الجمهوريـــــــــــــــة الجزائريـــــــــة الديمقراطيـــــــة الشعبيـــــة
وزارة التعليـــــــم العـــــــالي و البحـــــــث العلـــــمي
جامعة وهران للعلوم والتكنولوجيا محمد بوضياف
كلية الرياضيات و الاعلام الالي
République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur Et de la Recherche Scientifique
Université des Sciences et de la Technologie d'Oran Mohamed BOUDIAF
Faculté des Mathématiques et Informatique

Département *: Informatique*

Mémoire de fin d'études

L'apport de l'apprentissage automatique dans la prédiction de la structure des protéines

Pour l'obtention du diplôme de **Master**

Domaine : **Mathématiques – Informatique**

Filière : **Informatique**

Spécialité : **Systèmes d'Information et Données (SID)**

Présenté le : 02/ 06 /22
Par :

-BRAIKIA Houria

Jury	Nom et Prénom	Grade	Université
Président	Mme BOUZIANE Hafida	MCA	USTOMB
Encadrant	Mme NAIT BAHLOUL Sarah	MCB	USTOMB
Examinateur	Mme SAD HOUARI Nawal	MCA	USTOMB

2021/2022

Remerciement

En tout premier lieu, je remercie Dieu, tout puissant, de m'avoir donné la force pour survivre, ainsi que l'audace pour dépasser toutes les difficultés.
Je tiens aussi à adresser mes remerciements à ma famille, et plus précisément à mes parents qui m'ont toujours soutenus et encouragés. Ce présent travail a pu voir le jour grâce à leur soutien.
Je tiens à exprimer toute ma reconnaissance à l'encadrante du projet Madame Sarah NAIT BAHLOUL. Je la remercie de m'avoir encadré, orienté, aidé et conseillé.
Afin de n'oublier personne, mes vifs remerciements s'adressent à tous ceux qui m'ont aidée à la réalisation de ce modeste travail.

Dédicace

Ce modeste travail est dédié à mon oncle Dr Houari BOUKERMA décédé trop tôt, mon exemple éternel, mon soutien moral et source de joie et de bonheur, celui qui m'a toujours poussé et motivé dans mes études,
aussi à mes parents et à mon frère et ma sœur.

ملخص

البروتينات ضرورية للحياة ، فهي تهتم بكل الوظائف تقريبًا في الكائن الحي. البروتينات هي جزيئات كبيرة معقدة ، تتكون من سلاسل من الأحماض الأمينية ، بمجرد صنعها ، تنثني على نفسها لتتخذ شكلًا فريدًا. يرتبط شكل البروتين ارتباطًا وثيقًا بوظيفته ، والقدرة على التنبؤ بهذا الهيكل لها أهمية قصوى ، سواء في تحديد دوره في الجسم ، أو في تطوير الأدوية للأمراض التي يُعتقد أنها ناجمة عن البروتينات غير المطوية بشكل صحيح ، مثل مرض الزهايمر ومرض باركنسون. يحاول الباحثون منذ سنوات عديدة إيجاد طريقة لتحديد بنية البروتينات باستخدام مجموعة متنوعة من التقنيات التجريبية ، مثل الرنين المغناطيسي النووي وعلم البلورات بالأشعة السينية التي تعتمد على الكثير من التجربة والخطأ ، والتي يمكن أن تستغرق سنوات من العمل واستخدام المعدات المتخصصة بملايين الدولارات. هذا هو السبب في أن علماء الأحياء يتجهون إلى أساليب الذكاء الاصطناعي كبديل لهذه العملية. تم التعرف على أحدث إصدار من نظام الفافولد كحل لهذا التحدي الكبير من قبل منظمي مسابقة الكاسب. يوضح هذا الاختراع التأثير الذي يمكن أن يحدثه الذكاء الاصطناعي على الاكتشاف العلمي وقدرته على تسريع التقدم بشكل كبير في بعض المجالات الأساسية التي تشرح وتشكل عالمنا.

في هذه الأطروحة ، سوف نقدم مناهج مختلفة تعتمد على التعلم العميق المقترح لحل هذه المشكلة. وبالتالي ، سوف نقدم نهجًا يعتمد على بنية التشفير التلقائي التي تتنبأ بمصفوفة الاتصال من تسلسل حمض أميني معين والتعديلات التي أجريناها على هذا النهج.

الكلمات الدالة : البروتينات ، الأحماض الأمينية ، التسلسل ، هياكل البروتين ، الذكاء الاصطناعي ، التعلم العميق ، التشفير التلقائي ، مصفوفة الاتصال.

Résumé

Les protéines sont essentielles à la vie, elles s'occupent d'à peu près tout dans un organisme vivant. Ce sont de grandes molécules complexes, constituées de chaînes d'acides aminés, une fois fabriquées, elles se replient sur elles-même pour adopter une forme unique. La forme d'une protéine est étroitement liée à sa fonction, et la capacité de prédire cette structure est d'une importance primordiale, à la fois pour déterminer son rôle dans l'organisme, ainsi que pour développer des médicaments pour des maladies supposées être causées par des protéines mal repliées, telles que la maladie d'Alzheimer et la maladie de Parkinson. Des recherches scientifiques essaient depuis de nombreuses années de trouver une méthode pour déterminer la structure des protéines en utilisant une variété de techniques expérimentales, telles que la résonance magnétique nucléaire et la cristallographie aux rayons X qui dépendent de nombreux essais et erreurs, qui peuvent prendre des années de travail et nécessitent l'utilisation d'équipements spécialisés par des millions de dollars. C'est pourquoi les biologistes se tournent vers les méthodes d'intelligence artificielle (IA) comme alternative à ce processus. Dans une avancée scientifique majeure, la dernière version

du système d'IA AlphaFold a été reconnue comme une solution à ce grand défi par les organisateurs de la compétition CASP. Cette percée démontre l'impact que l'IA peut avoir sur la découverte scientifique et son potentiel pour accélérer considérablement les progrès dans certains des domaines les plus fondamentaux qui expliquent notre monde.

Dans ce mémoire, nous allons présenter différentes approches basées sur le deep learning proposées pour la résolution de ce problème. Ainsi, nous allons présenter une approche basée sur l'architecture auto-encodeur qui prédit la matrice de contact à partir d'une séquence d'acides aminés donnée et les modifications que nous avons apportées à cette approche.

Mots clés : Protéines, acide aminé, séquence, la structures des protéines, l'intelligence artificielle, deep learning, auto-encodeur, matrice de contact.

Abstract

Proteins are essential for life, they take care of just about everything in a living organism. They are large complex molecules, made up of chains of amino acids, once made, they fold up on themselves to adopt a unique shape. The shape of a protein is closely linked to its function, and the ability to predict this structure is of paramount importance, both in determining its role in the body, as well as in developing drugs for diseases thought to be caused by misfolded proteins, such as Alzheimer's disease and Parkinson's disease. Scientific research has been trying for many years to find a method to determine the structure of proteins using a variety of experimental techniques, such as nuclear magnetic resonance and X-ray crystallography which depend on a lot of trial and error, which can take years of work and require the use of specialized equipment by millions of dollars. This is why biologists are turning to artificial intelligence (AI) methods as an alternative to this process. In a major scientific breakthrough, the latest version of the AlphaFold AI system has been recognized as a solution to this great challenge by the organizers of the CASP competition. This breakthrough demonstrates the impact AI can have on scientific discovery and its potential to dramatically accelerate progress in some of the most fundamental areas that explain our world.

In this thesis, we will present different approaches based on deep learning proposed to solve this problem. Thus, we will present an approach based on the auto-encoder architecture that predicts the contact matrix from a given amino acid sequence and the modifications we have made to this approach.

Key words : Proteins, amino acid, sequence, protein structures, artificial intelligence, deep learning, auto-encoder, contact matrix.

TABLE DES MATIÈRES

TABLE DES FIGURES

ABBREVIATIONS

1D 1 dimension
2D 2 dimensions
3D 3 dimensions
4D 4 dimensions
AA Les acides aminés
ADAM Adaptive Moment Estimation
ADN acide désoxyribonucléique
ARN acide ribonucléique
CASP Critical Assessment of protein Structure Prediction
CM contact map
CNN Convolutional neural network
DL deep learning ou apprentissage profond
DM distance map
MSA Multiple sequence alignment
PDB Protein Data Bank
PSA protein structure annotation - l'annotation de la structure des protéines
PSSP prédiction de la structure secondaire des protéines
ReLU rectified linear unit
RL Apprentissage par renforcement - Reinforcement learning
TBM Template Based Modeling
TFM Template Free Modeling

INTRODUCTION GÉNÉRALE

À l'intérieur de chaque cellule de notre corps, des milliards de nanomachines moléculaires travaillent dur. Ce sont eux qui permettent à nos yeux de détecter la lumière, à nos neurones de se déclencher et aux instructions de notre ADN d'être lues, ce qui fait de nous une personne unique. Ces machines complexes sont **“ les protéines “**. Ils sous-tendent non seulement les processus biologiques de notre corps, mais tous les processus biologiques de chaque être vivant. Ce sont les éléments constitutifs de la vie.

Actuellement, il existe environ 100 millions de protéines distinctes connues, et de nombreuses autres sont découvertes chaque année. Chacune a une forme 3D unique qui détermine comment elle fonctionne et ce qu'elle fait. Si une protéine n'adopte pas la bonne structure (généralement à cause d'une mutation qui a changé sa séquence d'acides aminés), elle ne pourra pas assurer sa fonction. C'est le problème des maladies génétiques.

Déterminer la structure exacte d'une protéine est un processus coûteux et souvent long, ce qui signifie que nous ne connaissons que la structure 3D exacte d'une infime fraction des protéines connues de la science. Trouver un moyen de combler cet écart et de prédire la structure de millions de protéines inconnues pourrait non seulement nous aider à lutter contre les maladies et à trouver plus rapidement de nouveaux médicaments, mais peut-être aussi percer les mystères du fonctionnement de la vie elle-même.

Les techniques expérimentales pour déterminer les structures des protéines, telles que la résonance magnétique nucléaire [1] et la cristallographie aux rayons X [2] dépendent de nombreux essais et erreurs, qui peuvent prendre des années de travail, et nécessitent l'utilisation d'équipements spécialisés de plusieurs millions de dollars. C'est pourquoi les biologistes se tournent vers les méthodes d'intelligence artificielle comme alternative à ce processus.

Récemment, le domaine de la prédiction de la structure des protéines a connu de nombreuses avancées grâce aux approches basées sur le deep learning ou apprentissage profond (DL) comme le succès d'AlphaFold2 [3] en CASP14. Le concours Critical Assessment of protein Structure Prediction (CASP) évalue l'état de l'art dans la modélisation de la structure des protéines à partir de la séquence d'acides aminés.

Le premier concours CASP a eu lieu en 1994. Dans une avancée rapide en 2016 dans CASP12,

la prédiction de contact est apparue comme l'étape intermédiaire clé vers une prédiction précise de la structure. Pour la première fois, une méthode basée sur le DL, RaptorX-Contact [4], a atteint une précision d'environ 50%, soit près de deux fois plus de précision que la compétition CASP11. Peu après le CASP12, une version plus améliorée de RaptorX-Contact a été publiée [5].

Après l'annonce des résultats du concours CASP13 en 2018, les méthodes les plus performantes, notamment RaptorX-Contact [6] et AlphaFold1 [7], avaient mis à jour leurs méthodes pour prédire les « distogrammes » au lieu des seuls contacts. Dans CASP14, AlphaFold2 [3] a surpassé les autres méthodes et produit des modèles remarquablement précis qui ont obligé les organisateurs à déclarer que le problème de prédiction de la structure des protéines devrait être résolu.

Dans ce projet nous allons présenter différentes approches basées sur le deep learning proposées pour la résolution de ce problème. Notre mémoire est organisé en quatre chapitres. Dans le premier chapitre nous allons découvrir c'est quoi une protéine ? Quelles sont ses différentes structures ? et pourquoi la détermination de sa structure est un défi sur lequel les scientifiques travaillent depuis des décennies ? Nous soulignons dans le deuxième chapitre des étapes importantes et des progrès dans le domaine de la prédiction de la structure des protéines grâce aux méthodes basées sur le DL. Le troisième chapitre contient une introduction à l'apprentissage automatique et à l'apprentissage profond ainsi que leurs différents algorithmes. Dans le dernier chapitre nous allons présenter une approche basée sur le deep learning qui prédit la matrice de contact d'une séquence de protéine donnée, ainsi que les modifications que nous avons apportées à cette approche. Aussi, nous allons présenté les outils d'implémentation utilisés pour la réalisation de cette approche.

CHAPITRE

1

LES PROTÉINES

1 Introduction

Les corps des êtres vivants sont constitués de milliards de cellules, chacune ayant une fonction essentielle différente. La protéine est un composant important de la cellule qui a elle-même de nombreuses fonctions.

Dans ce chapitre nous allons découvrir c'est quoi une protéine ? Quelles sont ses différentes structures ? et pourquoi la détermination de sa structure est un défi sur lequel les scientifiques travaillent depuis des décennies ?

2 Qu'est ce qu'une protéine ?

Le terme «protéine» a déjà un sens dans le langage de tous les jours, ce sont des nutriments qu'on trouve dans la viande, le poisson, les œufs, etc. mais en biologie, les protéines sont plus spécifiquement des macromolécules de la cellule, dont elles constituent la « boîte à outils », lui permettant de digérer sa nourriture, produire son énergie, de fabriquer ses constituants, de se déplacer, etc [8].

Les protéines s'occupent de toutes les fonctions d'un organisme vivant telles que la catalyse des réactions biochimiques, le transport des nutriments, la reconnaissance et la transmission des signaux, elles servent de récepteurs, de transporteurs, d'anticorps, d'enzymes, etc.

Pour pouvoir assurer toutes ces fonctions, il existe un très grand nombre de protéines différentes, évidemment ces protéines sont largement spécifiques à une espèce, et donc quand on mange des protéines, elles sont découpées en briques élémentaires qui sont ensuite recyclées pour fabriquer nos propres protéines, ce qui permet cela, c'est le fait que toutes les protéines du monde vivant sont fabriquées à partir d'une liste très restreinte de briques : « les acides aminés » [9].

3 Les acides aminés (AA)

Le vivant utilise 21 types différents d'acides aminés, qui permettent de fabriquer toutes les protéines. Les acides aminés sont des molécules élémentaires qui obéissent à un schéma bien précis : Un atome de carbone, qu'on représente au centre, qui possède 4 liaisons et auquel sont attaché d'un côté un simple atome d'hydrogène, sur un autre, un groupe NH2, dit « amine », et de l'autre côté un groupe COOH dit « acide », d'où le nom **« acide aminé »**. Un radical est attaché sur la dernière liaison. Les 21 acides aminés du vivant ont chacun un radical différent (Figure 1.1) [9].

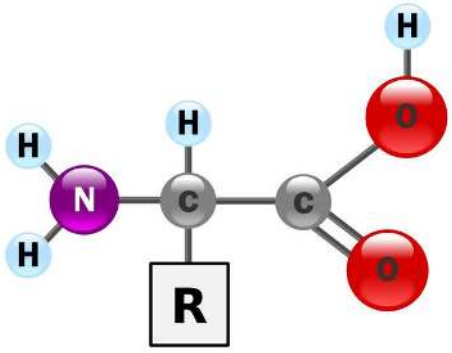


FIGURE 1.1 – Acide aminé

La figure 1.2 représente la liste d'AA avec leur nom, leur formule, et une lettre unique pour les désigner.

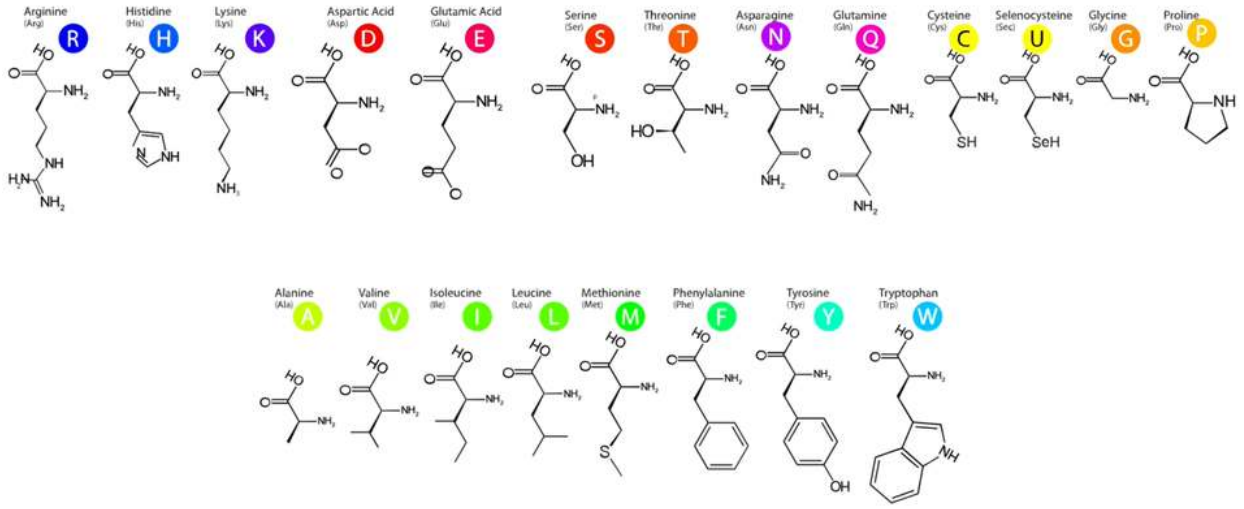


FIGURE 1.2 – Liste des acides aminés

Pour combiner les acides aminés et créer des protéines, ils sont simplement liés les uns aux autres pour former une chaîne, et donc pour la décrire, il suffit de donner la séquence de ces acides aminés.

La figure 1.3 montre un exemple d'une séquence qui décrit une protéine qu'on trouve dans le Sars-Cov2 (responsable du COVID-19) :

```
APTKVTFGDDTVIEVQGYKSVNITFELDERIDKVLNEKCSAYTVEL
GTEVNEFACVVADAVIKTLQPVSELLTPLGIDLDEWSMATYYLFDE
SGEFKLASHMYCSFYPPDE
```

FIGURE 1.3 – Séquence des acides aminés d'une protéine

De façon générale, quand une cellule de notre organisme doit fabriquer une protéine, il lui faut connaître le plan de montage, c'est-à-dire quels acides aminés enchainer, et dans quel ordre.

Les recettes de ces protéines - appelées gènes - sont codées dans notre acide désoxyribonucléique (ADN).

4 L'ADN

L'ADN est la molécule responsable de la transmission de l'information génétique héréditaire de génération en génération, présente dans toutes les cellules vivantes de l'organisme [10]. Une erreur dans cette recette génétique peut entraîner une protéine mal formée, ce qui peut entraîner une maladie ou la mort d'un organisme [9].

l'ADN est composé de quatre bases : l'adénine (A), la guanine (G), la cytosine (C) et la thymine (T). Il est formé de deux brins complémentaires enroulés en hélice nommée la structure en "double hélices" (Figure 1.4).

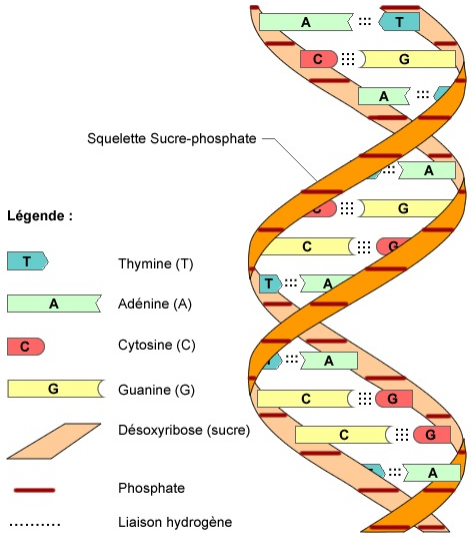


FIGURE 1.4 – L'ADN

Au début des années 1950, le biologiste Francis Crick a suggéré qu'il existe un flux unidirectionnel d'informations génétiques allant de l'ADN aux protéines en passant par l'acide ribonucléique (ARN) : "ADN → ARN → Protéines" [11].

5 L'ARN

L'ARN a une composition similaire à l'ADN (Figure 1.5). C'est une longue molécule linéaire constituée d'un nombre limité de nucléotides. Ces éléments constitutifs sont les quatre nucléotides : l'adénine (A), la guanine (G), la cytosine (C) et l'uracile (U), (l'uracile au lieu de thymine dans l'ADN).

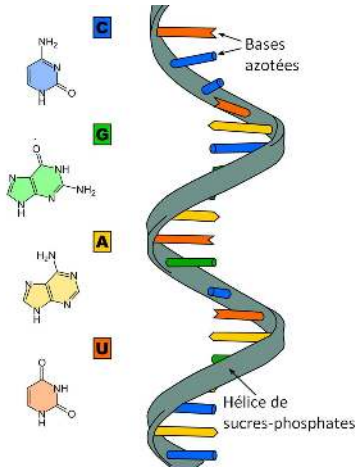


FIGURE 1.5 – L'ARN

L'ADN est transcrit en une molécule d'ARN (ARN messager (ARNm)), qui contient les mêmes informations de séquence que l'ADN, et ensuite cet ARNm est traduit en une séquence protéique selon le code génétique.

"***La transcription est le mécanisme qui permet de dupliquer un brin d'ADN sous forme d'ARN***".

La figure 1.6 montre le processus de transcription.

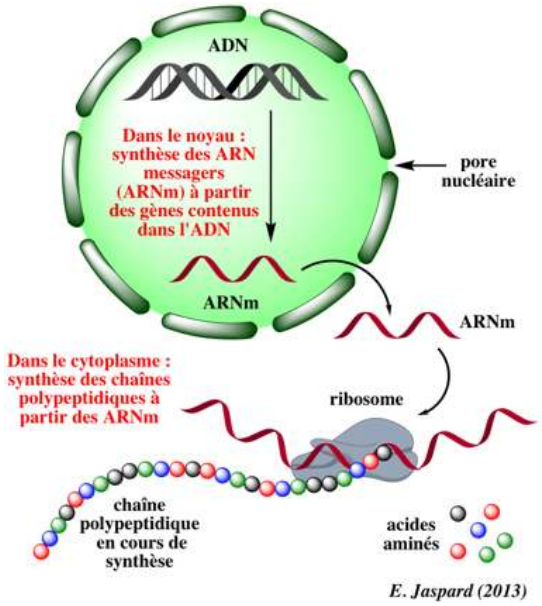


FIGURE 1.6 – Le processus de transcription

Le code génétique est la correspondance entre la séquence des quatre bases des acides nucléiques et la séquence des 21 acides aminés des protéines.

La figure 1.7 montre clairement cette correspondance.

Ce qui permet aux protéines d'agir, de se comporter comme des nanomachines moléculaires, c'est qu'une fois fabriquées, les chaînes se replient sur elles-mêmes pour adopter une forme bien précise. Chaque protéine a sa forme attitrée qui lui donne sa fonction.

1		2 U		2 C		2 A		2 G		3
	U	UUU	Phe	UCU	Ser	UAU	Tyr	UGU	Cys	U
		UUC		UCC		UAC		UGC		C
		UUA	Leu	UCA		UAA	STOP	UGA	STOP	A
		UUG		UCG		UAG		UGG	Trp	G
	C	CUU		CCU	Pro	CAU	His	CGU	Arg	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	Gln	CGA		A
		CUG		CCG		CAG		CGG		G
	A	AUU	Ile	ACU	Thr	AAU	Asn	AGU	Ser	U
		AUC		ACC		AAC		AGC		C
		AUA		ACA		AAA	Lys	AGA	Arg	A
		AUG	Met	ACG		AAG		AGG		G
	G	GUU	Val	GCU	Ala	GAU	Asp	GGU	Gly	U
		GUC		GCC		GAC		GGC		C
		GUA		GCA		GAA	Glu	GGA		A
		GUG		GCG		GAG		GGG		G

MS © cours-pharmacie.com

FIGURE 1.7 – Le code génétique

6 La forme des protéines

Comme nous l'avons dit, les protéines sont des longues chaînes d'acides aminés. Certains de ces acides aminés sont hydrophiles, d'autres sont hydrophobes.

À une température suffisante, les éléments hydrophobes ont généralement tendance à se replier vers le centre de la protéine pour fuir l'eau environnante, alors que les éléments hydrophiles restent à l'extérieur.

Ce n'est pas la seule force qui entre en jeu. Entre autres, les charges électriques jouent aussi un rôle. Finalement, la protéine est stabilisée par des liaisons entre acides aminés [12].

De la séquence au repliement, il existe quatre niveaux de structuration de la protéine (Figure 1.8) :

La séquence des acides aminés est appelée la structure primaire de la protéine (1 dimension (1D)).

Cette chaîne d'acides aminés se replie en structures secondaires (2 dimensions (2D)) locales comprenant des hélices alpha, des feuillets bêta et des coudes.

Ensuite, les éléments de la structure secondaire sont en outre repliés pour former une structure tertiaire (3 dimensions (3D)) en fonction des forces hydrophobes et des interactions entre les acides aminés, telles que la liaison hydrogène. La structure tertiaire décrit la structure tridimensionnelle de la protéine.

Enfin, plusieurs chaînes protéiques peuvent interagir ensemble ou s'assembler pour former la structure quaternaire des protéines (4 dimensions (4D)) [13].

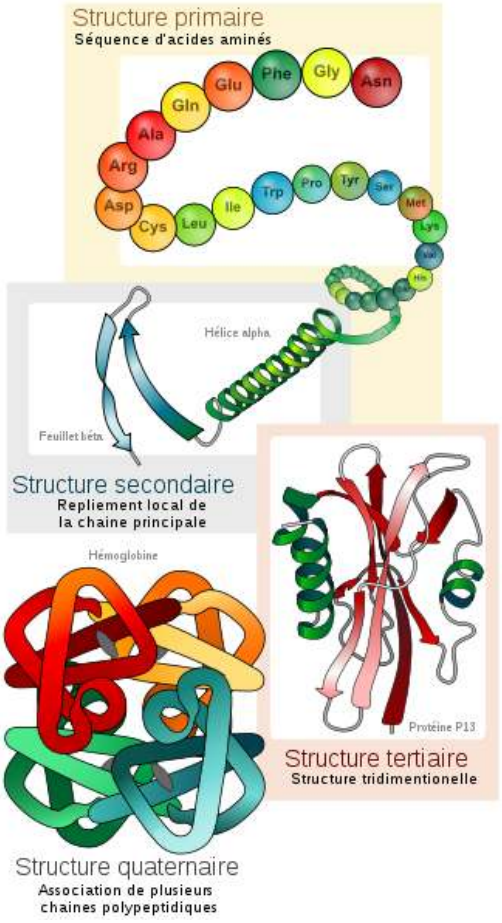


FIGURE 1.8 – Les quatre niveaux de structure des protéines

Ce qu'une protéine donnée peut faire dépend de sa structure 3D unique. De nombreuses expériences biochimiques ont montré que la fonction d'une protéine est déterminée par sa structure.

Par exemple, les protéines anticorps utilisées par notre système immunitaire sont en forme de «Y» et forment des crochets uniques (Figure 1.9). En s'accrochant aux virus et aux bactéries, ces protéines anticorps sont capables de détecter et de marquer les micro-organismes pathogènes pour leur élimination [9].

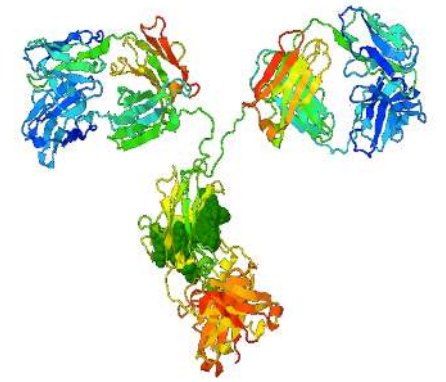


FIGURE 1.9 – Anti corps

Il y a aussi beaucoup de protéines qui servent de récepteurs et qui fonctionnent comme une serrure qui attend sa clé. La protéine s'active quand une molécule de forme complémentaire vient se lier à elle. C'est d'ailleurs ce mécanisme qu'il y a derrière le principe de beaucoup de médicaments, et aussi de drogues.

L'importance de la forme des protéines est telle qu'un certain nombre de maladies sont dues au fait que certaines protéines n'adoptent pas la bonne configuration. On pense que c'est notamment le cas pour des formes de phénylcétonurie, ou des maladies neurodégénératives comme les maladies d'Alzheimer ou de Parkinson [9].

La figure 1.10 montre la relation séquence-structure-fonction.

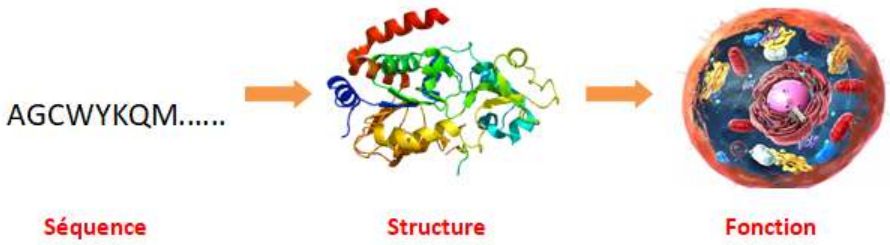


FIGURE 1.10 – la relation séquence-structure-fonction d'une protéine

Il est difficile de déterminer la structure des protéines expérimentalement. Plus la protéine est grosse, plus elle est difficile à modéliser, car il y a plus d'interactions entre les acides aminés à prendre en compte. Mais c'est plutôt facile de déterminer la séquence des acides aminés, il suffit d'aller lire les bons endroits d'ADN ou d'ARN par des techniques de séquençage de génome.

En fait, connaître la recette génétique d'une protéine ne veut pas dire connaître automatiquement sa forme, car l'ADN ne contient que des informations sur la séquence des acides aminés pas sur la façon dont ils se mettent en forme.

Par conséquent, le taux de découverte des structures est beaucoup plus lent que le taux d'identification des séquences en raison du coût et de la complexité. Les banques des séquences ne cessent de croître à l'instar des banques de structures protéiques [14].

7 Les banques de données

La base de données UniProt (Universal Protein Resource, https://www.uniprot.org) contient plus de 200 millions de séquences et le compte augmente environ 30 millions chaque année [15] [16]. Tandis que la PDB (Protein Data Bank, https://www.rcsb.org) recense, au 20 mai 2022 190 639 structures. 6 000 à 7 000 sont ajoutées chaque année.

La PDB est la principale banque internationale de structures tridimensionnelles, ces structures sont essentiellement déterminées par cristallographie aux rayons X ou par spectroscopie RMN [17].

En 2021, l'équipe des chercheurs britannique DeepMind en partenariat avec l'Institut européen de bioinformatique de l'EMBL (EMBL-EBI) ont crée la base de données sur la structure des protéines AlphaFold (AlphaFold DB, https://alphafold.ebi.ac.uk).

AlphaFold DB offre un accès libre à 992 316 prédictions de structure protéique. Ils prévoient en 2022 d'étendre la base de données pour couvrir une grande partie de toutes les protéines cataloguées (plus de 100 millions dans UniRef90 [18]) [19].

Toutes ces données sont accessibles publiquement à la communauté scientifique, cela permettra d'accélérer la recherche scientifique. Comprendre la structure et la fonction de la protéine en cause permet ensuite d'élaborer des thérapies adaptées, en particulier des thérapies géniques.

Prédire comment ces chaînes se replient dans la structure 3D complexe d'une protéine est ce que l'on appelle «Le problème du repliement des protéines». C'est un défi sur lequel des générations de chercheurs travaillent depuis longtemps [14] !

8 Le problème du repliement des protéines

Dans son discours de remerciement pour le prix Nobel de chimie de 1972, le biochimiste Christian Anfinsen a émis l'hypothèse thermodynamique qu'en théorie, toutes les informations qui régissent le repliement des protéines sont contenues dans leurs séquences primaires [20].

Les scientifiques ont essayé de trouver une méthode pour prédire de manière fiable la structure d'une protéine uniquement à partir de sa séquence d'acides aminés. Pour avoir la forme d'une protéine, on ne peut pas juste la mettre sous un microscope et la regarder mais, il faut utiliser des techniques de mesure comme la cristallographie aux rayons X [21].

8.1 Méthodes pour déterminer la forme d'une protéine

Au cours des cinq dernières décennies, les chercheurs ont été en mesure de déterminer les formes de protéines en laboratoire à l'aide de techniques expérimentales telles que la microscopie cryoélectronique [22], la résonance magnétique nucléaire (RMN) [1] et la cristallographie aux rayons X [2] (Figure 1.11), mais chaque méthode dépend de nombreux essais et erreurs, ce qui peut prendre

des années de travail et coûter des dizaines ou des centaines de milliers de dollars par structure protéique [14].

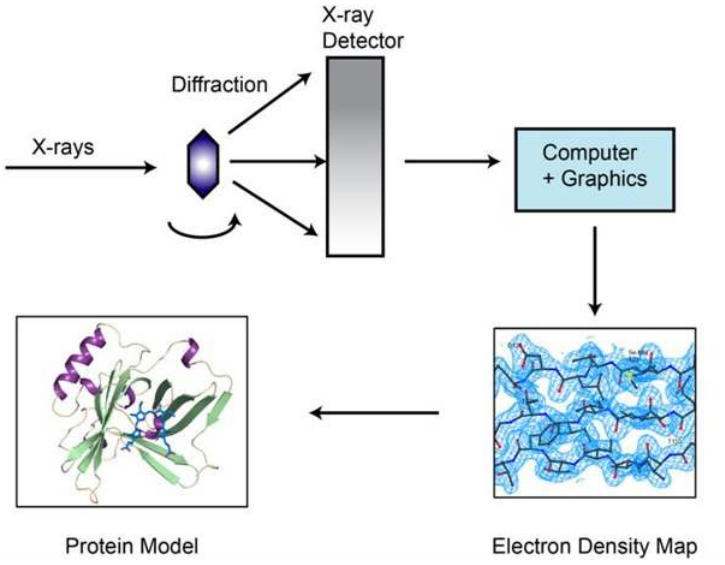


FIGURE 1.11 – La cristallographie aux rayons X

C'est pourquoi les biologistes se tournent vers les méthodes d'intelligence artificielle comme alternative à ce processus. L'apprentissage automatique (ML) et l'apprentissage profond (DL) ont prouvé leur efficacité dans le traitement du langage naturel, le traitement d'images, la vision par ordinateur, la reconnaissance vocale et d'autres domaines informatiques [23]. Ces succès ont attiré l'attention des chercheurs en bioinformatique - c'est ce que nous verrons au chapitre 2.

8.2 L'importance de la prédiction de la structure des protéines

La prédiction de la forme d'une protéine est d'une importance primordiale à la fois pour comprendre son rôle dans l'organisme, ainsi que pour développer des médicaments pour des maladies supposées être causées par des protéines mal repliées, telles que la maladie d'Alzheimer, la maladie de Parkinson, la maladie de Huntington et la fibrose kystique [24].

Des travaux ont déjà amélioré notre compréhension de nombreux processus fondamentaux de la santé et de la maladie comme la détermination de la structure de l'hémoglobine, la protéine des globules rouges chargée de transporter l'oxygène dans le corps, a aidé les chercheurs à comprendre comment une seule mutation peut provoquer la drépanocytose, aidant ainsi à développer des traitements pour cette maladie.

Ainsi, la détermination de la structure des protéines virales du SRAS-COV-2 a permis aux scientifiques de comprendre son fonctionnement, d'identifier des traitements et de développer de nouveaux vaccins [16].

Une compréhension du repliement des protéines aidera également à la conception des protéines, comme les enzymes biodégradables qui pourraient aider à gérer les polluants comme le plastique et l'huile, nous aidant à décomposer les déchets de manière plus respectueuse de notre environnement. En fait, les chercheurs ont déjà commencé à concevoir des bactéries pour sécréter des protéines qui rendront les déchets biodégradables et plus faciles à traiter [24].

Pour stimuler la recherche et mesurer les progrès sur les nouvelles méthodes d'amélioration de la précision des prédictions, un concours biennal mondial appelé CASP (Critical Assessment of protein Structure Prediction) a été créé en 1994 et est devenu la référence de techniques d'évaluation.

8.3 CASP

Pour pouvoir comparer de façon objective l'efficacité des différentes méthodes proposées partout dans le monde, le biologiste John Moult a proposé d'organiser une sorte de compétition opposant les différents algorithmes. Cette compétition a lieu tous les deux ans depuis 1994, et se déroule selon un protocole bien précis : Un comité d'organisation choisit des protéines dont on ne connaît que la séquence d'acides aminés.

D'un côté des expérimentateurs travaillent à déterminer la véritable structure de la protéine, le plus souvent avec des rayons X, et de l'autre chaque équipe qui prend part au concours s'efforce avec ses algorithmes de deviner à l'avance cette forme. Et à la fin on regarde qui a été le plus proche possible de la véritable forme.

Pour pouvoir classer les différentes méthodes, il faut un critère numérique, un score qui permette de les comparer. Ce qui est utilisé, c'est une quantité appelée **GDT** : On prend une prédiction pour la forme d'une protéine, on la compare avec la vraie forme déterminée par les expérimentateurs, et on compte quel pourcentage des acides aminés sont bien positionnés [9]. Voici un exemple dans la figure 1.12 - en bleu la prédiction et en vert la vraie structure.

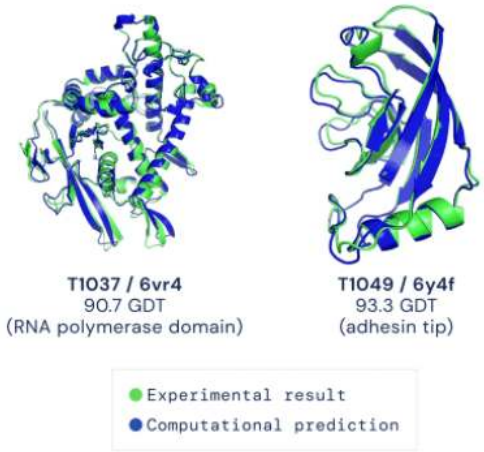


FIGURE 1.12 – CASP

Le meilleur score atteint dans cette compétition était en 2020 (CASP 14) par l'algorithme AlphaFold2 [3] de l'équipe DeepMind. Il s'agit d'une start up londonienne fondée par le neuroscientifique Demis Hassabis et rachetée ensuite par Google. Leurs méthodes se basent principalement sur le deep learning, le score GDT médian d'AlphaFold2 est de 92.4 dans l'ensemble de toutes catégories confondues. Cela signifie que leurs prédictions ont une erreur moyenne (RMSD) d'environ 1,6 Angströms, ce qui est comparable à 0,1 nanomètre. Même pour la catégorie de modélisation libre (Free Modeling) la plus difficile, AlphaFold2 atteint un score médian de 87,0 GDT [9] (Figure 1.13).

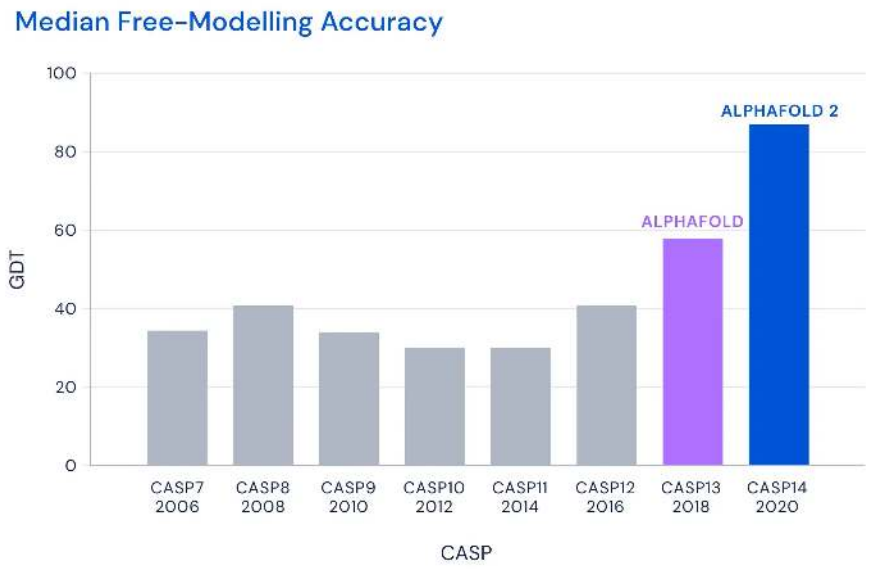


FIGURE 1.13 – Scores des gagnants de chaque édition CASP depuis 2006

C'est une médiane, sur certaines molécules ça peut être significativement moins bien que ça, mais sur d'autres c'est encore mieux. Ensuite même si on arrivait à prédire parfaitement la forme, tout ne serait pas gagné pour autant.

8.4 Défis

Il est très fréquent que dans les cellules les protéines s'associent en complexes, et que la forme d'une protéine ne soit pas forcément celle qu'elle aurait si elle était toute seule.

Les protéines s'influencent les unes les autres, ça dépend de la température, du Ph et on ne peut pas forcément les étudier en isolation [9].

Et puis sur le plan plus fondamental, même si on devient effectivement capable de prédire la forme des protéines grâce au Deep Learning, ça n'est pas pour autant que l'on comprendra comment elles acquièrent ces formes, par quel mécanisme et pour quelles raisons.

9 Conclusion

Dans ce chapitre nous avons commencé par définir les protéines et leurs composantes (les acides aminés). Ensuite nous avons détaillé les différentes structures d'une protéine ainsi que les molécules responsables de sa conception (ADN et ARN). Puis nous avons présenté les principales bases de données de protéines PDB, UniProt et la nouvelle AlphaFold DB. Enfin, nous avons défini le problème de repliement des protéines, son importance et ses défis. Les techniques de résolution de ce problème seront détaillées dans le prochain chapitre.

CHAPITRE

2

LA PRÉDICTION DE LA STRUCTURE DES PROTÉINES

1 Introduction

Dans le chapitre précédent, nous avons vu que l'obtention d'une description précise de la structure des protéines est une étape fondamentale vers la compréhension des bases de la biologie.

Récemment, le domaine de la prédiction de la structure des protéines a connu de nombreuses avancées grâce aux approches basées sur le Deep Learning (DL). Dans ce chapitre nous soulignons des étapes importantes et des progrès dans le domaine de la prédiction de la structure des protéines grâce aux méthodes basées sur le DL, comme observé dans les expériences CASP.

2 La prédiction de la structure tertiaire d'une protéine

Il existe deux approches principales, la modélisation basée sur des modèles (Template Based Modeling (TBM)), dans laquelle la structure est déterminée en se basant sur un modèle connu dans les base de données ; et la modélisation sans modèle (Template Free Modeling (TFM)), qui ne repose pas sur un modèle connu [25].

2.1 Template Based Modeling (TBM)

Les TBM comprennent à la fois les techniques de modélisation par enfilage et la modélisation par homologie (ou modélisation comparative).

2.1.1 La modélisation par homologie

Chaque séquence de protéine native adopte une structure unique. En d'autres termes, les protéines avec des séquences similaires ont tendance à se replier dans des structures similaires. Par exemple, si une séquence de structure inconnue (désigner U) a une similarité de séquence significative avec une protéine de structure connue (T), il est possible de construire un modèle 3D approximatif pour U en se basant sur l'hypothèse que U a simplement la même structure que T. Cette technique augmente efficacement le nombre de structures 3D connues [26].

Toutes les méthodes de modélisation par homologie actuelles consistent en cinq étapes séquentielles [27] (Figure 2.1) :

1. Rechercher des protéines avec des structures 3D connues qui sont liées à la séquence requête,
2. Sélectionner les structures qui seront utilisées comme modèles,
3. Aligner leurs séquences avec la séquence requête,
4. Construire le modèle pour la séquence requête en tenant compte de son alignement avec les structures des modèles,
5. Évaluer le modèle, en utilisant une variété de critères.

Ce processus peut être effectué par itération afin d'améliorer la qualité du modèle final.

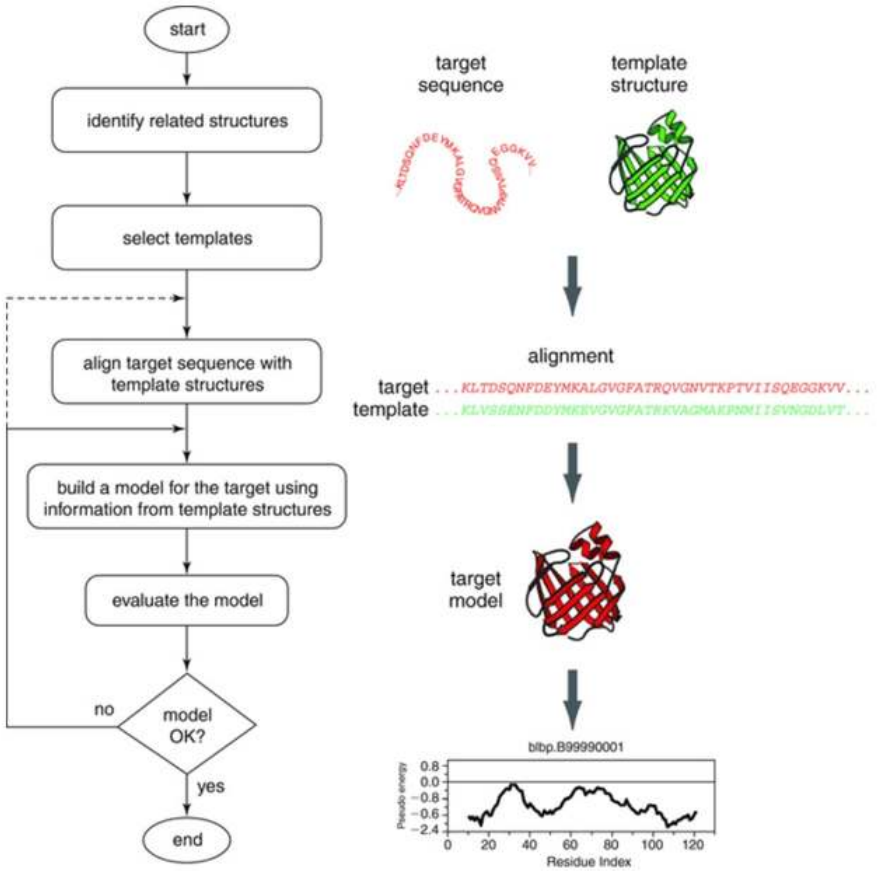


FIGURE 2.1 – Processus de modélisation par homologie

La modélisation par homologie est la méthode la plus précise lorsque la qualité des modèles est élevée, mais s'il n'existe aucune protéine homologue avec une structure 3D connue, la modélisation par enfilage est préférée [23].

2.1.2 La modélisation par enfilage

La modélisation par enfilage ou modélisation par reconnaissance des repliements (threading or fold recognition) se base sur la reconnaissance des coudes (Fold), c'est une méthode qui identifie les structures 3D qui se ressemblent avec un faible score d'alignement. Il s'agit d'un alignement d'une séquence à un morceau de structure [28] (Figure 2.2).

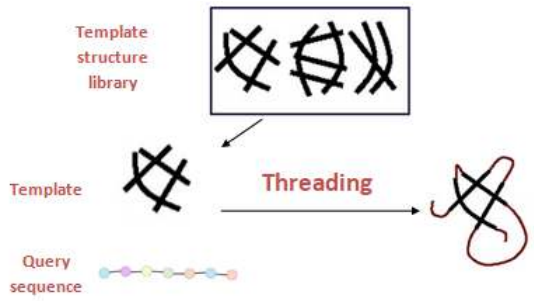


FIGURE 2.2 – Processus de modélisation par enfilage

Quand aucune structure de modèle connue significative n'est identifiée, la modélisation sans modèle (TFM) est la seule approche fiable pour construire des structures à partir de séquences de protéines [29].

2.2 Template Free Modeling (TFM)

Les méthodes TFM traditionnelles, telles que Rosetta [30], tentent de construire une structure tertiaire en assemblant les mini-structures de petits fragments de séquence selon les directives des fonctions énergétiques statistiques. D'autres outils tels que CONFOLD [31] utilisent des prédictions de contact inter-résidus comme contraintes de distance pour guider le repliement des protéines. D'autres méthodes de prédiction de structure tertiaire telles que Zhang Group [32], MULTICOM [33] et RaptorX [4] ont également été basées sur DL et les prédictions de contact et distance entre les résidus [29].

Le processus de TFM commence généralement par la construction d'un alignement de séquences multiples (MSA) de la protéine requête et des séquences associées (Figure 2.3). La séquence requête et de ses homologues sont ensuite utilisées pour prédire les caractéristiques structurelles locales (1D), telles que la structure secondaire et les angles de torsion du squelette, ou les caractéristiques non locales (2D), telles que les contacts résidu-résidu ou les distances inter-résidus à travers la chaîne protéique. Ces caractéristiques sont connues sous le nom de protein structure annotation - l'annotation de la structure des protéines (PSA) [25].

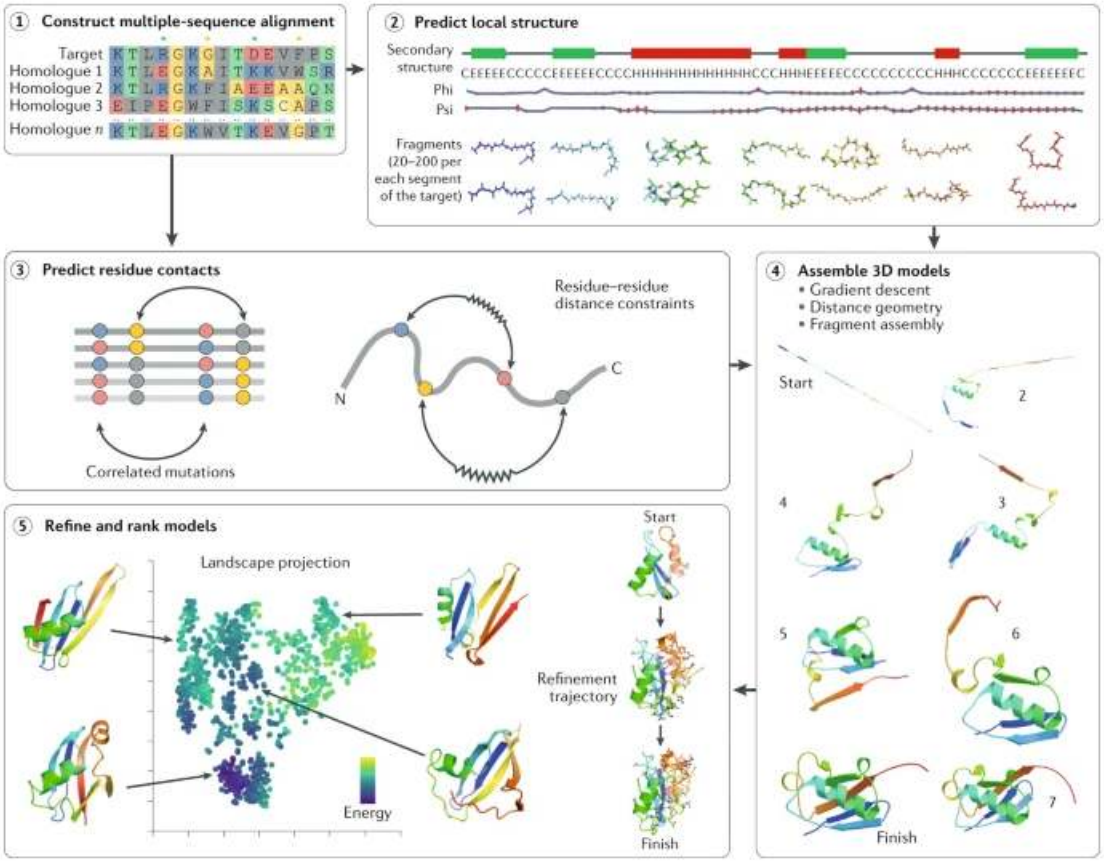


FIGURE 2.3 – Processus de prédiction de structure sans modèle (TFM)

3 L'annotation de la structure des protéines (PSA)

Les prédictions 1D et 2D PSA contiennent des informations simplifiées pour faciliter le processus de calcul et sont utilisées comme étape intermédiaire pour estimer la structure complète de la protéine [34].

3.1 1D

La prédiction 1D se concentre sur la prédiction des caractéristiques structurelles telles que la structure secondaire et l'accessibilité relative au solvant et la prédiction de la région désordonnée de chaque résidu le long de la séquence protéique.

Le problème de prédiction de la structure 1D est souvent considéré comme un problème de classification pour chaque acide aminé individuel dans la séquence protéique. L'entrée est une séquence primaire de protéine et la sortie est une séquence de caractéristiques prédites pour chaque acide aminé de la séquence.

Historiquement, la prédiction de la structure secondaire des protéines (PSSP) a été le problème 1D le plus étudié et a eu un impact fondamental sur le développement des méthodes de prédiction de la structure des protéines.

3.1.1 La prédiction de la structure secondaire d'une protéine (PSSP)

La PSSP est spécifiée par une séquence classant chaque acide aminé dans l'élément de structure secondaire correspondant [35] [36] (Figure 2.4).

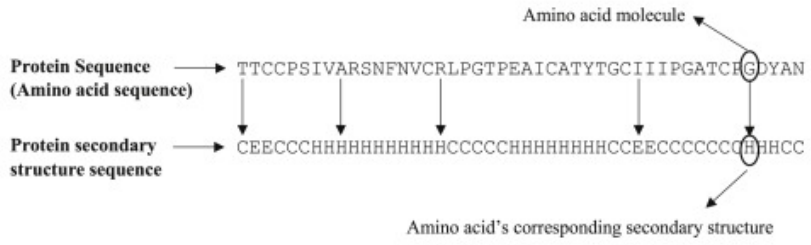


FIGURE 2.4 – La préduction de la structure secondaire

Deux classifications principales sont disponibles : la catégorisation à trois classes (hélice, brin et coude), et la catégorisation à huit classes, qui séparent davantage les trois classes précédentes [34] [37] (Figure 2.5).

8-class	3-class	Name
H	H	α-helix
E	E	β-strand
L	C	loop or irregular
T	C	β-turn
S	C	bend
G	H	3_{10}-helix
B	E	β-bridge
I	C	π-helix

FIGURE 2.5 – Les classifications de la structure secondaire de la proteine

Généralement, des algorithmes d'apprentissage automatique sont implémentés pour la PSSP, tels que les réseaux de neurones (NN), le K plus proches voisins (KNN) et le SVM [38].

Les annotations 1D ont été un sujet central depuis les années 60 tandis que l'attention se déplace progressivement vers des annotations 2D plus informatives et complexes.

3.2 2D

La prédiction 2D se concentre sur la prédiction de la relation spatiale entre les résidus. Les efforts récents pour les 2D PSA se concentrent sur la prédiction des matrices de contact (contact map (CM)) et des matrices de distance (distance map (DM)).

3.2.1 Les matrices de contact CM

Une paire d'acides aminés est dite en contact si la distance entre leurs atomes de carbone est inférieure ou égale à 8 Å (1 Ångström = 0,1 nanomètre), donc une matrice de contact est une matrice binaire de taille L x L (avec L est la longueur de la séquence protéique correspondante) avec des 1s à l'endroit où la distance est inférieure ou égale à 8 Å et des 0s à toutes les autres cellules [39].

De nombreux groupes ont développé des prédicteurs de CM en se basant sur différents types de réseaux de neurones profonds. RaptorX-Contact [4] a été la première a dépassé le seuil de 50% en terme de précision de la prédiction en CASP12. Ensuite, la nouvelle version de RapstorX-Contact [6] et l'approche TripletRes [40] se sont classés respectivement premier et deuxième dans la catégorie de la prédiction de contact en CASP13 [41] (Figure 2.6).

#	GR code	GR name	Domains Count	SUM Zscore (>-2.0)	Rank SUM Zscore (>-2.0)	AVG Zscore (>-2.0)	Rank AVG Zscore (>-2.0)
1	032	TripletRes	31	33.0683	1	1.0667	1
2	498	RaptorX-Contact	31	32.9798	2	1.0639	2
3	323	TripletRes_AT	30	24.0527	4	0.8684	3
4	180	ResTriplet	31	25.5417	3	0.8239	4
5	491	DMP	31	21.1262	5	0.6815	5

FIGURE 2.6 – Classement en CASP13 de la catégorie de la prédiction des matrices de contact

— **TripletRes**

TripletRes [40] commence par la collecte de Multiple sequence alignment (MSA) à travers des bases de données de séquences du génome entier et du métagénome, ensuite il construit trois matrices de caractéristiques co-évolutives (covariance matrix, precision matrix, pseudolikelihood maximization) qui sont entré à trois Convolutional neural network (CNN), puis ils sont fusionnés dans un seul CNN. Chaque CNN est composé de 24 couches convolutionnelles résiduelles avec un noyau de taille 3 3 64. L'entraînement de TripletRes a nécessité 4 GPU fonctionnant simultanément, en utilisant Adam et un taux d'abandon de 80% (Figure 2.7).

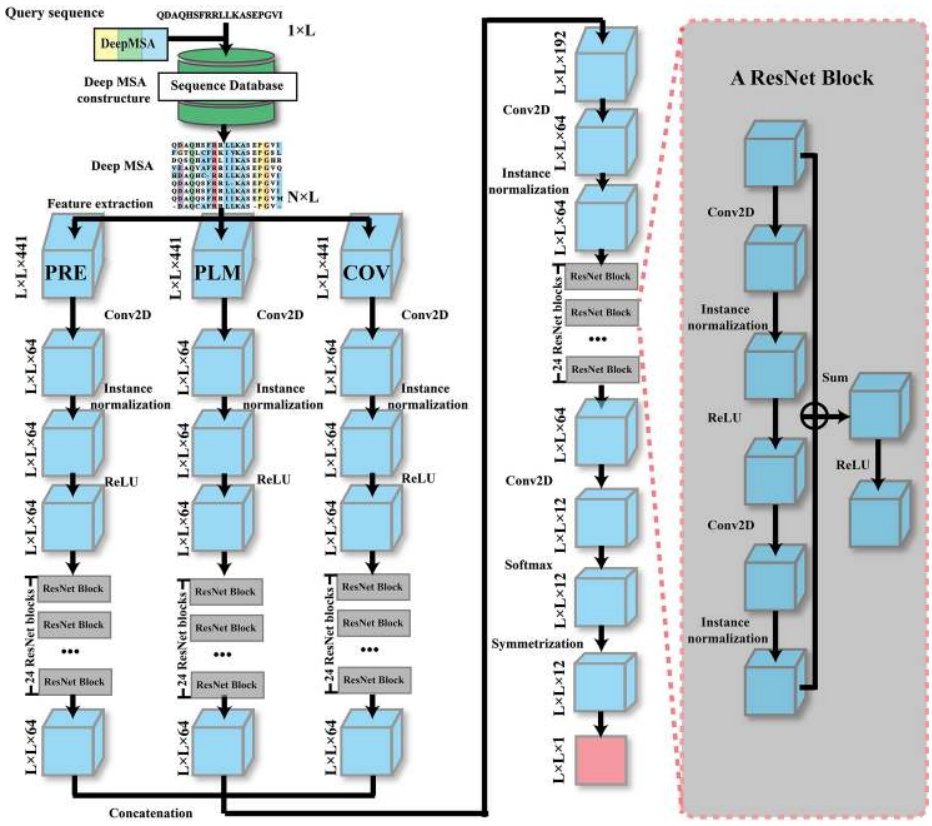


FIGURE 2.7 – Processus de l'approche TripletRes

— **RaptorX-Contact**

RaptorX-Contact [4] prédit les contacts en intégrant à la fois les informations de couplage évolutif (EC) et de conservation des séquences via un réseau de neurones ultra-profond formé par deux réseaux de neurones résiduels profonds. Le premier réseau résiduel effectue une série de transformations convolutives unidimensionnelles de caractéristiques séquentielles. Le deuxième réseau résiduel effectue une série de transformations convolutives bidimensionnelles d'informations par paires comprenant la sortie du premier réseau résiduel, des informations EC et un potentiel par paires (Figure 2.8).

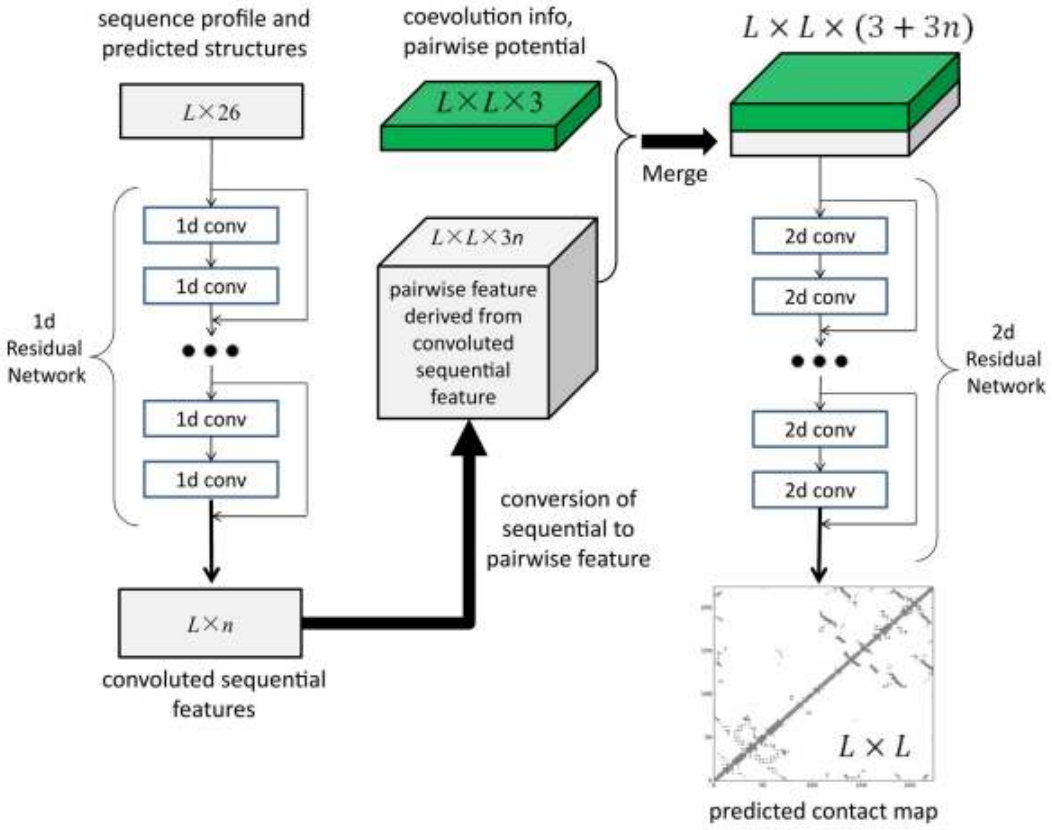


FIGURE 2.8 – Processus de l'approche RaptorX-Contact

La prédiction du contact inter-résidus était largement utilisée depuis plus d'une décennie dans le domaine de la prédiction de la structure des protéines. Cependant, récemment, le paradigme s'est déplacé vers la prédiction de la probabilité des intervalles de distance, appelés également matrices de distance (DM) ou distogrammes.

3.2.2 Les matrices de distance (DM)

La prédiction de la distance inter-résidus d'une protéine est la prédiction d'une matrice 2D de distance par paires d'acides aminés à partir de sa séquence.

Du point de vue de l'apprentissage automatique, le problème de prédiction de distance dans la prédiction de la structure des protéines peut être comparé au problème d'estimation de profondeur monoculaire en vision par ordinateur. Dans la prédiction de profondeur d'image, une matrice d'image est fournie en entrée et une matrice de profondeur est prédite en sortie où chaque pixel

a une profondeur prédite (distance de la caméra à l'objet). De même, la prédiction de distance prend un volume d'entrée tridimensionnel (hauteur × largeur × canaux) et produit une matrice de distance avec la même dimension que l'entrée (hauteur × largeur) mais avec un seul canal, et chaque pixel sur la carte représente une distance entre une paire de résidus dans la séquence (Figure 2.9).

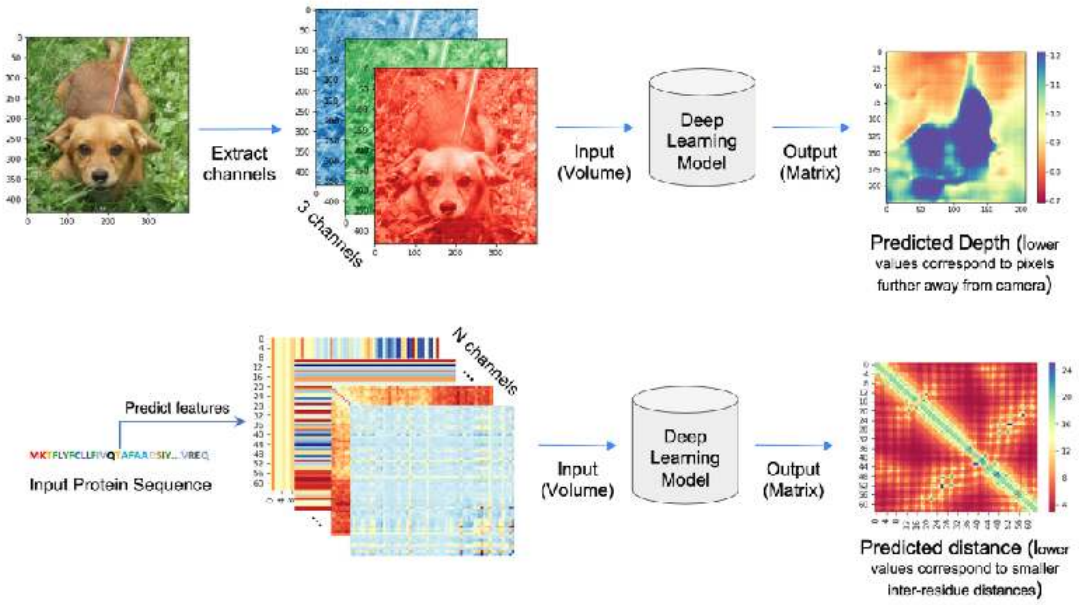


FIGURE 2.9 – Comparaison entre le problème d'estimation de profondeur monoculaire et le problème de prédiction de distance

AlphaFold1 [7] est l'approche qui a défendu l'idée d'utiliser les distogrammes pour la prédiction de la structure des protéines, elle a permis d'établir le fait que le distogramme prédit est le meilleur que la matrice de contact pour la prédiction de la structure des protéines.

— **AlphaFold1**

AlphaFold1 [7] a eu les meilleures performances en CASP13. Essentiellement, il utilise un réseau neuronal convolutif (CNN) qui est entrainer sur des structures de PDB pour prédire les distances entre n'importe quelle paire de résidus. En utilisant la représentation des aminés de la séquence de requête et les caractéristiques générées à partir de MSA, le réseau CNN prédit une distribution de probabilité discrète pour chaque paire. Cette distribution s'avère similaire aux vraies distances. Ensuite, les angles de torsion prédits du squelette et la distance par paire entre les résidus sont combinés pour former un score spécifique à la protéine. Enfin, la descente de gradient est appliquée sur le score spécifique de la protéine pour obtenir le modèle protéique final (Figure 2.10).

Le code est disponible en open source sur GitHub [1].

1. https://github.com/deepmind/deepmind-research/tree/master/alphafold_casp13

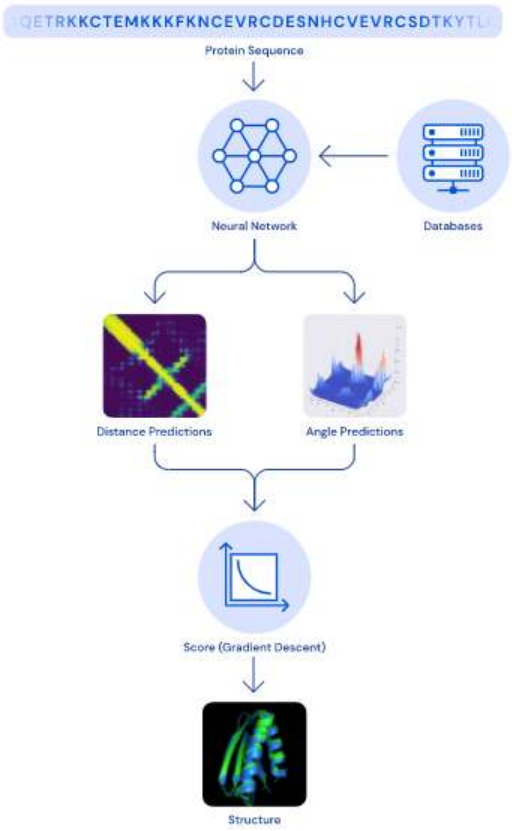


FIGURE 2.10 – Le processus d'AlphaFold1

Bien que le but ultime soit de prédire la structure 3D, les prédictions 1D et 2D sont souvent utilisées comme entrée pour les prédicteurs de coordonnées 3D (Figure 2.11).

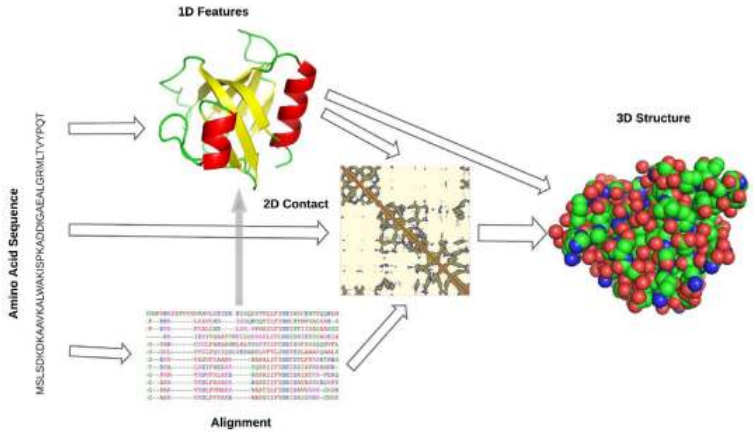


FIGURE 2.11 – L'utilisation des 1D-2D PSA dans la prédiction de la structure des protéines

En CASP14 (2020) DeepMind est venu avec un nouveau modèle complètement différent de AlphaFold1 appelé AlphaFold2 [3].

4 AlphaFold2

AlphaFold2 [3] utilise un système de réseau neuronal appelé transformer basé sur les mécanismes d'attention [2] qui prédit les coordonnées 3D de tous les atomes pour une protéine donnée en combinant les informations de la séquence d'acides aminés, les alignements de séquences multiples et les structures homologues [43]. La partie centrale du réseau neural, appelée Evoformer, consiste en une représentation de l'alignement de séquence multiple (MSA representation) et des relations par paires entre les différents acides aminés de la protéine (pair representation). La représentation des paires contient des informations sur les positions relatives des acides aminés dans la chaîne. Ces deux représentations sont mélangées et traitées par un ensemble de modules de réseaux de neurones. La première ligne de MSA est ensuite utilisée avec la représentation des paires pour prédire la structure finale [44]. Le processus d'AlphaFold2 est montré dans la figure 2.12.

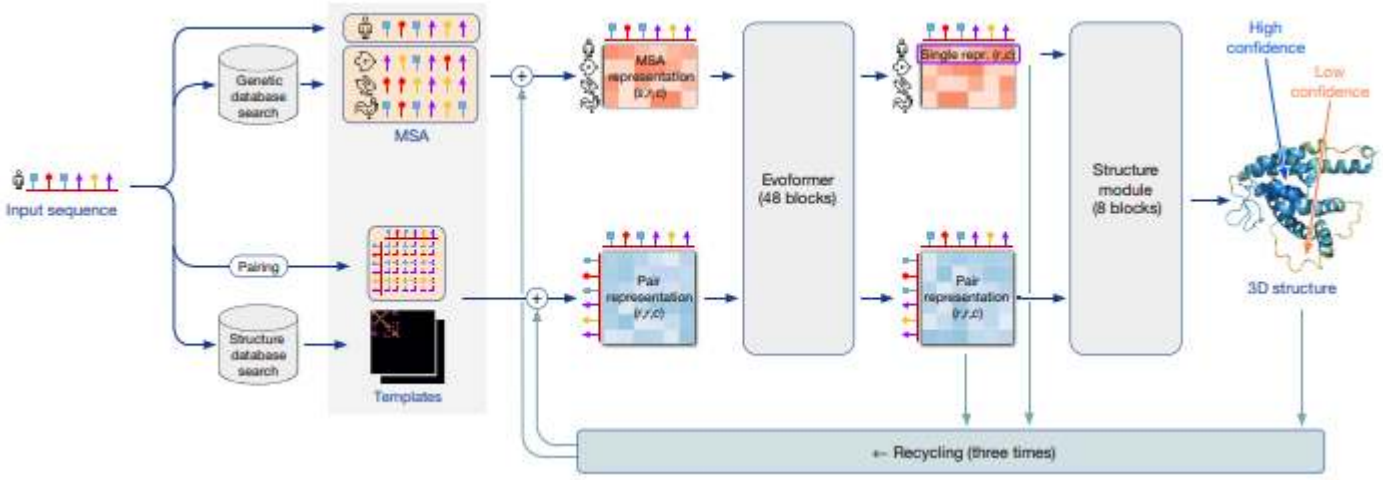


FIGURE 2.12 – Le processus de la méthode AlphaFold2

Le code est également mis en open source sur GitHub [3] avec les étapes d'installation pour pouvoir l'exécuter et prédire les structures protéiques. Le problème qui se pose c'est qu'il faut avoir beaucoup de ressources pour l'exécuter comme nous le montre la figure 2.13 :

FIGURE 2.13 – Les ressources nécessaire pour exécuter AlphaFold2

2. Les transformers ont été introduits en 2017 par une équipe de Google dans l'article "Attention Is All You Need" [42] et sont devenus les modèles de choix pour les problèmes de traitement du langage naturel, remplaçant les modèles récurrents tels que LSTM.
3. https://github.com/deepmind/alphafold

Pour cette raison DeepMind ont publié une version simplifiée d'AlphaFold2 sur Google Colab [4] qui permet en une touche d'avoir la structure protéique souhaitées.

Alors que la grande majorité des chaînes protéiques peuvent désormais être prédites avec une grande précision grâce au modèle AlphaFold2, la prédiction de la structure quaternaire (4D) reste encore un défi.

5 La prédiction de la structure 4D des protéines

L'objectif de la prédiction de la structure 4D est de prédire la structure d'une protéine constituée de deux chaînes protéiques ou plus. Plusieurs efforts en cours visent à adapter les méthodes 3D aux problèmes 4D comme DeepMind qui ont étendu AlphaFold2 [3] pour construire le modèle AlphaFold Multimer [44]. Il a démontré des performances supérieures par rapport aux approches existantes.

Les approches traditionnelles de la prédiction de la structure multimérique ont eu tendance à s'appuyer sur la modélisation basée sur des modèles et d'amarrage libre [5]. Les 3 principaux entrants de CASP14 Multimers Baker-experimental [46], Venclovas [47] et Takeda-Shitaka lab [48] ont tous utilisé des méthodologies basées sur ces approches [44].

Takeda-Shitaka a utilisé une approche purement basée sur des modèles, tandis que Baker-experimental et Venclovas ont utilisé une combinaison de modèles et d'amarrage libre. Baker-experimental a également utilisé un système basé sur DL pour déduire les contacts inter-chaînes à partir des informations de coévolution [44].

Inspiré par le récent succès d'AlphaFold2 au CASP14, il y a eu un certain nombre de tentatives pour appliquer le réseau AlphaFold2 entraîné à la prédiction de structure complexe comme RoseTTAFold [30], Bryant et al. [49], GRAMM [50], Ghani et al. [51] et Humphreys et al. [52] Toutes ces approches utilisent le système AlphaFold entraîné à une seule chaîne au moment de l'inférence, les prédictions sur les complexes sont obtenues simplement en modifiant l'entrée [44].

6 Discussion

Il existe des centaines de méthodes proposées comme solution au problème de prédiction de la structure des protéines. Évidemment, nous ne pouvions pas citer tout le monde, mais nous avons mentionné ceux qui, selon nous et les articles que nous avons lus, étaient performants. Torrisi et al. [53] regroupent dans leur article les approches récentes basées sur le deep learning pour les annotations structurelles de protéines 1D et 2D. Ainsi, il existe d'autres articles récents qui mettent en évidence les développements récents dans l'application de Deep Learning pour la prédiction de la structure des protéines 3D [39] [34] [25].

En raison du manque de ressources (GPU et espace de stockage), nous n'avons pas pu exécuter AlphaFold 2 [3] et RoseTTAFold [30] qui utilisent un modèle de deep learning transformer, nous sommes donc passés à AlphaFold 1 [7] qui prédit la matrice de distance, mais nous avons rencontré le même problème de ressources.

4. https://colab.research.google.com/github/sokrypton/ColabFold/blob/main/AlphaFold2.ipynb

5. L'amarrage moléculaire est une méthode qui calcule l'orientation préférée d'une molécule vers une seconde lorsqu'elles sont liées pour former un complexe stable. Connaître l'orientation préférée sert à prévoir la solidité de l'union entre deux molécules [45]

Nous nous sommes donc mis à la recherche d'une méthode d'une part pertinente et d'autre part exécutable et modifiable. Nous sommes passés par des dizaines de méthodes, sur certaines nous avons rencontré le même problème de ressources et d'autres étaient compliquées à appréhender.

Dans ce travail, nous nous sommes orientés vers une méthode qui permet la prédiction de la matrice de contact [54]. Cette méthode se base sur l'approche RaptorX-Contact [4] (vu dans la section 3.2.1). Nous rappelons que les CM ont permis de grandes avancées dans le domaine de la prédiction de la structure des protéines. Dans CASP12, l'approche RaptorX-Contact [4] a été la première a dépassé le seuil de 50%, soit près de deux fois plus de précision de la prédiction que la compétition CASP11, et en CASP13, sa nouvelle version s'est classé la 2ème dans la catégorie de la prédicition des CM.

C'est intéressant que les techniques actuelles basées sur DL fournissent une avancée significative dans les prédictions de la structure des protéines 1D-4D, ainsi que dans de nombreux autres problèmes connexes. Cependant, cela ne signifie pas qu'ils ont finalement "résolu" le problème du repliement des protéines.

7 Conclusion

Dans ce chapitre nous avons présenté les approches récentes les plus pertinentes pour la prédiction de la structure des protéines, ainsi que les différents outils basés sur ces approches et sur le deep learning. Nous avons clôturé par une discussion.

Dans le chapitre suivant nous présentons l'apprentissage automatique et l'apprentissage profond ainsi que leurs différents algorithmes.

CHAPITRE

3

L'APPRENTISSAGE AUTOMATIQUE

1 Introduction

L'apprentissage automatique, également appelé apprentissage machine ou apprentissage artificiel et en anglais machine learning, est un sous-domaine de l'intelligence artificielle qui donne aux ordinateurs la possibilité d'apprendre sans être explicitement programmés. Il s'intéresse à la construction d'algorithmes qui s'appuient sur un ensemble de données de certains phénomènes. Ces données peuvent provenir de la nature, être fabriquées à la main par l'homme ou générées par un autre algorithme.

Tom Mitchell en donne une définition un peu plus moderne à l'apprentissage automatique : "on dit qu'un programme informatique apprend de l'expérience E en ce qui concerne une tâche T et une mesure de performance P, si sa performance sur T, mesurée par P, s'améliore avec l'expérience E" [55].

On divise généralement l'apprentissage machine en trois grandes catégories : l'apprentissage supervisé, non supervisé et par renforcement [56].

1.1 L'apprentissage supervisé

Dans l'apprentissage supervisé, les données de formation qu'on fournie à l'algorithme comprennent les solutions, appelées étiquettes (ou labels en anglais).

Une tâche d'apprentissage supervisée typique est la classification. Elle consiste à examiner les caractéristiques d'un élément nouvellement présenté afin de l'affecter à une classe d'un ensemble prédéfini. L'un des exemples est le filtre anti-spam : il est formé avec de nombreux exemples d'e-mails avec leur classe (spam ou not spam), et il doit apprendre à classer les nouveaux e-mails, comme présenté dans la figure 3.1 [55].

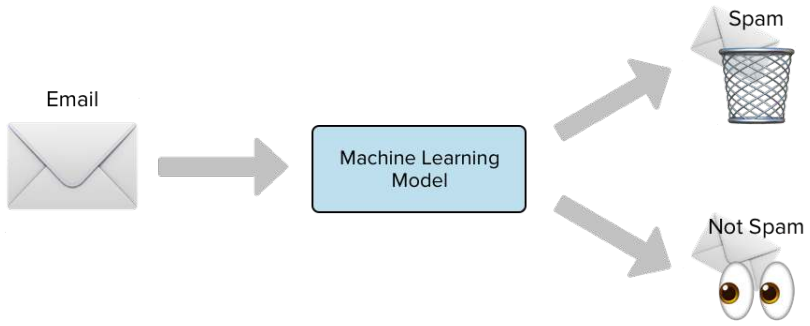


FIGURE 3.1 – Classification Spam/non spam

Une autre tâche typique consiste à prédire une valeur numérique cible, tel que le prix d'une voiture, en fonction d'un ensemble de caractéristiques (kilométrage, âge, marque, etc.) appelées prédicteurs. Ce type de tâche est appelé régression. Pour entraîner le système, nous devons lui donner de nombreux exemples de voitures, y compris leurs prédicteurs et leurs étiquettes (c'est-à-dire leurs prix) [55]. La figure 3.2 décrit une tâche de régression.

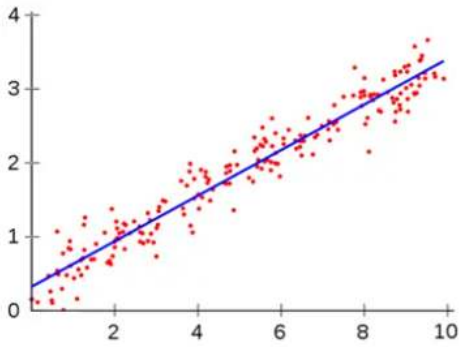


FIGURE 3.2 – Regression

1.2 L'apprentissage non supervisé

Dans l'apprentissage non supervisé, les données d'apprentissage ne sont pas étiquetées. Le modèle n'a pas de « réponses » dont il peut tirer des enseignements ; il doit donner un sens aux données en fonction des observations elles-mêmes.

L'apprentissage non supervisé nous permet d'aborder les problèmes avec peu ou pas d'idée de ce à quoi nos résultats devraient ressembler. Nous pouvons obtenir une structure à partir de données dont nous ne connaissons pas nécessairement l'effet des variables. Les tâches typiques d'apprentissage non supervisé sont le clustering et la réduction de dimension [55].

1.2.1 Le clustering

Le clustering est une segmentation d'une population hétérogène en sous-populations homogènes. Les sous-populations ne sont pas préétablies (Figure 3.3).

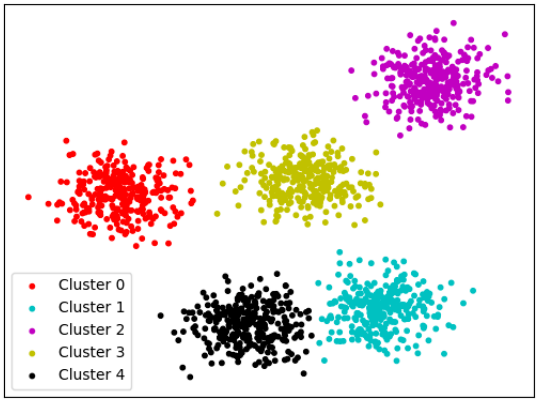


FIGURE 3.3 – Clustering

1.2.2 La réduction de la dimension

L'objectif de la réduction de la dimension est de simplifier les données sans perdre trop d'informations. Une façon d'y parvenir est de fusionner plusieurs caractéristiques corrélées en une seule. Par exemple, le kilométrage d'une voiture peut être très corrélé avec son âge, de sorte que l'algorithme de réduction de la dimensionnalité les fusionne en une seule caractéristique qui représente l'usure de la voiture. C'est ce qu'on appelle l'extraction de caractéristiques [55].

1.3 L'apprentissage par renforcement

Dans l'apprentissage par renforcement (Apprentissage par renforcement - Reinforcement learning (RL)), le modèle perçoit l'environnement, prend des mesures et effectue des ajustements et des choix basés sur l'état et la récompense ou la punition.

Le scénario général de l'apprentissage par renforcement est illustré par la figure 3.4.

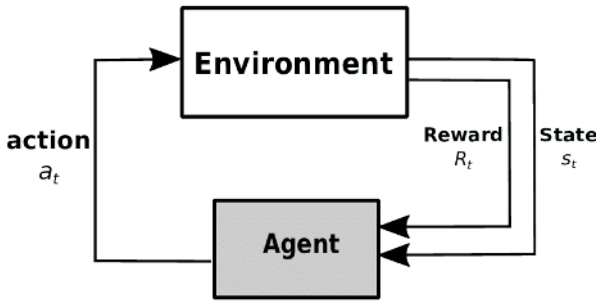


FIGURE 3.4 – Processus de RL

Le programme AlphaGo de DeepMind est un bon exemple d'apprentissage par renforcement. En mars 2016, il a battu le champion du monde Lee Sedol au jeu de Go. Il a appris sa politique gagnante en analysant des millions de parties, puis en jouant de nombreuses parties contre lui-même [55].

La figure 3.5 résume les algorithmes d'apprentissage automatique les plus importants de chaque catégorie.

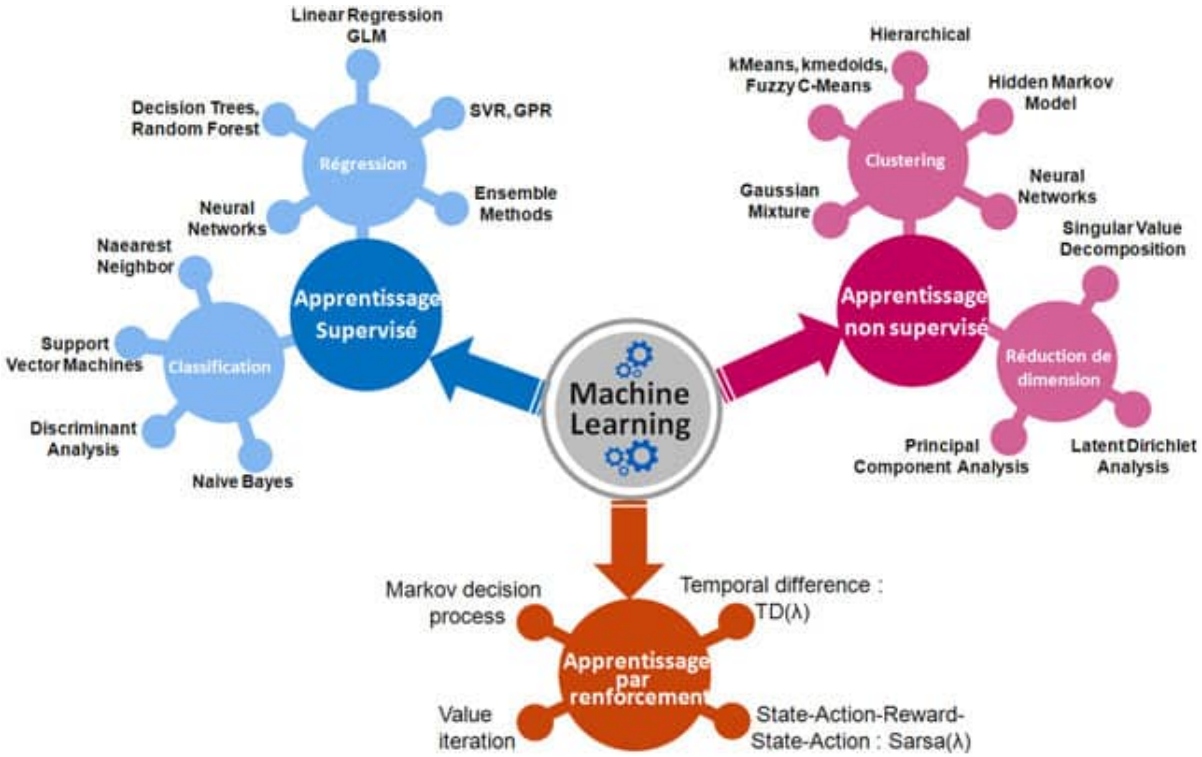


FIGURE 3.5 – Les types du machine learning

Lorsque les chercheurs ont atteint le point culminant de l'apprentissage automatique, le domaine de l'apprentissage profond (DL) est apparu. Une des grandes différences entre l'apprentissage profond et les algorithmes d'apprentissage automatique traditionnels c'est qu'il s'adapte bien, plus la quantité de données fournie est grande plus les performances d'un algorithme d'apprentissage profond sont meilleures [55] (Figure 3.6).

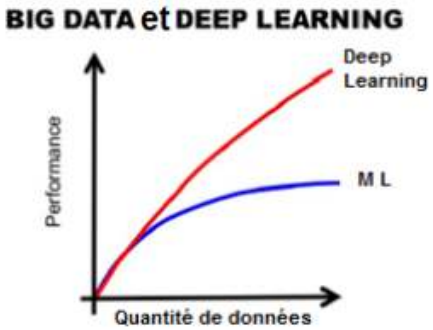


FIGURE 3.6 – Machine learning et deep learning

Autre différence entre l'apprentissage automatique et l'apprentissage profond c'est l'étape de l'extraction de caractéristiques. Dans les algorithmes d'apprentissage automatique traditionnels l'extraction de caractéristiques est faite manuellement, c'est une étape difficile et coûteuse en temps et requiert un spécialiste en la matière alors qu'en apprentissage profond cette étape est exécutée automatiquement par l'algorithme [57] (Figure 3.7).

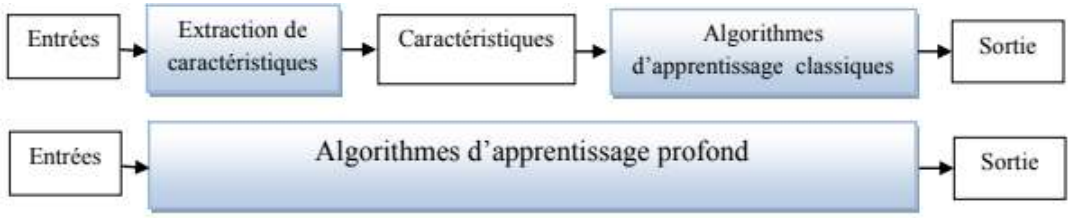


FIGURE 3.7 – L'étape extraction de données en machine learning et en deep learning

Alors, c'est quoi l'apprentissage profond ?

2 L'apprentissage profond

L'apprentissage profond est un sous-domaine de l'apprentissage automatique qui intègre des réseaux de neurones en couches successives afin d'apprendre des données de manière itérative, il est particulièrement utile lorsqu'on tente de détecter des tendances à partir de données non structurées [57] (Figure 3.8).

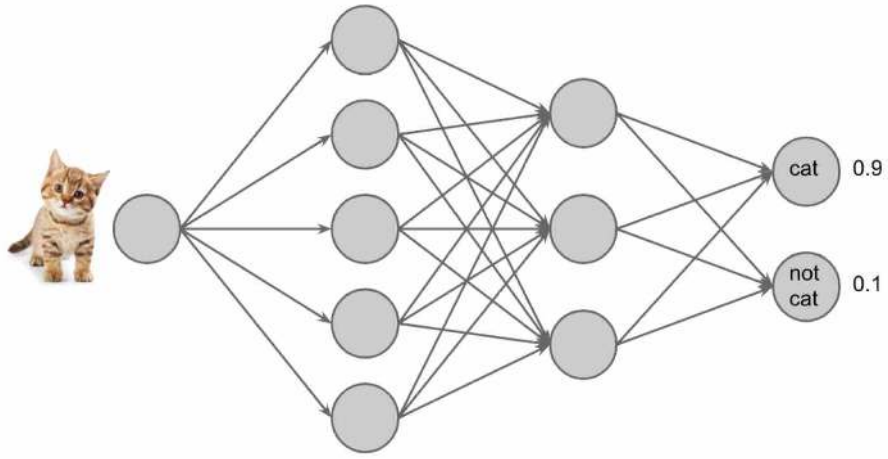


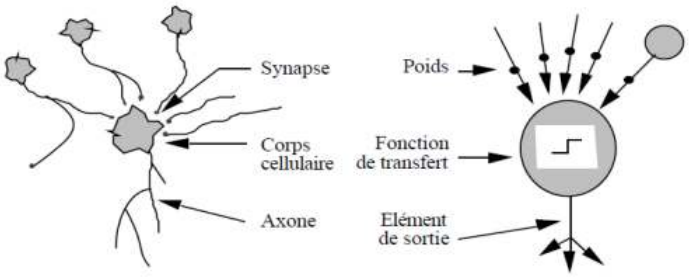
FIGURE 3.8 – Un réseau de neurones

2.1 Les réseaux de neurones

Les réseaux de neurones ont été développés pour simuler le système nerveux humain pour des tâches d'apprentissage automatique en traitant les unités de calcul dans un modèle d'apprentissage d'une manière similaire aux neurones humains. La figure 3.9 montre l'analogie entre le neurone biologique et le neurone artificiel [57].

Les réseaux de neurones sont composés de dizaines voire de centaines de couches de neurones, chacune recevant et interprétant les informations de la couche précédente. Un neurone est une fonction algébrique non linéaire et bornée, dont la valeur dépend de paramètres appelés coefficients ou poids. Les variables de cette fonction sont habituellement appelées "entrées" du neurone, et la valeur de la fonction est appelée sa "sortie".

Un neurone peut être représenté graphiquement comme indiqué sur la Figure 3.10 [57].



Neurone biologie	Neurone artificiel
Synapses	Poids de connexions
Axones	Signal de sortie
Dendrites	Signal d'entrée
Somma	Fonction d'activation

FIGURE 3.9 – L'analogie entre le neurone biologique et le neurone artificiel

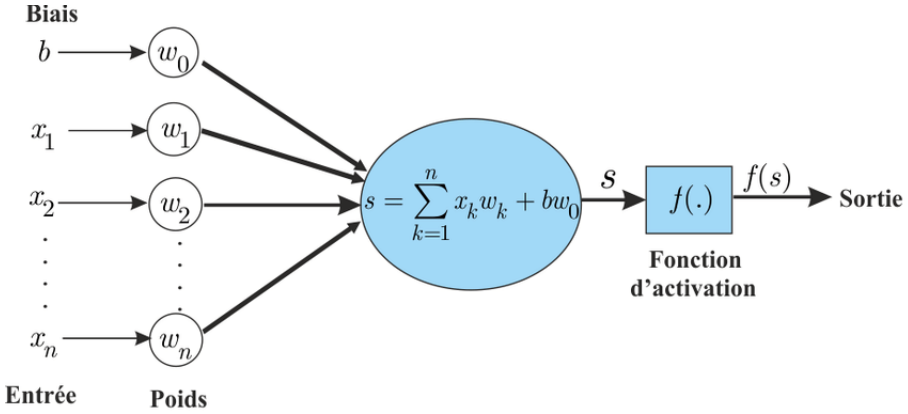


FIGURE 3.10 – Un neurone artificiel

— **Toutes les entrées (xj) :** sont directement les entrées du système.
— **Biais (b=W0) :** les entrées qui sont toujours mises à 1.
— **Poids (Wi) :** sont les facteurs multiplicateurs qui affectent l'influence de chaque entrée sur la sortie de neurone.
— **Noyau (Somme pondérée + Fonction d'activation) :**

$$F(x) = \sum_{i=0}^{n} (XiWi) - W0$$

.

2.1.1 La fonction d'activation

La fonction d'activation (ou fonction de transfert) sert à convertir le résultat de la somme pondérée des entrées d'un neurone en une valeur de sortie, cette conversion s'effectue par un calcul de l'état du neurone en introduisant une non-linéarité dans le fonctionnement du neurone. Le biais

b joue un rôle de seuil, quand le résultat de la somme pondérée dépasse ce seuil, l'argument de la fonction de transfert devient positif ou nul ; dans le cas contraire, il est considéré négatif [57].

Finalement si le résultat de la somme pondérée est :

— En dessous du seuil, le neurone est considéré comme non-actif.

— Aux alentours du seuil, le neurone est considéré en phase de transition.

— Au-dessus du seuil, le neurone est considéré comme actif.

Il existe plusieurs choix de fonctions d'activation, les plus courantes étant la fonction sigmoïde et la rectified linear unit (ReLU).

— **Sigmoïde**

Elle est la plus utilisée car elle introduit de la non-linéarité, mais c'est aussi une fonction continue, différentiable [57]. La fonction sigmoïde est définie dans la figure 3.11.

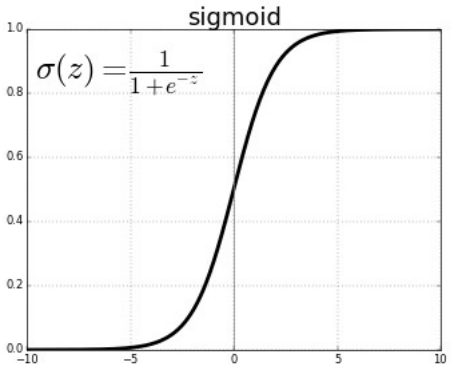


Figure 3.11 – La fonction sigmoïde

— **ReLU**

ReLU utilise la fonction f(z) = max(0,z), ce qui signifie que si la sortie est positive, elle produira la même valeur, sinon elle produira 0 [55]. La plage de sortie de la fonction est illustrée dans le visuel suivant (Figure 3.12) :

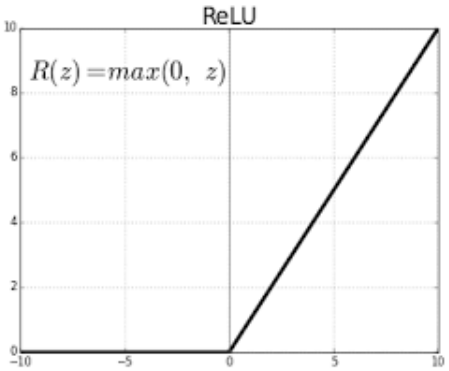


Figure 3.12 – La fonction ReLU

Après avoir construit un réseau de neurones, nous devons utiliser des métriques pour l'évaluer, comme la fonction de perte qui est utilisée comme une mesure pour l'aider à comprendre s'il apprend dans la bonne direction.

2.2 La fonction de perte

Pour formuler la fonction de perte (loss function) en termes simples, on peut considérer la note obtenue par un étudiant lors d'un examen. Supposons que cet étudiant se soit présenté à plusieurs tests sur le même sujet : quelle est la métrique à utiliser pour comprendre la performance pour chaque test ? Évidemment, le score du test. Supposons que cette étudiant à obtenu 56, 60, 78, 90 et 96 sur 100 dans cinq tests consécutifs, on peut constater clairement que l'amélioration des résultats est une indication des performances. De même, comment un réseau peut-il savoir s'il améliore son processus d'apprentissage à chaque itération ? Il utilise la fonction de perte, qui est analogue à la note de l'examen.

La fonction de perte mesure essentiellement la perte par rapport à la cible. Supposons que l'on développe un modèle pour prédire si un élève va réussir ou échouer et que la chance de réussir ou d'échouer soit définie par la probabilité. Ainsi, 1 indiquerait qu'il réussira avec une certitude de 100% et 0 indiquerait qu'il échouerait définitivement. Le modèle tire les leçons des données et prédit un score de 0,87 pour que l'élève réussisse. La perte réelle serait donc ici de 1,00 - 0,87 = 0,13. S'il répète l'exercice avec quelques mises à jour des paramètres afin de s'améliorer et obtient maintenant une perte de 0,40, il comprendrait que les changements qu'il a apportés n'aident pas le réseau à apprendre de manière appropriée [55].

En fonction du type de résultat des données, nous avons plusieurs fonctions de perte standard définies dans l'apprentissage automatique. Voici quelques unes :

— **Mean Squared Error (MSE) :** Différence moyenne au carré entre la valeur réelle et la valeur prévue. Le carré de la différence permet de pénaliser facilement le modèle pour une différence plus élevée. Ainsi, une différence de 3 entraînerait une perte de 9, mais une différence de 9 entraînerait une perte de 81.

$$\sum_{i=1}^{D}(x_i - y_i)^2$$

— **Mean Absolute Error (MAE) :** L'erreur moyenne absolue entre la valeur réelle et la valeur prédite.

$$\sum_{i=1}^{D}|x_i - y_i|$$

— **Binary cross-entropy :** Définit la perte lorsque les résultats catégoriques sont une variable binaire, c'est-à-dire avec deux résultats possibles : (réussite/échec) ou (oui/non), généralement utilisé pour des modèles de régression logistique.

$$(y \log(p) + (1 - y) \log(1 - p))$$

— **Categorical cross-entropy :** Définit la perte lorsque les résultats de la catégorie sont non binaires, c'est-à-dire >2 résultats possibles : (Oui/Non/Peut-être) ou (Type 1/ Type 2/... Type n).

$$\sum_{c=1}^{M} y_{o,c} \log(p_{o,c})$$

Dans un réseau de neurones, la fonction de perte prend les prédictions du réseau et la cible réelle et calcule un score de distance, en capturant la performance du réseau, comme le montre la figure 3.13 [55]. Sa valeur est mise à jour par un optimiseur.

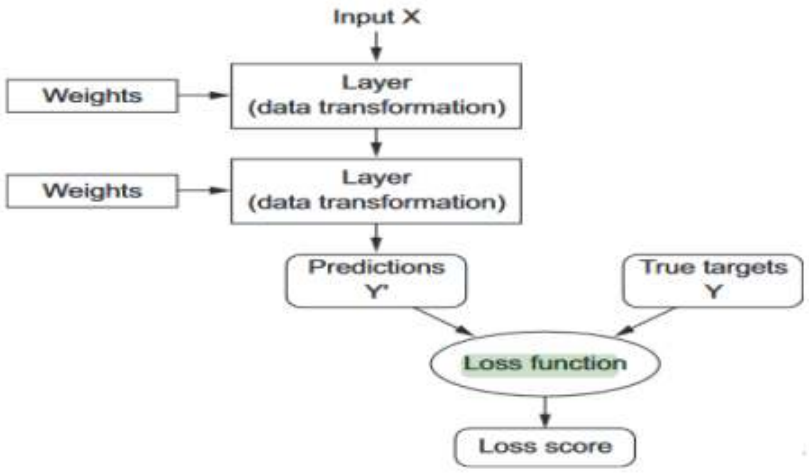


FIGURE 3.13 – La fonction de perte

2.3 Les optimiseurs

Pendant la formation, nous essayons de minimiser la perte de fonction et de mettre à jour les paramètres pour améliorer la précision. Les paramètres du réseau neuronal sont généralement les poids de liaison.

Ainsi, l'optimiseur est une méthode qui utilise le score obtenu par la fonction de perte pour ajuster un peu la valeur des poids, dans un sens qui fera baisser le score de perte et accélérer l'apprentissage, comme le montre la figure 3.14 [55].

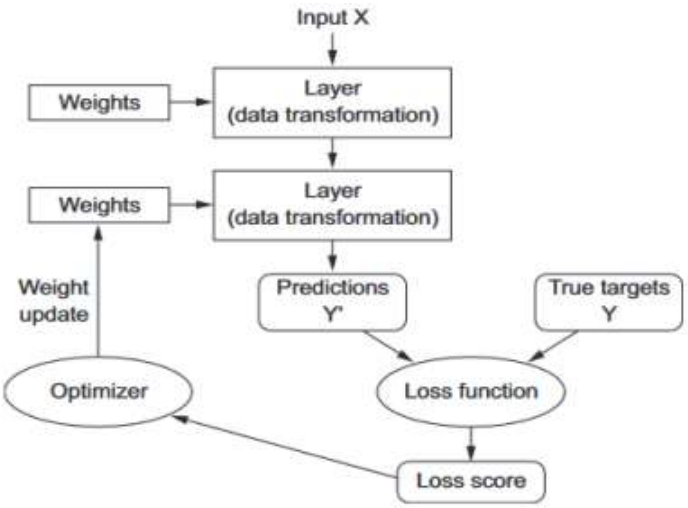


FIGURE 3.14 – L'optimiseur

ADAM, qui signifie « Adaptive Moment Estimation », est de loin l'optimiseur le plus populaire et le plus utilisé dans le domaine de l'apprentissage profond.

2.3.1 ADAM

Dans la plupart des cas, nous pouvons choisir l'optimiseur ADAM à l'aveuglette et oublier les autres possibilités d'optimisation. Cette technique d'optimisation calcule un taux d'apprentissage adaptatif pour chaque paramètre. Elle définit la dynamique et la variance du gradient de la perte et exploite un effet combiné pour mettre à jour les paramètres de poids [55].

Après que nous avons présenté les grandes lignes de l'apprentissage automatique et l'apprentissage profond, il est temps de présenter l'algorithme auto-encodeur basé sur les réseaux de convolution utilisé dans ce projet.

2.4 L'auto-encodeur

Les auto-encodeurs sont des algorithmes d'apprentissage non supervisé à base de réseaux de neurones artificiels, qui permettent de construire une nouvelle représentation d'un jeu de données.

L'architecture d'un auto-encodeur est constitué de deux parties : l'encodeur et le décodeur. L'encodeur est constitué par un ensemble de couches de neurones, qui traitent les données afin de construire de nouvelles représentations dites "encodées". À leur tour, les couches de neurones du décodeur reçoivent ces représentations et les traitent afin d'essayer de reconstruire les données de départ [58].

L'architecture la plus simple d'un auto-encodeur est semblable à un perceptron multicouches. La figure 3.15 schématise un auto-encodeur simple.

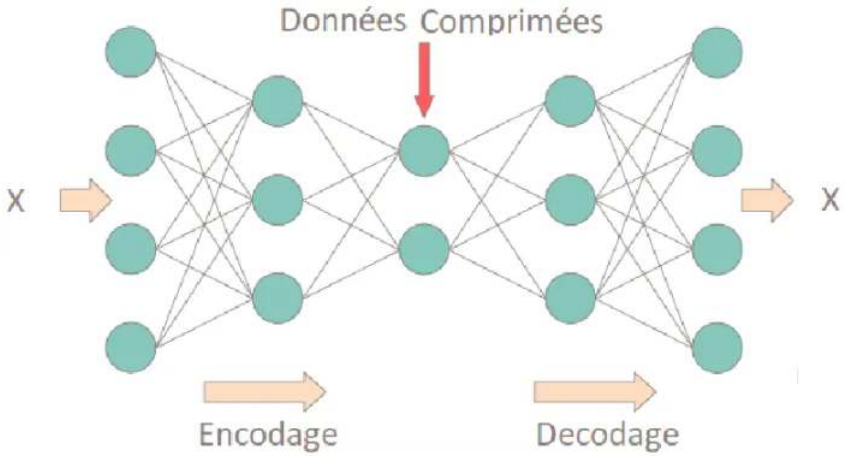


FIGURE 3.15 – L'architecture de l'auto-encodeur

Cependant, en fonction des données traitées, on peut utiliser différentes topologies de réseaux de neurones. Par exemple, des **couches convolutives** afin d'analyser des images ou des **couches de neurones récurrentes** pour traiter des séries temporelles ou des séquences.

2.5 Les réseaux de neurones à convolution

Les réseaux de neurones à convolution (Convolution Neural Network (CNN)) sont considérés comme un type spécialisé de réseau de neurones pour le traitement de données ayant une topologie

semblable à une grille, qui peuvent être considérées comme une grille 1D (Vecteur) et grille 2D de pixels (Matrice).

Les réseaux à convolution ont connu un succès considérable dans les applications pratiques nous citons par exemple reconnaissance de l'image et de la vidéo, les systèmes de recommandations et le traitement du langage naturel etc.

Généralement, il existe quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche fully-connected [57].

2.5.1 La couche de convolution (CONV)

Le rôle de cette première couche est d'analyser les images fournies en entrée et de détecter la présence d'un ensemble de caractéristiques [59].

Comme le montre la figure 3.16, la matrice la plus à gauche est la matrice d'entrée. Celui du milieu est généralement appelé matrice du noyau 'kernel'. La convolution est appliquée à cette matrice et le résultat est présenté comme la matrice la plus à droite [57].

FIGURE 3.16 – La couche de convolution (CONV)

Le processus de convolution est un produit élément par élément suivi d'une somme, comme illustré dans l'exemple. L'opération de convolution est généralement appelée noyau 'kernel' (Figure 3.17). Par différents choix de noyaux 'kernel', différentes opérations sur les images pourraient être réalisées. Les opérations incluent généralement identité, détection des contours, flou, netteté, etc. En introduisant des matrices aléatoires comme convolution opérateur, certaines propriétés intéressantes pourraient être découvertes [57].

FIGURE 3.17 – L'opération de convolution (kernel)

Lorsque nous parlons des CNN, nous nous référons généralement à un CNN bidimensionnel 2D qui est utilisé pour la classification des images. Mais il existe deux autres types de réseaux de neurones à convolution utilisés dans le monde réel, qui sont les CNN unidimensionnels 1D et tridimensionnels 3D [60]. Dans ce projet, le CNN unidimensionnel et bidimensionnel sera utilisé.

— **1D CNN**

Dans 1D CNN (Conv1D), le noyau se déplace dans 1 direction. Conv1D est principalement utilisé sur les données de séries chronologiques, les données sensorielles, et les données de l'accéléromètre.

La figure 3.18 montre des données recueillies à partir d'un accéléromètre qu'une personne porte au bras. Les données représentent l'accélération dans les 3 axes [60].

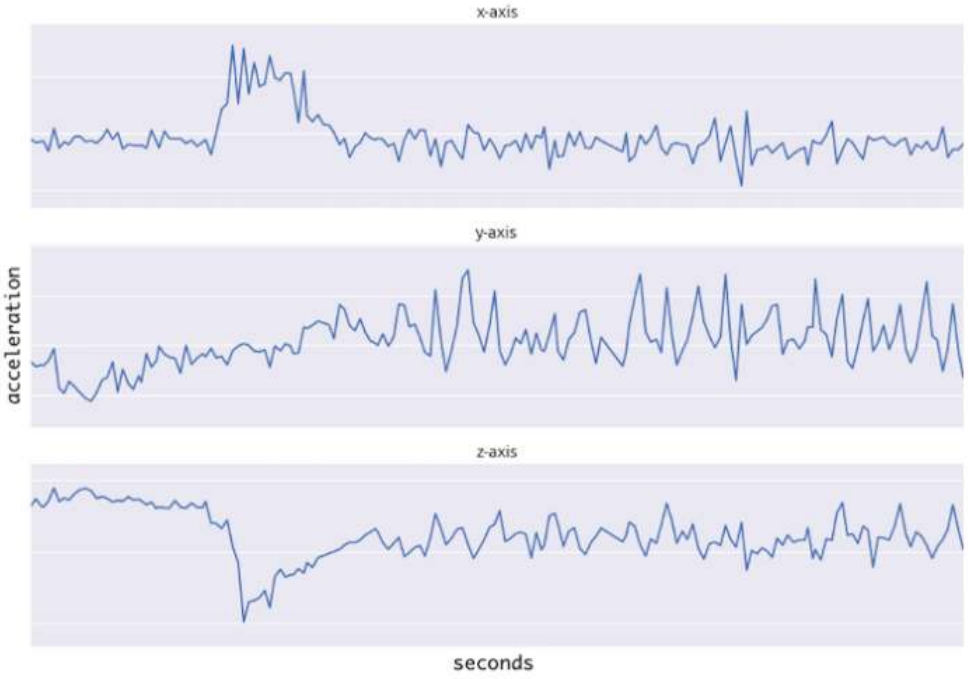


FIGURE 3.18 – Données de séries chronologiques d'un accéléromètre

1D CNN peut effectuer une tâche de reconnaissance d'activité à partir des données de l'accéléromètre, par exemple si la personne est debout, marche, saute, etc. Ces données ont 2 dimensions. La première dimension est le pas de temps et l'autre est les valeurs de l'accélération en 3 axes. Le graphique dans la figure 3.19 illustre comment le noyau se déplacera sur les données de l'accéléromètre. Chaque ligne représente l'accélération de la série chronologique pour certains axes. Le noyau ne peut se déplacer que dans une dimension le long de l'axe du temps [60].

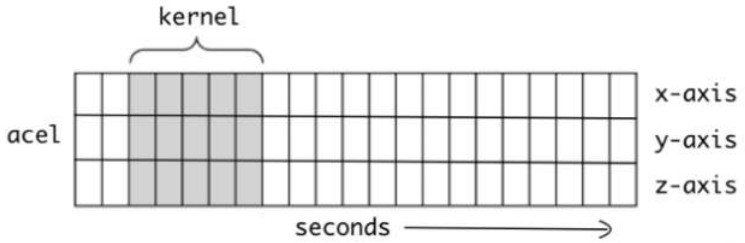


FIGURE 3.19 – Kernel glissant sur les données de l'accéléromètre

— **2D CNN**
Dans 2D CNN (Conv2D), le noyau se déplace dans 2 directions. Conv2D est généralement utilisé sur les données Image. Il est appelé CNN bidimensionnel car le noyau glisse le long de 2 dimensions sur les données, comme indiqué dans la figure 3.20 [60].

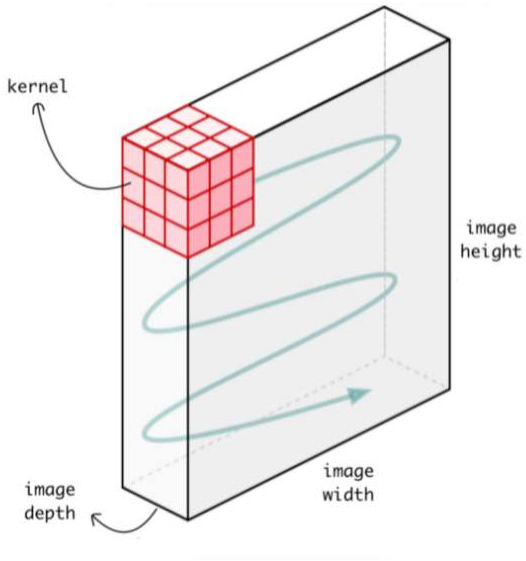


FIGURE 3.20 – Kernal glissant sur l'image

L'avantage d'utiliser CNN est qu'il peut extraire les caractéristiques spatiales des données à l'aide de son noyau, ce que d'autres réseaux sont incapables de faire. Par exemple, CNN peut détecter les bords, la distribution des couleurs, etc. dans l'image, ce qui rend ces réseaux très robustes dans la classification des images et d'autres données similaires contenant des propriétés spatiales [60].

— **3D CNN**
Dans CNN 3D (Conv3D), le noyau se déplace dans 3 directions comme le montre la figure 3.21.

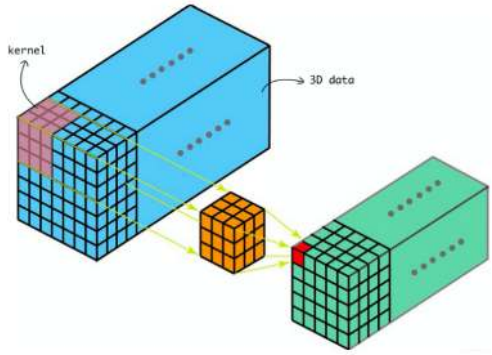


FIGURE 3.21 – Kernel glissant sur des données 3D

Conv3D est principalement utilisé avec des données d'image 3D. Comme les données d'imagerie par résonance magnétique (IRM) (Figure 3.22). Les données IRM sont largement utilisées pour examiner le cerveau, la moelle épinière, les organes internes et bien d'autres.

Une tomodensitométrie (TDM) est également un exemple de données 3D, qui sont créées en combinant une série d'images radiographiques prises sous différents angles autour du corps. Nous pouvons utiliser Conv3D pour classer ces données médicales ou en extraire des caractéristiques [60].

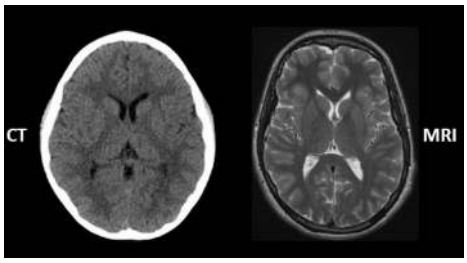


FIGURE 3.22 – Coupe transversale de l'image 3D du scanner et de l'IRM

2.5.2 La couche de pooling

La couche de Pooling est une opération généralement appliquée entre deux couches de convolution. Celle-ci reçoit en entrée les features maps formées en sortie de la couche de convolution et son rôle est de réduire la taille des images, tout en préservant leurs caractéristiques les plus essentielles [59].

Parmi les plus utilisés, on retrouve le max-pooling qui prend la valeur maximale de la fenêtre de filtre, ou encore l'average pooling dont l'opération consiste à conserver à chaque pas, la valeur moyenne de la fenêtre de filtre. La figure 3.23 montre max pooling sur une fenêtre 2×2 [57].

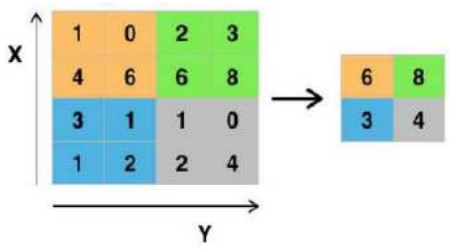


FIGURE 3.23 – Max pooling

2.5.3 La couche d'activation ReLU (Rectified Linear Units)

Cette couche remplace toutes les valeurs négatives reçues en entrées par des zéros. L'intérêt de ces couches d'activation est de rendre le modèle non linéaire et de ce fait plus complexe [59].

2.5.4 Couche entièrement connectée

Après plusieurs couches de convolution et de max-pooling, le raisonnement de haut niveau dans le réseau neuronal se fait via des couches entièrement connectées (ou Couche Fully Connected (FC)). Ces couches sont placées en fin d'architecture de CNN et sont entièrement connectées à tous les neurones de sorties (d'où le terme fully-connected) [59].

La figure 3.24 montre l'architecture d'un réseau de convolution pour la classification des images.

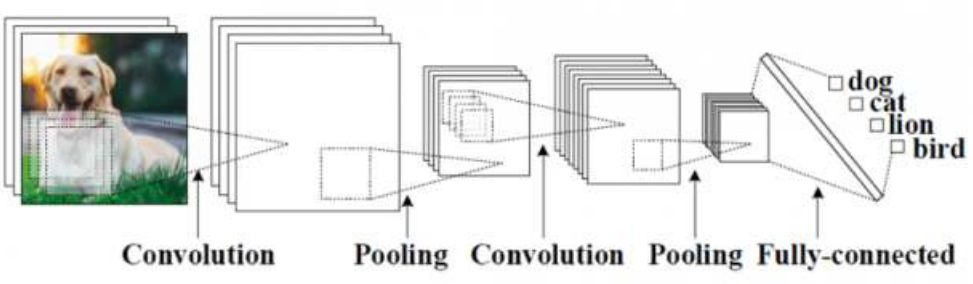


FIGURE 3.24 – Un réseau de convolution pour la classification des images

2.6 Les réseaux récurrents

Un réseau de neurones récurrents (Recurrent Neural Network RNN) est un réseau de neurones artificiels présentant des connexions récurrentes. Il est constitué d'unités (neurones) interconnectées interagissant non linéairement et pour lesquelles il existe au moins un cycle dans la structure. Les architectures de réseaux neuronaux récurrents peuvent prendre de nombreuses formes différentes. Un type commun consiste en un perceptron multicouche standard (MLP) plus des boucles ajoutées (Figure 3.25) [57].

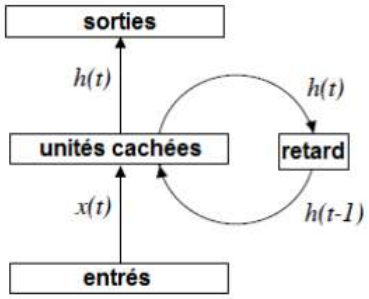


FIGURE 3.25 – L'architecture d'un réseau de neurones récurrents

Notez que le temps doit être discrédité, les activations étant mises à jour à chaque pas de temps. Une unité de délai doit être introduite pour conserver les activations jusqu'à leur traitement au prochain pas de temps [57].

3 Conclusion

Dans ce chapitre, nous avons présenté de manière globale les principaux axes de l'apprentissage automatique et l'apprentissage profond, leurs types et quelques algorithmes. Nous avons par la suite détaillé les auto-encodeurs qui sont les réseaux de neurones utilisés dans notre approche.

CHAPITRE

4

CONCEPTION ET IMPLÉMENTATION

1 Introduction

Le domaine de la prédiction de la structure des protéines a connu de nombreuses avancées grâce aux approches basées sur le deep learning (DL), comme le succès de l'approche RaptorX-Contact [5] en CASP12 dans la prédiction des matrices de contact, puis AlphaFold1 [7] en CASP13 dans la prédiction des matrices de distance, et plus récemment, AlphaFold2 [3] en CASP14 qui prédit directement les structures des protéines à partir de leurs séquences d'acides aminés sans passer par la prédiction des matrices de contact ou de distance.

Malheureusement, nous n'avons pas pu exécuter ces approches récentes en raison du manque de ressources (GPU et espace de stockage). Pour rappel, AlphaFold2 nécessite 3 TB de disque pour le stockage et multiple GPU pour l'exécution.

Nous nous sommes donc mis à la recherche d'une méthode d'une part pertinente et d'autre part exécutable et modifiable. Nous sommes passés par des dizaines de méthodes, sur certaines nous avons rencontré le même problème de ressources et d'autres étaient compliquées à appréhender. Enfin, nous nous sommes orientés vers une méthode qui permet la prédiction de la matrice de contact. Cette méthode se base sur l'approche RaptorX-Contact [4] (vu dans le chapitre 2 section 3.2.1) et est disponible sur GitHub [1]. Cette approche utilise le modèle de deep learning auto-encodeur basé sur les réseaux de convolution 1D et 2D (vu dans le chapitre 3 sections 3.4 et 3.5).

Dans ce chapitre, nous allons présenter cette méthode ainsi que les modifications que nous y avons apportées, mais avant cela nous présentons les outils utilisés dans notre travail.

1. https://github.com/lakmalnd/deep-protein-structure-modeling

2 Présentation des outils d'implémentation

Pour l'implémentation de cet algorithme, le langage Python a été utilisé, ainsi, nous l'avons exécuté sur la plateforme kaggle qui offre un accès gratuit au GPU de 30 heures par semaine.

2.1 Le langage python

Python est un langage de programmation interprété et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il s'est imposé comme le langage de référence pour les applications de machine learning car il est relativement puissant et dispose de plus de très nombreuses librairies qui permettent d'appeler des fonctions préprogrammées (Figure 4.1) [61].

FIGURE 4.1 – Python

Les bibliothèques utilisées dans l'algorithme sont : Numpy pour la manipulation des matrices, Matplotlib pour la visualisation des données et Tensorflow pour la création du modèle de deep learning (Figure 4.2).

```
import numpy as np
import matplotlib.pyplot as plt

from tensorflow.python.keras.layers import Embedding, Conv1D, MaxPooling1D, Dropout, Conv2DTranspose,
from tensorflow.python.keras import Sequential

import tensorflow.keras.backend as K

import tensorflow as tf
```

FIGURE 4.2 – Les bibliothèques utilisées

2.1.1 Numpy

NumPy (Numerical Python) est une bibliothèque de python qui comporte des fonctions permettant de manipuler des matrices ou tableaux multidimensionnels. Il permet d'effectuer rapidement et efficacement les opérations par rapport aux listes Python. Les tableaux NumPy utilisent d'abord moins de mémoire et d'espace de stockage, ce qui le rend plus avantageux que les tableaux traditionnels de python (Figure 4.3) [62].

FIGURE 4.3 – Numpy

2.1.2 Matplotlib

Matplotlib est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques [63].

On peut générer en quelques lignes de code des graphes, histogrammes, des spectres de puissance, des graphiques à barres, des graphiques d'erreur, des nuages de dispersion, etc (Figure 4.4) [64].

FIGURE 4.4 – Matplotlib

2.1.3 Tensorflow

TensorFlow est une bibliothèque open source, permettant d'exécuter des applications de machine learning et de deep learning. Cet outil dédié à l'apprentissage automatique a été développé par Google, et est fortement utilisé dans le domaine de l'intelligence artificielle (IA). Elle regroupe des modèles et des algorithmes de machine learning et de deep learning et simplifie le processus d'acquisition de données, d'entraînement des modèles, de génération de prédictions et de raffinement des résultats futurs (Figure 4.5) [65] [66].

FIGURE 4.5 – Tensorflow

2.2 La plateform kaggle

Kaggle est une plateforme web qui accueille la plus grande communauté de Data Science au monde. Il offre un environnement Jupyter Notebooks personnalisable et sans configuration. Sont accessibles gratuitement des GPU et une grande quantité de données et de codes publiés par la communauté. À l'intérieur de Kaggle, on trouve tout le code et les données dont on a besoin pour réaliser des projets de science des données. Il y a plus de 50 000 jeux de données publics et 400 000 notebooks publics disponibles pour tous (Figure 4.6) [67].

FIGURE 4.6 – kaggle

3 Dataset

Étant donné que les bases de données utilisées pour la prédiction de la structure des protéines sont énormes, cette méthode utilise une version simplifiée du dataset ProteinNet de CASP11 [68]. Ce dataset contient 5525 protéines (3700 pour l'entraînement et 1825 pour le test).

Le dataset est divisé en deux fichiers (Figure 4.7) :

— Le premier contient des chaînes d'acides aminés. Leurs tailles varient entre 64 et 128 acides aminés. Chaque acide aminé est représenté par un chiffre allant de 1 à 20 selon le nombre des acides aminés existant : A : 1, C : 2, D : 3, E : 4 etc.

— Le deuxième contient les coordonnées 3D de l'atome carbone central de chaque acide aminé (Figure 4.8).

```
In [3]: x_read = np.load('../input/datadata/amino_acid_sequences.npz')
        y_read = np.load('../input/datadata/C_alpha_atom_locations.npz')

        x_train = x_read['x_train']
        x_test = x_read['x_test']
        y_train = y_read['y_train']
        y_test = y_read['y_test']

        print(x_train.shape) # shape corresponding to: [Proteins, Amino Acids]
        print(x_test.shape) # shape corresponding to: [Proteins, Amino Acids]
        print(y_train.shape) # shape corresponding to: [Proteins, Amino Acids, x/y/y of C_alpha atoms]
        print(y_test.shape) # shape corresponding to: [Proteins, Amino Acids, x/y/y of C_alpha atoms]

(3700, 128)
(1825, 128)
(3700, 128, 3)
(1825, 128, 3)
```

FIGURE 4.7 – dataset

```
print(x_train[1])
print(y_train[1])
```

```
[ 6 16  7 11 11  9  8  8 16  9  9 20 15 10  4 10 20 16 11 10 18  3 10 10
 12  3 12  8 13 10 20  3  1 10 12  9  8 14 12  4  6 18  6  8 20  3  9 12
  5  8  9 16  8  4 10  8  9  3 15 11  9 16 12 16 16 10 17  3  1 10 17  6
 10  8 13  3  9  4 18 10 11  8 12 18  1  4 12 16  6  9  8 16 16  6  8  1
  1  8 15  9 12  8  8  3  1  3  4  8  9 16  9  1  8 16 16 11  8 17 13 16
  0  0  0  0  0  0  0  0]
[[ 0.0000e+00  0.0000e+00  0.0000e+00]
 [ 0.0000e+00  0.0000e+00  0.0000e+00]
 [ 1.4181e+03 -5.7870e+02  5.1320e+02]
 [ 1.1047e+03 -3.6760e+02  4.8360e+02]
 [ 1.0724e+03  8.2000e+00  4.6190e+02]
 [ 8.4200e+02  1.6000e+02  7.2350e+02]
 [ 7.3910e+02  5.1880e+02  8.0140e+02]
 [ 9.4580e+02  6.4610e+02  1.0956e+03]
 [ 7.2820e+02  6.7860e+02  1.4087e+03]
 [ 5.7360e+02  1.0145e+03  1.4993e+03]
 [ 7.7330e+02  1.0213e+03  1.8246e+03]
 [ 1.1014e+03  9.6190e+02  1.6386e+03]
 [ 1.0189e+03  1.2227e+03  1.3712e+03]
 [ 9.2870e+02  1.4885e+03  1.6313e+03]
```

FIGURE 4.8 – Les données du dataset

3.1 Pré-traitement des données

Pour le pré-traitement des données, il génère les matrices de contact de chaque séquence du dataset en utilisant les coordonnées 3D des atomes de carbone centrale de chaque paire d'acides aminés appelé C-alpha (Figure 4.9). Pour cela, il calcule la distance euclidienne entre ces atomes puis il utilise un seuil de distance de 9 angströms pour créer des cartes de contact binaires (Figure 4.10).

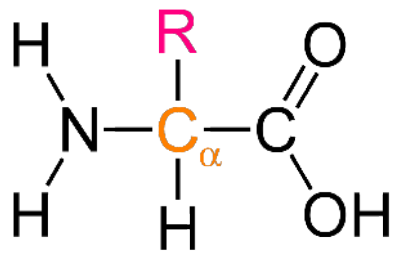


FIGURE 4.9 – Carbone central de l'acide aminé (c-alpha)

```
# distance calculation and thresholding
d_sqr = (y_train[i1,i2,0]-y_train[i1,i3,0])**2 + (y_train[i1,i2,1]-y_train[i1,i3,1])**2 + (y_train[i1,i2,2]-y_train[i1,i3,2])**2
cmap_train[i1, i2, i3, 0] = (d_sqr**0.5) > 900
```

FIGURE 4.10 – Calcul de distance entre les acides aminés

La figure 4.11 montre un exemple d'une matrice de contact d'une séquence du dataset.

```
In [6]: rand_idx = np.random.randint(0,cmap_test.shape[0])

        plt.imshow(cmap_test[rand_idx,:,:,0])
        plt.show()
```

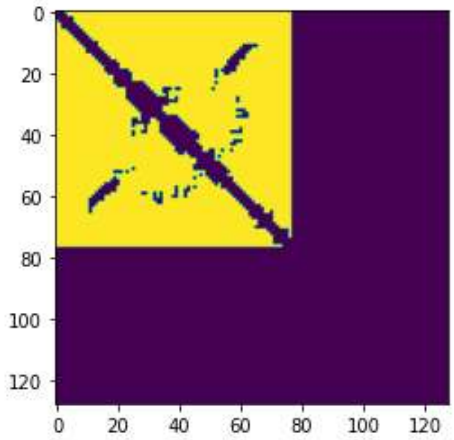


FIGURE 4.11 – Matrice de contact d'une séquence d'acides aminés

4 Création du modèle

Le modèle de deep learning est composé de :

— Une couche d'embedding, qui permet de convertir les entiers des acides aminés en un vecteur continu. Par exemple, l'acide aminé 'A' est représenté par la valeur 1 et l'acide aminé 'C' est représenté par la valeur 2. Cela ne signifie pas (acide aminé 'C') = 2 x (acide aminé 'A'), ces valeurs sont indépendantes. Une façon de se débarrasser de ces dépendances consiste à utiliser l'Embedding.

— Un modèle auto-encodeur qui consiste en deux réseaux de neurones, un encodeur basé sur les réseaux de convolution 1D et un décodeur basé sur les réseaux de convolution 2D, car l'entrée du modèle sera des séquences d'acides aminées (1 dimension) et la sortie sera des matrices de contact (2 dimensions). Ainsi, l'architecture du réseau de neurones doit prendre en considération la transformation des données de sortie de l'encodeur 1D en données d'entrée du décodeur 2D.

4.1 Embedding

L'embedding désigne un ensemble de techniques de machine learning qui visent à représenter les mots ou les phrases d'un texte par des vecteurs de nombres réels, décrits dans un modèle vectoriel (ou Vector Space Model) (Figure 4.12). Ces nouvelles représentations de données textuelles ont permis d'améliorer les performances des méthodes de traitement automatique des langues (ou Natural Language Processing), comme le Topic Modeling ou le Sentiment Analysis.

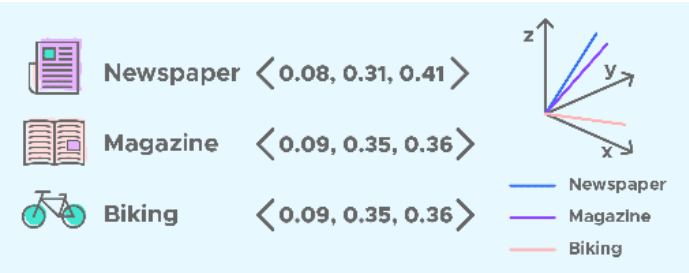


FIGURE 4.12 – L'embedding

Le word embedding repose sur la théorie linguistique fondée par Zellig Harris en 1956 et connue sous le nom de Distributional Semantics [69]. Cette théorie considère qu'un mot est caractérisé par son contexte, c'est-à-dire par les mots qui l'entourent. Ainsi, des mots qui partagent des contextes similaires partagent également des significations similaires.

Les algorithmes de word embedding sont le plus souvent employés pour décrire des mots à travers des vecteurs numériques, mais ils peuvent également être utilisés pour construire des représentations vectorielles de phrases entières, de données biologiques comme les séquences d'ADN, ou encore des réseaux représentés comme des graphes.

Il existe plusieurs approches de word embedding. Les premières remontent aux années 1960 et reposent sur des méthodes de réduction de dimensionnalité. Plus tard, de nouvelles techniques basées sur des modèles probabilistes et des réseaux de neurones ont permis d'obtenir de meilleures performances, comme Word2Vec, GloVe et la couche embedding qui est intégrée dans le réseau de neurones [70].

4.2 L'encodeur

L'encodeur est basé sur des couches de réseau de convolution 1D, des couches max pooling 1D et des couches dropout qui sont des couches typiques dans de nombreuses architectures de réseaux de neurones 1D. Puisque nos entrées sont en 1D, toutes les opérations sont en 1D dans l'encodeur.

Les résultats 1D obtenus seront convertis en matrices 2D selon l'opération expliquée ci-dessous.

4.3 La transformation de données de 1D en 2D

Si nous examinons en détail le processus de création de la matrice de contact, nous pouvons voir que ces matrices sont produites en associant toutes les paires d'acides aminés au sein d'une protéine. Cette association est réalisée à partir de la sortie 1D de l'encodeur dans l'architecture du réseau neuronal [54].

Cette opération est résumée par le schéma suivant (Figure 4.13) :

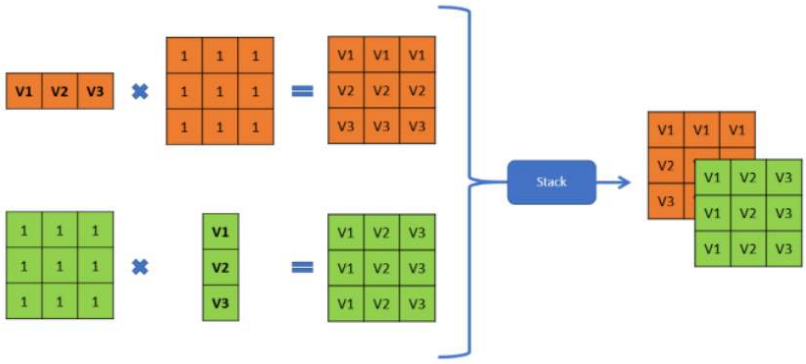


FIGURE 4.13 – La transformation des données en deux dimensions

4.4 Le décodeur

Après la conversion des données 1D en 2D, le décodeur inclut également des couches de convolutions 2D, des couches max pooling 2D et des couches dropout.

5 Résultats

Nous avons exécuté l'approche sur la plateforme kaggle [2], en utilisant le GPU. Le temps d'exécution d'une époque (epoch) est en moyenne de 12s (Figure 4.14).

```
fit_model2 = model2.fit(x_train, cmap_train, batch_size=100, epochs=500, verbose=1, shuffle=1, validation_data=(x_test,cmap_test))
```

```
Epoch 1/500
37/37 [==============================] - 15s 340ms/step - loss: 0.2724 - val_loss: 0.1695
Epoch 2/500
37/37 [==============================] - 12s 314ms/step - loss: 0.1351 - val_loss: 0.1673
```

FIGURE 4.14 – Exécution du modèle

2. https://www.kaggle.com/code/houriabraikia/contact-map-123

Puis nous avons généré deux modèles du modèle original après de nombreuses exécutions, pour étudier l'overfitting et l'underfitting d'un modèle (voir dessous). Dans le premier, nous avons minimisé le nombre de couches et d'époques et dans le deuxième, nous avons augmenté le nombre de couches et d'époques.

La fonction de perte Mean Square Error (MSE) a été utilisée pour l'évaluation des modèles, et l'optimiseur Adam pour son optimisation.

Rappelons que la fonction de perte représente une certaine mesure de la différence entre les valeurs observées des données et les valeurs calculées. C'est la fonction qui est minimisée dans la procédure d'ajustement d'un modèle par l'optimiseur. Elle quantifie à quel point un prédicteur donné est « bon » ou « mauvais », plus la perte est faible, meilleur est le travail du prédicteur pour modéliser la relation entre les données d'entrée et les cibles de sortie. Dans la plupart des projets de deep learning, la perte de données d'entraînement et de données de validation est généralement visualisée ensemble sur un graphique.

La perte d'entraînement est une métrique utilisée pour évaluer l'erreur du modèle sur l'ensemble d'entraînement. L'ensemble d'entraînement est une partie d'un ensemble de données utilisé pour former le modèle. La perte d'apprentissage est calculée en prenant la somme des erreurs pour chaque exemple de l'ensemble d'entraînement.

Alors que, la perte de validation est une métrique utilisée pour évaluer les performances d'un modèle sur l'ensemble de validation. L'ensemble de validation est une partie de l'ensemble de données mis de côté pour valider les performances du modèle. La perte de validation est similaire à la perte d'apprentissage et est calculée à partir d'une somme des erreurs pour chaque exemple dans l'ensemble de validation.

De plus, les pertes d'entraînement et de validation sont mesurées après chaque époque. Cela nous indique si le modèle a besoin d'autres réglages ou ajustements ou non. Ceci est généralement visualisé en traçant une courbe [71].

La figure 4.15 montre les fonctions de perte des trois modèles.

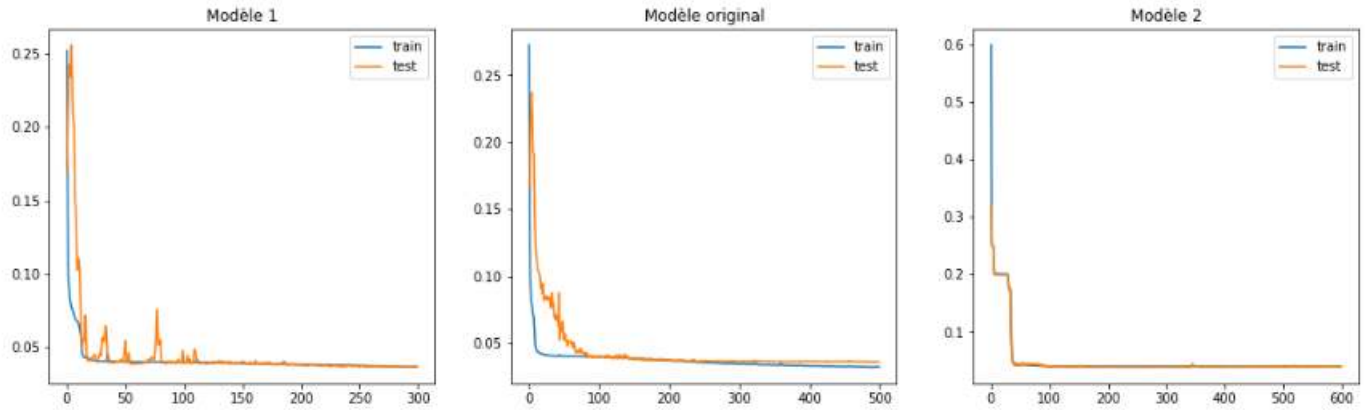


FIGURE 4.15 – Comparaison entre les fonctions de perte des trois modèles

Le but de la visualisation des courbes de perte est de diagnostiquer les performances du modèle et d'identifier s'il est en underfitting (sous-apprentissage), en overfitting (sur-apprentissage) ou en goodfitting (bon-apprentissage) (Figure 4.16).

Le sur-apprentissage représente un modèle qui a appris par cœur ses données d'entraînement, qui fonctionne donc bien sur le jeu d'entraînement mais pas de validation. Il effectue alors de

mauvaises prédictions sur de nouvelles, car elles ne sont pas exactement les mêmes que celles du jeu d'entraînement. Alors que, le sous-apprentissage représente un modèle qui n'arrive pas à déduire des informations du jeu de données. Il n'apprend donc pas assez et réalise de mauvaises prédictions sur le jeu d'entraînement. Il faut donc complexifier le réseau, car il ne taille pas bien par rapport aux types de données d'entrées. En effet, il n'arrive pas à capter la relation entre les données d'entrées et de sortie [72].

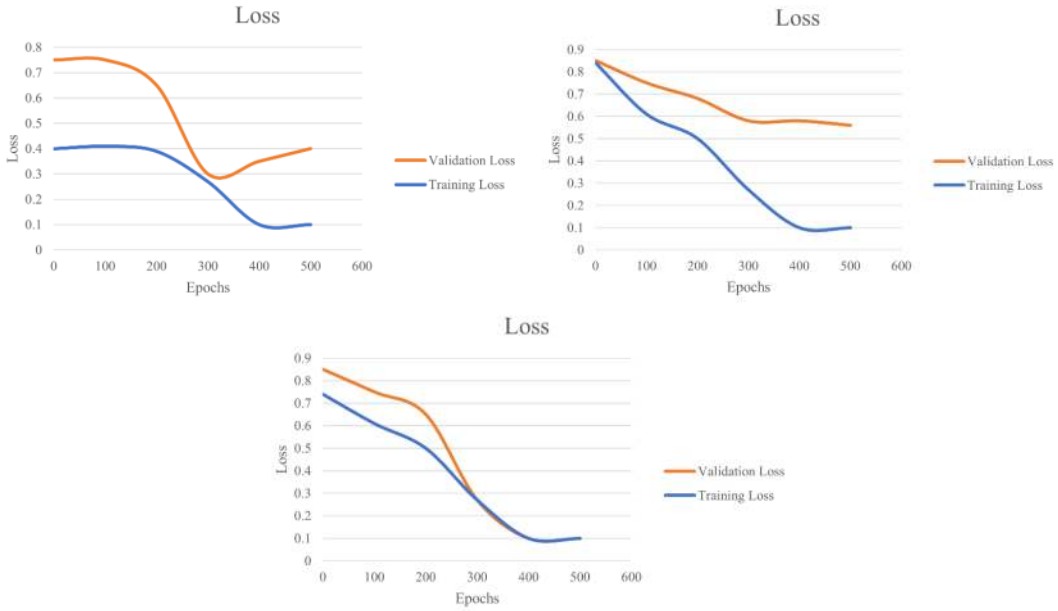


FIGURE 4.16 – Overfitting, Underfitting, Goodfitting

Les valeurs des fonctions de pertes des trois modèles sont organisées dans la figure 4.17 :

	Valeur de la fonction de perte de l'ensemble de données d'entraînement	Valeur de la fonction de perte de l'ensemble de données de test
Modèle original	0.0323	0.0359
Modèle 01	0.0368	0.0370
Modèle 02	0.0399	0.0398

FIGURE 4.17 – Les valeurs des fonctions de pertes des trois modèles

En comparant les résultats des trois modèles, on remarque que le modèle original a obtenu le taux d'erreur le plus bas. Alors, le modèle original contient le nombre de couches et d'époques idéal pour une prédiction précise de la matrice de contact.

La figure 4.18 montre les matrices de contact d'une séquence prédites à partir des trois modèles et sa matrice calculée à partir du dataset.

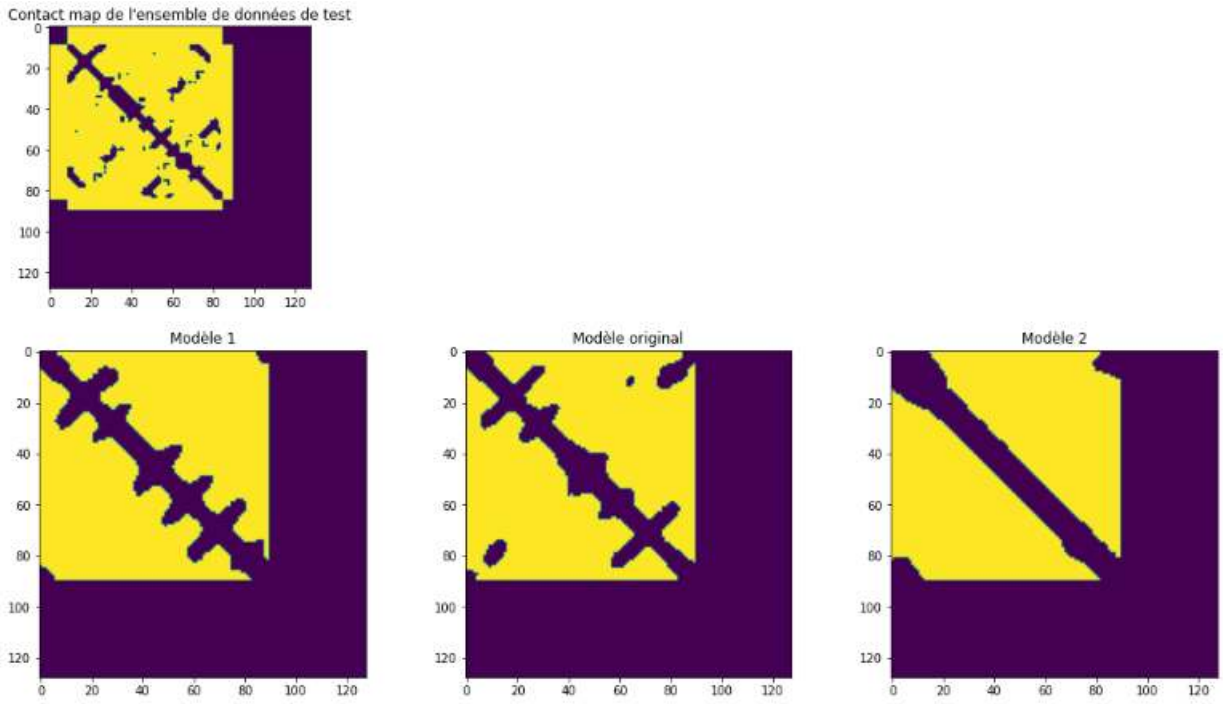


FIGURE 4.18 – Comparaison entre les résultats de prédiction des trois modèles

6 Conclusion

Dans ce chapitre nous avons présenté les outils d'implémentation utilisés pour la réalisation de cette approche. Ensuite, nous avons profondément expliqué l'approche utilisée qui sert à prédire la matrice de contact d'une séquence de protéine donnée. Aussi, nous avons présenté les modifications que nous avons apportées à cette méthode et enfin nous avons comparé les différents modèles.

CONCLUSION GÉNÉRALE

Les protéines sont les éléments constitutifs de la vie et sont responsables de la plupart des événements qui se produisent à l'intérieur des cellules. Le fonctionnement et l'action d'une protéine sont déterminés par sa forme tridimensionnelle. Pouvoir prédire cette forme à partir de sa seule séquence d'acides aminés serait une énorme avancée pour les sciences de la vie et la médecine. Cela faciliterait considérablement les travaux qui visent à mieux comprendre les éléments constitutifs des cellules et accélérerait la découverte de nouveaux médicaments.

Pendant des décennies, les expériences en laboratoire ont été le principal moyen de déterminer la structure des protéines telles que la résonance magnétique nucléaire [1] et la cristallographie aux rayons X [2]. Tandis que ces techniques sont devenues très coûteuses et trop lentes aux résultats souvent douteux, le recours vers les méthodes d'apprentissage automatique semble être une solution très prometteuse.

Plusieurs approches ont été proposées pour résoudre le problème de prédiction de la structure des protéines comme TripletRes [40] et RapstorX-Contact [4] qui prédisent les matrices de contact, AlphaFold1 [7] qui prédit les matrices de distance, et plus récemment, AlphaFold2 [3] et RossT-TAFold [30] ont été proposées pour prédire les coordonnées 3D des acides aminés directement à partir de la séquence primaire sans passer par les matrice de contact ou de distance, ils utilisent une architecture avancée de deep learning appelée les transformers.

Nous n'avons pas pu exécuter ces approches récentes en raison des tailles de bases de données utilisées et la complexité de ces approches. Donc, nous nous sommes orientés vers une approche qui fait la prédiction de la matrice de contact par l'apprentissage profond et plus précisément le modèle auto-encodeur et qui est basée sur l'approche RapstorX-Contact [4]. Cette approche est disponible sur GitHub [54] et utilise une version simplifiée du dataset ProteiNet [68].

Nous avons utilisé la plateforme kaggle qui offre un accès gratuit de 30 heures par semaine au GPU pour l'exécution, ainsi nous avons modifié l'approche pour étudier l'overfitting et l'underfitting d'un modèle en augmentant et en diminuant le nombre de couches et d'époques, puis la fonction de perte MSE a été utilisé pour évaluer les modèles et comparer entre eux. Les résultats obtenus ont montré que le modèle original contient un nombre de couches et d'époques idéal pour avoir un taux de perte minimal.

Comme perspective, cet algorithme peut être utilisé pour prédire les coordonnées 3D des acides aminés.

Les progrès annoncés aujourd'hui dans la prédiction de la structure des protéines nous donnent une nouvelle confiance dans le fait que l'intelligence artificielle deviendra l'un des outils les plus utiles de l'humanité pour repousser les frontières de la connaissance scientifique, et nous attendons avec impatience les nombreuses années de travail acharné et de découvertes à venir.

BIBLIOGRAPHIE

[1] Wikipédia. *s*pectroscopie rmn. https://fr.wikipedia.org/wiki/Spectroscopie_RMN. Consulté le : 2022-05-20.

[2] Wikipédia. *c*ristallographie aux rayons x. https://fr.wikipedia.org/wiki/Cristallographie_aux_rayons_X. Consulté le : 2022-05-20.

[3] Evans R. Pritzel A. et al. Jumper, J. Highly accurate protein structure prediction with alphafold. *Nature*, July 2021.

[4] Zhen Li Renyu Zhang Jinbo Xu Sheng Wang, Siqi Sun. Accurate de novo prediction of protein contact map by ultra-deep learning model. *PLoS Comput Biol*, January 2017.

[5] Jinbo Xu Sheng Wang, Siqi Sun. Analysis of deep learning methods for blind protein contact prediction in casp12. *Proteins. https://pubmed.ncbi.nlm.nih.gov/28845538/*, March 2018.

[6] Sheng Wang Jinbo Xu. Analysis of distance-based protein structure prediction by deep learning in casp13. *Proteins. https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25810*, August 2019.

[7] Evans R. Jumper J. et al. Senior, A.W. Improved protein structure prediction using potentials from deep learning. *Nature*, January 2020.

[8] Dr. SAD HOUARI Nawal. *Polycopié Bioinformatique et modélisation.* Université des Sciences et de la Technologie Mohamed Boudiaf d'Oran, Faculté des sciences de la nature et de la vie, Département du vivant et de lenvironnement, 2018 - 2019.

[9] David Louapre. *Le repliement des protéines : Résolu par l'intelligence artificielle AlphaFold ?* https://www.youtube.com/watch?v=0GewxRMME8o, 2020.

[10] Benseni Noureddine. Adaptation et implémentation de la méthode des réseaux de neurone pour la prédiction des structures secondaires de protéine. Master's thesis, Université Mohamed Boudiaf - M'sila, M'sila, 2016.

[11] Ahmed Majam. Prediction of protein structure classes using support vector machine (svm) classifier. Master's thesis, Université Mohamed Boudiaf - M'sila, M'sila, 2016.

[12] Mohamed Sayed Hassan Damien Imbs. *Bioinformatique, Travail d'étude*. Université de Nice Sophia Antipolis, 2016.

[13] Member IEEE Jianlin Cheng, Allison N. Tegge and IEEE Pierre Baldi, Senior Member. Machine learning methods for protein structure prediction. *Methodological Review*, 2008.

[14] Demis Hassabis Pushmeet Kohli Andrew Senior, John Jumper. *a*lphafold : Using ai for scientific discovery january 15, 2020. `https://www.deepmind.com/blog/alphafold-using-ai-for-scientific-discovery-2020`. Consulté le : 2022-05-19.

[15] TrEMBL. *u*niprotkb/trembl protein database release 2022/01 statistics. `https://www.ebi.ac.uk/uniprot/TrEMBLstats`. Consulté le : 2022-05-19.

[16] EMBL-EBI DeepMind. *a*bout alphafold protein structure database. `https://alphafold.ebi.ac.uk/about`. Consulté le : 2022-05-19.

[17] Wikipédia. *p*rotein data bank. `https://fr.wikipedia.org/wiki/Protein_Data_Bank`. Consulté le : 2022-05-19.

[18] UniRef90. *t*he uniprot reference clusters (uniref). `https://ftp.uniprot.org/pub/databases/uniprot/uniref/uniref90/`. Consulté le : 2022-05-20.

[19] EMBL-EBI DeepMind. *a*lphafold protein structure database. `https://alphafold.ebi.ac.uk`. Consulté le : 2022-05-19.

[20] DARWIN O.V. ALONSO ROGER ARMEN and VALERIE DAGGETT. The role of alpha-, 3(10)-, and pi-helix in helix–>coil transitions. *Department of Medicinal Chemistry, University of Washington, Seattle, Washington 98195, USA*, mars 2003.

[21] Guillaume Chakroun. *Prédiction de la structure d'une protéine*. Soluscience, 2004.

[22] Wikipédia. *c*ryo-microscopie électronique. `https://fr.wikipedia.org/wiki/Cryo-microscopie_électronique`. Consulté le : 2022-05-20.

[23] Jiarui Chen and Shirley W. I. Siu. Machine learning approaches for quality assessment of protein structures. *Department of Computer and Information Science, Faculty of Science and Technology, University of Macau,Macau, China*, April 2020.

[24] Demis Hassabis Andrew Senior, John Jumper. *a*lphafold : Using ai for scientific discovery december 2, 2018. `https://www.deepmind.com/blog/alphafold-using-ai-for-scientific-discovery`. Consulté le : 2022-05-19.

[25] Philip Bradley Brian Kuhlman. Advances in protein structure prediction and design. *Nature. https: // www. nature. com/ articles/ s41580-019-0163-x. pdf* , NOVEMBER 2019.

[26] Burkhard Rost. Protein structure prediction in 1d, 2d, and 3d. *European Molecular Biology Laboratory, Heidelberg, Germany*, 1998.

[27] Andras Fiser. Template-based protein structure modeling. *Methods Mol Biol. https: // www. ncbi. nlm. nih. gov/ pmc/ articles/ PMC4108304* , Jul 2014.

[28] Karim Mezhoud. *Prédiction des structure 3D des protéines PhD. Toxicologie, Protéomique, Bioinformatique*. PhD thesis, Centre national des Sciences et Technologies Nucléaires Sidi Thabet, Tunis, 2016.

[29] hiye Guo ie Hou anlin Cheng Jian Liu, ianqi Wu. Improving protein tertiary structure prediction by deeplearning and distance prediction in casp14. *Proteins. https ://doi.org/10.1002/prot.26186*, July 2021.

[30] Anishchenko I Dauparas J Ovchinnikov S Lee GR Wang J Cong Q Kinch LN Schaeffer RD Millán C Park H Adams C Glassman CR DeGiovanni A Pereira JH Rodrigues AV van Dijk AA Ebrecht AC Opperman DJ Sagmeister T Buhlheller C Pavkov-Keller T Rathinaswamy MK Dalwadi U Yip CK Burke JE Garcia KC Grishin NV Adams PD Read RJ Baker D. Baek M, DiMaio F. Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, Aug 2021.

[31] Cao R et al. dhikari B, Bhattacharya D. Confold : residue-residuecontact-guided ab initio protein folding. *Proteins. https://pubmed.ncbi.nlm.nih.gov/25974172/*, 2015.

[32] Zhang C et al. heng W, Li Y. Deep-learning contact-map guided pro-tein structure prediction in casp13. *Proteins. https://pubmed.ncbi.nlm.nih.gov/31365149/*, 2019.

[33] Cao R et al Hou J, Wu T. Protein tertiary structure modeling drivenby deep learning and contact distance prediction in casp13. *Proteins. https://onlinelibrary.wiley.com/doi/full/10.1002/prot.25697*, 2019.

[34] J.W. ; Choi S. ; Lee Y. Suh, D. ; Lee. Recent applications of deep learning methods on evo-lutionand contact-based protein structure prediction. *Internationnal Journal of Molecular Science*, June 2021.

[35] Member IEEE Jianlin Cheng, Allison N. Tegge and IEEE Pierre Baldi, Senior Member. Machine learning methods for protein structure prediction. *Methodological Review*, 2008.

[36] Hongzhi Yin Ngai-Man Cheung Leila Khalatbari M.R.Kangavari Saeid Hosseini. A multi-component learning machine to predict protein secondary structure. *Science Direct*, July 2019.

[37] Shusen Zhou, Hailin Zou, Chanjuan Liu, Mujun Zang, and Tong Liu. Combining deep neural networks for protein secondary structure prediction. *IEEE Access*, 2020.

[38] Mai S. Mabrouk3 Ahmed Y. Sayed Heba M. Afify, Mohamed B. Abdelhalim. Protein se-condary structure prediction (pssp) using different machine algorithms. *Egyptian Journal of Medical Human Genetics*, June 2021.

[39] B. ;Adhikari B. ; KC D.B. Pakhrin, S.C. ; Shrestha. Deep learning-based advances in protein structure prediction. *Internationnal Journal of Molecular Science*, May 2021.

[40] Eric W. Bell Wei Zheng Xiaogen Zhou Dong-Jun Yu Yang Zhang Yang Li, Chengxin Zhang. Deducing high-accuracy protein contact-maps from a triplet of coevolutionary matrices through deep residual convolutional networks. *PLoS Comput Biol*, March 2021.

[41] US National Institute of General Medical Sciences (NIH/NIGMS). *1*3th community wide experiment on the critical assessment of techniques for protein structure prediction. `https://predictioncenter.org/casp13/zscores_rrc.cgi`. Consulté le : 2022-05-20.

[42] Niki Parmar Jakob Uszkoreit Llion Jones Aidan N. Gomez Lukasz Kaiser Illia Polosukhin Ashish Vaswani, Noam Shazeer. Attention is all you need. *arxiv. https://arxiv.org/abs/1706.03762*, 2017.

[43] The AlphaFold team. *a*lphafold : a solution to a 50-year-old grand chal-lenge in biology november 30, 2020. `https://www.deepmind.com/blog/alphafold-a-solution-to-a-50-year-old-grand-challenge-in-biology`. Consulté le : 2022-05-20.

[44] Alexander Pritzel Natasha Antropova Andrew Senior Tim Green Augustin Žídek Russ Bates Sam Blackwell Jason Yim Olaf Ronneberger Sebastian Bodenstein Michal Zielinski Alex Bridgland Anna Potapenko Andrew Cowie Kathryn Tunyasuvunakool Rishub Jain Ellen Clancy Pushmeet Kohli John Jumper Demis Hassabis Richard Evans, Michael O'Neill. Protein complex prediction with alphafold-multimer. *bioRxiv*, March 2022.

[45] Wikipédia. *a*marrage (moléculaire). `https://fr.wikipedia.org/wiki/Amarrage_(moléculaire)`. Consulté le : 2022-05-23.

[46] Hahnbeom Park Ian R. Humphreys David Baker Minkyung Baek, Ivan Anishchenko. Protein oligomer modeling guided by predicted interchain contacts in casp14. *Proteins*, July 2021.

[47] Česlovas Venclovas Justas Dapkūnas, Kliment Olechnovič. Modeling of protein complexes in casp14 with emphasis on the interaction interface prediction. *Proteins*, June 2021.

[48] CASP. *ABSTRACT BOOK CRITICAL ASSESSMENT OF TECHNIQUES FOR PROTEIN STRUCTURE PREDICTION*. May-September 2020.

[49] Gabriele Pozzati Patrick Bryant and Arne Elofsson. Improved prediction of protein-protein interactions using alphafold2 and extended multiple-sequence alignments. *bioRxiv*, 2021.

[50] Ilya A Vakser. Evaluation of gramm low-resolution docking methodology on the hemagglutininantibody complex. *Proteins*, 1997.

[51] Akhil Jindal Omeir Khan George Jones Sergey Kotelnikov Dzmitry Padhorny Sandor Vajda Usman Ghani, Israel Desta and Dima Kozakov. Improved docking of protein models by a combination of alphafold2 and cluspro. *bioRxiv*, 2021.

[52] Minkyung Baek Aditya Krishnakumar Ivan Anishchenko Sergey Ovchinnikov Jing Zhang Travis J. Ness Sudeep Banjade Saket Bagde Viktoriya G. Stancheva Xiao-Han Li Kaixian Liu Zhi Zheng Daniel J. Barrero Upasana Roy Israel S. Fernández Barnabas Szakal Dana Branzei Eric C. Greene Sue Biggins Scott Keeney Elizabeth A. Miller J. Christopher Fromme Tamara L. Hendrickson Qian Cong Ian R. Humphreys, Jimin Pei and David Baker. Structures of core eukaryotic protein complexes. *bioRxiv*, 2021.

[53] Quan Le Mirko Torrisi, Gianluca Pollastri. Deep learning methods in protein structure prediction. *Computational and Structural Biotechnology Journal. https://www.researchgate.net/publication/338741992_Deep_Learning_methods_in_Protein_Structure_Prediction*, 2020.

[54] N. Lakmal Deshapriya. *p*rotein structure modeling with deep learning (tensorflow-keras). `https://github.com/lakmalnd/deep-protein-structure-modeling`. Consulté le : 2022-05-21.

[55] Hacene BELLAHMER. Implémentation et évaluation d'un modèle d'apprentissage automatique pour l'estimation de la valeur marchande de propriétés immobilières. Master's thesis, UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU, TIZI-OUZOU, 2020.

[56] Jeremie Sublime. L'apprentissage non-supervisé et ses contradictions. *Bulletin de la Société In-formatique de France, Société Informatique de France,*, 2022.

[57] MADOUI SOUMIA. L'utilisation du deep learning pour l'extraction du contenu des pages web. Master's thesis, Université Mohamed Khider – BISKRA, BISKRA, 2019.

[58] Le DAP Comité Éditorial. *a*uto-encodeur. `https://dataanalyticspost.com/Lexique/auto-encodeur/`. Consulté le : 2022-05-21.

[59] Gary B. *c*onvolutional neural network. https://datascientest.com/convolutional-neural-network. Consulté le : 2022-05-21.

[60] Shiva Verma. *u*nderstanding 1d and 3d convolution neural network | keras. https://towardsdatascience.com/understanding-1d-and-3d-convolution-neural-network-keras-9d8f76e29610. Consulté le : 2022-05-21.

[61] wikipedia. *p*ython. https://fr.wikipedia.org/wiki/Python_(langage). Consulté le : 2022-05-21.

[62] Data Transition Numérique. *m*aîtrisez l'analyse des données avec numpy python. https://www.data-transitionnumerique.com/numpy-python/. Consulté le : 2022-05-21.

[63] wikipedia. *m*atplotlib. https://fr.wikipedia.org/wiki/Matplotlib. Consulté le : 2022-05-21.

[64] Florian Fasmeyer. *m*atplotlib. https://he-arc.github.io/livre-python/matplotlib/index.html. Consulté le : 2022-05-21.

[65] Bastien L. *t*ensorflow. https://datascientest.com/tensorflow. Consulté le : 2022-05-21.

[66] journaldunet. *t*ensorflow. https ://www.journaldunet.fr/web-tech/guide-de-l-intelligence-artificielle/1501861-tensorflow-tout-savoir-sur-la-plateforme-de-deep-learning-de-google/. Consulté le : 2022-05-21.

[67] DataScientest. *k*aggle. https://datascientest.com/kaggle-tout-ce-quil-a-savoir-sur-cette-plateforme. Consulté le : 2022-05-21.

[68] Mohammed AlQuraishi. *p*roteinnet. https://github.com/aqlaboratory/proteinnet. Consulté le : 2022-05-21.

[69] Zellig S. Harris. Distributional structure. *https ://doi.org/10.1080/00437956.1954.11659520. WORD*, 1954.

[70] Le DAP Comité Éditorial. *e*mbedding. https://dataanalyticspost.com/Lexique/word-embedding/. Consulté le : 2022-05-21.

[71] Eugen (Baeldung). *t*raining and validation loss in deep learning. https://www.baeldung.com/cs/training-validation-loss-deep-learning. Consulté le : 2022-05-24.

[72] Bastien Maurice. *c*omprendre overfitting et underfitting. https://deeplylearning.fr/cours-theoriques-deep-learning/comprendre-overfitting-et-underfitting/. Consulté le : 2022-05-24.