الجمهورية الجزائرية الديمقراطية الشعبية
وزارة التعليم العالي والبحث العلمي

UNIVERSITÉ BADJI MOKHTAR - ANNABA
BADJI MOKHTAR – ANNABA UNIVERSITY

جامعة باجي مختار – عنابـــــــــــة

**Faculté :** Sciences de l'Ingénieur
**Département :** Informatique
**Domaine :** Mathématique-Informatique
**Filière :** Informatique
**Spécialité :** Gestion et Analyse des Données Massive

# Mémoire

## Présenté en vue de l'obtention du Diplôme de Master

**Thème :**

# Classification multi-label de localisation des protéines

**Présenté par :** *Zadoud Ahmed*

**Encadrant :** *Mohamed Ben Ali Yamina* *Pr* *UBMA*

## Jury de Soutenance :

| | | | |
|---|---|---|---|
| Azizi Nabiha | Pr | UBMA | Président |
| Mohamed Ben Ali Yamina | Pr | UBMA | Encadrant |
| Zenakhra Djamel | Dr | UBMA | Examinateur |

**Année Universitaire : 2020/2021**

# REMERCIEMENTS

*Je remercie « ALLAH » le tout puissant, qui ma donnée la foi, la force et la patience pour aller jusqu'au bout de ce travail*

*La réalisation de ce mémoire a été possible grâce au concours de plusieurs personnes à qui je voudrais témoigner toute ma reconnaissance.*

*Je voudrais tout d'abord adresser toute ma gratitude à mon encadreur de mémoire Mme Mohamed Ben Ali Yamina, pour sa patience, sa disponibilité et surtout ses judicieux conseils, qui ont contribué à alimenter ma réflexion.*

*Je désir aussi remercier les professeurs, qui m'ont fourni les outils nécessaires à la réussite de mes études universitaires, Je tiens à remercier spécialement mrs.Znakhra qui ma aider beaucoup en cour de cette memoire,*

*Mes sincères remerciements vont également aux membres du jury pour l'intérêt qu'ils ont accepté d'examiner mon travail et de l'enrichir par leurs propositions*

*Je voudrais exprimer ma reconnaissance envers les amis et collègues qui m'ont apporté leur support moral et intellectuel tout au long de ma démarche. Un grand merci à Talaa Ihab pour les conseils concernant la base de données, ils ont grandement facilité mon travail.*

# DEDICACES

*A mes chers parents, pour tous leurs sacrifices, leur amour, leur tendresse, leur soutien et leurs prières tout au long de mes études,*

*A ma sœur Sabrina pour leurs encouragements permanents, et leur soutien moral,*

*A mon frère Lounis pour leur appui et leur encouragement,*

*A toute ma famille pour leur soutien tout au long de mon parcours universitaire,*

*À mes amis(Hamza Zarzouni, Akrem Yousfi,Aoufi Dia et Saidi Aymen )Pour leur affection et leur soutien*

# TABLE DES MATIERES

## Table des matières

# TABLE DES ILLUSTRATIONS

# RESUME

La classification multi-label est une extension de la classification traditionnelle dans laquelle les classes ne sont pas mutuellement exclusives, chaque individu pouvant appartenir à plusieurs classes simultanément. Dans ce travail, nous proposons une approche qui contient un classifieur pour Classifier et prédire la position des protéines dans le corp humain, l'approche proposée est une classification multi-label en utilisant l'algorithme par transformation basé sur la régression logistique (binary relavence). La classification est composée de 4 étapes : l'étape de traitement des données, l'étape d'apprentissage de classifieur , l'étape de test ou on fait la prédiction de la position et la dernière étape est de évalué le classifieur (accuracy ,hamming loss..etc ). Les expérimentations ont montré que notre approche est efficace conservant des taux d'erreur de classification très faible est une stabilité très satisfaisante.

**Mots clés** : [Apprentissage automatique, classification multi-label, régression logistique,bio-informatique, cellule, pdoc, matrice de confusion, position …]

# ملخص

التصنيف متعدد العلامات هو امتداد للتصنيف التقليدي حيث لا تكون الفئات متعارضة ، حيث يكون كل فرد قادرًا على الانتماء إلى عدة فئات في وقت واحد. في هذا العمل ، نقترح نهجًا يحتوي على مصنف لتصنيف وتوقع موضع (موقع) البروتينات في جسم الإنسان ، والنهج المقترح هو تصنيف متعدد العلامات باستخدام الخوارزمية عن طريق التحويل بناءً على الانحدار اللوجستي (الصلة الثنائية) . يتكون التصنيف من 4 مراحل: مرحلة معالجة البيانات ، ومرحلة تعلم المصنف ، ومرحلة الاختبار حيث نقوم بالتنبؤ بالموقع ، والمرحلة الأخيرة هي تقييم المصنف (الدقة ، خسارة الطرق .. الخ). أظهرت التجارب أن نهجنا فعال، مع الحفاظ على معدلات خطأ تصنيف منخفضة للغاية واستقرار مرضٍ للغاية.

# INTRODUCTION GENERALE

## Introduction

La bio-informatique est une science à l'interface des disciplines numériques (l'informatique et les mathématiques) et des sciences de la vie (biochimie, biologie, microbiologie, écologie, épidémiologie). Étant donné que les scientifiques de la vie génèrent une quantité croissante de nouvelles données portant sur les génomes, les biomolécules, les organismes, leurs interactions et leur évolution, il y a un besoin croissant d'approches informatiques pour la manipulation, le stockage, la visualisation et l'analyse de ces données souvent très complexes.

Également, la bio-informatique joue un rôle important pour la recherche biomédicale. Les travaux sur les maladies génétiques et la génomique médicale sont en pleine croissance et l'avenir d'une médecine personnalisée dépend des approches de la bio-informatique. Par conséquent, les perspectives pour trouver un emploi sont excellentes pour les bio-informaticiens.

## Contexte du projet

Le développement de la technologie et l'explosion des informations partagées, a conduit à l'émergence d'une grande quantité de données dans plusieurs champs de travail et de différent type structurées, non structurées et inutilisées, en conséquence un nombre de techniques et méthodes pour traiter, explorer, classer, structurer et génère des nouvelles connaissances à partir de ces données sont apparues au cours des dernières années. Parmi ces techniques, la classification multi-label.

# Problématique

Plusieurs technique et méthode sont appliquer dans le domaine de la bio-informatique pour la classification et la prédiction des fonctions, domaine et position des protéines .Dans ce projet nous introduisons une idée pour construire un programme qui peuvent utiliser et des informations donner dans une base nomme genbase pour faire la classification, la prédiction sur les positions des protéines dans le corps humain.

# Objectifs

L'objectif principal de notre travail est comment classer les protéines en fonction de leur positionnement en utilisant la classification multi-label spécifiquement la régression logistique (binary relavence). Ce travail est basé sur des expérimentes des algorithmes de classification multi-label dans le domaine de classification des textes.

# Contenu du mémoire

Outre la partie introductive et la conclusion générale, le travail est organisé en 4 chapitres :

- Le 1er chapitre est consacré à l'apprentissage automatique (machine learning) ces approches et ces algorithmes .aussi en a mentionné les méthodes de classification en général.

- Dans le 2eme chapitre, on passera à une présentation détaillée sur la classification multi-label sa définition, ces approches et les règle de l'évaluation de cette dernière.

- Dans Le 3eme chapitre nous présentons les notions de base de la biologie ADN, ARN, protéines ou en a motionné les domaines et les défirent position des protéines dans le corps humain, aussi on a parlé de la bio-informatique sa définition, ces objectifs, etc.

- Dans Le dernier chapitre nous présentons la conception et l'implémentation en détails, nous validons la méthode de classification mentionnée dans le 2eme chapitre et nous montrons l'efficacité de notre travail avec des captures d'écrans et des fragments de code source, en expliquant chaque point.

# CHAPITRE 1
# APPRENTISSAGE AUTOMATIQUE ET LA CLASSIFICATION

L'apprentissage automatique est une discipline qui consiste à appliquer des algorithmes à des jeux de données afin d'en extraire des modèles. Ceux-ci peuvent à leur tour être appliquées sur des données similaires à des fins de prédiction. Avec suffisamment de données, il est possible de formuler une approximation de la relation entre toutes les variables d'entrée et les valeurs particulières dites « cible ». On peut alors appliquer cette formule sur de nouvelles données d'entrée pour prévoir la valeur cible associée. Cette approche diffère de la programmation conventionnelle où une application est développée à partir de règles préalablement définies. Même si les concepts fondamentaux de l'apprentissage automatique existent depuis un certain temps, le domaine a récemment pris de l'ampleur en raison d'une part de l'amélioration de la performance des processeurs (en particulier graphiques), et d'autre part grâce à la disponibilité de grandes quantités d'information. Ces deux composantes qui sont essentielles à l'obtention de prévisions précises. Étant donné qu'il existe déjà suffisamment de littérature sur l'histoire de l'apprentissage automatique.

## I. Apprentissage automatique

### 1- Définition :

L'**apprentissage automatique**. « **apprentissage machine**», est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'« apprendre » à partir de données, c'est-à-dire d'améliorer leurs performances à résoudre des tâches sans être explicitement programmés pour chacune. Plus largement, il concerne la conception, l'analyse, l'optimisation, le développement et l'implémentation de telles méthodes.

L'apprentissage automatique peut être appliqué à différents types de données, tels des graphes, des arbres, des courbes, ou plus simplement des vecteurs de caractéristiques, qui peuvent être des variables qualitatives ou quantitatives continues ou discrètes.

L'apprentissage automatique est très efficace dans les situations où des informations doivent être découvertes à partir de grands ensembles de données diverses et changeantes, par exemple : Big Data. Pour l'analyse de telles données, elle est bien plus efficace que les méthodes traditionnelles en termes de précision et de rapidité. [WEB 1]

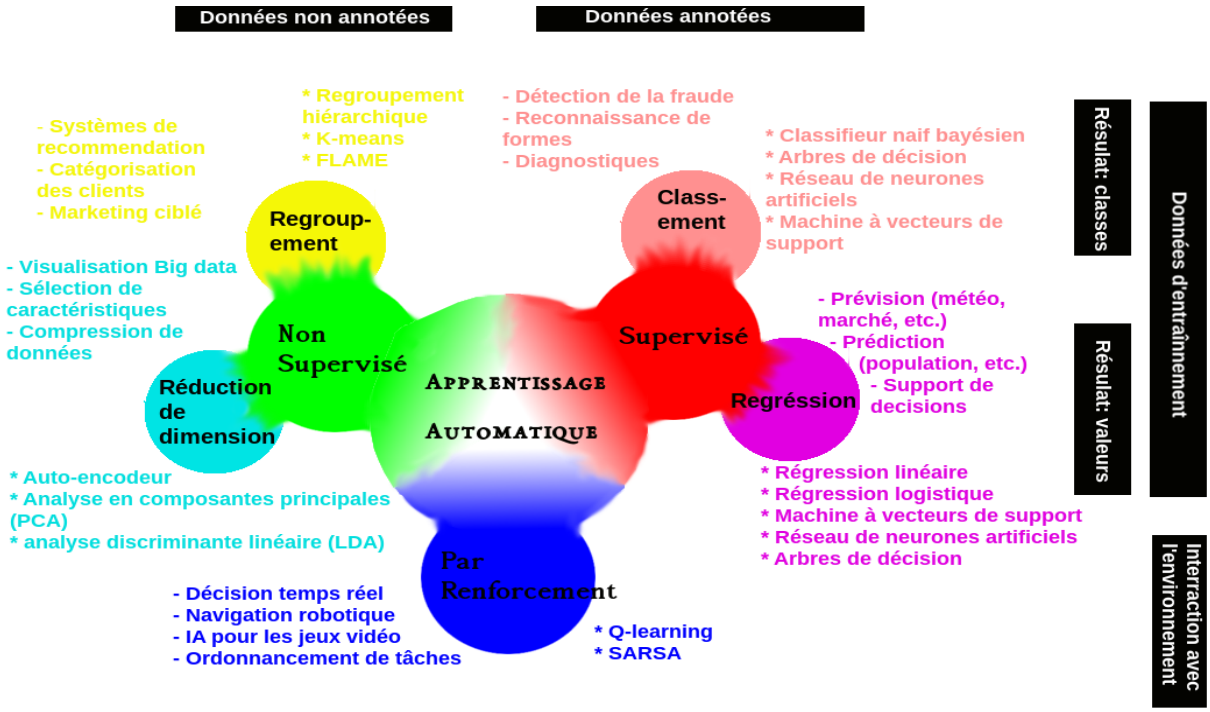



**Figure 01 :** Apprentissage automatique [WEB 2]

## 2- les phases de l'apprentissage automatique :

En général, l'apprentissage automatique se compose de 2 phases :

- La première phase est la conception du système qui est aussi appelée phase d'apprentissage ou d'apprentissage et l'estimation d'un modèle à partir de l'analyse des données. Cela comprend l'estimation d'une densité de probabilité ou la résolution d'une tâche pratique telle que la traduction d'un discours.
- La deuxième phase est un démarrage de la production. Les systèmes peuvent continuer à apprendre même lorsqu'ils sont déjà en production. Après détermination du modèle, la deuxième partie des données utiles à la réalisation de la tâche souhaitée est testée.

## 3- Approche de l'apprentissage automatique :

Des techniques d'apprentissage automatique sont nécessaires pour améliorer la précision des modèles prédictifs. Selon les informations disponibles lors de la phase d'apprentissage, l'apprentissage est qualifié de différentes manières. Si les données sont étiquetées (c'est-à-dire que la réponse à la tâche est connue pour ces données), il s'agit d'un apprentissage supervisé. On parle de classement ou de classement si les labels sont discrets, ou de régression s'ils sont continus. Si le modèle est appris de manière incrémentale en fonction d'une récompense reçue par le programme pour chacune des actions entreprises, on parle d'apprentissage par renforcement. Dans le cas le plus général, sans étiquette, on cherche à déterminer la structure sous-jacente des données (qui peut être une densité de probabilité) et il s'agit alors d'un apprentissage non supervisé.

### 3.1- Apprentissage supervisé :

L'apprentissage supervisé (supervised learning en anglais) est une tâche d'apprentissage automatique consistant à apprendre une fonction de prédiction à partir d'exemples annotés, au contraire de l'apprentissage non supervisé. On distingue les problèmes de régression des problèmes de classement. Ainsi, on considère que les problèmes de prédiction d'une variable quantitative sont des problèmes de régression tandis que les problèmes de prédiction d'une variable qualitative sont des problèmes de classification.

Les exemples annotés constituent une base d'apprentissage, et la fonction de prédiction apprise peut aussi être appelée « hypothèse » ou « modèle ». On suppose cette base d'apprentissage représentative d'une population d'échantillons plus large et le but des méthodes d'apprentissage supervisé est de bien généraliser, c'est-à-dire d'apprendre une fonction qui fasse des prédictions correctes sur des données non présentes dans l'ensemble d'apprentissage. [WEB 3]

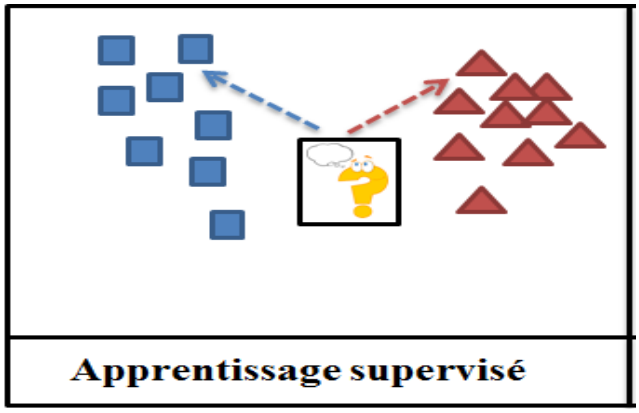



**Figure 02 :** Apprentissage supervisé [WEB 4]

### 3.2- Apprentissage non supervisé :

L’apprentissage non supervisé désigne la situation d'apprentissage automatique où les données ne sont pas étiquetées. Il s'agit donc de découvrir les structures sous-jacentes à ces données non étiquetées. Puisque les données ne sont pas étiquetées, il est impossible à l'algorithme de calculer de façon certaine un score de réussite.

L'absence d'étiquetage ou d'annotation caractérise les tâches d'apprentissage non-supervisé et les distingue donc des tâches d'apprentissage supervisé. [WEB 5]

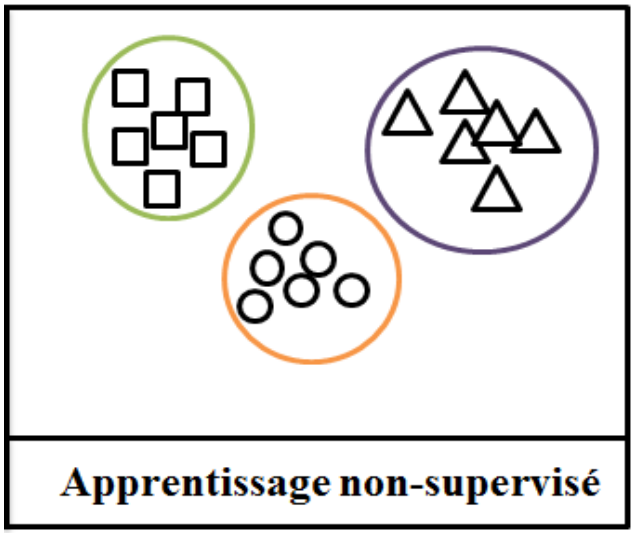



**Figure 03 :** Apprentissage non-supervisé [WEB 4]

### 3.3- Apprentissage semi-supervisé :

L'apprentissage semi-supervisé est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données étiquetées et non étiquetées. Il se situe ainsi entre l'apprentissage supervisé qui n'utilise que des données étiquetées et l'apprentissage non supervisé qui n'utilise que des données non étiquetées.

Il a été démontré que l'utilisation de données non étiquetées, en combinaison avec des données étiquetées, permet d'améliorer significativement la qualité de l'apprentissage.[WEB 6]

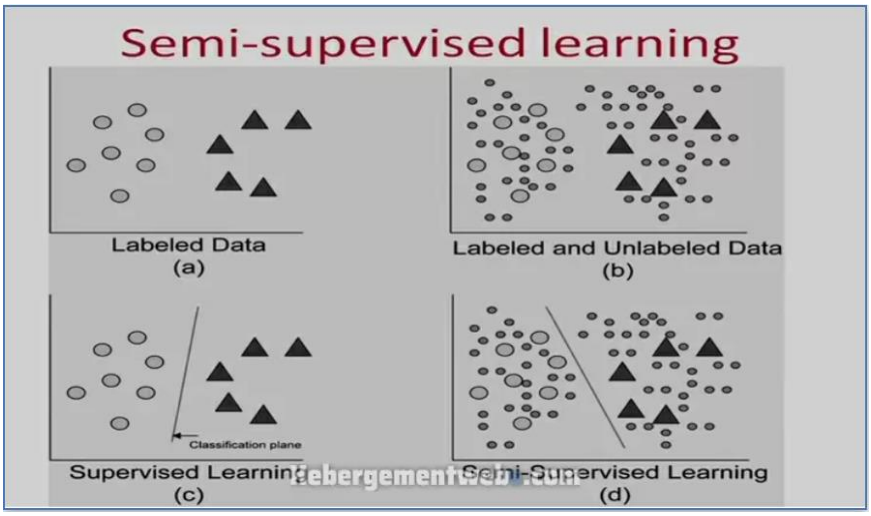



**Figure 04 :** Apprentissage semi-supervisé [WEB 7]

### 3.4- Apprentissage par renforcement :

L'apprentissage par renforcement consiste, pour un agent autonome à apprendre les actions à prendre, à partir d'expériences, de façon à optimiser une récompense quantitative au cours du temps. L'agent est plongé au sein d'un environnement, et prend ses décisions en fonction de son état courant. En retour, l'environnement procure à l'agent une récompense, qui peut être positif ou négatif. L'agent cherche, au travers d'expériences itérées, un comportement décisionnel (appelé stratégie ou politique, et qui est une fonction associée à l'état courant l'action à exécuter) optimal, en ce sens qu'il maximiser la somme des récompenses au cours du temps. [WEB 8]

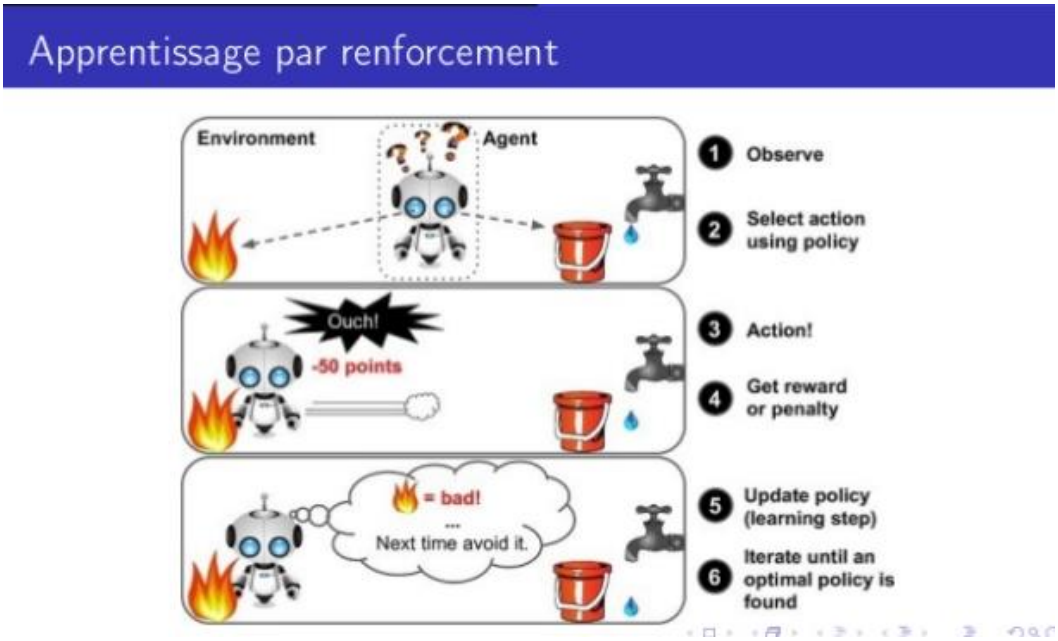



**Figure 05 :** Apprentissage par renforcement [WEB 9]

### 3.5- Apprentissage par transfert :

L'apprentissage par transfert (transfer learning en anglais) est l'un des champs de recherche de l'apprentissage automatique qui vise à transférer des connaissances d'une ou plusieurs tâches sources vers une ou plusieurs tâches cibles. Il peut être vu comme la capacité d'un système à reconnaître et appliquer des connaissances et des compétences, apprises à partir de tâches antérieures, sur de nouvelles tâches ou domaines partageant des similitudes. [WEB 10]

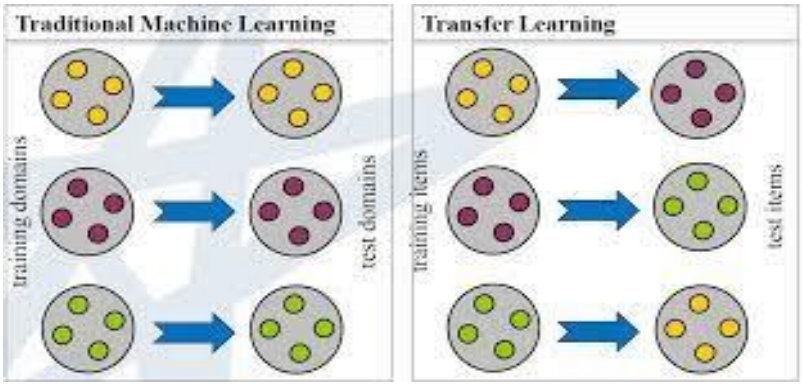



**Figure 06 :** Apprentissage par transfert [hal archive 1]

## 4- Algorithmes utilisés :

**4.1- Machine à vecteurs de support (SVM):** Une machine à vecteurs de support (SVM) est un algorithme d'apprentissage automatique qui analyse les données pour la classification et l'analyse de régression. SVM est une méthode d'apprentissage supervisé qui examine les données et les trie dans l'une des deux catégories. Un SVM génère une carte des données triées avec les marges entre les deux aussi éloignées que possible. Les SVM sont utilisées dans la catégorisation de texte, la classification d'images, la reconnaissance de l'écriture manuscrite et dans les sciences.

Une machine à vecteurs de support est également connue sous le nom de réseau de vecteurs de support (SVN). [WEB 11]

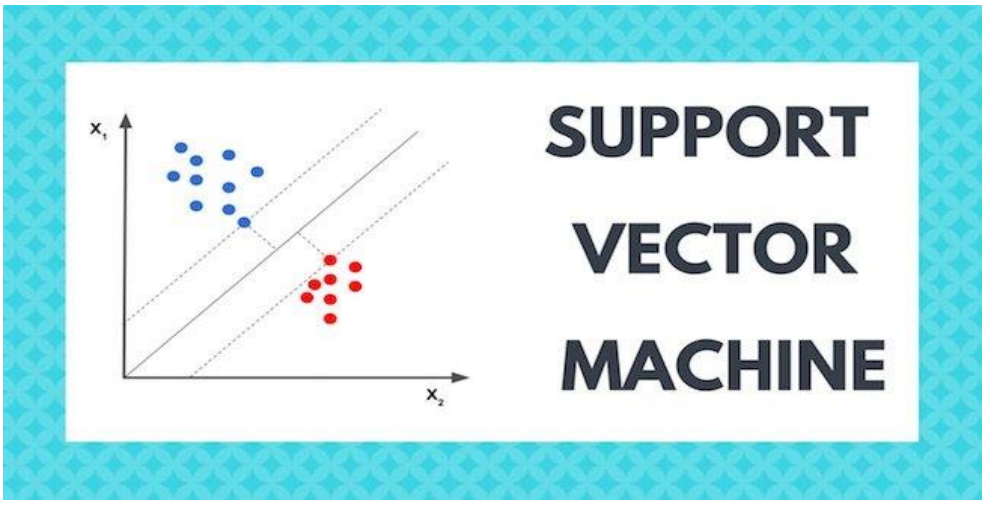



**Figure 07 :** Machine à vecteurs de support (SVM) [WEB 12]

**4.2- Les réseaux de neurones :** Un réseau de neurones artificiels est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement des neurones biologiques.

Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type probabiliste, en particulier bayésiens. Ils sont placés d'une part dans la famille des applications statistiques, qu'ils enrichissent avec un ensemble de paradigmes permettant de générer des classifications rapides (réseaux de Kohonen en particulier), et d'autre part dans la famille des méthodes de l'intelligence artificielle auxquelles ils fournissent un mécanisme perceptif indépendant des idées propres de l'implémenteur, et fournissant des informations d'entrée au raisonnement logique formel. [WEB 13]

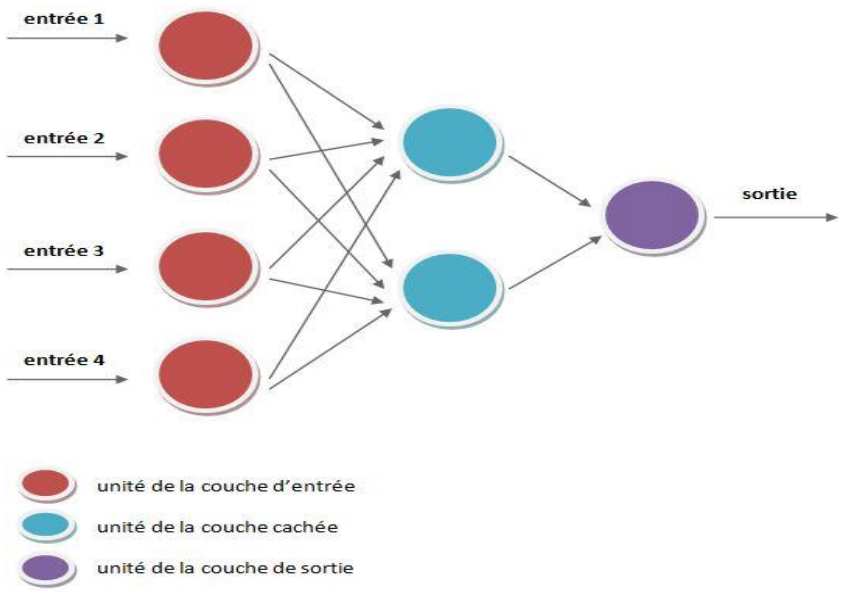



**Figure 08 :** Réseaux de neurones [WEB 14]

**4.3- Les arbres de décision :** Les arbres de décision (AD) sont une catégorie d'arbres utilisée dans l'exploration de données et en informatique décisionnelle. Ils emploient une représentation hiérarchique de la structure des données sous forme des séquences de décisions (tests) en vue de la prédiction d'un résultat ou d'une classe. Chaque individu (ou observation), qui doit être attribué(e) à une classe, est décrit(e) par un ensemble de variables qui sont testées dans les nœuds de l'arbre. Les tests s'effectuent dans les nœuds internes et les décisions sont prise dans les nœuds feuille. [WEB 15]

### 5- Etapes d’apprentissage automatique :

L'apprentissage automatique ne se résume pas à un ensemble d'algorithmes, mais suit une succession d'étapes :

1. Définir le problème à résoudre
2. Analyser et explorer les données
3. Extraction de caractéristiques
4. Choisir ou construire un modèle d’apprentissage
5. Entrainer, évaluer et optimiser
6. Test
7. Déployer

# II. Classification

### 1- Définition:

La classification est un processus de catégorisation d'un ensemble donné de données en classes. Elle peut être effectuée sur des données structurées ou non structurées. Le processus commence par prédire la classe de points de données donnés. Les classes sont souvent appelées cible, étiquette ou catégories.

La modélisation prédictive de classification est la tâche d'approximation de la fonction de mappage des variables d'entrée aux variables de sortie discrètes. L'objectif principal est d'identifier dans quelle classe/catégorie les nouvelles données entreront. [WEB 16]

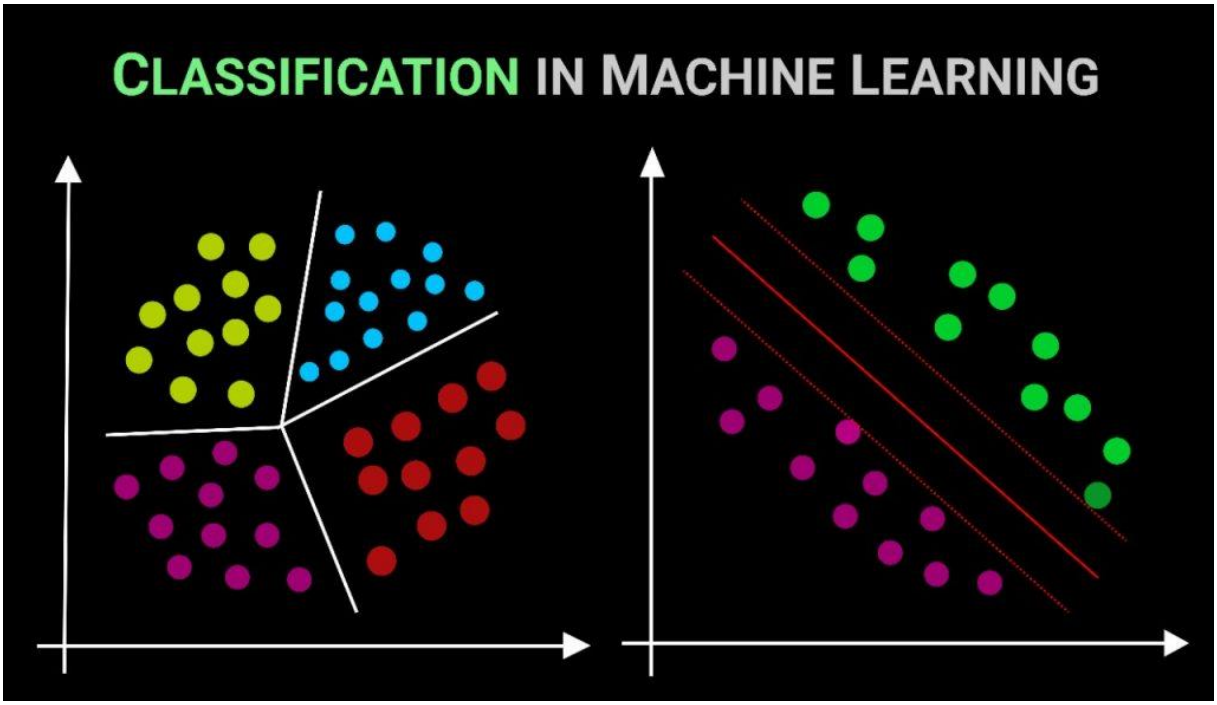



**Figure 09 :** Classification dans l'apprentissage automatique [WEB 17]

## 2- Types de classification en apprentissage automatique:

Les Types de classification en apprentissage automatique sont :

### 2.1- Classification Binaire (Binary Classification):

La classification binaire fait référence aux tâches de classification qui ont deux étiquettes de classe.

Les exemples comprennent:

- Détection de spam par courrier électronique (spam ou non).
- Prédiction de désabonnement (attrition ou non).
- Prédiction de conversion (acheter ou non).

En règle générale, les tâches de classification binaire impliquent une classe qui est l'état normal et une autre classe qui est l'état anormal.

Par exemple, "pas de spam" est l'état normal et "spam" est l'état anormal. Un autre exemple est « cancer non détecté » est l'état normal d'une tâche qui implique un test médical et « cancer détecté » est l'état anormal.

La classe pour l'état normal reçoit l'étiquette de classe 0 et la classe avec l'état anormal reçoit l'étiquette de classe 1.

Les algorithmes populaires qui peuvent être utilisés pour la classification binaire incluent :

- Régression logistique
- k-Les voisins les plus proches
- Arbres de décision
- Machine à vecteur de soutien
- Naïf Bayes

Certains algorithmes sont spécifiquement conçus pour la classification binaire et ne prennent pas en charge nativement plus de deux classes ; les exemples incluent la régression logistique et les machines à vecteurs de support. [WEB 18]

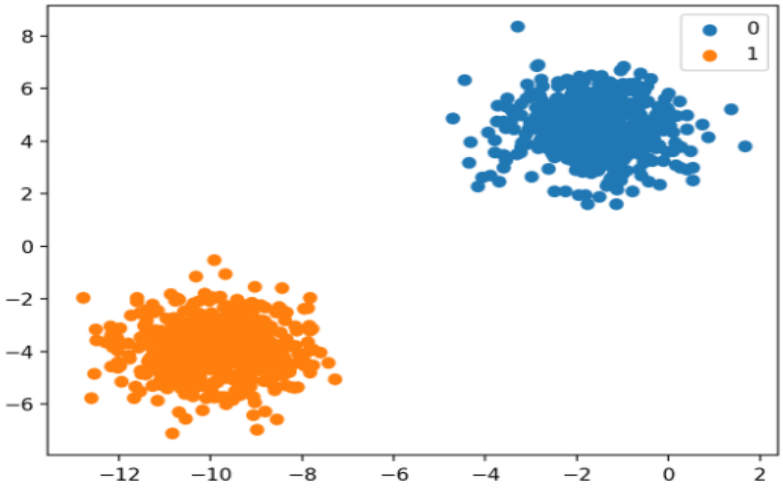



**Figure 10 :** Exemple de classification binaire [WEB 18]

## 2.2- Classification multi-classes:

La classification multi classe fait référence aux tâches de classification qui ont plus de deux étiquettes de classe.

Les exemples comprennent:

- Classement des visages.
- Classification des espèces végétales.
- Reconnaissance optique de caractères.

Contrairement à la classification binaire, la classification multi classe n'a pas la notion de résultats normaux et anormaux. Au lieu de cela, les exemples sont classés comme appartenant à l'une parmi une gamme de classes connues.

Le nombre d'étiquettes de classe peut être très important sur certains problèmes. Par exemple, un modèle peut prédire qu'une photo appartient à un parmi des milliers ou des dizaines de milliers de visages dans un système de reconnaissance faciale.

Les problèmes qui impliquent de prédire une séquence de mots, tels que les modèles de traduction de texte, peuvent également être considérés comme un type spécial de classification multi classe. Chaque mot de la séquence de mots à prédire implique une classification multi-classes où la taille du vocabulaire définit le nombre de classes possibles qui peuvent être prédites et peut avoir une taille de dizaines ou de centaines de milliers de mots.

Les algorithmes populaires qui peuvent être utilisés pour la classification multi-classes incluent :

- k-Les voisins les plus proches.
- Arbres de décision.
- Naïf Bayes.
- Forêt aléatoire.
- Renforcement des dégradés.

Les algorithmes conçus pour la classification binaire peuvent être adaptés pour être utilisés pour des problèmes multi-classes. [WEB 18]

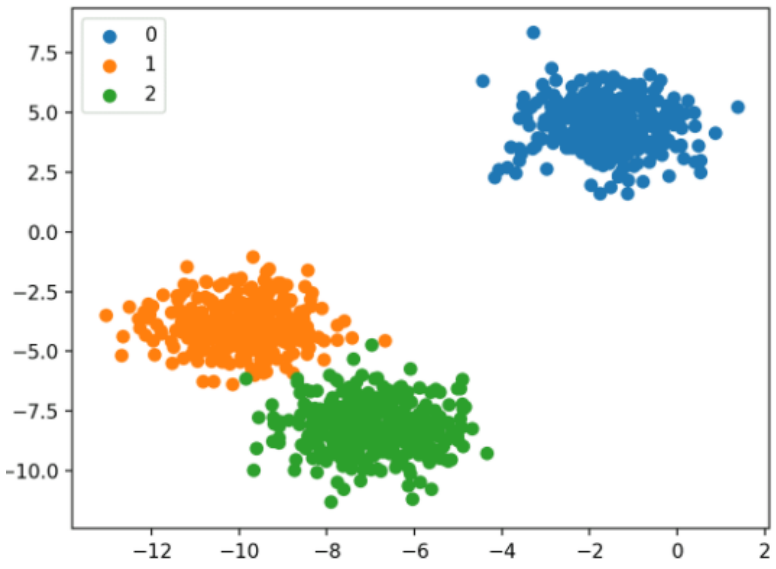



**Figure 11 :** Exemple de classification multi-classes [WEB 18]

Cela implique l'utilisation d'une stratégie d'ajustement de plusieurs modèles de classification binaire pour chaque classe par rapport à toutes les autres classes (appelé un contre un repos) ou un modèle pour chaque paire de classes (appelé un contre un).

### 2.2.1- One Vs Rest (One Vs All):

La stratégie One Vs Rest implique la formation d'un seul classificateur par classe, avec les échantillons de cette classe comme échantillons positifs et tous les autres échantillons comme négatifs. Cette stratégie nécessite que les classificateurs de base produisent un score de confiance à valeur réelle pour sa décision, plutôt qu'une simple étiquette de classe ; les étiquettes de classe discrète seules peuvent conduire à des ambiguïtés, où plusieurs classes sont prédites pour un seul échantillon. [WEB 19]

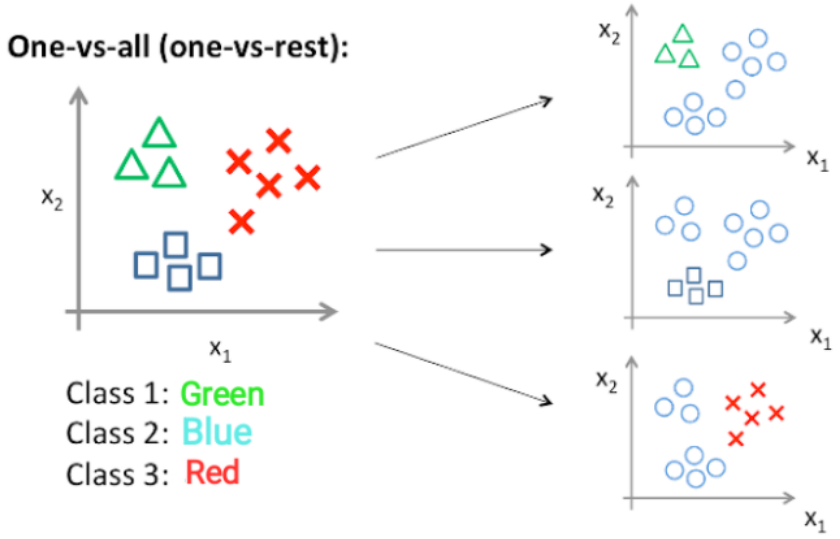



**Figure 12 :** Exemple de classification multi-classes de stratégie one vs all [WEB 19]

### 2.2.2- One Vs One:

Dans la réduction un contre un (OvO), on forme K (K − 1) / 2 classificateurs binaires pour un problème multi classe à K ; chacun reçoit les échantillons d'une paire de classes de l'ensemble d'apprentissage original, et doit apprendre à distinguer ces deux classes. Au moment de la prédiction, un schéma de vote est appliqué : tous les classificateurs K (K − 1) / 2 sont appliqués à un échantillon invisible et la classe qui a obtenu le plus grand nombre de prédictions « +1 » est prédite par le classificateur combiné. [WEB 19]

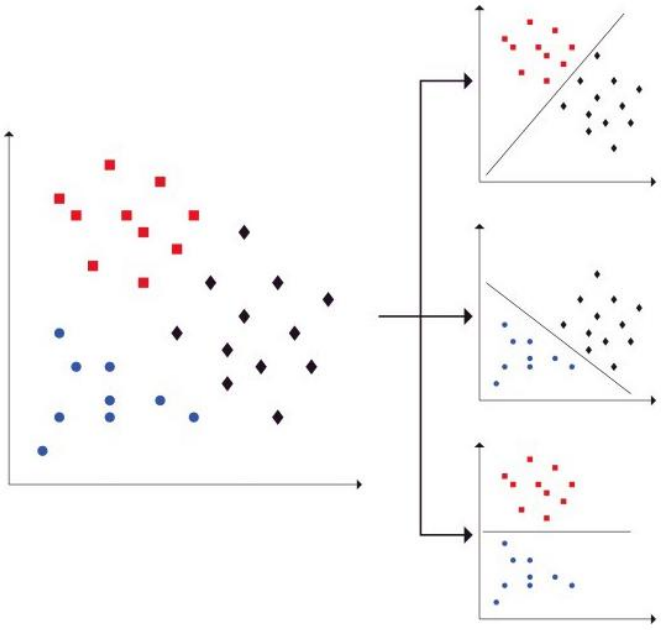


**Figure 13 :** Exemple de classification multi-classes de stratégie one vs one [WEB 19]

### 2.3- Classification multi-label:

Dans l'apprentissage automatique, la classification multi-label et le problème fortement lié de la classification multi-sorti sont des variantes du problème de classification où plusieurs étiquettes peuvent être attribuées à chaque instance. La classification multi-label est une généralisation de la classification multi-classe, qui est le problème en une seule étiquette de la catégorisation des instances précisément dans l'une de plus de deux classes; dans le problème multi-label, il n'y a aucune contrainte sur le nombre de classes auxquelles l'instance peut être affectée.

Formellement, la classification multi-label est le problème de trouver un modèle qui mappe les entrées x aux vecteurs binaires y (attribuant une valeur de 0 ou 1 pour chaque élément (étiquette) dans y). [WEB 20]

# Conclusion

Dans ce chapitre, nous avons passé en revue quelques méthodes et concepts liés au traitement des données et plus spécifiquement à l'apprentissage automatique, sa définition et ses approches puis nous avons parlé de classification. où nous avons répertorié les différents types de celui-ci. on a conclus qu'il existe plusieurs méthodes de classification différentes qui vous permettent de traiter et de comprendre de grandes quantités de données et de prévoir et d'acquérir de nouvelles connaissances à partir de ces données.

# CHAPITRE 2
# LA CLASSIFICATION MULTI-LABEL

L'apprentissage multi-label a des applications importantes dans de nombreux problèmes du monde réel comme la catégorisation de texte, la classification de scènes, l'annotation vidéo et la bio-informatique, où la tâche consiste à prédire pour un exemple un ensemble d'étiquettes dont la taille est a priori inconnue. Il est clair que le problème d'étiquette unique peut être considéré comme un cas particulier du problème d'apprentissage multi-label.

L'apprentissage multi-label est lié au classement multi-label. Le classement multi-label est la tâche d'apprendre une correspondance entre les instances et les classements sur l'ensemble de labels, de sorte que les labels pertinents soient mieux classés que les non pertinents.

## I. Classification multi-label (L'apprentissage multi-label)

### 1- Définition :

La classification multi-étiquette multi-label est la tâche d'affecter à une instance d'entrée plusieurs classes simultanément à partir d'un ensemble de classes disjointes ; les classes ne sont alors plus mutuellement exclusives. Dans ce contexte, on utilise souvent le terme "label" au lieu de "classe". Chaque instance est associée à un ensemble de labels pertinents. Les autres labels sont considérés comme non pertinentes. Contrairement à la classification mono- label, la tâche multi- labels est influencée par des corrélations latentes intrinsèques entre les étiquettes, dans le sens où l'appartenance d'une instance à une classe peut être utile pour prédire son ensemble d'étiquettes. Par exemple, un patient souffrant d'hypertension artérielle est plus susceptible de développer une maladie cardiaque qu'une autre personne, mais moins susceptible de développer une dystrophie musculaire. [hal archive 2]


**Figure 14 :** Classification multi-label [WEB 21]

## 2-Approches de l'apprentissage multi-label:

De nombreuses méthodes ont été proposées dans la littérature pour traiter les problèmes d'apprentissage multi-label. Les algorithmes existants peuvent être regroupés en trois catégories :

- **Les approches de transformation de problèmes :** cette méthode divise le problème multi-étiquettes en un ou plusieurs problèmes classiques à une seule étiquette.
- **Les algorithmes d'adaptation de problèmes :** cette catégorie généralise les algorithmes mono-étiquettes pour traiter directement les données multi-étiquetées.
- **Les méthodes d'ensemble :** cette catégorie intègre les mérites de ces deux approches précédentes.

Dans le paragraphe suivant, nous allons expliquer plus en détail chacune de ces catégories et décrire les principales caractéristiques des méthodes correspondantes. La figure 15 montre les différentes catégories et méthodes associées.

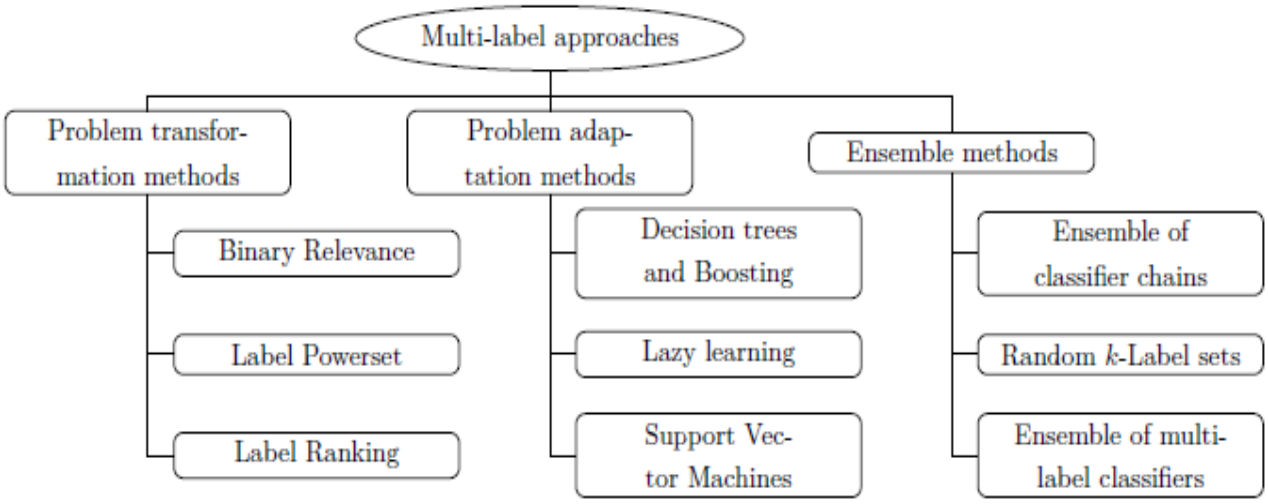



**Figure 15 :** Approches d'apprentissage multi-labels [hal archive 2]

### 2.1- Méthodes de transformation des problèmes:

La stratégie la plus simple dans l'apprentissage multi- labels est l'approche de transformation de problème qui peut être utilisée avec n'importe quel algorithme d'apprentissage. Dans ces méthodes, le problème de classification multi-label est transformé en un ou plusieurs problèmes de classification mono-label. Les solutions de ces problèmes sont ensuite combinées pour résoudre la tâche originale de l'apprentissage multi-label. Les méthodes de transformation des problèmes comprennent trois approches principales : pertinence binaire, ensemble de puissance des étiquettes et classement des labels. [hal archive 2]

#### 2.1.1- Pertinence binaire (Binary relevance BR): (L'approche utiliser dans notre travail)

L'approche de pertinence binaire (BR), également connue sous le nom de stratégie un contre tous, divise le problème d'apprentissage multi- labels avec Q classes possibles en Q problèmes de classification à une seule étiquette qui peuvent être résolus en entraînant Q classificateurs binaires (h1, . . ..$h_Q$) . Chaque classificateur q (q Є{1, . . .. Q}) est entraîné sur l'ensemble de données d'origine, et vise à déterminer la pertinence de son étiquette particulière pour une instance donnée. Lors de la classification d'une nouvelle instance x, BR produit l'union des étiquettes prédites positivement par les classificateurs binaires. Le classifieur multi-label est alors déterminé par : H(x) = {!q Є {Y|hq(x) = 1}. De nombreuses méthodes de classification classiques bien connues sont généralisées pour traiter le problème multi-étiquettes en utilisant l'approche BR, par exemple, les arbres de décision, SVM et k-NN. L'approche BR est simple à mettre en œuvre, et sa complexité est linéaire avec le nombre

d'étiquettes possibles. Cependant, BR ignore la corrélation entre les étiquettes en traitant chaque étiquette indépendamment. [hal archive 2]

**Observation :** Pour traiter les aspects négatifs de BR, la chaîne de classification (CC) introduite dans implique Q classificateurs binaires liés le long d'une chaîne. L'approche Label Powerset présentée dans la section suivante constitue également l'une des alternatives pour traiter ces aspects négatifs de l'approche BR.

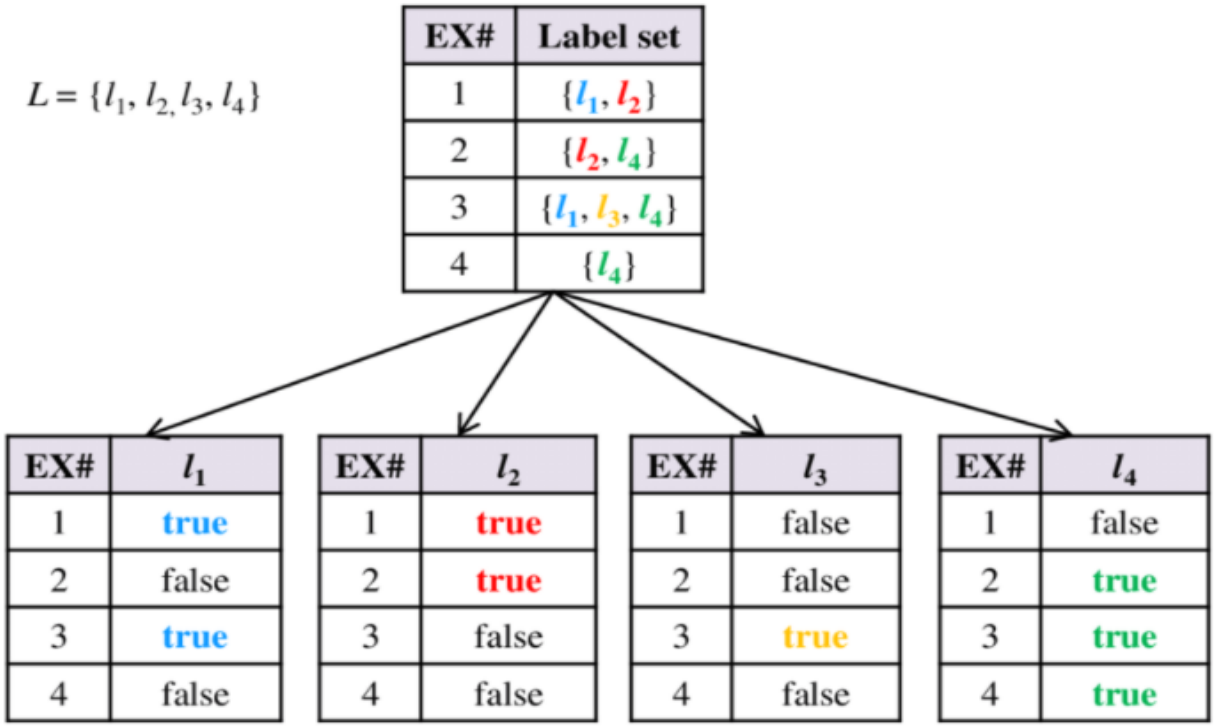



**Figure 16 :** Méthode de transformation de problème de pertinence binaire [WEB 22]

### 2.1.2- Ensemble de puissance d'étiquette (Label Powerset LP) :

Étant donné un ensemble de données d'apprentissage D avec n instances, l'approche Label Powerset (LP) considère chaque ensemble unique d'étiquettes dans D comme une seule étiquette, puis forme un classificateur à une seule étiquette. Le nombre de classes est majoré par $\min(2^Q, n)$. La complexité de LP repose sur la complexité du classifieur mono-étiquette par rapport au nombre de classes. Pour une nouvelle instance, l'approche LP génère la classe la plus probable qui est un ensemble d'étiquettes dans la représentation multi-étiquette d'origine. LP a l'avantage de prendre en compte les corrélations d'étiquettes. De plus, un aspect négatif de cette approche est qu'elle peut conduire à des ensembles de données déséquilibrés avec un grand nombre de classes associées à peu d'exemples. Ce problème est résolu en ne considérant que les combinaisons d'étiquettes fréquemment trouvées dans D

comme valeurs de classe pour le classificateur à une seule étiquette. Cette nouvelle approche est appelée méthode de transformation élaguée.

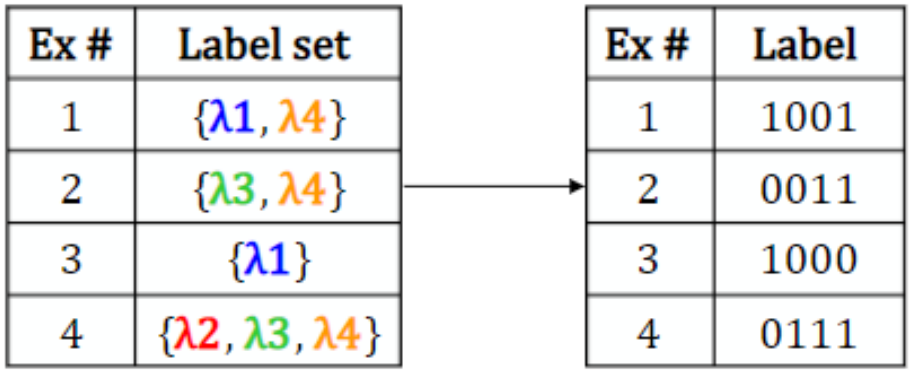



**Figure 17 :** Méthode de transformation de problème de Label Powerset LP [WEB 23]

### 2.1.3- Classement des étiquettes (Label Ranking (LR)):

L'approche Label Ranking (LR) apprend une correspondance entre les instances et les classements sur toutes les étiquettes possibles. Étant donné un classement de pertinence par rapport aux étiquettes pour une instance, la classification à une seule étiquette sélectionne l'étiquette (ou la classe) la plus pertinente pour cette instance. En revanche, dans le cas de plusieurs étiquettes, les étiquettes les plus élevées, et pas seulement l'étiquette supérieure, sont liées à cette instance. Pour sortir l'ensemble d'étiquettes pour une instance invisible, l'approche LR divise l'ensemble d'étiquettes ordonné en étiquettes pertinentes et non pertinentes. Le but de l'apprentissage multi-label est donc de trouver une fonction de notation qui attribue un score ou une valeur à chaque couple. Pour déterminer les classes qui doivent être attribuées à une instance particulière, une valeur seuil est introduite. L'approche LR ne modélise pas explicitement les corrélations d'étiquettes. [hal archive 2]

## 2.2- Méthodes d'adaptation aux problèmes:

Les méthodes d'adaptation aux problèmes personnalisent les algorithmes d'apprentissage automatique traditionnels afin de gérer directement les concepts multi-labels. Ces méthodes ont l'avantage de se concentrer sur un algorithme spécifique. Un autre avantage est que ces méthodes utilisent l'ensemble de données d'apprentissage (instances et étiquettes) à la fois pour entraîner un classificateur multi-label. En général, les performances de ces algorithmes sont meilleures dans les problèmes difficiles du monde réel que celles des méthodes de transformation de problèmes, au prix d'une complexité plus élevée. Plusieurs adaptations d'algorithmes d'apprentissage traditionnels ont été proposées dans la littérature, certaines d'entre elles étendent toutes les étiquettes et instances simultanément, comme les arbres de décision et les machines à vecteurs de support multi-étiquettes, tandis que d'autres

considèrent chaque classe séparément, comme le k-plus proche multi-étiquettes, méthode des voisins (MLkNN). Ces méthodes sont brièvement passées en revue ci-dessous. [hal archive 2]

### 2.2.1- Arbres de décision et Boosting :

**Les arbres de décision** : sont parmi les méthodes de classification les plus populaires en apprentissage automatique. À partir de toutes les instances d'apprentissage, un arbre de décision crée un modèle prédictif ayant une structure arborescente, qui peut être considérée comme un partitionnement de l'ensemble de données d'apprentissage. Les nœuds de l'arborescence représentent des attributs qui sont connectés à des branches qui mènent à des nœuds enfants (partitions). Pour une instance invisible, la classe cible est prédite en suivant le chemin des nœuds et des branches du nœud racine à la feuille terminale. De nombreux algorithmes d'arbres de décision ont été étendus au domaine de classification multi-label. un algorithme multi-label est proposé en modifiant la formule d'entropie. Il a été évalué dans un premier temps sur des données multi-marquées issues de la génomique fonctionnelle.

**Le boosting** : Le boosting **a** été appliqué à l'apprentissage multi-label et en particulier à la catégorisation de textes. Les méthodes basées sur le boost incluent deux versions légèrement différentes de la méthode d'apprentissage d'ensemble AdaBoost (AdaBoost.MH et AdaBoost.MR). La première extension vise à prédire l'ensemble de toutes les étiquettes correctes pour une instance de requête, tandis que la seconde est conçue pour trouver un classificateur qui classe les étiquettes de sorte que les étiquettes correctes reçoivent les rangs les plus élevés.

**Observation :** AdaBoost.MH est combiné avec l'algorithme d'arbres de décision alternés pour produire l'Adapted Decision Tree Boosting, ADTBoost.MH.

### 2.2.2- Machines à vecteurs de support:

Les machines à vecteurs de support (SVM) ont été largement utilisées avec des méthodes de transformation de problèmes, en particulier avec l'approche BR, pour résoudre le problème d'apprentissage multi-étiquette. Cependant, à notre connaissance, une seule méthode a été proposée dans la littérature afin d'étendre l'algorithme SVM pour traiter directement le problème multi-label.

Rank-SVM définit un modèle linéaire basé sur un système de classement combiné à un prédicteur de taille d'ensemble d'étiquettes. Ce modèle est décomposé en deux parties. Le premier suit un système de classement, qui classe les étiquettes en fonction de leurs valeurs de

sortie. La seconde définit un prédicteur de taille d'ensemble (t(x)) qui peut servir de valeur seuil pour différencier les étiquettes pertinentes des autres.

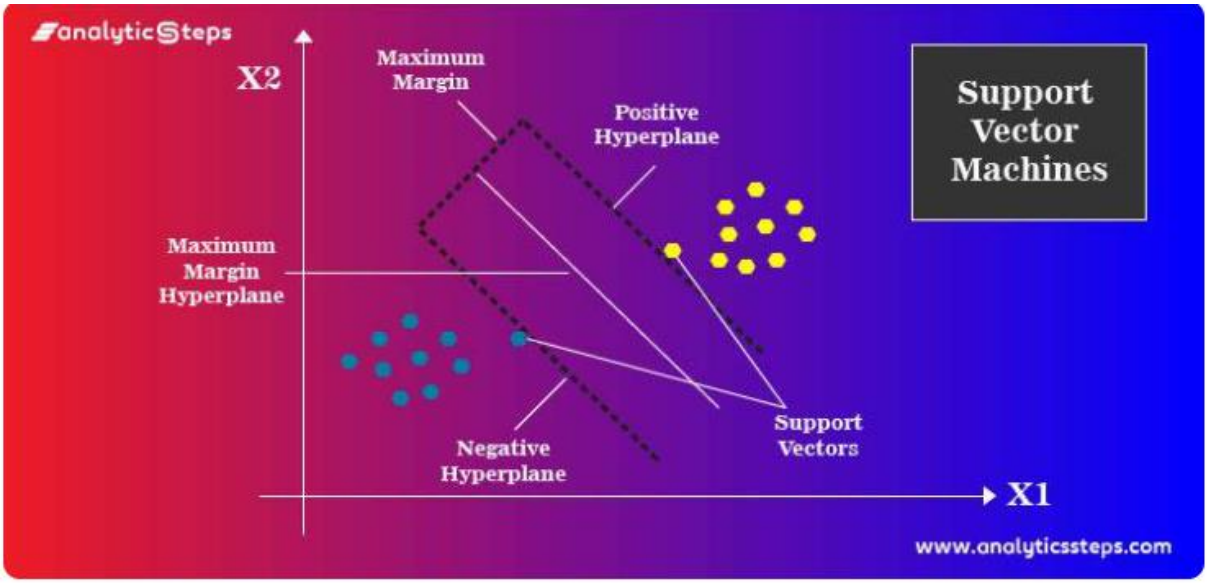



**Figure 18 :** fonctionnement de Support Vector Machine (SVM) dans l'apprentissage automatique [WEB 24]

### 2.2.3-Multi-Label-kNN (ML-kNN)

Les k-voisins les plus proches en multi-label (ML-kNN) sont une extension de l'algorithme des k-voisins les plus proches (kNN). Plusieurs variantes pour l'apprentissage multi-label (ML-kNN) de l'algorithme d'apprentissage k- Nearest Neighbors (kNN) ont été proposées. La récupération des ML-KNN est la même que dans l'algorithme kNN traditionnel.

ML-kNN est une méthode de type BR qui combine l'algorithme standard de kNN avec une inférence bayésienne. En phase d'apprentissage, ML-kNN estime les probabilités a priori et a posteriori de chaque label à partir des exemples d'apprentissage. Pour un nouvel exemple xi, ML-kNN calcule ses k plus proches voisins puis mesure la fréquence de chaque label dans ce voisinage. Cette fréquence est ensuite combinée avec les probabilités estimées dans la phase d'apprentissage pour déterminer son ensemble de labels en suivant le principe du maximum a posteriori (MAP).

## 2.3- Méthodes d'ensemble:

Les méthodes d'ensemble incorporent des classificateurs de transformation et d'adaptation de problème. Beaucoup de ces méthodes peuvent être agrégées pour produire un nouveau classificateur pour l'apprentissage multi-étiquettes. Les méthodes d'ensemble peuvent atténuer les inconvénients d'un classificateur de base en ajoutant un ensemble de classificateurs. Plusieurs méthodes d'ensemble ont été proposées, parmi lesquelles : ensemble

de chaînes de classificateurs, ensembles aléatoires de k-labels et ensemble de classificateurs multi-labels. [hal archive 2]

### 2.3.1- Ensemble de chaînes de classificateurs(ECC):

L'ensemble de chaînes de classificateurs (ECC) utilise des chaînes de classificateurs (CC) comme classificateurs de base. ECC a été proposé pour atténuer l'effet de l'ordre des classificateurs dans CC, en entraînant un ensemble de classificateurs CC, $C_1, \ldots, C_m$. Chaque Ck peut être entraîné avec un ordre de chaîne aléatoire sur un sous-ensemble aléatoire de D. Étant donné une instance invisible, les prédictions de différents classificateurs sont rassemblées et combinées pour chaque étiquette afin que chaque étiquette reçoive un certain nombre de votes. Pour sortir le jeu d'étiquettes multiples final, un seuil est utilisé pour sélectionner les étiquettes les plus pertinentes. En général, ECC surpasse BR et CC tout en maintenant une complexité de calcul acceptable.

### 2.3.2- Ensembles aléatoires de k-labEL:

La méthode RAndom k-labEl sets (RAKEL) est liée aux approches de transformation de problèmes et en particulier à l'approche LP [96]. RAKEL récupère le problème de LP présenté par le grand nombre d'étiquettes avec peu d'exemples par classe. Cette méthode tire un sous-ensemble aléatoire de taille k de toutes les étiquettes et forme un classificateur multi-labels basé sur LP pour chacun des ensembles d'étiquettes. Étant donné une nouvelle instance, les décisions de tous les classificateurs LP sont combinées à l'aide d'un simple processus de vote pour déterminer l'ensemble final d'étiquettes. RAKEL a un certain nombre de paramètres qui doivent être optimisés pour obtenir des performances quasi optimales.

Cela peut être difficile lorsque le nombre d'exemples de formation est insuffisant.

### 2.3.3- Ensemble de classificateurs multi-labels:

Pour améliorer les performances des classificateurs multi-labels et résoudre le problème de déséquilibre (très peu d'instances pour certains labels), les auteurs ont proposé Ensemble of Multi-Label classifiers (EML). EML utilise des ensembles hétérogènes d'apprenants multi-étiquettes, qui consistent en un ensemble de classificateurs formés individuellement :$h_1, \ldots, h_q$, chaque classificateur multi-étiquette hk appartient à un groupe d'adaptation différent. Pour une instance de test x, chaque classificateur individuel hk produit un vecteur de dimension Q $p_k = [p_{1k}, \ldots, p_{Qk}]$. Chaque pik représente la probabilité que l'appartenance de l'instance x à la classe !i ait été correctement affectée par le classificateur k. Les sorties de ces classificateurs q sont agrégées en utilisant différentes techniques de

combinaison basées sur des méthodes de moyenne et de vote pondéré. Les résultats montrent que ces approches apportent des améliorations significatives en s'attaquant aux problèmes de déséquilibre de classe et de corrélation d'étiquettes.

# 3- Métriques d'évaluation:

## 3.1- Mesures basées sur les prédictions: [hal archive 2]

**Hamming loss:** La métrique de Hamming loss pour l'ensemble d'étiquettes est définie comme la fraction d'étiquettes dont la pertinence est mal prédite :

| | |
|---|---|
| $\mathcal{H}Loss(\mathcal{H},\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \Delta \hat{Y}_i\rvert}{Q},$ | **(1)** |

**Où Δ désigne la différence symétrique entre deux ensembles.**

**Accuracy:** La métrique de précision donne un degré moyen de similitude entre les ensembles d'étiquettes de vérité prédite et terrain :

| | |
|---|---|
| $Accuracy(\mathcal{H},\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap \hat{Y}_i\rvert}{\lvert Y_i \cup \hat{Y}_i\rvert}.$ | **(2)** |

**Précision :** la métrique de précision calcule la proportion de prédictions réellement positives :

| | |
|---|---|
| $\mathcal{P}recision(\mathcal{H},\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap \hat{Y}_i\rvert}{\lvert \hat{Y}_i\rvert}.$ | **(3)** |

**Recall :** cette métrique estime la proportion de vrais libellés qui ont été prédits comme positifs :

| | |
|---|---|
| $\mathcal{R}ecall(\mathcal{H},\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m}\frac{\lvert Y_i \cap \hat{Y}_i\rvert}{\lvert Y_i\rvert}.$ | **(4)** |

**Mesure F1 :** la mesure F1 est définie comme la moyenne harmonique de la précision et du rappel. Il est calculé comme :

| | |
|---|---|
| $\mathcal{F}1(\mathcal{H},\mathcal{S}) = \frac{1}{m}\sum_{i=1}^{m}\frac{2\lvert Y_i \cap \hat{Y}_i\rvert}{\lvert Y_i\rvert + \lvert \hat{Y}_i\rvert}.$ | **(5)** |

### 4-Domaines d’application :

L'apprentissage multi-label a des applications importantes dans de nombreux problèmes du monde réel : , l'annotation vidéo et la bio-informatique

- Classification des images
- Prédiction de la fonction des gènes (la bio-informatique)
- Classification des textes
- Classification vidéo
- la classification de scènes

# Conclusion

Dans ce chapitre, j'ai présenté une étude sur la classification multi-label. Une analyse des méthodes multi-label existantes a été présentée. Ces méthodes peuvent être divisées en trois catégories selon la façon dont elles utilisent les données multi-étiquettes, puis j'ai décrit comment évaluer les classificateurs multi-étiquettes. Enfin, j'ai mentionné plusieurs applications réelles de l'apprentissage multi-label.

# CHAPITRE 3
# LA BIO-INFORMATIQUE ET LES NOTIONS DE BIOLOGIE

La vie sur Terre est apparue il y a un peu plus de trois milliards d'années Tout un organisme vivant, est constitué de cellules très diverses. Ces cellules diffèrent par leur fonction, mais aussi par leur forme, leur taille et bien sûr leur masse. Les calculs réalisés estiment que le nombre total de cellules dans le corps humain s'élève à quelque 30.000 milliards.

Dans la cellule il y a des molécules de petite taille (micromolécule) et aussi d'autres qui ont une grande taille (les macromolécules).

Nous retrouvons trois types de macromolécules sont impliqués dans cette unité cellulaire : les protéines, les ADNs, les ARNs. Ces macromolécules sont les molécules les plus importantes dans l'organisme des vivants.

Dans ce chapitre nous présenterons la bio-informatique en deux parties principales. Dans ces parties nous nous focaliserons sur quelques notions liées à biologie. Ensuite, nous détaillerons.

## 1- Bio-informatique

La bio-informatique est un domaine de la science informatique qui a à voir avec l'analyse de séquences de molécules biologiques. Il fait généralement référence aux gènes, à l'ADN, à l'ARN ou aux protéines, et est particulièrement utile pour comparer des gènes et d'autres séquences dans des protéines et d'autres séquences au sein d'un organisme ou entre organismes, en examinant les relations évolutives entre les organismes et en utilisant les modèles qui existent à travers des séquences d'ADN et de protéines pour comprendre quelle est leur fonction. Vous pouvez considérer la bio-informatique comme essentiellement la partie linguistique de la génétique. C'est-à-dire que les spécialistes de la linguistique recherchent des modèles dans le langage, et c'est ce que font les spécialistes de la bio-informatique : rechercher des modèles dans des séquences d'ADN ou de protéines. [WEB 25]

### 1.1- Définition:

La bio-informatique est une sous-discipline de la biologie et de l'informatique concernée par l'acquisition, le stockage, l'analyse et la diffusion de données biologiques, le plus souvent des séquences d'ADN et d'acides aminés. La bio-informatique utilise des programmes informatiques pour diverses applications, notamment la détermination des fonctions des gènes et des protéines, l'établissement de relations évolutives et la prédiction des formes tridimensionnelles des protéines. [WEB 25]

### 1.2- Objectifs de la bio-informatique

Les principaux objectifs de la bio-informatique sont :

- De gérer les données de manière à permettre un accès facile aux informations existantes et de soumettre de nouvelles entrées au fur et à mesure qu'elles sont produites
- Développer des outils technologiques qui aident à analyser les données biologiques;
- D'utiliser ces outils pour analyser les données et interpréter les résultats d'un point de vue biologique. [WEB 26]

### 1.3- Les données de la bio-informatique

Les données classiques de la bio-informatique comprennent :

- Séquences d'ADN de gènes ou de génomes complets
- Séquence d'acides aminés de protéineset les structures tridimensionnelles des protéines, des acides nucléiques et des complexes protéine-acide nucléique.

Les flux de données « -omiques » supplémentaires comprennent :

- La transcriptomique, le modèle de synthèse d'ARN à partir d'ADN
- La protéomique, la distribution des protéines dans les cellules
- L'interactomique, les modèles d'interactions protéine-protéine et protéine-acide nucléique
- La métabolomique, la nature et les schémas de circulation des transformations de petites molécules par les voies biochimiques actives dans les cellules.

Dans chaque cas, il est intéressant d'obtenir des données complètes et précises pour des types de cellules particuliers et d'identifier des modèles de variation au sein des données.

La bio-informatique a été stimulée par la grande accélération des processus de génération de données en biologie. Les méthodes de séquençage du génome montrent peut-être les effets les plus spectaculaires. En 1999, les archives de séquences d'acides nucléiques

contenaient un total de 3,5 milliards de nucléotides, légèrement plus que la longueur d'un seul génome humain ; une décennie plus tard, ils contenaient plus de 283 milliards de nucléotides, soit la longueur d'environ 95 génomes humains. [WEB 26]

### 1.4- les banques des données en bio-informatique

En a plusieurs banque de donnée dans le domaine de la bio-informatique : [WEB 27]

| Nom | Description |
|---|---|
| **BIOCAT** | Annuaire de logiciels d'intérêt général en biologie moléculaire et génétique. |
| **BLOCKS** | Alignements multiples de segments sans insertions correspondant aux régions les mieux conservées de Prosite. Base automatiquement générée par la recherche des régions les plus conservées dans les groupes de protéines de PROSITE. |
| **PDB** | (Protein Data Bank). PDB : Coordonnées 3D des protéines dont la structure a été déterminée |
| **PFAM** | (Protein FAMily database). Collection de familles alignées de protéines, générées automatiquement ou semi-automatiquement par la méthode "Hidden Markov Models" (HMMs). |
| **PRODOM** | PROtein DOMain Database).Base de sites et patterns biologiquement significatifs. PRODOM est une compilation automatisée des domaines homologues (alignements multiples et consensus) détectés dans SwissProt. ProDom CG : Complete Genomes Protein Domain Database. |
| **PROSITE** | (Database of protein families and domains |
| **IPI** | International Protein Index est une banque protéomique humaine et de souris construite sur des données d'Ensembl, SWISS-PROT/TrEMBL et RefSeq. |

### 1.5- Stockage et récupération des données

En bio-informatique, les banques de données sont utilisées pour stocker et organiser les données. Beaucoup de ces entités collectent des séquences d'ADN et d'ARN à partir d'articles scientifiques et de projets sur le génome. De nombreuses bases de données sont entre les mains de consortiums internationaux. Pour s'assurer que les données de séquences sont librement disponibles, les revues scientifiques exigent que les nouvelles séquences nucléotidiques soient déposées dans une base de données accessible au public comme condition de publication d'un article. (Des conditions similaires s'appliquent aux structures d'acide nucléique et de protéine.)

La principale base de données de la structure macromoléculaire biologique est la banque mondiale de données sur les protéines (wwPDB), un effort conjoint du Research Collaboratory for Structural Bioinformatics (RCSB) aux États-Unis, de la Protein Data Bank Europe (PDBe) de l'Institut européen de bio-informatique dans le Royaume-Uni et la Protein Data Bank Japan de l'Université d'Ōsaka.

La récupération d'informations à partir des archives de données utilise des outils standards pour l'identification des éléments de données par mot-clé ; par exemple, on peut taper "myoglobine aardvark" dans Google et récupérer la séquence d'acides aminés de la molécule. D'autres algorithmes recherchent des banques de données pour détecter des similitudes entre des éléments de données. [WEB 26]

## 2- Notions de base biologique

Avant d'entamer la partie bio-informatique on va d'abord citer et définir quelques notions qui concernent la biologie y compris ADN, ARN…etc.

### 2.1- Chromosomes

Un chromosome est une longue molécule d'ADN contenant tout ou partie du matériel génétique d'un organisme. La plupart des chromosomes eucaryotes comprennent des protéines d'emballage appelées histones qui, aidées par des protéines chaperons, se lient à la molécule d'ADN et la condensent pour maintenir son intégrité. Ces chromosomes présentent une structure tridimensionnelle complexe, qui joue un rôle important dans la régulation transcriptionnelle. [WEB 28]

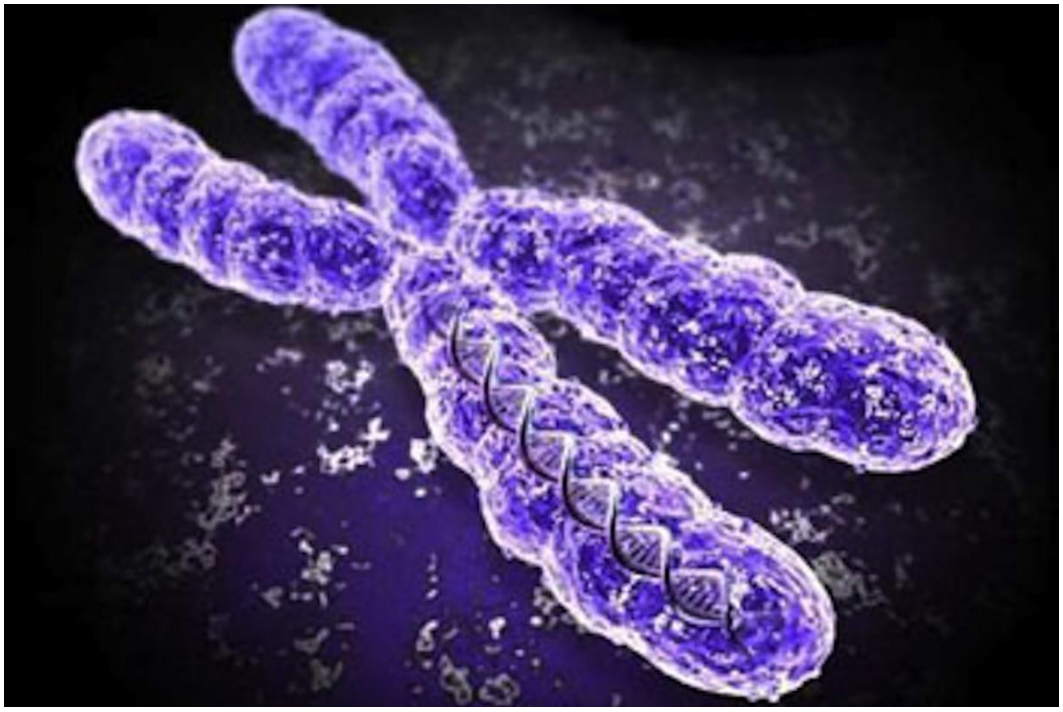


**Figure 19 :** chromosome [WEB 29]

## 2.2- ADN

### 2.2.1- Définition

Longue molécule que l'on retrouve dans tous les organismes. La structure originale de l'ADN, formée de deux brins complémentaires enroulés en hélice (double hélice), lui permet de se dupliquer en deux molécules identiques entre elles et identiques à la molécule mère lors du phénomène de réplication ou duplication. On dit que l'ADN est le support de l'hérédité. Il est présent dans le noyau des cellules eucaryotes, les cellules procaryotes, dans les mitochondries ainsi que dans les chloroplastes. Il est à la base de processus biologiques aboutissant à la production des protéines. [WEB 30]

### 2.2.2- Les composants de l'ADN

L'ADN est une grande molécule formée d'un grand nombre de nucléotides. [WEB 31]
Le nucléotide de l'ADN est constitué de trois éléments principaux :

1. Un groupe phosphate

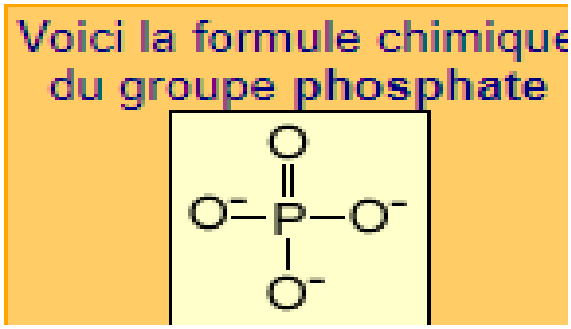



**Figure 20** : formule chimique de groupe phosphate [WEB 31]

2. Un sucre à 5 carbones (pentose) : le désoxyribose


**Figure 21 :** le désoxyribose [WEB 31]

3. Une base azotée qui peut être soit :
   - La cytosine (C)
   - La thymine (T)
   - L'adénine (A)
   - La guanine (G)

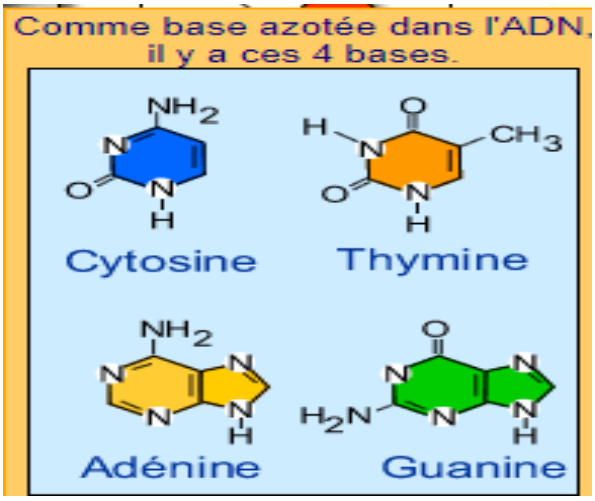



**Figure 22 :** les bases azotées [WEB 31]

Il existe donc 4 types de nucléotides qui se lient l'un à la suite de l'autre et forme un polynucléotide.

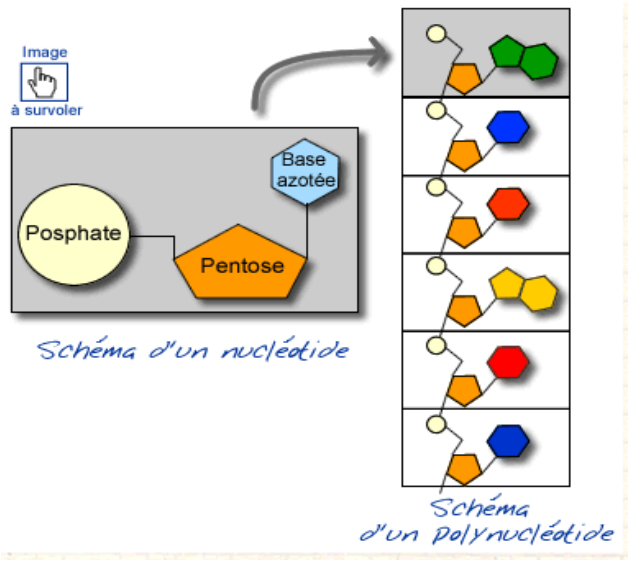



**Figure 23 :** Les composants de l'ADN [WEB 31]

**Figure 24 :** Brin d'ADN avec ses bases azotées [WEB 32]

## 2.3- ARN

### 2.3.1- Définition [WEB 33]

L'acide ribonucléique (ARN) est une molécule polymère essentielle dans divers rôles biologiques dans le codage, le décodage, la régulation et l'expression des gènes. L'ARN et l'acide désoxyribonucléique (ADN) sont des acides nucléiques. Avec les lipides, les protéines et les glucides, les acides nucléiques constituent l'une des quatre macromolécules majeures essentielles à toutes les formes de vie connues. Comme l'ADN, l'ARN est assemblé sous la forme d'une chaîne de nucléotides, mais contrairement à l'ADN, l'ARN se trouve dans la nature sous la forme d'un simple brin replié sur lui-même, plutôt que d'un double brin apparié.

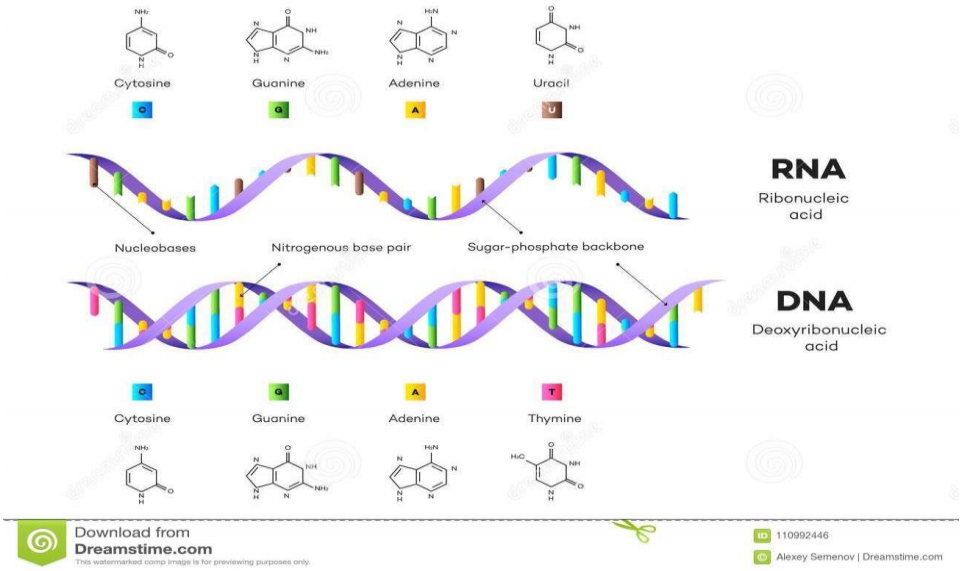



**Figure 25 :** L'acide ribonucléique (ARN) [WEB 34]

### 2.3.2- Les types d'ARN

Chez les procaryotes et les eucaryotes, il existe trois principaux types d'ARN : l'ARN messager (ARNm), l'ARN ribosomique (ARNr) et l'ARN de transfert (ARNt). Ces 3 types d'ARN sont discutés ci-dessous. [WEB 35]

**ARN messager (ARNm)**

L'ARNm ne représente que 5% de l'ARN total dans la cellule. L'ARNm est le plus hétérogène des 3 types d'ARN en termes de séquence de bases et de taille. Il porte le code génétique complémentaire copié, à partir de l'ADN lors de la transcription, sous forme de triplets de nucléotides appelés codons.

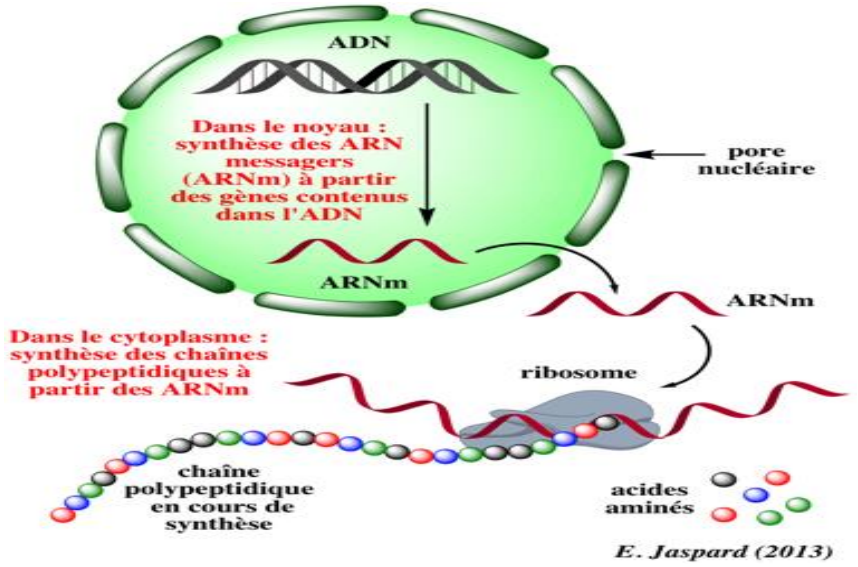
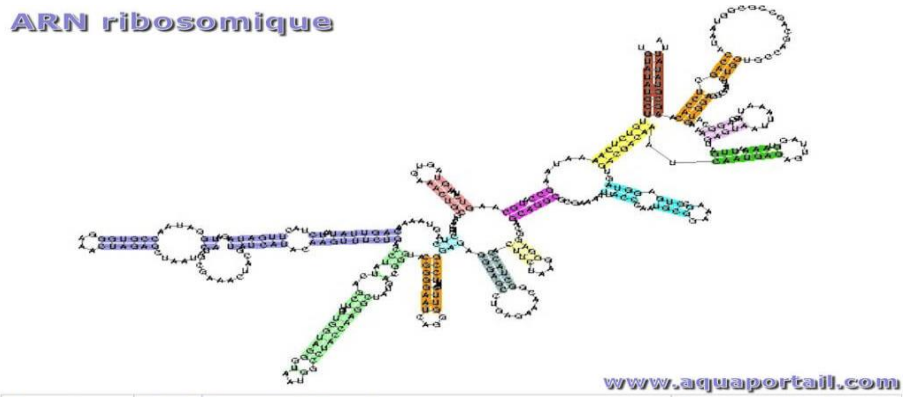



**Figure 26 :** ARNm [WEB 36]

**ARN ribosomique (ARNr)**

Les ARNr se trouvent dans les ribosomes et représentent 80 % de l'ARN total présent dans la cellule. Les ribosomes sont composés d'une grande sous-unité appelée 50S et d'une petite sous-unité appelée 30S, chacune étant composée de ses propres molécules d'ARNr spécifiques. Les différents ARNr présents dans les ribosomes comprennent les petits ARNr et les grands ARNr, qui appartiennent respectivement aux petites et grandes sous-unités du ribosome.




| Domaine | Taille | Sous-unité majeure | Sous-unité mineure |
|---|---|---|---|
| Procaryotes | 70 S | 50 S (5 S: 120 nt, 23 S: 2906 nt) | 30S (16S: 1542 nt) |
| Eucaryotes | 80 S | 60 S (5 S: 121 nt, 5,8 S: 156 nt, 28 S: 5070 nt) | 40S (18S: 1869 nt) |

**Figure 27 :** ARNr [WEB 37]

**ARN de transfert (ARNt)**

L'ARNt est le plus petit des 3 types d'ARN, possédant environ 75-95 nucléotides. Les ARNt sont un composant essentiel de la traduction, où leur fonction principale est le transfert d'acides aminés lors de la synthèse des protéines. Par conséquent, ils sont appelés ARN de transfert.

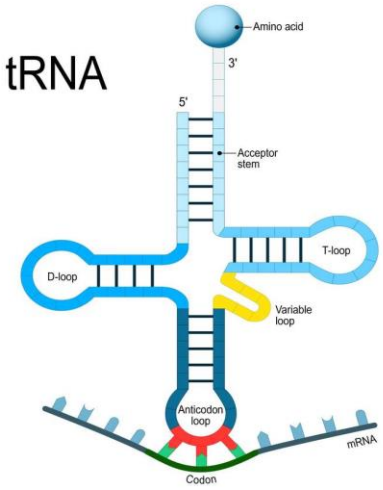



**Figure 28:** ARNt [WEB 38]

## 2.4- Le protéine

### 2.4.1- Définition

Les protéines ont été définies comme étant des macromolécules biologiques présentes dans toutes les cellules vivantes, mais des travaux récents montrent qu'il existe aussi des centaines voire des milliers de micro- ou nano-protéines. Elles sont formées d'une ou de plusieurs chaînes polypeptidiques. Chacune de ces chaînes est constituée de l'enchaînement de résidus d'acides aminés liés entre eux par des liaisons peptidiques. [WEB 39]

### 2.4.2- Fabrication des protéines

Les protéines sont codées par les gènes et synthétisées par le ribosome au cours du processus de traduction de l'ARN. Elles sont ainsi créées par l'incorporation successive d'acides aminés, maintenus entre eux grâce à la formation de liaisons peptidiques, selon l'ordre indiqué par la succession des codons sur l'ARN. [WEB 40]

### 2.4.3- Structure des protéines

Suivant l'ordre des acides aminés et leurs interactions entre eux, les protéines prennent des conformations particulières, qui sont essentielles pour leur fonctionnalité. Il existe des structures rencontrées très fréquemment : [WEB 41]

**a) La structure primaire :** ou séquence, d'une protéine correspond à la succession linéaire des acides aminés (ou résidus) la constituant sans référence à une configuration spatiale. Les protéines sont donc des polymères d'acides aminés, reliés entre eux par des liaisons peptidiques. La structure primaire d'une protéine est le fruit de la traduction de l'ARNm en séquence protéique par le ribosome.

**b) La structure secondaire**

La chaîne principale contient trois liaisons covalentes par acide aminé. La liaison peptidique étant une liaison plane, il reste deux liaisons simples autour desquelles la rotation est possible. On peut donc déterminer la conformation du squelette d'un acide aminé à partir de deux angles dièdres,

- L'angle dièdre φ est défini par les quatre atomes successifs du squelette : CO-NH-Cα-CO, le premier carbonyle étant celui du résidu précédent.
- L'angle dièdre ψ est défini par les quatre atomes successifs du squelette : NH-Cα-CO-NH, le second amide étant celui du résidu suivant.

Donc il y a deux structures secondaires principales :

• les hélices φ

• les feuillets ψ.

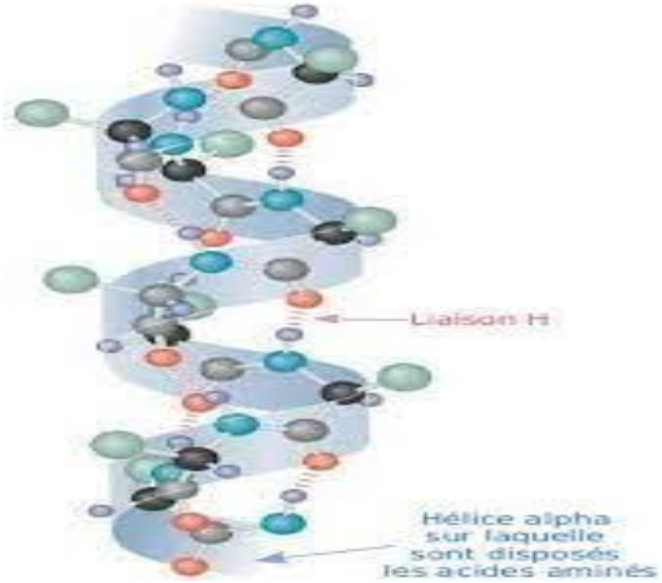



**Figure 29:** les hélices φ [WEB 42]

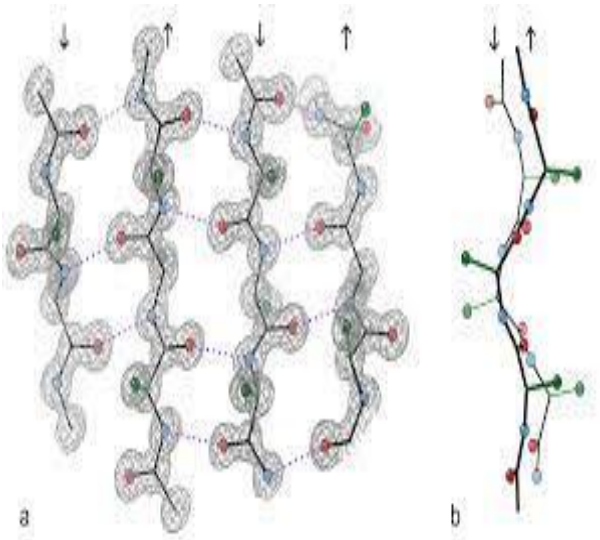



**Figure 30:** les feuillets ψ [WEB 43]

**c) La structure tertiaire**

La structure tertiaire d'une protéine correspond au repliement de la chaîne polypeptidique dans l'espace. On parle plus couramment de structure tridimensionnelle. La structure tridimensionnelle d'une protéine est intimement liée à sa fonction: lorsque cette structure est cassée par l'emploi d'agents dénaturants ou chaotropiques, la protéine perd sa fonction: elle est dénaturée.

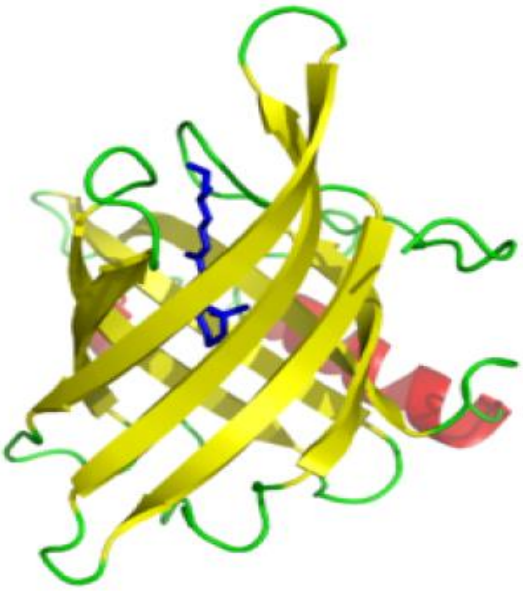


**Figure 31:** La structure tertiaire [WEB 44]

## 3- Les différents domaines de protéine

### 3.1-Définition de domaine

Un domaine protéique est une partie d'une protéine capable d'adopter une structure de manière autonome ou partiellement autonome du reste de la molécule. C'est un élément modulaire de la structure des protéines qui peuvent ainsi être composées de l'assemblage de plusieurs de ces domaines. On parle alors de protéine multi domaines.

Les domaines protéiques forment en général une structure compacte et stable, et peuvent parfois être produits de manière indépendante (par génie génétique, coupure protéolytique...). [WEB 45]

### 3.2- Les différents domaines de protéine

Exemples de domaines :

#### 3.2.1-Domaine de fermeture à glissière leucine basique (domaine bZIP) :

Présent dans de nombreuses protéines eucaryotes se liant à l'ADN. Une partie du domaine contient une région qui médie les propriétés de liaison à l'ADN spécifiques à la séquence et la fermeture à glissière Leucine qui est requise pour la dimérisation de deux régions de liaison à l'ADN.

#### 3.2.2-Domaine effecteur de la mort (DED) :

Permet la liaison protéine-protéine par des interactions homotypiques (DED-DED). Les caspases protéases déclenchent l'apoptose via des cascades protéolytiques. La pro-caspase-8 et la pro-caspase-9 se lient à des molécules adaptatrices spécifiques via des domaines DED, ce qui conduit à l'auto activation des caspases.

#### 3.2.3-Domaine de liaison à la phosphotyrosine (PTB) :

Les domaines PTB se lient généralement aux résidus de tyrosine phosphorylés. On les trouve souvent dans les protéines de transduction du signal.

#### 3.2.4-Domaine d'homologie 2 Src (SH2) :

Les domaines SH2 sont souvent trouvés dans les protéines de transduction du signal. Les domaines SH2 confèrent la liaison à la tyrosine phosphorylée (pTyr). Nommé d'après le domaine de liaison à la phosphotyrosine de l'oncogène viral src, qui est lui-même une tyrosine kinase.

#### 3.2.5-Domaine de liaison à l'ADN à doigt de zinc (ZnF_GATA) :

Les protéines contenant le domaine ZnF_GATA sont généralement des facteurs de transcription qui se lient généralement à la séquence d'ADN [AT] GATA [AG] des promoteurs.

## 4- Les positions des protéines dans les cellules humaines

Depuis les positions on nomme quelque position existante dans notre travail :

**4.1-Nucleus :** Le noyau contient le matériel génétique (ADN), sous la forme d'un complexe ADN-protéines appelé chromatine et composé de plusieurs unités discontinues appelées chromosomes. La chromatine apparaît sous forme diffuse3 pendant les phases séparant les divisions cellulaires (interphase). Elle se condense au moment de la division cellulaire. [WEB 46]

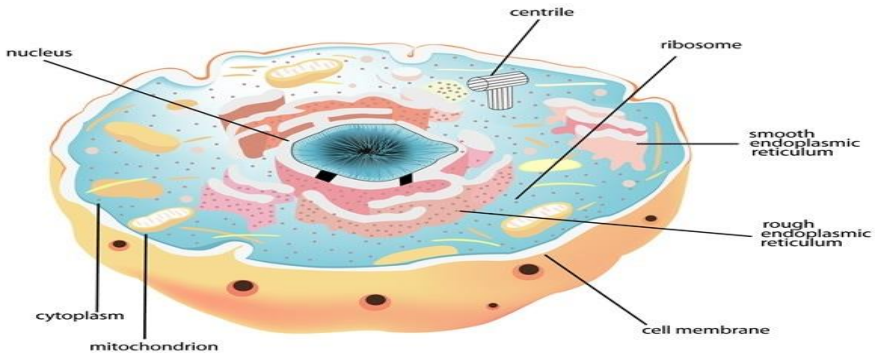



**Figure 32:** Cellule et noyau [WEB 47]

**4.2-Mevalonate pathway :**

La voie du mévalonate (MP) est une voie anabolique fournissant des métabolites pour de multiples processus cellulaires chez les eucaryotes, soulignant ainsi son importance pour presque tous les organismes vivants, y compris les humains.

Le mévalonate qui est produit à partir d'acétoacétyl-CoA par HMGCR est ensuite transformé en stérol isoprénoïdes, tels que le cholestérol, qui est un précurseur indispensable des acides biliaires, des lipoprotéines et des hormones stéroïdes, et en un certain nombre de molécules hydrophobes, notamment des isoprénoïdes non stéroïdiens, tels que le dolichol, l'hème-A, l'ARNt d'isopentényl et l'ubiquinone. [WEB 48]

**4.3-Tissu musculaire :**

Les protéines sont le matériau de base de la structure des tissus. Ils sont le composant le plus important du muscle squelettique strié. Leur classification est corrélée à la structure histologique du tissu musculaire. Les protéines musculaires peuvent être divisées en formes contractile, régulatrice, sarcoplasmique et extracellulaire. Les plus importantes sont les protéines contractiles actine et myosine. Parmi les protéines régulatrices, la troponine, la tropomyosine, la protéine M, la bêta-actine, la gamma-actine et la protéine C ont une grande importance. [WEB 49]

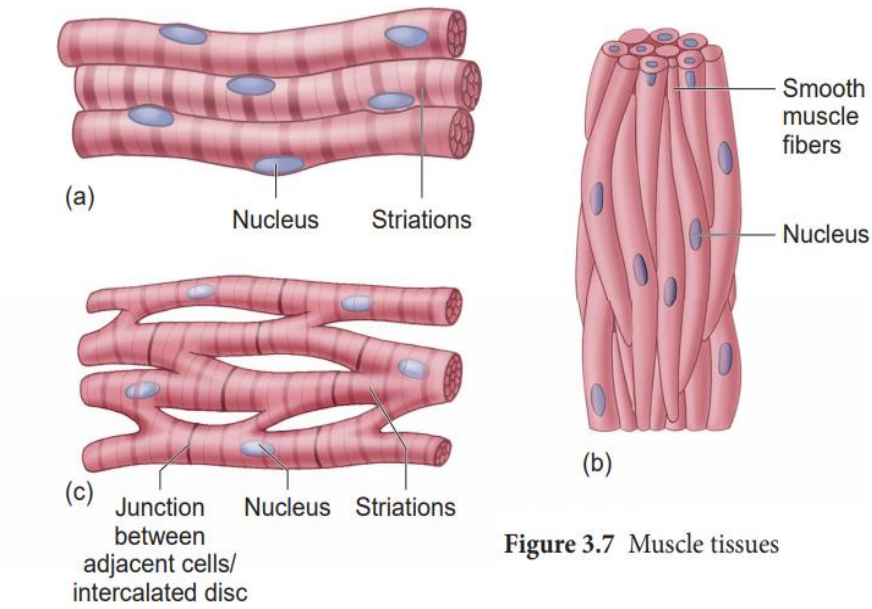



**Figure 33 :** Tissu musculaire [WEB 50]

## 4.4-Cytoplasm :

Le cytoplasme est une solution épaisse qui remplit chaque cellule et est entourée par la membrane cellulaire. Il est principalement composé d'eau, de sels et de protéines telles que les protéines tyrosine kinases—protéines cytoplasmiques tyrosine kinases. Bien que le cytoplasme puisse sembler n'avoir aucune forme ou structure, il est en réalité très organisé. Un cadre d'échafaudages protéiques appelé cytosquelette fournit au cytoplasme et à la cellule leur structure. [WEB 51]

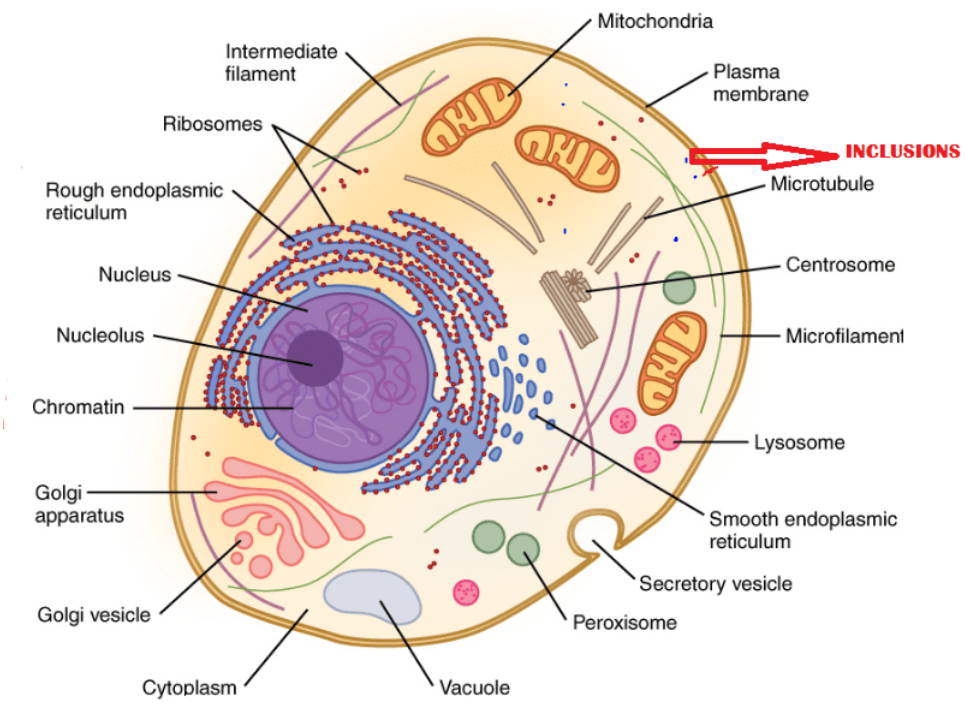



**Figure 34 :** Cytoplasm [WEB 52]

### 4.5-Membrane :

Le deuxième composant principal des membranes plasmiques est la variété des protéines. Une protéine membranaire est une molécule de protéine qui est attachée ou associée à la membrane d'une cellule ou d'un organite. Les protéines membranaires peuvent être classées en deux groupes en fonction de la manière dont la protéine est associée à la membrane : (1) les protéines membranaires intégrales et (2) les protéines membranaires périphériques. [WEB 53]

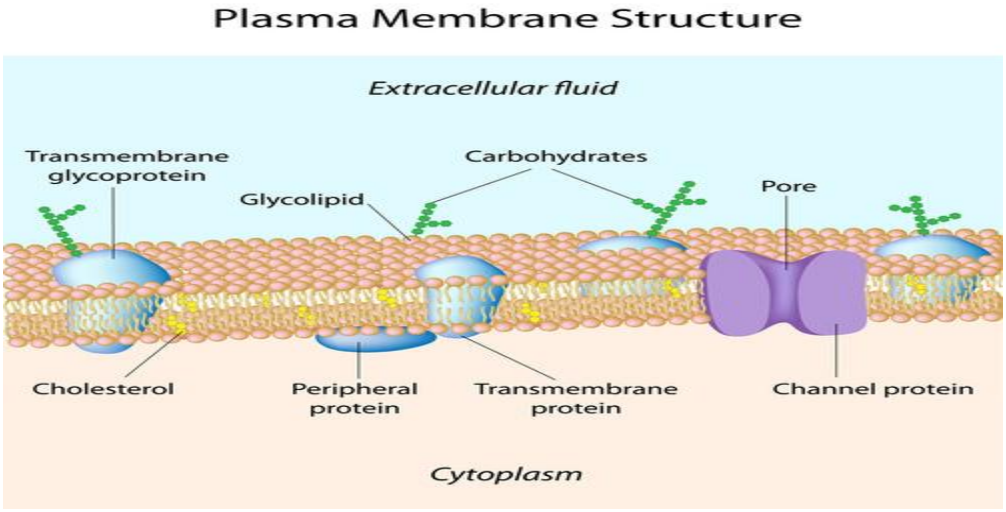



**Figure 35 :** Membrane [WEB 53]

# Conclusion

Dans ce chapitre, on a commencé par présenter les notions de base de la biologie où on a parlé de l'ADN et l'ARN détaillons leur définition et les types d'ARN . Ensuit on a parlé sur les protéines leur structure, leur domaine et on a mentionné quelque position des protéines dans le Corp humain .Enfin on a défini la bio-informatique, les donnes en bio-informatique et les objectifs principales de cette dernière.

# CHAPITRE 4
# CONCEPTION ET IMPLEMENTATION

## 1. Introduction

Dans cette partie du mémoire nous présentons la phase d'implémentation. La démonstration de programme, de l'importation des bibliothèques nécessaires jusqu'à l'obtention des résultats à analyser et l'évaluation de classification. Nous nous attacherons dans ce chapitre à détailler les choix d'implémentation et les résultats de classification**.**

## 2. Conception

### 2.1- Architecture fonctionnelle de l'application :

Nous allons appliquer une approche pour prédire les positions des protéines dans le corp humain. La figure suivant démontre les étapes de notre travail :

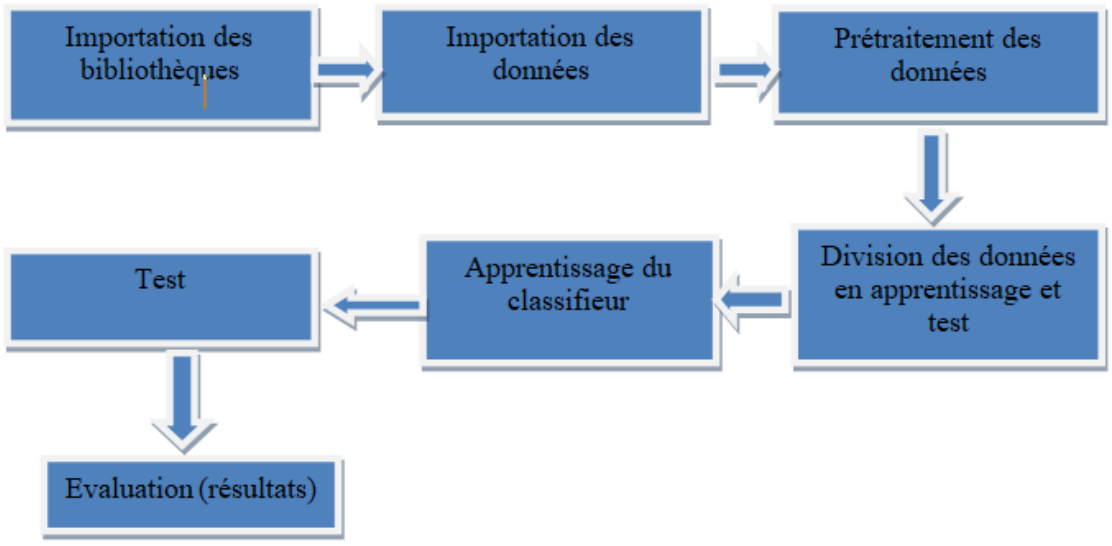
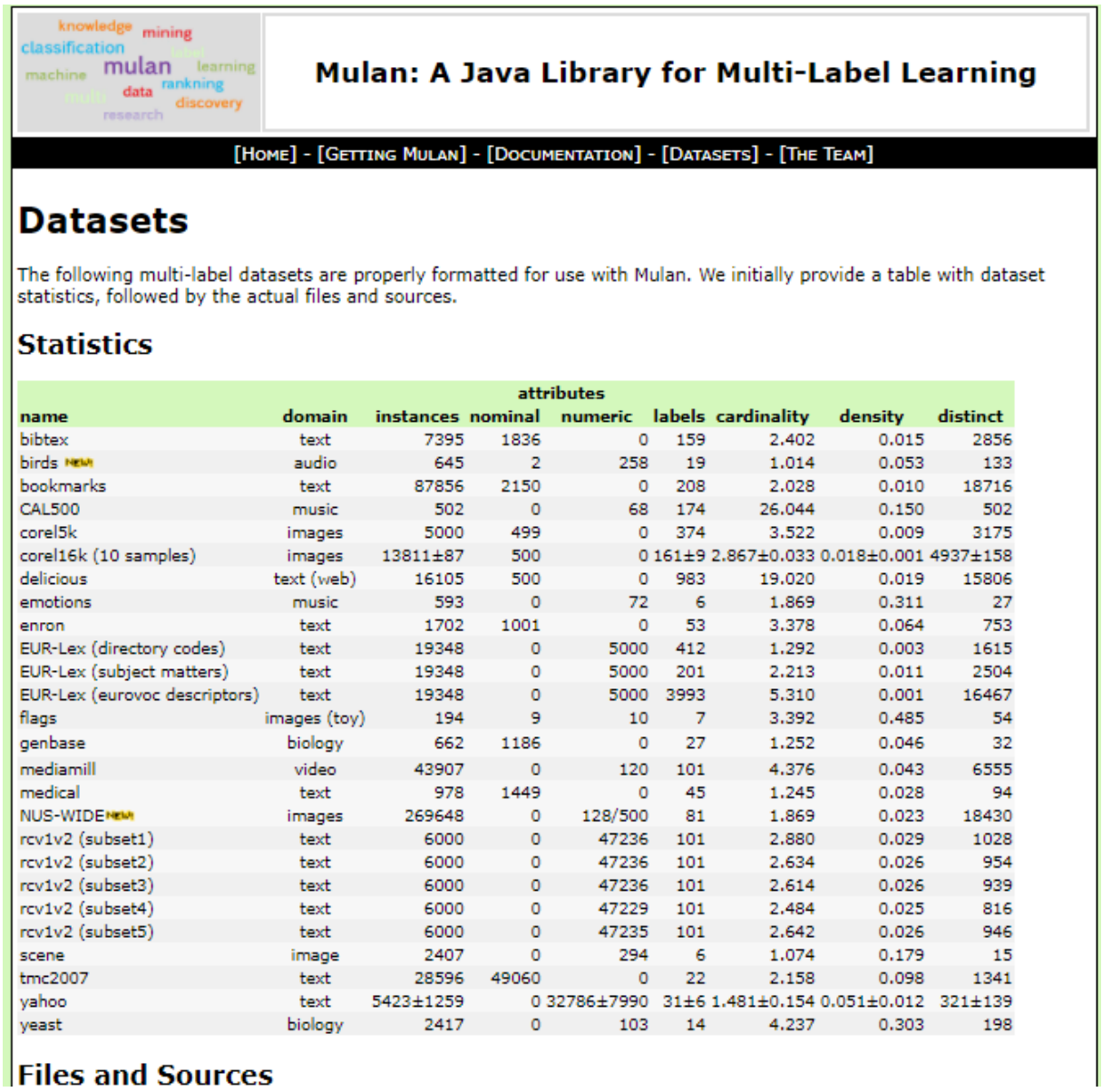



**Figure 36 :** Étapes de l'implémentation

## 2.2- Source de la base :

On a téléchargé notre base depuis le site suivant :

http://mulan.sourceforge.net/datasets-mlc.html



Mulan: A Java Library for Multi-Label Learning

[HOME] - [GETTING MULAN] - [DOCUMENTATION] - [DATASETS] - [THE TEAM]

# Datasets

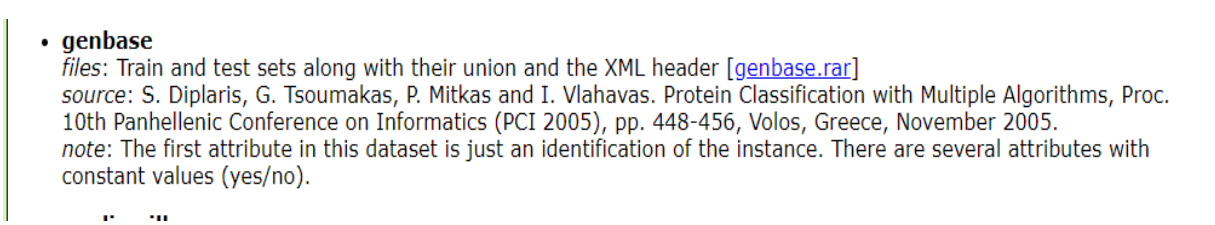
The following multi-label datasets are properly formatted for use with Mulan. We initially provide a table with dataset statistics, followed by the actual files and sources.

## Statistics

| | | | attributes | | | | | |
|---|---|---|---|---|---|---|---|---|
| name | domain | instances | nominal | numeric | labels | cardinality | density | distinct |
| bibtex | text | 7395 | 1836 | 0 | 159 | 2.402 | 0.015 | 2856 |
| birds NEW | audio | 645 | 2 | 258 | 19 | 1.014 | 0.053 | 133 |
| bookmarks | text | 87856 | 2150 | 0 | 208 | 2.028 | 0.010 | 18716 |
| CAL500 | music | 502 | 0 | 68 | 174 | 26.044 | 0.150 | 502 |
| corel5k | images | 5000 | 499 | 0 | 374 | 3.522 | 0.009 | 3175 |
| corel16k (10 samples) | images | 13811±87 | 500 | 0 | 161±9 | 2.867±0.033 | 0.018±0.001 | 4937±158 |
| delicious | text (web) | 16105 | 500 | 0 | 983 | 19.020 | 0.019 | 15806 |
| emotions | music | 593 | 0 | 72 | 6 | 1.869 | 0.311 | 27 |
| enron | text | 1702 | 1001 | 0 | 53 | 3.378 | 0.064 | 753 |
| EUR-Lex (directory codes) | text | 19348 | 0 | 5000 | 412 | 1.292 | 0.003 | 1615 |
| EUR-Lex (subject matters) | text | 19348 | 0 | 5000 | 201 | 2.213 | 0.011 | 2504 |
| EUR-Lex (eurovoc descriptors) | text | 19348 | 0 | 5000 | 3993 | 5.310 | 0.001 | 16467 |
| flags | images (toy) | 194 | 9 | 10 | 7 | 3.392 | 0.485 | 54 |
| genbase | biology | 662 | 1186 | 0 | 27 | 1.252 | 0.046 | 32 |
| mediamill | video | 43907 | 0 | 120 | 101 | 4.376 | 0.043 | 6555 |
| medical | text | 978 | 1449 | 0 | 45 | 1.245 | 0.028 | 94 |
| NUS-WIDE NEW | images | 269648 | 0 | 128/500 | 81 | 1.869 | 0.023 | 18430 |
| rcv1v2 (subset1) | text | 6000 | 0 | 47236 | 101 | 2.880 | 0.029 | 1028 |
| rcv1v2 (subset2) | text | 6000 | 0 | 47236 | 101 | 2.634 | 0.026 | 954 |
| rcv1v2 (subset3) | text | 6000 | 0 | 47236 | 101 | 2.614 | 0.026 | 939 |
| rcv1v2 (subset4) | text | 6000 | 0 | 47229 | 101 | 2.484 | 0.025 | 816 |
| rcv1v2 (subset5) | text | 6000 | 0 | 47235 | 101 | 2.642 | 0.026 | 946 |
| scene | image | 2407 | 0 | 294 | 6 | 1.074 | 0.179 | 15 |
| tmc2007 | text | 28596 | 49060 | 0 | 22 | 2.158 | 0.098 | 1341 |
| yahoo | text | 5423±1259 | 0 | 32786±7990 | 31±6 | 1.481±0.154 | 0.051±0.012 | 321±139 |
| yeast | biology | 2417 | 0 | 103 | 14 | 4.237 | 0.303 | 198 |

## Files and Sources

**Figure 37 :** Source de donnée



- **genbase**
  *files*: Train and test sets along with their union and the XML header [genbase.rar]
  *source*: S. Diplaris, G. Tsoumakas, P. Mitkas and I. Vlahavas. Protein Classification with Multiple Algorithms, Proc. 10th Panhellenic Conference on Informatics (PCI 2005), pp. 448-456, Volos, Greece, November 2005.
  *note*: The first attribute in this dataset is just an identification of the instance. There are several attributes with constant values (yes/no).

**Figure 38 :** Téléchargé Genbase

# 3. Implémentation

## 3.1- Les outils utilisés

### 3.1.1- Python

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages. [WEB 54]


**Figure 39 :** Logo de langage python [WEB 55]

### 3.1.2-Anaconda Navigator

Anaconda Navigator est une interface utilisateur graphique (GUI) de bureau incluse dans la distribution Anaconda® qui vous permet de lancer des applications et de gérer facilement les packages, les environnements et les canaux conda sans utiliser de commandes de ligne de commande. Navigator peut rechercher des packages sur Anaconda.org ou dans un référentiel Anaconda local. Il est disponible pour Windows, macOS et Linux. [WEB 56]

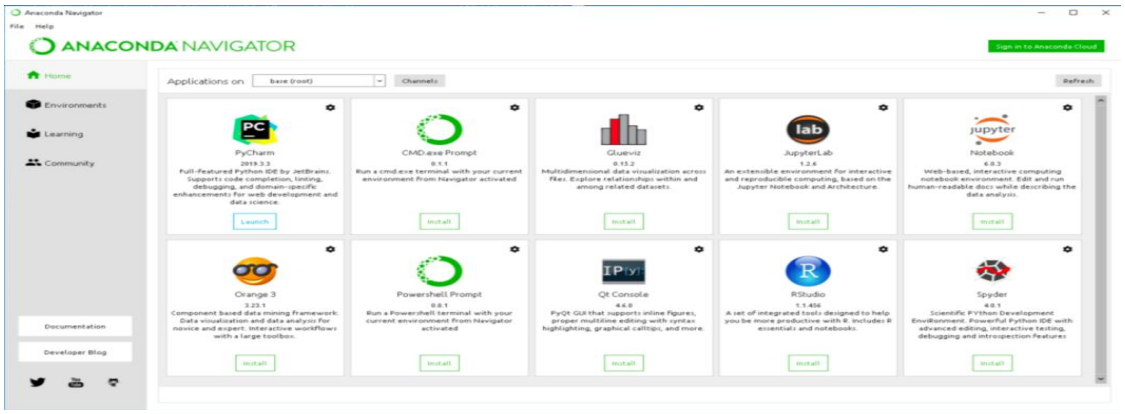



**Figure 40 :** Plateforme d'Anaconda3

### 3.1.3-Jupyter Notebook

L'application Jupyter Notebook est une application serveur-client qui permet de modifier et d'exécuter des documents de notebook via un navigateur Web. L'application Jupyter Notebook peut être exécutée sur un bureau local ne nécessitant aucun accès Internet (comme décrit dans ce document) ou peut être installée sur un serveur distant et accessible via Internet.

En plus d'afficher/éditer/exécuter des documents de bloc-notes, l'application Jupyter Notebook dispose d'un « tableau de bord » (Notebook Dashboard), un « panneau de configuration » affichant les fichiers locaux et permettant d'ouvrir des documents de bloc-notes ou d'arrêter leurs noyaux. [WEB 57]

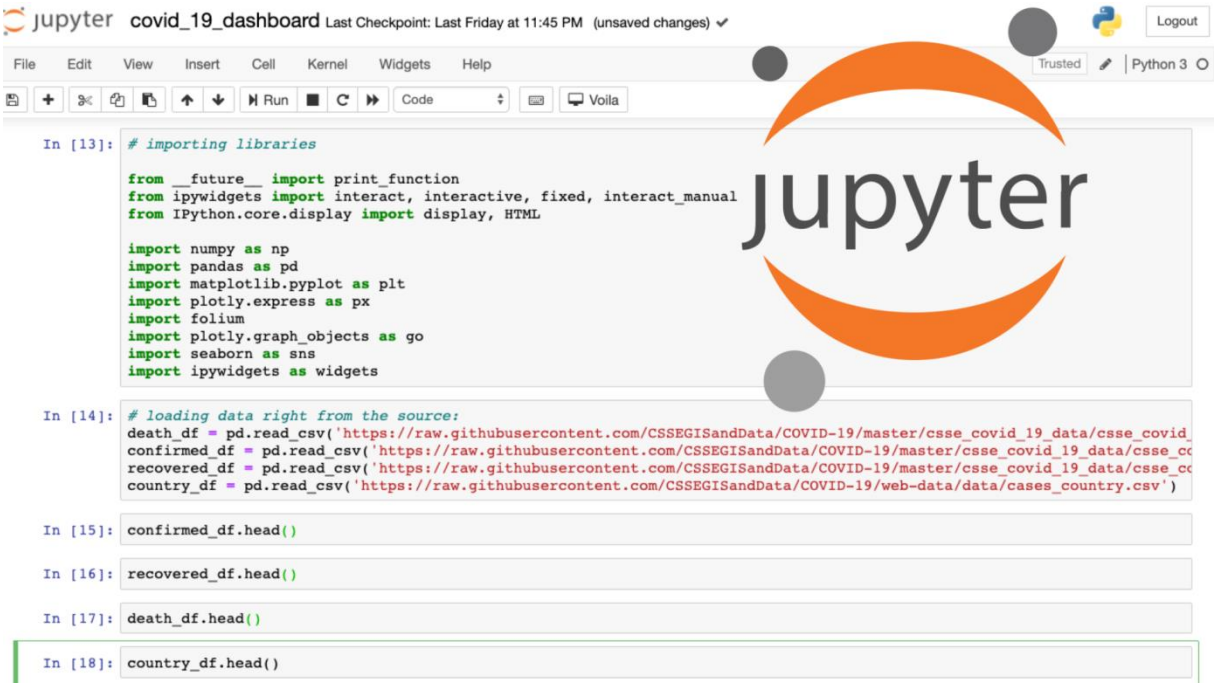



**Figure 41** : Plateforme de jupyter notebook [WEB 58]

## 3.2- Les étapes de l'implémentation

Les étapes de l'implémentation sont :

1. L'importation des bibliothèques.
2. L'importation du dataset.
3. Prétraitement des données.
4. Phase d'apprentissage de classifieur.
5. Phase de test.
6. L'évaluation.

### 3.2.1- Importation des bibliothèques (package)

La première partie du programme est l'importation des bibliothèques nécessaires pour mener à bien les différentes tâches. Pour ce faire nous avons eu recours à des bibliothèques principales ci-dessous.

```
from numpy import mean
from numpy import std
from sklearn.datasets import make_multilabel_classification
from skmultilearn.dataset import load_dataset,load_from_arff
from sklearn.model_selection import RepeatedKFold
from keras.models import Sequential
import numpy as np
from sklearn.metrics import hamming_loss, multilabel_confusion_matrix
from sklearn.metrics import accuracy_score
from sklearn.metrics import f1_score
from sklearn.metrics import precision_score
from sklearn.metrics import recall_score
from skmultilearn.problem_transform import BinaryRelevance
from sklearn.linear_model import LogisticRegression
import pandas as pd
import math
import seaborn as sns
```

**Figure 42**: Importation des bibliothèques

- **Numpy :**

NumPy est le package fondamental pour le calcul scientifique en Python. Il s'agit d'une bibliothèque Python qui fournit un objet tableau multidimensionnel, divers objets dérivés (tels que des tableaux et des matrices masqués) et un assortiment de routines pour des opérations rapides sur des tableaux, notamment mathématiques, logiques, manipulation de forme, tri, sélection, E/S , transformées de Fourier discrètes, algèbre linéaire de base, opérations statistiques de base, simulation aléatoire et bien plus encore. [WEB 59]

- **Scikit-learn :**

Scikit-learn (anciennement scikits.learn et également connu sous le nom de sklearn) est une bibliothèque logicielle gratuite d'apprentissage automatique pour le langage de programmation Python. Il comporte divers algorithmes de classification, de régression et de clustering, notamment les machines vectorielles de support, les forêts aléatoires, l'amplification de gradient, k-means et DBSCAN, et est conçu pour interagir avec les bibliothèques numériques et scientifiques Python NumPy et SciPy. [WEB 60]

- **scikit-multilearn :**

scikit-multilearn est un module Python capable d'effectuer des tâches d'apprentissage multi-étiquettes. Il est construit sur divers packages scientifiques Python (numpy, scipy) et suit une API similaire à celle de scikit-learn. [WEB 61]

- **Keras :**

Keras est une bibliothèque open source de composants de réseaux neuronaux écrits en Python. Keras est capable de fonctionner sur TensorFlow, Theano, PlaidML et d'autres. La bibliothèque a été développée pour être modulaire et conviviale, mais elle a initialement commencé dans le cadre d'un projet de recherche pour le système d'exploitation intelligent neuro-électronique ouvert ou ONEIROS. [WEB 62]

- **Pandas :**

Pandas est une boîte à outils d'analyse de données basée sur Python qui peut être importée à l'aide d'import pandas as pd. Il présente une gamme variée d'utilitaires, allant de l'analyse de plusieurs formats de fichiers à la conversion d'une table de données entière en un tableau matriciel NumPy. Cela fait des pandas un allié de confiance dans la science des données et l'apprentissage automatique. [WEB 63]

- **Math :**

La bibliothèque mathématique Python nous donne accès à certaines fonctions et constantes mathématiques courantes en Python, que nous pouvons utiliser tout au long de notre code pour des calculs mathématiques plus complexes. La bibliothèque est un module Python intégré, vous n'avez donc pas besoin de faire d'installation pour l'utiliser. [WEB 64]

- **Seaborn :**

Seaborn est une bibliothèque en Python principalement utilisée pour créer des graphiques statistiques. C'est également une bibliothèque de visualisation de données construite sur matplotlib et étroitement intégrée aux structures de données pandas en Python. La visualisation est la partie centrale de Seaborn qui aide à l'exploration et à la compréhension des données. [WEB 65]

### 3.2.2- L'importation du dataset

On utilise la bibliothèque de scikit-multilearn pour télécharger et importer la base de donnée **Genbase.arff :**

```
from skmultilearn.dataset import load_dataset,load_from_arff
```

On utilise l'instruction suivante pour effectuer le téléchargement et importation de la base de données :

```
Entrée [14]: X_train, y_train, feature_names_train, label_names_train = load_dataset('genbase', 'train')
             X_test, y_test, feature_names_test, label_names_test = load_dataset('genbase', 'test')

             genbase:train - exists, not redownloading
             genbase:test - exists, not redownloading
```

**Figure 43**: l'instruction de téléchargement et importation de la base de données

**Observation :**

- Comme on a déjà exécuté l'instruction avant donc le résultat dit que la base existe déjà.
- On n'a pas besoin de séparer la base en train et test ca l st séparer d'origine.

### 3.2.3- Prétraitement des données

La troisième étape consiste au nettoyage des données. Pour ce faire nous avons écrit une fonctions **cleaning()**. Cet fonctions vont supprimer les classes qui ont un 0 dans tout les protéines, ce qui signifie qu'aucune des protéines de l'ensemble de donnée (train) n'appartient a ces classes.

```
Entrée [33]: def cleaning():
                 global X_train, y_train, X_test, y_test, label_names_train, label_names_test
                 X_train = X_train.toarray()
                 y_train = y_train.toarray()
                 X_test = X_test.toarray()
                 y_test = y_test.toarray()
                 counter_cleaning = 0
                 for y in y_train.T:
                     if np.all(y == 0):
                         X_train = np.delete(X_train, counter_cleaning, 1)
                         y_train = np.delete(y_train, counter_cleaning, 1)
                         X_test = np.delete(X_test, counter_cleaning, 1)
                         y_test = np.delete(y_test, counter_cleaning, 1)
                         label_train = label_names_train.pop(counter_cleaning)
                         label_names_test.pop(counter_cleaning)
                         print("Removing the following label : ", label_train)
                     elif np.all(y == 1):
                         X_train = np.delete(X_train, counter_cleaning, 1)
                         y_train = np.delete(y_train, counter_cleaning, 1)
                         X_test = np.delete(X_test, counter_cleaning, 1)
                         y_test = np.delete(y_test, counter_cleaning, 1)
                         label_train = label_names_train.pop(counter_cleaning)
                         label_names_test.pop(counter_cleanskmultilearning)
                         print("Removing the following label : ", label_train)
                     else:
                         counter_cleaning += 1
```

**Figure 44**: la fonction cleaning ()

Le résultat de la fonction est :

- Supprimer les classes PDOC00660 et PDOC50199.

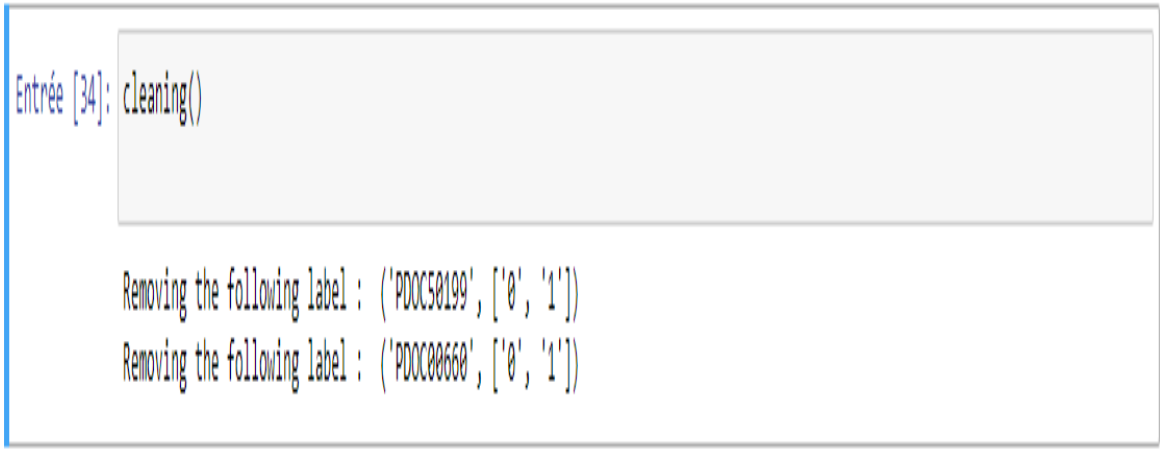



**Figure 45:** Le résultat de la fonction cleaning ()

### 3.2.4- Phase d'apprentissage de classifieur

La 4ème étape du programme consiste à la classification multi-label qui se décompose en plusieurs parties.

**a) Phase d'apprentissage**

Cette partie consiste à l'apprentissage du classifieur qui utilise la régression logistique (binary relevance) pour construire un modèle pour chaque label.

```
clf = BinaryRelevance(LogisticRegression())
clf.fit(X_train, y_train)
```

**Figure 46:** l'apprentissage du classifieur

**Clf :** L'instance d'estimateur clf (pour classifier) est d'abord ajustée au modèle ; c'est-à-dire qu'il doit apprendre du modèle. Cela se fait en passant notre ensemble d'entraînement à la méthode fit ().

### 3.2.5- Phase de test

Pour tester notre classification en a fait une expérimente pour prédicter les classes des 4 premières protéines

```
Entrée [8]: print(X_test.shape)
            X =X_test[0:4,:]
            print(X)

            (199, 1184)
            [[649.   1.   1. ...   1.   1.   1.]
             [ 46.   1.   1. ...   1.   1.   1.]
             [356.   1.   1. ...   1.   1.   1.]
             [382.   1.   1. ...   1.   1.   1.]]
```

**Figure 47:** l'initialisation des 4 premières protéines

```
Entrée [9]: y_pred = clf.predict(X).toarray()
            y_pred

Out[9]: array([[0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0],
                   [1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0],
                   [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0],
                   [0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                    0, 0, 0]], dtype=int64)
```

**Figure 48:** la prédiction des 4 premières protéines en vecteur 0 et 1

Les classes sont des codes prosite(protéine documentation ) donc pour voir le nom de classe et la position dans le corp humain il faut tapez le code de la classe dans la recherche de site :

**https://prosite.expasy.org/**

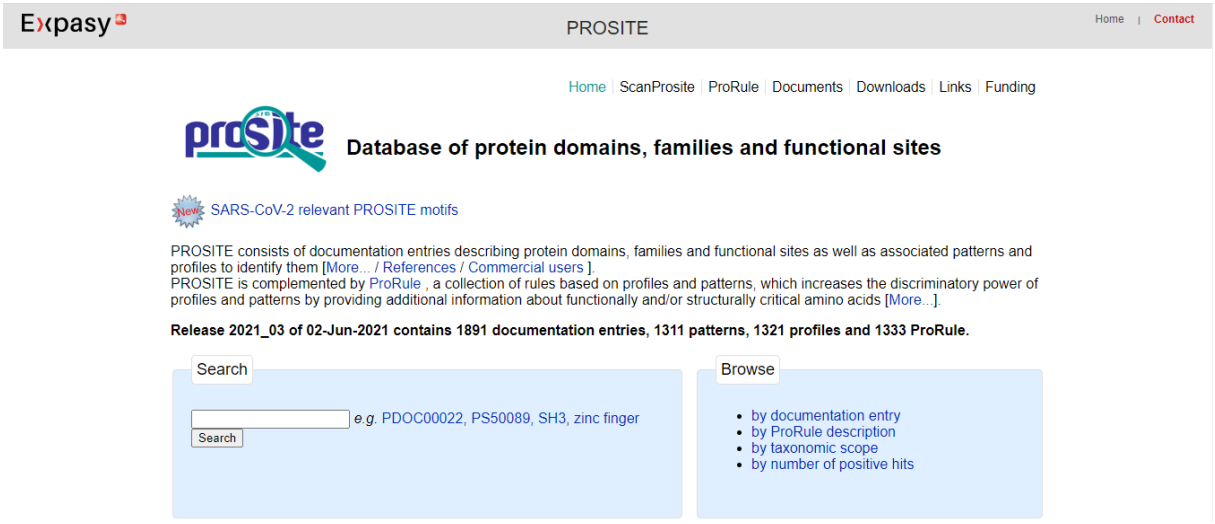



**Figure 49:** Prosite

Apres de tapez tout les classes et trouver leur nom et positionnement, on crée deux liste une des noms des classes et l'autre les positions.

On a insérer ces dernier  avec le même ordre des classes dans la base (PDOC)

```
Entrée [10]: cl=["Cyclophilin","Kinesin motor domain signature and profile","Pertactine","HMG-CoA ","Hsp20","C2 domain","Phosphoinositide phos

Entrée [11]: po=["nucleus","microtubule","tracheal epithelial cells","Mevalonate pathway","muscle tissue","membrane binding face","metabolism
```

**Figure 50:** les noms et les positions

Enfin on a fait la prédiction finale des positions des 4 premières protéines

```
Entrée [44]:
for pre in y_pred :
    for i in range (len(pre)):
        if pre[i]==1:
            print (label_names_test[i],"name:    ",cl[i],"        ","location:  ",po[i])
    print()

('PDOC00064', ['0', '1']) name:     HMG-CoA           location:   Mevalonate pathway

('PDOC00154', ['0', '1']) name:     Cyclophilin          location:   nucleus

('PDOC00791', ['0', '1']) name:     Hsp20          location:   muscle tissue

('PDOC00791', ['0', '1']) name:     Hsp20          location:   muscle tissue
```

**Figure 51:** la prédiction finale

### 3.2.6-Phase d'évaluation

Dans cette phase nous avons effectué les fonctions d'évaluation tel que **precision**, **accuracy**, **recall**, **F1-mesurent** et **Hamin_loss** car Hamming_loss représente le pourcentage de labels mal prédits**.** Avant nous allons expliquer les principes des fonctions d'évaluations utilisées. Pour les foncions d'évaluation tel que la **precision, recall**, et F**1-score** on a le paramètre « **average »** qui prend deux valeurs (**micro** et **macro**) dans le cas de la classification multi-label. La valeur micro permet de calculer le score globalement et prend en compte le déséquilibre des labels alors que la valeur macro calcule le score de chaque label sans retourner la moyenne pondérée des scores de tous les labels et aussi elle ne prend pas en compte le déséquilibre des labels. Compte tenu de cela nous avons opté pour la valeur micro pour le paramètre « **average** » de nos fonctions d'évaluation (**precision, recall**, et **F1-score).**

- **Hamming loss:** La métrique de Hamming loss pour l'ensemble d'étiquettes est définie comme la fraction d'étiquettes dont la pertinence est mal prédite
- **Accuracy:** La métrique de précision donne un degré moyen de similitude entre les ensembles d'étiquettes de vérité prédite et terrain.

- **Précision :** la métrique de précision calcule la proportion de prédictions réellement positives

- **Recall :** cette métrique estime la proportion de vrais libellés qui ont été prédits comme positifs

- **Mesure F1 :** la mesure F1 est définie comme la moyenne harmonique de la précision et du rappel. Il est calculé comme :

```
def print_pred(y, title):
    print(title)
    print("hamming loss: ")
    print(hamming_loss(y_test, y))
    print("accuracy:")
    print(accuracy_score(y_test, y))
    print("f1 score:")
    print("micro")
    print(f1_score(y_test, y, average='micro'))
    print("macro")
    print(f1_score(y_test, y, average='macro'))
    print("precision:")
    print("micro")
    print(precision_score(y_test, y, average='micro'))
    print("macro")
    print(precision_score(y_test, y, average='macro'))
    print("recall:")
    print("micro")
    print(recall_score(y_test, y, average='micro'))
    print("macro")
    print(recall_score(y_test, y, average='macro'))
```

**Figure 52:** code d'évaluation

```
Entrée [37]: y_pred = clf.predict(X_test).toarray()
             accuracy = print_pred(y_pred, "genbase")

             genbase
             hamming loss:
             0.001407035175879397
             accuracy:
             0.964824120603015
             f1 score:
             micro
             0.9853862212943633
             macro
             0.7147816905126446
             precision:
             micro
             1.0
             macro
             0.72
             recall:
             micro
             0.9711934156378601
             macro
             0.7101190476190475
```

**Figure 53:** Résultat d'évaluation

Ou l'instruction **y_pred = clf.predict(X_test).toarray ()** est l'instruction de prédiction de classfieur.

### 3.3- matrice de confusion

Une matrice de confusion est une matrice qui permet de mesurer la qualité d'un modèle de classification**.**

- Sur **les lignes** de la matrice, on trouve les classes réelles.
- Sur **les colonnes**, on retrouve les prévisions calculées par le modèle.

```
Entrée [36]: def print_confusion_matrix(confusion_matrix, axes, class_label, class_names, c_accuracy, fontsize=12):
                 df_cm = pd.DataFrame(confusion_matrix, index=class_names, columns=class_names, )
                 try:
                     heatmap = sns.heatmap(df_cm, annot=True, fmt="d", cbar=False, ax=axes)
                 except ValueError:
                     raise ValueError("Confusion matrix values must be integers.")
                 heatmap.yaxis.set_ticklabels(heatmap.yaxis.get_ticklabels(), rotation=0, ha='right', fontsize=fontsize)
                 heatmap.xaxis.set_ticklabels(heatmap.xaxis.get_ticklabels(), rotation=45, ha='right', fontsize=fontsize)
                 axes.set_ylabel('Real label')
                 axes.set_xlabel('Predicted label')
                 axes.set_title("Class : " + class_label[0] + ", Acc = " + "{:.2%}".format(c_accuracy))
```

**Figure 54:** code de matrice de confusion

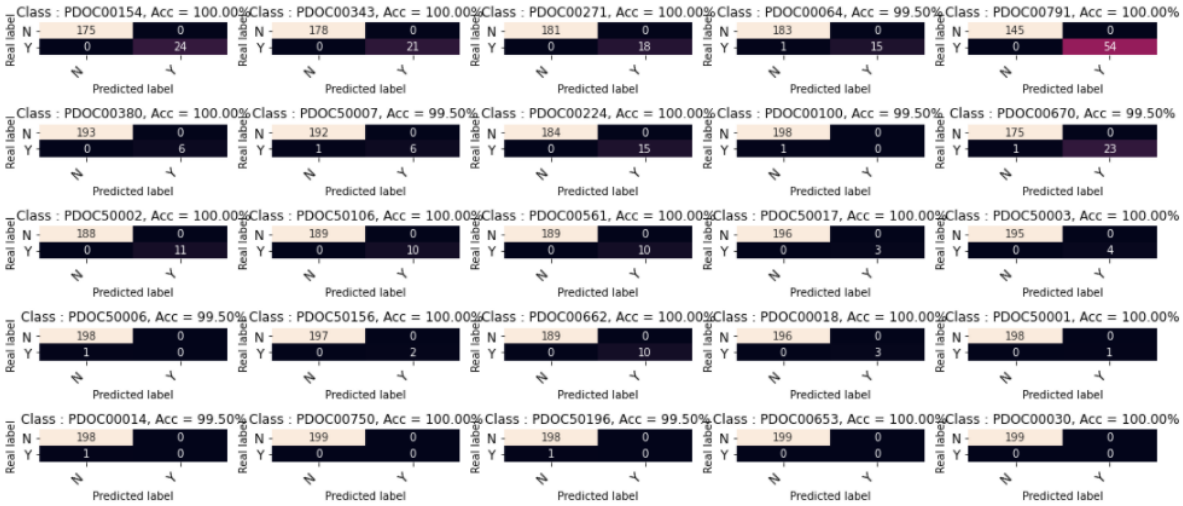



**Figure 55:** matrice de confusion

# Conclusion

Dans cette section du mémoire nous avons présenté les outils utilisés et les différentes parties de notre phase d'implémentation. Ainsi nous avons présenté quelques bibliothèques qui ont été indispensables et nous avons interprété les différents résultats obtenus par une évaluation pour déterminer des valeurs optimales qui optimisent des fonctions de mesures.

# CONCLUSION ET PERSPECTIVES

L'objectif principal de ce mémoire a été d'expérimenter un algorithme de classification multi-label. Proposons différentes perspectives pour classifier et prédire la position des proteines dans le corp humain.
Les démarches que nous avons eu à mener à savoir de la définition de la bio-informatique et plus précisément la classification des protéines selon leur position, jusqu'à l'implémentation de la méthode.

Durant cette phase de ce mémoire nous avons fait une généralité sur la bio-informatique, l'apprentissage automatique (machine learning) et la classification multi-label. La méthode que nous avons eu a utilisé est la classification multi-label (binary relavence BR), un classifieur obtenu a été tester sur une très grande base de données qui est celle (Genbase), le classifieurr nous a donnée des très bons résultats avec un taux de 0.96 une précision 96%.

Ce fut une tâche assez difficile, mais c'était très intéressant en même temps et cela m'a permis de découvrir les vraies difficultés qui peuvent être rencontrées dans les problèmes de classification multi-labels. Les ensembles de données multi-labels souffriraient très probablement d'un déséquilibre des étiquettes, donc comme perspectives de recherches futures, nous envisageons :

- De développer des techniques pour améliorer les resultats(accuracy)
- De tester le maximum de classifieurs qui sont adaptées aux ensembles de données multi-étiquettes.

# Bibliographie

## Bibliographie

[WEB 1]Apprentissage automatique. (s.d.). Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Apprentissage_automatique

[WEB 2] image apprentissage non Supervise. (s.d.). Récupéré sur github
https://projeduc.github.io/intro_apprentissage_automatique/introduction.html

[WEB 3] apprentissage Supervise. (s.d.). Récupéré sur math.univ-toulouse
https://www.math.univ-toulouse.fr/~agarivie/mydocs/apprentissageSupervise.pdf

[WEB 4] image apprentissage Supervise. (s.d.). Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Fichier:Apprentissage_Supervis%C3%A9_Vs_Non_Supervis

[WEB 5] apprentissage non Supervise. (s.d.). Récupéré sur dataanalyticspost
https://dataanalyticspost.com/Lexique/apprentissage-non-supervise/

[WEB 6] Apprentissage_semi-supervis. (s.d.). Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Apprentissage_semi-supervis

[WEB 7] image Apprentissage_semi-supervis. (s.d.). Récupéré sur hebergementwebs
https://www.hebergementwebs.com/blockchain/quatre-nouvelles-methodes-d-apprentissage-automatique-pour-analyser-les-ensembles-de-donnees-blockchain

[WEB 8] apprentissage par renforcement.(s.d.).Récupéré sur wikipedia
https://en.wikipedia.org/wiki/Reinforcement_learning

[WEB 9] image apprentissage par renforcement.(s.d.).Récupéré sur
https://www.slideshare.net/seml147/apprentissage-par-renforcement-85757171

[WEB 10] Apprentissage_par_transfert.(s.d.).Récupéré sur slideshare
https://fr.wikipedia.org/wiki/Apprentissage_par_transfert

[hal archive 1] image Apprentissage_par_transfert.(s.d.).Récupéré sur
François Meunier, Christophe Marsala, Laurent Castanié. 3DRESC-TF : Apprentissage par transfert pour la réutilisation de connaissances en classification d'objets 3D. Conférence Nationale sur les Applications Pratiques de l'Intelligence Artificielle, Jul 2017, Caen, France. ffhal-01561517

[WEB 11] SVM.(s.d.).Récupéré sur scikit-learn
http://scikit-learn.org/stable/modules/svm.html
[WEB 12] image SVM.(s.d.).Récupéré sur learnopencv
https://learnopencv.com/support-vector-machines-svm/

[WEB 13] reseau de neurones artificiels.(s.d.).Récupéré sur lebigdata
https://www.lebigdata.fr/reseau-de-neurones-artificiels-definition

[WEB 14] image reseau de neurones artificiels.(s.d.).Récupéré sur wikiversity
https://fr.wikiversity.org/wiki/R%C3%A9seaux_de_neurones/Les_neurones_en_r%C3%A9seaux

[WEB 15] Arbres Decision.(s.d.).Récupéré sur cedric.cnam
http://cedric.cnam.fr/vertigo/cours/ml2/coursArbresDecision.html

[WEB 16] classification.(s.d.).Récupéré sur edureka
https://www.edureka.co/blog/classification-in-machine-learning/

[WEB 17] image classification.(s.d.).Récupéré sur datamahadev
https://datamahadev.com/classification-algorithms-explained-in-30-minutes/

[WEB 18] types of classification- n machine learning.(s.d.).Récupéré sur machinelearningmastery
https://machinelearningmastery.com/types-of-classification-in-machine-learning/

[WEB 19] multi-class classification one vs all one vs one.(s.d.).Récupéré sur towardsdatascience
https://towardsdatascience.com/multi-class-classification-one-vs-all-one-vs-one-94daed32a87b

[WEB 20] Multi-label classification.(s.d.).Récupéré sur wikipedia
https://en.wikipedia.org/wiki/Multi-label_classification

[hal archive 2] multi-label classification.(s.d.).Récupéré sur
Sawsan Kanj. Learning methods for multi-label classification . Machine Learning [stat.ML]. Université de technologie de Compiègne; Université Libanaise (Liban), 2013. English. tel-01435796

[WEB 21] image multi-label classification.(s.d.).Récupéré sur github
https://gombru.github.io/2018/05/23/cross_entropy_loss/

[WEB 22] image Binary-relevance.(s.d.).Récupéré sur esearchgate
https://www.researchgate.net/figure/Binary-relevance-problem-transformation
method_fig13_311470881

[WEB 23] image Label Powerset.(s.d.).Récupéré sur slideplayer
https://slideplayer.com/slide/4824574/ LP

[WEB 24] image support-vector-machine.(s.d.).Récupéré sur analyticssteps
https://www.analyticssteps.com/blogs/how-does-support-vector-machine-algorithm-works-machine-learning

[WEB 25] Bioinformatics.(s.d.).Récupéré sur genome
https://www.genome.gov/genetics-glossary/Bioinformatics
[WEB 26] Bioinformatics.(s.d.).Récupéré sur britannica
https://www.britannica.com/science/bioinformatics

[WEB 27] les banques des données.(s.d.).Récupéré sur morissardjerome
http://morissardjerome.free.fr/infobiogen/www.infobiogen.fr/services/bdd/getdb_groupe7fa.html?group=-1

[WEB 28] Chromosome.(s.d.).Récupéré sur wikipedia
https://en.wikipedia.org/wiki/Chromosome

[WEB 29] image Chromosome.(s.d.).Récupéré sur medicinus
https://www.medicinus.net/chromosomes/

[WEB 30] ADN.(s.d.).Récupéré sur ctu-environnement
https://www.actu-environnement.com/ae/dictionnaire_environnement/definition/acide_desoxyribonucleique_adn.php4

[WEB 31] composant ADN.(s.d.).Récupéré sur unamur
https://www.unamur.be/sciences/enligne/transition/biologie/Fichesderevision/revision2%20fonctionnement/adn.htm#

[WEB 32] image Brin d'ADN avec ses bases azotées.(s.d.).Récupéré sur genomequebec-education
http://www.genomequebec-education-formations.com/education-concepts-adn

[WEB 33] ARN.(s.d.).Récupéré sur wikipedia
https://en.wikipedia.org/wiki/RNA

[WEB 34] image ARN.(s.d.).Récupéré sur dreamstime
https://fr.dreamstime.com/structure-mol%C3%A9culaire-l-adn-arn-illustration-%C3%A9ducative-vecteur-d-infographic-image110992446

[WEB 35] Types of RNA mRNA rRNA and tRNA.(s.d.).Récupéré sur news-medical
https://www.news-medical.net/life-sciences/-Types-of-RNA-mRNA-rRNA-and-tRNA.aspx

[WEB 36] image ARNm.(s.d.).Récupéré sur biochimej.univ-angers
http://biochimej.univ-angers.fr/Page2/COURS/7RelStructFonction/2Biochimie/1SyntheseProteines/1Synthese Prot.htm

[WEB 37] image ARNr.(s.d.).Récupéré sur aquaportail
https://www.aquaportail.com/definition-14593-arn-ribosomique.html

[WEB 38] image ARNt.(s.d.).Récupéré sur futura-sciences
https://www.futura-sciences.com/sante/definitions/genetique-arnt-657/

[WEB 39] Protéine.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Prot%C3%A9ine

[WEB 40] fabrication de Protéine.(s.d.).Récupéré sur futura-sciences
https://www.futura-sciences.com/sante/definitions/biologie-proteine-237/

[WEB 41] Structure de Protéine.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Structure_des_prot%C3%A9ines

[WEB 42] image des hélices φ.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/H%C3%A9lice_alpha

[WEB 43] image des feuillets ψ.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Feuillet_b%C3%AAta

[WEB 44] image de La structure tertiaire.(s.d.).Récupéré sur unice
http://sites.unice.fr/site/ffontaine/structurebio/co/chap3.html

[WEB 45] domaine.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Domaine_prot%C3%A9ique

[WEB 46] Noyau.(s.d.).Récupéré sur wikipedia
https://fr.wikipedia.org/wiki/Noyau_(biologie)

[WEB 47] image de Noyau.(s.d.).Récupéré sur news-medical
https://www.news-medical.net/life-sciences/Structure-and-Function-of-the-Cell-Nucleus-(French).aspx

[WEB 48] mevalonate pathway.(s.d.).Récupéré sur sciencedirect
https://www.sciencedirect.com/topics/biochemistry-genetics-and-molecular-biology/mevalonate-pathway

[WEB 49] Tissu musculaire.(s.d.).Récupéré sur pubmed.ncbi
https://pubmed.ncbi.nlm.nih.gov/18630139/

[WEB 50] image Tissu musculaire.(s.d.).Récupéré sur brainkart
https://www.brainkart.com/article/Muscle-Tissue_33169/

[WEB 51] cytoplasm.(s.d.).Récupéré sur nature
https://www.nature.com/scitable/definition/cytoplasm-280/

[WEB 52] image cytoplasm.(s.d.).Récupéré sur readbiology
https://readbiology.com/cytoplasm/

[WEB 53] membrane.(s.d.).Récupéré sur ck12
https://www.ck12.org/book/ck-12-biology-advanced-concepts/section/3.12/

[WEB 54] python.(s.d.).Récupéré sur journaldunet
https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/
[WEB 55] image python.(s.d.).Récupéré sur lebigdata
https://www.lebigdata.fr/python-langage-definition

[WEB 56] anaconda.(s.d.).Récupéré sur anaconda
https://docs.anaconda.com/anaconda/navigator/

[WEB 57] jupyter-notebook.(s.d.).Récupéré sur jupyter-notebook-beginner-guide.readthedocs https://jupyter-notebook-beginner-guide.readthedocs.io/en/latest/what_is_jupyter.html

[WEB 58] image jupyter-notebook.(s.d.).Récupéré sur towardsdatascience https://towardsdatascience.com/the-complete-guide-to-jupyter-notebooks-for-data-science-8ff3591f69a4

[WEB 59] numpy.(s.d.).Récupéré sur numpy https://numpy.org/doc/stable/user/whatisnumpy.html

[WEB 60] Scikit-learn.(s.d.).Récupéré sur wikipedia https://en.wikipedia.org/wiki/Scikit-learn

[WEB 61] scikit-multilearn.(s.d.).Récupéré sur awesomeopensource https://awesomeopensource.com/project/scikit-multilearn/scikit-multilearn

[WEB 62] keras.(s.d.).Récupéré sur deepai https://deepai.org/machine-learning-glossary-and-terms/keras

[WEB 63] pandas.(s.d.).Récupéré sur educative https://www.educative.io/edpresso/what-is-pandas-in-python

[WEB 64] math.(s.d.).Récupéré sur stackabuse https://stackabuse.com/the-python-math-library

[WEB 65] seaborn.(s.d.).Récupéré sur towardsdatascience https://towardsdatascience.com/seaborn-python-8563c3d0ad41