الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جامعة الإخوة منتوري قسنطينة 1
Frères Mentouri Constantine 1 University
Université Frères Mentouri Constantine 1

Université Frères Mentouri Constantine
Faculté des Sciences de la Nature et de la Vie
Département de Biologie appliquée

جامعة الاخوة منتوري قسنطينة
كلية علوم الطبيعة والحياة
قسم البيولوجيا التطبيقية

Mémoire présenté en vue de l'obtention du diplôme de Master

Domaine : Sciences de la Nature et de la Vie
Filière : Sciences biologiques
Spécialité : Bioinformatique
N° d'ordre :
N° de série :

Intitulé :

Approche DL pour l'identification des classes d'ARN codants et non codants

Présenté par : Nouha BENMENIA
Khaoula SMAILI

Le 19/06/2023

Jury d'évaluation :

Président de jury : Pr. Ines BELLIL
Encadreur : Dr. Khaled BOULAHROUF
Examinateur : Dr. Amira GHERBOUDJ

Année universitaire
2022 – 2023

REMERCIEMENT

Nos sincères remerciements s'adressent à notre encadreur, Monsieur Khaled BOULAHROUF, Maître de Conférences au département de Microbiologie de l'Université Frères Mentouri de Constantine 1, pour avoir dirigé ce travail.

Nous tenons à exprimer notre gratitude pour sa compréhension, sa disponibilité, sa patience et sa générosité scientifique. Ses conseils précieux et ses encouragements tout au long de ce mémoire ont été d'une grande aide.

Nous souhaitons également remercier chaleureusement Madame Ines BELIL, Professeur au département de Biologie Appliquée de l'Université Frères Mentouri de Constantine 1, pour l'honneur qu'elle nous fait en présidant le jury de ce mémoire. Nous adressons nos remerciements à l'examinateur de ce mémoire, Madame Amira Gherboudj, Maître de Conférences de catégorie A au département de Biologie Appliquée de l'Université Frères Mentouri de Constantine 1. Nous sommes ravis de votre participation au jury de ce mémoire.

Nous aimerions également exprimer notre profonde gratitude envers Monsieur Salah ALIOUANE, doctorant en Bioinformatique, pour son immense aide, sa confiance, sa générosité et son soutien tout au long de notre travail.

Enfin, nous tenons à remercier tous ceux qui nous ont apporté leur soutien, leurs conseils techniques et leur expertise lors de cette expérience. Votre présence à nos côtés a été inestimable et nous vous exprimons notre reconnaissance la plus sincère.

Résumé

Ce mémoire présente une approche basée sur l'apprentissage profond (*Deep Learning*) pour l'identification des classes d'ARN codants et non codants. L'ARN est une molécule essentielle dans la biologie, et sa classification en ARN codants (ARNm) et ARN non codants est cruciale pour comprendre les mécanismes de régulation génétique. Dans ce travail, nous avons utilisé deux ensembles de données : un ensemble d'ARN non codants provenant de la base de données Rfam, et un ensemble d'ARN codants provenant de la base de données RefSeq. Les données ont été prétraitées pour les rendre compatibles avec les algorithmes d'apprentissage automatique, notamment en utilisant l'encodage one-hot, le *padding* et le *tokenizer*. Ensuite, un modèle d'apprentissage en Deep Learning a été construit en utilisant une architecture de réseau de neurones convolutionnels (*CNN*) avec des couches d'*embedding*, de convolution, de max pooling et de classification. Le modèle a été entraîné sur les données d'ARN codants et non codants, en utilisant une répartition des données avec 90% pour l'apprentissage et 10% pour les tests. Les performances du modèle ont été évaluées en utilisant la mesure de précision (*accuracy*), qui représente le pourcentage de prédictions correctes par rapport au nombre total d'instances. Les résultats obtenus ont démontré une précision élevée sur les données d'entraînement (99,99%) et sur les données de test (99,87%), ce qui indique une performance prometteuse du modèle dans l'identification des classes d'ARN codants et non codants. En conclusion, cette approche basée sur l'apprentissage profond offre une méthode efficace pour l'identification des classes d'ARN codants et non codants. Les résultats obtenus suggèrent que cette approche pourrait être utilisée dans des études plus larges et pour l'analyse de grandes quantités de données d'ARN, contribuant ainsi à une meilleure compréhension des mécanismes de régulation génétique.

Mots clés : ARN ; Classification ; Apprentissage profond ; Réseau de neurones convolutionnels, Encodage one-hot.

Abstract

This thesis presents a deep learning-based approach for the identification of coding and non-coding RNA classes. RNA is an essential molecule in biology, and its classification into coding (mRNA) and non-coding RNA is crucial for understanding genetic regulatory mechanisms. Two datasets were used in this study: a dataset of non-coding RNAs from the Rfam database, and a dataset of coding RNAs from the RefSeq database. The data was preprocessed to make it compatible with machine learning algorithms, including one-hot encoding, padding, and tokenization. Subsequently, a deep learning model was constructed using a convolutional neural network (CNN) architecture with embedding, convolution, max pooling, and classification layers. The model was trained on the coding and non-coding RNA data, using a data split of 90% for training and 10% for testing. The model's performance was evaluated using the accuracy metric, which represents the percentage of correct predictions compared to the total number of instances. The results demonstrated high accuracy on the training data (99.99%) and testing data (99.87%), indicating promising performance of the model in identifying coding and non-coding RNA classes.In conclusion, this deep learning-based approach offers an effective method for identifying coding and non-coding RNA classes. The results suggest that this approach could be utilized in larger studies and for analyzing large amounts of RNA data, contributing to a better understanding of genetic regulatory mechanisms.

Keywords: RNA; Classification; Deep Learning; Convolutional Neural Networks; One-hot Encoding.

ملخص

تقدم هذه الرسالة نهجًا يعتمد على التعلم العميق لتحديد فئات الحمض الريبي النووي المشفر وغير المشفر. الحمض الريبي هو جزيء أساسي في علم الأحياء، وتصنيفه إلى حمض الريبي المشفر وغير المشفر ضروري لفهم آليات التنظيم الوراثي. في هذا العمل، تم استخدام مجموعتي بيانات: مجموعة الحمض الريبي غير المشفر من قاعدة بيانات Rfam ، ومجموعة الحمض الريبي المشفر من قاعدة بيانات RefSeq . تم معالجة البيانات لتكون متوافقة مع خوارزميات التعلم الآلي، بما في ذلك الترميز الثنائي، والتعبئة والرمز. ثم تم بناء نموذج تعلم عميق باستخدام هندسة الشبكات العصبية التركيبية مع طبقات التضمين، والتحويل، والتجميع الأقصى، والتصنيف. تم تدريب النموذج على بيانات الحمض الريبي المشفر وغير المشفر باستخدام توزيع البيانات بنسبة 90٪ للتدريب و10٪ للاختبار. تم تقييم أداء النموذج باستخدام قياس الدقة الذي يمثل نسبة التنبؤات الصحيحة مقارنةً بإجمالي عدد الحالات. أظهرت النتائج المحصلة دقة عالية على بيانات التدريب (99.99٪) وعلى بيانات الاختبار (99.87٪)، مما يشير إلى أداء واعد للنموذج في تحديد فئات الحمض الريبي المشفر وغير المشفر. في الختام، تقدم هذه النهج المعتمد على التعلم العميق طريقة فعالة لتحديد فئات الحمض الريبي المشفر وغير المشفر. تشير النتائج المحصلة إلى أنه يمكن استخدام هذا النهج في دراسات أوسع وتحليل كميات كبيرة من بيانات الحمض الريبي، مما يسهم في فهم أفضل لآليات التنظيم الوراثي.

الكلمات الرئيسية: الحمض الريبي النووي؛ تصنيف؛ التعلم العميق؛ الشبكات العصبية التركيبية الانطباعية؛ الترميز الثنائي.

LISTE DES FIGURES

LISTE DES TABLEAUX

LISTE DES ABREVIATIONS

ADN :	Acide Désoxyribonucléique
ARN :	Acide Ribonucléique
ARNm :	Acide Ribonucléique messager
lncRNA :	*Long non-coding RNA* (ARN non-codant long)
ARNt :	Acide Ribonucléique de transfert
ARNr :	Acide Ribonucléique ribosomique
sncRNA :	*Small non-coding RNA* (ARN non codant court)
miARN :	*MicroARN*
snoRNA :	*Small nucleolar RNA* (ARN nucléolaire petit)
piARN :	*Piwi-interacting RNA* (ARN interagissant avec Piwi)
RT-PCR :	*Reverse Transcription Polymerase Chain Reaction* (Réaction en chaîne par polymérase avec transcription inverse)
ADNc :	Acide Désoxyribonucléique complémentaire
RNA-Seq :	*RNA sequencing* (séquençage d'ARN)
FISH :	*Fluorescence in situ Hybridization* (Hybridation in situ en fluorescence)
CNN :	*Convolutional Neural Networks* (Réseaux de neurones convolutionnels)
RBM :	*Restricted Boltzmann Machine*
LSTM :	*Long Short-Term Memory* (Mémoire à Long Terme et Court Terme)
RNN :	*Recurrent Neural Networks* (Réseaux de neurones récurrents)
DL :	*Deep Learning* (Apprentissage profond)
ECG :	Electrocardiogramme
EEG :	Electroencéphalogramme
Acc :	Accuracy (précision)
DNN :	*Deep Neural Network* (Réseaux de neurones profonds)

TABLE DE MATIERES

INTRODUCTION

INTRODUCTION

La classification est une tâche essentielle dans de nombreux domaines, et la biologie ne fait pas exception. Dans le domaine de l'ARN, la classification est un enjeu majeur pour comprendre les différentes fonctions de ces molécules clés de la vie cellulaire [1]. Les ARN peuvent être classés en deux grandes catégories : les ARN codants, qui sont traduits en protéines, et les ARN non codants, qui ont des fonctions régulatrices et structurales importantes [2]. La classification automatique des ARN en ARN codants et non codants est un défi important dans le domaine de la biologie moléculaire, qui peut être résolu en utilisant des techniques de reconnaissance de formes et d'apprentissage approfondi [3].

La classification automatique des ARN est un domaine de recherche en constante évolution, qui peut contribuer à la compréhension des mécanismes biologiques des ARN [4]. La classification des ARN peut également aider à identifier de nouvelles cibles thérapeutiques pour le traitement de maladies liées à des anomalies dans les ARN [5].

Dans cette étude, nous allons nous intéresser à la classification des ARN en ARN codants et non codants en utilisant des techniques de reconnaissance de formes et d'apprentissage approfondi. Nous allons explorer différentes méthodes de classification, en nous concentrant en particulier sur les réseaux de neurones, une méthode qui s'est avérée performante pour la classification d'images et de données de séquences [6].

Notre travail s'inscrit dans le cadre de la classification des ARN en utilisant le Deep Learning, on vise à classifier les ARN en deux grandes classes : les ARN codants et les ARN non codants. Le résultat de la classification sera exploité pour comprendre la fonction biologique de chaque type d'ARN et pour aider à la découverte de nouveaux ARN ayant des rôles importants dans les processus biologiques. Ce travail a comme objectif de simplifier la tâche aux chercheurs du domaine, ainsi faciliter la compréhension de la fonction des différents types d'ARN.

Le plan du mémoire est le suivant : la première partie se compose de trois chapitres qui traitent de l'état de l'art dans le domaine de l'ARN. Le premier chapitre aborde l'ARN en détail, en mettant l'accent sur sa structure, sa fonction et son rôle dans la cellule [7]. Le deuxième chapitre se concentre sur les classes d'ARN. Nous explorerons les principales catégories d'ARN, en examinant leurs caractéristiques distinctives et leurs fonctions biologiques spécifiques [8]. Le troisième chapitre est consacré à l'apprentissage approfondi, une méthode

d'apprentissage automatique qui a connu des avancées majeures dans le domaine de la biologie. Nous examinerons les principes fondamentaux de l'apprentissage approfondi et ses applications spécifiques dans l'étude et la classification de l'ARN [4].

La deuxième partie de ce mémoire fournira une vue d'ensemble complète de la recherche en présentant le matériel utilisé, notamment les données biologiques, la machine et les outils informatiques. Les méthodes appliquées seront également détaillées, en mettant en évidence les étapes du prétraitement et de l'apprentissage. De plus, les résultats obtenus à partir du modèle construit seront exposés. Enfin, la discussion des résultats permettra d'analyser ces derniers à la lumière des objectifs de recherche. Les résultats seront interprétés et comparés avec d'autres études pertinentes qui seront également effectuées afin de mettre en évidence les points forts, les limitations et les perspectives futures de la recherche.

PARTIE 01 : RECHERCHE BIBLIOGRAPHIQUE

CHAPITRE 01 : L'ARN

L’ARN

1. Définition de l’ARN

Au cours de notre apprentissage en génétique et en biologie moléculaire, nous avons été initiés à la notion fondamentale de l'ADN, l'acide désoxyribonucléique, en tant que molécule de base de la vie. Doté d'un code génétique qui porte l'information dans ses séquences, l'ADN renferme les instructions nécessaires à la survie de tous les êtres vivants [9].

Cependant, malgré son rôle important, l'ADN ne fonctionne pas en solitaire. Pour que l'ADN puisse exercer son pouvoir de régulation sur tous les aspects physiques et fonctionnels de la vie, un système extrêmement efficace travaille en coulisses, orchestré par une autre molécule d'acide nucléique : l'ARN (acide ribonucléique).

Dans cette partie, nous nous intéressons à la définition de l'ARN, ses caractéristiques et son rôle fondamental dans la cellule.

L'ARN est un long polymère non ramifié de fragments monophosphate de ribonucléoside reliés entre eux par des liaisons phosphodiester **(figure 1)**. Les ARN eucaryotes et procaryotes sont essentiellement des molécules simples brin. Les blocs de construction de base non assemblés de l'ARN (et de l'ADN) sont appelés nucléotides. Ces blocs de construction sont constitués de trois composants clés : un pentose (sucre à 5 carbones ; ribose dans le cas de l'ARN, désoxyribose dans le cas de l'ADN), au moins un groupe phosphate et une base azotée. Une base azotée liée à un sucre pentose est appelée nucléoside. Lorsqu'un groupe phosphate est ajouté, le composite, un ester de phosphate du nucléoside, est appelé nucléotide **(figure 2)** [9].

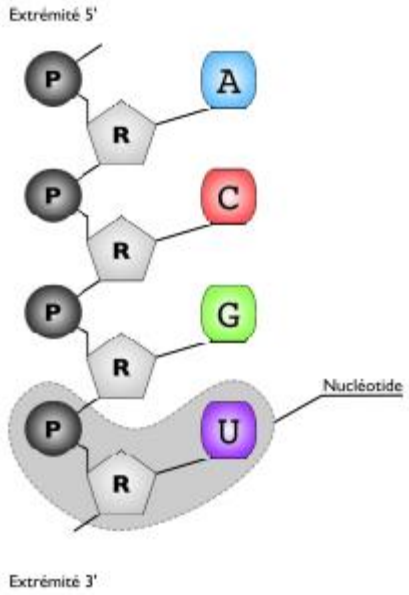


Figure 01. Structure d'un brin d'ARN : Chaque groupe phosphate (P) est lié à un sucre (R) qui est à son tour lié à une base azotée (A, C, G, U).

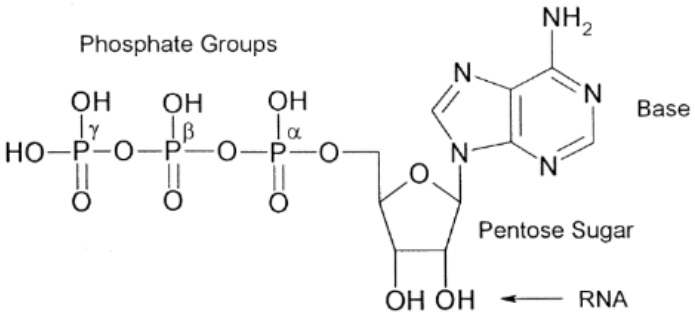


Figure 02. Un bloc d'ARN

Règles de complémentarité des bases azotées dans l'ARN selon la règle de *Watson-Crick :*

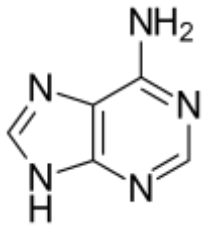
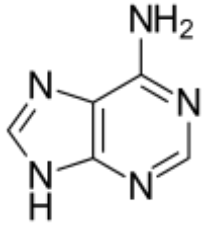
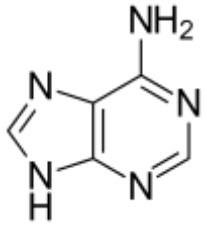
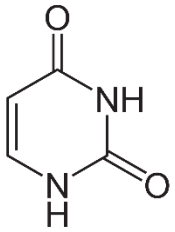
Les appariements de bases azotées dans l'ARN **(tableau 01)** suivent la règle de Watson-Crick, qui se décompose comme suit :

- La base adénine (A) s'associe à l'uracile (U) par deux liaisons hydrogène (A = U).
- La base guanine (G) s'associe à la cytosine (C) par trois liaisons hydrogène (G ≡ C).

Ces appariements faibles permettent à l'ARN de s'apparier avec lui-même, formant des structures en épingle à cheveux, ou de former des liaisons avec d'autres ARN, voire avec

l'ADN [10].

Tableau 01. Les 4 bases azotées de l'ARN [11]

La base azotée	Formule	Structure	Groupes
Adénine	$C_5H_5N_5$		purines
Guanine	$C_5H_5N_5O$		
Cytosine	$C_4H_5N_3O$		pyrimidines
Uracile	$C_4H_4N_2O_2$		

2. Rôle de l'ARN

L'acide ribonucléique (ARN) joue un rôle fondamental au sein de la cellule. Il est impliqué dans une grande diversité de fonctions essentielles à la vie. L'une de ses principales fonctions est d'agir comme un messager, permettant la transmission de l'information génétique contenue dans l'ADN vers les ribosomes, où elle est traduite en protéines fonctionnelles. Cependant, l'ARN ne se limite pas uniquement à ce rôle de messager. Il est également impliqué dans la régulation de l'expression des gènes, la réplication et la réparation de l'ADN, ainsi que dans d'autres processus cellulaires importants. [12]

3. Caractéristiques de l'ARN

3.1.La synthèse de l'ARN

L'ARN est produit par le processus de transcription. La transcription est le mécanisme par lequel une séquence d'ARN est synthétisée en copiant la séquence d'un gène spécifique. Ce processus se décompose généralement en trois étapes : l'initiation, l'élongation et la terminaison, qui sont schématisées dans la **figure 03**.

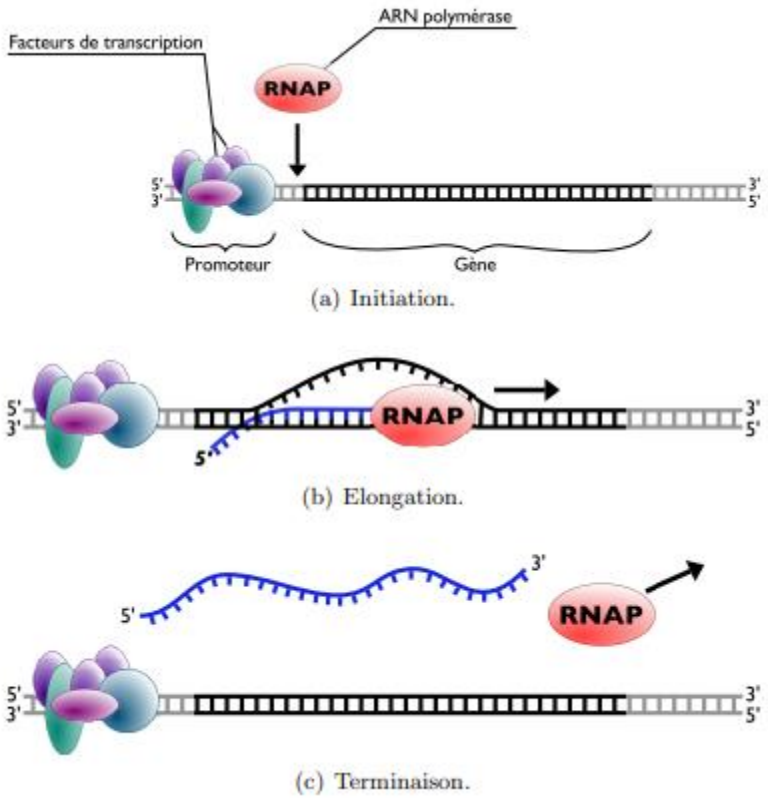


Figure 03. Les étapes de la transcription

3.1.1 Initiation

Durant cette phase, l'ARN polymérase, un complexe protéique, se fixe sur une région spécifique de l'ADN appelée site promoteur, située en amont du gène à transcrire. La liaison entre l'ADN et l'ARN polymérase permet l'ouverture de la double hélice et catalyse l'insertion des ribonucléotides pour former un brin d'ARN.

Chez les procaryotes, le site promoteur contient la boîte de *PRIBNOW*, marquant le début de la transcription. En revanche, chez les eucaryotes et les archées, l'équivalent de la boîte de *PRIBNOW* est la boîte TATA. Ces boîtes, ainsi que d'autres éléments régulateurs tels que la boîte CAAT, interviennent dans la régulation de la transcription.

3.1.2 Elongation

Au cours de cette étape, l'ARN polymérase progresse séquentiellement sur le brin d'ADN codant, de l'extrémité 3' vers l'extrémité 5', qui correspond au brin complémentaire du brin contenant la séquence à transcrire. Les ribonucléotides sont incorporés de manière complémentaire, produisant ainsi une copie conforme de la région à transcrire.

3.1.3 Terminaison

La transcription se termine lorsque l'ARN polymérase rencontre une séquence spécifique appelée terminateur. Chez les procaryotes, le terminateur est généralement une région riche en guanine (G) et cytosine (C) qui forme une structure en tige-boucle, suivie d'une série d'adénines (A) sur l'ADN. Les mécanismes de terminaison de la transcription chez les eucaryotes sont moins connus [13].

Après la formation de l'ARN, commence le processus de maturation de l'ARN, qui fait référence aux modifications post-transcriptionnelles qui se produisent sur l'ARN nouvellement synthétisé. Ces modifications sont essentielles pour la stabilité, la fonction et la régulation de l'ARN.

CHAPITRE 02 : LES CLASSES D'ARN

LES CLASSES D'ARN

La variation dans la formation des séquences d'ARN codants et non codants est principalement due à des variations génomiques qui se produisent pendant la maturation de l'ARN [14]. Lors de la maturation, les ARN pré-messagers subissent des processus complexes, tels que l'épissage [15], la polyadénylation et la modification chimique, qui déterminent finalement si un ARN pré-messager donnera naissance à un ARN messager (ARNm) codant une protéine ou à un ARN non codant.

Les ARNm sont des ARN pré-messagers qui ont subi des modifications spécifiques et qui contiennent des régions codantes qui peuvent être traduites en séquences d'acides aminés, formant ainsi les protéines. Cependant, les ARN non codants sont des ARN pré-messagers qui ne subissent pas les modifications nécessaires pour être traduits en protéines, et ils peuvent remplir diverses fonctions régulatrices ou structurales au sein de la cellule.

Les variations génomiques qui se produisent au niveau des sites d'épissage, des sites de polyadénylation ou d'autres régions régulatrices peuvent influencer la formation des séquences d'ARN codants et non codants, conduisant ainsi à une diversité fonctionnelle dans le transcriptome. Ces variations génomiques contribuent à la complexité et à la régulation précise de l'expression des gènes dans les organismes vivants [16].

Les ARN se divisent en deux grandes classes, qui sont catégorisées selon leurs fonctions : les ARN codants (ARNm) et les ARN non codants [17].

1. ARN codant (ARNm)

Les ARN codants, également appelés ARN messagers (ARNm), sont une classe spécifique d'ARN qui code pour une protéine. Après la transcription de l'ADN génomique, le pré-ARNm est synthétisé et subit un processus d'épissage, au cours duquel les introns sont éliminés. L'ARNm mature est ensuite transporté dans le cytoplasme, où il subit la traduction en protéine. Les ARNm ne représentent qu'une petite fraction, soit environ 2,3 %, du génome [18].

L'ARNm est une molécule d'acide ribonucléique (ARN) qui joue un rôle d'intermédiaire dans la synthèse des protéines. Il s'agit d'une copie temporaire d'une partie de l'ADN correspondant à un ou plusieurs gènes. L'ARNm est utilisé par les cellules comme modèle pour la production de protéines. Le concept d'ARNm a été proposé et démontré par Jacques Monod et François Jacob en 1960 [19].

Chez les eucaryotes, l'ARNm est généralement monocistronique, ce qui signifie qu'il correspond à un seul gène et code une seule protéine. Il est composé d'une séquence codante, qui contient les codons pour les acides aminés, encadrée par des régions non codantes telles que les régions 5'UTR (non traduite en protéine) et 3'UTR. L'ARNm eucaryote subit également des processus de maturation, tels que l'ajout d'une coiffe en 5' et d'une queue poly-A en 3', ainsi que l'épissage des introns pour éliminer les séquences non codantes.

En revanche, chez les procaryotes, l'ARNm est souvent polycistronique, ce qui signifie qu'il peut coder plusieurs protéines. Dans ce cas, l'ARNm contient des régions codantes successives pour différentes protéines, séparées par des séquences non codantes appelées régions intergéniques. Les gènes adjacents dans l'ARNm procaryote sont traduits séquentiellement pour produire les différentes protéines. Dans ces organismes qui n'ont pas de noyau, la transcription des ARNm et leur traduction en protéines sont généralement couplées [20].

2. ARN non-codant

Les ARN non codants forment une classe diversifiée d'ARN qui jouent un rôle dans la régulation de l'expression génétique. Ils se composent d'un grand nombre d'ARN différents, chacun ayant des fonctions spécifiques. Leur taille peut varier considérablement. Certains sont relativement courts, appelés petits ARN non codants (10-200 nucléotides), tandis que d'autres peuvent être beaucoup plus longs, connus sous le nom de longs ARN non codants (des milliers de nucléotides). Cette diversité de tailles reflète la diversité des mécanismes d'action.

Les ARN non codants se localisent dans le génome de différentes manières. Ils peuvent être présents sur le brin sens ou antisens, chevauchant ou non les exons d'un gène transcrit, et se trouver dans les introns ou les régions intergéniques (entre deux gènes codants) [21].

La classification des ARN non codants se fait en fonction de leur taille. Ils se divisent en deux grandes classes :

2.1. Long ARN non codant (*lncRNA*)

Les ARN longs non codants (*lncRNA*) sont une classe d'ARN qui ne code pas directement de protéines mais ont une fonction régulatrice dans la cellule. Ils ont une taille supérieure à 200 nucléotides et leur localisation peut varier en fonction de leur fonction, se trouvant soit dans le noyau, soit dans le cytoplasme. Il existe différents types d'ARN longs

non codants dans cette classe, dont deux exemples importants sont les ARN ribosomaux (ARNr) et les ARN de transfert (ARNt) [22].

2.1.1. ARNt

Les ARN de transfert (ARNt) sont des molécules d'ARN présentes dans les cellules et jouent un rôle essentiel dans la synthèse des protéines. Chaque cellule contient différentes espèces d'ARNt qui se distinguent chromatographiquement et ont des longueurs variables de 76 à 90 nucléotides. Environ 400 ARNt ont été identifiés, dont environ 200 ont été purifiés à partir d'organismes eucaryotes. Environ 50 de ces ARNt sont d'origine organellaire tandis que les autres sont codés dans le noyau.

Les ARNt présentent une structure et des fonctions conservées chez les bactéries (procaryotes) et les eucaryotes, à l'exception des ARNt mitochondriaux qui ont des caractéristiques structurelles différentes en raison de l'utilisation d'un code génétique variant. La régulation de la transcription des gènes d'ARNt chez les eucaryotes est étudiée afin de comprendre comment la cellule optimise la traduction de l'ARNm pour permettre la synthèse des protéines.

La biosynthèse de l'ARNt est essentielle pour assurer la production d'ARNt correct en quantités suffisantes pour correspondre aux codons de l'ARNm traduit. La composition en codons de l'ARNm influence l'efficacité de la traduction et la vitesse d'élongation du polypeptide [23].

2.1.2. ARNr

Les ARN ribosomiques (ARNr) sont des molécules d'ARN présentes dans les ribosomes, qui sont les structures responsables de la traduction des ARN messagers (ARNm) en protéines. Les ARNr jouent un rôle essentiel dans cette activité. Chez les eucaryotes, la petite sous-unité 40S du ribosome est composée d'un ARNr spécifique appelé 18S, associé à 33 protéines ribosomiques (RPS) constituant la petite sous-unité. La grande sous-unité 60S est composée de trois ARNr distincts : 5S, 5.8S et 28S, ainsi que de 47 protéines ribosomiques (*RPL*) constituant la grande sous-unité.

Les ARNr sont essentiels pour la structure globale du ribosome et son bon fonctionnement. Ils supportent l'activité peptidyl-transférase du ribosome, qui catalyse la formation des liaisons peptidiques, un processus clé dans la synthèse des protéines [24].

2.2.Petit ARN (*sncRNA*)

Les petits ARN (*sncRNA*) sont une catégorie d'ARN de petite taille qui ne codent pas directement des protéines. Leur taille est généralement d'environ 200 nucléotides. Ils se localisent à la fois dans le noyau et le cytoplasme des cellules.

Les petits ARN remplissent diverses fonctions, agissant comme des régulateurs de l'expression génétique en interagissant avec les ARNm et en contrôlant leur stabilité ou leur traduction [25]. Il existe plusieurs types d'ARN dans cette classe, notamment les microARN (miARN), les *snoRNA* (*small nucleolar RNA*) et les *piARN* (*Piwi-interacting RNA*) [26].

2.2.1. miARN

Les microARN (miARN) sont de petites molécules d'ARN d'environ 21 à 23 nucléotides de longueur, découvertes en 1993. Ils jouent un rôle fondamental dans la régulation post-transcriptionnelle de l'expression des gènes en bloquant la traduction des ARNm cibles.

Leur propre expression est finement régulée, ce qui leur permet de participer à de nombreux processus biologiques importants [27].

2.2.2. snoRNA

Les *small nucleolar RNA* (*snoRNA*) sont de petits ARN présents dans les nucléoles des cellules. Ils sont composés de 60 à 300 nucléotides et jouent un rôle essentiel dans la modification et la maturation des ARN ribosomaux, des petits ARN nucléaires et d'autres ARN cellulaires. Les *snoARN* se divisent en deux catégories : les *snoRNA* à boîte C/D, responsables de la méthylation en 2'-O-ribose, et les *snoRNA* à boîte H/ACA, qui dirigent la pseudouridylation des nucléotides [28].

2.2.3. piARN

Les *Piwi-interacting RNA* (*piRNA*) sont de petites molécules d'ARN non codant qui proviennent de transcriptions précurseurs constituées de longues chaînes simples. Ces molécules ont une longueur de 21 à 35 nucléotides et interagissent avec les protéines PIWI pour former le complexe de silençage des piARN (*piRISC*). Les piARN jouent un rôle essentiel dans la répression des transposons, la modification des motifs de méthylation de l'ADN, le silençage des éléments transposables, la régulation de l'expression génique et la lutte contre les infections virales [29].

3. Techniques d'identification d'ARN

Les techniques d'identification de l'ARN ont connu des avancées significatives au fil des années, ce qui a permis une meilleure compréhension de la diversité et de la fonction des ARN. Dans cette section, nous présenterons les principales techniques couramment utilisées pour identifier les ARN (acides ribonucléiques).

3.1 Northern blot

Cette technique classique de détection et de quantification des ARN spécifiques est un protocole bien établi et largement utilisé dans le domaine de la biologie moléculaire. Elle comprend plusieurs étapes clés qui permettent une analyse précise des ARN cibles.

Initialement, *Northern Blot* repose sur l'utilisation de sondes d'ARN complémentaires marquées. Ces sondes synthétiques sont conçues de manière à être complémentaires à la séquence spécifique des ARN cibles que l'on souhaite détecter. Elles peuvent être marquées avec des marqueurs tels que les isotopes radioactifs ou les molécules fluorescentes, ce qui permet leur détection ultérieure. Une fois les sondes d'ARN marquées préparées, elles sont ajoutées à l'échantillon contenant les ARN à analyser. Elles se lient spécifiquement aux ARN cibles présents dans l'échantillon, formant ainsi des complexes d'hybridation. Cette étape est essentielle car elle permet de cibler exclusivement les ARN d'intérêt parmi la multitude d'ARN présents dans l'échantillon. Par la suite, les complexes d'ARN hybrides formés sont soumis à une électrophorèse sur gel. L'échantillon est déposé dans des puits d'un gel d'agarose ou de polyacrylamide, puis une différence de potentiel est appliquée à travers le gel. Les ARN, qui sont chargés négativement en raison de leur nature d'acide nucléique, migrent à travers le gel en fonction de leur taille. Cette migration permet de séparer les ARN en fonction de leur poids moléculaire, les ARN de petite taille se déplaçant plus rapidement que ceux de grande taille. Une fois la séparation par électrophorèse terminée, les ARN présents dans le gel sont transférés sur une membrane solide. Ce transfert est effectué en plaçant la membrane, généralement composée de nitrocellulose ou de nylon, sur le gel et en appliquant une pression ou une électrophorèse transversale pour transférer les ARN de manière efficace. Ce transfert immobilise les ARN sur la membrane tout en préservant leur position relative par rapport à leur séparation obtenue sur le gel. Enfin, les ARN immobilisés sur la membrane sont détectés à l'aide de différentes méthodes. L'autoradiographie est souvent utilisée lorsque des sondes d'ARN radioactives sont employées. Dans cette méthode, la membrane est exposée à un film photographique sensible aux radiations émises par les sondes radioactives. Les signaux

radioactifs se manifestent sous la forme de taches ou de bandes sur le film, indiquant ainsi la présence des ARN cibles. Lorsque des sondes d'ARN fluorescentes sont utilisées, une détection directe de la fluorescence peut être réalisée en utilisant des techniques d'imagerie appropriées [30].

3.2 RT-PCR (Reverse Transcription - Polymerase Chain Reaction)

La *RT-PCR* est une technique couramment utilisée en biologie moléculaire pour amplifier de manière spécifique les ARN présents dans un échantillon. Cette méthode est basée sur deux étapes essentielles : la transcription inverse et la polymérase en chaîne (PCR).

La première étape de la *RT-PCR* consiste en la transcription inverse, qui convertit les ARN en ADN complémentaire (ADNc). Cela est réalisé en utilisant une enzyme appelée transcriptase inverse, qui synthétise une séquence d'ADN complémentaire à partir de l'ARN cible. L'ADNc est donc une copie de l'ARN d'origine, mais sous forme d'ADN.

Une fois l'ADNc synthétisé, la deuxième étape de la RT-PCR intervient, qui est la polymérase en chaîne (PCR). La PCR amplifie spécifiquement l'ADNc, en utilisant des amorces spécifiques qui se lient aux extrémités de la séquence d'ADNc cible. Ces amorces permettent à une ADN polymérase de synthétiser des copies complémentaires de la séquence d'ADNc lors de cycles de chauffage et de refroidissement répétés. Ainsi, à chaque cycle, la quantité d'ADNc est doublée, ce qui conduit à une amplification exponentielle de la séquence cible.

La technique de *RT-PCR* présente plusieurs avantages significatifs. Tout d'abord, elle permet une amplification spécifique des ARN cibles, facilitant ainsi la détection de faibles concentrations d'ARN spécifiques au sein d'un échantillon complexe. De plus, elle se caractérise par une grande sensibilité et une grande spécificité, offrant ainsi une méthode précise pour la détection et la quantification des ARN [31].

3.3 Hybridation in situ

L'hybridation in situ est une technique puissante qui permet la visualisation et la localisation précise des ARN spécifiques à l'intérieur des cellules ou des tissus. Elle utilise des sondes d'ADN ou d'ARN marquées avec des marqueurs fluorescents ou radioactifs, qui se lient de manière complémentaire aux séquences d'ARN cibles dans leur emplacement

naturel. Lors de cette technique, les échantillons cellulaires ou tissulaires sont fixés sur des supports, tels que des lames de microscope. Les sondes marquées sont ensuite appliquées sur les échantillons, permettant ainsi leur hybridation avec les ARN correspondants. La spécificité de cette technique repose sur la complémentarité entre les séquences des sondes et celles des ARN cibles, ce qui assure une liaison sélective. Une fois l'hybridation réalisée, les signaux de liaison entre les sondes et les ARN cibles peuvent être détectés. Si les sondes sont marquées avec des fluorochromes, une observation au microscope à fluorescence permet de visualiser les signaux fluorescents émis par les sondes liées aux ARN cibles. Cette approche offre une excellente résolution spatiale, permettant d'observer la localisation subcellulaire des ARN d'intérêt. A l'inverse, si les sondes sont marquées avec des isotopes radioactifs, la détection des signaux de liaison peut être réalisée par autoradiographie. Les rayonnements émis par les isotopes radioactifs sont capturés par un film photographique, produisant des taches ou des bandes correspondant aux emplacements où les sondes ont lié les ARN cibles. L'hybridation in situ permet une analyse directe des ARN dans leur contexte cellulaire ou tissulaire, sans avoir besoin d'extraire ou d'amplifier les ARN avant l'expérimentation. Cela garantit la préservation de la localisation et de l'abondance relative des ARN dans leur état natif. De plus, cette technique offre une grande sensibilité, permettant la détection d'ARN à faible niveau d'expression [32].

3.4 *RNA-Seq* (Séquençage de l'ARN)

Le *RNA-Seq*, ou séquençage de l'ARN, est une technique avancée permettant de séquencer l'ensemble des ARN présents dans un échantillon biologique. Cette méthode utilise les technologies de séquençage à haut débit pour générer des millions de séquences d'ARN à partir de l'échantillon étudié. Ces séquences d'ARN sont ensuite alignées sur un génome de référence approprié afin d'identifier et de quantifier les ARN spécifiques. L'analyse de ces séquences repose sur leur alignement sur un génome de référence. Cette étape permet d'associer chaque séquence d'ARN à une position spécifique dans le génome et d'identifier ainsi l'ARN d'origine. En comparant les séquences alignées avec les annotations génomiques, il est possible de déterminer les gènes exprimés, les isoformes d'ARN, les variations génétiques, etc. En outre, la quantification des ARN peut être réalisée en mesurant le nombre de séquences alignées sur chaque gène ou isoforme d'ARN, ce qui permet d'estimer leur niveau d'expression relatif.

Grâce à sa capacité à fournir une vue d'ensemble complète du transcriptome, le RNA-Seq a révolutionné la compréhension des régulations géniques et des processus biologiques.

Il a été largement utilisé dans divers organismes, notamment les levures, les plantes, les animaux et les humains, etc. Finalement, l'analyse par *RNA-Seq* offre de multiples avantages par rapport aux méthodes traditionnelles de quantification des ARN. Premièrement, elle permet une couverture exhaustive de l'ensemble du transcriptome, car elle peut détecter et quantifier tous les ARN présents dans l'échantillon, y compris ceux qui sont peu abondants. De plus, le séquençage à haut débit génère un grand nombre de séquences, ce qui permet d'obtenir une représentation précise de l'expression des ARN dans l'échantillon (**figure 04**) [33].

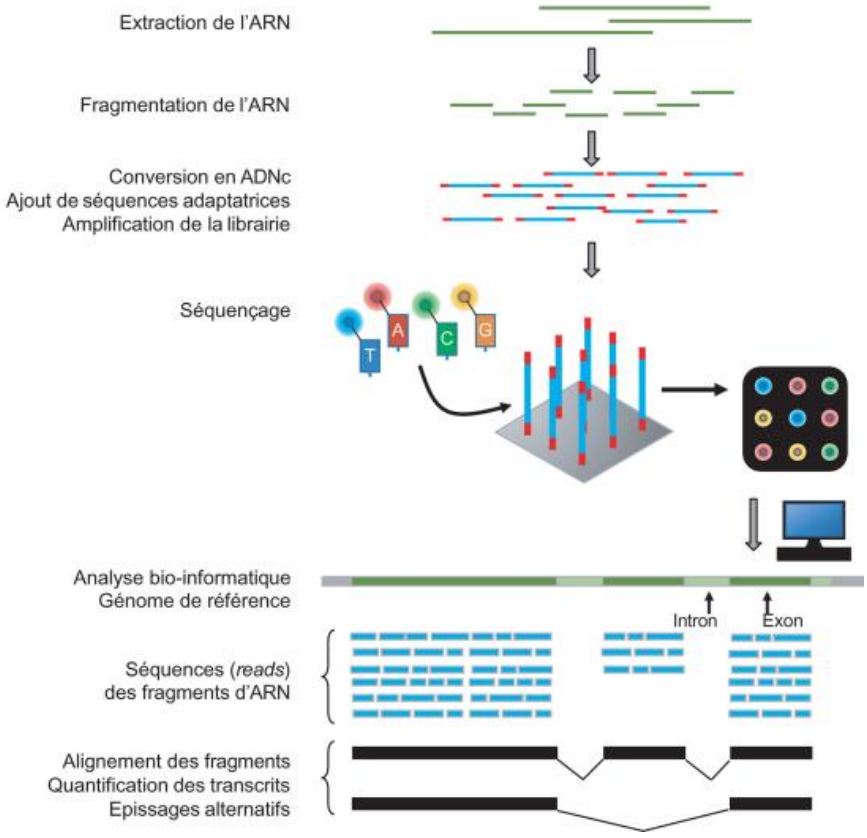


Figure 04. Les étapes de la technique de séquençage d'ARN

3.5 Microarray

La méthode de *microarray* utilise des puces à ADN ou des sondes d'ARN pour permettre la détection et la quantification simultanée d'un grand nombre d'ARN. Les ARN cibles sont préalablement marqués et hybridés avec les sondes spécifiques présentes sur la puce. Cette hybridation se produit lorsque les sondes complémentaires se lient spécifiquement

aux ARN cibles. Une fois les ARN cibles hybridés, ils sont détectés en utilisant des techniques de fluorescence. Dans cette méthode, les puces à ADN ou les sondes d'ARN contiennent des séquences complémentaires spécifiques à différents ARN d'intérêt. L'échantillon d'ARN à analyser est généralement marqué avec des fluorochromes ou d'autres marqueurs qui émettent une fluorescence lorsqu'ils sont excités par une source lumineuse appropriée. Lorsque l'échantillon d'ARN marqué est appliqué sur la puce ou en présence des sondes d'ARN, une réaction d'hybridation se produit, permettant aux ARN cibles de se lier spécifiquement aux sondes correspondantes sur la puce. Après une étape de lavage pour éliminer les ARN non liés, la puce est analysée à l'aide d'un scanner à fluorescence. Le scanner détecte la fluorescence émise par les ARN cibles qui se sont hybridés avec les sondes sur la puce. Les intensités de fluorescence sont enregistrées et permettent de quantifier la présence et l'abondance relative des ARN cibles dans l'échantillon. La *microarray* permet la détection simultanée de nombreux ARN dans un échantillon, ce qui en fait une approche efficace pour l'analyse génomique et l'étude des profils d'expression génique. Elle a été largement utilisée dans des domaines : la génomique fonctionnelle, la recherche sur le cancer, la découverte de médicaments et d'autres applications en biologie moléculaire et en médecine [34].

3.6 FISH (Fluorescence in situ Hybridization)

La technique de *FISH* est largement utilisée en biologie moléculaire pour visualiser et localiser des ARN spécifiques dans les cellules. Cette méthode repose sur l'utilisation de sondes d'ARN spécifiques, marquées avec des molécules fluorescentes, qui se lient de manière complémentaire aux ARN cibles présents dans les cellules. L'hybridation entre les sondes marquées et les ARN cibles se produit dans les conditions appropriées, ce qui permet de détecter directement les ARN spécifiques au microscope à fluorescence.

L'avantage fondamental de la technique *FISH* est sa capacité à fournir une localisation précise des ARN au sein des cellules, ce qui permet d'étudier leur expression spatiale. Cette méthode est particulièrement utile pour comprendre les processus de régulation génétique, car elle permet de visualiser l'expression de gènes spécifiques dans différents types de cellules ou de tissus. Pour réaliser une expérience de *FISH*, les cellules sont généralement fixées sur une lame de microscope et traitées avec des sondes d'ARN marquées spécifiques. Après l'hybridation, les échantillons sont soumis à une série de lavages pour éliminer les sondes non liées, puis ils sont observés au microscope à fluorescence. Les signaux fluorescents provenant des sondes liées aux ARN cibles sont détectés et peuvent être enregistrés par imagerie. Les

images obtenues fournissent des informations précieuses sur la localisation et la distribution des ARN spécifiques dans les cellules étudiées. Cette technique peut être utilisée dans divers domaines de recherche, dans l'étude du développement embryonnaire, la caractérisation des tumeurs, l'identification de microorganismes spécifiques et l'analyse des interactions entre les gènes et les protéines. Elle offre également la possibilité d'observer les changements dynamiques de l'expression des ARN au fil du temps [35].

CHAPITRE 03 : L'APPRENTISSAGE APPROFONDI

L’APPRENTISSAGE APPROFONDI

1. Définition de l’apprentissage approfondi

L'apprentissage profond, également connu sous le nom de *deep learning*, est une approche de l'apprentissage automatique qui vise à modéliser des abstractions de haut niveau à partir de données en utilisant des réseaux de neurones profonds. Il s'agit d'une branche de l'intelligence artificielle qui vise à créer des modèles capables d'apprendre et de prendre des décisions de manière autonome en traitant de grandes quantités de données.

Au cours des dernières années, l'apprentissage profond a connu une popularité croissante en raison de sa capacité à traiter des données complexes et non structurées telles que des images, des textes ou des signaux audio. Il a été appliqué avec succès à de nombreux domaines, y compris la reconnaissance d'images, la compréhension du langage naturel, la traduction automatique, la reconnaissance vocale, la recommandation de produits, la prédiction de séries temporelles, et bien d'autres.

Une caractéristique clé de l'apprentissage profond est l'utilisation de réseaux de neurones profonds, qui sont des architectures composées de multiples couches de neurones artificiels. Chaque couche traite les données reçues de la couche précédente et les transforme à l'aide de fonctions non linéaires. Ces couches successives permettent au réseau de capturer des niveaux d'abstraction de plus en plus élevés, ce qui lui confère une capacité d'apprentissage et de généralisation puissante.

L'apprentissage profond repose généralement sur des modèles d'apprentissage supervisé, où le réseau est entraîné sur un ensemble de données étiquetées pour apprendre à effectuer des prédictions précises. Cependant, des approches d'apprentissage non supervisé, telles que les réseaux de neurones autoencodeurs ou les réseaux de neurones génératifs, sont également utilisées pour découvrir des structures et des représentations cachées dans les données.

En bioinformatique et en médecine, l'apprentissage profond a été appliqué à diverses tâches, telles que l'analyse de séquences d'ADN et d'ARN, la prédiction de structures de protéines, le diagnostic médical à partir d'images médicales, la découverte de médicaments et la médecine de précision. Ces applications ont montré des améliorations significatives par

rapport aux approches traditionnelles, ce qui a suscité un grand intérêt dans la communauté scientifique et médicale [36].

2. Les réseaux neurones

Un réseau de neurones est un système interconnecté d'unités de traitement, inspiré de manière générale par le fonctionnement des neurones biologiques. Il est composé de plusieurs éléments de traitement simples appelés nœuds ou unités (**figure 05**).

La capacité de traitement du réseau est stockée dans les poids des connexions entre ces unités, qui sont ajustés lors d'un processus d'apprentissage à partir d'un ensemble de modèles d'entraînement.

La mise en œuvre d'un réseau de neurones comprend généralement les étapes suivantes :

- **Acquisition de l'ensemble de données d'entraînement et de test :** Les données utilisées pour entraîner et évaluer le réseau sont collectées et préparées.
- **Entraînement du réseau :** Les poids des connexions entre les unités du réseau sont ajustés en utilisant un algorithme d'apprentissage. Cela implique de présenter les données d'entraînement au réseau, de calculer les sorties prédites et de comparer ces prédictions aux sorties attendues, puis d'ajuster les poids en fonction de l'erreur obtenue.
- **Prédiction avec les données de test :** Une fois que le réseau est entraîné, il peut être utilisé pour effectuer des prédictions sur de nouvelles données (les données de test) en utilisant les poids ajustés.

Les réseaux neuronaux profonds, également appelés réseaux neuronaux profonds, sont des types spécifiques de réseaux de neurones qui sont composés de plusieurs couches de nœuds. Ces réseaux utilisent différentes architectures et techniques pour résoudre des problèmes complexes dans divers domaines, tels que la vision par ordinateur, le traitement du langage naturel, la reconnaissance de motifs, etc. Les réseaux neuronaux profonds ont montré des performances impressionnantes dans de nombreuses applications et ont contribué aux avancées récentes de l'apprentissage en profondeur [37].

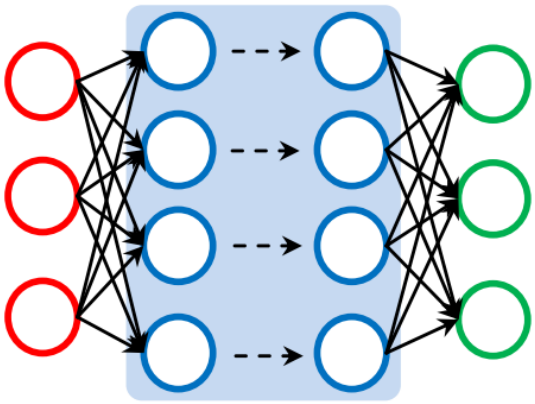


Figure 05. Exemple d'un réseau de neurone artificiel

3. Les architectures des réseaux neuronaux les plus utilisées

3.1.Réseau neuronal convolutif (Convolutional Neural Network)

Les réseaux neuronaux convolutifs (*CNN*) sont des architectures de réseaux neuronaux qui tirent leur inspiration du fonctionnement du cortex visuel du cerveau. Ils sont conçus pour la perception et l'analyse de données structurées, telles que des images ou des signaux temporels. Les *CNN* sont caractérisés par un flux d'informations unidirectionnel, allant des entrées aux sorties, et sont particulièrement adaptés au traitement de données avec une structure de grille, comme les images.

Les *CNN* sont composés de plusieurs couches qui se succèdent. Les principales couches sont les couches de convolution et de *pooling* (ou sous-échantillonnage). Les couches de convolution appliquent des filtres sur les données d'entrée pour extraire des caractéristiques locales, tandis que les couches de *pooling* réduisent la dimension spatiale des données en sélectionnant les valeurs les plus importantes. Ces couches de convolution *pooling* sont

souvent regroupées en modules, et plusieurs modules peuvent être empilés pour former un réseau profond.

Ensuite, les sorties des couches convolutives sont généralement connectées à des couches entièrement connectées, similaires aux réseaux neuronaux classiques, qui effectuent des opérations de classification ou de prédiction finale. L'apprentissage des CNN se fait par rétropropagation du gradient, où les poids des connexions sont ajustés pour minimiser l'erreur entre les prédictions du réseau et les étiquettes de sortie attendues.

Les CNN ont connu un grand succès dans des tâches telles que la reconnaissance d'images, la détection d'objets, la segmentation d'images et le traitement du langage naturel. Leur capacité à extraire automatiquement des caractéristiques pertinentes à partir des données en fait des outils puissants pour des problèmes complexes [38].

3.2.Autoencodeur (*Autoencoder*)

L'objectif principal des autoencodeurs est d'apprendre une représentation comprimée des données qui capture les caractéristiques les plus importantes. En réduisant la dimensionnalité des données, les autoencodeurs peuvent être utilisés pour extraire des caractéristiques pertinentes et effectuer des tâches telles que la compression de données, la détection d'anomalies ou la génération de données synthétiques.

Les autoencodeurs parcimonieux sont une variation des autoencodeurs dans laquelle une contrainte est imposée pour encourager certaines unités à être inactives ou à produire des valeurs proches de zéro. Cette contrainte favorise l'émergence de représentations plus parcimonieuses, c'est-à-dire des représentations qui ne sont activées que pour un sous-ensemble restreint des données. Cela peut être utile pour la sélection automatique des caractéristiques les plus discriminantes et pour améliorer l'interprétabilité des modèles.

Les autoencodeurs sont largement utilisés dans divers domaines, tels que la vision par ordinateur, le traitement du langage naturel, la recommandation de produits et la détection d'anomalies [39].

3.3.Machine de Boltzmann restreinte (Restricted Boltzmann Machine ou RBM)

La machine de Boltzmann restreinte (*Restricted Boltzmann Machine ou RBM)* est un type de réseau neuronal artificiel utilisant un algorithme d'apprentissage non supervisé pour construire des modèles génératifs non linéaires à partir de données non étiquetées [40]. Son fonctionnement repose sur l'idée d'entraîner le réseau à maximiser une fonction (telle qu'un produit ou un logarithme) de la probabilité d'un vecteur dans les unités visibles, afin de

reconstruire probabilistiquement l'entrée. Ainsi, la *RBM* apprend la distribution de probabilité des données d'entrée.

La *RBM* se compose de deux couches : la couche visible et la couche cachée. Chaque unité de la couche visible est connectée à toutes les unités de la couche cachée, tandis qu'il n'existe pas de connexions entre les unités de la même couche. Cette architecture permet à la RBM de capturer des relations complexes entre les variables d'entrée et d'apprendre des représentations efficaces des données en extrayant des caractéristiques significatives.

L'apprentissage de la *RBM* se fait par itérations, où les activations des unités visibles sont propagées aux unités cachées, puis les activations des unités cachées sont rétropropagées aux unités visibles. Ce processus d'activation et de rétropropagation permet à la RBM d'ajuster les poids des connexions afin de reconstruire de manière probabiliste les données d'entrée. Une fois entraînée, la *RBM* peut être utilisée pour générer de nouvelles données similaires à celles sur lesquelles elle a été formée.

Les *RBM* ont été largement utilisées dans des applications telles que la recommandation de produits, la classification de textes, la détection d'anomalies, etc. Elles ont également été combinées avec d'autres architectures de réseaux neuronaux, comme les réseaux de neurones profonds, pour améliorer les performances dans des tâches plus complexes.

3.4.Mémoire à court et long terme (*Long Short-Term Memory ou LSTM*)

LSTM (Long Short-Term Memory) est une implémentation spécifique des réseaux neuronaux récurrents (RNN) qui a été proposée pour la première fois par *Hochreiter et al.* en 1997 [33]. Il se distingue par sa capacité à conserver la connaissance des états antérieurs, ce qui le rend adapté aux tâches nécessitant une mémoire à long terme ou une conscience de l'état [34]. Les *LSTM* sont particulièrement efficaces pour apprendre à partir de données d'entrée séquentielles et pour construire des modèles qui tirent parti du contexte et des états précédents. Le bloc cellulaire d'un *LSTM* conserve les informations pertinentes des états précédents. Les portes d'entrée, d'oubli et de sortie régulent les flux de données entrant dans la cellule, ce qui est conservé dans la cellule et les valeurs de la cellule utilisées pour calculer la sortie du bloc *LSTM* respectivement [42,43].

L'objectif de la formation des architectures d'apprentissage en profondeur est d'optimiser les paramètres de poids dans chaque couche. Dans un réseau de neurones profond, chaque couche traite les informations transmises par la couche précédente en combinant des

fonctionnalités plus simples pour en créer de plus complexes. Cela permet d'apprendre les représentations hiérarchiques les plus appropriées à partir des données [44].

4. Le processus de l'apprentissage approfondi dans le cas de Classification des séquences

Le processus d'apprentissage approfondi (*Deep Learning, DL*) dans le cas de la classification des séquences comprennent plusieurs étapes essentielles pour créer un modèle efficace. Ces étapes peuvent être décrites comme suit :

a) Collecte et préparation des données : Cette étape consiste à collecter les données nécessaires pour l'entraînement et le test du modèle de classification. Les données doivent être nettoyées, prétraitées et divisées en ensembles d'entraînement, de validation et de test [45].

b) Choix de l'architecture du modèle : Le choix de l'architecture du modèle est crucial pour la classification des séquences. Il peut s'agir d'un réseau de neurones convolutif (*CNN*) pour la classification d'images, d'un réseau récurrent (*RNN*) ou d'un réseau *LSTM* pour la classification de séquences temporelles. L'architecture choisie dépend de la nature des données et du problème de classification spécifique [46].

c) Entraînement du modèle : Cette étape implique l'entraînement du modèle en utilisant les données d'entraînement. Le modèle est exposé aux données et ajuste ses paramètres pour minimiser l'erreur de prédiction. L'optimisation est généralement réalisée à l'aide d'algorithmes de descente de gradient, tels que la rétropropagation [47,48].

d) Validation du modèle : Le modèle entraîné est évalué à l'aide des données de validation. Cela permet de mesurer sa performance et d'ajuster les hyperparamètres si nécessaire. La validation croisée peut également être utilisée pour évaluer la robustesse du modèle en le testant sur plusieurs ensembles de validation [49].

e) Évaluation du modèle : Une fois que le modèle est entraîné et validé, il est évalué à l'aide des données de test. Cela permet de mesurer sa performance réelle sur des données inconnues. Les métriques d'évaluation, telles que l'exactitude, la précision, le rappel et la F-mesure, sont utilisées pour évaluer la qualité des prédictions du modèle [50].

f) Interprétation des résultats : Les résultats obtenus par le modèle sont interprétés et analysés pour comprendre les facteurs qui influencent la classification. Cela peut inclure l'identification des caractéristiques importantes, la visualisation des activations du modèle ou l'analyse des erreurs de classification afin d'améliorer le modèle et de mieux comprendre le problème étudié [51].

5. Les applications de deep learning en biologie

Avec le développement de la recherche, de nombreux algorithmes d'apprentissage profond appliqués aux structures de données des réseaux biologiques qui contiennent beaucoup d'informations entre les organismes. L'exploration des réseaux biologiques est importante pour comprendre la corrélation interne des biomolécules, la découverte de médicaments, le traitement des maladies et le mécanisme d'action des micro-organismes [52].

Dans cette section, l'application de l'apprentissage en profondeur peut être présenté dans les réseaux biologiques par le biais de la recherche de données génomiques, de la recherche de données protéomiques, de la recherche de données transcriptomiques, de la découverte de médicaments, de la biologie des maladies et de la recherche de données sur le microbiome, etc. il est présenté dans la **figure 06**.

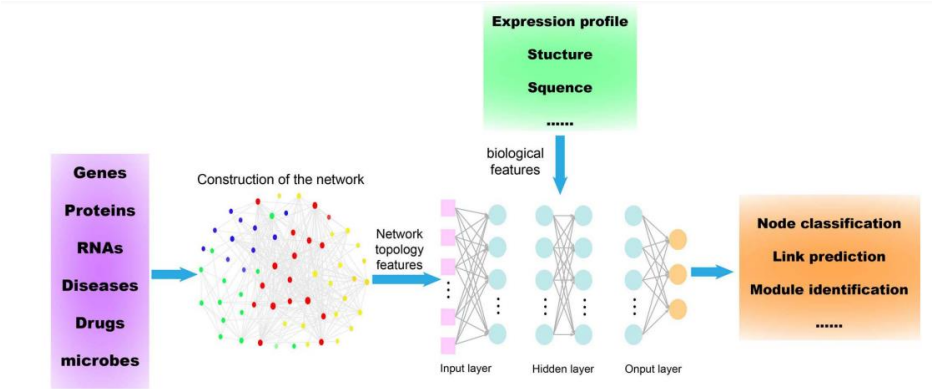


Figure 06. Les applications de l'apprentissage approfondi dans les données biologiques

- **En génomique**

L'apprentissage approfondi permet d'annoter plus précisément les régions fonctionnelles du génome en reconnaissant les motifs et les caractéristiques associés à ces régions. De plus, il est utilisé pour classifier les variants génétiques en fonction de leur impact sur la fonction protéique ou leur association avec des maladies, ce qui permet d'identifier plus efficacement les variants pathogènes. L'apprentissage approfondi peut également prédire la structure tridimensionnelle de l'ADN et de l'ARN à partir des séquences génomiques, fournissant ainsi des informations précieuses sur les interactions moléculaires et les régions fonctionnelles. De plus, il est utilisé pour reconnaître les motifs régulateurs dans les séquences génomiques, aidant ainsi à prédire les régions régulatrices et à mieux comprendre le contrôle de l'expression des gènes. L'utilisation de l'apprentissage approfondi dans la génomique permet une analyse et une interprétation plus avancées des données génomiques, contribuant ainsi à une meilleure compréhension du génome, à l'identification de variants génétiques pathogènes et à des avancées significatives dans les applications médicales et biotechnologiques [53].

- **En protéomique**

L'utilisation de l'apprentissage approfondi dans le domaine de la protéomique a révolutionné la prédiction de la structure des protéines. Grâce à des décennies de recherche et de développement, les méthodes statistiques ont évolué vers des techniques d'apprentissage automatique de plus en plus complexes, pour finalement adopter les méthodes d'apprentissage profond. Ces dernières années, les réseaux neuronaux convolutifs, récurrents et à propagation avant ont été largement utilisés pour capturer des informations structurales complexes à différents niveaux de détail. Ces avancées ont permis d'obtenir des prédictions de plus en plus précises et fiables sur la structure des protéines. L'apprentissage profond a également permis d'explorer les annotations de structure des protéines à une et deux dimensions, en passant des méthodes statistiques simples aux algorithmes sophistiqués et intensifs en calcul. En outre, les bases de données protéiques ont considérablement augmenté en taille, fournissant ainsi des ensembles de données plus riches pour l'apprentissage des modèles d'apprentissage profond. Grâce à cette approche, il est maintenant possible d'exploiter les connaissances sur l'évolution et la co-évolution pour améliorer les prédictions de la structure des protéines.

L'apprentissage approfondi joue donc un rôle central dans les pipelines de prédiction des structures protéiques, ouvrant la voie à de nouvelles opportunités et défis passionnants dans le domaine de la protéomique [54].

- **En biologie moléculaire**

L'apprentissage approfondi est utilisé pour prédire la structure des protéines à partir de leur séquence d'acides aminés, ce qui est essentiel pour comprendre leur fonction et leur interaction avec d'autres molécules. Cela aide à la conception de médicaments ciblés et à la recherche de nouvelles cibles thérapeutiques. Ensuite, il est utilisé pour l'annotation fonctionnelle des génomes en identifiant les régions régulatrices, les promoteurs et d'autres éléments importants dans la régulation de l'expression génique. Cela permet de mieux comprendre les mécanismes de régulation et de prédire la fonction des gènes. De plus, cet apprentissage est utilisé dans le domaine de la classification cellulaire, pour identifier et caractériser différents types cellulaires à partir de données de transcription, d'imagerie et de cytométrie en flux. Cela aide à comprendre la diversité cellulaire et les interactions cellulaires au sein des tissus et des organes. Finalement, l'apprentissage approfondi joue un rôle important dans la prédiction et le diagnostic des maladies. Il permet de développer des modèles de prédiction de maladies à partir de données cliniques et moléculaires, ce qui peut contribuer à un diagnostic précoce et à des interventions thérapeutiques plus précises [55].

- **En microbiologie**

L'apprentissage approfondi peut être utilisé pour classifier le microbiome et effectuer des analyses taxonomiques en se basant sur les données génomiques des microorganismes. Cela permet d'identifier et de caractériser les différentes populations microbiennes présentes dans un échantillon. De plus, l'apprentissage approfondi contribue à l'étude de l'écologie microbienne en analysant les données environnementales et les profils métaboliques des microorganismes. Il permet de révéler des interactions complexes et des dynamiques difficilement détectables par les méthodes traditionnelles. En ce qui concerne les pathogènes et l'épidémiologie, l'apprentissage approfondi aide à la détection et à la prédiction des maladies infectieuses. En analysant des données cliniques et génomiques, il permet d'identifier les marqueurs de virulence des pathogènes, de prévoir l'évolution des épidémies

et de développer des stratégies de prévention. Enfin, l'apprentissage approfondi accélère la découverte de médicaments antimicrobiens en analysant de vastes bases de données chimiques et en prédisant l'activité antimicrobienne de nouvelles molécules. En résumé, l'apprentissage approfondi offre des avantages considérables à la microbiologie en exploitant les données massives disponibles et en améliorant notre compréhension des microorganismes et de leur impact sur les écosystèmes et la santé humaine [56].

- **En bioinformatique**

Dans le domaine des omiques, il est utilisé pour l'analyse des données massives de séquençage génomique, de séquençage d'ARN et de protéomique. L'apprentissage approfondi permet d'identifier des motifs complexes dans les séquences d'ADN et d'ARN, de prédire la structure tridimensionnelle des protéines, d'annoter les gènes et de découvrir de nouvelles associations génétiques avec des maladies. En ce qui concerne l'imagerie biomédicale, le deep learning est utilisé pour l'analyse d'images médicales telles que les radiographies, les IRM et les scans CT. Il permet la détection automatique de lésions, la segmentation des tissus, la classification de différentes pathologies et la prédiction de résultats cliniques. Il permet également l'analyse d'images cellulaires, permettant de caractériser et de classer différents types de cellules et de suivre leur évolution au fil du temps. Dans le domaine du traitement des signaux biomédicaux, cet apprentissage est utilisé pour l'analyse de signaux tels que l'électrocardiogramme (ECG), l'électroencéphalogramme (EEG) et d'autres signaux physiologiques. Il permet la détection automatique de motifs et de caractéristiques dans les signaux, la classification de maladies, la prédiction de résultats cliniques et l'aide au diagnostic médical. Il est également appliqué à l'analyse des réseaux biologiques, en aidant à comprendre les interactions entre les gènes, les protéines et les voies biologiques. Il permet la prédiction de nouvelles interactions protéine-protéine, l'identification de régulateurs clés et la modélisation des réseaux de régulation génétique [57].

- **En médecine**

L'utilisation de l'apprentissage approfondi dans la biologie et la médecine permet de combiner des données complexes et riches en informations pour résoudre des problèmes spécifiques. Les algorithmes d'apprentissage approfondi ont montré des résultats

impressionnants dans ces domaines, offrant de nouvelles perspectives pour la classification des patients, la compréhension des processus biologiques fondamentaux et l'amélioration des traitements. Bien que l'apprentissage approfondi n'ait pas encore révolutionné la biomédecine ni résolu tous les défis majeurs, il représente un outil prometteur pour accélérer les enquêtes humaines et apporter des changements significatifs tant en laboratoire que dans les soins aux patients. Les avancées dans ce domaine ouvrent de nouvelles opportunités pour transformer et améliorer divers aspects de la biologie et de la médecine [58].

PARTIE 02 : MATERIEL ET METHODES

MATERIEL ET METHODES

1. Matériel

1.1. Données biologiques : Les données utilisées pour ce travail sont 2 *dataset* de chez l'espèce *homosapiens*, présentés dans le **tableau 02.**

Tableau 02. Les informations des données biologiques utilisées

Type de dataset	Base de données	Date d'extraction	Fichier format	Taille du fichier	Nombre des séquences
ARN non codants	Rfam[1]	08/04/2023	FASTA	1265 Ko	6320
ARN codant (ARNm)	RefSeq[2]	04/05/2023	FASTA	12107 Ko	7129

1.2. Configuration de la machine : Ses caractéristiques sont détaillées dans le **tableau 03.**

Tableau 03. Les caractéristiques de l'ordinateur utilisé

Ordinateur	Caractéristiques
Processeur	Intel(R) Celeron(R) CPU N2840 @ 2.16GHz, 2159 MHz, 2 cœur(s), 2 processeur(s) logique(s)
Mémoire installée RAM	4,00 Go CPU N2840 @ 2.16GHz,
Stockage	HDD de 500 gb
Système d'Exploitation	Windows 10 famille
Type de système	Système d'exploitation 64 bits

1.3. Outils et bibliothèques informatiques : Nous décrivons brièvement les outils et les bibliothèques utilisés pour réaliser le travail dans cette section :

1.3.1. Environnement de travail : Le travail est réalisé en utilisant les outils présentés dans le **tableau 04.**

[1] https://rfam.org/
[2] https://www.ncbi.nlm.nih.gov/refseq/

Tableau 04. Principaux outils utilisés

Outil	Description
Python[3] 3.10	Un langage de programmation open source, interprété et haut niveau, conçu pour être facile à lire et à écrire. Il est utilisé dans les domaines de la science des données et de l'analyse, ainsi que dans le développement Web, l'automatisation système et la création de logiciels de bureau. Python possède une grande bibliothèque standard, ainsi que de nombreuses bibliothèques tierces pour la manipulation de données, la visualisation, l'apprentissage automatique, etc.

1.3.2. Bibliothèques python : afin d'effectuer ce travail, plusieurs bibliothèques python ont été utilisées, les principales sont présentés dans le **tableau 05.**

Tableau 05. Les bibliothèques python utilisées

Bibliothèque	Description
Pandas[4]	Une bibliothèque open source de traitement et d'analyse de données pour Python. Elle fournit des structures de données flexibles pour la manipulation de données tabulaires et de séries temporelles. Les principales structures de données de Pandas sont les objets *Series* et *DataFrame*. Les objets *Series* sont des tableaux unidimensionnels de données étiquetées, tandis que les objets *DataFrame* sont des tableaux bidimensionnels de données étiquetées, similaires à des tableaux dans une feuille de calcul. Pandas fournit également des outils pour importer et exporter des données à partir de différents formats de fichier, ainsi que pour nettoyer, transformer et manipuler des données.
Sklearn[5]	Une bibliothèque open-source en Python spécialisée dans l'apprentissage automatique (*machine learning*). Elle offre une vaste gamme d'algorithmes d'apprentissage automatique, des outils de prétraitement des données et des utilitaires pour évaluer et optimiser les modèles.

[3] https://www.python.org/
[4] https://pandas.pydata.org/
[5] https://pypi.org/project/sklearn/

Tensorflow[6]	Une bibliothèque open source de machine learning développée par Google. Elle permet de construire et d'entraîner des réseaux de neurones profonds pour une variété de tâches, telles que la classification, la reconnaissance d'images, la reconnaissance vocale, la traduction de langues, et bien d'autres encore. TensorFlow fournit une API flexible pour la création de modèles, ainsi qu'une interface graphique pour la visualisation des graphes de calcul. Elle est également compatible avec les GPU pour accélérer les calculs, et peut être utilisée en conjonction avec d'autres bibliothèques de Python, telles que NumPy et Pandas.
Numpy[7]	NumPy est une bibliothèque Python open-source largement utilisée pour effectuer des calculs numériques et manipuler des tableaux multidimensionnels de manière efficace. Son nom est une abréviation de "*Numerical Python*". NumPy fournit des structures de données puissantes, des fonctions de manipulation de tableaux et des outils pour effectuer des opérations mathématiques avancées.
Matplotlib[8]	Une bibliothèque populaire de visualisation de données en Python. Elle permet de créer une grande variété de graphiques, de tracés et de visualisations pour représenter visuellement des données sous forme de graphiques en 2D et 3D. Matplotlib est une bibliothèque flexible et puissante, largement utilisée dans les domaines scientifiques, de l'analyse de données et de la visualisation.

[6] https://www.tensorflow.org/
[7] https://numpy.org/
[8] https://matplotlib.org/

2. MÉTHODES

Cette partie représente les méthodes utilisées afin de parvenir à la prédiction de la classification des ARN codant/non codant.

2.1.Prétraitement des données

La phase de prétraitement des données consiste en plusieurs étapes visant à obtenir un ensemble de données propre et prêt à être utilisé dans le modèle d'apprentissage. Ces étapes ont été appliquées aux deux ensembles de données d'ARN de manière parallèle. Voici une liste générale des étapes qui ont été utilisées :

2.1.1. Conversion des deux fichiers, au format FASTA, en fichiers CSV. La conversion en format CSV permet une meilleure manipulation et une meilleure visibilité des données.

2.1.2. Lecture des fichiers CSV en tant que *dataframes* Pandas contenant la Description, les Séquences et la Taille. Voir les **figures 07 et 08.**

```
df
```

	Description	Sequence	Length
0	lcl|NM_001256071.3_cds_NP_001243000.2_1 [gene=...	ATGGAGTGTCCTTCGTGCCAGCATGTCTCCAAGGAGGAAACCCCCA...	15624
1	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898
2	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454
3	lcl|NM_001199319.2_cds_NP_001186248.1_1 [gene=...	ATGAAGAGCGATTCTTCGACCTCTGCAGCCCCCCTCAGGGGGCTCG...	771
4	lcl|NM_001204504.3_cds_NP_001191433.1_1 [gene=...	ATGGCAGGGGCCGGAGGCCAGCACCACCCTCCGGGCGCCGCTGGAG...	3495
...	...	...	...
7124	lcl|NM_001244847.2_cds_NP_001231776.1_1 [gene=...	ATGAAGATCCCGGTCCTTCCTGCCGTGGTGCTCCTCTCCCTCCTGG...	258
7125	lcl|NM_001206426.2_cds_NP_001193355.1_1 [gene=...	ATGGCAGGTCCAGAAAGTGATGCGCAATACCAGTTCACTGGTATTA...	177
7126	lcl|NM_001271680.2_cds_NP_001258609.1_1 [gene=...	ATGTCTCTTCCAAGGCTAAAAGTAAACATCAGAACATATGATAATT...	207
7127	lcl|NM_001205266.2_cds_NP_001192195.1_1 [gene=...	ATGAGGGTCTTGTATCTCCTCTTCTCGTTCCTCTTCATATTTCTGA...	195
7128	lcl|NM_001242853.1_cds_NP_001229782.1_1 [gene=...	ATGAGGGTCTTGTTTTTTGTCTTTGGAGTCCTTTCCTTGATGTCCA...	213

7129 rows × 3 columns

Figure 07. *Dataset* d'ARNm sous forme d'un *dataframe*

```
df
```

	Description	Sequence	Length
0	RF00001_AF095839_1_346-228 5S_rRNA	GCGTACGGCCATACTATGGGGAATACACCTGATCCCGTCCGATTTC...	119
1	RF00001_AY245018_1_1-119 5S_rRNA	GCTATCGGCCATACTAAGCCAAATGCACCGGATCCCTTCCGAACTC...	119
2	RF00001_X52048_1_2-120 5S_rRNA	TGCTACGATCATACCACTTAGAAAGCACCCGGTCCCATCAGACCCC...	119
3	RF00001_M28193_1_1-119 5S_rRNA	AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTG...	119
4	RF00001_X14816_1_860-978 5S_rRNA	ACCAACGGCCATACCACGTTGAAAGTACCCAGTCTCGTCAGATCCT...	119
...	...	...	...
6315	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131
6316	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163
6317	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172
6318	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142
6319	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134

6320 rows × 3 columns

Figure 08. *Dataset* d'ARNnc sous forme d'un *dataframe*

2.1.3. Filtrage

a) Le premier filtrage a été effectué en se basant sur la taille des séquences. Les séquences dont la taille varie entre 100 et 800 nucléotides ont été conservées pour les ARN non codants, tandis que les séquences des ARN messagers (ARNm) dont la taille varie entre 800 et 3000 nucléotides ont été conservées. Les plages de tailles ont été déterminées en fonction de la plus petite et de la plus grande séquence. Voir les **figures 09 et 10**.

```
filtered_df = pd.read_csv('//content/updated_mRNA_dataset.csv')
filtered_df = filtered_df[filtered_df['Length'].between(800, 3000)]
filtered_df.to_csv('mRNA_new_dataset.csv', index=False )
```

```
import pandas as pd
df = pd.read_csv('mRNA_new_dataset.csv')
df
```

	Description	Sequence	Length
0	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898
1	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454
2	lcl|NM_001243795.2_cds_NP_001230724.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245
3	lcl|NM_001243794.2_cds_NP_001230723.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245
4	lcl|NM_001199241.2_cds_NP_001186170.1_1 [gene=...	ATGGAGCCTTCATCTCTTGAGCTGCCGGCTGACACAGTGCAGCGCA...	1398
...	...	...	...
4374	lcl|NM_001257309.1_cds_NP_001244238.1_1 [gene=...	ATGGCAGACGGCTGTTGTCCTGGAAACACCACAGCCATTCCAGCTG...	1026
4375	lcl|NM_001243116.2_cds_NP_001230045.1_1 [gene=...	ATGGACCTGCAGAATGACCTAGGCCAGACAGCCCTGCACCTGGCAG...	813
4376	lcl|NM_001256700.2_cds_NP_001243629.1_1 [gene=...	ATGATGAATTCTGACCAGAAGGCAGTGAAATTCCTGGCAAATTTTT...	996
4377	lcl|NM_001270500.2_cds_NP_001257429.1_1 [gene=...	ATGAGGGCCCTGGTGCTTCTGCTGTCCCTGTTCCTGCTGGGTGGCC...	942
4378	lcl|NM_001271831.2_cds_NP_001258760.2_1 [gene=...	ATGGATCCTGAGGTGACCTTGCTGCTGCAGTGCCCTGGCGGGGGCC...	813

4379 rows × 3 columns

Figure 09. Premier filtre d'ARNm

```
filtered_df = pd.read_csv('updated_ncRNA_dataset.csv')
filtered_df = filtered_df[filtered_df['Length'].between(100, 800)]
filtered_df.to_csv('ncRNA_newdataset.csv', index=False )
```

```
import pandas as pd
df = pd.read_csv('ncRNA_newdataset.csv')
df
```

	Description	Sequence	Length
0	RF00001_AF095839_1_346-228 5S_rRNA	GCGTACGGCCATACTATGGGGAATACACCTGATCCCGTCCGATTTC...	119
1	RF00001_AY245018_1_1-119 5S_rRNA	GCTATCGGCCATACTAAGCCAAATGCACCGGATCCCTTCCGAACTC...	119
2	RF00001_X52048_1_2-120 5S_rRNA	TGCTACGATCATACCACTTAGAAAGCACCCGGTCCCATCAGACCCC...	119
3	RF00001_M28193_1_1-119 5S_rRNA	AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTG...	119
4	RF00001_X14816_1_860-978 5S_rRNA	ACCAACGGCCATACCACGTTGAAAGTACCCAGTCTCGTCAGATCCT...	119
...	...	...	...
4710	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131
4711	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163
4712	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172
4713	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142
4714	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134

4715 rows × 3 columns

Figure 10. Premier filtre d'ARNnc

b) Le deuxième filtrage a été réalisé pour supprimer les séquences contenant le caractère "N", qui représente n'importe quel nucléotide et peut entraîner des erreurs dans les résultats. Voir les **figures 11 et 12.**

```
df = pd.read_csv('mRNA_new_dataset.csv')
df = df[~df['Sequence'].str.contains('N')]
df.to_csv('no_N.csv', index=False)
df
```

	Description	Sequence	Length
0	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898
1	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454
2	lcl|NM_001243795.2_cds_NP_001230724.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245
3	lcl|NM_001243794.2_cds_NP_001230723.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245
4	lcl|NM_001199241.2_cds_NP_001186170.1_1 [gene=...	ATGGAGCCTTCATCTCTTGAGCTGCCGGCTGACACAGTGCAGCGCA...	1398
...	...	...	...
4374	lcl|NM_001257309.1_cds_NP_001244238.1_1 [gene=...	ATGGCAGACGGCTGTTGTCCTGGAAACACCACAGCCATTCCAGCTG...	1026
4375	lcl|NM_001243116.2_cds_NP_001230045.1_1 [gene=...	ATGGACCTGCAGAATGACCTAGGCCAGACAGCCCTGCACCTGGCAG...	813
4376	lcl|NM_001256700.2_cds_NP_001243629.1_1 [gene=...	ATGATGAATTCTGACCAGAAGGCAGTGAAATTCCTGGCAAATTTTT...	996
4377	lcl|NM_001270500.2_cds_NP_001257429.1_1 [gene=...	ATGAGGGCCCTGGTGCTTCTGCTGTCCCTGTTCCTGCTGGGTGGCC...	942
4378	lcl|NM_001271831.2_cds_NP_001258760.2_1 [gene=...	ATGGATCCTGAGGTGACCTTGCTGCTGCAGTGCCCTGGCGGGGGCC...	813

4379 rows × 3 columns

Figure 11. Deuxième filtre de *dataset* d'ARNm

```
df = pd.read_csv('ncRNA_newdataset.csv')
df = df[~df['Sequence'].str.contains('N')]
df.to_csv('no_N_2.csv', index=False)
df
```

	Description	Sequence	Length
0	RF00001_AF095839_1_346-228 5S_rRNA	GCGTACGGCCATACTATGGGGAATACACCTGATCCCGTCCGATTTC...	119
1	RF00001_AY245018_1_1-119 5S_rRNA	GCTATCGGCCATACTAAGCCAAATGCACCGGATCCCTTCCGAACTC...	119
2	RF00001_X52048_1_2-120 5S_rRNA	TGCTACGATCATACCACTTAGAAAGCACCCGGTCCCATCAGACCCC...	119
3	RF00001_M28193_1_1-119 5S_rRNA	AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTG...	119
4	RF00001_X14816_1_860-978 5S_rRNA	ACCAACGGCCATACCACGTTGAAAGTACCCAGTCTCGTCAGATCCT...	119
...	...	...	...
4710	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131
4711	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163
4712	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172
4713	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142
4714	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134

4617 rows × 3 columns

Figure 12. Deuxième filtre de *dataset* d'ARNnc

c) Le troisième filtrage a été effectué pour supprimer les séquences en double dans chaque ensemble de données. Voir les **figures 13 et 14.**

```
df = pd.read_csv('no_N.csv')
df.drop_duplicates(subset='Sequence', keep='first', inplace=True)
df.to_csv('filtred_mRna_data.csv', index=False)
df
```

	Description	Sequence	Length
0	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898
1	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454
2	lcl|NM_001243795.2_cds_NP_001230724.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245
4	lcl|NM_001199241.2_cds_NP_001186170.1_1 [gene=...	ATGGAGCCTTCATCTCTTGAGCTGCCGGCTGACACAGTGCAGCGCA...	1398
5	lcl|NM_001252078.2_cds_NP_001239007.1_1 [gene=...	ATGGCGGAAGGCGGAGCGGCGGATCTGGACACCCAGCGGTCTGACA...	2946
...	...	...	...
4372	lcl|NM_001260509.2_cds_NP_001247438.1_1 [gene=...	ATGTCTGCACTCCGAAGGAAATTTGGGGACGATTATCAGGTAGTGA...	927
4374	lcl|NM_001257309.1_cds_NP_001244238.1_1 [gene=...	ATGGCAGACGGCTGTTGTCCTGGAAACACCACAGCCATTCCAGCTG...	1026
4376	lcl|NM_001256700.2_cds_NP_001243629.1_1 [gene=...	ATGATGAATTCTGACCAGAAGGCAGTGAAATTCCTGGCAAATTTTT...	996
4377	lcl|NM_001270500.2_cds_NP_001257429.1_1 [gene=...	ATGAGGGCCCTGGTGCTTCTGCTGTCCCTGTTCCTGCTGGGTGGCC...	942
4378	lcl|NM_001271831.2_cds_NP_001258760.2_1 [gene=...	ATGGATCCTGAGGTGACCTTGCTGCTGCAGTGCCCTGGCGGGGGCC...	813

3008 rows × 3 columns

Figure 13. Troisième filtre de *dataset* d'ARNm

```
df = pd.read_csv('no_N_2.csv')
df.drop_duplicates(subset='Sequence', keep='first', inplace=True)
df.to_csv('filtred_ncRNA_dataset.csv', index=False)
df
```

	Description	Sequence	Length
0	RF00001_AF095839_1_346-228 5S_rRNA	GCGTACGGCCATACTATGGGGAATACACCTGATCCCGTCCGATTTC...	119
1	RF00001_AY245018_1_1-119 5S_rRNA	GCTATCGGCCATACTAAGCCAAATGCACCGGATCCCTTCCGAACTC...	119
2	RF00001_X52048_1_2-120 5S_rRNA	TGCTACGATCATACCACTTAGAAAGCACCCGGTCCCATCAGACCCC...	119
3	RF00001_M28193_1_1-119 5S_rRNA	AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTG...	119
4	RF00001_X14816_1_860-978 5S_rRNA	ACCAACGGCCATACCACGTTGAAAGTACCCAGTCTCGTCAGATCCT...	119
...	...	...	...
4612	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131
4613	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163
4614	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172
4615	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142
4616	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134

4617 rows × 3 columns

Figure 14. Troisième filtre de *dataset* d'ARNnc

2.1.4. L'ajout de la colonne de type d'ARN dans chaque *dataset*, dans les **figures 15 et 16.**

```
df = pd.read_csv('filtred_mRna_data.csv')
df['RNA Type'] = 'coding'
df.to_csv('updated_mRNA_dataset.csv', index=False)
df
```

	Description	Sequence	Length	RNA Type
0	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898	coding
1	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454	coding
2	lcl|NM_001243795.2_cds_NP_001230724.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245	coding
3	lcl|NM_001199241.2_cds_NP_001186170.1_1 [gene=...	ATGGAGCCTTCATCTCTTGAGCTGCCGGCTGACACAGTGCAGCGCA...	1398	coding
4	lcl|NM_001252078.2_cds_NP_001239007.1_1 [gene=...	ATGGCGGAAGGCGGAGCGGCGGATCTGGACACCCAGCGGTCTGACA...	2946	coding
...	...	...	...	...
3003	lcl|NM_001260509.2_cds_NP_001247438.1_1 [gene=...	ATGTCTGCACTCCGAAGGAAATTTGGGGACGATTATCAGGTAGTGA...	927	coding
3004	lcl|NM_001257309.1_cds_NP_001244238.1_1 [gene=...	ATGGCAGACGGCTGTTGTCCTGGAAACACCACAGCCATTCCAGCTG...	1026	coding
3005	lcl|NM_001256700.2_cds_NP_001243629.1_1 [gene=...	ATGATGAATTCTGACCAGAAGGCAGTGAAATTCCTGGCAAATTTTT...	996	coding
3006	lcl|NM_001270500.2_cds_NP_001257429.1_1 [gene=...	ATGAGGGCCCTGGTGCTTCTGCTGTCCCTGTTCCTGCTGGGTGGCC...	942	coding
3007	lcl|NM_001271831.2_cds_NP_001258760.2_1 [gene=...	ATGGATCCTGAGGTGACCTTGCTGCTGCAGTGCCCTGGCGGGGGCC...	813	coding

3008 rows × 4 columns

Figure 15. *Dataframe* d'ARNm *dataset* avec la colonne de type d'ARN

```
df = pd.read_csv('filtred_ncRNA_dataset.csv')
df['RNA Type'] = 'non-coding'
df.to_csv('updated_ncRNA_data.csv', index=False)
df
```

	Description	Sequence	Length	RNA Type
0	RF00001_AF095839_1_346-228 5S_rRNA	GCGTACGGCCATACTATGGGGAATACACCTGATCCCGTCCGATTTC...	119	non-coding
1	RF00001_AY245018_1_1-119 5S_rRNA	GCTATCGGCCATACTAAGCCAAATGCACCGGATCCCTTCCGAACTC...	119	non-coding
2	RF00001_X52048_1_2-120 5S_rRNA	TGCTACGATCATACCACTTAGAAAGCACCCGGTCCCATCAGACCCC...	119	non-coding
3	RF00001_M28193_1_1-119 5S_rRNA	AGTTACGGCCATACCTCAGAGAATATACCGTATCCCGTTCGATCTG...	119	non-coding
4	RF00001_X14816_1_860-978 5S_rRNA	ACCAACGGCCATACCACGTTGAAAGTACCCAGTCTCGTCAGATCCT...	119	non-coding
...	...	...	...	...
4612	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131	non-coding
4613	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163	non-coding
4614	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172	non-coding
4615	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142	non-coding
4616	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134	non-coding

4617 rows × 4 columns

Figure 16. *Dataframe* d'ARNnc *dataset* avec la colonne de type d'ARN

2.1.5. Fusion des deux ensembles de données, afin de les traités ensemble. Voir la **figure 17.**

```
mRNA_df = pd.read_csv('updated_mRNA_dataset.csv')
ncRNA_df = pd.read_csv('updated_ncRNA_data.csv')
merged_df = pd.concat([mRNA_df, ncRNA_df], ignore_index=True)
merged_df.to_csv('dataset.csv', index=False)
merged_df
```

	Description	Sequence	Length	RNA Type
0	lcl|NM_001252624.2_cds_NP_001239553.1_1 [gene=...	ATGTACAACATGCGGCGATTAAGTCTTTCACCCACCTTTTCAATGG...	2898	coding
1	lcl|NM_001243101.2_cds_NP_001230030.1_1 [gene=...	ATGGATGTCTCTCTTTGCCCAGCCAAGTGTAGTTTCTGGCGGATTT...	2454	coding
2	lcl|NM_001243795.2_cds_NP_001230724.1_1 [gene=...	ATGACCAAGGCCCGGCTGTTCCGGCTGTGGCTGGTGCTGGGGTCGG...	1245	coding
3	lcl|NM_001199241.2_cds_NP_001186170.1_1 [gene=...	ATGGAGCCTTCATCTCTTGAGCTGCCGGCTGACACAGTGCAGCGCA...	1398	coding
4	lcl|NM_001252078.2_cds_NP_001239007.1_1 [gene=...	ATGGCGGAAGGCGGAGCGGCGGATCTGGACACCCAGCGGTCTGACA...	2946	coding
...	...	...	...	...
7620	RF02535_AFEY01343643_1_18075-17945 IRES	ACTTCCAATGCAATGGCTGCAGTGAAGCTATAATTATAGCCTTGTA...	131	non-coding
7621	RF02535_AAPE02009951_1_24083-24245 IRES	ATTCCCAGTGCTGCACCGAGAGGACCTGTCTCCTGTGGACTGGAAG...	163	non-coding
7622	RF02535_ABQO011108623_1_28-199 IRES	AGTGCAACGGCTGCACCGAAGGCACAATCGTAGCCTTGTATTTCAC...	172	non-coding
7623	RF02535_AAPE02044716_1_11582-11441 IRES	ATTCCCGCTGCTGCACAGAGAGGACCCGTGTCCCGTGGACTGGGAG...	142	non-coding
7624	RF02535_AEYP01041088_1_4708-4575 IRES	AGTCCCAATATTGCATCCAACAGGATTTGGAATTTCTAGAGAATTG...	134	non-coding

7625 rows × 4 columns

Figure 17. La fusion des deux *datasets* utilisant de la fonction concat()

2.2. Apprentissage

Pour pouvoir utiliser les données dans le modèle d'apprentissage, il est nécessaire de les transformer, car le modèle d'apprentissage ne prend pas en charge les chaînes de caractères. Voici les transformations qui ont été effectuées :

2.2.1. *One hot encoding*

Une méthode de conversion des données pour les préparer à un algorithme et obtenir une meilleure prédiction. Avec *one hot encoding*, nous convertissons chaque valeur catégorielle en une nouvelle colonne catégorielle et attribuons une valeur binaire de 1 ou 0 à ces colonnes. Chaque valeur entière est représentée par un vecteur binaire. En utilisant *one hot encoder* sur la colonne « *RNA Type* » le résultat est présenté dans le **tableau 06**.

Tableau 06. Le résultat de *one hot encoder*

	ARN codant	ARN non-codant
ARN codant	1	0
ARN non-codant	0	1

2.2.2. *Padding*

Consiste à ajouter des zéros à nos chaînes de caractères dans le but d'avoir des séquences ayant toutes la même taille.

- Pour homogénéiser la taille des séquences, le *padding* a été appliqué en ajoutant des zéros aux séquences de longueur inférieure à 3000 nucléotides. Cela permet de les aligner sur la taille désirée. Voir son utilisation dans la **figure 18.**

2.2.3. *Tokenizer*

Le *tokenizer* est un outil qui divise les textes en morceaux plus petits, tels que les mots ou les caractères individuels. Il permet de transformer le texte en éléments distincts qui peuvent être utilisés pour l'analyse et le traitement des données textuelles. Voir son utilisation dans la **figure 18.**

```
sentences = np.array(df['Sequence'])
```

```
max_length = 3000
padding_type='same'
```

```
tokenizer = Tokenizer(char_level=True)
tokenizer.fit_on_texts(sentences)

word_index = tokenizer.word_index

X = tokenizer.texts_to_sequences(sentences)
X = sequence.pad_sequences(X, maxlen=max_length)
```

```
word_index
```

```
{'a': 1,
, 'g': 2,
, 'c': 3,
, 't': 4,
, 'r': 5,
, 'y': 6,
, 'k': 7,
, 'm': 8,
, 's': 9,
, 'w': 10,
, 'h': 11}
```

Figure 18 : L'utilisation du *padding* et *tokenizer*

2.2.4. Construction du modèle

Une fois les données préparées, nous procédons à la construction du modèle de type *Sequential* **(figure 19).** Le modèle est composé des couches suivantes :

1. Couche *Embedding* : Cette couche convertit chaque vecteur en une matrice en attribuant des valeurs numériques à chaque élément du vecteur. Les options de cette couche incluent le nombre de valeurs possibles pour chaque élément du vecteur, la taille de l'*embedding* et la taille du vecteur.

2. Couche *Conv1D* : Il s'agit d'un réseau de neurones convolutionnel (*CNN*) composé de 8 filtres, avec un kernel de taille 3 unités et une fonction d'activation de type ReLU.

3. Couche *MaxPooling1D* : Cette couche effectue un regroupement maximal en sélectionnant les éléments les plus importants pour la classification. Elle produit une matrice contenant les caractéristiques les plus significatives de la sortie de la couche *CNN*.

4. Couche *Flatten* : Cette couche convertit la sortie de la couche *MaxPooling1D*, qui est une matrice, en un vecteur.

5. La couche Dense : Cette couche représente un réseau de neurones artificiel standard qui est responsable de la prise de décision pour la prédiction.

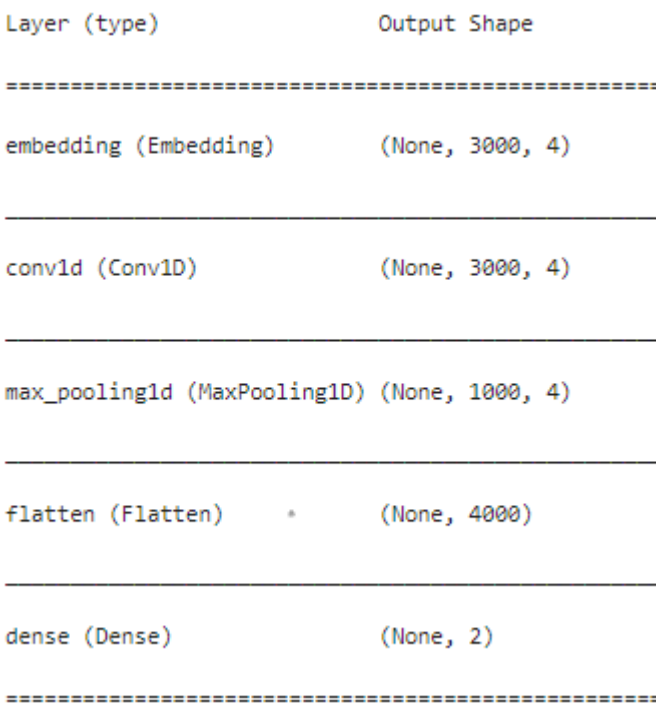


Figure 19. Le modèle utilisé

2.2.5. Répartition des données

La répartition des données est effectuée comme suit : 90% des données sont utilisées pour l'apprentissage du modèle, tandis que les 10% restants sont réservés aux tests.

Ces 10% de données de test sont ensuite divisés en deux parties égales : 50% sont utilisés pour le test pendant la phase d'apprentissage du modèle, tandis que les 50% restants sont réservés à l'évaluation du modèle une fois l'apprentissage terminé.

2.2.6. L'Apprentissage du modèle

Pour lancer l'apprentissage, la fonction "fit" de *TensorFlow* est utilisée. Cette fonction prend en compte les données d'apprentissage et de test, ainsi que le nombre d'epoch souhaité

(une epoch correspond à un cycle d'apprentissage sur l'ensemble des données) que le modèle doit parcourir pour terminer son entraînement.

RESULTATS ET DISCUSSION

RESULTATS ET DISCUSSION

1. Résultats

1.1.Résultats du prétraitement

Plusieurs étapes ont été réalisées lors du prétraitement des données afin de nettoyer le dataset et de conserver uniquement les informations pertinentes pour l'apprentissage. Les résultats obtenus à chaque étape sont les suivants :

- Le dataset initial comprenait un total de 7129 séquences d'ARN codants et 6320 séquences d'ARN non-codants chez l'*homosapiens*.
- Après le prétraitement des séquences d'ARN **(figure 20)**, nous avons obtenu un ensemble de 7625 séquences d'ARN, 3008 séquences d'ARN codants et 4617 séquences d'ARN non-codants.

```
non-coding    4617
,coding        3008
,Name: RNA Type, dtype: int64
```

```
df.shape
```

```
(7625, 4)
```

Figure 20. Le nombre de séquences après le prétraitement

1.2.Matrice de confusion

Une fois l'apprentissage terminé, le modèle a été évalué pour vérifier son efficacité en utilisant une matrice de confusion. La matrice de confusion est un outil qui permet d'évaluer les performances de classification d'un modèle en comparant les prédictions du modèle avec les véritables étiquettes de classe.

La matrice de confusion obtenue pour notre modèle est présentée dans la **figure 21**. Elle affiche les prédictions du modèle pour les classes d'ARN codants et non codants, ainsi que les véritables étiquettes de classe. Les éléments de la matrice sont comptés en fonction de la correspondance entre les prédictions du modèle et les véritables étiquettes de classe.

Dans notre matrice de confusion, les colonnes représentent les classes prédites par le modèle, tandis que les lignes représentent les véritables classes d'ARN. Les valeurs dans la matrice indiquent le nombre d'instances appartenant à chaque combinaison de classe prédite et véritable classe d'ARN.

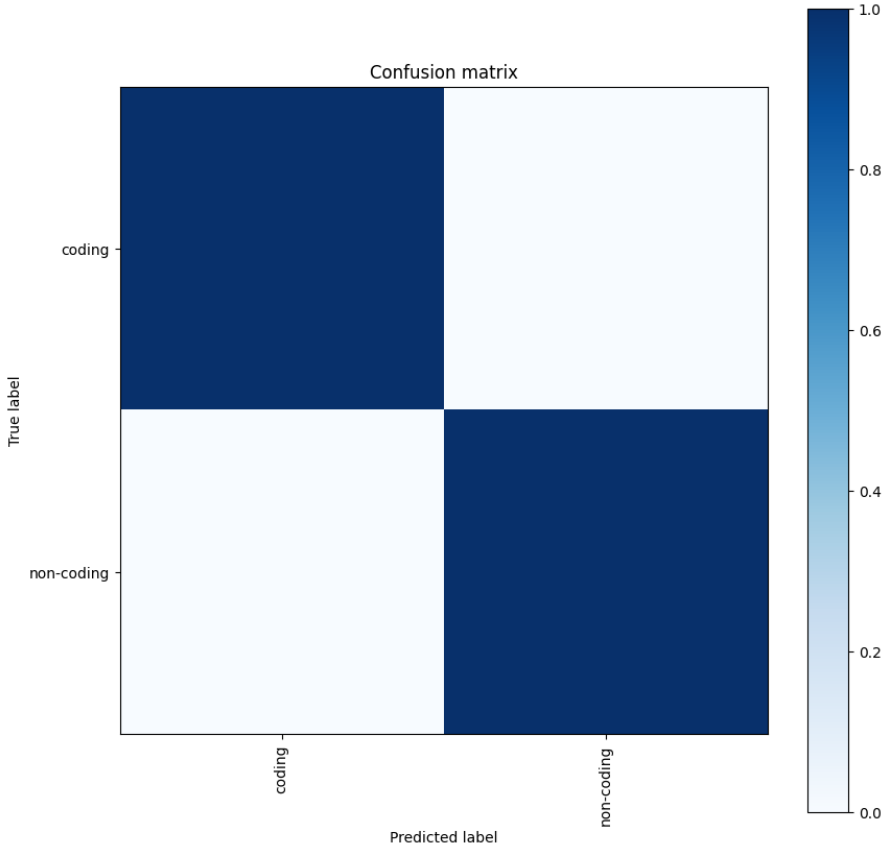


Figure 21. Matrice de confusion du modèle

1.3.Précision (*Accuracy*)

Pour évaluer les performances de notre modèle, nous avons calculé la précision (*Accuracy*) sur les données d'entraînement et de test. La précision est une mesure couramment utilisée pour évaluer la performance d'un modèle de classification et elle représente le pourcentage de prédictions correctes par rapport au nombre total d'instances.

Dans notre étude, nous avons utilisé un *dataset* de 7625 séquences d'ARN pour entraîner et tester notre modèle. Les résultats obtenus pour la précision sont les suivants :

- Précision sur les données d'entraînement (train-acc) : 0.9998542698921598

- Précision sur les données de test (test-acc) : 0.9986893840104849

2. Discussion

2.1.Discussion des résultats de l'apprentissage

-Précision sur les données d'entraînement (*train-acc*) : 0.9998542698921598. Cela signifie que le modèle a obtenu une précision élevée de près de 99,99% lors de l'évaluation sur les données d'entraînement. Cela suggère que le modèle a appris avec succès à reconnaître les classes d'ARN codants et non-codants dans l'ensemble d'entraînement.

- Précision sur les données de test (*test-acc*) : 0.9986893840104849. Cette mesure indique que le modèle a également obtenu une précision élevée de près de 99,87% lors de l'évaluation sur les données de test. Cela démontre que le modèle généralise bien et est capable de prédire avec précision les classes d'ARN codants et non-codants sur de nouvelles données.

Ces résultats de précision élevée suggèrent que l'approche de *deep learning* utilisée dans notre étude est prometteuse pour l'identification des classes d'ARN codants et non-codants.

Dans notre travail, nous avons utilisé un ensemble de 7625 séquences d'ARN pour développer un modèle prédictif visant à distinguer les ARN codants des ARN non-codants. Notre modèle s'appuie sur un réseau neuronal artificiel spécialement conçu pour cette tâche. Grâce à ce modèle, nous sommes en mesure d'identifier le type d'ARN, qu'il s'agisse d'un ARN codant pour une protéine ou d'un ARN non-codant.

L'importance de notre travail réside dans la prédiction du type d'ARN, c'est-à-dire s'il code ou non pour une protéine. Notre méthode a réussi à classer les ARN codants et non-codants avec une précision et une exactitude de 99%. Ces résultats obtenus démontrent que notre modèle peut jouer un rôle efficace dans la prédiction et la classification des ARN.

Plusieurs travaux de recherche ont été réalisés dans le domaine de l'identification des classes d'ARN codants et non codants en utilisant des approches basées sur l'apprentissage profond. Ces travaux partagent l'objectif commun d'améliorer la compréhension des mécanismes de régulation génétique en permettant une classification précise des ARN.

Dans une étude similaire, les chercheurs ont utilisé un modèle de réseau de neurones récurrents (RNN) pour la classification des ARN [59]. Ils ont prétraité les données en utilisant l'encodage one-hot et le padding, tout comme nous l'avons fait dans notre approche. Leur modèle a été entraîné sur un grand ensemble de données comprenant à la fois des ARN codants et non codants, et ils ont obtenu des résultats prometteurs avec une précision de classification élevée [59].

Une autre étude a exploré l'utilisation de réseaux de neurones convolutionnels (CNN) pour l'identification des classes d'ARN [60]. Les chercheurs ont également utilisé l'encodage one-hot et le padding pour prétraiter les données, et ils ont construit un modèle CNN avec des couches de convolution et de pooling. Leur modèle a été entraîné sur un ensemble de données contenant différentes classes d'ARN, et les résultats ont montré une précision élevée dans la classification des ARN codants et non codants [60].

Il convient de noter que différentes approches de prétraitement des données et d'architecture de modèles peuvent être utilisées dans ces travaux similaires. Certains chercheurs ont exploré l'utilisation de techniques avancées telles que les réseaux de neurones récurrents bidirectionnels (Bi-RNN) ou les modèles d'apprentissage par transfert pour améliorer les performances de classification [61].

De plus, certains travaux ont utilisé des ensembles de données plus vastes et plus diversifiés pour évaluer la généralisation de leurs modèles [62].

Dans l'ensemble, ces travaux similaires confirment l'efficacité des approches basées sur l'apprentissage profond pour l'identification des classes d'ARN codants et non codants. Ils démontrent également l'importance des techniques de prétraitement des données et du choix approprié de l'architecture du modèle pour obtenir de bons résultats de classification. Des études futures pourraient se concentrer sur la comparaison et l'évaluation approfondie de différentes approches et architectures, ainsi que sur l'application de ces modèles à des ensembles de données plus vastes et à des problèmes de classification plus complexes.

CONCLUSION

CONCLUSION

En conclusion, ce mémoire a présenté une approche basée sur l'apprentissage profond (Deep Learning) pour l'identification des classes d'ARN codants et non codants. L'utilisation d'un modèle de réseau de neurones convolutionnels (CNN) combiné à des techniques de prétraitement telles que l'encodage one-hot, le *padding* et le *tokenizer* s'est révélée efficace pour la classification des ARN.

Les résultats obtenus ont démontré une précision élevée sur les données d'entraînement et de test, ce qui indique la capacité du modèle à généraliser et à effectuer des prédictions précises. Cela suggère que l'approche proposée peut être utilisée avec succès dans des études plus larges et pour l'analyse de grandes quantités de données d'ARN.

Il convient de souligner que les travaux similaires ont également exploré différentes architectures de modèles, prétraitement des données et ensembles de données pour l'identification des classes d'ARN. Certaines études ont utilisé des réseaux de neurones récurrents (RNN) ou des techniques avancées telles que les réseaux de neurones récurrents bidirectionnels (Bi-RNN) pour améliorer les performances de classification. De plus, l'utilisation d'ensembles de données vastes et diversifiés a été privilégiée pour évaluer la généralisation des modèles.

En combinant les connaissances acquises à partir de ces travaux similaires, il est possible d'améliorer davantage l'approche proposée en explorant de nouvelles architectures de modèles, des techniques de prétraitement avancées et des ensembles de données plus vastes. Cela pourrait contribuer à une meilleure compréhension des mécanismes de régulation génétique et à l'identification précise des classes d'ARN codants et non codants.

En conclusion, l'utilisation de l'apprentissage profond pour l'identification des classes d'ARN offre une méthode prometteuse qui peut ouvrir de nouvelles perspectives dans la recherche en biologie moléculaire et la compréhension des processus génétiques.

CONCLUSION

REFERENCES BIBLIOGRAPHIQUES

REFERENCES BIBLIOGRAPHIQUES

[1] Smith MA, Gesell T, Stadler PF. Mattick JS. The functional genomics of noncoding RNA. Science. 2019;324(5923):257-262.

[2] Esteller M. Non-coding RNAs in human disease. Nat Rev Genet. 2011;12(12):861-874.

[3] Ma J, Yu MK, Fong S. Deep learning in bioinformatics. J Mol Biol. 2020;432(1):323-339.

[4] Angermueller C, Pärnamaa T, Parts L, Stegle O. Deep learning for computational biology. Mol Syst Biol. 2021;17(6):e9729.

[5] Gurtan AM, Rinn JL, Garber M. The expanding scope of noncoding RNAs in the nucleus. Nat Struct Mol Biol. 2016;23(11):1114-1121.

[6] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-444.

[7] Mattick JS, Makunin IV. Non-coding RNA. Hum Mol Genet. 2006;15(suppl_1):R17-R29.

[8] Guttman M, Rinn JL. Modular regulatory principles of large non-coding RNAs. Nature. 2012;482(7385):339-346.

[9]Jacquard A. L'ADN et la vie. Le Portique. 2000;5. Available from: http://journals.openComparaison avec d'autres travaux. Accessed May 7, 2023.

[10] Lamoril J, Ameziane N, Deybach J-C, Bouizegarène P, Bogard M. Le monde complexe et mouvant des ARN. Première partie. Immuno-analyse et biologie spécialisée. 2010;25:4-25.

[11] Lodish H, Berk A, Matsudaira P, Kaiser CA, Krieger M, Scott MP, Darnell J. Biologie moléculaire de la cellule (3e éd.). Bruxelles, Belgique: de Boeck; 2005. ISBN 978-2804148027.

[12] Lamoril J, Ameziane N, Deybach J-C, Bouizegarène P, Bogard M. Le monde complexe et mouvant des ARN. Première partie. Immuno-analyse et biologie spécialisée. 2010;25:4-25.

[13] Housset C, Raisonnier A. Cours de biologie moléculaire du CHU Pitié Salpêtrière. 2006-2007.

[14] Moore MJ, Proudfoot NJ. Pre-mRNA processing reaches back to transcription and ahead to translation. Cell. 2009;136:688-700.

[15] Dujardin G, Daguenet É, Bernard DG, Flodrops M, Durand S, Chauveau A, ... Corcos L. L'épissage des ARN pré-messagers : quand le splicéosome perd pied. Médecine/Sciences. 2016;32(9):1103-1110.

[16] Bhartiya D, Scaria V. Genomic variations in non-coding RNAs: Structure, function and regulation. Genomics. Advance online publication. 2016. doi: 10.1016/j.ygeno.2016.01.005.

[17] Fontaine A. Classification d'ARN codants et d'ARN non-codants. Bio-informatique [q-bio.QM]. Université des Sciences et Technologie de Lille - Lille I, France; 2009.

[18] Frith MC, Pheasant M, Mattick JS. The amazing complexity of the human transcriptome. European Journal of Human Genetics. 2005;13:894-897.

[19] Van Lint S, Heirman C, Thielemans K, Breckpot K. mRNA: From a chemical blueprint for protein production to an off-the-shelf therapeutic. Human Vaccines & Immunotherapeutics. 2013;9(2):265-274.

[20] Kushner SR. mRNA Decay in Prokaryotes and Eukaryotes: Different Approaches to a Similar Problem. Critical Review, IUBMB Life. 2004;56(10):585-594.

[21] Ponting CP, Oliver PL, Reik W. Evolution and functions of long noncoding RNAs. Cell. 2009;136:629-641.

[22] Derrien T, Guigó R. Des longs ARN non codants humains activateurs de la transcription des gènes [Long non-coding RNAs with enhancer-like function in human cells]. Médecine/Sciences. 2011;27(4):359-361. doi: 10.1051/medsci/2011274009

[23] Sharp SJ. Structure and Transcription of Eukaryotic tRNA Genes. CRC Critical Reviews in Biochemistry. Year;19:107-144.

[24] Warner JR. The economics of ribosome biosynthesis in yeast. Reviews. 1999;437-440.

[25] Hüttenhofer A, Schattner P. The principles of guiding by RNA: chimeric RNA-protein enzymes. Nature Reviews Genetics. 2006;7:475-482.

[26] Lamoril J, Bouizegarène P, Bogard M. Le monde complexe et mouvant des ARN. Seconde partie: les microARNs. Immuno-analyse et biologie spécialisée. 2010;25:219-240.

[27] Dong H, Lei J, Ding L, Wen Y, Ju H, Zhang X. MicroRNA: Function, Detection, and Bioanalysis. Chemical Reviews. Year;Volume(Issue):Page numbers.

[28] Liang J, Wen J, Huang Z, Chen X-p, Zhang B-x, Chu L. Small Nucleolar RNAs: Insight Into Their Function in Cancer. Review. Date;Page numbers.

[29] Tutar L. Introductory Chapter: Noncoding RNAs - A Brief Introduction. In: Noncoding RNAs in Health and Disease. Intech; 2022. p. 1-10. doi: 10.5772/intechopen.108165.

[30] Gurman SP, Hamilton AJ. Improved northern blot method for enhanced detection of small RNA. Nature Protocols. 2008;3(6):1077. doi: 10.1038/nprot.2008.67

[31] Hernandez-Rodriguez P, editor. Polymerase Chain Reaction. In: InTech; 2012. doi: 10.5772/2204

[32] Jensen E. Technical Review: In Situ Hybridization. The Anatomical Record. Advance online publication. Year. doi: 10.1002/ar.22944

[33] Wang Z, Gerstein M, Snyder M. RNA-Seq: A revolutionary tool for transcriptomics. Nature Reviews Genetics. 2009;10:57-63.

[34] Sealfon SC, Chu TT. RNA and DNA microarrays. Methods in Molecular Biology. 2011;671:3-34. doi: 10.1007/978-1-59745-551-0_1

[35] Liehr T, editor. Fluorescence In Situ Hybridization (FISH) Application Guide. Year. pp. 107-169.

[36] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-444.

[37] Gurney K. An introduction to neural networks. CRC Press. 1997.

[38] Wang Z. Understanding Convolutional Neural Networks for NLP. arXiv preprint arXiv:1809.08895. n.d.

[39] Goodfellow I, Bengio Y, Courville A. Deep Learning. MIT Press. 2016.

[40] Teh YW, Hinton GE. Rate-coded Restricted Boltzmann Machines for Face Recognition. Proceedings of the 13th Annual Conference on Neural Information Processing Systems (NIPS 2000). Year;Pages.

[41] Hochreiter S, Schmidhuber J. Long Short-term Memory. Neural Computation. 1997;9(8):1735-1780.

[42] Metz C. Apple is bringing the AI Revolution to your Phone. Wired. 2016.

[43] Gers FA, Schmidhuber J, Cummins FA. Learning to Forget: Continual Prediction with LSTM. Neural Computation. 2000;12(10):2451-2471.

[44] Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in Bioinformatics. 2017;18(5):851-869.

[45] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction (2nd edition). Springer. 2009.

[46] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015;521(7553):436-444.

[47] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press. 2016.

[48] Ruder S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:1609.04747. 2016.

[49] Arlot S, Celisse A. A survey of cross-validation procedures for model selection. Statistics Surveys. 2010;4:40-79.

[50] Powers DM. Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. Journal of Machine Learning Technologies. 2011;2(1):37-63.

[51] Olah C, Mordvintsev A, Schubert L. Feature visualization. Distill. 2017;2(11):e7.

[52] Poplin R, Chang P-C, Alexander D, Schwartz S, Colthurst T, Ku A, Newburger D, Dijamco J, Nguyen N, Afshar PT, Gross SS, Dorfman L, McLean CY, DePristo MA. Universal variant calling with deep neural networks. Nature Biotechnology. 2017;35(10):915-921.

[53] Webb S. Deep learning for biology. Nature. 2018;554:555-557.

[54] Torrisi M, Pollastri G, Le Q. Deep learning methods in protein structure prediction. Computational and Structural Biotechnology Journal. 2020;18:1301-1310.

[55] Hudson IL. Data Integration Using Advances in Machine Learning in Drug Discovery and Molecular Biology: Artificial Neural Networks (Vol. 2190). In: Advances in Experimental Medicine and Biology. Springer; 2021. pp. 167-184. ISBN: 978-1-0716-0825-8.

[56] Jiang Y, Luo J, Huang D, Liu Y, Li D. Machine Learning Advances in Microbiology: A Review of Methods and Applications. Frontiers in Microbiology. 2022;13:925454. doi: 10.3389/fmicb.2022.925454.

[57] Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in Bioinformatics. 2017;18(5):851-869. doi: 10.1093/bib/bbw068.

[58] Ching T, Himmelstein DS, Beaulieu-Jones BK, Kalinin AA, Do BT, Way GP, Greene CS. Opportunities and obstacles for deep learning in biology and medicine. Journal of the Royal Society Interface. 2018;15(141):20170387. doi: 10.1098/rsif.2017.0387.

[59] Xiong Y, Jin J, Zhang J, et al. DeepCRISPR: optimized CRISPR guide RNA design by deep learning. Genome Biol. 2018;19(1):80.

[60] Zhou J, Troyanskaya OG. Predicting effects of noncoding variants with deep learning-based sequence model. Nat Methods. 2015;12(10):931-934.

[61] Chen X, Li Y, Gao J, Zhang X, Sun L. Coding-Noncoding RNA Classification with Multi-Attention Convolutional Neural Network. IEEE/ACM Trans Comput Biol Bioinform. 2020;17(3):1062-1070.

[62] Zhou J, Theesfeld CL, Yao K, Chen KM, Wong AK, Troyanskaya OG. Deep learning sequence-based ab initio prediction of variant effects on expression and disease risk. Nat Genet. 2018;50(8):1171-1179.

Soutenue le :	Présenté par :
19/06/2023	BENMENIA Nouha & SMAILI Khaoula

Thème :

Approche DL pour l'identification des classes d'ARN codants et non codants

Mémoire Présenté en vue de l'obtention du Diplôme de Master en :

Bioinformatique

Domaine : Science de la nature et la vie

Département de Biologie Appliquée

Ce mémoire présente une approche basée sur l'apprentissage profond (Deep Learning) pour l'identification des classes d'ARN codants et non codants. L'ARN est une molécule essentielle dans la biologie, et sa classification en ARN codants (ARNm) et ARN non codants est cruciale pour comprendre les mécanismes de régulation génétique. Dans ce travail, nous avons utilisé deux ensembles de données : un ensemble d'ARN non codants provenant de la base de données Rfam, et un ensemble d'ARN codants provenant de la base de données RefSeq. Les données ont été prétraitées pour les rendre compatibles avec les algorithmes d'apprentissage automatique, notamment en utilisant l'encodage one-hot, le padding et le tokenizer. Ensuite, un modèle d'apprentissage en Deep Learning a été construit en utilisant une architecture de réseau de neurones convolutionnels (CNN) avec des couches d'embedding, de convolution, de max pooling et de classification. Le modèle a été entraîné sur les données d'ARN codants et non codants, en utilisant une répartition des données avec 90% pour l'apprentissage et 10% pour les tests.Les performances du modèle ont été évaluées en utilisant la mesure de précision (accuracy), qui représente le pourcentage de prédictions correctes par rapport au nombre total d'instances. Les résultats obtenus ont démontré une précision élevée sur les données d'entraînement (99,99%) et sur les données de test (99,87%), ce qui indique une performance prometteuse du modèle dans l'identification des classes d'ARN codants et non codants.En conclusion, cette approche basée sur l'apprentissage profond offre une méthode efficace pour l'identification des classes d'ARN codants et non codants. Les résultats obtenus suggèrent que cette approche pourrait être utilisée dans des études plus larges et pour l'analyse de grandes quantités de données d'ARN, contribuant ainsi à une meilleure compréhension des mécanismes de régulation génétique.

Mots clés : ARN ; Classification ; Apprentissage profond ; Réseau de neurones convolutionnels, Encodage one-hot.

Jury d'évaluation :

Encadreur : Dr. Khaled BOULAHROUF

Examinateur 1 : Pr. Ines BELLIL

Examinateur 2 : Dr. Amira GHERBOUDJ