,

4.2.1 Précision

La prédiction c'est l'objectif fondamental pour une approche de classification. Ce but consiste à prédire correctement le groupe ou la classe d'un nouvel élément se trouvant dans la base du test. La précision permet de calculer le nombre d'éléments correctement classés rapporté au nombre total d'éléments attribués. La précision est évaluée par la relation suivante :

$$Precision_i = \frac{a_i}{b_i}. \quad (4.1)$$

a_i c'est le nombre d'éléments correctement attribués à la classe i et b_i indique le nombre d'éléments attribués à la classe i . L'évaluation et la précision d'une approche de classification se fait par l'utilisation de deux échantillons (apprentissage et test). Le calcul du taux d'erreur ou du taux de bonne classification se fait par l'évaluation du nombre d'éléments classés dans l'échantillon de test. La matrice de confusion est considérée comme étant une bonne structure de données qui permet de reconnaître la précision et la performance de la méthode de classification utilisée (Tableau 4.1).

4.2.2 Rappel

Le rappel indique le nombre d'éléments correctement attribués rapporté au nombre d'élément appartenant à une classe donnée. Nous pouvons calculer ces métriques en utilisant une matrice de confusion multi-classes. Cette métrique est utilisée souvent dans le domaine de recherche de documents ; en effet, il est défini par le nombre d'éléments (documents) pertinents retrouvés au regard du nombre d'éléments pertinents que possède la base de données. Ce critère est utilisé aussi pour évaluer la performance de la méthode de classification. Il est défini par la relation suivante :

$$Rappel_i = \frac{a_i}{c_i}. \quad (4.2)$$

a_i est le nombre d'éléments correctement attribués à la classe i et c_i indique le nombre d'éléments appartenant à la classe i .

4.2.3 F-mesure

La f-mesure est un critère d'évaluation des algorithmes de classification. Elle est calculée à l'aide de la précision et du rappel. La f-mesure est considérée comme étant une moyenne harmonique de la précision et du rappel. Elle est définie par la relation suivante :

$$f - mesure = \frac{(1 + \beta^2) * Precision * Rappel}{((\beta^2 * Precision) + Rappel)}. \quad (4.3)$$

Avec $\beta^2 = 1$.

4.2.4 Matrice de confusion

C'est une structure de données qui sert à mesurer la qualité d'une méthode de classification. Chaque colonne de la matrice représente le nombre d'oc-

currences d'un groupe (classe) estimé, tandis que chaque ligne définit l'effectif d'occurrences d'une classe réelle (référence). Les données appliquées pour chacune de ces classes doivent être différentes. Le tableau 4.1 représente le nombre d'occurrences d'une classe estimée, tandis que chaque ligne représente le nombre d'occurrences d'une classe réelle (ou de référence). Les données utilisées pour chacun de ces groupes doivent être différentes.

Tableau 4.1 – Matrice de confusion

		Réel					
		C_1	C_2	...	C_i	...	C_n
Prédit	C_1	c_1^1	c_1^2	...	c_1^i	...	c_1^n
	C_2	c_2^1	c_2^2	...	c_2^i	...	c_2^n

	C_i	c_i^1	c_i^i	...	c_i^n

	C_n	c_n^1

Où :

- c_i^i : prédiction correcte.
- c_i^j avec $i \neq j$: prédiction incorrecte.

La précision d'une classe correspond au pourcentage théorique de chance d'obtenir une bonne classification des éléments et le rappel d'une classe c'est le rapport du nombre d'éléments bien classés au nombre d'éléments de la classe disponibles.

Le taux de bonne classification (accuracy) est défini comme étant un rapport entre le nombre d'éléments correctement groupés et l'effectif total d'éléments.

$$Precision(C_i) = \frac{c_i^i}{\sum_{j=1}^n c_i^j}. \tag{4.4}$$

$$Rappel(C_i) = \frac{C_i^i}{\sum_{j=1}^n C_j^i}. \quad (4.5)$$

$$Accuracy = \frac{\sum_{i=1}^n C_i^i}{\sum_{i,j=1}^n C_i^j}. \quad (4.6)$$

4.3 Evaluation et interprétation des résultats obtenus

Les bases de données des séquences d'ADN utilisées :

4.3.1 Les bases de données des séquences d'ADN utilisées

La classification des séquences d'ADN est un problème fréquemment étudié et traité par les biologistes pour interpréter et analyser automatiquement les connaissances obtenues.

Ces séquences sont stockées dans des bases de données selon plusieurs formes. Dans le domaine biologique, il existe plusieurs bases de données qui contiennent l'ensemble des séquences nucléiques publiques avec leurs indications par exemple les bases de données de la banque GenBank et la banque Swiss-Prot contient des séquences protéiques expertisées. Le premier réflexe d'un biologiste qui dispose d'une séquence nouvelle d'ADN est de parcourir ces bases, afin d'y trouver les séquences similaires et de faire succéder (hériter) à la nouvelle séquence les connaissances qui leur sont associées. De même nous pouvons trouver des bibliothèques des familles des séquences d'ADN qui sont créés manuellement, pour les génomes qui sont largement traités, tels que la souris et l'humain.

Les bases de données utilisées dans le cadre de notre travail, sont des bases qui concernent :

- Les bactéries ;
- Les espèces eucaryotypes : nous trouvons des bases de données de séquences d'ADN qui concernent les organismes suivants : Plant, Porifera, Protostomias, Vertebra, Cnidaria et Fungi ;
- Des bases de données qui contiennent des genes des sequences d'ADN des espèces microbiennes (Ces bases sont : HOG100, HOG200 et HOG300) ;
- Les barcode de séquences d'ADN (code-barre génétiques) pour les espèces suivantes : Cypraeidae, Drosophila, Inga, Bats, Fishes, Birds, Fungi et Algae

Tous les organismes vivants possèdent un fragment d'ADN qui s'appelle barcode moléculaire. Ce fragment d'ADN est identique chez les organismes appartenant à la même espèce, et permet alors d'identifier l'espèce à laquelle appartient un individu en ne connaissant que la séquence de ce fragment d'ADN.

Dans le cadre de notre travail, nous avons utilisé ces bases de données qui contiennent des barcodes pour ces espèces : Bats, Fishes, Birds, Fungi,.



FIGURE 4.1 – Codes-barres d'ADN

4.3.2 Interprétation des résultats obtenus

4.3.2.1 Principe de l'approche ROCB(Réseaux d'ondelettes-Codification Binaire)

Notre approche de classification consiste à :

1. Convertir les séquences d'ADN en utilisant la codification binaire.
2. Appliquer la méthode de densité spectrale de puissance pour obtenir un signal convenable. L'utilisation de la densité spectrale de puissance (Power Spectrum) pour distinguer la composition des nucléotides au niveau d'une séquence d'ADN et résoudre la variété de la taille des séquences.
3. Créer un réseau d'ondelettes en utilisant un algorithme incrémental.
4. Approximer les signaux des séquences d'ADN en utilisant les réseaux d'ondelettes. Ces réseaux sont construits en utilisant une bibliothèque d'ondelettes. Cette approximation nous permet de construire de modèles pour chaque élément d'apprentissage. Chaque élément sera identifié par les poids de connexions et les ondelettes de la couche cachée.
5. Classifier les caractéristiques obtenues en utilisant un algorithme de classification ou en utilisant un métrique qui mesure la similarité par exemple la distance euclidienne,
6. Evaluer la classification obtenue en utilisant un critère d'évaluation par exemple la précision, le rappel...

A.Classification des séquences d'ADN des bactéries

Pour évaluer la performance de notre approche nous avons effectué plusieurs expérimentations en utilisant des bases de données contenant des séquences d'ADN pour des bactéries(bacillus-subtilis, aeropyrum-pernix, aquifex-aeolicus et buchnera-sp)[A.Dakhli 2014a]. Ces bases sont subdivisées en deux parties : une partie réservée à la construction du modèle et l'échantillon d'apprentissage et une deuxième partie destinée à l'évaluation et l'échan-

tillon des tests. Les expériences ont été faites sur des bases de taille différente (Tableau 4.2).

Tableau 4.2 – Distribution des séquences d'ADN d'une base de données des bactéries

Echantillons d'ADN	Nombre de séquences d'ADN	Apprentissage	Test
bacillus- subtilis	80	60	20
aeropyrum- pernix	1322	882	440
aquifex- aeolicus	1120	747	373
buchnera- sp	1430	954	476

Tableau 4.3 – Précision de la classification des bases de données d'ADN en utilisant RNA,SVM et RO(notre approche)

Base d'ADN	Taille	RN		SVM		RO(notre approche)	
		Nombre de classe	Précision (%)	Nombre de classe	Précision (%)	Nombre de classe	Précision (%)
bacillus-subtilis	697	4	98.4	5	92.9	8	98.98
aeropyrum-pernix	219	3	92.5	4	90.2	5	98.8
aquifex-aeolicus	432	3	96.3	2	80.4	4	97.5
buchnera-sp	907	5	89.7	4	41.7	5	87.9

Nous avons utilisé le critère de précision pour évaluer la performance de classification des séquences d'ADN. Pour classifier les séquences d'ADN nous avons appliqué les méthodes suivantes : SVM (Support Vector Machine), le réseau de neurones artificiels et le réseau d'ondelettes proposé.

Le Tableau 4.3 montre la performance de notre approche pour la classification des échantillons d'ADN des bactéries. Nous pouvons remarquer que notre approche est nettement plus efficace que les autres méthodes. L'approche proposée donne une précision de classification égale à 98.98% et un nombre de classe égale à 8 groupes qui ont une similarité forte (figure 4.3) pour la base de données bacillus-subtilis. Les résultats présentent la performance de notre approche devant les autres méthodes. En effet, elle a une précision moyenne de 95.795%. Le nombre de classe obtenu augmente la prédiction des séquences d'ADN pour notre approche.

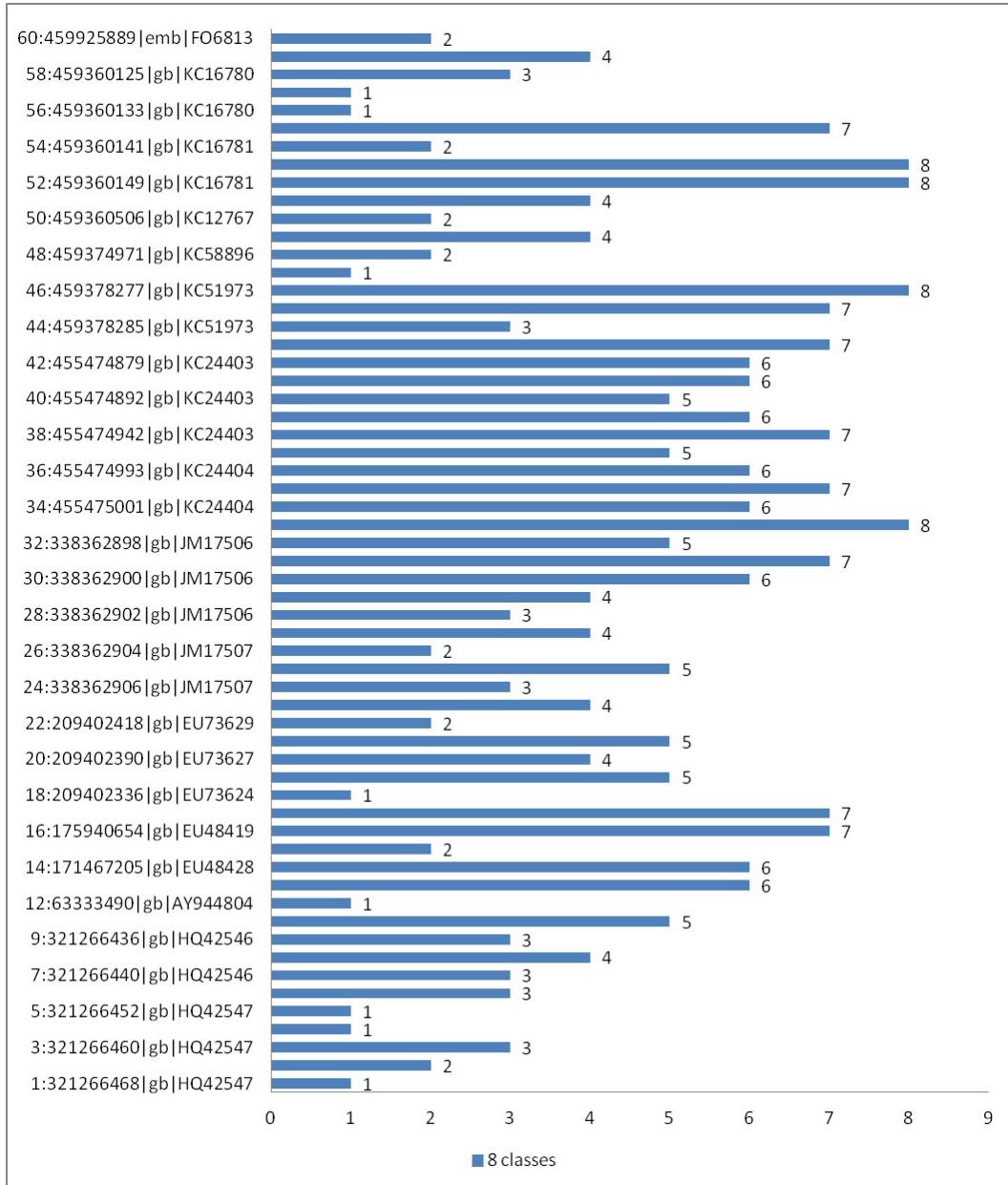


FIGURE 4.2 – Représentation des classes obtenues à partir de la base de données bacillus-subtilis

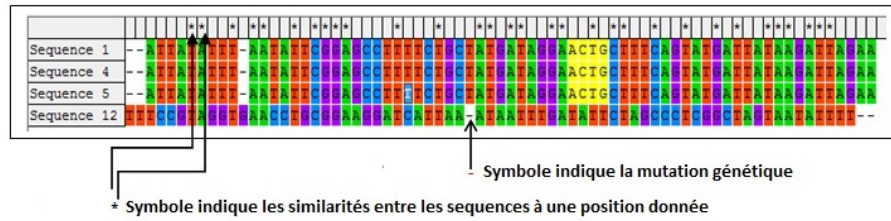


FIGURE 4.3 – Comparaison des séquences d'ADN de la classe 1 pour la base de données bacillus-subtilis

Les figures 4.2 et 4.3 montrent que les séquences d'ADN de la classe 1 ont des similarités biologiques et par la suite ces ressemblances génétiques peuvent assurer une fonction biologique similaire au niveau des cellules des organismes. De même cette similarité biologique des séquences permet de montrer un type biologiquement significatif dans la classification des séquences d'ADN.

La classification des ces espèces montre les degrés de relation biologique qui se trouve entre eux. Cette relation peut expliquer l'évolution historique des espèces dans le temps. La classe 1 qui regroupe les séquences (1, 4, 5,12,..) justifie bien la relation biologique qui existe entre les organismes identifiés par ces séquences. Donc d'après ces résultats nous sommes arrivés à extraire des interprétations et des analyses biologiques qui aident à la prise de décision. Les résultats obtenus sont donc très importants dans le domaine biologique spécifiquement dans l'étude moléculaire des espèces qui s'intéresse à expliquer et à étudier la séquence d'ADN comme étant un support de l'information génétique nécessaire dans la vie des espèces.

La figure 4.2 montre la répartition des séquences d'ADN entre les classes obtenues à partir de la base de données bacillus-subtilis (8 classes). La répartition des séquences d'ADN dans la même classe montre la forte ressemblance ou similarité entre les brins d'ADN. Cette similarité s'explique par la forte précision obtenues qui vaut 98.98%.

La ressemblance ou similarité de deux séquences d'ADN peut être expliquée en prenant comme postulat de départ que toutes les espèces sont issues d'un

même ancêtre commun. Selon cette théorie, des mutations ont eu lieu au cours de l'évolution générant des séquences qui ont donné naissance à des espèces de natures différentes(figure 4.4).

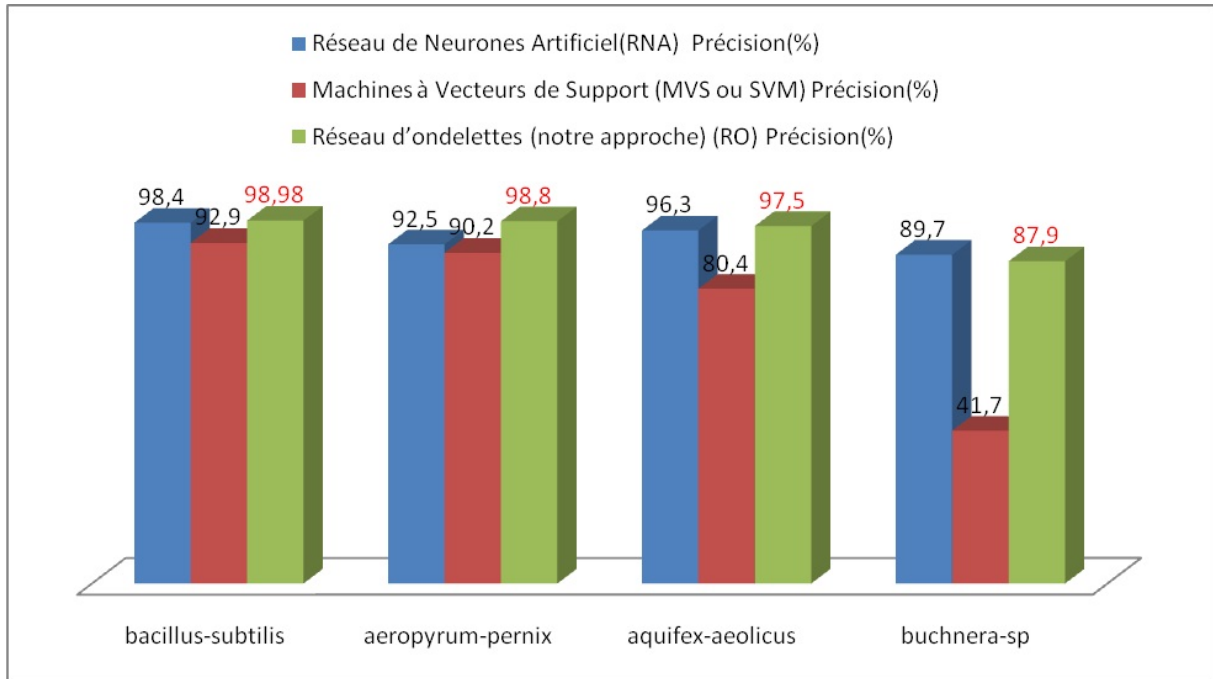


FIGURE 4.4 – Evolution de la précision pour chaque base de données d'ADN.

Tableau 4.4 – Le Temps d'apprentissage de notre approche (RO) et les méthodes RNA et SVM

Base d'ADN	Taille	RN		SVM	RO
		Temps d'apprentissage(sec.)	Temps d'apprentissage(sec.)	Temps d'apprentissage(sec.)	Temps d'apprentissage(sec.)
bacillus-subtilis	697	63.235	60.235	17.611	
aeropyrum-pernix	219	45.785	42.986	27.971	
aquifex-aeolicus	432	61.856	56.235	44.335	
buchnera-sp	907	98.985	96.542	82.805	

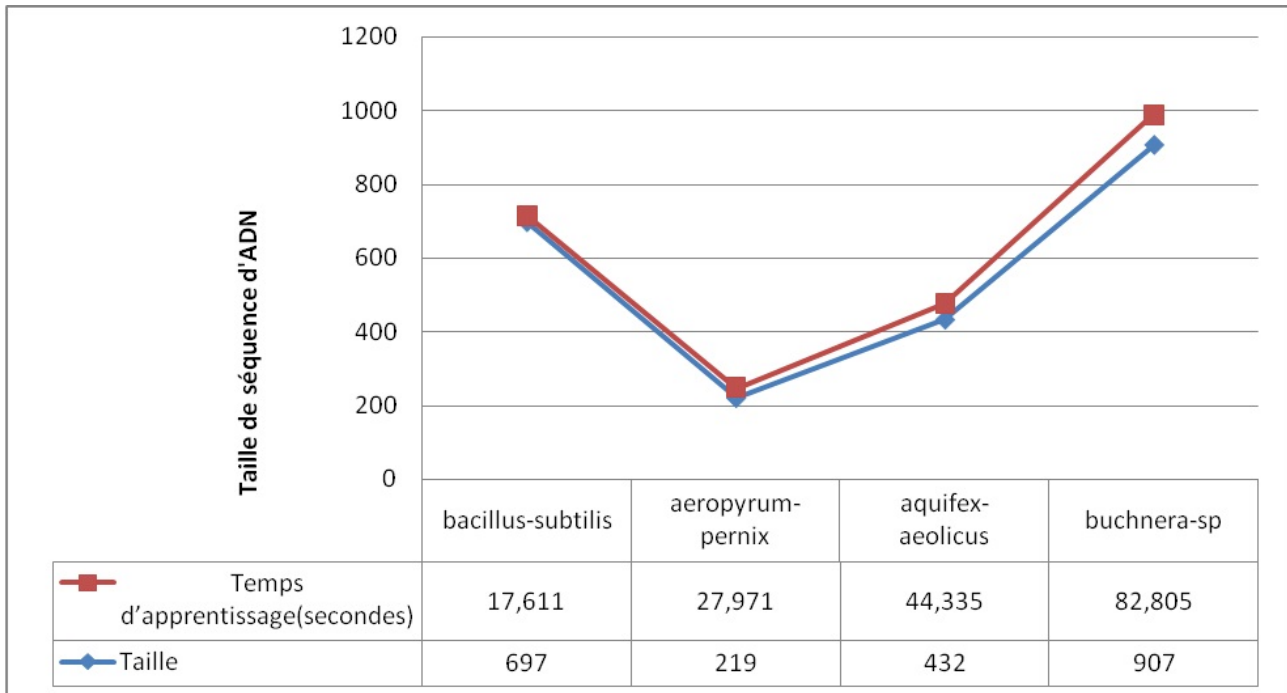


FIGURE 4.5 – Evolution du temps d'apprentissage de notre approche..

Le tableau 4.4 permet de constater que la taille des séquences d'ADN classées n'influe pas sur le calcul de précision car ce critère s'intéresse au nombre des séquences d'ADN classées ou non. Par contre la taille des séquences a une influence sur l'approximation et le temps d'exécution d'apprentissage (tableau 4.4)(Figure 4.5). Elle peut donc provoquer la performance du réseau de point de vue complexité et décision ou classification.

La figure 4.5 montre que le temps d'apprentissage augmente avec la taille des séquences d'ADN. De même ce temps varie avec la taille d'échantillon d'apprentissage.

Nous constatons aussi que le temps d'exécution d'apprentissage dépend de la taille des séquences d'ADN. Lorsque la taille est égale à 907 nucléotides le temps d'apprentissage est égal à 82.805 secondes par contre le temps d'apprentissage vaut 27.971 secondes pour une taille qui est égale à 219 nucléotides. Donc la taille des séquences d'ADN peut influencer la complexité

et la performance de l'approche utilisée.

Nous pouvons remarquer que notre méthode est nettement plus efficace aussi que les autres méthodes de point de vue temps d'apprentissage et approximation[A.Dakhli 2014a].

B.Classification des séquences d'ADN des espèces eucaryotypes

Les bases de données de séquences d'ADN des espèces eucaryotypes sont subdivisées en deux échantillons : le premier échantillon pour l'apprentissage et le deuxième pour le test (tableau 4.5).

Tableau 4.5 – Distribution des bases de données en deux échantillons (Apprentissage et test)[A.Dakhli 2014b]

Bases de données	Total	Apprentissage	Test
Plant	30	20	10
Porifera	21	14	7
Protostomia	256	171	85
Vertebrata	1024	683	341
Cnidaria	34	25	12
Fungi	52	35	17
Total	1420	948	472

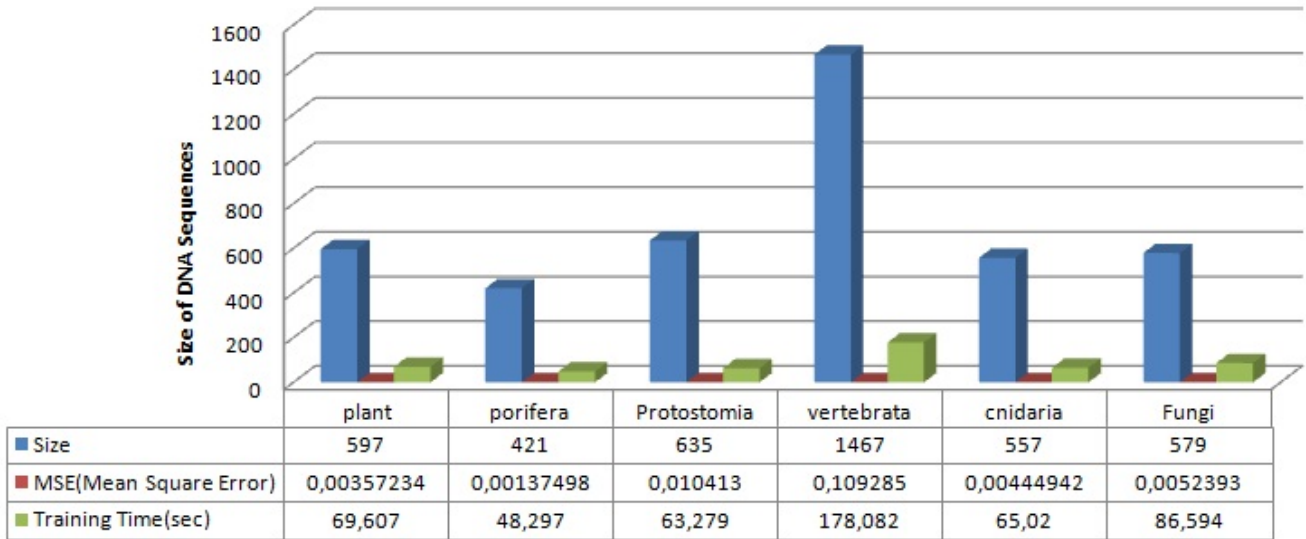


FIGURE 4.6 – Temps d'apprentissage de l'approximation des signaux de séquences d'ADN

Durant la phase d'apprentissage, notre système se prépare pour construire les modèles ou les classes des séquences d'ADN se trouvant dans la base d'apprentissage. Au cours de cette phase il y a approximation de chaque séquence d'ADN. Cette approximation se fait par une décomposition suivie par une reconstruction du signal. L'estimation de performance de cette phase se mesure par MSE (Mean Square Error). La Figure 4.6 montre une faible MSE qui vaut 0.00137498 secondes qui concerne l'approximation des séquences d'ADN de la base « porifera » de même, le resultat montre que le temps d'apprentissage dépend de la taille de séquence d'ADN ; en effet l'approximation de la base « vertebrata » qui contient la plus longue séquence d'ADN exige un temps très lent (taille=1467 nucléotides et temps d'apprentissage=178.082 secondes) tout en la comparant aux séquences se trouvant dans la base « porifera » (taille=421 nucléotides et au temps d'apprentissage=48.297 secondes)[A.Dakhli 2014b].

Tableau 4.6 – Matrice de confusion de classification des séquences d'ADN en utilisant notre réseau d'ondelettes

Classes	Classes prédites						Taux de classification (%)
	1	2	3	4	5	6	
Plant	269	0	0	0	0	4	98.666
Porifera	0	285	5	10	0	0	95
Protostomia	0	0	300	0	0	0	100
Vertebrata	0	0	5	250	5	5	93.333
Cindaria	0	15	1	2	282	0	94
Fungi	10	0	0	0	0	290	96.666
Rappel(%)	96.732	93.443	96.463	95.89	98.258	96.96	

Les données indiquées dans le tableau 4.6, permettent de constater que notre approche a réalisé un taux de bonne classification très important. En effet, les séquences d'ADN sont presque correctement bien classées (100% pour le protostomia, 98.666% pour la classe plant, 96.666% pour le fungi,). D'après ces résultats, nous avons constaté que notre approche développée a une bonne performance.

De même d'après ces résultats nous pouvons extraire des connaissances biologiques qui expliquent le phénomène de l'évolution des espèces durant leurs vies ; nous remarquons que quelques exemples des séquences d'ADN ont des similarités qui coïncident avec d'autres séquences ou espèces se trouvant dans d'autres classes.

Par exemple, nous constatons qu'il existe quatre espèces de la classe plant, classées dans la classe Fungi, de même pour la classe porifera et la classe vertebrata, Donc nous pouvons conclure la traçabilité et l'évolution des ses espèces dans l'histoire (Figure 4.7)[\[A.Dakhli 2014b\]](#).

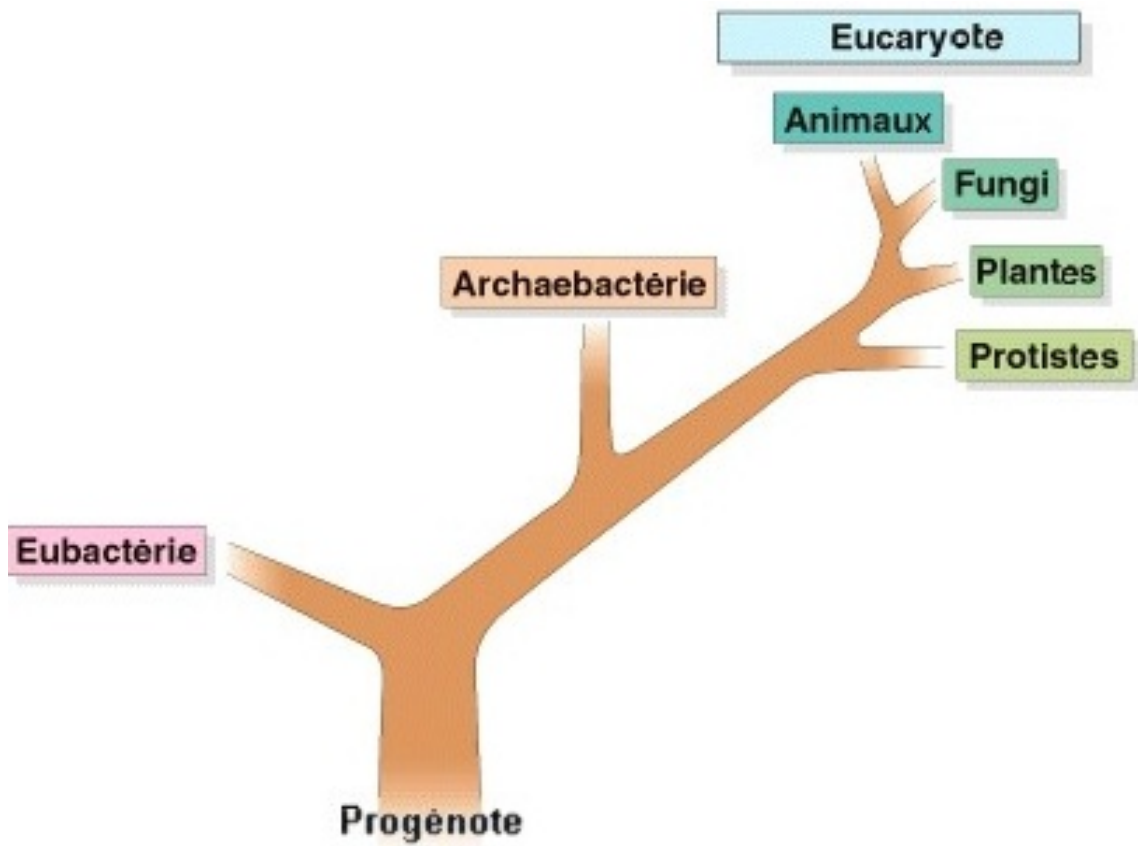


FIGURE 4.7 – Arbre phylogénique

Tableau 4.7 – Taux de classification des séquences d'ADN en utilisant le réseau de neurones probabiliste(RNP) et le réseau d'ondelettes RO (notre approche)

Bases de données	Taux de classification en utilisant RNP%	Taux de classification en utilisant RO%
Plant	73.3	98.666
Porifera	80.0	95
Protostomia	86.7	100
Vertebrata	96.8	93.333
Cnidaria	94.12	94
Fungi	65.38	96.666
Taux de classification moyens(%)	82.71	96.2775

Les Tableaux 4.6 et 4.7 présentent les taux de classification de notre approche. Les résultats obtenus montrent la performance de classification de notre approche qui s'explique par le bon taux de classification en comparaison avec le réseau de neurones qui est basé sur la probabilité.

Notre méthode a un taux de bonne classification moyenne égale à 96.2775% par contre le réseau de neurone a un taux de classification qui vaut 82.71% ; en effet, les séquences de la classe « protostomia » sont parfaitement classées. Les classes « plant », « porifera » et fungi ont aussi un bon taux de classification.

4.3.2.2 Principe de l'approche ROEIIP(Réseaux d'ondelettes-Electron-ion interaction pseudopotential)

Notre approche de classification consiste à :

1. Convertir les séquences d'ADN en utilisant EIIP (Electron-ion interaction pseudopotential). Cette méthode indique l'énergie localisée au niveau de

chaque nucléotide.

2. Appliquer la méthode de densité spectrale de puissance pour obtenir un signal convenable. L'utilisation de la densité spectrale de puissance (Power Spectrum) pour distinguer la composition des nucléotides au niveau d'une séquence d'ADN et résoudre la variété de la taille des séquences.
3. Créer un réseau d'ondelettes en utilisant un algorithme incrémental ;
4. Approximer les signaux des séquences d'ADN en utilisant les réseaux d'ondelettes. Ces réseaux sont construits en utilisant une bibliothèque d'ondelettes. Cette approximation nous permet de construire des modèles pour chaque élément d'apprentissage. Chaque élément sera identifié par les poids de connexions et les ondelettes de la couche cachée.
5. Classifier les caractéristiques obtenues en utilisant un algorithme de classification ou en utilisant un métrique qui mesure la similarité par exemple la distance euclidienne.
6. Evaluer la classification obtenue en utilisant un critère d'évaluation par exemple la précision et le rappel.

A.Classification des barcodes des séquences d'ADN

Pour évaluer la performance de notre approche, nous avons effectué plusieurs expérimentations sur des bases de données qui contiennent des barcodes de séquences d'ADN (code-barre génétiques) pour les espèces suivantes : Cypraeidae, Drosophila, Inga, Bats, Fishes, Birds, Fungi et Algae[A.Dakhli 2015]

Tableau 4.8 – MSE de l'approximation d'un signal d'une séquence d'ADN en utilisant le réseau d'ondelettes[A.Dakhli 2015][A.Dakhli 2016]

Séquences d'ADN	Taille	MSE(Mean Square Error)	Temps d'exécution (sec)
Cypraeidae	614	0.002854	69.607
Drosophila	663	0.002471	46.297
Inga	1.838	0.003041	63.279
Bats	659	0.001092	45.145
Fishes	419	0.005841	36.08
Birds	255	0.004587	22.25
Fungi	510	0.005874	38.333
Algae	1.128	0.000145	10.223

L'approximation d'un signal d'une séquence d'ADN par notre approche, consiste à décomposer et à reconstruire le signal en utilisant les fonctions d'ondelettes se trouvant dans la couche cachée.

La mesure de la performance de cette approximation se fait à l'aide de MSE. Le tableau 4.8 montre la performance de l'approche développée. En effet, les valeurs de MSE sont faibles de même, les temps d'exécution d'apprentissage sont acceptables.

Nous constatons aussi que les tailles des séquences d'ADN influencent sur le temps d'exécution pendant la phase d'apprentissage[A.Dakhli 2015].

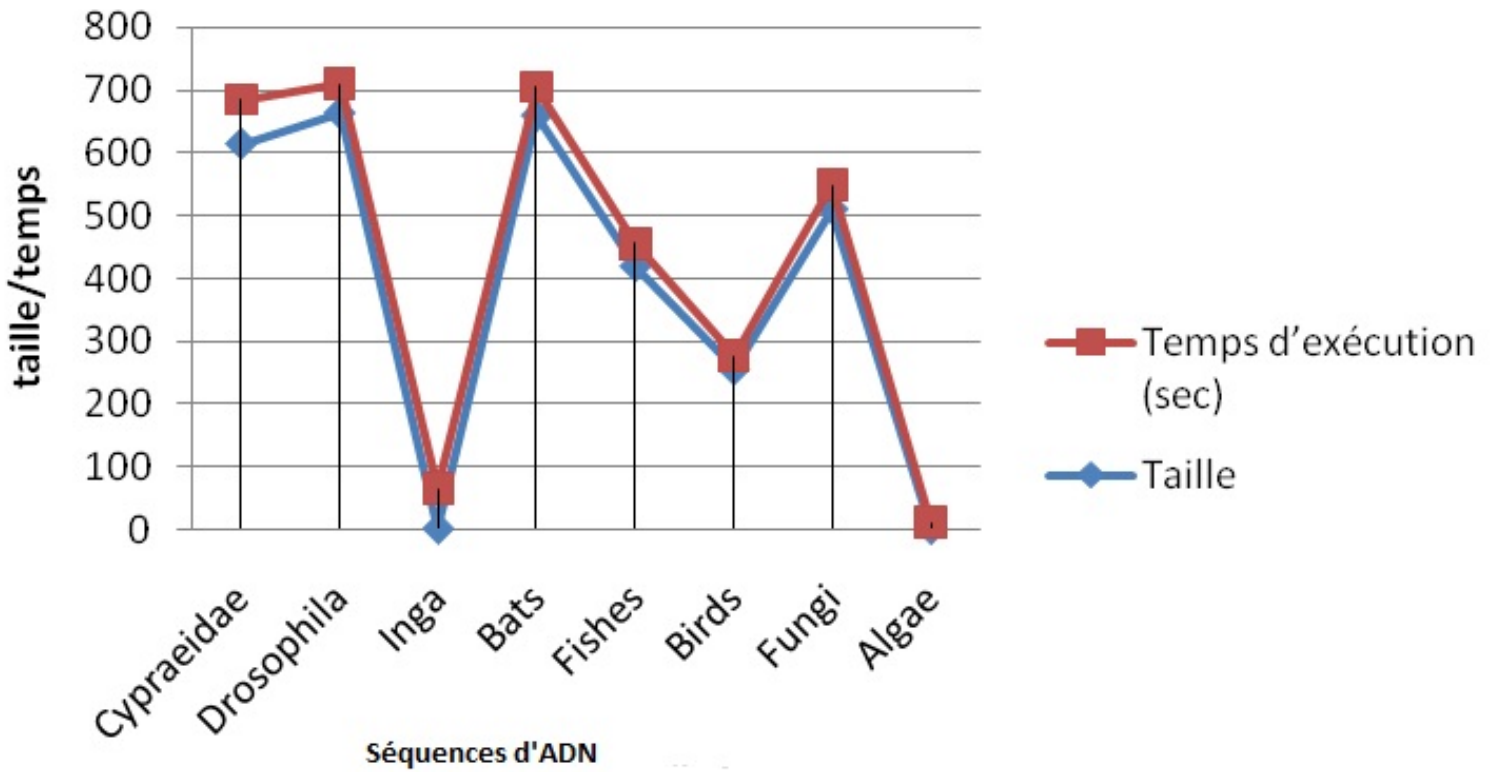


FIGURE 4.8 – Temps d'exécution en fonction de la taille de séquence d'ADN

La figure 4.8 montre bien la relation qui existe entre le temps d'exécution d'apprentissage et de la taille de séquence d'ADN ; la taille des brins d'ADN a une influence sur le temps d'exécution d'entraînement. Une grande dimension provoque une durée d'exécution importante.

De même cette situation permet aussi d'augmenter la capacité de stockage, donc la taille des séquences d'ADN peut influencer la performance de réseau de classification du point de vue complexité[A.Dakhli 2016].

Tableau 4.9 – Matrice de confusion de la classification de barcodes de séquences d'ADN

Classes actuelles	Classes prédites							Taux de classification(%)
	1	2	3	4	5	6	7	
Cypraeidaet	108	0	0	0	1	2	0	98.181
Drosophila	0	9	0	0	0	0	0	100
Inga	0	0	16	0	1	0	0	94.118
Bats	0	0	0	28	1	1	0	93.333
Fishes	0	0	0	0	29	1	0	96.667
Birds	0	0	0	1	1	57	1	95
Fungi	0	0	0	0	0	0	4	100
Rappel(%)	100	100	100	96.55	87.87	93.4	100	
Taux de classification moyen(%)								96,630

Le tableau 4.9 présente la matrice de confusion comme étant un critère pour évaluer la performance de notre méthode de classification. Dans ce tableau nous constatons la répartition des bonnes classifications par classe ainsi que le taux de classification total pour l'ensemble des séquences d'ADN. A l'aide de notre approche nous avons pu aboutir à un taux de classification global égal à 96,630%.

Les séquences d'ADN sont effectivement bien classées en Cypraeidae(98.181%). Il existe une légère confusion avec la classe Fishes et la classe Birds de même nous constatons que les séquences d'ADN sont effectivement bien classées en Drosophila et Fungi (100%).

D'après ce tableau nous pouvons aussi constater la similarité entre les barcodes de séquences d'ADN.

Par exemple, deux séquences d'ADN de la classe « Bats » sont classées respectivement dans la classe 5 et 6 et par suite nous pouvons confirmer l'identité entre ces séquences d'ADN qui peut être expliquée par l'évolution historique des ses espèces.

Tableau 4.10 – Taux de classification qui concerne notre approche et des autres méthodes (SVM,Jrip,J48,Naive Bayes)

Base de données de barcode d'ADN	SVM	Jrip	J48	Naïve Bayes	Réseau d'ondelettes
Cypraeidae	94.32	86.93	91.76	93.18	98.181
Drosophila	98.28	94.83	91.38	96.55	100
Inga	89.83	88.14	88.14	91.53	94.118
Bats	100	100	98.15	100	96.553
Fishes	95.50	90.09	92.79	97.30	96.667
Birds	98.42	84.86	91.80	94.32	95
Fungi	80	50	60	70	100
Algae	100	60	60	100	100
Taux de classification moyen(%)	94.54	81.85	84.25	92.86	97,162375

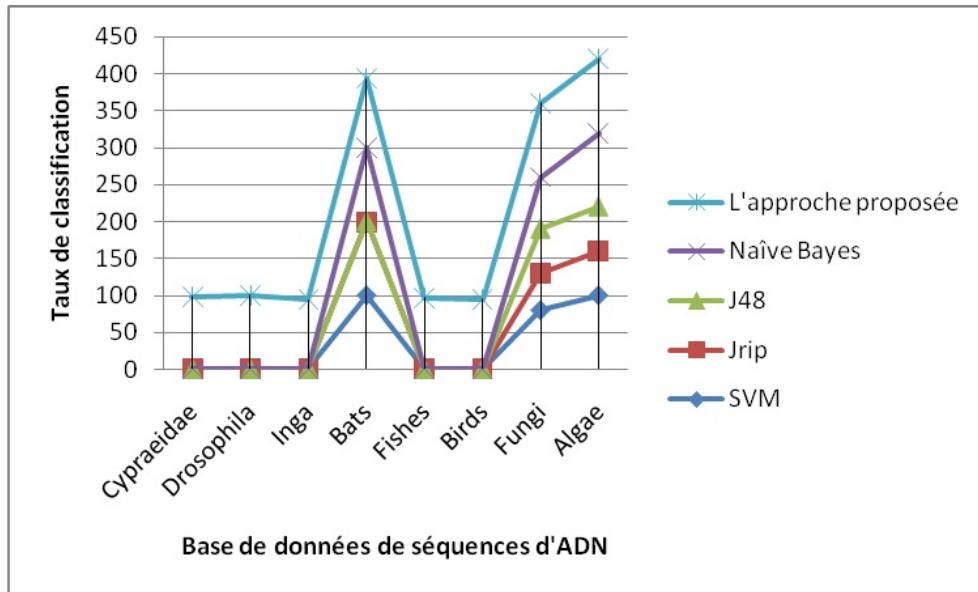


FIGURE 4.9 – Comparaison des taux de classification de l'approche proposée avec d'autres classifieurs

Le tableau 4.10 et la figure 4.9 dressent la comparaison entre les taux de classification de notre approche et les autres méthodes de classification (SVM, Jrip, J48, Naive Bayes). Cette comparaison montre bien la performance de notre approche devant les autres méthodes ; les séquences d'ADN des classes suivantes : Drosophila, Fungi et Algae (100%) sont correctement classées. De même notre approche a un taux de classification moyenne qui vaut 97,162375%.

Nous constatons aussi la performance de la méthode SVM pour classifier les séquences d'ADN. En se basant sur les résultats trouvés nous pouvons signaler que notre approche permet d'obtenir une meilleure performance que l'autres méthodes.

B.Classification des sequences d'ADN des espèces microbiennes (HOG100, HOG200 et HOG300)

Les bases de données HOG100, HOG200 et HOG300 sont utilisées pour

mieux évaluer notre approche. Ces bases contiennent des genes de sequences d'ADN des espèces microbiennes. La taille de ces sequences varie entre 100 à 300 nucléotides.

Le tableau 4.11 présente les details de ces bases de données :

Tableau 4.11 – Bases de données des sequences d'ADN des espèces microbiennes (HOG100, HOG200 et HOG300)

Base de données	Nombre de famille	Nombre de séquence d'ADN	Taille moyenne de séquences	Taille de base de données(MB)
HOG100	100	9648	1484	15.1
HOG200	200	22585	1557	37.0
HOG300	300	27825	1448	42.6

Pour évaluer la performance de notre approche nous avons fait plusieurs simulations en utilisant des données test et des données d'apprentissage. Ces données sont présentées dans le tableau 4.12 :

Tableau 4.12 – Distribution des données d'apprentissage et de test

Base de données	Totale	Apprentissage	Test
HOG100	500	300	200
HOG200	600	400	200
HOG300	700	600	100

Les résultats des expériences ont été effectués pour prouver l'efficacité de notre approche proposée. Les paramètres d'évaluation à savoir précision, rappel et F-mesures sont utilisés pour comparer notre approche avec d'autres méthodes concurrentielles.

L'influence des tailles des vecteurs de caractéristiques des séquences d'ADN sur la performance des méthodes de classifications

La molécule d'ADN est formée de nucléotides. Elle est constituée de deux chaînes de nucléotides complémentaires. Les molécules d'ADN sont les plus grosses molécules du monde vivant et sont présentes dans tous les organismes vivants. Une molécule d'ADN est une double hélice composée de deux brins enroulés l'un autour de l'autre ; nous disons que l'ADN est bicaténaire (contrairement à l'ARN, qui est monocaténaire).

Chacun de ces brins est constitué d'un enchaînement de bases dites puriques (guanine, G ; adénine, A) et pyrimidiques (cytosine, C ; thymine, T). Ces brins d'ADN sont caractérisés par de tailles variables.

L'ADN, présent dans l'organisme, est une molécule qui peut être gigantesque avec un enchaînement linéaire de millions de nucléotides.

Comment la taille d'une séquence d'ADN a une influence sur la performance de classification.

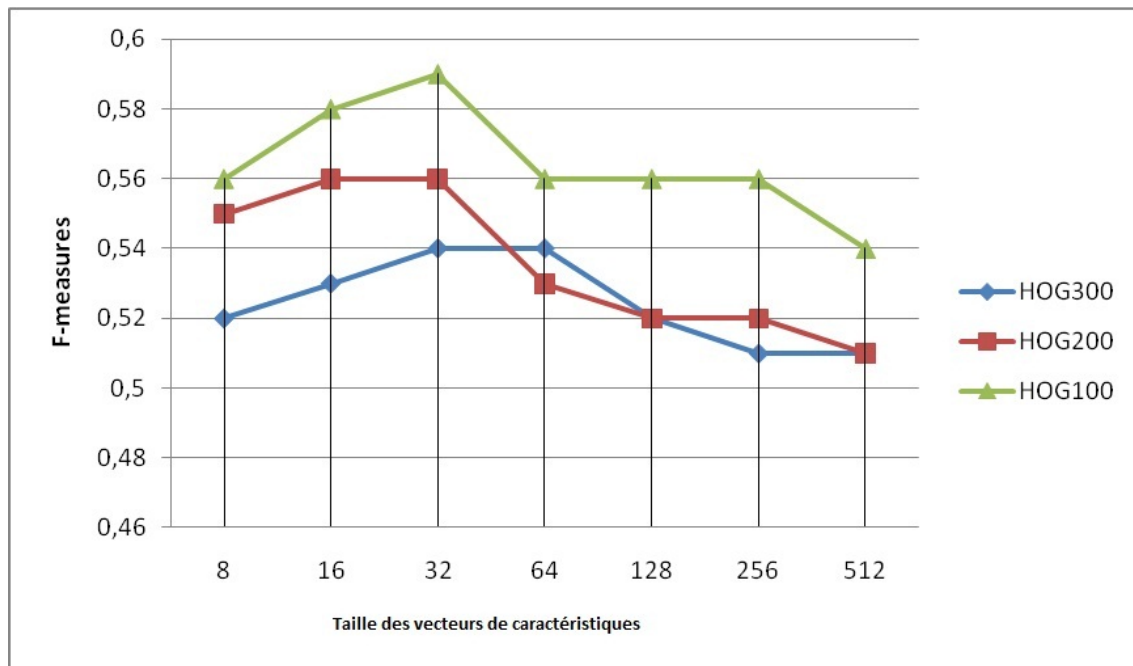


FIGURE 4.10 – Les résultats de classification d'une méthode basée sur les ondelettes utilise les vecteurs de caractéristiques (OVC) des différents ensembles de données ayant des tailles différentes.

La figure 4.10 montre que le vecteur caractéristique modèle, basé sur les ondelettes(OVC) (WFV) donne un meilleur résultat de regroupement lorsque la longueur du vecteur de caractéristique est 32 nucléotides pour HOG100 et HOG200, et un meilleur résultat obtenu lorsque le vecteur est 64 nucléotides. Cependant, la différence de la classification des résultats était très faible entre 32 et 64 nucléotides.

Par conséquent, 32 nucléotides était la longueur préférée des vecteurs de caractéristiques dans les tests.

Un vecteur de caractéristique au-delà de 64 nucléotides ne peut pas obtenir un bon résultat de regroupement.

Un vecteur de caractéristique plus courte peut réduire le temps de calcul, ce qui est très utile dans le traitement de séquences d'ADN à grande échelle.

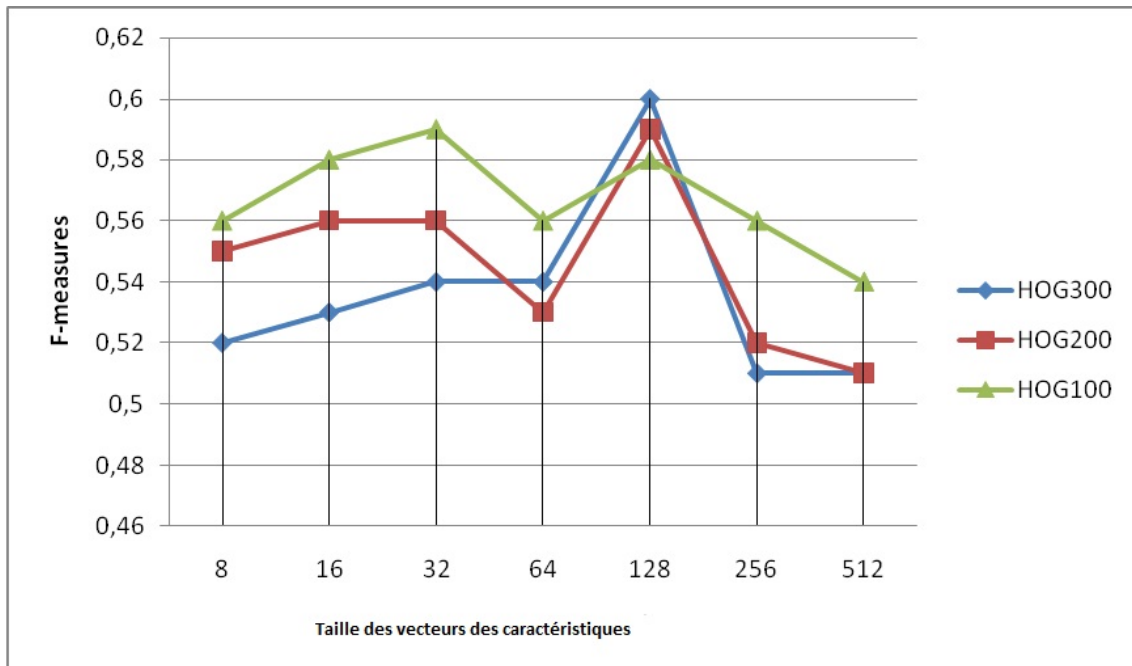


FIGURE 4.11 – Les résultats de classification de notre approche (WNN-Lasso) qui utilise les vecteurs de caractéristiques des différents ensembles de données ayant des tailles différentes.

La figure 4.11 montre que le modèle WNN-Lasso (Notre approche proposée) permet d'obtenir un meilleur résultat de regroupement lorsque la longueur du vecteur caractéristique était de 128 sur les différents ensembles de données (HOG100, HOG200 et HOG300). Dans la méthode OVC(WFV), nous obtenons un meilleur résultat lorsque le vecteur est 32 nucléotides.

Cependant, la différence de classement des résultats était très faible entre 32 (Les F-mesures moyennes = 0,57) et 128 (Les F-mesures moyennes = 0,59). Par conséquent, 32 nucléotides était la longueur préférée des vecteurs caractéristiques dans notre test.

En conséquence, un vecteur de caractéristique supérieure à 32 nucléotides ne conduit pas un bon résultat de regroupement. Un vecteur de caractéristique plus courte peut réduire le temps de calcul, ce qui est très utile dans le traitement de séquences d'ADN à grande échelle.

Les résultats montrent bien que notre approche est performante pour les différentes tailles de séquences d'ADN.

Résultats de classification des différentes méthodes de classification :

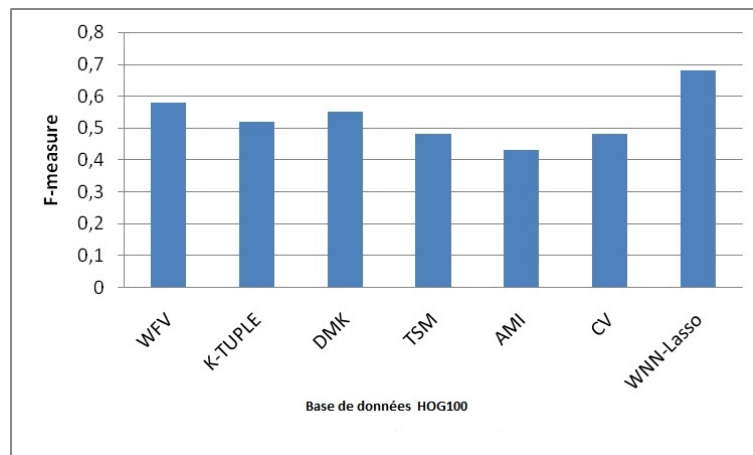


FIGURE 4.12 – F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG100.

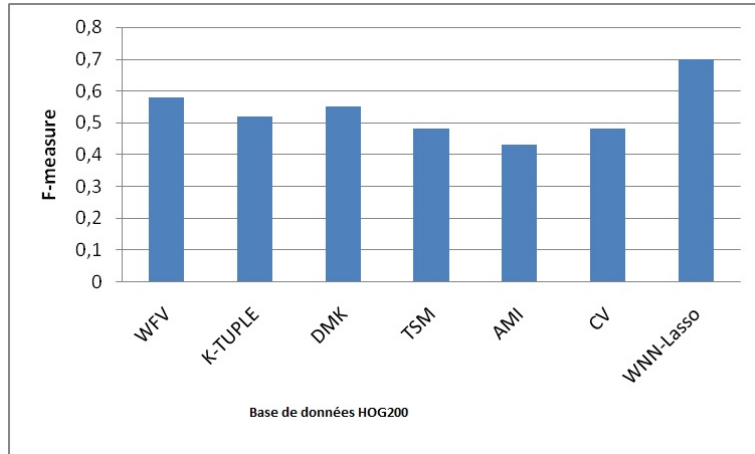


FIGURE 4.13 – F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG200.

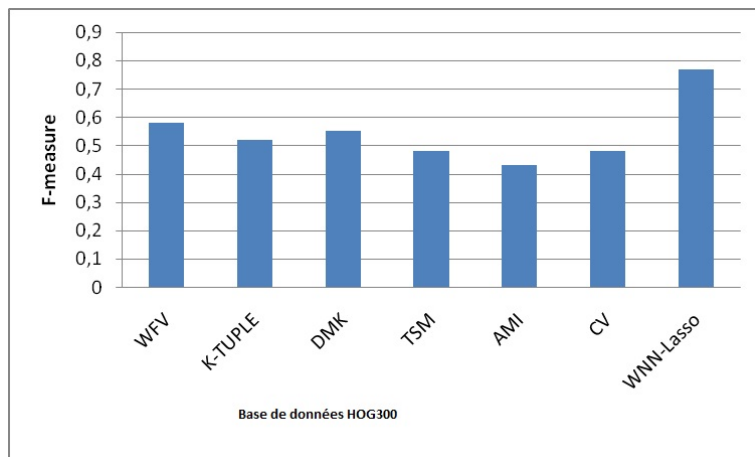


FIGURE 4.14 – F-mesure des méthodes d'alignements, WFV et notre approche(WNN-Lasso) pour la classification de base de données HOG300.

Les figures 4.12,4.13 et 4.14 montrent les résultats de classification de notre approche proposée (WNN-Lasso), le WFV et les cinq autres modèles qui utilisent l'alignement, à savoir, k-tuple, DMK, TSM, AMI et CV. La méthode WNN-Lasso fixe la longueur du vecteur de caractéristique à 128 nucléotides pour toutes les séquences d'ADN sur les trois ensembles de données (HOG100, HOG200 et HOG300). Pour les cinq autres modèles, les vecteurs de caractéristiques ont été affectés de façon significative par la taille de la fenêtre coulissante. La fenêtre coulissante pour TSM est 2, alors que pour le reste elle est 3. Etant donné que la longueur de chaque codon est trois dans l'ADN, il peut être avantageux de conserver l'information génétique de la séquence d'ADN. Ces figures illustrent les résultats de clustering de tous les modèles. Elles montrent les valeurs de F-mesure des méthodes qui sont utilisées pour classifier les trois ensembles de données (HOG100, HOG200 et HOG300). Il est clair que la performance de WNN-Lasso était la meilleure. Le résultat de l'approche WFV est moins performant que notre approche proposée mais elle est plus élevée que les autres méthodes de classification qui utilisent le principe d'alignement pour classifier les séquences d'ADN.

Tableau 4.13 – Les résultats de classification de l'approche proposée (WNN-Lasso) et les autres méthodes basées sur le principe de l'alignement pour les différents ensembles de données

	WNN-Lasso	WFV	K-tuple	DMK	TSM	AMI
Dataset	f-mesure/ Nbr.classes	f-mesure/ Nbr.classes	f-mesure/ Nbr.classes	f-mesure/ Nbr.classes	f-mesure/ Nbr.classes	f-mesure/ Nbr.classes
HOG100	0.68/1200	0.58/854	0.52/451	0.55/658	0.48/127	0.42/189
HOG200	0.7/879	0.57/845	0.53/566	0.55/754	0.48/840	0.43/412
HOG300	0.78/897	0.58/185	0.52/576	0.54/256	0.48/230	0.43/465

Le tableau 4.13 présente les résultats de classification de l'approche proposée (WNN-Lasso) et les modèles basés sur l'alignement (K-uplet,

DMK, TSM, AMI...), Sur les différents ensembles de données (HOG100, HOG200 et HOG300). Les méthodes basées sur l'alignement sont moins performantes que le WNN-Lasso (notre approche proposée) et le WFV.

Cependant, les méthodes d'alignement sont conçues pour trouver la similarité entre les séquences d'ADN qui respectent un seuil de similitude d'identité donné. Ces méthodes utilisent la notion de similarité pour comparer et classer les séquences d'ADN. Le degré de similarité est appliqué pour mesurer la ressemblance entre les séquences.

Il existe plusieurs termes permettant de nommer la ressemblance entre deux séquences biologiques. La similarité est une quantité qui se mesure en pourcentage d'identité, identité elle-même définie comme une ressemblance parfaite entre deux séquences. L'homologie quand à elle est une propriété de séquences qui a une connotation évolutive. Deux séquences sont dites homologues si elles possèdent un ancêtre commun. L'homologie présente la particularité d'être transitive. Si une séquence D'ADN S1 est homologue à S2 et S2 homologue à une séquence S3 alors S1 est homologue à S3 même si S1 et S3 se ressemblent très peu.

L'homologie se mesure par la similarité. Nous considérons qu'une similarité significative est signe d'homologie sauf si les séquences présentent une faible complexité. L'inverse n'est par contre pas vrai. Une absence totale de similarité ne veut pas dire non-homologie.

Temps d'exécution

Tableau 4.14 – Temps d'exécution des méthodes pour les différents ensembles de données

Dataset	Model	Taille de vecteur d'extraction	Temps construction de vecteur d'extraction	Temps de k-means	Totale Temps d'exécution
HOG100	WNN-Lasso	128	5.3569	75.645	81.0019
	WFV	32	8.4857	102.2634	110.7491
	K-tuple	64	7.9544	763.5223	771.4767
	DMK	64	30.2172	1550.4054	1580.6226
	TSM	12	32.7890	576.7237	609.5127
	AMI	4	167.8232	326.0293	493.8525
	CV	64	24.5168	2715.1169	2739.6337
HOG200	Our Method(WNN-Lasso)	128	16.758	350.664	367.422
	WFV	32	20.5239	645.8376	666.3615
	K-tuple	64	19.6306	3010.5426	3030.1732
	DMK	64	74.3083	7210.1629	7284.4712
	TSM	12	82.2772	2144.6954	2226.9726
	AMI	4	410.8531	1059.0261	1459.8792
	CV	64	60.3402	9247.6313	9307.9715
HOG300	Our Method(WNN-Lasso)	128	15.556	854.656	870.212
	WFV	32	24.1259	1349.6459	1373.7718
	K-tuple	64	22.5723	5560.3319	5582.9042
	DMK	64	85.1456	13785.8160	13870.9616
	TSM	12	92.8571	3329.9724	3422.8295
	AMI	4	471.9211	1577.3361	2049.2572
	CV	64	69.3221	16609.7183	16679.0404

Le tableau 4.14 montre que l'approche proposée(WNN-Lasso) améliore la précision de la classification des séquences d'ADN de même, elle réduit le temps d'exécution durant la procédure de la classification. Ce tableau prouve la performance de notre approche du point de vue complexité et décision devant les autres méthodes qui utilisent le principe d'alignement qui provoque l'augmentation de la complexité d'une façon exponentielle.

De même nous constatons que le temps de construction des vecteurs caracté-

ristiques était beaucoup plus court que les autres modèles. Donc WNN-Lasso est plus appropriée pour des quantités énormes des séquences d'ADN. La classification par les modèles (K-uplet, DMK, TSM et AM) étudiés dépend des séquences d'ADN et de la structure des alignements. L'application de l'alignement multiple (Un grand nombre de séquences peuvent être comparées simultanément) est un problème NP-complets. L'utilisation de ces méthodes est pratiquement très coûteux et nécessite une infrastructure de calcul importante.

Notre modèle permet d'assurer la classification et la prédiction des séquences d'ADN.

Cette classification obtenue nous permet d'analyser et interpréter biologiquement les relations qui existent entre les séquences d'ADN de la même classe.

Tableau 4.15 – Erreur quadratique moyenne(EQM) d'approximation des séquences d'ADN en utilisant l'approche proposée.

Base de données	Taille de séquence d'ADN	EQM	Temps d'exécution(sec)
HOG100	128	0.009958	9.3569
HOG200	32	0.002582	5.758
HOG300	64	0.007231	7.556

Pendant la phase de l'approximation, notre approche décompose le signal d'entrée pour chaque séquence, puis elle reconstruit le signal d'entrée. L'estimation de la performance de cette phase a été mesurée par l'erreur quadratique moyenne (MSE). Le tableau 4.15 montre que l'erreur quadratique moyenne (MSE) obtenue est très faible (0,002582). De même le temps d'exécution est court relativement à la taille de la séquence d'ADN. Les résultats montrent que la taille de la séquence d'ADN augmente le temps d'exécution durant la phase d'apprentissage.

Le temps d'exécution dépend de la taille de la séquence d'ADN. Lorsque la taille est égale à 128 nucléotides, le temps d'apprentissage est égal à 9.3569 secondes.

4.4 Conclusion

Ce chapitre nous a permis de présenter les résultats expérimentaux de l'application de notre approche comme étant un classificateur non supervisé. Pour évaluer sa performance nous avons utilisé plusieurs bases de données qui contiennent des séquences d'ADN de différentes tailles.

Au début, nous avons présenté les différents critères utilisés pour mesurer la performance d'une méthode de classification automatique. Dans cette section nous avons défini les métriques les plus utilisés dans le domaine d'analyse de données. Ensuite, on a dressé les différentes bases de données qui sont utilisées pour faire des simulations pour notre approche proposée. Enfin, nous avons dressé les résultats expérimentaux élaborés en appliquant notre approche de même nous avons fait des interprétations et des analyses des résultats fournis. Ces analyses concernent la biologie moléculaire qui étudie les séquences d'ADN comme étant une information génétique, d'une façon détaillée.

Conclusion et perspectives

Dans cette thèse, nous avons travaillé sur plusieurs objectifs fixés à priori. Les travaux réalisés ont été présentés sur la base d'un ensemble de contributions touchants le domaine de bio-informatique. Les résultats obtenus confirment les performances de notre classifieur de séquences d'ADN en termes de précision et de robustesse. L'objectif général de nos travaux de thèse était la classification des séquences d'ADN en utilisant un réseau d'ondelettes comme un classificateur.

Au début, nous avons appliqué la codification binaire ou la méthode énergétique en utilisant la technique Electron-ion interaction pseudopotential (EIIP) pour codifier les séquences d'ADN. Ensuite, nous avons utilisé la densité spectrale de puissance (Power Spectrum) pour distinguer la composition des nucléotides au niveau d'une séquence d'ADN afin d'obtenir un signal convenable qui a été utilisé pour alimenter notre réseau d'ondelettes.

Pour réaliser cette classification nous avons utilisé un réseau d'ondelettes bêta qui est construit à l'aide d'un algorithme incrémental et une méthode de sélection basée sur les moindres carrés tronqués (Least Trimmed Squares : LTS) ou la technique de sélection par orthogonalisation.

Nous avons utilisé une approche incrémentale pour construire notre réseau par l'ajout d'une ondelette à chaque fois dans la couche cachée. La sélection de cette ondelette à partir d'une bibliothèque, a été réalisée par une méthode de sélection. L'application de cette stratégie nous a permis de construire un réseau d'ondelettes performant du point de vue complexité et décision.

Les résultats obtenus prouvent la performance de notre réseau élaboré. Pour bien évaluer notre système nous avons utilisé plusieurs bases de données des séquences d'ADN par exemple : bactéries, eucaryotypes,, et des bases de données contenant des barcodes de séquences d'ADN.

Pour mesurer les performances de notre classificateur, nous avons utilisé des

métriques d'évaluation basées sur la précision, le rappel et la matrice de confusion. Les regroupements des séquences d'ADN obtenus nous ont permis de faire des analyses et des interprétations biologiques qui concernent la traçabilité et l'évolution des espèces dans l'histoire.

Les travaux accomplis durant notre thèse ont ouvert des perspectives de travaux futurs. Nous avons jugé possible d'adapter notre modèle afin de classifier d'autres types d'ADN intitulés puce à ADN (DNA-microarray ou DNA chip). Cette puce contient les profils d'expression de gènes d'ADN. Ces profils indiquent le niveau d'expression de gènes dans les cellules. Chaque nucléotide est représenté par plusieurs attributs. Pour réduire ou sélectionner les variables pertinentes on peut comparer les performances de notre système comparé à une méthode d'analyse en composantes principale (ACP) par exemple.

De même, il est possible d'améliorer la construction de topologie du réseau d'ondelettes en utilisant plusieurs autres algorithmes incrémentaux par exemple : MTower, MTiling-real, Upstart, ou bien l'utilisation d'une méthode hybride qui combine l'arbre de décision et les algorithmes génétiques.

Publications Scientifiques

Publications dans des conférences internationales

[A.Dakhli 2016a]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR , Wavelet neural network initialization using LTS for DNA Sequence Classification,Advanced Concepts for Intelligent Vision Systems.Acivs2016,LNCS, volume 10016,pp 661-673.

[A.Dakhli 2016b]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR, DNA Sequence Classification using Power Spectrum and Wavelet neural network ,16th International Conference on Hybrid Intelligent Systems (HIS 2016)),Volume 96, 2016, Pages 418-427.

[A.Dakhli 2016]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR , Wavelet Neural Networks for DNA Sequence Classification Using the Genetic Algorithms and the Least Trimmed Square. Procedia Computer Science 96 (2016) 418-427 (KnowledgeBased and Intelligent Information Engineering Systems : Proceedings of the 20th International Conference KES-2016).

Publications dans des journaux internationaux

[A.Dakhli 2014a]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR, Classification DNA Sequences of Bacterias using Multi Library Wavelet Networks , International Journal of Biomedical Science Bioinformatics IJBSB.Volume 1 : Issue 1,23-29, 30 September, 2014.

[A.Dakhli 2014b]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR ,Unsupervised classification of Eukaryotic DNA sequences using Multi Library Wavelet Networks, International Journal of Computer Science Issues, Vol. 11, Issue 2, No 1, March 2014 ISSN (Print) : 1694-0814 | ISSN (Online) : 1694-0784.

[A.Dakhli 2015]

Abdesselem DAKHLI, Wajdi BELLIL, Chokri BEN AMAR, Unsupervised Classification of DNA Barcodes Species Using Multi-Library Wavelet Networks, World Academy of Science, Engineering and Technology International Journal of Computer, Electrical, Automation, Control and Information Engineering Vol. :9, No. :4, 2015.

Bibliographie

- [a 1995] M. Cannon a et J.J.E. Slotine. *SpaceFrequency Localized Basis Function Networks for Nonlinear System Estimation and Control*. *Neurocomputing*, vol. 9, no. 3, pages 293-342, 1995. (Cité en page 94.)
- [Abbasifard 2014] M. Reza Abbasifard, B. Ghahremani et H. Naderi. *A Survey on Nearest Neighbor Search Methods*. *International Journal of Computer Applications*, vol. 95, no. 12, pages 39 - 52, 2014. (Cité en page 26.)
- [Abe 2006] T. Abe, H. Sugawara, M. Kinouchi et T. Ikemura. *Novel Phylogenetic Studies of Genomic Sequence Fragments Derived from Uncultured Microbe Mixtures in Environmental and Clinical Samples*. *DNA Research*, vol. 12, no. 5, pages 281-290, 2006. (Cité en pages 45 et 46.)
- [A.Dakhli 2014a] A.Dakhli, W. Bellil et C.Ben Amar. *Classification DNA Sequences of Bacterias using Multi Library Wavelet Networks*. *International Journal of Biomedical Science Bioinformatics IJBSB*, vol. 1, no. 1, pages 23-29, 2014. (Cité en pages 146 et 153.)
- [A.Dakhli 2014b] A.Dakhli, W. Bellil et C.Ben Amar. *Unsupervised classification of Eukaryotic DNA sequences using Multi Library Wavelet Networks*. *IJCSI International Journal of Computer Science Issues*, vol. 11, no. 1, pages 64-73, 2014. (Cité en pages xi, 153, 154 et 155.)
- [A.Dakhli 2015] A.Dakhli, W. Bellil et C.Ben Amar. *Unsupervised Classification of DNA Barcodes Species Using MultiLibrary Wavelet Networks*. *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 9, no. 4, pages 910-916, 2015. (Cité en pages xi, 158 et 159.)
- [A.Dakhli 2016] A.Dakhli, W. Bellil et C.Ben Amar. *Wavelet Neural Networks for DNA Sequence Classification Using the Genetic Algorithms and the*

- Least Trimmed Square*. KnowledgeBased and Intelligent Information Engineering Systems, vol. 96, no. 10, pages 418-427, 2016. (Cit  en pages xi, 159 et 160.)
- [Alberts 2014] B. Alberts, A. Johnson, J. Lewis, D. Morgan, M. Raff, K. Roberts et P. Walter. *Molecular biology of the cell*, page 1464. 6. Garland Science, 2014. (Cit  en pages 9 et 11.)
- [Alimi 2003] A. M. Alimi. *Beta neurofuzzy systems*. Task Quarterly Journal, vol. 7, no. 1, page 23 41, 2003. (Cit  en page 70.)
- [AlKhateeb 2011] J. H. AlKhateeb, O. Pauplin et J. Jiang J. Ren. *Performance of hidden Markov model and dynamic Bayesian network classifiers on handwritten Arabic word recognition*. Knowledge-Based Systems, vol. 24, pages 680-688, 2011. (Cit  en page 27.)
- [Amar 2005] C. Ben Amar, M. Zaied et A. Alimi. *Beta wavelets. Synthesis and application to lossy image compression*. Advances in Engineering Software, vol. 36, no. 7, page 459 474, 2005. (Cit  en page 72.)
- [Amar 2006] C. Ben Amar, W. Bellil et M.A. Alimi. *Beta function and its derivatives : A new wavelet family*. Transactions on Systems, Signals Devices, vol. 1, no. 3, pages 275-293, 2006. (Cit  en page 72.)
- [Anderson 1989] J. A. Anderson et E. Rosenfeld. *Neurocomputing : Foundations of research*, page 752. 1. A Bradford Book, 1989. (Cit  en pages 54 et 55.)
- [Aniba 2010] M. Radhouene Aniba, O. Poch et J. D. Thompson. *Survey and Summary Issues In bioinformatics benchmarking : the case study of multiple sequence alignment*. Nucleic Acids Res., vol. 38, no. 21, pages 7353-7363, 2010. (Cit  en page 31.)
- [Antonini 1992] M. Antonini, M. Barlaud, P. Mathieu et I. Daubechies. *Image coding using wavelet transform*. IEEE Transactions on Image Processing, vol. 1, no. 2, pages 205 - 220, 1992. (Cit  en page 69.)

- [b. Arniker 2012] S. b. Arniker et H. K. Kwan. *Advanced Numerical Representation of DNA Sequences*. International Proceedings of Chemical, Biological Environmenta, vol. 31, no. 1, 2012. (Cité en page 113.)
- [Bakshi 1993] B. R. Bakshi et G. Stephanopoulos. *Wavenet : a multiresolution, hierarchical neural network with localized learning*. American Institute of Chemical Engineers, vol. 39, no. 1, 1993. (Cité en page 64.)
- [Bellil 2004] W. Bellil, C. Ben Amar, M. Zaied et M.A. Alimi. *La fonction Beta et ses dérivées : vers une nouvelle famille d'ondelettes*. First International Conference on Signal, System and Design, SCS'04., pages 201-207, 2004. (Cité en page 72.)
- [Bellil 2008] W. Bellil, C. Ben Amar et M.A. Alimi. *Comparison between Beta Wavelets Neural Networks, RBF Neural Networks and Polynomial Approximation for 1D, 2D Functions Approximation*. International Journal of Computer, Electrical, Automation, Control and Information Engineering, vol. 2, no. 1, pages 189-194, 2008. (Cité en page 74.)
- [B.Fritzke 1994] B.Fritzke. *Fast learning with incremental RBF networks*. Neural Processing Letters, vol. 1, no. 1, pages 2 - 5, 1994. (Cité en page 85.)
- [Bhatia 2010] N. Bhatia. *Survey of Nearest Neighbor Techniques*. (IJCSIS) International Journal of Computer Science and Information Security, vol. 8, no. 2, pages 302 - 305, 2010. (Cité en page 26.)
- [Biswas 2015] S. K. Biswas et M. M. Alam Mia. *Image Reconstruction Using Multi Layer Perceptron (MLP) And Support Vector Machine (SVM) Classifier And Study Of Classification Accuracy*. International Journal of Scientific Technology Research, vol. 4, no. 2, pages 226 – 231, 2015. (Cité en page 27.)

- [Breiman 1984] L. Breiman, R. A. Friedman, R. A. Olshen et C. G. Stone. *Classification and Regression Trees*. Pacific Grove, CA : Wadsworth, 1984. (Cité en page 24.)
- [Burge 2002] C. Burge. *Bioinformaticists Will Be Busy Bees*. Genome Technology, vol. 17, 2002. (Cité en page 6.)
- [Campedel 2008] M. Campedel, I. Kyrgyzov et H. Maître. *Consensual clustering for unsupervised feature selection. Application to SPOT5 satellite images indexing*. JMLR : Workshop and Conference Proceedings, vol. 4, pages 48-59, 2008. (Cité en page 25.)
- [Carpenter 1992] G. A. Carpenter, S. Grossberg, N. Markuzon et J. H. Reynolds. *Fuzzy ARTMAP : A neural network architecture for incremental supervised learning of analog multidimensional maps*. IEEE Transactions on Neural Networks, vol. 3, no. 5, pages 698 -713, 1992. (Cité en page 40.)
- [Cavalieri 2005] D. Cavalieri et C. De Filippo. *Bioinformatic methods for integrating wholegenome expression results into cellular networks*. Drug Discov Today, vol. 10, no. 10, pages 727-734, 2005. (Cité en page 36.)
- [Chan 2008] C. Kenneth Chan, A. L Hsu, S. K Halgamuge et S. Tang. *Binning sequences using very sparse labels within a metagenome*. BMC Bioinformatics, vol. 9, no. 1, pages 1-17, 2008. (Cité en page 46.)
- [Chatterji 2008] S. Chatterji, I. Yamazaki, Z. Bai et J. Eisen. *CompostBin : A DNA composition based algorithm for binning environmental shotgun reads*. International conference on Research in computational molecular biology, pages 17-28, 2008. (Cité en page 46.)
- [Chen 1989] S. Chen, S.A. Billings et W. Luo. *Orthogonal Least Squares Methods and Their Application to Nonlinear System Identification*. International Journal of Control, vol. 50, no. 5, pages 1873-1896, 1989. (Cité en page 100.)

- [Chen 1991] S. Chen, C. F. N. Cowan et P. M. Grant. *Orthogonal least squares learning algorithm for radial basis function networks*. IEEE Transactions on Neural Networks, vol. 5, no. 5, pages 302 - 309, 1991. (Cité en page 102.)
- [Cohen 2004] J. Cohen. *Bioinformatics An Introduction for Computer Scientists*. ACM Computing Surveys, vol. 36, no. 2, pages 122-158, 2004. (Cité en page 7.)
- [Crosby 2004] K. Crosby et P. Gabbert. *classification of intron and exon sequences using the SPRINT algorithm*. Computational Systems Bioinformatics Conference, CSB 2004 Proceedings, 2004. (Cité en page 43.)
- [Daubechies 1990] I. Daubechies. *The wavelet transform, timefrequency localization and signal analysis*. IEEE Transactions on Information Theory, vol. 36, no. 5, pages 961 - 1005, 1990. (Cité en pages 65 et 68.)
- [Daubechies 1992] I. Daubechies. Ten lectures on wavelets, page 350. 1. CBMSNSF Regional Conference Series in Applied Mathematics, 1992. (Cité en page 69.)
- [Daugelaite 2013] J. Daugelaite, A. O' Driscoll et R. D. Sleator. *An Overview of Multiple Sequence Alignments and Cloud Computing in Bioinformatics*. ISRN Biomathematics, 2013. (Cité en page 32.)
- [Denoeux 1995] T. Denoeux. *A k-nearest neighbor classification rule based on Dempster-Shafer theory*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 25, no. 5, pages 804- 813, 1995. (Cité en page 26.)
- [Diaz 2009] N. N Diaz, L. Krause, A. Goesmann, K. Niehaus et T. W Nattkemper. *TACOA a Taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach*. BMC Bioinformatics, vol. 10, no. 1, pages 1-16, 2009. (Cité en pages 45 et 46.)

- [Dunkin 1997] N. Dunkin, J. ShaweTaylor et P. Koiran. *A New Incremental Learning Technique*. Neural Nets WIRN VIETRI96, pages 112-118, 1997. (Cité en page 85.)
- [Eddy 2002] S. R. Eddy. *A memory-efficient dynamic programming algorithm for optimal alignment of a sequence to an RNA secondary structure*. BMC Bioinformatics, vol. 3, no. 18, 2002. (Cité en page 34.)
- [Elhadi 2012] G. F. Elhadi, R. M. Farouk et A. T. Issa. *Protein sequence for clustering DNA based on Artificial Neural Networks*. IJCSI International Journal of Computer Science, vol. 9, no. 3, pages 161 -167, 2012. (Cité en page 38.)
- [F. Sousa 2005] J. Tendeiro F. Sousa. *A Validation Methodology in Hierarchical Clustering*. International Symposium of Applied Stochastic Models and Data Analysis (ASMDA), pages 396-403, 2005. (Cité en page 29.)
- [Fahlman 1990] S. E. Fahlman et C. Lebiere. *The cascadedcorrelation learning architecture*. Advances in Neural Information Processing Systems 2, pages 524-532, 1990. (Cité en pages 85 et 88.)
- [Gallant 1986] S. Gallant. *Three Constructive Algorithms for Network Learning*. Eighth Annual Conference of the Cognitive Science Society, pages 652 - 660, 1986. (Cité en page 92.)
- [Gavarraju 2016] L. N. Jayaprada Gavarraju, P. Jeevana Jyothi et K. Kar-teeka Pawan. *A Literature Survey on Multiple Sequence Alignment Algorithms*. International Journal of Advanced Research in Computer Science and Software Engineering, vol. 6, no. 3, pages 280-288, 2016. (Cité en page 32.)
- [Gilbert 2000] D. R. Gilbert, M. Schroeder et J. van Helden. *Interactive visualization and exploration of relationships between biological objects*. TRENDS in biotechnology, vol. 18, no. 12, pages 487-494, 2000. (Cité en page 36.)

- [Hebb 1949] D.O. Hebb. *The Organization of Behavior*. Brain Research Bulletin, vol. 50, no. 5, pages 60-78, 1949. (Cité en page 55.)
- [Iyengar 2002] S. Sitharama Iyengar, E. C. Cho et V.V. Phoha. Foundations of wavelet networks and applications, page 288. 1. Chapman and Hall/CRC, 2002. (Cité en page 87.)
- [Jach 2010] A. E Jach et J. Miguel Marin. *Classification of Genomic Sequences via Wavelet Variance and a Self-Organizing Map with an Application to Mitochondrial DNA*. Statistical Applications in Genetics and Molecular Biology, vol. 9, no. 1, 2010. (Cité en page 38.)
- [Jeng 2006] C. Jeng, I. Yang, K. Hsieh et C. Lin. *Bacteria Classification on Power Spectrums of Complete DNA Sequences by SelfOrganizing Map*. Neural Information Processing, vol. 9, no. 3, pages 53-57, 2006. (Cité en page 39.)
- [Juditsky 1994] A. Juditsky, Q. Zhang, B. Delyon, P. Yves Glorennec et A. Benveniste. *Wavelets in identification wavelets, splines, neurons, fuzzies : how good for identification*. AS Signal Processing and Control, 1994. (Cité en page 95.)
- [Kanungo 2002] T. Kanungo, D. M. Mount, N. S. Netanyahu et C. D. Piatko. *An efficient k-means clustering algorithm : analysis and implementation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 24, no. 7, pages 881-892, 2002. (Cité en page 27.)
- [Katoh 2010] K. Katoh et H. Toh. *Parallelization of the MAFFT multiple sequence alignment program*. Bioinformatics, vol. 26, no. 15, pages 1899-900, 2010. (Cité en page 31.)
- [Koren 2014] A. Koren, R. Handsaker, N. Kamitaki, R. Karlic, S. Ghosh, P. Polak, K. Eggan et S. A. McCarroll. *Genetic Variation in Human DNA Replication Timing*. Cell, vol. 159, no. 5, page 10151026, 2014. (Cité en page 19.)

- [Krause 2008] L. Krause, N.N. Diaz, A.Goesmann, S.Kelley, T. W. Nattkemper, F.Rohwer, R. A. Edwards et J. Stoye. *Phylogenetic classification of short environmental DNA fragments*. Nucleic Acids Research, vol. 36, no. 7, pages 2230-2239, 2008. (Cité en page 46.)
- [Lampinen 1992] J. Lampinen et E. Oja. *Clustering properties of hierarchical self-organizing maps*. J. Math. Imag. Vis, vol. 2, no. 2â3, pages 261-272, 1992. (Cité en page 29.)
- [Li 2010] H. Li et N. Homer. *A survey of sequence alignment algorithms for next-generation sequencing*. Briefings in Bioinformatics, vol. 11, no. 5, pages 473-483, 2010. (Cité en page 23.)
- [Loewenstern 1995] D. Loewenstern, H. Hirsh, P. Yianilos et M. Noordewier. *DNA sequence classification using compressionbased induction*. Center for Discrete Mathematics Theoretical Computer Science, 1995. (Cité en page 39.)
- [Mallat 1989] S. G. Mallat. *A Theory for Multiresolution Signal Decomposition : The Wavelet Representation*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, pages 674-693, 1989. (Cité en pages 64, 65 et 81.)
- [Marucci 2014] E. A. Marucci, G. F. D. Zafalon, J. C. Momente, L. A. Neves, C. R. ValÃˆancio, A. R. Pinto, A. M. Cansian, R. C. G. de Souza, Y. Shiyou et J. M. Machado. *An Efficient Parallel Algorithm for Multiple Sequence Similarities Calculation Using a Low Complexity Method*. BioMed Research International, pages 1-6, 2014. (Cité en page 31.)
- [Masai 2010] H. Masai, S. Matsumoto, Z. You, N. Yoshizawa-Sugata et M. Oda. *Eukaryotic chromosome DNA replication : where, when, and how*. Annual Review of Biochemistry, vol. 130, pages 79-89, 2010. (Cité en page 19.)

- [Mavrommatis 2007] K. Mavrommatis, N. Ivanova, K. Barry et N. C Kyrpides. *Use of simulated data sets to evaluate the fidelity of metagenomic processing methods*. Nature Methods, vol. 4, no. 1, pages 495-500, 2007. (Cité en page 46.)
- [Mazouzi 2014] A. Mazouzi, G. Velimezi et J. Loizou. *DNA replication stress : Causes, resolution and disease*. Experimental Cell Research, vol. 329, no. 1, page 8593, 2014. (Cité en page 20.)
- [McCulloch 1943] W. S. McCulloch et W. Pitts. *A logical calculus of the ideas immanent in nervous activity*. The bulletin of mathematical biophysics, vol. 5, no. 4, pages 115-133, 1943. (Cité en page 55.)
- [McHardy 2007] A. Carolyn McHardy, H. Garcia Martin, A. Tsirigos, P. Hugenholtz et I. Rigoutsos. *Accurate phylogenetic classification of variable length DNA fragments*. Nature Methods, vol. 4, no. 1, pages 63-72, 2007. (Cité en pages 44 et 46.)
- [Meyer 1985] Y. Meyer. *Principe d'incertitude, bases hilbertiennes et algèbres d'opérateurs*. Séminaire Bourbaki,, vol. 662, no. 15, pages 209-223, 1985. (Cité en page 65.)
- [Meyer 1990] Y. Meyer. *Ondelettes et opérateurs : Ondelettes*, page 215. 1. Hermann, 1990. (Cité en page 81.)
- [Minsky 1969] M. Minsky et S. Papert. *Perceptrons : An introduction to computational geometry*, page 268. 1. MIT Press, 1969. (Cité en page 55.)
- [Mohamed 2006] S. Mohamed, D. Rubin et T. Marwala. *Multiclass Protein Sequence Classification Using Fuzzy ARTMAP*. IEEE International Conference on Systems, Man and Cybernetics, vol. 2, pages 1676-1681, 2006. (Cité en page 40.)

- [Mohamed 2007] S. Mohamed, D. Rubin et T. Marwala. *Incremental Learning for Classification of Protein Sequences*. International Joint Conference on Neural Networks, pages 19 -24, 2007. (Cité en page 40.)
- [Mohamed 2013] M. Mohamed, A. A. AlMehdhar, M. Bamatraf et M. R. Girgis. *Enhanced Self-Organizing Map Neural Network for DNA Sequence Classification*. Intelligent Information Management,, vol. 5, pages 25-33, 2013. (Cité en page 39.)
- [Muller 2003] H. M. Muller et S. E. Koonin. *Vector space classification of DNA sequences*. Journal of Theoretical Biology, vol. 223, no. 2, pages 161-169, 2003. (Cité en page 40.)
- [Musavi 1992] M. T. Musavi, D. M. Hummels, A. J. Laffely et S. P. Kennedy. *Noise density estimation using neural networks*. Neural Networks for Signal Processing [1992] II, pages 484 - 492, 1992. (Cité en page 63.)
- [Nair 2010] V. V. Nair, K. Vijayan, D. P. Gopinath et A. S. Nair. *ANN based Genome Classifier using Frequency Chaos Game Representation*. International Journal of Engineering and Technology, vol. 2, no. 3, pages 308-312, 2010. (Cité en page 37.)
- [Park 1991] J. Park et I. W. Sandberg. *Universal Approximation Using Radial-BasisFunction Networks*. Neural Computation, vol. 3, pages 246-257, 1991. (Cité en page 75.)
- [Patel 2009] A. Kumar Patel et P. Kumar. *Binary classification of uncharacterized proteins into DNA binding/non-DNA binding proteins from sequence derived features using ANN*. Digest Journal of Nanomaterials and Biostructures, vol. 4, no. 4, pages 775-782, 2009. (Cité en page 39.)
- [Pati 1993] Y. C. Pati et P. S. Krishnaprasad. *Analysis and synthesis of feed-forward neural networks using discrete affine wavelet transformations*. IEEE Transactions on Neural Networks, vol. 4, no. 1, pages 73 - 85, 1993. (Cité en pages 64 et 93.)

- [Pavlov 1927] P. I. Pavlov. *Conditioned Reflexes : An investigation of the physiological activity of the cerebral cortex*. Annals of Neurosciences, vol. 17, no. 3, pages 136-141, 1927. (Cité en page 54.)
- [Payan 2006] F. Payan et M. Antonini. *Mean square error approximation for wavelet-based semiregular mesh compression*. IEEE Transactions on Visualization and Computer Graphics (TVCG), vol. 12, no. 4, pages 649-657, 2006. (Cité en page 69.)
- [P.K.Srimani 2013] P.K.Srimani et S.Mahesh. *Multiclass Tumour classification by using SVM classifiers*. American International Journal of Research in Science, Technology, Engineering Mathematics, vol. 3, no. 1, pages 103 - 108, 2013. (Cité en page 27.)
- [Polikar 2001] R. Polikar, L. Upda, S.S. Upda et V. Honavar. *Learn++ : an incremental learning algorithm for supervised neural networks*. IEEE Transactions on Systems, Man, and Cybernetics, vol. 31, no. 4, pages 497 - 508, 2001. (Cité en page 85.)
- [Rani 2012] S. Rani et S. Kaur. *Cluster Analysis Method for Multiple Sequence Alignment*. International Journal of Computer Applications, vol. 43, no. 14, pages 19-25, 2012. (Cité en page 36.)
- [Reese 2010] E. Reese et V. V Krishnan. *Classification of DNA sequences based on thermal melting profiles*. Bioinformation, vol. 4, no. 10, 2010. (Cité en page 42.)
- [Rosen 2008] G. Rosen, E. Garbarine, D. Caseiro, R. Polikar et B. Sokhansanj. *Metagenome Fragment Classification Using NMer Frequency Profiles*. Advances in Bioinformatics, pages 1-12, 2008. (Cité en page 44.)
- [Rosenblatt 1958] F. Rosenblatt. *The Perceptron : A Probabilistic Model for Information Storage and Organization in The Brain*. Psychological Review, vol. 65, no. 6, pages 386-408, 1958. (Cité en page 55.)

- [Rosenblatt 1962] F. Rosenblatt. *Principles of neurodynamics : perceptrons and the theory of brain mechanisms*. Anatomy and Physiology Cybernetics, pages 245-248, 1962. (Cité en page 55.)
- [Rousseeuw 1987] P. J. Rousseeuw et A. M. Leroy. Robust regression and outlier detection, page 360. 1. Wiley Series in Probability and Statistics, 1987. (Cité en page 122.)
- [Ruffalo 2011] M. Ruffalo, T. LaFramboise et M. Koyutürk. *Comparative analysis of algorithms for next-generation sequencing read alignment*. Bioinformatics, vol. 27, no. 20, pages 2790-2796, 2011. (Cité en page 21.)
- [Russell 2003] S. J. Russell et P. Norvig. Artificial intelligence a modern approach, pages 1-1045. 2. Library of Congress Cataloging in Publication Data, 2003. (Cité en page 54.)
- [Sandberg 2001] R. Sandberg, G. Winberg, C. Bränden, A. Kaske, I. Ernberg et J. Cöster. *Capturing Whole-Genome Characteristics in Short Sequences Using a Naive Bayesian Classifier*. Genome Research, vol. 11, no. 8, pages 1404-1409, 2001. (Cité en pages 44 et 46.)
- [Satish 1993] L. Satish et B. I. Gururaj. *Use of Hidden Markov Models for Partial Discharge Pattern Classification*. IEEE Transactions on Electrical Insulation, vol. 28, no. 2, pages 172-182, 1993. (Cité en page 27.)
- [Seo 2010] T. Seo. *Classification of nucleotide sequences using support vector machines*. Journal of Molecular Evolution, vol. 71, no. 4, pages 250-267, 2010. (Cité en page 41.)
- [Shi 2012] L. Shi et H. Huang. *DNA Sequences Analysis Based on Classifications of Nucleotide Bases*. Advances in Intelligent and Soft Computing, vol. 137, pages 379-384, 2012. (Cité en page 113.)
- [Stoppiglia 1997] H. Stoppiglia. *Méthodes statistiques de sélection de modèles neuronaux ; applications financières et bancaires*. Thèse de Doctorat de l'Université Paris 6, 1997. (Cité en page 100.)

- [Teeling 2004] H. Teeling, A. Meyerdierks, M. Bauer et F. Oliver Glöckner. *Application of tetranucleotide frequencies for the assignment of genomic fragments*. Environ Microbiol, vol. 6, no. 9, pages 938-947, 2004. (Cit  en pages 45 et 46.)
- [Tibshirani 1996] R. Tibshirani. *Regression shrinkage and selection via the lasso*. Journal of the Royal Statistical Society. Series B, vol. 58, no. 1, pages 267-288, 1996. (Cit  en pages 112, 124, 126 et 128.)
- [Urbani 1995] D. Urbani. *M thodes statistiques de selection d'architectures neuronales : application a la conception de mod les de processus dynamiques*. th se de doctorat de l'Universit  Pierre et Marie Curie, 1995. (Cit  en page 100.)
- [Vesanto 2000] J. Vesanto et E. Alhoniemi. *Clustering of the self-organizing map*. IEEE Trans Neural Netw, vol. 11, no. 3, pages 586-600, 2000. (Cit  en page 29.)
- [Vinga 2003] S. Vinga et J. Almeida. *Alignmentfree sequence comparison a review*. Bioinformatics, vol. 19, no. 4, pages 513-523, 2003. (Cit  en page 113.)
- [Volfovsky 2001] N. Volfovsky, B. J Haas et S. Salzberg. *A clustering method for repeat analysis in DNA sequences*. Genome biology, vol. 2, no. 8, 2001. (Cit  en page 33.)
- [Waghodekar 2015] P. Waghodekar et K. Bhosle. *Survey of Efficient and Fast Nearest Neighbor Search For Spatial Query on Multidimensional Data*. (IJESTA) International Journal of Engineering Studies and Technical Approach, vol. 01, no. 12, pages 6 – 15, 2015. (Cit  en page 26.)
- [Watson 1953] J. Watson et F. Crick. *A Structure for Deoxyribose Nucleic Acid*. Nature, vol. 3, no. 171, pages 737-738, 1953. (Cit  en pages 7 et 9.)

- [Wu 1993] C. Wu, M. Berry, Y. Fung et J. McLarty. *Classification of nucleotide sequences using support vector machines*. Proc Int Conf Intell Syst Mol Biol, vol. 1, pages 429-437, 1993. (Cité en page 41.)
- [Zaied 2008] M. Zaied. *Etude des réseaux d'ondelettes Bêta : Application à la reconnaissance de visages*. Thèse de doctorat, Laboratoire REGIM-ENIS. Sfax, 2008. (Cité en pages ix et 118.)
- [Zhang 1992] Q. Zhang et A. Benveniste. *Wavelet networks*. IEEE Transactions on Neural Networks, vol. 3, no. 6, pages 889 - 898, 1992. (Cité en pages 64, 77 et 87.)
- [Zhang 1993] Q. Zhang. *Regressor selection and wavelet network construction*. Decision and Control, 1993., Proceedings of the 32nd IEEE Conference on, vol. 4, no. 2, pages 3688 - 3693, 1993. (Cité en pages 95 et 100.)
- [Zhang 1995] J. Zhang, G. G. Walter, Y. Miao et W. N. Wayne Lee. *Wavelet neural networks for function learning*. IEEE Transactions on Signal Processing, vol. 43, no. 6, pages 1485-1497, 1995. (Cité en page 94.)
- [Zhang 1997] Q. Zhang. *Using wavelet network in nonparametric estimation*. IEEE Transactions on Neural Networks, vol. 8, no. 2, pages 227 - 236, 1997. (Cité en page 95.)
- [Zhang 2003] Z. Zhang et M. Gerstein. *Patterns of nucleotide substitution, insertion and deletion in the human genome inferred from pseudogenes*. Nucleic Acids Res, vol. 31, no. 18, pages 5338-5348, 2003. (Cité en page 17.)
- [Zhao 2001] J. Zhao, X. Wen Yang, J. Ping Li et Y. Yan Tang. *DNA Sequences Classification Based on Wavelet Packet Analysis*. Wavelet Analysis and Its Applications, vol. 2251, pages 424-429, 2001. (Cité en page 43.)