République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Abbes LAGHROUR KHENCHELA
Faculté des Sciences et de Technologie
Département des Mathématiques et Informatique

Mémoire présenté en vue de l'obtention du diplôme de Master
Spécialité :Sécurité et Technologie web
Filière :Informatique
Domaine : Mathématiques et Informatique

THÈME

Application des méthodes d'apprentissage automatique pour l'inférence des réseaux de régulation génétique

Présenté par :

Laghmassi Leila

Sekkiou Amel

Dirigé par :

M.AZIZI NABIL

Promotion : 2016/2017

Remerciements

En premier lieu nous tenons à remercier dieu le tout puissant qui nous donner la volonté et la force pour accomplir ce travail.

Nous tenons à remercier M.AZIZI Nabil qui a encadré ce mémoire. Nous le remercie d'abord pour son encadrement, ses conseils avisés, tant scientifiques que pédagogiques et pour la patience, la gentillesse avec lesquelles il les a prodigués, pour son soutien, sa disponibilité et pour la collaboration étroite dans laquelle nous avons travaillé et son aide qui nous a permis de mener à bien ce mémoire.

Nous remercier également tous les enseignants du département « informatique » qui ont contribué à notre formation.

Enfin, nous remercions nos familles, nos amis et tous ceux qui, de près ou de loin, nous ont soutenu.

Dédicace

A ceux qui mon offre le bonheur :

Mon très chère père

Ma très chère mère

Mes très chères frères :Aymen et Houssam

Mes très chères sœurs :Hend ,Naima,Sabrina

A tous les personnes qui m'aiment

Laghmassi Leila

Dédicaces

A ceux qui mon offre le bonheur :

Mon très chère père

Ma très chère mère

Mes très chères frères :Omer et Abdaljalil

Mes très chères sœurs :semouna et assia, dalal,la petite belle Ikram

A tous les personnes qui m'aiment

Sekkiou Amel

Résumé

Ces dernières années ont vu une extension rapide en terme de données biologique (séquences ADN, protéines, gènes ...etc). L'importante quantité d'information disponible a naturellement rapproché l'informatique et la biologie a priori distincts pour créer la bioinformatique. L'enjeu majeur dans ce contexte est de disposer des outils permettant l'analyse d'une manière fiable et efficace de ce type de données. Le data mining dans sa forme fondamentale est d'extraire une connaissance non triviale, implicite, précédemment inconnue et potentiellement utile à partir d'une grande quantité de données. L'application des techniques de data mining à la bioinformatique a fait la naissance d'une nouvelle discipline émergente appelée biodata mining. La régulation génétique est un processus biologique, à l'intérieur de la cellule, qui définit si la transcription d'un gène est activée ou inhibée. Ce mécanisme de régulation impliquant des milliers de gènes permettent aux cellules vivantes de s'adapter aux changements de l'environnement afin de se nourrir, de se développer . . . etc. La compréhension des réseaux de régulation génétique est ainsi nécessaire pour comprendre le mécanisme de fonctionnement de la cellule. L'étude expérimentale de est un défi couteux, en revanche les méthodes d'apprentissage automatique représente une direction prometteuse qui à travers une démarche inductive visent à extraire les réseaux de régulation génétique à partir de données d'expression génétique sous forme de puce à ADN. Notre travail vise à explorer les potentiels des méthodes d'apprentissage automatique pour l'inférence des réseaux de régulation génétique.

Mots clés : bioinformatique, data mining , apprentissage automatique , réseaux de régulation génétique, données d'expression génétique, puce à ADN.

Abstract

In last years we have seen a rapid extension in terms of biological data (DNA sequences, proteins, genes ... etc). The large amount of information available has naturally brought together computing and biology a priori distinct to create bioinformatics. The major challenge in this context is to have the tools to analyze this type of data in a reliable and effective way. Data mining in its fundamental form is to extract a non-trivial, implicit, previously unknown and potentially useful knowledge from a large amount of data. The application of data mining techniques to bioinformatics has spawned a new emerging discipline called biodata mining. Genetic regulation

is a biological process within the cell that defines whether the transcription of a gene is activated or inhibited. This regulatory mechanism involving thousands of genes allows living cells to adapt to changes in the environment in order to feed and develop. . . etc. Understanding of genetic regulation networks is thus necessary to understand the mechanism of functioning of the cell. Experimental study is a costly challenge, while automatic learning methods represent a promising direction which, through an inductive approach, aims to extract the genetic regulation networks from gene expression data in the form of a DNA chip . Our work aims to explore the potentials of automatic learning methods for the inference of genetic regulation networks.

Key words : bioinformatics, data mining, automatic learning, genetic regulation networks, gene expression data, DNA chip.

Table des matières

Liste des tableaux

Table des figures

Introduction générale

Aujourd'hui la bio-informatique commence à se faire connaître, cela fait un peu plus de dix ans qu'elle se développe.De façon très générale, on peut inclure dans une définition de la bio-informatique toutes les applications de l'informatique à la biologie. Celles-ci sont extrêmement nombreuses, et incluent des domaines aussi différents que l'étude *in silico* de la connectique des neurones, le traitement quantitatif et qualitatif d'images microscopiques, la gestion des échantillons et des données expérimentales dans les grands laboratoires industriels, ou la modélisation de l'évolution de populations animales dans des conditions écologiques spécifiques. Dans tous ces cas, l'informatique apporte des outils indispensables à l'analyse de phénomènes biologiques, à la formulation de nouvelles hypothèses, ou à la gestion de données expérimentales. De plus en plus couramment, les biologistes font appel à l'informatique pour les aider à résoudre des problèmes à tous les niveaux, et créent donc des interfaces multiples entre le monde de la biologie et celui de l'informatique. D'autre part,la croissance des données biologique tel que cellule,gène,protéine,séquence. . .,ce quantité de données besoin des supports de stockage(puce ADN) et aussi des systèmes informatique pour manipuler ces données. La bioinformatique n'est pas une discipline en soi. Elle résulte du besoin des biologistes d'analyser les données qu'ils produisent en quantités de plus en plus importantes, et d'intégrer ces données dans un cadre scientifique rigoureux.

L'ensemble des interactions concertées entre ces entités constituent des réseaux de régulation biologique dont l'élucidation est l'un des objectifs majeurs de la biologie des systèmes. Elle repose essentiellement sur la mise en évidence et la caractérisation à un niveau global des relations entre ces entités.

Parmi les différents mécanismes de régulation à l'œuvre dans la cellule, il est communément admis que la régulation transcription elle joue un rôle prépondérant. Ce processus de régulation génétique est d'autant plus important qu'il est pour l'instant le plus aisément observable, du fait notamment de la disponibilité de techniques expérimentales adaptées telles que les puces à ADN. Les données qui en sont issues rendent compte de l'activité transcriptionelle de l'ensemble des gènes d'un échantillon de cellules dans des conditions expérimentales spécifiques. L'exploitation de ces données doit permettre l'extraction de connaissances en vue d'améliorer la compréhension de certains processus normaux ou pathologiques et de cerner de nouvelles cibles thérapeutiques. Ces travaux de recherche concernent l'apprentissage automatique des réseaux de

régulation transcriptionnelle, à partir de données de transcriptome. Cette tâche est généralement entreprise de la manière suivante. Dans un premier temps, une classe de modèles mathématiques permettant de d'écrire les interactions entre des gènes régulateurs et leurs gènes cibles est choisie. Les données sont ensuite utilisées afin d'apprendre à la fois le graphe d'interaction et les paramètres du modèle représentant le réseau de régulation.

Organisation du mémoire

Ce manuscrit se compose de quatre chapitre :

— Le premier chapitre est consacré, en premier lieu,à la définition de quelques notions importantes comme cellule, gène,protéine...,plus quelque définition important tel que la bioinformatique et leurs axes de recherche.

— Dans le deuxième chapitre nous nous sommes intéressé à le Data Mining et leurs processus et tâches tel que la classification,et certains méthodes d'apprentissage automatique soit en approche non supervisée ou semi-supervisée ou supervisée.

— Dans le troisième chapitre présente quelque travaille proposée dans cet axe de recherche.

— Le quatrième chapitre, quant à lui, est consacré à la présentation de notre contribution pour l'inférer des réseaux de régulation génétique.

Chapitre 1

Les notions biologique

Introduction

De nos jours, la biologie s'inscrit dans une perspective systémique selon laquelle les cellules, qui sont les composantes de base du vivant, sont régies par le fonctionnement coordonné de multiples entités : gênes, protéines et métabolites. L'ensemble des interactions concertées entre ces entités constituent des réseaux de régulation biologique dont l'élucidation est l'un des objectifs majeurs de la biologie des systèmes. Elle repose essentiellement sur la mise en évidence et la caractérisation à un niveau global des relations entre ces entités. Parmi les différents mécanismes de régulation à l'œuvre dans la cellule, il est communément admis que la régulation transcriptionnelle joue un rôle prépondérant. Ce processus de régulation génétique est d'autant plus important qu'il est pour l'instant le plus aisément observable, du fait notamment de la disponibilité de techniques expérimentales adaptées telles que les puces à ADN. Les données qui en sont issues rendent compte de l'activité transcriptionnelle de l'ensemble des gênes d'un échantillon de cellules dans des conditions expérimentales spécifiques. L'exploitation de ces données doit permettre l'extraction de connaissances en vue d'améliorer la compréhension de certains processus normaux ou pathologiques et de cerner de nouvelles cibles thérapeutiques. Ces travaux de recherche concernent l'apprentissage automatique des réseaux de régulation transcriptionnelle, à partir de données de transcriptome.

1.1 Qu'est-ce qu'une cellule ?

Une cellule est l'élément de base fonctionnel et structural qui compose les tissus et les organes des êtres vivants. Elle contient l'information génétique de l'individu et est à l'origine de la création biologique. Complexe, elle est constituée de divers éléments dont une membrane qui lui permettent d'être autonome. Ce qui ne l'empêche pas d'entrer en interaction avec les autres cellules. Les êtres vivants sont créés à la base par une unique cellule qui se divise par un phénomène itératif (appelé mitose) pour composer le corps humain. La multiplication des cellules est un phénomène régulé qui permet le développement du corps humain et des différents organes. La prolifération anormale et anarchique de cellules constitue un élément fondamental pour la constitution d'un cancer.[3]

[2]

1.2 Acide aminé (AA)

Petite molécule dont l'enchaînement compose les protéines - on dit qu'une protéine est un polymère d'acides aminés (les monomères). Il existe 20 acides aminés différents utilisés pour fabriquer les protéines.[26]

1.3 Gène

Fragment d'ADN portant les informations nécessaires à la fabrication d'une ou plusieurs protéine(s). Un gène comprend la séquence en nucléotides qui sera transcrite puis traduite en acides aminés, mais aussi des séquences permettant de réguler cette fabrication de protéine en fonction des conditions cellulaires. La longueur d'un gène peut varier de quelques centaines, à plus d'un million de nucléotide.[26]

1.4 Génome

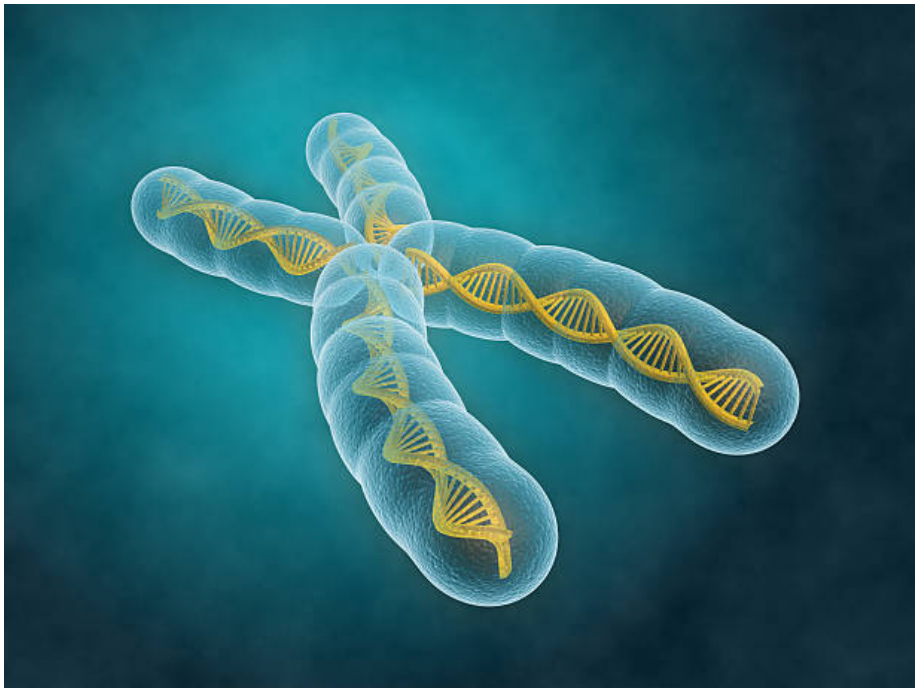
Ensemble de l'information génétique d'un organisme. Une copie du génome est présente dans chacune de ses cellules. Le génome est transmis de génération en génération. Par extension, le génome se réfère aussi au support physique de cette information génétique, c'est-à-dire la macromolécule d'ADN.[26]

1.5 Génomique

Étude des génomes. Son objectif est de séquencer l'ADN d'un organisme et de localiser sur celui-ci tous les gènes qu'il porte, puis de caractériser leurs fonctions.[26]

1.6 Chromosome

Les chromosomes sont les bâtonnets contenus dans le noyau de chaque cellule du corps et qui portent l'information génétique (on dit aussi le matériel génétique). Toutes les cellules du corps comportent 22 paires de chromosomes. [4]



[5]

1.7 Acide nucléique

Polymère formé par l'enchaînement de nucléotides. Les acides nucléiques jouent un rôle fondamental dans le stockage, le maintien et le transfert de l'information génétique. Il existe deux types d'acide nucléique : l'acide ribonucléique (ARN) et l'acide désoxyribonucléique(ADN).[26]

1.8 Acide désoxyribonucléique (ADN)

Support biochimique de l'information génétique chez tous les êtres vivants (à l'exception de quelques virus qui utilisent l'ARN). Principal composant des chromosomes, l'ADN se présente

le plus souvent sous forme de deux longs filaments (ou chaînes) torsadés l'un dans l'autre pour former une structure en double hélice. Chacune de ces chaînes est un polymère formé de l'assemblage de quatre nucléotides différents, désignés par l'initiale de la base azotée qui entre dans leur composition : A (Adénine), C (Cytosine), G (Guanine) et T(Thymine).[26]

1.9 Acide ribonucléique (ARN)

Dans les cellules, on distingue plusieurs types d'ARN suivant leur fonction. Les trois types principaux sont : les ARN messagers, les ARN de transfert et les ARN ribosomaux. L'ARN est un acide nucléique constitué d'une seule chaîne de nucléotides, de structure analogue à celle de l'ADN. Il existe cependant des différences chimiques entre ces deux acides nucléiques qui donnent à l'ARN certaines propriétés particulières. L'ARN est produit par transcription de l'ADN.[26]

1.9.1 ARN messager (ARNm)

Photocopie du gène, il sert à transférer l'information génétique de son lieu de stockage (le chromosome) jusqu'au lieu de synthèse des protéines (les ribosomes). Les ARNm des cellules eucaryotes doivent subir une maturation, comprenant souvent un processus d'excision de leurs introns et d'épissage de leurs exons avant leur traduction en protéines.[26]

1.9.2 ARN ribosomal (ARNr)

Constituant principal des ribosomes, la machinerie cellulaire où a lieu la traduction en protéines de l'information contenue dans les ARNm.[26]

1.9.3 ARN de transfert (ARNt)

Petits ARN responsables du transport des acides aminés jusqu'aux ribosomes lors de la traduction des ARNm : chaque ARNt transporte un acide aminé, de façon spécifique. Sa séquence comporte une série de trois nucléotides, nommée anticodon, qui reconnaît le codon (cf code génétique) correspondant à l'acide aminé qu'il transporte.[26]

1.10 Promoteur

Courte séquence spécifique d'ADN, située au début des gènes, sur laquelle se fixe l'enzyme qui effectue la transcription (l'ARN polymérase). Etant nécessaire pour que la transcription débute, le promoteur est indispensable au fonctionnement d'un gène.[26]

1.11 Protéine

L'un des quatre matériaux de base de tout organisme, avec les glucides, les lipides et les acides nucléiques. Les protéines sont formées d'un enchaînement spécifique d'acides aminés (de quelques dizaines à plusieurs centaines).[26]

1.12 Ribosomes

Machinerie cellulaire, constituée de protéines et d'ARN (les ARNr), responsable de la traduction des ARNm.[26]

1.13 Nucléotide

Motif structural de base (monomère) des acides nucléiques, formé de l'assemblage de plusieurs molécules : un sucre (ribose pour l'ARN, désoxyribose pour l'ADN), un acide phosphorique et une base azotée (dans le cas de l'ARN cette base peut être l'Adénine - A, la Cytosine - C, la Guanine - G ou l'Uracile - U ; idem dans le cas de l'ADN, excepté que l'Uracile est remplacé par la Thymine - T).[26]

1.14 Facteur de transcription(TF)

Les facteurs de transcription sont les protéines qui permettent la transcription,donc sont des régulateurs essentiels et spécifiques du processus d'expression des gènes en fonction des besoins de la cellule.[22]

1.15 EXPRESSION DE L'INFORMATION GENETIQUE

1.15.1 La molécule d'ADN

Les molécules d'ADN sont les plus grosses molécules du monde vivant et sont présentent dans tous les organismes vivants. Une molécule d'ADN est une double hélice composée de deux brins enroulés l'un autour de l'autre ; on dit que l'ADN est bicaténaire (contrairement à l'ARN, qui est monocaténaire).

Chacun de ces brins est constitué d'un enchaînement de bases dites puriques (guanine, G ; adénine, A) et pyrimidiques (cytosine, C ; thymine, T).Les bases sont reliées entre elles à l'intérieur d'un brin d'ADN par des sucres des oses ,appelés désoxyriboses, et par des acides phosphoriques.[29]

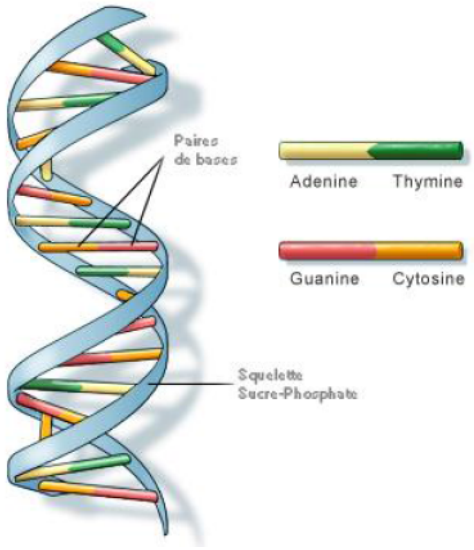
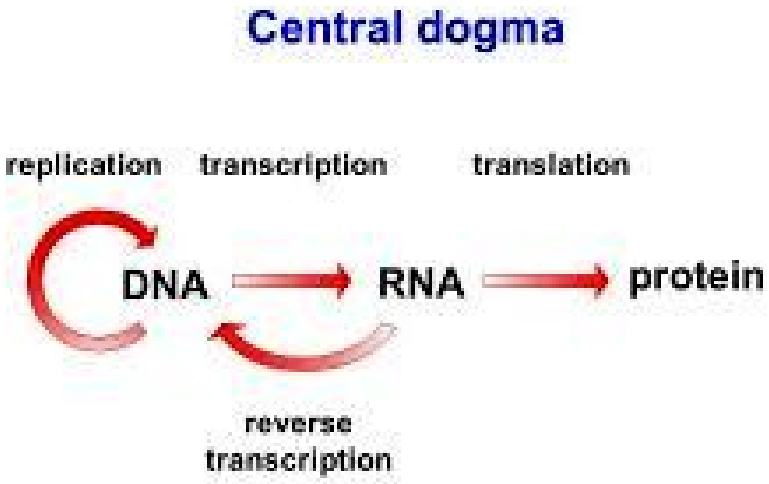


Figure 1.1 – Structure de l'ADN
[29]

1.16 Le dogme central

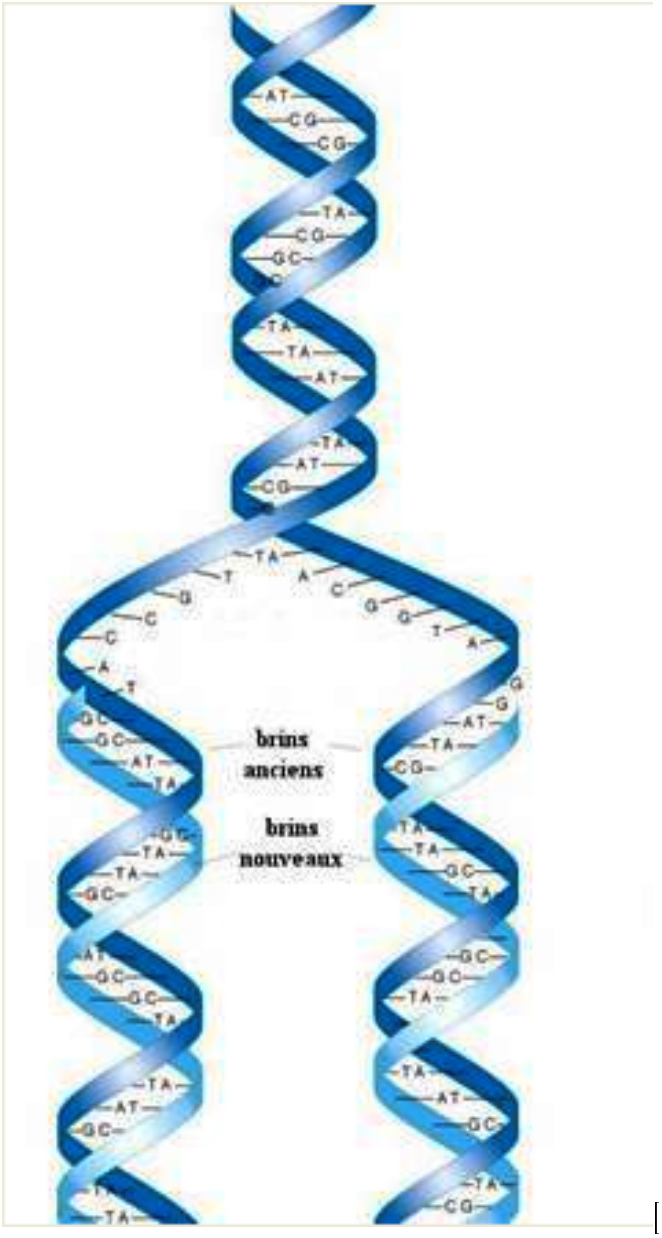


[8]

1.16.1 La réplication

Mécanisme de synthèse de l'ADN permettant de transmettre l'information génétique d'une cellule ou d'un organisme à sa descendance. Chaque molécule-fille d'ADN est constituée d'un

brin de la molécule-mère, qui sert de modèle à un nouveau brin. Ceci conduit à la duplication des molécules d'ADN de tout le génome.[22]



[?]

1.16.2 La transcription

Processus permettant la copie de l'ADN en ARN, ou de l'ARN en ARN messager dans le cas de certains virus. C'est la première étape du processus qui permet de passer de l'ADN à la protéine, ou plus concrètement du gène à son produit.[1]

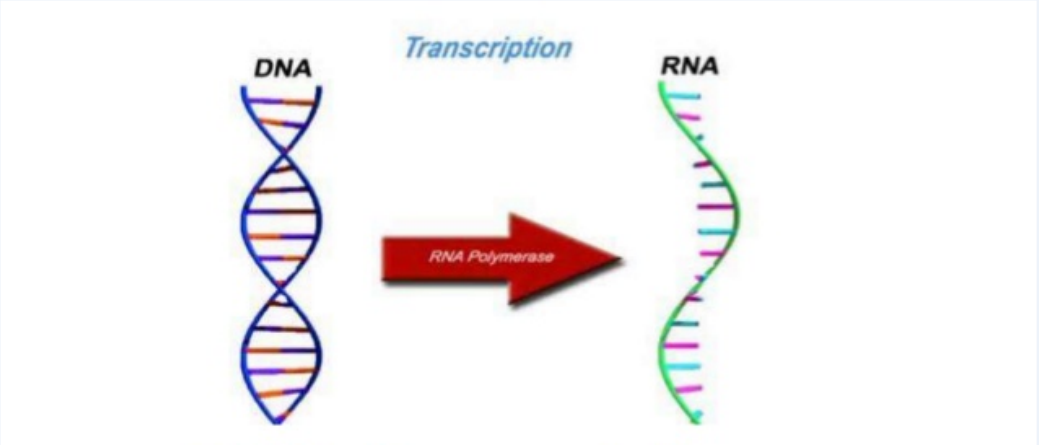


Figure 1.2 – DNA Transcription
[1]

1.16.3 La molécule d'ARN :

La molécule d'ARN est constituée d'un enchaînement de ribonucléotides : l'adénine (A), la cytosine (C), la guanine (G) et l'uracile (U), reliés entre eux par des liaisons nucléotidiques (voir la Figure 1.3). L'ordre de ces nucléotides est dicté par la séquence des désoxyribonucléotides portés par la séquence ADN dont ils sont issus suite au processus de transcription. Les ribonucléotides de l'ARN diffèrent des désoxynucléotides de l'ADN par la présence d'un groupement OH en 20 du ribose (et non d'un H comme le désoxyribose de l'ADN, voir Figure 1.3), mais aussi par le fait que la thymine (T) est substituée par l'uracile (U).[29] À l'inverse de l'ADN, la plupart du temps structuré en double hélice, l'ARN peut adopter des conformations différentes (en simple brin, en tige boucle, dots) liées à sa fonction.

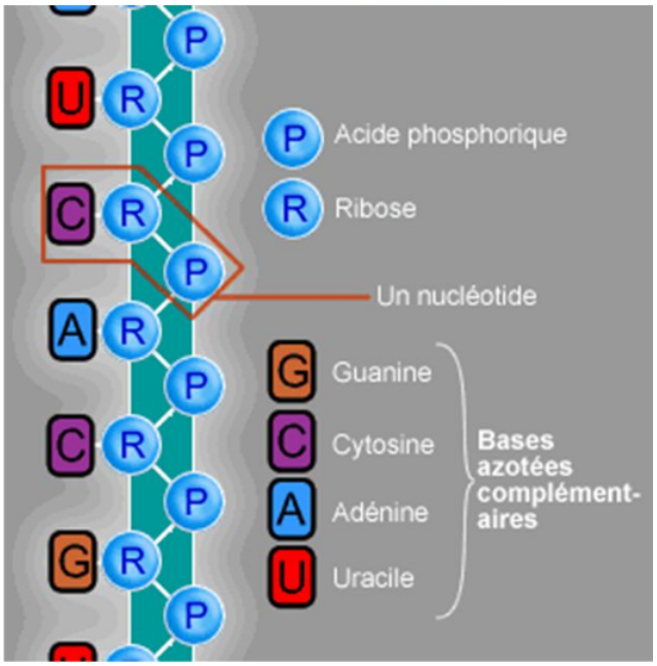


Figure 1.3 – La molécule d'ARN
[?]

1.16.3.1 De l'ARN aux Fonctions Biologiques

Tout comme l'alphabet de l'ADN, celui de l'ARN se compose de quatre bases complémentaires deux à deux. Comme expliqué dans la section précédente il est facile de déduire la séquence d'ARN qui sera issue d'une séquence ADN en utilisant cette complémentarité des bases. L'étape suivant la transcription est la traduction. Au cours de ce processus la molécule d'ARN, composée des quatre bases de son alphabet, est traduite en une protéine, dont l'alphabet compte 22 acides aminés. Il existe donc une table de traduction de la composition en bases azotées de la molécule d'ARN en acides aminés protéiques : c'est le **code génétique**.[29]

1.16.3.2 Le code génétique

On appelle code génétique le code de correspondance assurant la traduction du message génétique par chaque cellule. Concrètement, la molécule nommée ARN messager porte plusieurs séquences de bases azotées. Chaque trinôme de bases représente une séquence qui code ou symbolise un acide animé, à la base de la formation des protéines. Ainsi, le code génétique permet-il de traduire les informations cryptées dans notre matériel génétique afin de produire les protéines nécessaires au fonctionnement de notre organisme.[22]

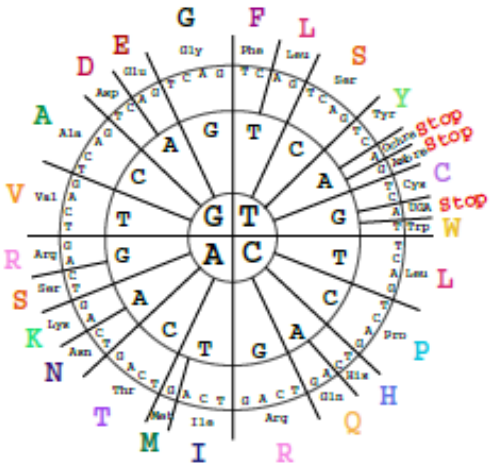
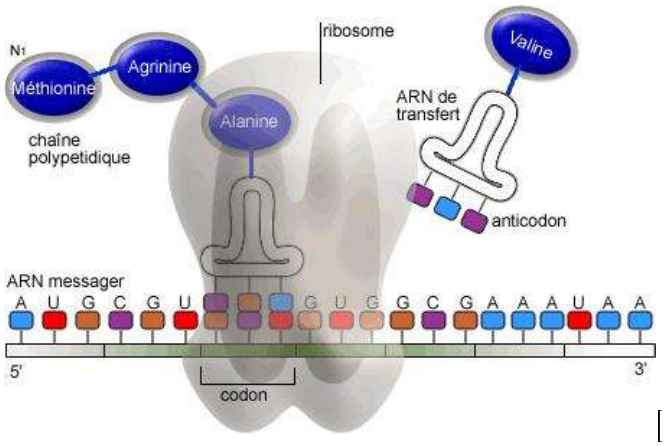


Figure 1.4 – Code génétique le plus fréquemment utilisé pour la traduction. [29]

1.16.4 La traduction

Étape de la synthèse (fabrication) des protéines au cours de laquelle le brin d'ARN messager obtenu lors de la transcription est converti en une chaîne d'acides aminés qui donnera une protéine. [26]



[10]

1.17 Bioinformatique

1.17.1 Définition :

La bio-informatique moderne est née de la convergence de deux aspects de la recherche en biologie : le stockage des séquences moléculaires sur ordinateurs sous la forme de bases données et

l'application d'algorithmes mathématiques pour l'alignement des séquences d'acides nucléiques et protéiques.

La plupart des définitions de la bio-informatique suggèrent l'interaction entre la biologie, les technologies de l'information et les sciences informatiques (les mathématiques).

D'après [15], « la bio-informatique est la discipline de l'analyse de l'information biologique, en majorité sous la forme de séquences génétiques et de structures de protéines. C'est le décryptage de la « bio-information » (« *Computational Biology* » en anglais) ».

Andrade et Sander, dans Bioi formatics : from genome data to biological knowledge, Current Opinion in Biotechnology (1997), présentent une définition plus large de la bio-informatique. Selon ces auteurs, « Bioinformatics is a science of recent creation that uses biological data, completed by computational methods, to derive new biological knowledge ».

Cette définition, plus moderne, sous-entend que la bio-informatique ne se limite évidemment pas à l'analyse des séquences. Un objectif fondamental est la volonté d'intégration de données de différentes natures, celles relatives aux séquences mais aussi celles concernant les marqueurs moléculaires, les données phénotypiques, etc.

La bioinformatique est une approche *in silico* de la biologie traditionnelle qui vient compléter les approches classiques *in situ* (dans le milieu naturel), *in vivo* (dans l'organisme vivant) et *invitro* (en éprouvette).

La bio-informatique est une branche théorique et pratique de la biologie. Sur le plan théorique, sa finalité est la synthèse des données biologiques à l'aide de modèles et de théories en énonçant des hypothèses généralisatrices et en formulant des prédictions. Sur le plan pratique, son but est de proposer des méthodes et des logiciels pour la sauvegarde, la gestion et le traitement de données biologiques. Par souci de clarté, les Anglo-saxons, utilisent deux termes pour distinguer ces deux aspects de la bio-informatique. Associé au terme de "bioinformatics" pour l'aspect pratique, ils utilisent le terme générique de « *biocomputing* » ("*computational biology*" pour les Américains) pour désigner l'aspect théorique.[30]

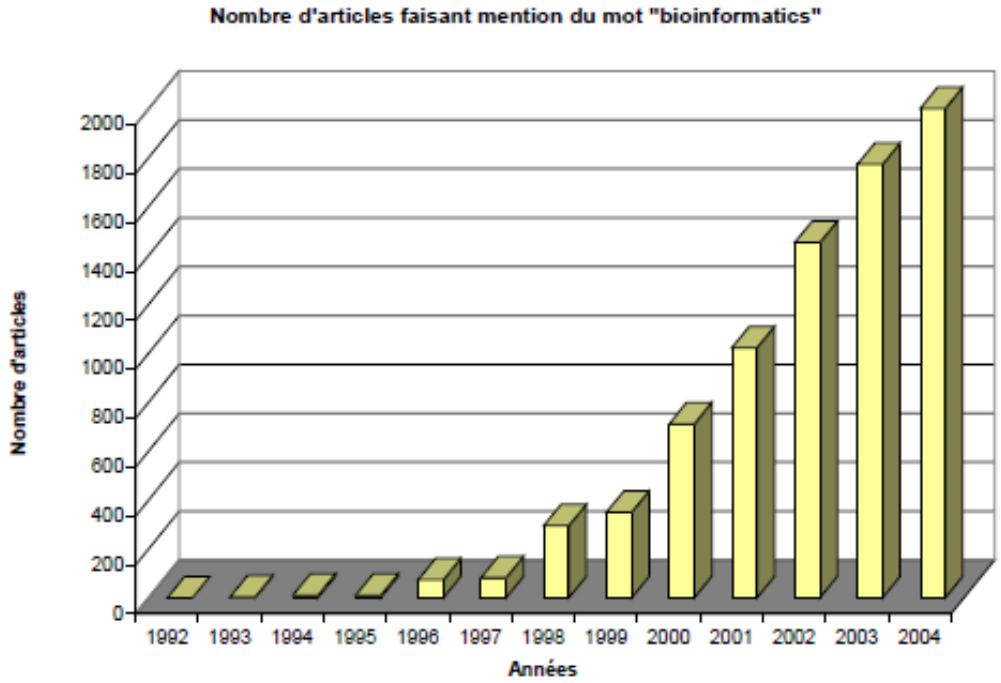


FIGURE 1.5 – La bio-informatique dans la littérature scientifique de 1992 à nos jours (source : PubMed). De 1992 à 2004, croissance exponentielle du nombre d'articles référencés dans PubMed sous le terme « bioinformatics ».

[30]

1.17.2 Histoire du terme « bio-informatique »

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

L'apparition de la bioinformatique n'est donc pas une conséquence de la génomique (séquençage d'un génome et son interprétation), mais plutôt une de ses fondations.[17]

1.17.3 Buts

La bioinformatique est l'étude de l'information biologique. Ce n'est pas simplement l'application à la biologie de l'informatique ; c'est une branche à part entière de la biologie. La bioinformatique actuelle se concentre surtout sur l'étude des séquences d'ADN et sur le repliement des protéines, donc travaille surtout au niveau moléculaire. De nombreux bioinformaticiens travalent également à l'élaboration d'outils biologiques permettant de résoudre des problèmes de l'informatique classique.[17]

1.17.4 Quelques problèmes de bioinformatique :

1.17.4.1 Alignements de séquences

Un alignement de séquences biologiques x et y (ADN, ARN ou protéine) consiste, à partir d'une distance définie pour chaque $couple(X_i; Y_i)$), à trouver le positionnement relatif d'une séquence par rapport à l'autre pour minimiser la somme des distances sur tous les couples (i.e avec un décalage constant k).[13]

[13]

Intuitivement on veut maximiser le nombre de lettres qui "correspondent" dans les deux séquences.

Deux versions du problème :

1. **Alignement global** : on prend deux séquences et on cherche le meilleur alignement sur l'ensemble des deux séquences. Exemple : comparer deux séquences d'ADN théoriquement identiques, entre deux individus proches.
2. **Alignement local** : on cherche un très bon alignement, mais on accepte qu'il ne concerne qu'une toute petite sous-partie des deux séquences. **Exemple :** trouver les parties communes de l'ADN de l'homme et de la truite.[13]

Biologiquement, à quoi ça sert ?

Les alignements de séquence sont très utilisés en biologie moderne. On considère que deux séquence proches ont un ancêtre commun récent et partagent donc, en plus d'une similarité de séquence, une similarité au niveau, par exemple, de la fonction biologique.

On peut ainsi chercher à aligner les séquences génétiques de l'homme et de la souris, pour pouvoir ensuite faire des expériences de génétique chez la souris, et en tirer des conclusions chez l'homme. Cela permet également d'étudier la génétique de certains organismes, et de dé-couvrir des traces de leur évolution récente (i.e duplication de gènes ou de génomes).[13]

1.17.4.2 Prédiction fonctionnelle de protéines :

Une pro. . . quoi ?

Une protéine est une chaîne linéaire d'acides aminés. On en trouve 20 (en réalité 23) chez l'ensemble des êtres vivants, toujours les mêmes. Les protéines représentent la plus grande partie des molécules du vivant, et ont des rôles fonctionnels très variés, aussi bien mécaniques qu'enzymatiques, de transport.[13]

Comment prédire informatiquement la fonction d'une protéine ?

- Par recherche de signaux peptides
- Par similarité de séquence
- Par similarité de structure (quand vous l'avez)
- Par des méthodes d'apprentissage (*machine learning, neural networks,HMM*)
- Par prédiction de sa structure.[13]

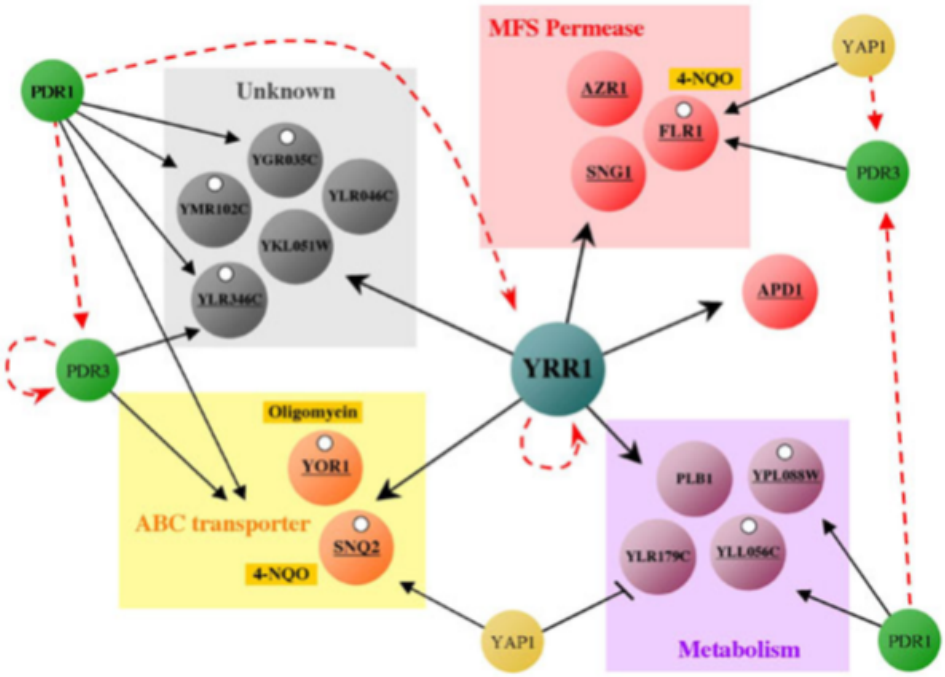
1.17.4.3 Problématiques de réseaux biologiques

Qu'est-ce qu'un réseau biologique ?

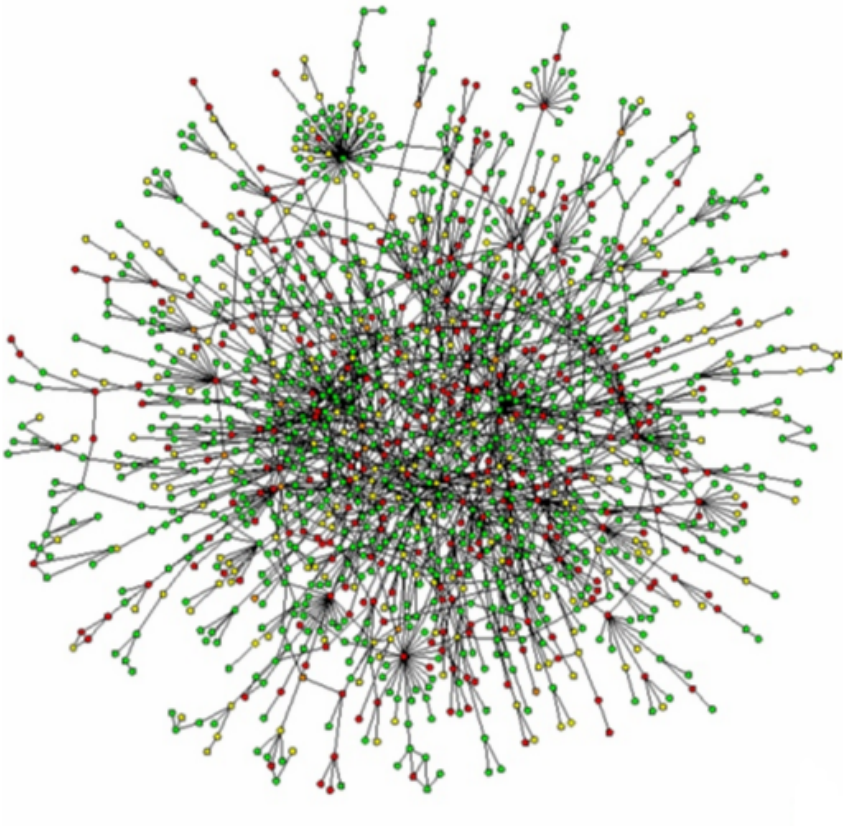
Un réseau biologique est une représentation de la circulation d'un certain type d'information dans la cellule. Il en existe plusieurs types :

- Réseau génétique ou de régulation : le gène A régule l'expression du gène B.
- Réseau d'interaction protéine-protéine : La protéine A interragit physiquement avec la protéine B.
- Réseau de signalisation : la protéine A transmet un signal informatif à la protéine B.
- Réseau métabolique : l'ensemble des réactions chimiques dans une cellule.[13]

Quelques exemples


MFS Permease
YAP1
PDR1
Unknown
AZR1
4-NQO
FLR1
SNG1
PDR3
YGR035C
YMR102C
YLR046C
YKL051W
YLR346C
APD1
PDR3
YRR1
Oligomycin
YOR1
PLB1
YPL088W
ABC transporter
SNQ2
YLR179C
YLL056C
4-NQO
YAP1
Metabolism
PDR1

[13]



[13]

1.18 La régulation génique

1.18.1 Réseau de régulation

Interactions complexes entre les gènes et leurs produits (ARN et protéines) régissant l'activité de la cellule afin de lui permettre de s'adapter en permanence aux variations de son environnement. Il existe de véritables cascades d'interactions, faisant souvent intervenir des boucles de rétroaction positive et négative.[26]

Exemple :

Chez Escherichia coli les gènes *lacZ*, *lacY* et *lacA* (appelés opéron lactose), localisés côte à côte sur le génome, codent pour 3 enzymes impliquées dans le transport et la digestion du lactose. En absence de lactose, une protéine répresseur empêche la synthèse de ces 3 enzymes. En présence de lactose, une molécule dérivée du lactose se fixe sur le répresseur, activant la synthèse des 3 gènes. Le lactose peut alors être utilisé par la cellule.

1.18.2 Pourquoi réguler ses gènes ?

Toutes les cellules contiennent le génome complet de l'organisme, mais :[35]

- Chaque cellule a des besoins spécifiques et différents ;
- Elles doivent s'adapter aux changements du milieu extérieur ;
- Économiser un processus coûteux
  - Inhibition de la synthèse de protéines liées à une activité inutile ;
  - Économie de matière première et d'énergie ;
  - Disponibilité de la machinerie cellulaire pour d'autres gènes.

1.18.3 Inférence de réseaux de régulation

L'expression des gènes est régulée par des protéines appelées facteurs de transcription (TF), qui, en se fixant sur des séquences en amont des gènes cibles (TG), activent ou répriment leur transcription. Les *TFs* résultant eux-mêmes de gènes ayant été transcrits, ces relations entre gènes peuvent être représentées sur un graphe dirigé, où les nœuds et les arêtes représentent respectivement les gènes et les interactions TF-TG. Un tel graphe est appelé *réseau de régulation génique* (GRN). La connaissance et la compréhension des interactions TF-TG ont de nombreuses applications, de la modélisation et la simulation de réseaux *in silico* à l'identification de nouvelles cibles thérapeutiques potentielles. L'inférence des GRNs à partir de données de puces peut être approchée statistiquement comme une série de tâches d'apprentissage supervisé, où l'expression de chaque TG est prédite par l'expression des *TFs* le régulant. Une hypothèse couramment

admise est la parcimonie de ces réseaux : on suppose que seul un petit nombre de *TFs* régule un TG. Il s'agit donc également d'un problème de sélection de variables où le but est d'identifier l'ensemble des *TFs* interagissant avec chaque TG.[24]

1.18.4 Mécanismes de Régulation de l'Expression des Gènes

La cellule a développé des mécanismes qui lui permettent de réprimer ou d'activer l'expression des gènes selon les conditions environnementales de la cellule (stress ou autre). Cette régulation permet, entre autres, à l'organisme d'adapter son métabolisme à son environnement mais surtout, il permet l'expression différentielle du génome selon la spécialisation de la cellule ou la période du développement cellulaire.

Au cours de l'expression des gènes toutes les étapes en partant de la séquence ADN jusqu'au produit final, protéine ou ARN, sont régulées par divers mécanismes. Ainsi la transcription puis la traduction mais aussi les étapes de maturation et les produits eux-mêmes sont soumis à des mécanismes de régulation permettant de moduler, d'accroître ou de décroître, la quantité d'ARN et de protéines synthétisés.

La transcription dépendant de facteurs de transcription leur présence ou absence influe sur le taux de transcription. On en distingue deux principales classes :

— Les éléments cis-régulateurs sont des séquences ADN de 6 à 15 nucléotides de long le plus souvent en amont de la séquence codante à environ 5 000 nucléotides du gène d'intérêt en moyenne.

— Les éléments trans-régulateurs sont des facteurs de transcription se fixant spécifiquement aux régions cis-régulatrices de manière à activer ou inhiber la séquence codante. Elles sont généralement situées sur un autre chromosome.

Le dernier niveau de régulation, la régulation post-traductionnelle, réfère au contrôle de la quantité de protéines actives, par la régulation de l'expression du gène codant cette protéine ou de sa stabilité. La protéine produite peut ainsi être modifiée chimiquement. Ces modifications peuvent jouer sur sa conformation spatiale et donc sur son activité. Enfin des complexes enzymatiques peuvent détruire ces protéines.[29]

Conclusion

A travers ce chapitre, nous avons met en évidence les notions biologiques de base tell que (la Cellule, Gène, Génome, ADN, ARN...) et définition d'expression de l'information génétique. Nous avons vu que le dogme central (Réplication (ADN à lui-même), la transcription(ADN-ARN), la traduction(ARN-Protéine)). Nous avons vu aussi dans ce chapitre le domaine de bio-informatique : leur définition, historique, le but de ce domaine, et aussi nous avons présenté quelques problèmes de bio-informatique (dans ce travail intéressé à la Problématiques de réseaux

biologiques). Nous avons montré l'importance de réseau de régulation(présenté leur définition et le mécanismes de régulation de l'expression des gènes).

Chapitre 2

Data mining

Introduction

Durant ces dernières années, on assiste à une forte augmentation tant dans le nombre que dans le volume des informations mémorisées par des bases de données scientifiques, économiques, financières, administratives, médicales, biologiques, etc. Le stockage en lui-même ne pose pas de réelles difficultés du point de vue informatique, mais le besoin d'interpréter ou de trouver de nouvelles relations entre les éléments stockés dans ces bases a suscité beaucoup d'intérêt. Ainsi, la mise au point de nouvelles techniques informatiques est devenue un thème important pour bon nombre de chercheurs.

La faculté d'apprendre est essentielle à l'être humain pour reconnaître une voix, une personne, un objet.... On distingue en général deux types d'apprentissage : l'apprentissage « par cœur » qui consiste à mémoriser telles quelles des informations, et l'apprentissage par généralisation où l'on apprend à partir d'exemples un modèle qui nous permettra de reconnaître de nouveaux exemples. Pour les systèmes informatiques, il est facile de mémoriser un grand nombre de données (textes, images, vidéos...), mais difficile de généraliser. Par exemple, il leur est difficile de construire un bon modèle d'un objet et d'être ensuite capable de reconnaître efficacement cet objet dans de nouvelles images. L'apprentissage automatique est une tentative de comprendre et de reproduire cette faculté d'apprentissage dans des systèmes artificiels. Il nous semble donc approprié d'utiliser des techniques issues de ce domaine pour découvrir et modéliser des connaissances liant texte et image, et pouvoir ainsi réduire le fossé sémantique. Dans ce chapitre, nous introduirons dans un premier temps le data Mining et ses différentes Techniques. Nous présenterons ensuite l'apprentissage automatique et leur défirent techniques de classification supervisé, non supervisé et semi supervisé qui nous servant utilisée dans notre approche . Puis, nous aborderons les différents méthodes de chaque ces technique tel que svm, k-means ,etc . et nous terminerons par l'étude de performances et nous motionné quelque méthodes tel que la courbe roc, f-mesure,etc.

2.1 Définition de la fouille de données

La *fouille de données* est un domaine qui est apparu avec l'explosion des quantités d'informations stockées, avec le progrès important des vitesses de traitement et des supports de stockage.

La *fouille de données* vise à découvrir, dans les grandes quantités de données, les informations précieuses qui peuvent aider à comprendre les données ou à prédire le comportement des données futures. Le *datamining* utilise depuis sont apparition plusieurs outils de statistiques et d'intelligence artificielle pour atteindre ses objectifs.

La fouille de données s'intègre dans le processus d'extraction des connaissances à partir des données ECD ou *(KDD : Knowledge Discovery from Data en anglais)*. Ce domaine en pleine expansion est souvent appelé le *data mining*.

La fouille de données est souvent définie comme étant le processus de découverte des nouvelles connaissances en examinant de larges quantités de données (stockées dans des entrepôts) en utilisant les technologies de reconnaissance de formes de même que les techniques statistiques et mathématiques. Ces connaissances, qu'on ignore au début, peuvent être des corrélations, des patterns ou des tendances générales de ces données.

La science et l'ingénierie modernes sont basées sur l'idée d'analyser les problèmes pour comprendre leurs principes et leur développer les modèles mathématiques adéquats. Les données expérimentales sont utilisées par la suite pour vérifier la correction du système ou l'estimation de quelques paramètres difficiles à la modélisation mathématiques. Cependant, dans la majorité des cas, les systèmes n'ont pas de principes compris ou qui sont trop complexes pour la modélisation mathématique. Avec le développent des ordinateurs, on a pu rassembler une très grande quantité de données à propos de ces systèmes. La fouille de données vise à exploiter ces données pour extraire des modèles en estimant les relations entre les variables (entrées et sorties) de ses systèmes. En effet, chaque jour nos banques, nos hôpitaux, nos institutions scientifiques, nos magasins, ...produisent et enregistrent des milliards et des milliards de données.

La fouille de données représente tout le processus utilisant les techniques informatiques (y compris les plus récentes) pour extraire les connaissances utiles dans ces données. Actuellement, La fouille de données utilise divers outils manuels et automatiques : on commence par la description des données, résumer leurs attributs statistiques (moyennes, variances, covariance,...), les visualiser en utilisant les courbes, les graphes, les diagrammes, et enfin rechercher les liens significatifs potentiels entre les variables (tel que les valeurs qui se répètent ensemble). Mais la description des données toute seule ne fournit pas un plan d'action. On doit bâtir un modèle de prédiction basé sur les informations découvertes, puis tester ce modèle sur des données autres que celles originales.

La fouille de données a aujourd'hui une grande importance économique du fait qu'elle permet

d'optimiser la gestion des ressources (humaines et matérielles). Elle est utilisée par exemple dans [18] :

- organisme de crédit : pour décider d'accorder ou non un crédit en fonction du profil du demandeur de crédit, de sa demande, et des expériences passées de prêts ;
- optimisation du nombre de places dans les avions, hôtels, ...) sur-réservation
- organisation des rayonnages dans les supermarchés en regroupant les produits qui sont généralement achetés ensemble (pour que les clients n'oublient pas bêtement acheter un produit parce qu'il est situé à l'autre bout du magasin). Par exemple, on extraira une règle du genre : "les clients qui achètent le produit X en fin de semaine, pendant l'été, achètent généralement également le produit Y" ;
- organisation de campagne de publicité, promotions,...(ciblage des offres)
- diagnostic médical : "les patients ayant tels et tels symptômes et demeurant dans des agglomérations de plus de 104 habitants développent couramment telle pathologie" ;
- analyse du génome ;
- classification d'objets (astronomie,...) ;
- commerce électronique ;
- analyser les pratiques et stratégies commerciales et leurs impacts sur les ventes ;
- moteur de recherche sur internet : fouille du web ;
- extraction d'information depuis des textes : fouille de textes ;
- évolution dans le temps de données : fouille de séquences.

2.2 Processus du data mining

Il est très important de comprendre que le data mining n'est pas seulement le problème de découverte de modèles dans un ensemble de données. Ce n'est qu'une seule étape dans tout un processus suivi par les scientifiques, les ingénieurs ou toute autre personne qui cherche à extraire les connaissances à partir des données. En 1996 un groupe d'analystes définit le data mining comme étant un processus composé de cinq étapes sous le standard CRISP-DM *(Cross-Industry Standard Process for Data Mining)* comme schématisé ci dessous : Ce processus, composé de cinq étapes, n'est pas linéaire, on peut avoir besoin de revenir à des étapes précédentes pour corriger ou ajouter des données. Par exemple, on peut découvrir à l'étape d'exploration (5) de nouvelles données qui nécessitent d'être ajoutées aux données initiales à l'étape de collection (2).[18]

Décrivons maintenant ces étapes :

1. **Définition et compréhension du problème** :
   Dans la plus part des cas, il est indispensable de comprendre la signification des données et le domaine à explorer. Sans cette compréhension, aucun algorithme ne va donner un résultat fiable.

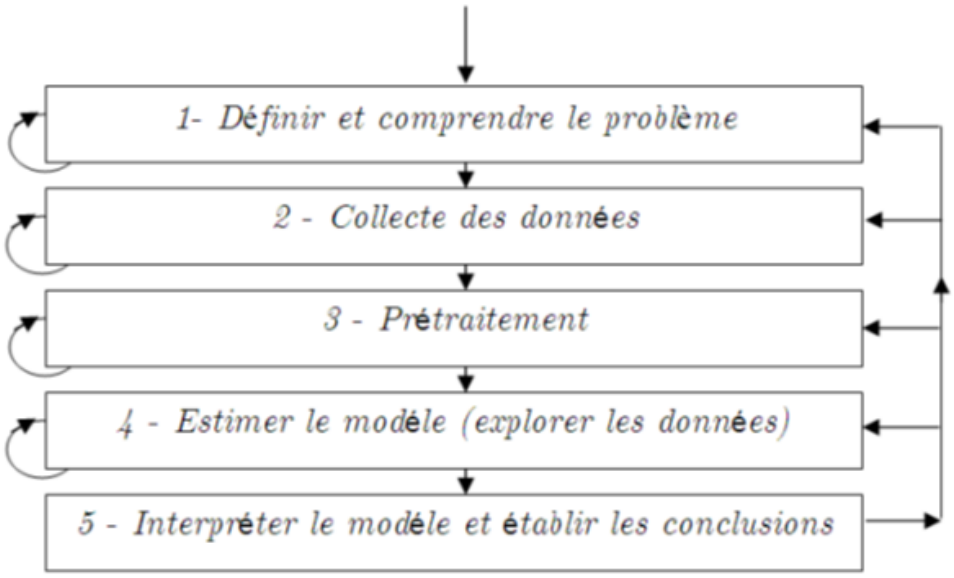


Figure 2.1 – Processus de data mining *(CRISP-DM)* [18]

En effet, Avec la compréhension du problème, on peut préparer les données nécessaires à l'exploration et interpréter correctement les résultats obtenus. Généralement, le data mining est effectué dans un domaine particulier (banques, médecine, biologie, marketing, ...etc) où la connaissance et l'expérience dans ce domaine jouent un rôle très important dans la définition du problème, l'orientation de l'exploration et l'explication des résultats obtenus. Une bonne compréhension du problème comporte une mesure des résultats de l'exploration, et éventuellement une justification de son coût. C'est-à-dire, pouvoir évaluer les résultats obtenus et convaincre l'utilisateur de leur rentabilité.

2. **Collecte des données :**
   dans cette étape, on s'intéresse à la manière dont les données sont générées et collectées. D'après la définition du problème et des objectifs du data mining, on peut avoir une idée sur les données qui doivent être utilisées. Ces données n'ont pas toujours le même format et la même structure. On peut avoir des textes, des bases de données, des pages web, ...etc. Parfois, on est amené à prendre une copie d'un système d'information en cours d'exécution, puis ramasser les données de sources éventuellement hétérogènes (fichiers, bases de données relationnelles, temporelles,...).

   Quelques traitements ne nécessitent qu'une partie des données, on doit alors sélectionner les données adéquates. Généralement les données sont subdivisées en deux parties : une utilisée pour construire un modèle et l'autre pour le tester. On prend par exemple une partie importante (suffisante pour l'analyse) des données (80 %) à partir de laquelle on

construit un modèle qui prédit les données futures. Pour valider ce modèle, on le teste sur la partie restante (20 %) dont on connaît le comportement.

3. **Prétraitement :**

   Les données collectées doivent être "préparées" .

   Avant tout, elles doivent être nettoyées puisqu'elles peuvent contenir plusieurs types d'anomalies : des données peuvent être omises à cause des erreurs de frappe ou à causes des erreurs dues au système lui-même, dans ce cas il faut remplacer ces données ou éliminer complètement leurs enregistrements.

   Des données peuvent être incohérentes c-à-d qui sortent des intervalles permis, on doit les écarter où les normaliser. Parfois on est obligé à faire des transformations sur les données pour unifier leur poids. Un exemple de ces transformations est la normalisation des données qui consiste à la projection des données dans un intervalle bien précis [0,1] ou [0,100] par exemple.

   Un autre exemple est le lissage des données qui considère les échantillons très proches comme étant le même échantillon. Le pré-traitement comporte aussi la réduction des données qui permet de réduire le nombre d'attributs pour accélérer les calculs et représenter les données sous un format optimal pour l'exploration.

   Une autre méthode de réduction est celle de la sélection et suppression des attributs dont l'importance dans la caractérisation des données est faible, en mesurant leurs variances. On peut même réduire le nombre de données utilisées par le data mining en écartant les moins importantes. Dans la majorité des cas, le pré-traitement doit préparer des informations globales sur les données pour les étapes qui suivent tel que la tendance centrale des données (moyenne, médiane, mode), le maximum et le minimum, le rang, les quartiles, la variance, ... etc. Plusieurs techniques de visualisation des données telles que les courbes, les diagrammes, les graphes,...etc, peuvent aider à la sélection et le nettoyage des données. Une fois les données collectées, nettoyées et pré-traitées on les appelle entrepôt de données *(data warehouse).*

4. **Estimation du modèle :**

   Dans cette étape, on doit choisir la bonne technique pour extraire les connaissances (exploration) des données.

   Des techniques telles que les réseaux de neurones, les arbres de décision, les réseaux bayésiens, le clustering, ...sont utilisées.

   Généralement, l'implémentation se base sur plusieurs de ces techniques, puis on choisit le bon résultat. Dans le reste de ce rapport on va détailler les différentes techniques utilisées dans l'exploration des données et l'estimation du modèle.

5. **Interprétation du modèle et établissement des conclusions :**

   généralement, l'objectif du data mining est d'aider à la prise de décision en fournissant

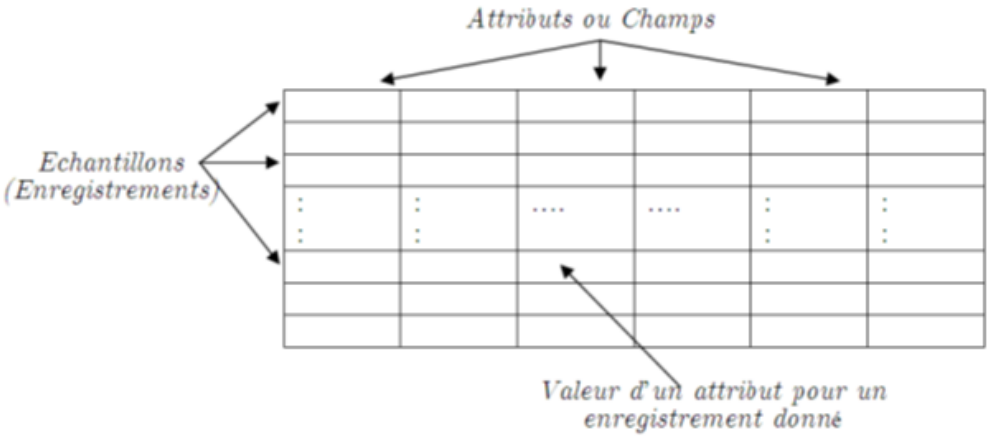
des modèles compréhensibles aux utilisateurs. En effet, les utilisateurs ne demandent pas des pages et des pages de chiffres, mais des interprétations des modèles obtenus.

Les expériences montrent que les modèles simples sont plus compréhensibles mais moins précis, alors que ceux complexes sont plus précis mais difficiles à interpréter.[18]

2.3 Quel type de données fouiller ?

Le composant de base d'un processus de data mining est l'ensemble d'échantillons représentants les données à explorer. Chaque échantillon est présenté sous forme de ligne caractérisée par un ensemble d'attributs.

Dans le cas des bases de données un échantillon est un enregistrement composé d'un ensemble de champs. Généralement, il convient de représenter les enregistrements sous forme de points dans un espace de m dimensions où m est le nombre d'attributs.[18]



[18]

Une donnée est, donc, un enregistrement au sens des bases de données, que l'on nomme aussi "individu" (terminologie issue des statistiques) ou "instance" (terminologie orientée objet en informatique) ou même "tuple" (terminologie base de données) et "point" ou "vecteur" parce que finalement, d'un point de vue abstrait, une donnée est un point dans un espace euclidien ou un vecteur dans un espace vectoriel.

Une données est caractérisée par un ensemble de "champs", de "caractères", ou encore d' "attributs" (en suivant les 3 terminologies précédemment évoquées : bases de données, statistiques et conception orientée objet).

Les attributs ou les champs sont de deux types : numériques où catégoriels. Les attributs numériques qui comportent les variables réelles ou entières tel que la longueur, le poids, l'âge, ... sont caractérisés par une relation d'ordre ($5 < 7$:5) et une mesure de distance (D(5 ; 7 :5)

= 2 :5).Les attributs catégoriels (appelées aussi symboliques) tel que la couleur, l'adresse ou le groupe sanguin ne possèdent aucune de ces caractéristiques. Deux variables catégorielles ne peuvent être qu'égales ou différentes. Il est clair que la relation d'ordre dans le cas des attributs numériques permet de calculer dans un ensemble d'enregistrements, un max, un min, une moyenne, une distance, ...etc. Alors que dans le cas d'attributs catégoriels ça sera impossible.

comment calculer la moyenne, la variance ou la distance entre des adresses ? Dans ce cas, de nouvelles mesures doivent être développées pour chaque technique de fouille de données. Théoriquement, plus le nombre d'échantillons est important, meilleure est la précision de l'analyse. Mais en pratique, beaucoup de difficultés peuvent être rencontrées avec les bases de données gigantesques (des milliards d'enregistrements ou des Gigabytes).

En effet, Les bases de données de nos jours sont immenses au point où elles épuisent même les supports de stockage, et nécessitent pour être analysées les machines les plus puissantes et les techniques les plus performantes.

Un premier problème avec les bases de données immenses, est celui de leur préparation à l'analyse, puisque la qualité des données analysées influence directement sur les résultats d'analyse. La préparation doit prendre compte d'un certain nombre de points :

- Les données doivent être précises : les noms doivent être écrits correctement, les valeurs doivent être dans les bons intervalles et doivent être complètes,
- Les données doivent être enregistrées dans les bon formats : une valeur numérique ne doit pas être enregistrée sous format caractère, une valeur entière ne doit pas être réelle,...etc,
- La redondance doit être éliminée ou au moins minimisée,
- ...etc.

Dans le cas d'un nombre limité d'échantillons, la préparation peut être semi-automatique ou même manuelle, mais dans notre cas d'immenses BDD la préparation automatique s'impose et des techniques automatiques de vérification et normalisation doivent intervenir. Le problème majeur des BDD immenses apparaît lors de leur exploration, le temps d'exploration et la précision doivent être pris en compte. Toutes les techniques d'exploration qu'on va voir fixent des critères d'arrêt soit sur le temps d'exécution ou sur la précision des résultats atteints.

Les enregistrements sont regroupés dans des tables et dans des bases de données de différents types et l'analyse effectuée et le choix de ses outils dépendent fortement du type de la base de données à analyser. En fait, les bases de données relationnelles, les bases de données transactionnelles, les systèmes avancés de bases de données, les streams et les bases de données spatiales représentent les types les plus utilisés.[18]

2.4 Les tâches de la fouille de données

Beaucoup des problèmes intellectuels, économiques ou même commerciaux peuvent être exprimés en termes des six tâches suivantes [18] :

- La classification.
- L'estimation.
- Le groupement par similitude (règles d'association).
- L'analyse des clusters.
- La description.

particulière prise comme but en termes de ces données. Le groupement par similitude et l'analyse des clusters sont des tâches non-supervisées où le but est d'établir un certain rapport entre toutes la description appartient à ces deux catégories de tâche, elle est vue comme une tâche supervisée et non-supervisée en même temps.

- **Classification :**

  La classification est la tâche la plus commune de la fouille de données qui semble être une tâche humaine primordiale. Afin de comprendre notre vie quotidienne, nous sommes constamment obligés à classer, catégoriser et évaluer.

  La classification consiste à étudier les caractéristiques d'un nouvel objet pour l'attribuer à une classe prédéfinie. Les objets à classifier sont généralement des enregistrements d'une base de données, la classification consiste à mettre à jours chaque enregistrement en déterminant la valeur d'un champ de classe.

  Le fonctionnement de la classification se décompose en deux phases :

  * La première étant la phase d'apprentissage. Dans cette phase, les approches de classification utilisent un jeu d'apprentissage dans lequel tous les objets sont déjà associés aux classes de références connues. L'algorithme de classification apprend du jeu d'apprentissage et construit un modèle.
  * La seconde phase est la phase de classification proprement dite, dans laquelle le modèle appris est employé pour classifier de nouveaux objets.

- **L'estimation :**

  L'estimation est similaire à la classification à part que la variable de sortie est numérique plutôt que catégorique. En fonction des autres champs de l'enregistrement l'estimation consiste à compléter une valeur manquante dans un champ particulier.

  Par exemple on cherche à estimer la lecture de tension systolique d'un patient dans un hôpital, en se basant sur l'âge du patient, son genre, son indice de masse corporelle et le niveau de sodium dans son sang.

  La relation entre la tension systolique et les autres données vont fournir un modèle d'estimation. Et par la suite nous pouvons appliquer ce modèle dans d'autres cas.

— **Le groupement par similitude :** (Analyse des associations et de motifs séquentiels)
Le groupement par similitude consiste à déterminer quels attributs "vont ensemble". La tâche la plus répandue dans le monde du business, est celle appelée l'analyse d'affinité ou l'analyse du panier du marché, elle permet de rechercher des associations pour mesurer la relation entre deux ou plusieurs attributs. Les règles d'associations sont, généralement, de la forme "Si <antécédent>, alors <conséquent>".
Les trois premières tâches sont des exemples de la fouille supervisée de données dont le but est d'utiliser les données disponibles pour créer un modèle décrivant une variable.

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

— **La description :**
Parfois le but de la fouille est simplement de décrire se qui se passe sur une base de données compliquée en expliquant les relations existantes dans les données pour premier lieu comprendre le mieux possible les individus, les produit et les processus présents dans cette base.
Une bonne description d'un comportement implique souvent une bonne explication de celui-ci. Dans la société Algériennes nous pouvons prendre comme exemple comment une simple description, "les femmes supportent le changement plus que les hommes", peut provoquer beaucoup d'intérêt et promouvoir les études de la part des journalistes, sociologues, économistes et les spécialistes en politiques. [18]

2.5 Apprentissage automatique :

2.5.1 Définitions d'apprentissage automatique

L'apprentissage automatique fait référence au développement, à l'analyse et à l'implémentation de méthodes qui permettent à une machine d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.

Des systèmes complexes peuvent être analysés, y compris pour des données associées à des

valeurs symboliques (ex : sur un attribut numérique, non pas simplement une valeur numérique, juste un nombre, mais une valeur probabilisée, c'est-à-dire un nombre assorti d'une probabilité ou associé à un intervalle de confiance) ou un ensemble de modalités possibles sur un attribut numérique ou catégoriel. L'analyse peut même concerner des données présentées sous forme de graphes ou d'arbres, ou encore de courbes (par exemple, la courbe d'évolution temporelle d'une mesure ; on parle alors de données continues, par opposition aux données discrètes associées à des attributs-valeurs classiques).

En tous les cas, il consiste à utiliser des ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations, que ce soit des données-exemples ou des expériences passées .Lorsque l'on connait le bon modèle de traitement à utiliser, alors pas besoin de faire de l'apprentissage.[40]

L'apprentissage automatique peut être utile lorsque :

- On n'a pas d'expertise sur le problème. Premier exemple : "robot navigant sur Mars".
- On a une expertise, mais on ne sait pas comment l'expliquer. Premier exemple : "reconnaissance de visages".
- Les solutions au problème changent dans le temps. Premier exemple : "routage de Paquets".
- Les solutions doivent être personnalisées. Premier exemple : "biométrie".
- Il y'a deux phases d'apprentissage
  1. On présente des exemples au système.
  2. Le system « apprend » à partir des exemples.

Donc le système modifie graduellement ses paramètres ajustables pour que sa sortie ressemble à la sortie désirée.

L'apprentissage automatique (*machine learning* en anglais), un des champs d'étude de l'intelligence artificielle, est la discipline scientifique concernée par le développement, l'analyse et l'implémentation de méthodes automatisables qui permettent à une machine (au sens large) d'évoluer grâce à un processus d'apprentissage, et ainsi de remplir des tâches qu'il est difficile ou impossible de remplir par des moyens algorithmiques plus classiques.[40]

2.5.2 Applications

L'apprentissage automatique est utilisé pour doter des ordinateurs ou des machines de systèmes de : perception de leur environnement : vision, reconnaissance d'objets (visages, schémas, langages naturels, écriture, formes syntaxiques, etc.) ; moteurs de recherche ; aide aux diagnostics, médical notamment, bio-informatique, schéma informatique ; interfaces cerveau-machine ; détection de fraudes à la carte de crédit, analyse financière, dont analyse du marché boursier ; classification des séquences d'ADN ; jeu ; génie logiciel ; sites Web adaptatifs ou mieux adaptés ;

locomotion de robots ; etc.[40]

Exemples :

* Un système d'apprentissage automatique peut permettre à un robot ayant la capacité de bouger ses membres mais ne sachant initialement rien de la coordination des mouvements permettant la marche, d'apprendre à marcher. Le robot commencera par effectuer des mouvements aléatoires, puis, en sélectionnant et privilégiant les mouvements lui permettant d'avancer, mettra peu à peu en place une marche de plus en plus efficace.
* La reconnaissance de caractères manuscrits est une tâche complexe car deux caractères similaires ne sont jamais exactement égaux. On peut concevoir un système d'apprentissage automatique qui apprend à reconnaître des caractères en observant des « exemples », c'est-à-dire des caractères connus.[40]

2.5.3 Types d'apprentissage

Les algorithmes d'apprentissage peuvent se catégoriser selon le type d'apprentissage qu'ils emploient [40] :

- **L'apprentissage supervisé**
- **L'apprentissage non-supervisé**
- **L'apprentissage par renforcement (semi –supervisé)**

Il est Illustré selon le schéma suivant :

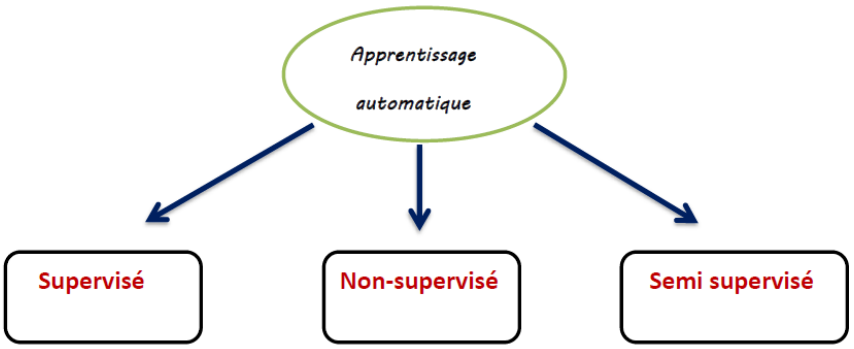


Figure 2.2 – Différents types d'apprentissage [40]

2.5.4 L'apprentissage supervisé

2.5.4.1 Introduction

La notion de prédiction fait référence à une stratégie particulière d'Apprentissage Automatique (AA), appelée apprentissage supervisé. Ce domaine d'étude à part entière, peut être considéré comme une sous-thématique de l'Intelligence Artificielle (IA). De façon synthétique, l'apprentissage supervisé consiste à faire émerger d'un ensemble de données d'entrainement préclassifiées, les caractéristiques nécessaires et suffisantes pour permettre de classer correctement une nouvelle donnée. Dans ce type d'approche, les classes sont connues à l'avance, cette connaissance est utilisée dans le processus d'apprentissage.[40]

2.5.4.2 Définition

Si les classes sont prédéterminées et les exemples connus, le système apprend à classer selon un modèle de classement ; on parle alors d'apprentissage supervisé (ou d'analyse discriminante).

Un expert doit préalablement étiqueter des exemples. Le processus se passe en deux phases. Lors de la première phase (hors ligne, dite d'apprentissage), il s'agit de déterminer un modèle des données étiquetées.

La seconde phase (en ligne, dite de test) consiste à prédire l'étiquette d'une nouvelle donnée, connaissant le modèle préalablement appris.

Parfois il est préférable d'associer une donnée non pas à une classe unique, mais une probabilité d'appartenance à chacune des classes prédéterminées (on parle alors d'apprentissage supervisé probabiliste).[40]

Exemple :

L'analyse discriminante linéaire ou les SVM en sont des exemples typiques.

Autre exemple :

en fonction de points communs détectés avec les symptômes d'autres patients connus (les « exemples »), le système peut catégoriser de nouveaux patients au vu de leurs analyses médicales en risque estimé (probabilité) de développer telle ou telle maladie.

2.5.4.3 Principe

1. Un expert est employé pour étiqueter correctement des exemples.
2. L'apprenant doit alors trouver ou approximer la fonction qui permet d'affecter la bonne étiquette à ces exemples.[40]

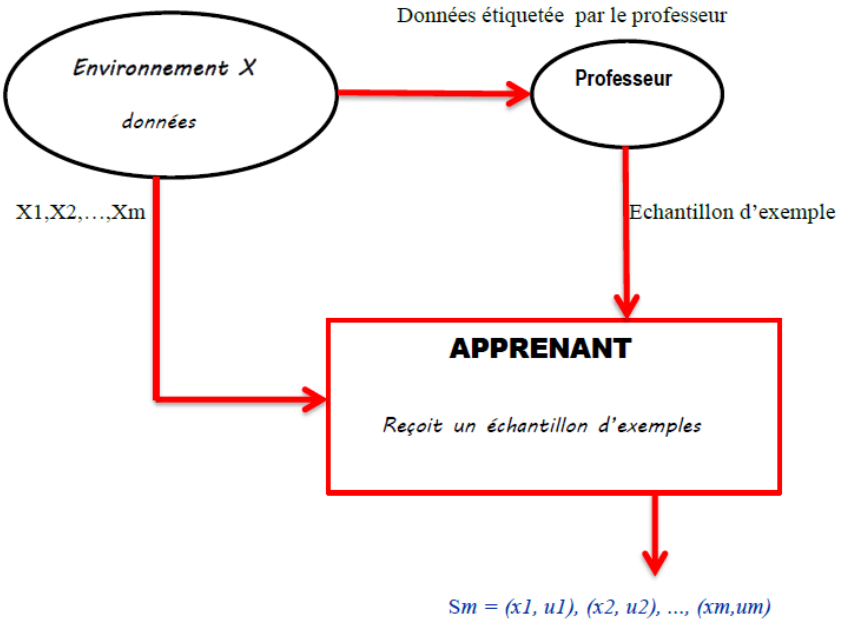


FIGURE 2.3 – Système de fonctionnement de l'apprentissage supervisé [40]

2.5.4.4 Les Différents buts d'apprentissage supervisé

Buts principaux

- Approximer au mieux la sortie désirée pour chaque entrée observée.
- Obtenir un « bon » modèle : la prévision obtenue est proche de la vraie valeur.
- Obtenir rapidement un modèle rapide : temps de construction du modèle et temps nécessaire à l'obtention d'une prévision.
- Pouvoir garantir les performances : avec une probabilité de 1-r ; la prévision sera bonne à peu près.[40]

Buts annexes :

- Obtenir un modèle compréhensible : comment le modèle prend il la décision ?
- Obtenir un modèle modifiable : pouvoir prendre en compte de nouvelles données ; s'adapter à un environnement changeant, etc.[40]

Buts mathématiques :

- Apprendre une projection entre des observations X en entrée et des valeurs associées Y en sortie.[40]

2.5.4.5 Quelques algorithmes d'apprentissage supervisé

La plupart des algorithmes d'apprentissage supervisés tentent de trouver un modèle (une fonction mathématique) qui explique le lien entre des données d'entrée et les classes de sortie.

Ces jeux d'exemples sont donc utilisés par l'algorithme.[40]

Il existe de nombreuses méthodes d'apprentissage supervisé parmi eux :

- Méthode des k plus proches voisins.
- Machine à vecteurs de support.
- Réseau de neurones.
- Arbre de décision.
- Classification naïve bayésienne.

k plus proches voisins

La méthode des k-plus proches voisins *(noté K-PPV ou K-NN pour K-Nearst-Neighbors en anglais)* consiste à déterminer pour chaque nouvel individu que l'on veut classer, la liste des k-plus proches voisins parmi les individus déjà classé. L'individu est affecté à la classe qui contient le plus d'individus parmi ces k-plus proches voisins. Cette méthode nécessite de choisir une distances (la plus classique est la distance Euclidienne), et donc le nombre$k$ de voisins à prendre en compte. Cette méthode supervisée et non-paramétrique est souvent performante. De plus son apprentissage est assez simple.[6]

Machine à vecteurs de support

Cette technique - initiée par [42] - tente de séparer linéairement les exemples positifs des exemples négatifs dans l'ensemble des exemples. Chaque exemple doit être représenté par un vecteur de dimension n. La méthode cherche alors l'hyperplan qui sépare les exemples positifs des exemples négatifs, en garantissant que la marge entre le plus proche des positifs et des négatifs soit maximale. Intuitivement, cela garantit un bon niveau de généralisation car de nouveaux exemples pourront ne pas être trop similaires à ceux utilisés pour trouver l'hyperplan mais être tout de même situés franchement d'un côté ou l'autre de la frontière. L'efficacité des *SVM* est supérieure à celle de toutes les autres méthodes sur la classification de textes. Son efficacité est aussi très bonne pour la reconnaissance de formes. Un autre intérêt est la sélection de Vecteurs Supports qui représentent les vecteurs discriminant grâce auxquels est déterminé l'hyperplan. Les exemples utilisés lors de la recherche de l'hyperplan ne sont alors plus utiles et seuls ces vecteurs supports sont utilisés pour classer un nouveau cas. Cela en fait une méthode très rapide.[6]

Réseau de neurones

Les réseaux de neurones sont des approximateurs universels parcimonieux ; ils peuvent donc être utilisés pour modéliser ou commander tout processus, statique ou dynamique, non linéaire : en raison de leur parcimonie, ils sont avantageux par rapport aux autres approximateurs et notamment au flou - dès que le processus à modéliser ou à commander possède plus de deux

ou trois entrées. Néanmoins, comme toute autre technique, les réseaux de neurones sont soumis à des contraintes : étant des outils statistiques, ils traitent uniquement de données numériques, dont le nombre et la représentativité doivent être convenables même si, leur parcimonie leur permet d'utiliser moins de données que d'autres méthodes statistiques. S'il est possible de tirer profit, pour la conception du réseau, des connaissances, même imprécises,que l'on peut avoir sur le processus, il faut qu'elles soient sous forme mathématique : les réseaux de neurones ne permettent pas de traiter aisément des données linguistiques.[6]

Arbre de décision

Les arbres de décision sont les plus populaires des méthodes d'apprentissage. L'apprentissage se fait par partitionnement récursif selon des règles sur les variables explicatives suivant les critères de partitionnement et les données, on dispose de différentes méthodes, dont CART, CHAID.... Ces méthodes peuvent s'appliquer à une variable. Deux types d'arbres de décisions sont ainsi définis [6] :

- **Arbre de classification** : la variable expliquée est de type nominal. A chaque étape du partitionnement, on cherche à réduire l'impureté totale des deux nœuds fils par rapport au nœud père.
- **Arbre de régression** : la variable expliquée est de type numérique et il s'agit de prédire une valeur la plus proche possible de la vraie valeur.

Construire un tel arbre consiste à définir un nœud, chaque nœud permettant de faire une partition des objets en 2 groupes sur la base d'une des variables explicatives. Il convient donc :
→ définir un critère permettant de sélectionner le meilleur nœud possible à une étape donnée ;
→ définir quand s'arrête le découpage, en définissant un nœud terminal (feuille) ;
→ D'attribuer au nœud terminal la classe ou la valeur la plus probable
→ D'élaguer l'arbre quand le nombre de nœuds devient trop important en sélectionnant un sous arbre optimal à partir de l'arbre maximal ;
→ Valider l'arbre à partir d'une validation croisée ou d'autres techniques.

RANDOM FOREST

RANDOM FOREST est une technique d'apprentissage supervisée qui combine une technique d'agrégation, le BAGGING, et une technique particulière d'induction d'arbres de décision. Les forêts aléatoires ont été inventées par *Breiman* en 2001. Elles sont en général plus efficaces que les simples arbres de décision mais possède l'inconvénient d'être plus difficilement interprétables. Leur construction se base sur le bootstrap (ou le bagging). On subdivise l'ensemble de données en plusieurs parties par le bootstrap puis on apprend un arbre de décision à partir de chaque partie. Un nouvel exemple est testé par tous les arbres construits et sa classe est la classe

majoritaire.[11]

2.5.5 Apprentissage non supervisé (Clustering)

2.5.5.1 Principe

L'apprentissage non supervisé consiste à apprendre à classer sans supervision. Au début de processus nous ne disposons ni de la définition des classes, ni de leurs nombres. C'est l'algorithme de classification qui va déterminer ces informations. Nous ne disposons pas non plus de données en entrée qui sont déjà classées, c'est aussi à l'algorithme de découvrir par lui-même la structure plus ou moins cachée des données et de former des groupes d'individus dont les caractéristiques sont communes. L'apprentissage non supervisé est utilisé dans plusieurs domaines tels que [40] :

- Médecine : Découverte de classes de patients présentant des caractéristiques physiologiques communes.
- Le traitement de la parole : construction de système de reconnaissance de la voie humaine.
- Archéologie : regroupement des objets selon leurs époques.
- Traitement d'images.
- Classification de documents.

2.5.5.2 Quelques méthodes de la classification non supervisé

Il y a plusieurs et on distingue 2 sortes de méthodes principales généralement connus [40] :

* Méthode des K-means

* Classification hiérarchique

K-means

C'est une méthode dont le but est de diviser des observations en k partitions dans lesquelles chaque observation appartient à la partition avec la moyenne la plus proche. Nous citons deux méthodes connues sur le principe de k-means sont [40] :

→ Méthodes de centres mobiles ;

→ Méthodes des nuées dynamiques.

- **Méthode de entres mobiles** Cette méthode consiste à construire une partition en k classes en sélectionnant k individus commence, des classes tirés au hasard de l'ensemble d'individus. Après cette sélection, on affecte chaque individu au centre le plus proche en créant k classes, les centres des classes seront remplacer par les centres de gravité et nouveaux classes seront crées par le même principe. Généralement la partition obtenue est localement optimale car elle dépend du choix initial des centres. Pour cela les résultats entre deux exécutions de l'algorithme sont significativement variés.

- **Méthode de nuées dynamiques** Dans ce cas, le problème posé est la recherche d'une partition en k (k fixé) classes d'une ensemble de n individus. C'est un algorithme itératif. Soit I une population d'individus, cette population est représentable sur R et forme un nuage de n points. On cherche à constituer une partition en k classes sur i. chaque classe est représentée par son centre, également appelé noyau, constitué du petit sous-ensemble de la classe qui minimise le critère de dissemblance.

Classification hiérarchique

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

— Toute classe est non vide.
— Tout individu appartient à une (et même plusieurs) classes.
— Deux classes distinctes sont disjointes, ou vérifient une relation d'inclusion (l'une d'elle est incluse dans l'autre)
— Toute classe est la réunion des classes qui sont incluse dans elle.

L'avantage de cette méthode est qu'elle n'est soumise à aucune initialisation particulière de paramètre(s) ce qui la rend déterministe, et en outre, que le nombre de classe n'a pas à être fixé a priori. Cependant, ce type de méthode impose le calcul de la matrice des distances de tous les points d'observation avec tous les autres, et cette masse de calculs est beaucoup trop importante compte tenu du temps que nous voulons consacrer à cette étape. Parmi les méthodes non-supervisées les plus utilisées, citons deux types d'approches :[40]

Classification hiérarchique ascendante

La *CAH* permet de construire une hiérarchie entière des objets sous la forme d'un "arbre" dans un ordre ascendant. On commence en considérant chaque individu comme une classe et on essaye de fusionner deux ou plusieurs classes appropriées (selon une similarité) pour former une nouvelle classe. Le processus est itéré jusqu'à ce que tous les individus se trouvent dans une même classe. Cette classification génère un arbre que l'on peut couper à différents niveaux pour obtenir un nombre des classes plus ou moins grand. Différentes mesures de la distance interclasses peuvent être utilisées : la distance euclidienne, la distance inferieur (qui favorise la création de classes de faible inertie) ou la distance supérieure (qui favorise la création de classes d'inertie plus importante) etc. le cas de la classification ascendante hiérarchique, à partir des éléments, on forme des petites classes ne comprenant que des individus très semblables, puis à partir de celle-ci, on construit des classes de moins en moins homogènes, jusqu'à obtenir la classe tout entière.[40]

Classification hiérarchique descendante

Dans la *CDH*, on considérant tous les individus comme une seule classe au début, on divise successivement les classes en classes plus raffinées. Le processus marche jusqu'à ce que chaque classe contienne un seul point ou bien si l'on atteint un nombre de classes désiré.[40]

2.5.6 Apprentissage Semi-Supervisé

2.5.6.1 Introduction

Effectué de manière probabiliste ou non, généralement si on a des données et on est toujours dans le terme d'apprentissage alors il vise à faire apparaître la distribution sous-jacente des « exemples » dans leur espace de description. Il est mis en œuvre quand des données (ou « étiquettes ») manquent ....

Le modèle doit utiliser des exemples non-étiquetés pouvant néanmoins renseigner.[40]

2.5.6.2 Définition

C'est une classe de techniques d'apprentissage automatique qui utilise un ensemble de données étiquetées et non-étiquetés. 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. Un autre intérêt provient du fait que l'étiquetage de données nécessite l'intervention d'un utilisateur humain. Lorsque les jeux de données deviennent très grands, cette opération peut s'avérer fastidieuse. Dans ce cas, l'apprentissage semi-supervisé, qui ne nécessite que quelques étiquettes, revêt un intérêt pratique évident.

Un exemple d'apprentissage semi-supervisé est le Co-apprentissage, dans lequel deux classificateurs apprennent un ensemble de données, mais en utilisant chacun un ensemble de caractéristiques différentes, idéalement indépendantes. Si les données sont des individus à classer en hommes et femmes, l'un pourra utiliser la taille et l'autre la pilosité par exemple.

En effet, étiqueter un échantillon (données, textes, pages WEB,...) est une opération coûteuse car elle nécessite un « expert ».[40]

2.5.6.3 Quelques algorithmes d'apprentissage semi-supervisé

- EM Semi-Supervisé.
- Co-training.
- Traductive SVM's.
- Algorithmes à base de graphes.

Les SVM transductifs

Le principe des *SVM transductifs* reste toujours de trouver la séparation avec la plus vaste marge possible. Cependant, le principe d'apprentissage transductif implique que la règle de classement est uniquement apprise pour classer les données non étiquetées. *Les SVM tranductifs*, recherchent la frontière de classement avec la plus vaste marge possible une fois les données non étiquetées classées. Ainsi l'optimisation se fait à la fois sur la taille de la marge et sur les étiquettes des données non classées. D'un point de vue théorique ceci consiste à minimiser une borne sur l'erreur de l'échantillon test. Contrairement aux SVM standards, le problème d'optimisation est non convexe et pose par conséquent des problèmes combinatoires. Il ne peut pas être résolu exactement quand le nombre de données non classées excède 100. Des approches heuristiques permettent de faire face à ce problème d'intractabilité. C'est par exemple le cas des SVM_{light}, qui nécessitent en pratique de fixer la proportion d'exemples étiquetés positivement et négativement afin d'éviter l'obtention de solutions dégénérées. Une autre possibilité qui permet de traiter le problème des *SVM transductifs* est d'utiliser des outils de programmation semi-définie positive. Celle-ci consiste à relaxer les contraintes imposées. Cette méthode permet de traiter des situations allant jusqu'à 1000 données non étiquetées. Les *SVM transductifs*, comme les SVM classiques sont particulièrement bien adaptés pour résoudre des problèmes de classification de données en grande dimension. Les *SVM transductifs* ont notamment montré de bonnes performances en classification de texte.[41]

L'auto-apprentissage

La première approche à effectuer de l'apprentissage semi-supervisé est *l'auto-apprentissage*. Elle consiste à apprendre une règle de classement à partir de l'une des méthodes décrites précédemment uniquement sur les données classées. Ensuite une fraction des données non classées est classée à partir de la règle apprise. La règle est ensuite réapprise à partir des données classées à la base et des données classées à l'étape précédente qui sont maintenant considérées comme classées. Ces étapes sont itérées jusqu'à ce que toutes les données non classées soient classées [52, 26, 1]. L'intérêt pratique de cette méthode est qu'elle permet d'adapter n'importe qu'elle méthode de classification supervisée au cadre semi-supervisée. Cependant, le comportement de l'auto-apprentissage ainsi que ses conséquences dépendent fortement de la méthode d'apprentissage supervisée utilisée. De plus l'absence de résultat théorique sur ce à quoi correspond l'auto-apprentissage rend difficile la compréhension de ce qui est effectivement fait et des améliorations qui peuvent en être attendues.[41]

2.6 Évaluation des performances d'un classifieur

L'existence réelle de grands biais, dans la distribution et la répartition des classes a été observée dans différents domaines par[16],[20], [28] et [38]. Par exemple, dans la prise de décision médicale, les épidémies peuvent faire augmenter l'incidence d'une maladie avec le temps. Dans la détection de fraudes, la proportion des fraudes varie de manière significative de mois en mois et de zone en zone, [21]. Dans chacun des exemples à deux classes, la prédominance d'une classe peut changer radicalement sans pour autant altérer fondamentalement les caractéristiques de la classe. Si les proportions d'éléments positifs ou/et négatifs changent dans une base de test, nous voulons que le système d'évaluation des performances ne soit pas perturbé. Pour cela, nous allons comparer deux méthodes applicables aux classifieurs à deux classes, la courbe Précision-Rappel et la courbe ROC. Les mesures que nous allons évoquer utilisent la matrice de confusion, tableau2.1, qui permet la différenciation des erreurs selon chaque classe en vue d'évaluer un classifieur. Définissons maintenant plusieurs mesures de manière formelle :

Catégorie Ci		Jugement Expert	
		Oui	Non
Jugement classifieur	Oui	TP_i	FP_i
	Non	FN_i	TN_i

Table 2.1 – Matrice de contingence de la classe C_i

— Le taux de vrais positifs ("True positive rate"),

$$TPr = \frac{TP}{Pos} = \frac{TP}{TP + FN}$$

— Le taux de vrais négatifs ("True negative rate"),

$$TNr = \frac{TN}{Neg} = \frac{TN}{TN + FN}$$

— Le taux de faux positifs ("False positive rate"),

$$FPr = \frac{FP}{Neg} = \frac{FP}{FP + TN}$$

— Le taux de faux négatifs ("False negative rate"),

$$FNr = \frac{FN}{Pos} = \frac{FN}{FN + TN}$$

— Le taux de bonne classification ou l'exactitude (accuracy)

$$acc = tbc = Pos * TPr + Neg * (1 - FPr)$$

— La précision

$$prec = \frac{TP}{PPos} = \frac{TP}{TP + FP}$$

— Le rappel (recall)

$$rec = TPr = \frac{TP}{Pos} = \frac{TP}{TP + FN}$$

Maintenant que nous avons caractérisé notre problème (estimation des taux de bonne et mauvaise classification, évaluation du type d'erreur,...) via la matrice de confusion, nous allons représenter les performances des systèmes de classification à l'aide de la courbe Précision–Rappel puis de la courbe ROC.

2.6.1 Courbe Précision–Rappel

Nous allons maintenant traiter de la mesure "*précision*" et de la mesure "*rappel*". Ces mesures très utilisées en recherche documentaire (information retrieval) permettent d'évaluer les performances (la pertinence) du retour d'information vis-à-vis d'une requête,[32],[31]. La *Précision* et le *Rappel* sont généralement utilisés pour traiter des bases de documents statiques ; cependant, ils peuvent être utilisés dans des environnements dynamiques tels que la fouille de pages web, lorsque le nombre de documents non pertinents correspondant à une requête augmente à la vitesse de création de pages web sur Internet. Considérons la figure 2.4 qui présente deux classifieurs évalués par la courbe *Précision-Rappel*. Dans la figure 2.4a, la base de test est équilibrée dans la distribution des classes (1 :1). Pour la courbe2.4b, les mêmes classifieurs sont utilisés, mais cette fois, le nombre d'éléments négatifs est dix fois plus important (1 :10). Nous constatons que les courbes *Précision-Rappel*, figures 2.4a et 2.4b, diffèrent sensiblement, ce qui indique une sensibilité de la représentation par rapport à la distribution. Il s'agit donc d'un problème important mettant en évidence la faiblesse de la courbe *Précision–Rappel*. De ce fait, nous devons faire appel à une autre représentation plus robuste vis-à-vis de la structure des bases de test. La **courbe ROC**, que nous allons maintenant présenter intervient dans ce sens.

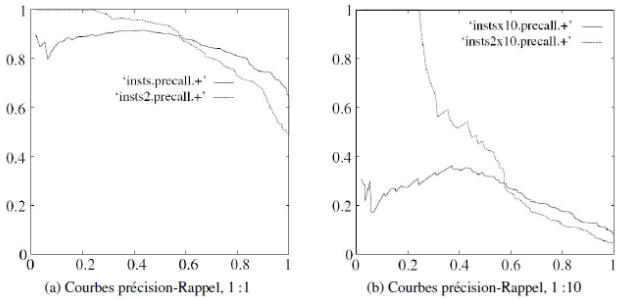
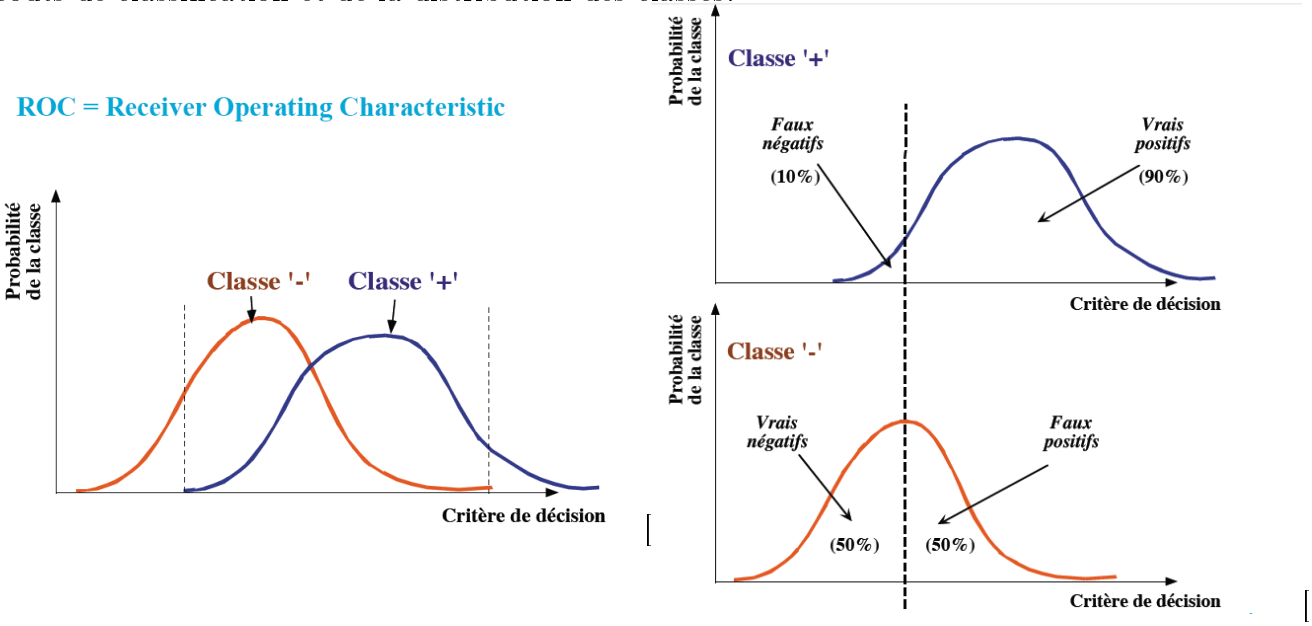


Figure 2.4 – Courbes Précision-Rappel vis-à-vis de la distribution des classes, [20]

2.6.2 Courbe ROC

La courbe ROC *(Receiver Operating Characteristics)* offre à la fois une vision graphique et une mesure pertinentes de la performance d'un classifieur. Elle possède de nombreux avantages par rapport aux mesures de rappel et précision par classe : : la performance est synthétisée par une unique mesure qui ne dépend pas des proportions de classe. Cet avantage se transforme néanmoins en inconvénient lorsqu'il s'agit de revenir rapidement aux mesures de rappel et de précision par classe ou d'estimer le comportement du classifieur selon les classes. Les mesures de rappel et précision sont en effet utiles car elles caractérisent précisément le comportement du classifieur sur chacune des classes. En particulier, lorsque les classes sont déséquilibrées, ces mesures fournissent des indications de performance plus représentatives et concrètes qu'un unique score pour le classifieur.[?] En général, les courbes ROC sont basées sur le taux de vrai positifs (tpr) et le taux de faux positifs (fpr). Il s'agit là de rapports qui ne dépendent donc pas de la distribution des classes. Cette méthode robuste permet de s'affranchir de la connaissance des coûts de classification et de la distribution des classes.



[?]

2.6.3 F–mesure

La *F-mesure* est un indicateur de synthèse communément utilisé pour évaluer les algorithmes de classification de données textuelles, à partir de la précision et du rappel. Elle est utilisée indifféremment pour les classifications et les catégorisations.

$$\text{F-mesure} = (1 + \beta^2) \cdot \frac{Precision \cdot Rappel}{(\beta^2 \cdot Precision) + Rappel}$$

La F-mesure correspond à une moyenne harmonique de la précision et du rappel. Le paramètre b permet de pondérer la précision ou le rappel et vaut généralement 1, [36]. La mesure devient :

$$\text{F-mesure} = 2 \cdot \frac{Precision \cdot Rappel}{Precision + Rappel}$$

L'avantage de ce choix est que lorsque la précision est égale au rappel, on obtient : $Prcision = Rappel =$ F-mesure. Ceci facilite la lecture et on recherche à maximiser la F-mesure en maximisant simultanément la précision et le rappel. Le problème que pose cette méthode est qu'elle ne permet pas la différenciation des erreurs et reste sensible à la distribution des classes car basée sur la précision et le rappel. Nous allons maintenant nous intéresser à l'aire sous la courbe ROC afin de comparer les deux mesures. Il est donc préférable d'utiliser la seconde mesure qui est à notre disposition à savoir : l'aire sous la courbe ROC.

conclusion

La classification est une étape importante dans l'analyse de données qui consiste à regrouper les données en classes similaires.Il existe donc une quantité importante de méthodes pour la classification de données.Toutes sont issues des recherches sur l'apprentissage *(" Machine Learning ")*. Dans ce chapitre, nous avons défini la notion de data mining, en donnant les différents technologies. Nous avons aussi présenté la notion d'apprentissage automatique et leur défirent méthodes de classification .puis nous définir les méthodes de préférence. Dans le chapitre suivant, nous présentons un état de l'art des travaux qui ont étudié de l'application de datamining pour l'inférence de réseau de régulation génétique.

Chapitre 3

La problématique

Introduction

L'inférence du réseau du gènes depuis les données d'expression est une tâche très difficile. Plusieurs méthodes ont été développé dans ce domaine mais une évaluation compressive couvre les méthode supervisée et non supervisé, et donne des lignes prédéfinis pour leur application pratique .Nous avons effectué une évaluation approfondie des méthodes d 'inférence sur des données d' expression simulées et expérimentales. Un environnement semi-supervisé avec un petit nombre d'échantillons seulement positifs, a surpassé les techniques non supervisées.

Problématique

L'idée de l'inférence est d'utiliser des données accessibles en très grande quantité, ainsi qu'un modèle, pour reconstruire la structure du réseau biologique considéré.

Les données accessibles à grande échelle sont les interactions protéine-protéine et le niveau d'expression de l'ensemble des gènes dans des conditions contrôlées. Cela s'applique essentiellement au réseau de de régulation.

Pour créer un classifieur (binaire) par apprentissage supervisé, on a besoin de deux classes, l'une est l'ensemble des éléments étiquetés par + (positif) et l'autre est l'ensemble des éléments étiquetés par (-) négatifs. Quand il y a une absence totale ou partielle d'étiquette dans une classe le problème est dit apprentissage semi-supervisé.

Les données décrivant les réseaux de régulation génétique sont rares et on ne connaît généralement qu'une petite partie des vraies interactions. La situation est encore pire pour les données négatives (non-interactions) parce que la validation expérimentale (dans les laboratoires) vise en grande partie à détecter (non à exclure) les interactions. les méthodes supervisées sont donc limitées à de petits ensembles d'apprentissage, ce qui affecte négativement leurs performances.

Les méthodes d'apprentissage semi supervisées consiste généralement en deux étapes :

1. la sélection d'un ensemble des négatifs fiables à partir de l'ensemble non étiqueté (*unlabeled data*).
2. application d'une méthode supervisée.

État de l'art des méthodes d'inférence de réseaux de régulation génétique

3.0.1 Les méthodes non supervisées

Les méthodes de classification non supervisée sont des techniques de regroupement (clustering) où un processus automatique sépare les données observées en groupes distincts sans aucune connaissance préalable des classes existantes. Les algorithmes de clusterisation groupent les gènes en fonction de leur profil d'expression basée sur une métrique qui calcule la similarité entre deux profils. La plupart des algorithmes utilisent le coefficient de corrélation statistique ou la distance Euclidienne. Ce sont les algorithmes les plus souvent utilisés pour l'analyse des données pour les puces à ADN.

Par exemple [39] regroupe presque toute les méthodes non supervisées utilisée dans la littérature telques CLR, ARACNE, MRNET et MRNET-B qui sont une partie de R package 'minet'. Et ont été appelés avec leurs paramètres par défaut, avec l'exception de ARACNE Avec le paramètre par défaut $eps = 0.0$, ARACNE a mal fonctionné et ils ont utilisé $eps = 0.2$. d'une manière similaire, l'auteur a appliqué la méthode GENIE , MINE et PCIT *(Partial Correlation and Information Theory)* qui ont été installés et évalués avec des paramètres par défaut. Toutes les autres méthodes ont été mises en œuvre selon leurs publications respectives. SPEARMAN-C, EUCLID et SIGMOID sont des implémentations de nos propres algorithmes d'inférence.

La corrélation

Les méthodes d'inférence de réseau basées sur la corrélation supposent que les niveaux d'expression corrélés entre deux gènes sont indicatifs d'une interaction régulatrice. Les coefficients de corrélation vont de +1 à -1 et un coefficient de corrélation positif indique une activation Interaction, alors qu'un coefficient négatif indique une interaction inhibitrice. La mesure de corrélation commune par *Pearson* est définie comme :

$$corr(X_i, X_j) = \frac{cov(X_i, X_j)}{\sigma(X_i).\sigma(X_j)}$$

Où X_i et X_j sont les niveaux d'expression des gènes i et j, $cov(.,.)$ désigne la covariance, et $\sigma(.)$ est l'écart type.

La mesure de corrélation de *Pearson* Suppose des valeurs normalement distribuées, une hypothèse qui ne se limite pas nécessairement aux données d'expression génique. Par conséquent, les mesures fondées sur le classement sont fréquemment Utilisé, les mesures par *Spearman* et *Kendall* étant les plus courantes.

La méthode de *Spearman* est simplement le coefficient de corrélation de *Pearson* pour les valeurs d'expression classées, et le coefficient

$$\tau(X_i, X_i) = \frac{con(X'_i, X'_j) - dis(X'_i, X'_j)}{\frac{1}{2}n(n-1)}$$

Où X'_i et X'_j sont les profils d'expression classés des gènes i et j. $Con(.,.)$ désigne le nombre de concordant et $dis(.,.)$ le nombre de paires dis-concordant dans X'_i et X'_j , Les deux profils étant de longueur n. Parce que notre évaluation de la précision de prédiction ne fait pas de distinction entre l'inhibition et l'activation Interactions, les pondérations d'interaction prédites sont calculées comme la valeur absolue des coefficients de corrélation [39]

$$w(ij) = |corr(X_i, X_j)|$$

.

L'information mutuelle(IM)

Les méthodes d'inférence du réseau d'information mutuelle comprennent une sous-catégorie de méthodes d'inférence du réseau, qui détermine les interactions régulatrices entre les gènes en fonction de l'information mutuelle par paires.

Dans une première étape, ces méthodes nécessitent le calcul de la matrice d'information mutuelle (**MIM**), une matrice carrée dont l'élément mim_{ij} est donné par l'information mutuelle entre X_i et X_j :

$$mim_{ij} = I(X_i; X_j)$$

Où X_i et X_j sont des variables aléatoires dénotant les niveaux d'expression des gènes i et j, respectivement.

Les principaux avantages des réseaux d'information mutuelle pour l'inférence des réseaux réglementaires transcriptionnels sont les suivants :

- La complexité informatique est abordable. Cela résulte du fait que seuls $\begin{pmatrix} n \\ 2 \end{pmatrix}$ appels d'informations mutuelles, basés sur des distributions de probabilité bivariées, sont nécessaires pour calculer le **MIM**.
- Le nombre d'échantillons requis est plutôt faible, car seule la distribution bivariée doit être estimée.

Dans ce qui suit, nous examinons d'abord deux méthodes d'inférence de réseau à la fine pointe de la technologie basées sur des informations mutuelles par paires. Nous procédons en décrivant deux estimateurs d'information mutuelle couramment utilisés.[37]

ARACNE

Le logiciel **ARACNE** proposé par [12] débute par le calcul de l'ensemble des Information mutuelles (IMs) par paire de gènes et retient uniquement les relations dont l'IM est supérieur à un seuil défini par permutation des données. A partir de cet ébauche de réseau l'algorithme analyse successivement tout les triplets de gènes formant une clique. Le principe de DPI *(Data Porcessing Inequality)* est alors testé pour chacun de ces triplets et vise à supprimer la relation dont l'IM est la plus faible parmi les trois relations composant la clique. Du fait que ce processus repose uniquement sur des comparaisons d'IM, celui-ci est peu sensible à la précision du calcul des IMs dès lors que le biais est commun à l'ensemble des gènes. Une condition supplémentaire permet de conserver les trois relations d'une clique si aucune des IMs est significativement plus faible.

De même [27] dans l'algorithme **CLR**, ne recourent pas aux mesures conditionnelles mais proposent de corriger l'IM d'une $paire(G_i, G_j)$ en fonction de l'IM de chacun de ces deux gènes avec tous les autres gènes du réseau. Un nouveau score est ainsi proposé, basé sur la distance euclidienne de l'IM entre G_i et G_j avec les distributions de l'IM de (G_i, G_k) et de (G_j, G_k) pour chaque gène G_k du réseau. Ce nouveau score agit de la même manière qu'un test de significativité qui permet de sélectionner une relation qui se démarque du bruit de fond local aux deux gènes reliés. Le réseau final est obtenu grâce à un seuil sur ces IMs corrigées.

MRNET

MRNET utilise l'IM entre les profils d'expression et un algorithme de sélection de fonctionnalité [minimum-redondance-maximum-relevance (**MRMR**)] pour inférer les interactions entre les gènes.Plus précisément, la méthode place chaque gène dans le rôle d'un gène cible j avec tous les autres gènes V comme régulateurs. L'IM entre le gène cible et les régulateurs est calculé et la méthode **MRMR** est appliquée pour sélectionner le meilleur sous-ensemble des régulateurs. **MRMR** étape par étape construit un ensemble S en sélectionnant les gènes i^{MRMR} avec la plus grande valeur MI et la plus petite redondance basée sur la définition suivante [39] : $i^{MRMR} = argmax_{i \epsilon V \S}(s_i)$

Avec

$$s_i = u_i - r_i$$

. Le terme de pertinence

$$u_i = I(X_i, X_j)$$

est donc l'IM entre le gène i et la cible j, et le terme de redondance r_i est défini comme :

$$r_i = \frac{1}{|S|} \Sigma_{k \in S} I(X_i, X_k)$$

Les poids d'interaction w_{ij} sont finalement calculés comme

$$w_{ij} = max(s_i, s_j)$$

.

MRNET-B

MRNET-B est une modification de **MRNET** qui remplace la stratégie de sélection avancée pour identifier le meilleur sous-ensemble des gènes régulateurs par un retour stratégie de sélection suivie d'un remplacement séquentiel.[39]

CLR(Context Likelihood of Relatedness)

L'algorithme **CLR** est une extension de l'approche du réseau de pertinence. Cette dernière approche a été introduite pour le regroupement de gènes et appliquée avec succès pour inférer des relations entre l'expression de l'ARN et la susceptibilité chimiothérapeutique. L'approche des réseaux de pertinence consiste à inférer un réseau dans lequel une paire de gènes (X_i, X_j) sont liés par un bord si l'information mutuelle $I(X_i; X_j)$ est supérieure à un seuil donné θ.La complexité de la méthode est $O(n^2)$ puisque toutes les interactions par paires sont considérées. L'algorithme **CLR** obtient un score de la distribution empirique de l'information mutuelle pour chaque paire de gènes. En particulier, au lieu de considérer l'information $I(X_i; X_j)$ entre les gènes X_i et X_j elle estime un score

$$w_{ij} = \sqrt{z_i^2 + z_j^2}$$

où

$$z_i = max(0, \frac{I(X_i; X_j - \mu_i)}{\sigma_i})$$

.

Les paramètres μ_i et σ_i sont la moyenne et l'écart type de la distribution empirique des valeurs d'information mutuelle $I(X_i; X_k)$ de X_i avec toutes les autres variables X_k $(k = 1, \ldots, n)$. L'algorithme **CLR** a une complexité dans $O(n^2)$. Il a été appliqué avec succès pour déchiffrer le réseau réglementaire de transcription de E. coli.[37]

3.0.2 Les méthodes supervisée

Il existe de nombreuses méthodes de classification supervisée (analyse discriminante, analyse de voisinage, machines à support vectoriel (SVM), etc.) permettant de prédire la classe du gène,l'objectif étant de minimiser le taux d'erreur de prédiction.

Nous nous contentons à la méthode supervisée la plus célèbre pour l'inférence des réseaux

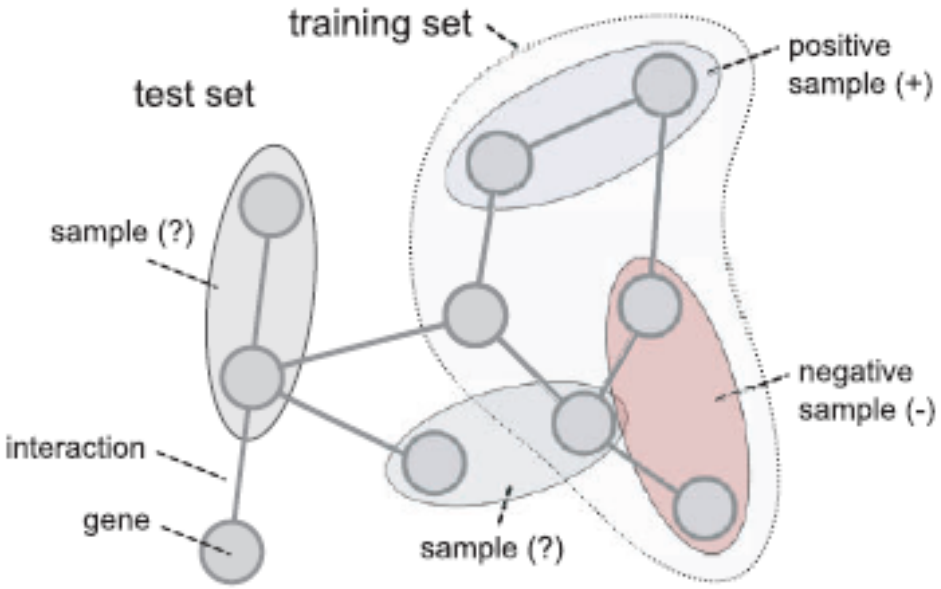


FIGURE 3.1 – Extraction of samples for the training and test set from a gene interaction network [39]

de régulation génétique à savoir **SIRENE** qui est une méthode générale pour inférer de nouvelles relations de régulation entre *TF* (facteurs de transcription) connu et tous les gènes d'un organisme. Il nécessite deux types de données en tant qu'entrées. Tout d'abord, chaque gène de l'organisme doit être caractérisé par certaines données, dans notre cas, un vecteur de valeurs d'expression dans un compendium de profils d'expression. Deuxièmement, une liste des relations de régulation connues entre le *TF* connu et certains gènes est nécessaire. Plus précisément, pour chaque *TF*, nous avons besoin d'une liste de gènes connus pour être réglementé par le *TF* et, si possible, une liste de gènes dont on ne sait pas réglementer. De telles listes peuvent généralement être construites à partir de bases de données publiquement disponibles de règlements caractérisés expérimentalement, par exemple RegulonDB pour les gènes E.coli. Bien que de telles bases de données ne contiennent généralement pas d'informations sur l'absence de réglementation. Lorsque ces données sont disponibles, **SIRENE** divise le problème de l'inférence du réseau réglementaire dans une classification binaire multiple sous-problèmes, un sous-problème étant associé à chaque *TF*. Plus précisément, pour chaque *TF*, **SIRENE** forme un classificateur binaire pour discriminent entre les gènes connus pour être réglementés et les gènes connus pour ne pas être réglementés par le *TF*, en fonction des données qui caractérisent les gènes (par exemple, les données d'expression). La raison d'être de cette approche est-ce que, bien que nous n'ayons aucune hypothèse concernant le rapport entre le niveau d'expression mesuré d'un *TF* et ses cibles, nous supposons que si deux gènes sont régulés par le même *TF*, ils sont susceptibles d'afficher des modèles d'expression similaires. Dans notre mise en œuvre, nous utilisons une **SVM**

pour résoudre les problèmes de classification binaire, mais tout autre algorithme de classification binaire supervisée pourrait en principe être utilisé. Une fois formé, le modèle associé à un *TF* donné est capable d'affecter à chaque nouveau gène, non utilisé pendant la formation, un score qui tend à être positif et grand lorsqu'il croit, basé sur les données qui caractérisent le gène, que le gène est réglementé par le *TF*. La dernière étape consiste à combiner tous les scores des différents modèles pour classer les interactions du gène *TF* candidat dans une liste unique en diminuant le score.[19]

En résumé, **SIRENE** décompose le problème difficile du gène l'inférence du réseau réglementaire dans un grand nombre de sous-problèmes qui tentent d'estimer les modèles locaux pour caractériser les gènes réglementés par chaque *TF*. Une approche similaire a été proposée par [14] pour inférer des graphiques non dirigés et testé avec succès sur la reconstruction des réseaux métaboliques et PPI. Ici, nous sommes confrontés à un problème légèrement différent, car le graphique que nous souhaitons déduire est dirigé et il suffit d'inférer des modèles locaux pour prédire les gènes réglementés par un *TF* donné.[19]

3.0.3 les méthodes semi-supervisée

Les données qui décrivent les réseaux organisationnels sont dispersés et nous ne savons généralement qu'une petite partie des interactions réelles. La situation est encore pire pour les données négatives (non-interactions) en raison de la vérification expérimentale vise essentiellement à détecter, mais ne pas exclure les interactions.

Le cas que tous les échantillons dans l'ensemble d'apprentissage peuvent être étiquetées comme positif ou négatif est rarement donné dans les problèmes d'inférence des réseaux de régulation et les méthodes supervisée sont limitée à des petits ensembles d'apprentissage, ce qui affecte négativement leur performance.

Les méthodes semi-supervisée tentent de tirer parti les avantages des échantillons non étiquetées dans l'ensemble d'apprentissage par la prise de leur distribution en compte, et même peuvent être apprises par seulement des données positivement étiquetées.

La figure 3.2 montre de l'étiquetage requis des données pour les différentes approches. Les méthodes supervisées exigent que tous les échantillons dans l'ensemble d'entraînement soient étiquetés, alors que les méthodes non supervisées ne nécessitent aucun étiquetage.[39]

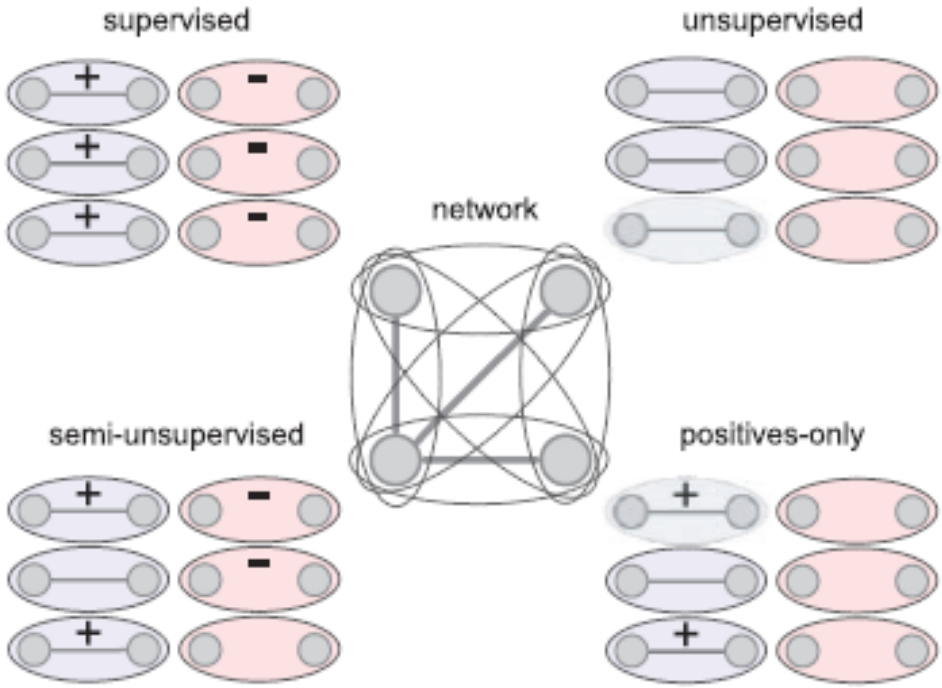


Figure 3.2 – Original labeling of samples for supervised,unsupervised, semi-supervised and positives-only prediction methods. All the six samples within a sample set are generated by a four-node network with three interactions

[39]

parmi les méthodes semi-supervisée dans la littérature on peut citer :

Bagging - SVM

Baggin SVM est un ensemble des techniques qui améliore généralement la performance des individuels de classificateurs lorsqu'ils sont instables ou ne sont pas corrélés entre eux. Les apprentissages positifs ont une structure particulière qui conduit à des classificateurs instables en raison de la contamination positive de l'ensemble non marqué qui peut être avantageusement exploité par une procédure de type bagging. L'approche collecte le résultat d'une énorme classification de nombre (par exemple 1000), où chaque classifieur,F_i, est formé avec les exemples positifs connus, P_{tfi}, et un ensemble aléatoire de candidats NC négatifs tirés uniformément d'$\cup_{tfi}$, considérés comme des exemples négatifs. Le classificateur d'ensemble, F, marque un exemple non marqué g en faisant la moyenne des scores obtenus par cet exemple à chaque épreuve :

$$F(g) = \frac{\Sigma_{i \epsilon T_g} F_i(g)}{|T_g|}$$

Où g est un membre tiré d'$\cup_{tfi}$ F_i est le *i-th* classifier, et T_g est l'ensemble de classifieurs partiels qui n'ont pas été formés avec g, c'est-à-dire que l'exemple non marqué g n'a pas été dessiné par

la sélection aléatoire.[33]

Spy-SVM

Spy-SVM est une technique proposée dans l'article de [33] comme suite. Un pourcentage de points positifs connus,$(s_1, s_2, \ldots, s_k)$,sélectionnés au hasard à partir de P_{tfi}, qui agissent comme "*espions*", sont envoyés à l'ensemble non étiqueté $\cup_{tfi}$. Un algorithme de classification SVM est formé avec des exemples positifs (sans les espions) et l'ensemble non marqué (avec les espions) supposés négatifs. Les espions devraient se comporter de manière identique aux exemples positifs inconnus appartenant à $\cup_{tfi}$, ce qui permet d'inférer de manière fiable le comportement des exemples positifs inconnus. Un seuil t est utilisé pour décider si un exemple dans $\cup_{tfi}$ est un négatif fiable ou non. Les exemples avec une probabilité d'être positifs,$P(f(x) = +1))$, inférieurs à t sont les exemples négatifs les plus probables. Le seuil est calculé intuitivement comme le minimum de la probabilité d'être positif d'espions, c'est-à-dire

$$t = minP(f(s_1) = +1), P(f(s_2) = +1), ..., P(f(s_k) = +1)$$

. Cela signifie que tous les exemples d'espionnage devraient être classés comme positifs.[33]

PSoL - Positive Sample only Learning

PSoL sélectionne un fort exemple négatif en utilisant la mesure de distance euclidienne. L'algorithme commence par un candidat négatif qui est l'exemple le plus éloigné de P_{tfi} calculé comme le maximum de la distance minimale par rapport aux éléments de P_{tfi}. Plus de candidats négatifs sont sélectionnés à partir de l'ensemble non marqué $\cup_{tfi}$ satisfaisant les contraintes qui sont différentes des exemples positifs connus et les plus éloignées des négatives précédemment sélectionnées. L'algorithme suppose que les exemples négatifs dans l'ensemble non marqué sont situés loin des points positifs et des exemples négatifs sélectionnés précédents. La dernière condition assure que l'ensemble négatif couvre l'ensemble des exemples négatifs dans l'ensemble non marqué. Compte tenu de cet ensemble négatif initial, la méthode *PSoL* élargit itérativement le jeu négatif en utilisant un SVM à deux classes formées avec des points positifs connus et la sélection négative actuelle. L'expansion du jeu négatif est répétée jusqu'à ce que la taille de l'ensemble non marqué restant dépasse un nombre prédéfini. À cette dernière étape, les points de données non étiquetés avec les valeurs les plus importantes de la fonction de décision positive sont déclarés comme positifs. [33]

conclusion

Les techniques de classification pour l'analyse de puces à ADN ont été largement utilisées pour identifier des groupes de gènes partageant des profils d'expression similaires et les résultats obtenus sont très concluants. Néanmoins, ces méthodes ne permettent de découvrir qu'une partie des relations parmi toutes les relations potentielles entre les gènes, les classes recherchées doivent être vérifiées sur l'ensemble des expérimentations et un gène ne peut appartenir qu'à une seule classe. En effet, ces méthodes ne donnent pas la relation exacte qui peut exister entre deux gènes ou deux groupes. L'application des algorithmes de classification supervisée ou non supervisée ou semi-supervisée aux données de puces ADN pose un problème : chaque gène étant considéré comme une variable, le nombre de variables(environ 10000) est trop grand comparé au nombre d'observations disponibles(environ 100). Dans le chapitre 4, on va posé notre proposition dans ce problème d'inférence de réseau génétique.

Chapitre 4

La réalisation

Introduction

Habituellement, les classifieurs binaires prennent en entrée deux ensembles de données, l'un contenant des données étiquetées positivement et l'autre des données étiquetées négativement. L'objectif étant d'apprendre un séparateur de ces deux ensembles de données. Malheureusement, il arrive souvent que ces ensembles contiennent uniquement des données étiquetées positivement et des données non étiquetées à la fois positives et négatives.

Dans ce chapitre nous allons d'abord expliquer les outils utilisés dans la réalisation de notre application, le langage de programmation utilisé dans le développement, la base de données utilisée. On va décrire les outils nécessaires pour réaliser notre projet, à savoir : Le langage Python et un corpus sous forme d'une Puce ADN (ainsi que d'autres données) sur lesquelles on applique notre modèle de classification. Sans oublier de mettre un coup d'œil sur les méthodes choisies qui sont kmeans (non supervisée), SVM(supervisée). Ces méthodes seront appliquées sur notre propre corpus sous forme des matrices que nous avons recueilli à partir de Puce ADN. Enfin, nous allons présenter les résultats obtenus par notre implémentation.

4.1 Les outils utilisés :

4.1.1 Python

Python est un langage de programmation objet, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions ; il est ainsi similaire à R, Perl, Ruby, Scheme, Smalltalk et Tcl. Le langage Python est placé sous une licence libre et fonctionne sur la plupart des plates-formes informatiques, des supercalculateurs aux ordinateurs centraux, de Windows à Unix avec notamment GNU/Linux en passant par macOS, ou encore Android, iOS, et aussi avec Java ou encore .NET. Il est conçu pour optimiser la productivité des programmeurs en offrant des outils de haut niveau et une syntaxe simple à utiliser. Il est également apprécié par certains pédagogues qui y trouvent un langage où la syntaxe, clairement séparée des mécanismes de bas

niveau, permet une initiation aisée aux concepts de base de la programmation. [9]

4.1.2 Pourquoi nous avons utilisé python ?

- La facilité d'apprentissage ;
- La montée de la data science et du machine learning ;
- Python tourne sur presque toutes les plateformes ;
- La rapidité de développement. Un programme Python de 50 lignes peut représenter dans d'autres langages, des programmes de plusieurs centaines de lignes. Ce qui fait qu'en fin de compte, même avec un programmeur Python pas assez rapide, on peut gagner beaucoup de temps au niveau du développement ;
- Python est ouvert aux autres langages et technologies. Python a choisi de collaborer avec les autres langages et technologies. On peut intégrer l'interpréteur Python dans son propre programme, et ajouter le langage comme système de Scripting (c'est le cas de Blender) ou plugin (comme avec Sublime Text). On peut appeler du code Python depuis d'autres langages et vice-versa.
- Python n'a pas besoin d'un EDI. Avec Python, on peut se passer des outils complexes comme Visual Studio ou Eclipse. On peut programmer en Python avec un simple éditeur de texte à coloration syntaxique ;[23]

4.1.3 Les puces à ADN

Les puces à ADN ou biopuces ;utilisées pour analyser l'expression d'un gène parmi un nombre important de gènes dont la fonction dans l'organisme est connue ou inconnue. Elles sont efficaces pour vérifier l'homologie entre deux brins d'ADN, pour détecter des polymorphismes ou des mutations génétiques au sein de l'organisme. Toutefois, dans une définition technique plus exacte, elle représente un arrangement ordonné de plusieurs milliers de gènes séquencés, identifiés et imprimés sur un support solide imperméable, généralement fait de verre, de silicium ou bien de membrane en nylon. Chaque gène, quelque soit le support sur lequel il est imprimé, correspond à un fragment d'ADN génomique, d'ADN complémentaire ou des oligonucléotides chimiquement synthétisés.[25]

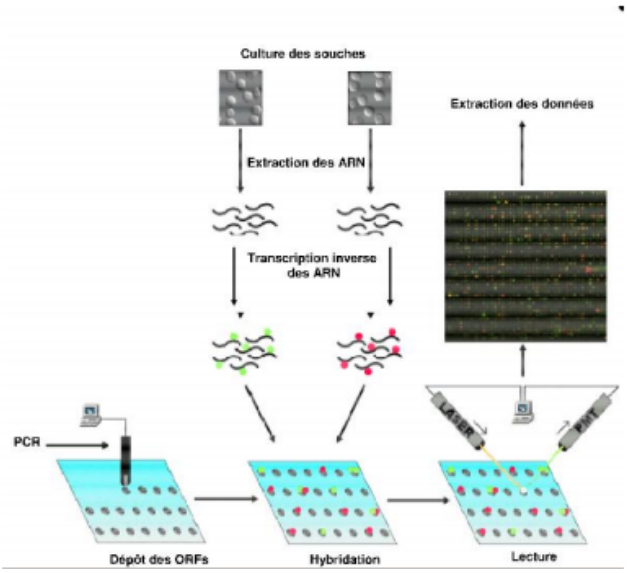


Figure 4.1 – Schématisation de la technique d'analyse du transcriptome par la technologie des puces à DNA

[25]

4.1.4 De Données(Puce ADN) à la Connaissance(réseau génétique) :

Le processus d'extraction de connaissances à partir des données d'expression est décrit dans la figure 4.2. Il consiste d'une part à analyser les images et à normaliser les données et d'autre part à appliquer des méthodes statistiques et informatiques pour obtenir de la connaissance utile pour les experts.[34]

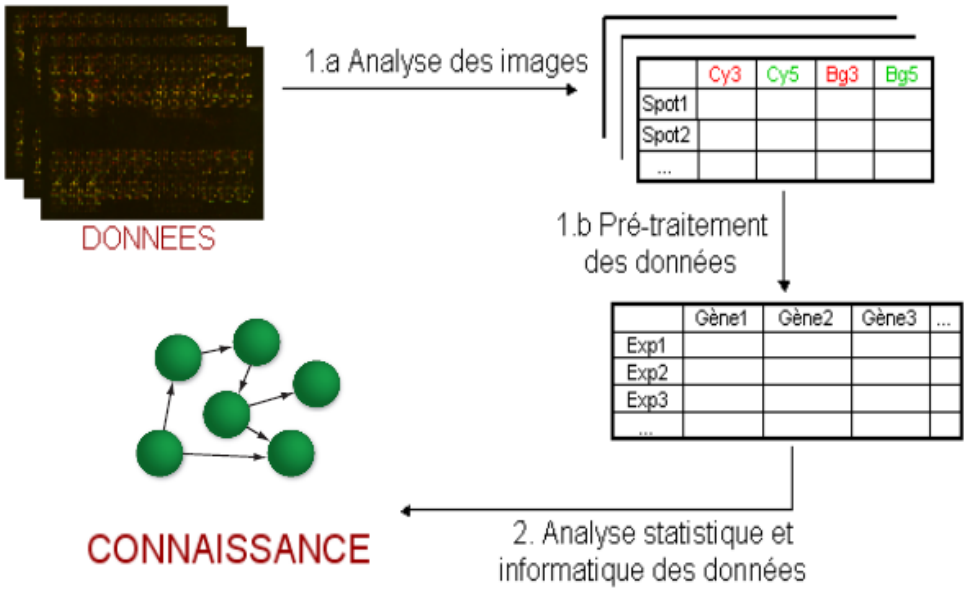


Figure 4.2 – Traitement des données d'expression [34]

4.1.5 Différentes techniques de datamining pour les puces à ADN :

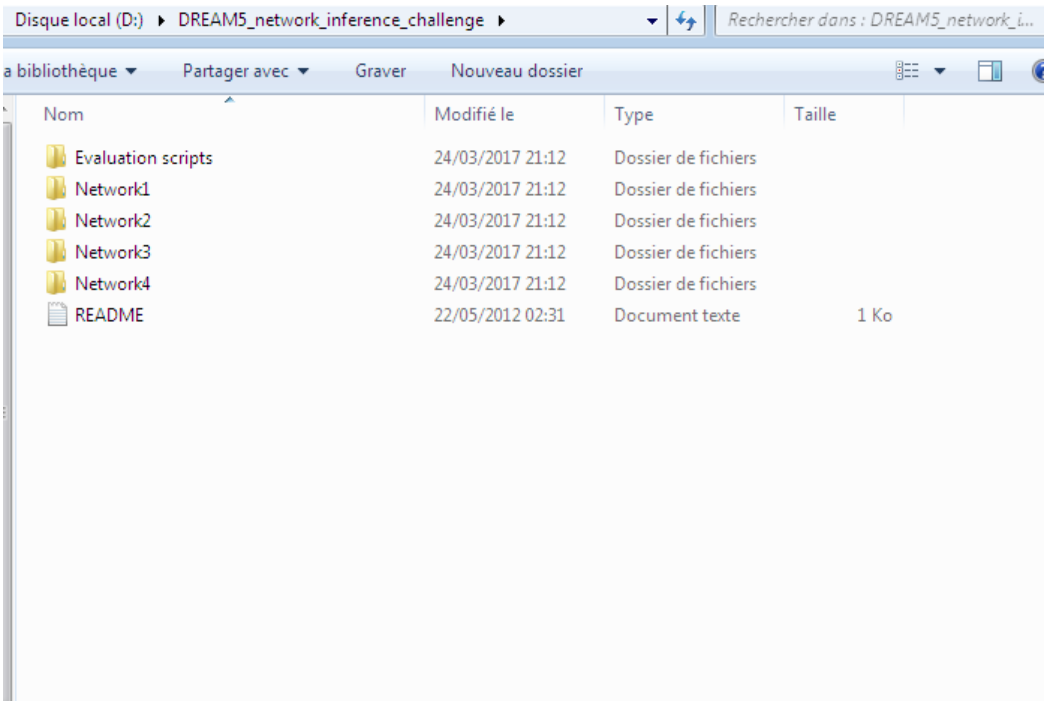
Les résultats de puces à ADN sont des matrices représentant les niveaux d'expression des gènes sous plusieurs conditions étudiées. A chaque gène est associé un profil d'expression. Le but de l'analyse est d'identifier les gènes ayant des profils semblables (ou responsable des phénomènes étudiés), ces gènes peuvent être régulés par les même facteurs de transcription ou intervenir dans le même processus biologique. En raison du nombre important de gènes et de la complexité des réseaux géniques, les techniques de datamining, d'apprentissage et de statistique sont avérées un outil très utile pour l'analyse de profils d'expression. Parmi les techniques utilisées, nous citons la classification (svm, reseau de neurones...), le clustering (kmeans), et règles d'association. L'un des objectifs de l'analyse des données d'expression consiste à classer les profils en fonction de leur différence d'expression selon certains facteurs biologiques en K classes (tumeur, cancer, type de bactérie ...). Les algorithmes de classification sont définis comme des méthodes de répartition d'un ensemble d'objets (points ou vecteurs) en plusieurs sous-ensembles, sur la base de leurs similarités ou dissimilarités. Le but est de construire des groupes qui minimisent la variabilité intra-groupe tout en maximisant les distances inter-groupes. Plus précisément, ils visent à trouver l'ensemble des groupes (gènes ou échantillons) dont les membres sont très similaires mais distants des autres membres sur la base de leur profil d'expression. Les algorithmes de classification se regroupent en deux grandes catégories : les approches supervisées et non supervisées. Les méthodes non supervisées groupent les objets sans connaissance a priori. Ces techniques sont dites exploratoires et sont essentiellement employées pour la découverte de classes. A l'inverse,

les méthodes supervisées utilisent de la connaissance a priori. Parmi les méthodes utilisées, citons la classification hiérarchique, les cartes topologiques de Kohonen et les méthodes dites des nuées dynamiques. Dans ce cadre, les problèmes de choix du nombre de classes et de méthodes de classification deviennent des problèmes de choix de modèles, et il n'existe pas de méthode générique pour choisir le nombre de classes ni la meilleure méthode de classification (cluster, gene cluster,. . .) .

4.2 Les données :

DREAM5_network_inference_challenge :

Ce répertoire contient les données d'entrée, les gold standards et les noms de gènes authentiques pour les quatre networks.



NETWORKS :

- Network1 : In silico
- Network2 : S. aureus
- Network3 : E. coli
- Network4 : S. cerevisiae

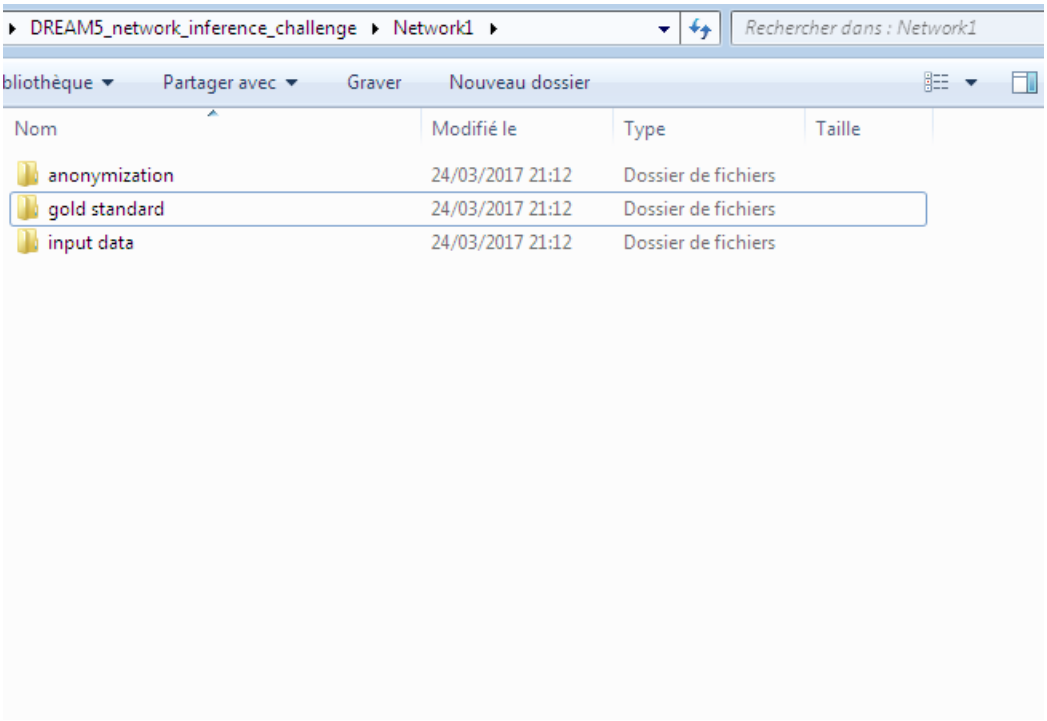
3 Networks réels : E. coli, S. cerevisiae, S. aureus (pas gold standard pour le dernier, mais les prédictions communautaires seront vérifiées expérimentalement).

1 réseau simulé : In silico.

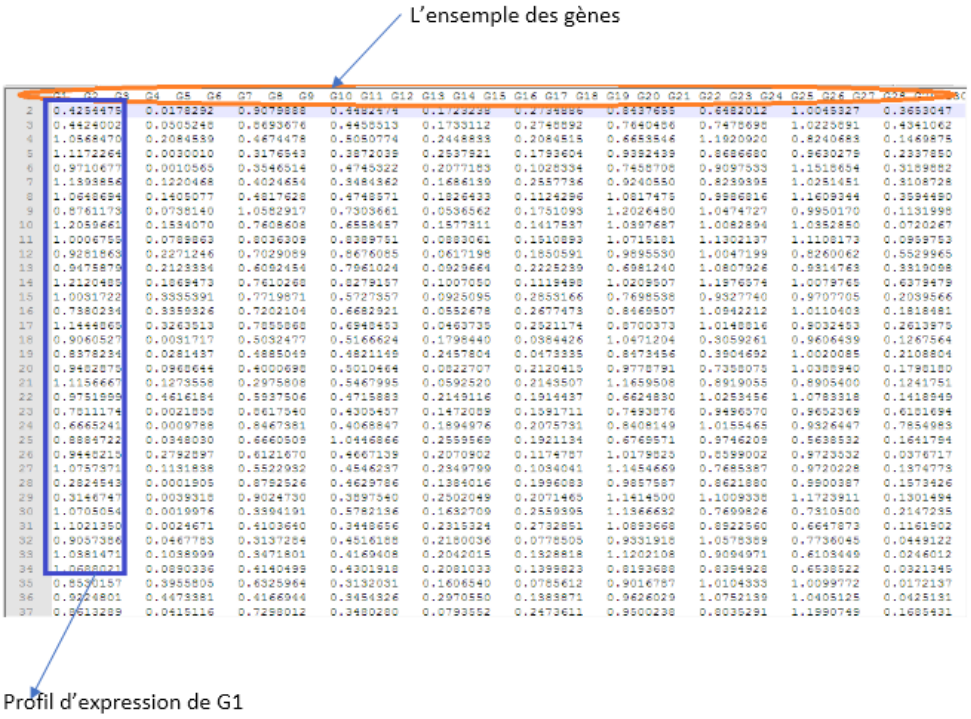
Les TF potentiels sont censés être connus à l'avance.

Network	# TFs	# Genes	# Chips
in-silico	195	1643	805
S. aurus	99	2810	160
E. coli	334	4511	805
S. cerevisiae	333	5950	536

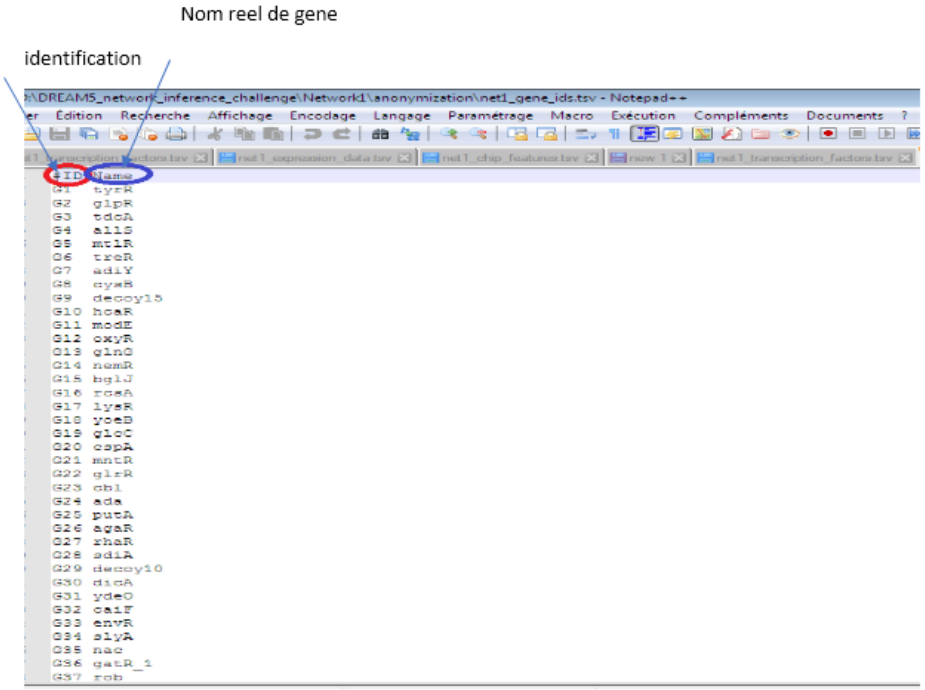
Network1 :



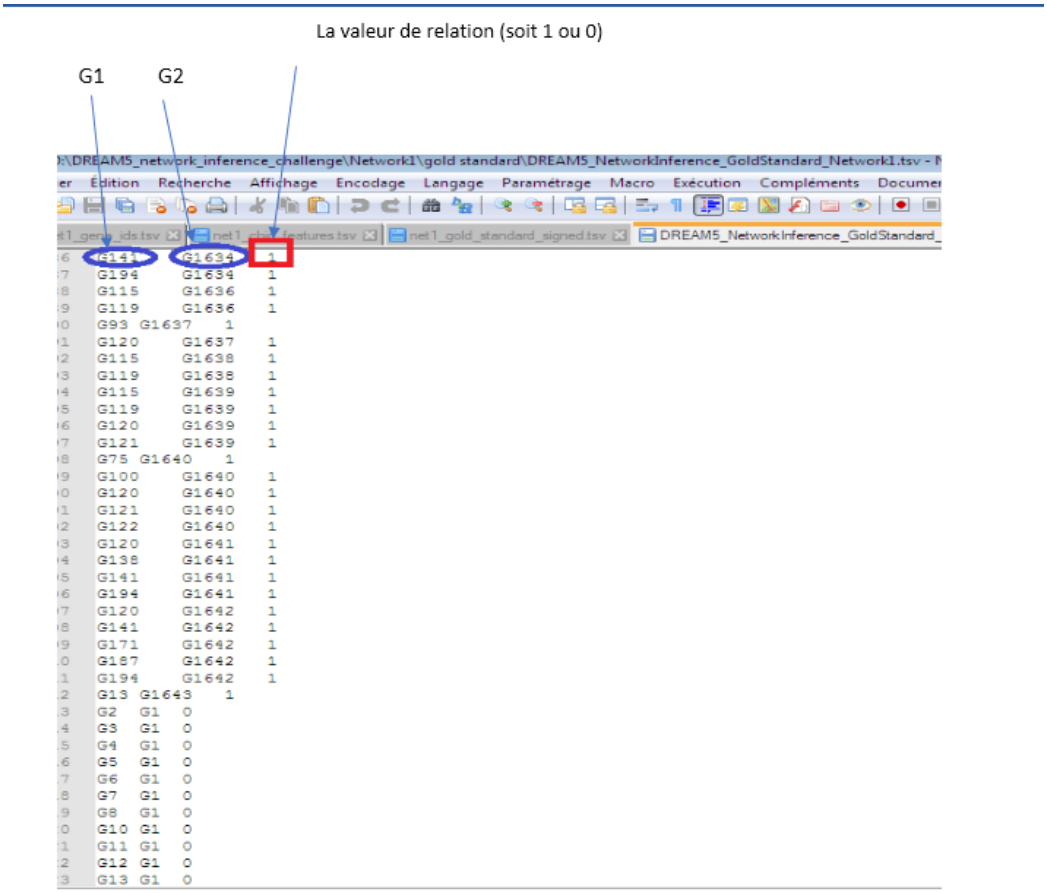
Network1/Input Data/net1_expression_data :



Network1/anonymization/net1_gene_ids :



Network1/goldstandard/DREAM5_NetworkInference_GoldStandard_Network1 :



4.3 Notre proposition :

1. l'ensemble positif (interaction) est les l'ensemble des gènes qui ont une relation avec les facteurs de transcription. Le reste des gènes est l'ensemble non étiqueté (unlabeled).
2. Afin d'extraire des négatifs fiables on va procéder comme suit :
   — rassembler les gènes similaires de l'ensemble non étiqueté en clusters
   — mesurer la distance (mesure de similarité) entre un représentant de chaque cluster (barycentre par exemple) avec les éléments de la classe positif (comparaison un par un). les clusters les plus loins sont ceux qui ont plus de chance d'être des négatifs. Dans cette étape on va choisir la mesure de similarité la plus fiable (z-score, information mutuelle, spearman ...etc)
3. apprendre un calssifieur SVM sur les deux classes positif et négatif (extraite)
4. mesurer les performances (auroc par exemple) de notre classifieur en utilisant un ensemble de test

4.4 Application sur les données (utiliser le langage python) :

Nous allons travailler en premier temps sur les données du répertoire (`DREAM5_network_inference_c`

- Un gène est représenté par un vecteur de valeurs d'expression génétique. Regarder le fichier (`input data\net1_expression_data.tsv`) les lignes représentent les conditions (échantillons), et les colonnes représentent les gènes le gène G1 par exemple sera représenter par le vecteur [0.4254475, 0.4424002, 1.0568470, 1.1172264,...,0.6812067]
- le fichier (`input data\net1_transcription_factors.tsv`) contient l'ensemble des facteurs de transcription
- le fichier (`gold standard\DREAM5_NetworkInference_GoldStandard_Network1.tsv`) contient les interactions (relations) entre les gènes (1 pour dire qu'il y a une relation et 0 sinon) dans notre cas on ne s'intéresse qu'aux relations 1 les autres sont considérées comme non étiquetées
- le fichier (`anonymization\net1_gene_ids.tsv`) contient les noms des gènes.

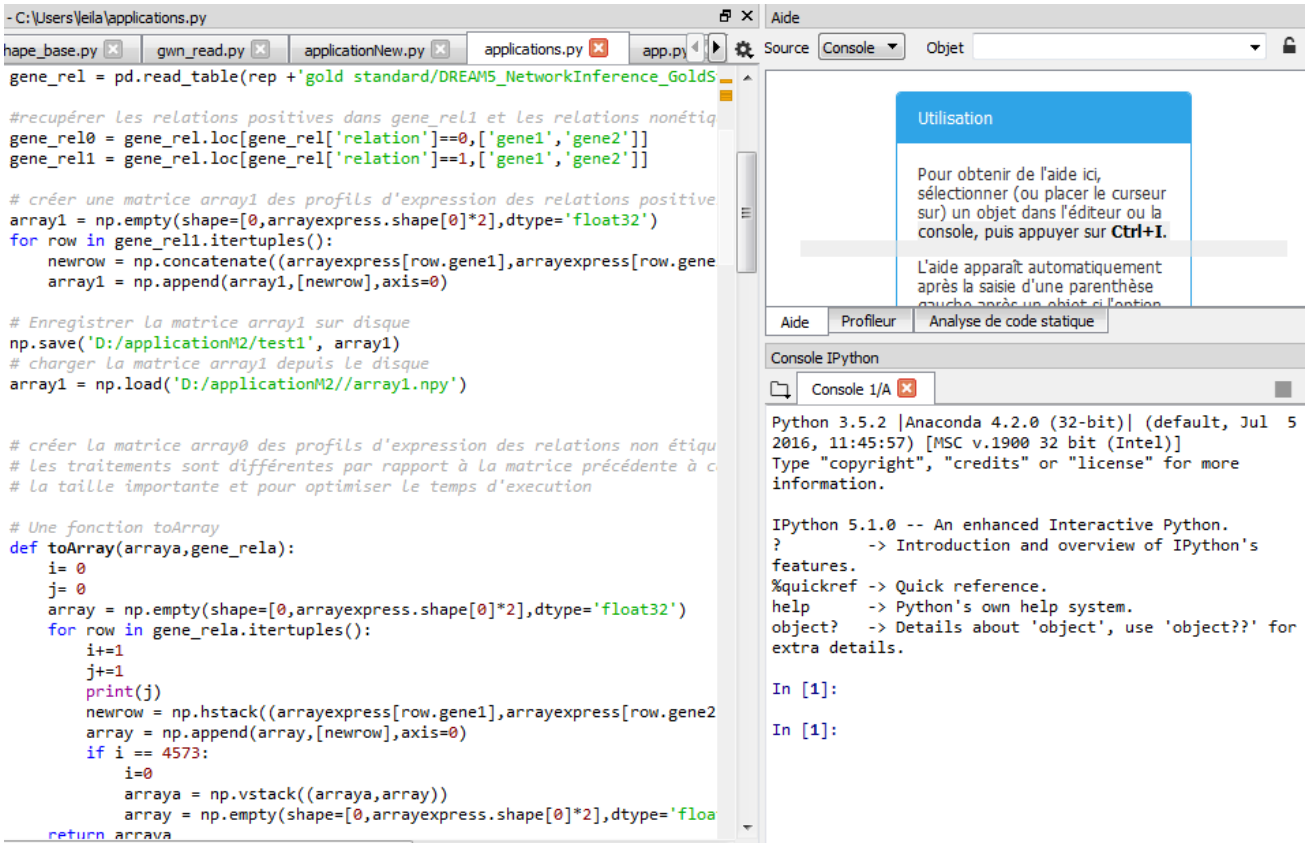
4.4.1 validation croisée (Cross-validation)

L'échantillon original est divisé en k échantillons, puis un des k échantillons sont sélectionnés comme ensemble de test et les (k-1) autres échantillons constitueront l'ensemble d'apprentissage. L'erreur de prédiction moyenne est calculée pour ces échantillons. L'opération est répétée pour un autre échantillon de validation parmi les (k-1) échantillons qui n'ont pas encore été utilisés pour la validation du modèle jusqu'à ce que tous les k sous échantillons seront utilisés une fois comme ensemble de validation. La moyenne des k erreurs est enfin calculée pour estimer l'erreur de prédiction.[7]

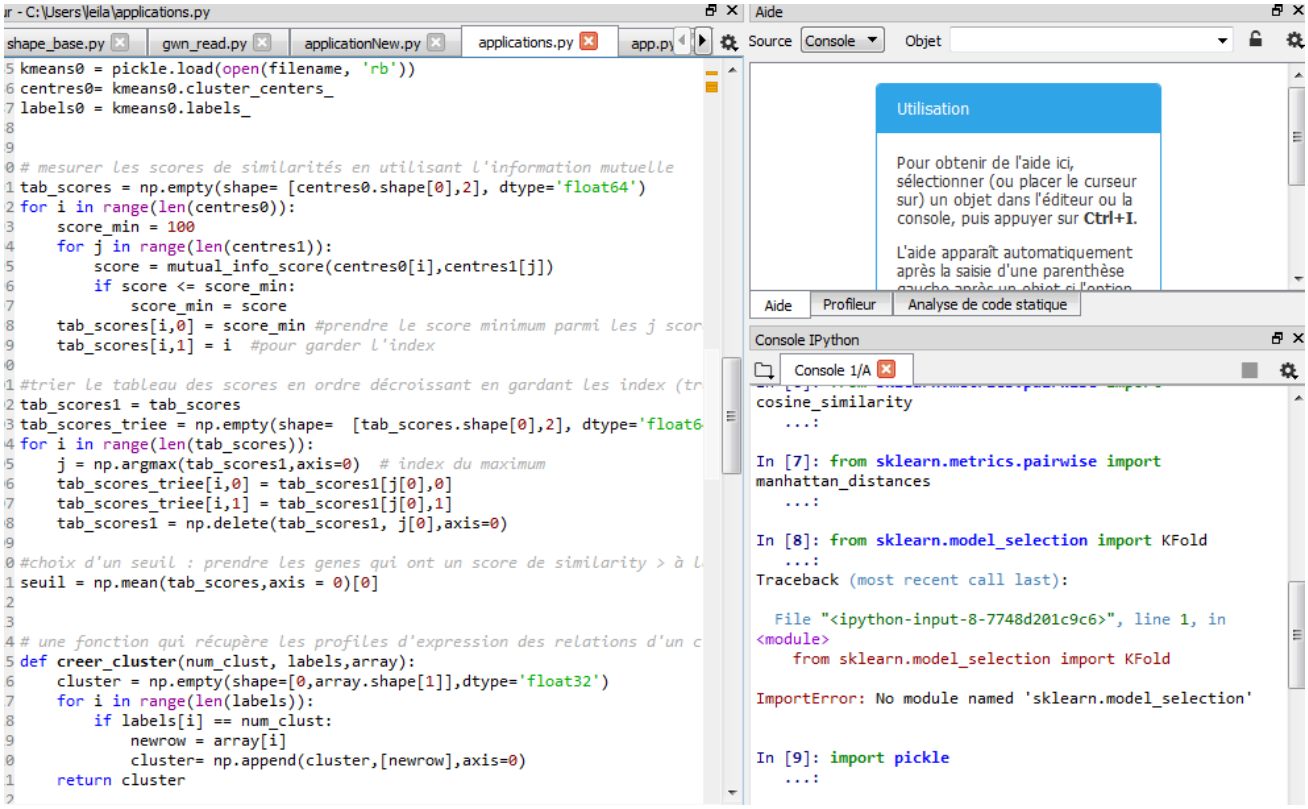
4.5 Démarche de l'implémentation (utilisé Python) :

Dans les premiers pas on a récupéré les données dans des matrices et après nous avons récupéré les gènes étiqueté positif(1) dans la matrice (numpy.array) `gene-rel1` et les gènes non étiqueté(0) dans la matrice `gene-rel0`. Ces relations nous aideront a créer des nouvelles matrices de profils d'expression des relations.

Nous avons trouvé une difficulté pour crée les profils d'expression de `gene-rel0` parce-que la taille de la matrice générée est très grand pour cela on fait recours à des traitements spécifique pour optimiser le temps d'exécution et la taille de la mémoire utilisée.



En suite on a utilisée la méthode *Kmeans* pour rassembler les relations similaires en clusters soit pour les échantillons positif et aussi pour les échantillons non étiqueté ;et après l'extraction des clusters on va mesurer les scores de similarités en utilisant l'information mutuelle. le clustering dans cette étape est utilisé pour minimiser le nombre de comparaison des profils d'expression similaires.



Par la suite pour obtenir l'échantillon des relations qui ont une forte chance d'êtres des négatifs on va comparer les distances entre le barycentre des clusters non encore étiquetées avec ceux des clusters positifs. Les clusters les plus loin avec un certain seuil (la moyenne des distances) sont considérés comme des négatifs.

Après la construction des deux classes (positif et négatif) on peut finalement appliquer la méthode *SVM* pour créée notre modèle de classification qui nécessite la division de notre corpus en deux parties : la partie base d'apprentissage et l'autre pour le test.

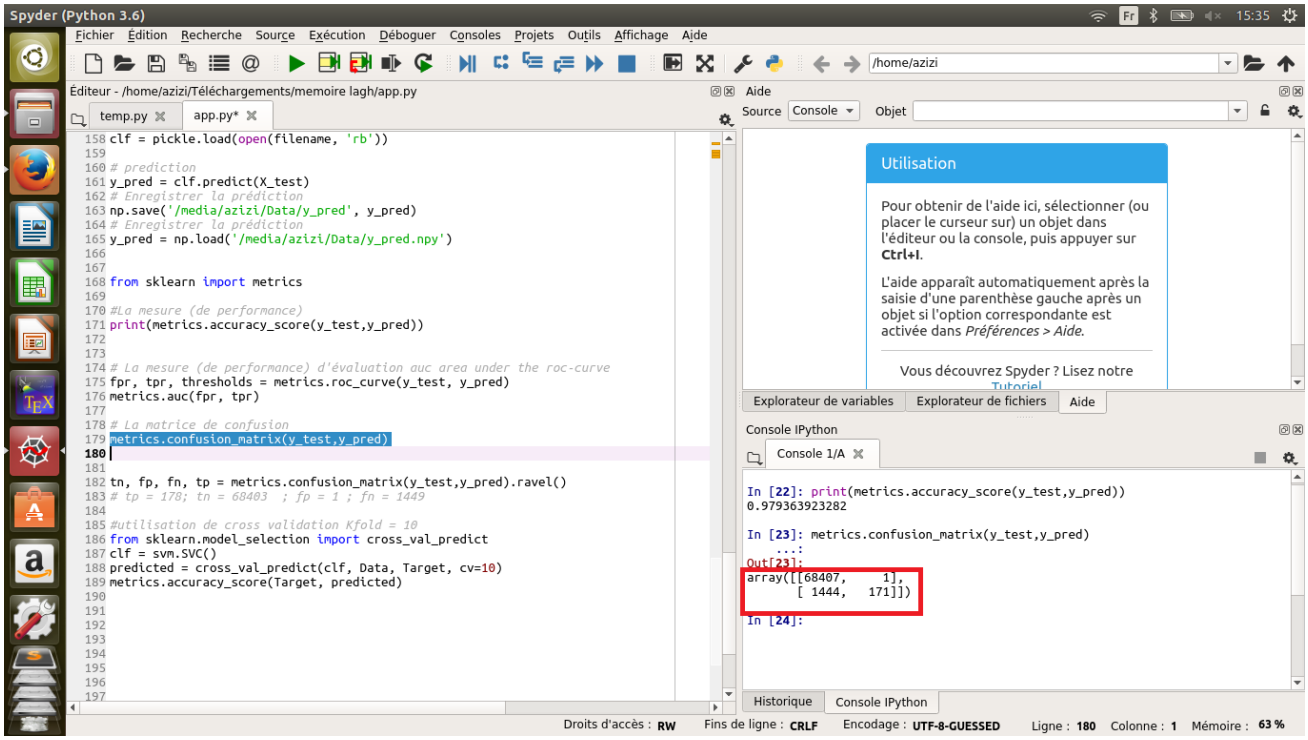
```
from sklearn import metrics

#La mesure (de performance)
print(metrics.accuracy_score(y_test,y_pred))


# La mesure (de performance) d'évaluation auc area under the roc-curve
fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred)
auc = metrics.auc(fpr, tpr)

# La matrice de confusion
metrics.confusion_matrix(y_test,y_pred)


tn, fp, fn, tp = metrics.confusion_matrix(y_test,y_pred).ravel()
# tp = 171; tn = 68407  ; fp = 1 ; fn = 1444

#utilisation de cross validation Kfold = 10
from sklearn.model_selection import cross_val_predict
clf = svm.SVC()
predicted = cross_val_predict(clf, Data, Target, cv=10)
metrics.accuracy_score(Target, predicted)

#------Plot auc
import matplotlib.pyplot as plt
plt.figure()
lw = 2
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('ROC Curve')
plt.legend(loc="lower right")
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.plot(fpr, tpr, color='darkorange',
         lw=lw, label='ROC curve (area = %0.2f)' % auc)
plt.plot([0, 1], [0, 1], color='navy', lw=lw, linestyle='--')
```

```
Utilisation

Pour obtenir de l'aide ici, sélectionner (ou placer le curseur sur) un objet dans l'éditeur ou la console, puis appuyer sur Ctrl+I.

L'aide apparaît automatiquement après la saisie d'une parenthèse

Aide | Profileur | Analyse de code statique

Console IPython
Console 1/A
cosine_similarity
   ...:

In [7]: from sklearn.metrics.pairwise import
manhattan_distances
   ...:

In [8]: from sklearn.model_selection import KFold
   ...:
Traceback (most recent call last):

  File "<ipython-input-8-7748d201c9c6>", line 1, in
<module>
    from sklearn.model_selection import KFold

ImportError: No module named 'sklearn.model_selection'


In [9]: import pickle
   ...:
```

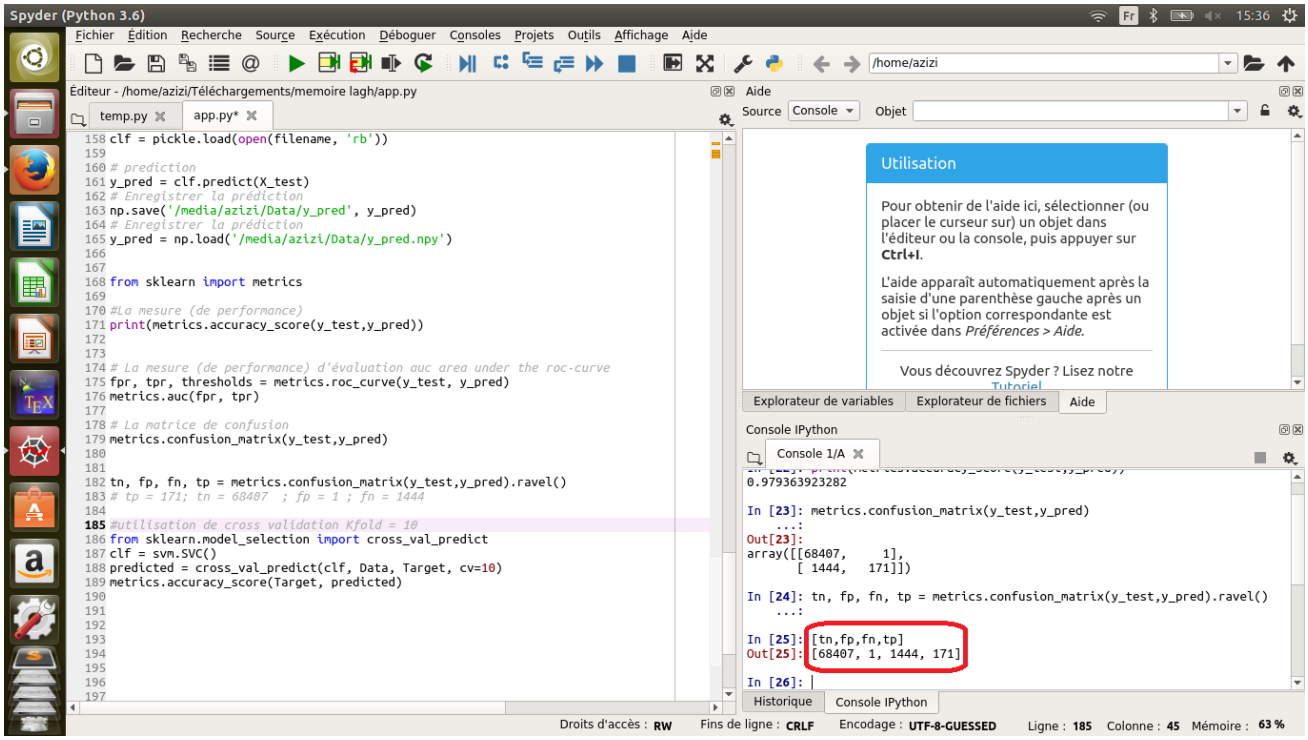
4.6 Présentation des résultats

Ci-dessous les résultats après l'importation du package `metrics` qui nous permet de connaitre les mesures de performances tel que taux de succès=0.979292518173

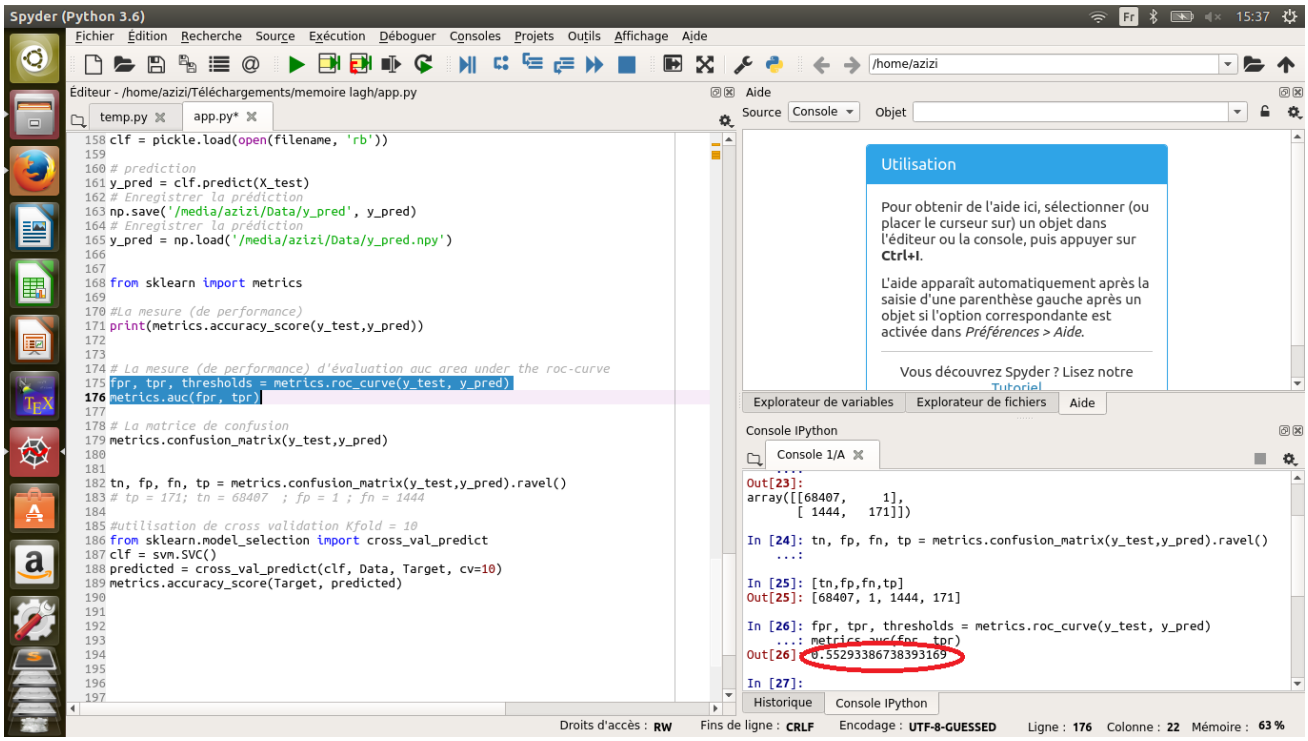
```
Spyder (Python 3.6)
Fichier Édition Recherche Source Exécution Déboguer Consoles Projets Outils Affichage Aide
Éditeur - /home/azizi/Téléchargements/memoire lagh/app.py
temp.py | app.py*
158 clf = pickle.load(open(filename, 'rb'))
159
160 # prediction
161 y_pred = clf.predict(X_test)
162 # Enregistrer la prédiction
163 np.save('/media/azizi/Data/y_pred', y_pred)
164 # Enregistrer la prédiction
165 y_pred = np.load('/media/azizi/Data/y_pred.npy')
166
167
168 from sklearn import metrics
169
170 #La mesure (de performance)
171 print(metrics.accuracy_score(y_test,y_pred))
172 # le taux de reconaissance = 0.979292518173
173
174
175 # La mesure (de performance) d'évaluation auc area under the roc-curve
176 fpr, tpr, thresholds = metrics.roc_curve(y_test, y_pred)
177 metrics.auc(fpr, tpr)
178 # auc = 0.5524942364900016
179
180 # La matrice de confusion
181 metrics.confusion_matrix(y_test,y_pred)
182 #array([[68403,     1],
183 #       [ 1449,   170]])
184
185 tn, fp, fn, tp = metrics.confusion_matrix(y_test,y_pred).ravel()
186 # tp = 170; tn = 68403  ; fp = 1 ; fn = 1449
187
188 #utilisation de cross validation Kfold = 10
189 from sklearn.model_selection import cross_val_predict
190 clf = svm.SVC()
191 predicted = cross_val_predict(clf, Data, Target, cv=10)
192 metrics.accuracy_score(Target, predicted)
193
194
195
196
197

Aide
Source Console Objet
Utilisation
Pour obtenir de l'aide ici, sélectionner (ou placer le curseur sur) un objet dans l'éditeur ou la console, puis appuyer sur Ctrl+I.
L'aide apparaît automatiquement après la saisie d'une parenthèse gauche après un objet si l'option correspondante est activée dans Préférences > Aide.
Vous découvrez Spyder ? Lisez notre Tutoriel
Explorateur de variables | Explorateur de fichiers | Aide
Console IPython
Console 1/A
In [22]: print(metrics.accuracy_score(y_test,y_pred))
0.979363923282
In [23]:
Historique | Console IPython
Droits d'accès : RW  Fins de ligne : CRLF  Encodage : UTF-8-GUESSED  Ligne : 172  Colonne : 1  Mémoire : 63 %
```



Les chiffres 68407 et 1 et 1444,171 sont les éléments de la matrice de confusion : les lignes représente les observations et les colonnes représente les prédictions

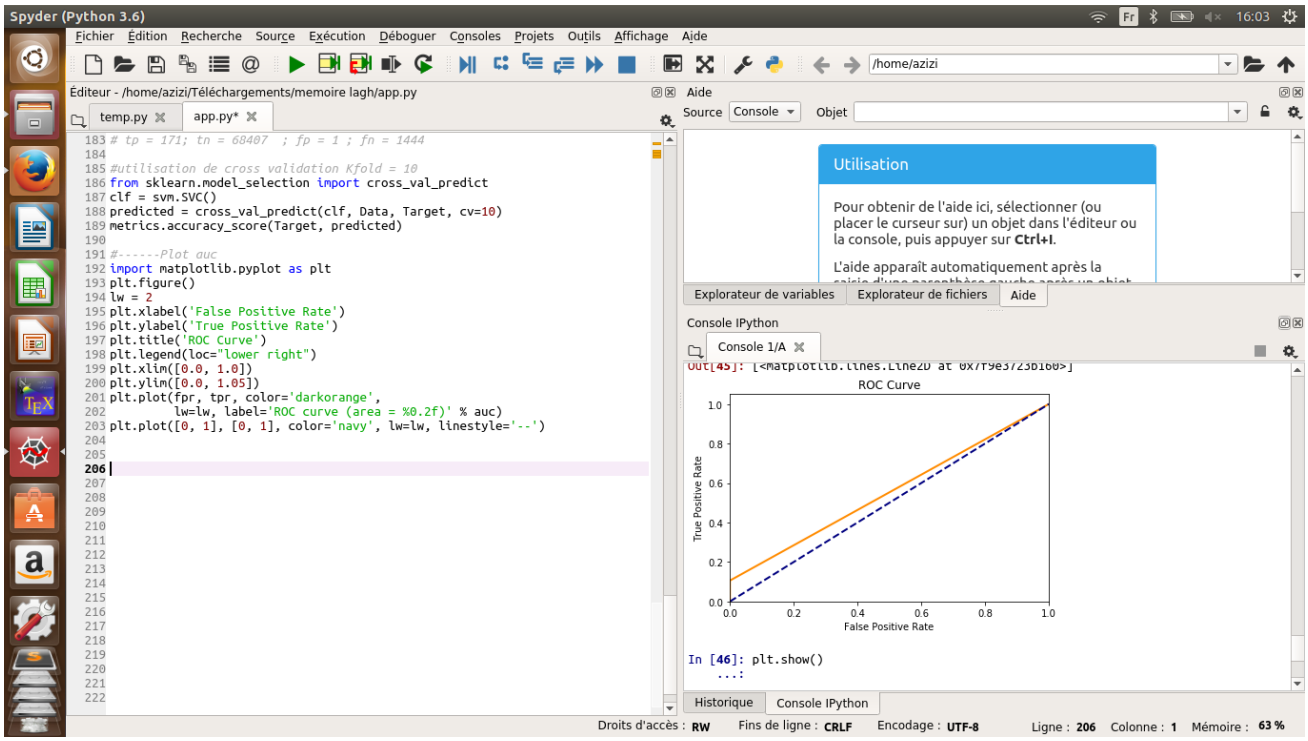


La matrice de confusion présente à cet ordre :tn signifier la true négatif=68407 ;fp signifier : false positif=1 ;fn signifier : false négatif=1444 ;tp signifier : true positif=171. Le chiffre



sélectionner en rouge présenté le mésure de performance Auc=0.55249423464900016

Ci-dessous le graphique de courbe roc à partir des true positif(tp) et false positif(fp)



Conclusion

Le but de ce chapitre est la réalisation d'un modèle informatique à traité les données biologique sous forme d'une puce ADN pour inférer un réseau de régulation génétique à partir a ces données. On a essayé de faire un programme informatique sous python pour traité ces données qui stocker en puce ADN. En premier temps, nous avons proposé une nouvelle méthode pour la sélection des négatifs fiable à partir des données non étiquetées.Nous avons commencé à utiliser les outils nécessaires , à savoir :la puce ADN,le langage Python. Par la suite on a appliquer l'algorithme*Kmeans* pour extraire des clusters pour obtenir les gènes susceptibles d'êtres des négatifs, après la récupération des données dans des matrices ; Nous avons appliqué l'algorithmes *SVM*, puis on a mesuré les performances de cet algorithmes par la mesure de performance AUROC.

Conclusion générale et perspectives

Notre travail s'inscrit dans le cadre du l'utilisation de l'informatique à une domaine biologique que s'appel la bioinformatique et plus précisément l'application des méthodes d'apprentissage automatique pour inférer un réseau de régulation génétique.

Pour mener à bien ce rapport nous avons commencé par une étude bibliographique en deux premiers chapitres et un état de l'art dans le troisième chapitre. Un premier chapitre les notions biologique de base.Dans ce chapitre en a essayé à comprendre les différentes notions et définitions de ce nouveau domaine(cellule,gène,acide aminée,chromosome...etc),et présente dans ce chapitre Le dogme central comporte généralement trois étapes : la réplication,la transcription et la traduction.Nous avons vu aussi le domaine bioinformatique et les axes de recherche dans ce domaine.

Au cours du deuxième chapitre nous avons vu le terme de Data Mining en français(la fouille de données) et plus particulièrement la classification (catégorisation) durant lequel on a met l'accent sur les méthodes utilisées que ce soit supervisées (nécessitent une connaissance a priori dite apprentissage) ou non supervisées ou semi-supervisée.

Dans le troisième chapitre nous avons vus quelque travaux étudiés et applicable dans ce cadre de recherche sous forme des algorithmes d'apprentissage automatique pour inférer des réseaux de régulations génétique qui permet de distinguer en trois axes :la première approche basé sur les méthodes d'apprentissage automatique non supervisée tel que ARANCE,MRNET,MRNET-b ;la deuxième s'intéresse aux méthodes d'apprentissage semi-supervisée tel que :PSoL,bagging SVM et Spy-SVM et la troisième c'est les méthodes supervisée tel que la méthode SIRENE.

Le dernier chapitre a été consacré à notre contribution.En premier temps, nous avons proposé une nouvelle méthode pour la sélection des négatifs fiable à partir des données non étiquetées.Nous avons commencé à utiliser les outils nécessaires , à savoir :la puce ADN,le langage Python. Par la suite on a appliquer l'algorithme*Kmeans* pour extraire des clusters pour obtenir les gènes susceptibles d'êtres des négatifs, après la récupération des données dans des matrices ; Nous avons appliqué l'algorithmes *SVM*, puis on a mesuré les performances de cet algorithmes par la mesure de performance AUROC.

La difficulté majeure dans les méthodes d'apprentissage automatique en générale et spécialement dans ce travail est l'énorme besoin en terme de temps (apprentissage) qui nécessite parfois une journée entière, et en terme de mémoire puisque nous traitons des données avec des tailles

importantes par exemple la taille de la matrice `array0` est 274380 x 1610 qui contient des réels (chacun sur 4 octets) c à d 1.65 Go de mémoire RAM juste pour cette matrice, et comme notre PC ne dispose que 4 GO, à chaque fois où nous lançons l'apprentissage pour extraire les clusters nous recevons le message de la mémoire insuffisante.

Comme perspectives on pourrait penser à améliorer notre programme pour qu'il soit plus optimal en terme de temps d'exécution et de la mémoire utilisée .

On pourrait aussi en extraire d'autres problématiques tel que la recherche d'un seuil optimal pour choisir un cluster négatif qui donne plus de performances.

Envisager la réutilisation de cette méthode qui pourrait être appliquée sur d'autres données non biologique pour la classification dite apprentissage positif et non étiqueté (PU learning) tel que la classification de textes, web, ...etc.

Bibliographie

[1] biologie-transcription. Doc Intrnet.

[2] Cellul. Internet.

[3] cellule-definition. Doc Intrnet.

[4] Chromosom. Internet.

[5] Chromosome. Internet.

[6] classification. Doc Internet.

[7] Cross-validation,. Internet.

[8] Dogome central. Internet.

[9] Python. Internet.

[10] Traduction. Internet.

[11] Didacticiel - etudes de cas, 2016. Doc Internet.

[12] K. Basso C. Wiggins G. Stolovitzky R. Dalla Favera A. Califano A. A. Margolin, I. Nemenman. Aracne : an algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context.

[13] Marc Bailly-Bechet. Quelques problèmes de bioinformatique„ 2009-2010.

[14] al Bleakley K. Supervised reconstruction of biological networks with local models.

[15] Abergel C Claverie J.-M, Audic S. La bioinformatique : une discipline stratégique pour l'analyse et la valorisation des génomes, 1999.

[16] S.Stern Clearwater. A rulelearning program in high energy physics event classification„ 1991.

[17] Mohamed Sayed Hassan Damien Imbs. Bioinformatique.

[18] DJEFFAL Dr. Abdelhamid. Cours fouille de données avancée, 2015. cours Master 2 IDM.

[19] Jean-Philippe Vert Fantine Mordelet. Sirene : supervised inference of regulatory networks.

[20] Povost.F Fawcett.T. Adaptative fraud detection, volume 1 of data mining and knowledge discovery, 1997.

[21] Povost.F Fawcett.T. Robust classification systems for imprecise environments, 2001.

[22] Rioul François. Code génétique - définition.

[23] Michael Guilloux. Chroniqueur actualités.

[24] Anne-Claire HAURY. *Sélection de variables à partir de données d'expression.* Theses, l'École nationale supérieure des mines de Paris, décembre 2012.

[25] Tropical Hygiene. Genome resource facility grf.

[26] Quinkal Isabelle. Quelques termes-clef de biologie moléculaire et leur définition,, 2003.

[27] J.T. Thaden I. Mogno J. Wierzbowski G. Cottarel S. Kasif J.J. Collins J.J. Faith, B. Hayete and T.S. Gardner. Large-scale mapping and validation of escherichia coli transcriptional regulation from a compendium of expression profiles.

[28] Matwin.S Kubat.M, Holte.R. Machine learnong for the detection of oils spills in satellite radar images.

[29] Bourgeade Laetitia. *Inférence des acteurs de la régulation des expressions géniques.* PhD thesis, l'Université de Bordeaux.

[30] Nolwenn LE MEUR. *De l'Acquisition des Données de Puces à ADN vers leur Interprétation : Importance du Traitement des Données Primaires.* PhD thesis, UNIVERSITE DE NANTES, FACULTE DE MEDECINE.

[31] Lewis.D. Evaluating text categorization. in proc. workshop on speech and natural language.

[32] Lewis.D. Representation quality in text classification :an introduction and experiment. in proc. workshop on speech and natural language, 1990.

[33] Pietro Zoppoli Michele Ceccarelli Luigi Cerulo, Vincenzo Paduano. A negative selection heuristic to predict new transcriptional targets.

[34] AGIER. M. De l'analyse de données d'expression à la reconstruction de réseaux de gènes.

[35] Hélène Touzet Maude Pupin, Laurent Noé. Notions de biologie.

[36] Métais E Nakache D. Evaluation :nouvelle approche avec juges, 2005.

[37] Sushmita Roy Manolis Kellis Patrick E. Meyer, Daniel Marbach. Information-theoretic inference of gene networks using backward elimination.

[38] Neri.F Saitta.L. Learning in the "real world".

[39] Melissa J. Davis Stefan R. Maetschke, Piyush B. Madhamshettiwar and Mark A. Ragan. Supervised, semi-supervised and unsupervised inference of gene regulatory networks.

[40] Ziani Soheyb Tabet aoul Walid Houcine. Clustering hiérarchique de données à base de ward, 2013.

[41] Vandewalle Vincent. Les modèles de mélange, un outil utile pour la classification semi-supervisee.

[42] V.Vapnik. The nature of statistical learning theory., 1995. Doc Internet.