UNIVERSITÉ DU QUÉBEC À MONTRÉAL

MÉTHODE DE MODÉLISATION ET DESIGN D'UNE PROTÉINE POUR LA CAPTURE DE SF4

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

ADEKOUDJO ADE-DAYO NASSIR

JUILLET 2025

REMERCIEMENTS

Je tiens premièrement à remercier Dieu pour m'avoir permis de commencer et de mener ce travail à terme.

Je tiens, ensuite à remercier et à exprimer toute ma gratitude à mon directeur de mémoire Monsieur Reinharz, Vladimir, Professeur à l'Université du Québec à Montréal, pour sa confiance, sa patience, sa disponibilité et par-dessus tout pour ses précieux conseils.

Je remercie aussi le laboratoire LACIM pour sa bourse qu'il m'a octroyée au cours de ma maitrise.

Merci à tous les professeurs de l'UQAM avec qui j'ai suivi des cours durant tout mon parcours.

Je tiens à exprimer ma profonde gratitude envers ma famille pour l'amour, le soutien inébranlable et la compréhension qu'ils m'ont témoignés tout au long de ce parcours académique.

À toutes celles et ceux qui, de près ou de loin, ont contribué à la réalisation de ce mémoire, je leur adresse mes sincères remerciements : leur appui et leurs encouragements ont été une source précieuse de motivation dans l'aboutissement de ce travail.

À tous ceux qui se sentent fiers de ce travail.

TABLE DES MATIÈRES

TABLE DES FIGURES

LISTE DES TABLEAUX

ACRONYMES

SF4 Cluster Fer-Soufre (Fe-S) de type [4Fe-4S].

ADN Acide Désoxyribonucléique.

ARN Acide ribonucléique.

RFdiffusion RosettaFoldDiffusion.

GPU Graphic Processing Unit.

CPU Central Processing Unit.

SUF (mobilization of sulfur.

NIF nitrogen fixation.

ISC iron-sulfur cluster.

BLAST Basic Local Alignment Search Tool.

PDB Protein Data Bank.

RCSB Research Collaboratory for Structural Bioinformatics.

MSA Multiple Sequence Alignment.

NPNN Acide ribonucléique.

CCIS1 Coiled-Coil Iron-Sulfur 1.

EPR résonance paramagnétique électronique.

UV-vis ultraviolet-visible.

CASP Critical Assessment of Structure Prediction.

PAE Acide ribonucléique.

RMSD Root Mean Square Deviation.

MEME Multiple Em for Motif Elicitation.

EM Expectation-Maximization.

CD-HIT Cluster Database at High Identity with Tolerance.

SDH succinate déshydrogénase.

QFR fumarate oxydoréductase.

PYMOL Python Molecular Graphics System.

PAM Point Accepted Mutation.

BLOSUM Blocks Substitution Matrix.

PLDDT Predicted Local Distance Difference Test.

PTM Predicted TM-score.

IDDT Interface Difference of Distances.

UPGMA Unweighted Pair Group Method with Arithmetic Mean.

Nombres

α alpha.

β beta.

Unités

Å angström.

RÉSUMÉ

Les protéines métalliques, et en particulier celles contenant des clusters fer-soufre ([Fe-S]), jouent un rôle central dans de nombreux processus biologiques essentiels tels que la respiration cellulaire, le transfert d'électrons ou encore la régulation génétique. Parmi elles, les clusters [4Fe-4S] (ou SF4) se distinguent par leur complexité structurale et leur sensibilité à l'oxygène, rendant leur intégration dans les protéines artificielles particulièrement difficile. Ce mémoire s'inscrit dans cette problématique et explore la possibilité de concevoir, par des approches computationnelles, une protéine artificielle capable d'intégrer un cluster [4Fe-4S] via les mécanismes cellulaires naturels. En mobilisant des outils récents de modélisation moléculaire, tels que AlphaFold et RFdiffusion, cette étude vise à identifier un signal structurel, potentiellement un motif séquentiel conservé responsable de l'insertion naturelle du cluster dans les protéines connues. Ce signal est ensuite greffé à une protéine d'intérêt, dans le but de concevoir une entité hybride capable de détourner les mécanismes d'assemblage intracellulaire. Après une phase de traitement et d'analyse des structures disponibles, des variants ont été générés et validés in silico afin d'évaluer leur compatibilité structurale et fonctionnelle avec le cluster SF4. Ce travail ouvre ainsi la voie à une nouvelle approche pour la conception rationnelle de protéines hybrides capables d'accueillir des cofacteurs métalliques complexes, avec des implications potentielles en biotechnologie, en bioénergie et en médecine.

Le code source de l'implémentation est disponible sur le dépôt suivant : https://gitlab.info.uqam.ca/adekoudjo.ade-dayo_nassir/sf4ligand.

Mots clés : Clusters fer-soufre (Fe-S), [4Fe-4S] / SF4, Conception de protéines, intelligence artificielle, AlphaFold, RFdiffusion, Bio-ingénierie, Protéines hybrides, Biologie computationnelle.

INTRODUCTION

Les protéines métalliques occupent une place importante dans de nombreux processus biologiques cruciaux tels que la photosynthèse, la respiration, l'oxydation de l'eau, la réduction de l'oxygène moléculaire et la fixation de l'azote (Johnson *et al.*, 2005). Parmi ces protéines, celles incorporant des clusters fer-soufre ([Fe-S]) sont omniprésents dans tous les organismes vivants et se distinguent par leur rôle indispensable dans des mécanismes complexes, notamment en catalysant des réactions redox et en stabilisant des états d'oxydation variables (Rees, 2002). Les clusters fer-souffre sont composés d'un à huit atomes de fer (Fe) reliés par des ligands de souffre (S). Les clusters [4Fe-4S] ou encore SF4 sont composés de 4 atomes de fer reliés par des ligands de souffre. Grâce à leur flexibilité structurelle exceptionnelle et à leurs propriétés chimiques et électroniques variées, les clusters [Fe-S] jouent un rôle clé dans plusieurs processus tels que le transfert d'électrons, la fixation et l'activation de substrats, le stockage du fer et du soufre, la régulation de l'expression des gènes, ainsi que dans l'activité des enzymes (Johnson *et al.*, 2005).

Ces dernières années, la conception de protéines contenant des clusters [Fe-S] a suscité un intérêt croissant, tant pour élucider leurs mécanismes naturels que pour concevoir de nouvelles protéines dotées de propriétés spécifiques. Cependant, la création de protéines artificielles intégrant des clusters [4Fe-4S] reste un défi de taille, principalement en raison de la complexité de leur structure et de leur sensibilité aux perturbations environnementales, le cluster [4Fe-4S] étant très réactif avec l'oxygène, les expériences doivent se faire en milieu anaérobiques (Boyd *et al.*, 2014). La conception informatique d'une protéine qui interagit avec le cluster [4Fe-4S] (Grzyb *et al.*, 2010), permet d'explorer et de prédire des interactions au niveau atomique et offre une voie prometteuse pour surmonter ces difficultés.

Avec aujourd'hui des outils fonctionnels de modélisation moléculaire et de biologie computationnelle comme AlphaFold (Jumper *et al.*, 2021b) et RFdiffusion (Watson *et al.*, 2023) qui permettent de faire de bonnes prédictions et conceptions pour plusieurs protéines notamment des monomères, des assemblages symétriques complexes et la fixation de métaux pour ne citer que ceux là. Ces outils nous permettrons alors de faire des variants de la protéine interagissant avec le clus-

ter [4Fe-4S] (Grzyb *et al.*, 2010). Ces variants pourront donc être utilisés comme point de départ avec ces outils pour faire une première validation informatique, ce qui nous permettra d'être plus confiant dans les résultats escomptés.

Étant donné la complexité autour du processus d'insertion du cluster [4Fe-4S] dans une cellule de protéine, nous ne savons pas exactement comment le processus se fait naturellement (Lill, 2009). Par hypothèse nous supposons qu'il existe un signal particulier dans la structure des protéines contenants le cluster [4Fe-4S] reconnu par un mécanisme cellulaire qui insère le cluster dans la protéine.
L'objectif de ce travail est de détourner un signal indiquant à la cellule de capturer et insérer le cluster [4Fe-4S]. Nous disposons d'un ensemble de protéines qui intègrent ce mécanisme, où la cellule interagit directement avec le cluster [4Fe-4S]. Nous savons que la protéine d'intérêt peut également interagir avec ce cluster. La question que nous tentons de résoudre ici est la suivante : peut-on d'abord identifier le signal (c'est-à-dire un motif de structure de séquence) dans les séquences protéiques, puis l'intégrer à notre protéine d'intérêt, de manière à ce que la cellule puisse accomplir ce processus complexe, celui de le lier au cluster [4Fe-4S] ? Dans ce travail, nous chercherons donc à identifier le signal commun aux protéines qui contiennent le cluster [4Fe-4S], puis à concevoir une protéine capable d'incorporer ce signal.

Ce mémoire vise à explorer l'utilisation de méthodes informatiques pour la conception d'une protéine qui détourne le mécanisme de la cellule afin d'intégrer un cluster [4Fe-4S].

Cette étude pourrait jeter les bases d'avancées futures dans le domaine des protéines synthétiques et leur application en biotechnologie, en bioénergie ou en médecine.

0.1 Les protéines

Les protéines sont des biomolécules essentielles présentes dans toutes les formes de vie, jouant des rôles diversifiés dans l'organisme. Elles participent à des processus tels que les réactions enzymatiques, la signalisation cellulaire, le transport moléculaire et la régulation des tissus et des organes. Une protéine est formée d'une ou plusieurs chaînes d'acides aminés, appelées polypep-

tides, qui sont reliées par des liaisons peptidiques. La formation des peptides et des protéines résulte de la liaison d'acides aminés entre eux par une réaction de condensation, générant des liaisons peptidiques(Buxbaum *et al.*, 2007, chapitre 2).

L'étude des structures protéiques est organisée en quatre niveaux hiérarchiques : les structures primaire, secondaire, tertiaire et quaternaire. Chaque niveau de complexité est associé à des défis spécifiques, notamment en termes de prédiction, de modélisation et d'analyse. Cette section explore ces différents niveaux.

Structure primaire : La structure primaire d'une protéine correspond à la séquence linéaire des acides aminés qui la compose (Buxbaum *et al.*, 2007, chapitre 2). Cette séquence dicte la structure tridimensionnelle et la fonction biologique de la protéine. Les acides aminés dans une chaîne polypeptidique sont liés par des liaisons peptidiques entre le groupe carboxyle d'un résidu et le groupe amine du suivant. Elle peut également être représentée comme une séquence sur un alphabet de plusieurs caractères.

Structure secondaire : La structure secondaire d'une protéine désigne l'ensemble des motifs de repliement réguliers et répétitifs adoptés par la chaîne polypeptidique. Cette organisation est maintenue grâce à des liaisons hydrogène formées entre les groupes amine et groupes cétone des liaisons peptidiques, qui présentent respectivement des charges partielles positive et négative (Buxbaum *et al.*, 2007, chapitre 2). Deux motifs principaux sont largement répandus : l'hélice α, une structure enroulée où les liaisons hydrogène se forment entre un résidu et celui situé quatre positions plus loin, et le feuillet β, composé de brins alignés stabilisés par des liaisons hydrogène interchaînes, pouvant être parallèles ou antiparallèles. Toutefois le feuillet beta et les brins peuvent être éloignés dans la séquence. Chaque résidu de la protéine appartient soit à une hélice α, brin β ou aucun. La protéine dans la Figure 0.1, a quatre hélice α jointes par des régions non-structurées.

Structure tertiaire : La structure tertiaire d'une protéine correspond à son repliement tridimensionnel, stabilisé par des interactions entre les chaînes latérales des acides aminés, comme les interactions hydrophobes, les liaisons hydrogène, les ponts salins et les ponts disulfure, assurant sa forme et sa fonction (Buxbaum *et al.*, 2007, chapitre 2). La structure tertiaire fait référence à la

position 3D dans l'espace de tous les atomes de la protéine.

Structure quaternaire : La structure quaternaire d'une protéine correspond à l'association de plusieurs chaînes polypeptidiques qui forment un complexe fonctionnel, stabilisé par des interactions ioniques et hydrophobes. Selon le nombre et la nature des sous-unités, on parle par exemple de dimères ou d'hétérodimères. Cette organisation est essentielle au bon fonctionnement de nombreuses protéines, comme l'hémoglobine, composée de sous-unités répétées appelées protomères (Buxbaum *et al.*, 2007, chapitre 2).

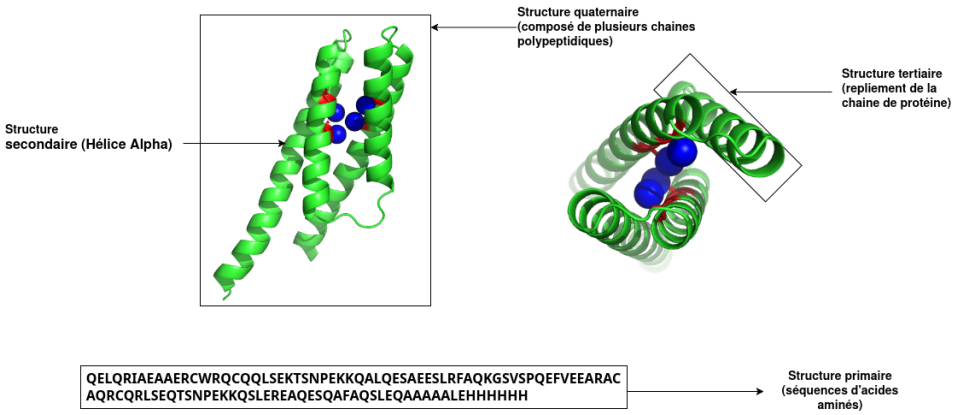

Figure 0.1 Figure montrant une structure de protéine en vert avec le cube SF4 en bleu et les acides aminés de cystéine en rouge. Séquence et structure prédite par AlphaFold de CCIS1 (Coiled-Coil Iron-Sulfur 1) vues de dessus et de côté. Les résidus sont tous en hélices.

0.2 Protéine d'intérêt : Coiled-Coil Iron Sulfur 1 (CCIS1)

De base il convient de notifier qu'il n'existe aucune expérience qui détermine la structure de CCIS1 (Coiled-Coil Iron-Sulfur 1). Néanmoins il existe un modèle (Grzyb *et al.*, 2010) qui nous sert de point de départ dans nos prédictions. La **protéine CCIS1 (Coiled-Coil Iron-Sulfur 1)** (Grzyb *et al.*, 2010) est une **protéine artificielle de novo** comme illustrée à la Figure 0.1 ayant été conçue pour **lier un cluster SF4** au sein d'un **faisceau stable de quatre hélices** α. Sa structure repose sur un noyau hydrophobe où des cystéines stratégiquement placées assurent une coordination précise du fer, la distinguant des protéines FeS naturelles. Inspirée du motif CXXC de la **tryptophanyl-ARNt syn-**

thétase de Thermotoga Maritima (PDB ID 2G36) (Han *et al.*, 2010), CCIS1 a été optimisée grâce à une approche computationnelle combinée à des méthodes expérimentales, notamment avec ProtCAD (Xu et Dunbrack Jr, 2023) pour la conception de sa configuration symétrique et ROSETTA (Simons *et al.*, 1997) pour l'ajustement de sa séquence et de ses boucles structurales. Sa fonctionnalité a été validée par des analyses spectroscopiques telles que le dichroïsme circulaire, l'UV-Vis et l'EPR, confirmant un assemblage correct et stable du cluster SF4. De plus, son architecture unique permet une extension modulaire vers des protéines multi-clusters par duplication de son site de liaison, ouvrant des perspectives pour la conception de nouvelles métalloprotéines.
La séquence de la protéine CCIS1 a 97 acides aminés et est définie ci-dessous :

```
QELQRIAEAWERCWRQCQQLSEKTSNPEKKHALQEEADESLRFAQKGSVSP
QEFVEDARCAQCRQLSEQTSNPEKKQSLEQANEESQNFQAWLEQAA
```

Avec cette séquence nous reprenons plusieurs tests de prédiction de la structure avec AlphaFold et les résultats obtenus correspondent au modèle contenu dans l'article (Grzyb *et al.*, 2010). Ces résultats viennent donc confirmer le modèle.

0.3 Algorithmes sur les protéines

Le but est de faire des prédictions en vue de comprendre différentes structures pour différents niveaux d'organisation. Chaque niveau comporte différentes questions relatives aux protéines qui font l'objet de nos préoccupations dans le cadre de notre projet. Dans les sections qui suivent nous allons donc explorer différentes méthodes pour tenter d'y répondre.

0.3.1 La structure primaire

Dans le cadre de notre travail, il s'agira principalement d'explorer deux défis algorithmiques, au niveau de la séquence, que sont l'alignement multiple et l'identification de motifs.
Une méthode populaire pour l'alignement multiple des séquences est l'algorithme d'alignement progressif reposant sur la construction successive d'alignements par paires. Un outil reposant sur cette stratégie est Clustal W (Chenna *et al.*, 2003) qui se base sur un algorithme de programma-

tion dynamique notamment celui de Needleman-Wunsch (Needleman et Wunsch, 1970), sur un algorithme de clustering par voisinage notamment **le neighbor-joining** (Saitou et Nei, 1987) ou **UPGMA (Unweighted Pair Group Method with Arithmetic Mean)** (Sokal et Michener, 1958) et sur l'alignement progressive des séquences par ordre décroissant de similarité.
Un autre défi en lien avec la structure primaire concerne l'identification des motifs au sein d'un ensemble de séquences. Pour ce faire l'approche utilisée est l'algorithme de **espérance-maximisation (Expectation-Maximization, EM)** (Dempster *et al.*, 1977), utilisé pour estimer les paramètres d'un modèle probabiliste lorsque certaines variables ne sont pas directement observables. Cet algorithme est utilisé par l'outil **MEME (Multiple Expectation Maximization for Motif Elicitation)** (Bailey *et al.*, 1994) et implémenté dans MemeSuite (Bailey *et al.*, 2015). Nous l'utilisons dans ce travail.
Un autre outil que nous utilisons au niveau de la structure primaire est **cd-hit (Cluster Database at High Identity with Tolerance)** (Li *et al.*, 2001) que nous utilisons pour faire des clusters avec différents seuils d'identité.

0.3.2 Les structures secondaire, tertiaire et quaternaire

Nous souhaitons, dans le cadre de ce projet, nous attarder sur la prédiction de la structure d'une séquence dans le but de vérifier la viabilité de notre modèle. Pour ce faire, il s'agira de valider et de discuter les résultats obtenus par aphafold basé sur les réseaux de neuronnes profonds. Ces dernières années, il a été constaté qu'Alphafold basé sur les réseaux neuronaux profonds donnait de bons résultats dans des cas particuliers (Kryshtafovych *et al.*, 2021). Nous en discutons dans la section 1.1.

0.3.3 La conception des structures de protéine avec la diffusion

Ici nous voulons concevoir un protéine avec la structure du CCIS1 (Coiled-Coil Iron Sulfur1) auquel nous associons le motif identifié dans les protéines contenant un cluster SF4. La diffusion est un phénomène physique qui consiste à propager des particules d'un milieu dense vers un milieu moins dense. Ce phénomène a inspiré des algorithmes d'intelligence artificielle pour concevoir des

modèles comme RFdiffusion (Watson *et al.*, 2023) que nous utilisons pour ce travail. RFdiffusion est basé sur un processus de diffusion et de diffusion inverse pour concevoir des protéines.

Le chapitre suivant traitera de ce qui a déjà été fait en ce qui concerne le repliement de protéines notamment de l'état de l'art de la prédiction des protéines, de la conception des protéines et de la biogenèse des protéines SF4.

Le terme SF4 sera utilisé pour désigner les clusters [4Fe-4S] pour la suite de ce travail.

CHAPITRE 1

PRÉDICTION, CONCEPTION ET BIOGENÈSE DES PROTÉINES SF4

Ce chapitre vise à donner plus d'informations sur les outils utilisés en ce qui concerne la prédiction et la conception des protéines et exposer des connaissances en ce qui concerne la biogenèse des protéines.

1.1 Prédiction de protéines

1.1.1 État de l'art

Le défi du repliement des protéines était formulé depuis 1965, soulevant des interrogations essentielles sur le rôle des séquences d'acides aminés dans la détermination des structures tridimensionnelles des protéines, sur la rapidité de ce processus et sur la possibilité de prédire les structures protéiques via des algorithmes à partir des séquences génétiques.

En 1965, Anthony V. Guzzo (Guzzo, 1965) expliquent des corrélations entre les séquences d'acides aminés et les structures secondaires dans certaines protéines. Ils ont identifié la proline, l'acide aspartique, l'acide glutamique et l'histidine comme essentiels pour provoquer des interruptions dans les structures hélicoïdales, bien que les trois derniers ne suffisent pas à eux seuls. Ils ont aussi ajouté qu'une chaîne d'acides aminés d'au moins six résidus, sans aucun de ces acides aminés, est probablement hélicoïdale. Ces règles appliquées aux séquences de la protéine du virus de la mosaïque du tabac et du lysozyme ont permis de prédire des structures secondaires qui correspondent aux observations expérimentales connues pour ces protéines. Ces règles permettent ainsi de compendre comment certaines séquences d'acides aminés influencent les régions hélicoïdales et non hélicoïdales dans les structures secondaires des protéines.

Le développement d'un algorithme capable de prédire la structure tridimensionnelle des protéines à partir de leurs séquences d'acides aminés représente un défi majeur en biochimie. Le

projet CASP, initié en 1994 (Moult *et al.*, 1995), est un concours biannuel où des groupes de recherche du monde entier se réunissent pour prédire et valider des structures de protéines à partir de séquences cibles. Les avancées dans ce domaine reposent en grande partie sur la base de données Protein Data Bank (PDB) (Berman *et al.*, 2000), qui contient aujourd'hui plus de 200 000 structures protéiques. Toutefois, de nombreux défis persistent en matière de prédiction précise des structures, en particulier lorsque les séquences cibles ne correspondent à aucune séquence connue dans la PDB. Trois catégories de prédiction ont été explorées en 1994 : la modélisation comparative ou par homologie, la reconnaissance de repliements et le pliage ab initio. Malgré les avancées technologiques et méthodologiques, prédire les structures des protéines, surtout celles sans analogues proches dans la PDB, reste complexe. Des améliorations continues sont nécessaires, notamment pour mieux comprendre les interactions physiques sous-jacentes et pour réduire la dépendance aux structures déjà connues, ce qui ouvrirait la voie à des avancées dans l'étude des protéines et leur fonctionnement.

En 2018, soit 24 ans plus tard le CASP13 (Kryshtafovych *et al.*, 2019) voit concourir les groupes AD7, Zhang, MULTICOM, QUARK avec respectivement AlphaFold1 (Senior *et al.*, 2019), I-TASSER (Zheng *et al.*, 2019), MULTICOM (Hou *et al.*, 2019), QUARK (Zheng *et al.*, 2019) , où AlphaFold1 a excellé dans la catégorie topologie (Free Modeling) pour prédire de nouveaux repliements protéiques. En effet CASP13 (Kryshtafovych *et al.*, 2019) regroupait plusieurs catégories notamment la Modélisation de haute précision, la topologie (anciennement Modélisation libre), la prédiction des contacts, l'affinage, l'assemblage, l'Estimation de la précision, l'assistance par données et la pertinence biologique. Leur méthode combine l'analyse co-évolutive et les réseaux de neurones profonds pour identifier et transformer les couplages co-évolutifs en cartes de contact. L'édition suivante CASP14 (Kryshtafovych *et al.*, 2021) une version améliorée de AlphaFold1, AlphaFold2 (Jumper *et al.*, 2021a) (Jumper *et al.*, 2021b), remporte de nouveau la catégorie Free Modeling en apportant une amélioration substantielle en terme de précision des prédictions de structure et dans l'utilisation d'une architecture de réseau plus avancée qui permet une intégration plus globale et efficace aussi bien de la structure que des séquences.

1.1.2 Origine de AlphaFold

AlphaFold est un outil d'intelligence artificielle conçu par le groupe AD7 de DeepMind qui prédit la structure en trois dimensions des protéines à partir de leur séquence d'acides aminés. Il s'appuie sur des alignements de séquences multiples de protéines similaires et sur un réseau de neurones basé sur l'apprentissage profond pour calculer les distances entre les paires d'acides aminés. Il existe trois versions du programme, AlphaFold 1 (Senior *et al.*, 2020), AlphaFold 2 (Jumper *et al.*, 2021b) et AlphaFold3 (Abramson *et al.*, 2024) qui se différencient par leurs méthodes algorithmiques et leurs architectures distinctes.

1.1.3 Mode de fonctionnement de AlphaFold2

AlphaFold 1 (Senior *et al.*, 2020), lancé en 2018, reposait sur une méthode combinant des principes de physique locale et un potentiel guide basé sur la reconnaissance de motifs . Cependant, cette approche avait tendance à exagérer les interactions entre les résidus voisins dans la séquence protéique, ce qui entraînait un sur-ajustement des prédictions. En conséquence, les modèles générés présentaient des structures secondaires légèrement amplifiées par rapport à la réalité, avec une surreprésentation des hélices alpha et des feuillets bêta, souvent plus réguliers, rigides ou étendus que ce qui est observé dans les structures protéiques réelles.

AlphaFold 2 (Jumper *et al.*, 2021b) , sorti en 2020, repose sur une approche avancée combinant apprentissage profond, reconnaissance des motifs à travers des alignements multiples de séquences, prédiction des distributions de distances continues entre résidus, et estimation des angles de torsion pour construire des structures tridimensionnelles de protéines avec une précision remarquable.

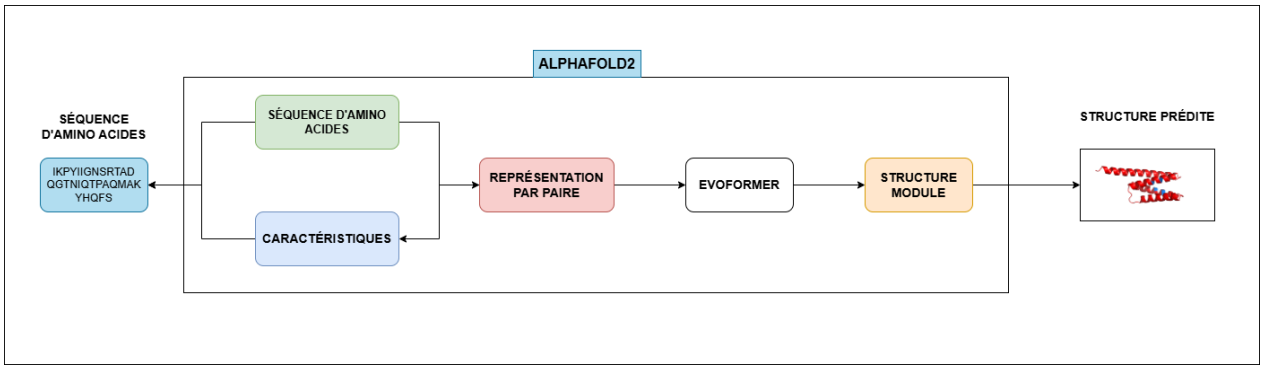

Figure 1.1 Figure montrant le passage d'une séquence dans la structure de AlphaFold2.

L'amélioration significative des prédictions de structures protéiques par AlphaFold résulte de l'adoption de nouvelles **architectures de réseaux neuronaux** et de **méthodes d'entraînement** prenant en compte les contraintes évolutives, physiques et géométriques des protéines. Le réseau AlphaFold2 utilise une architecture innovante qui intègre simultanément les alignements multiples de séquences (MSA) et les interactions entre paires de résidus. Il propose également une nouvelle représentation des sorties accompagnée d'une fonction de perte optimisée pour une prédiction précise de la structure de bout en bout. De plus, il intègre une architecture d'attention équivariante, exploite des pertes intermédiaires pour affiner progressivement les prédictions, utilise une perte MSA masquée pour un entraînement conjoint avec la structure, et applique des techniques d'auto-distillation ainsi que des auto-évaluations de précision pour l'apprentissage à partir de séquences protéiques non étiquetées.

L'Evoformer constitue le coeur de réseau AlphaFold2. Son objectif principal est d'identifier et d'analyser les relations entre les paires de résidus au sein de la séquence d'entrée. Pour ce faire l'Evoformer traite les entrées à travers plusieurs couches répétées, en produisant une matrice de taille nombre de séquences par nombre de résidus représentant un MSA transformé et une matrice de taille nombre de résidus par nombre de résidus représentant les relations entre paires de résidus. Les blocs Evoformer intègrent à la fois des éléments exploitant des mécanismes d'attention et d'autres ne reposant pas sur cette approche comme le moyenne des produits extérieurs(outer product mean), le réseau Feed-Foward, le dropout et le triangulation multiplicative update.

Avant d'appliquer les triangulations sur les relations entre résidus, les relations mises à jour issues de la MSA sont intégrées dans la représentation des paires, permettant d'ajuster les arêtes du graphe. Ensuite, ces données sont réinjectées pour pondérer les informations MSA par ligne, suivies d'une nouvelle série d'analyses en ligne et en colonne.
Ce processus itératif est exécuté dans **l'Evoformer**, affinant progressivement les modèles de la MSA initiale et les représentations des paires, garantissant ainsi une représentation optimisée pour la génération de la structure finale.
L'Evoformer est suivie du **module de structure** qui applique des rotations et translations à chaque acide aminé de la protéine pour obtenir une première estimation de la **structure de la protéine en 3D**. Il impose également des contraintes physiques et chimiques basées sur les angles de liaison atomique et les angles de torsion. Les modèles ajustés, ainsi que la sortie du module de structure, passent par le traitement de l'ancien modèle et du module de structure **trois fois supplémentaires**, soit **quatre cycles au total**, avant d'obtenir le résultat final.

1.2 Machine learning et design de protéines : RFDiffusion

RFdiffusion (Watson *et al.*, 2023) est utilisé pour prédire une nouvelle séquence d'acides aminés qui devrait se replier en une structure tridimensionnelle cible. Il utilise des algorithmes de machine learning qui ont été formés sur des bases de données de protéines connues, où la relation entre la séquence et la structure est bien caractérisée.
RFdiffusion est utilisé par exemple, pour générer des oligomères symétriques, pour concevoir des liants, pour concevoir des protéines avec des plis spécifiques souhaités pour l'échafaudage de motifs fonctionnels ainsi que pour l'échafaudage de motifs symétriques et aussi pour la mise en place de ces derniers dans des échafaudages symétriques.

1.2.1 L'idée de la diffusion

En apprentissage automatique, la méthode de diffusion consiste à ajouter progressivement du bruit à des données telles que des images ou des séquences les transformant ainsi de manière contrôlée. Suite à cela, un modèle d'intelligence artificielle, généralement un réseau de neurones,

est entraîné à inverser ce processus afin de reconstruire fidèlement les données originales à partir du bruit généré (Croitoru *et al.*, 2023).

1.2.2 RoseTTAFold

Cette partie vient poser les bases de RFDiffusion (Watson *et al.*, 2023). En effet RFdiffusion et RoseTTAFold (Baek *et al.*, 2021) ont été conçus par le même groupe. Étant donné que RoseTTAFold est utilisé dans RFdiffusion, nous en parlons d'abord afin de pouvoir introduire RFdiffusion.

Baek et al. ont en 2021 (Baek *et al.*, 2021) proposé une nouvelle approche, RoseTTAFold, pour prédire avec précision la structure des protéines. Ils ont ajouté une troisième voie de structure parallèle utilisant des coordonnées 3D, en plus des informations sur la séquence d'acides aminés en 1D et les cartes de distances en 2D.
Cette architecture permet aux informations de circuler entre les différents niveaux, ce qui permet au réseau de raisonner collectivement sur les relations entre les séquences, les distances et les coordonnées. En raison des limitations de mémoire du matériel informatique, les modèles ont été entraînés sur des cultures discontinues de segments de séquence d'entrée plutôt que sur de grandes protéines.
Deux approches ont été utilisées pour générer les structures 3D finales : en utilisant pyRosetta pour générer des modèles à atomes entiers à partir des prédictions de distance et d'orientation, ou en transmettant les données moyennes à une composante du modèle capable de gérer la géométrie spatiale, afin de générer directement l'ossature de la protéine.
Cette nouvelle méthode présente l'avantage d'une faible utilisation de la mémoire GPU et de produire des modèles de chaînes latérales complets, mais nécessite une utilisation de CPU pour l'étape de modélisation de la structure de pyRosetta.
RoseTTAFold marche assez bien lorsqu'il n'y a pas d'alignement de séquences. Pour profiter à cela il est utilisé pour des modèles de diffusion.

1.2.3 RFdiffusion

Nous ajoutons RFdiffusion dans notre projet parce qu'elle nous permet de trouver la bonne séquence qui va se replier en une structure cible. RFdiffusion nous propose une séquence que nous testons avec AlphaFold2 afin de nous assurer qu'elle soit consistante entre différents modèles. RFdiffusion (Watson *et al.*, 2023) repose sur le concept de diffusion inverse, qui est une méthode issue des modèles de diffusion employés en intelligence artificielle pour créer des images. Ce concept s'inscrit dans un cadre plus vaste connu sous le nom de processus de diffusion stochastique, souvent utilisé dans le traitement du signal et des données (Blau *et al.*, 2022).
RFdiffusion commence par utiliser RosettaFold pour générer une première prédiction de la structure de protéines. RosettaFold fournit une estimation initiale de la structure 3D à partir de la séquence de la protéine.
Par la suite, la méthode de diffusion est employée pour perfectionner la structure initiale. Cette approche vise à affiner et améliorer la prédiction en modélisant des détails et des variations qui pourraient ne pas avoir été entièrement capturés dans la première estimation.
Cela se réalise en produisant des échantillons additionnels à partir de la structure initiale et en ajustant progressivement la prédiction.
RFdiffusion génère une séquence d'acides aminés qui est conçue pour adopter une structure cible donnée, tandis que RoseTTAFold est utilisé pour vérifier que cette séquence se replie effectivement selon la structure souhaitée. Cette approche permet de concevoir des séquences compatibles avec une architecture tridimensionnelle précise, en combinant génération et validation.
Aussi convient-il de notifier que RoseTTAFold est l'architecture utilisée pour RFdiffusion mais d'autres architectures auraient pu être utilisées notamment AlphaFold (Jumper *et al.*, 2021b), OmegaFold (Wu *et al.*, 2022), ESMFold (Lin *et al.*, 2023) et bien d'autres architectures encore.

1.2.4 Mode de fonctionnement de RFdiffusion

Dans la pratique, RFDiffusion est un modèle génératif inversé qui prône l'élimination itérative du bruit à chaque étape.
L'approche de Watson et al. (Watson *et al.*, 2023) a consisté donc à prendre un réseau existant de prédiction de structure du nom de RosettaFold, et à l'ajuster (fine-tuning) pour qu'il devienne un

réseau de débruitage dans un modèle de diffusion.
En résumé, le travail de RosettaFold (Baek *et al.*, 2021) est de prendre en entrée une séquence et des coordonnées pour essayer de prédire de vraies structures de protéine. Ceci étant, Watson et al. ont développé un nouvel outil, RFdiffusion, qui s'appuie sur les fondations de RoseTTAFold tout en allant bien au-delà. Il s'agit d'un modèle génératif, conçu pour produire des séquences capables d'adopter des structures protéiques cibles, en tirant parti des forces de RoseTTAFold en prédiction de structure.
Une fois le choix du réseau définit et l'architecture mise en place deux stratégies d'entraînement ont été adoptées : **l'approche proche des modèles de diffusion**, où les prédictions à chaque étape sont indépendantes des étapes précédentes (Trippe *et al.*, 2022) (Anand et Achim, 2022) et **l'autoconditionnement** inspiré du recyclage dans AlphaFold2 car Watson et al. se sont rendu compte que le fait de donner accès à un modèle sur une prédiction antérieure pouvait l'aider à obtenir de meilleurs résultats sur la prédiction suivante.
RFdiffusion prend donc le résultat de prédiction précédente qu'il réintroduit dans le modèle à la prédiction suivante. RFDiffusion génère des squelettes de protéines, trouve des séquences avec la Protein Message Passing Neural Network (ProteinMPNN) (Dauparas *et al.*, 2022) et ensuite fait la prédiction avec AlphaFold. Il compare ensuite le Root Mean Square Deviation (RMSD) entre AlphaFold et la conception sachant que plus faible est le RMSD, meilleur est le design. Une autre métrique aussi qui est observée c'est la confiance dans la prédiction par AlphaFold caractérisée par le Predicted Aligned Error (PAE) (Jumper *et al.*, 2021b).

1.2.5 Les limites de RFDiffusion

Si RFdiffusion (Watson *et al.*, 2023) a démontré des capacités impressionnantes dans la conception de protéines, Watson et al. s'efforcent actuellement de remédier à certaines limites auxquelles est confrontée cette méthode.
L'efficacité des modèles d'apprentissage profonds tels que la RFdiffusion dépend de manière significative de la qualité et de la quantité des données d'apprentissage. Dans le domaine de la conception de protéines, les structures de protéines expérimentales à haute résolution restent

rares, en particulier pour les protéines complexes et les nouveaux repliements.

Le manque de données peut restreindre la faculté du modèle à s'appliquer à des structures de protéines inédites pour lui. La prédiction des structures de protéines nécessite une compréhension approfondie des interactions complexes entre les acides aminés, qui englobent à la fois les interactions locales et les interactions à longue portée sur l'ensemble de la chaîne de protéines.

La saisie précise de ces interactions à longue portée représente un défi important pour les modèles d'apprentissage profond, ce qui peut se traduire par des prédictions moins précises pour des structures de protéines spécifiques.

À l'instar de nombreux modèles d'apprentissage profond, RFdiffusion est très sensible à la sélection des hyperparamètres, qui régissent le processus d'apprentissage. L'identification des hyperparamètres optimaux pour une tâche particulière de conception de protéines nécessite une expérimentation méticuleuse et des connaissances spécialisées, ce qui en fait une entreprise longue et difficile.

Bien que RFdiffusion puisse produire des séquences, il peut être difficile de déchiffrer le raisonnement sous-jacent à ces prédictions. Ce manque d'interprétabilité rend complexe l'évaluation de la fiabilité des prédictions, en plus de rendre difficile l'identification des biais potentiels ou des erreurs introduites par le modèle.

À tout prendre, il est clair que Watson et al. s'efforcent activement de contourner ces limites en employant diverses méthodes, dont le perfectionnement des architectures des modèles, l'intégration de connaissances biophysiques additionnelles, le développement de nouvelles méthodes de collecte de données, et la conception de techniques visant à améliorer l'interprétabilité et à diminuer les coûts de calcul.

Malgré ces défis, RFdiffusion reste un outil intéressant pour la conception de protéines, offrant un potentiel important pour faire progresser notre compréhension de la structure et de la fonction des protéines. Au fur et à mesure que le domaine de la conception de protéines progresse, nous pouvons nous attendre à de nouvelles améliorations des capacités et des limites de la RFdiffusion et d'autres méthodes basées sur l'apprentissage profond.

1.3 Biogenèse des protéines SF4

1.3.1 La conception in vivo

Pour comprendre le mécanisme de formation des protéines fer souffre, des études ont été faites sur ce qui se fait déja dans le vivant (Jagilinki *et al.*, 2020) (Grzyb *et al.*, 2012) et il en ressort que c'est un processus qui se fait naturellement dans les cellules. Une hypthèse a été établie selon laquelle il existe des protéines spécifiques dans la cellule qui sont responsables de l'assemblage des atomes de fer et de souffre en cluster SF4. Jagilinki (Jagilinki *et al.*, 2020) et al. ont énuméré trois systèmes d'assemblage et d'insertion des clusters SF4 dans les protéines cibles : ISC (iron-sulfur cluster), SUF (mobilization of sulfur), NIF (nitrogen fixation).

Le système ISC est une machinerie cellulaire impliquée dans l'assemblage et la maturation des clusters SF4 dans des conditions normales de croissance. Il repose sur des protéines qui assurent l'extraction du soufre, la formation du cluster et son transfert aux protéines cibles.

Le système SUF est une machinerie cellulaire impliquée dans l'assemblage et la réparation des clusters SF4 lorsque la cellule subit un stress oxydatif ou un manque de fer. Il repose sur l'opéron sufABCDSE pour extraire le soufre, former le cluster et le transférer aux protéines nécessaires. Ce système joue un rôle clé dans l'adaptation aux environnements riches en oxygène et permet la survie dans des conditions difficiles.

Le système NIF est une voie biologique essentielle à la formation des clusters SF4, principalement impliquée dans la maturation de la nitrogénase, une enzyme clé de la fixation de l'azote. Présent chez certaines bactéries fixatrices d'azote, il repose sur des protéines spécifiques qui assemblent et insèrent ces clusters dans les protéines cibles.

Les morceaux du mécanisme d'assemblage des protéines SF4 sont certes connus, cependant enlever le cube SF4 dans une structure pourrait avoir des incidences majeures non seulement sur la structure mais aussi sur la fonction de la structure. Aussi nous ne savons pas exactement ce qui est reconnu dans la protéine pour recevoir ou capturer le cube ou encore le passer à une protéine qui en est dépourvue.

1.4 Quelques outils utilisés dans notre travail.

Cette rubrique fait référence aux différents outils utilisés aussi bien dans ce travail que pour la rédaction de ce mémoire. En effet, nous nous sommes appuyés sur certains outils pour réaliser ce travail, outils qui nous ont considérablement facilité la tâche. Les lignes suivantes feront un inventaire de tous ces outils.

1.4.1 Biopython

Biopython (Cock *et al.*, 2009) est un projet open source, développé par des développeurs bénévoles, qui fournit un large éventail de bibliothèques python et ce dans le but de résoudre bon nombre de problèmes en bio-informatique.
Biopython contient différents modules qui nous ont été d'une aide précieuse telle que la lecture et l'écriture de différents formats de fichiers de séquences et d'alignements de séquences multiples, le traitement de structures moléculaires en 3D, l'interaction avec des outils courants tels que BLAST (Altschul *et al.*, 1990), ClustalW (Thompson *et al.*, 1994) l'accès aux principales bases de données en ligne, ainsi que pour l'accès à des méthodes pour l'apprentissage statistique.

1.4.2 Texshade

TEXShade est un outil utilisé dans le domaine de la bio-informatique pour analyser et visualiser des données structurales. Il permet notamment de créer des représentations graphiques telles que des cartes de similarité ou des alignements de séquences, mettant en lumière les zones conservées ou divergentes. Voici un aperçu de ses principales fonctionnalités :
TEXShade (Beitz, 2000) est utilisé pour générer des graphiques qui présentent de manière visuelle les similitudes et les variations entre les séquences biologiques alignées, facilitant ainsi la visualisation des zones conservées et des variations au sein des séquences.
Avec TEXShade, il est possible de personnaliser les représentations graphiques en ajustant les couleurs, les échelles et les styles de présentation, ce qui permet de rendre les données d'intérêt plus clairement visibles et compréhensibles.

TEXShade est utile pour les chercheurs en génétique, biologie moléculaire et bio-informatique, car il facilite l'analyse et la visualisation de données complexes de manière précise et instructive.

1.4.3 Pymol

PyMOL (Schrödinger, LLC, 2015) (Python Molecular Graphics System), est un outil puissant et polyvalent utilisé pour la visualisation et l'analyse de structures moléculaires en 3D. Voici un résumé de ses principales caractéristiques et fonctionnalités :
PyMOL se distingue par sa capacité à générer des visualisations 3D interactives et de haute qualité de divers types de structures moléculaires, telles que les protéines, les acides nucléiques, les petites molécules et les complexes.
PyMOL propose une gamme étendue d'options pour représenter les molécules, comme les modèles bâtonnets et billes, les rubans en cartoon, ainsi que les surfaces. Les utilisateurs peuvent ajuster le niveau de détail et les schémas de coloration pour une meilleure clarté visuelle.
PyMOL offre une manipulation interactive des molécules en 3D, permettant aux utilisateurs de les faire pivoter, zoomer et déplacer pour les observer sous différents angles.
PyMOL propose des fonctionnalités pour analyser les interactions moléculaires, les distances entre les atomes, les angles entre les liaisons, ainsi que pour calculer d'autres propriétés pertinentes.
PyMOL permet aux utilisateurs d'écrire des scripts en Python pour automatiser des tâches répétitives, effectuer des calculs complexes et étendre ses fonctionnalités. Cette capacité de scriptage rend PyMOL hautement personnalisable et adaptable aux besoins spécifiques de la recherche.
En conclusion, PyMOL est un outil utile pour les chercheurs de divers domaines scientifiques, leur permettant de visualiser et d'analyser des structures moléculaires en 3D.

CHAPITRE 2
MÉTHODOLOGIE

2.1 Idée générale

L'objectif est d'intégrer de manière artificielle le cluster SF4 dans des protéines d'intérêt , ce qui pose des défis, car ce cluster réagit au contact de l'oxygène. Pourtant, certaines cellules parviennent déjà à le faire naturellement. Des expériences sont donc menées pour comprendre comment insérer le cube SF4 dans différentes structures de protéines. Nous cherchons à insérer ce cluster dans des structures de protéines spécifiques de manière similaire. Il existe une protéine d'intérêt, le CCIS1 0.2 (Grzyb *et al.*, 2010) capable de recevoir un cluster SF4, et nous souhaitons identifier un motif dans les protéine qui facilite cette insertion. Le but final est de lier le cluster SF4 à ce motif et ainsi réussir à **laisser la cellule intégrer le cluster dans des protéines** qui n'en contiennent pas initialement.

Dans cette partie nous décrivons les différents procédés qui nous ont conduits à trouver des interfaces pour récupérer les clusters SF4, extraire leurs séquences respectives et comment nous les utilisons pour ce projet.
Nous avons au prime abord défini le problème dans cette section 2.1. Ensuite dans les sections 2.2 et 2.3 nous tenterons de décrire le processus qui nous a permis de trouver, de récupérer les structures de protéines contenant le cluster SF4 pour trouver ou localiser à l'intérieur le cluster SF4 et leur proximité par rapport à différentes parties de la structure des protéines.
Ensuite, dans les sections 2.4 et 2.5 nous allons extraires toutes les chaînes polypeptidiques contenant le cluster SF4, classifier les résultats obtenus en groupes par des arbres et des alignements et récupérer les groupes intéressants. Nous aborderons ultérieurement, aux sections 2.6 et 2.7, l'identification de motifs au sein de la séquence, suivie de la détermination des suites alternatives nous permettant de capturer le cluster SF4. Donc suivant l'ordre de ce qui a été prévu nous allons commencer par utiliser la séquence de notre protéine d'intérêt, pour voir si la prédiction de la structure fonctionne avec elle puis ensuite effectuer le design de séquences alternatives qui ont

toujours la bonne structure suivant les prédictions de AlphaFold2 tout en ayant les bonnes propriétés pour pouvoir interagir avec le cube SF4.

Ci-joint la figure 2.1 retraçant ce que nous allons faire tout au long de ce travail ainsi que les fonctions et outils que nous allons utiliser.

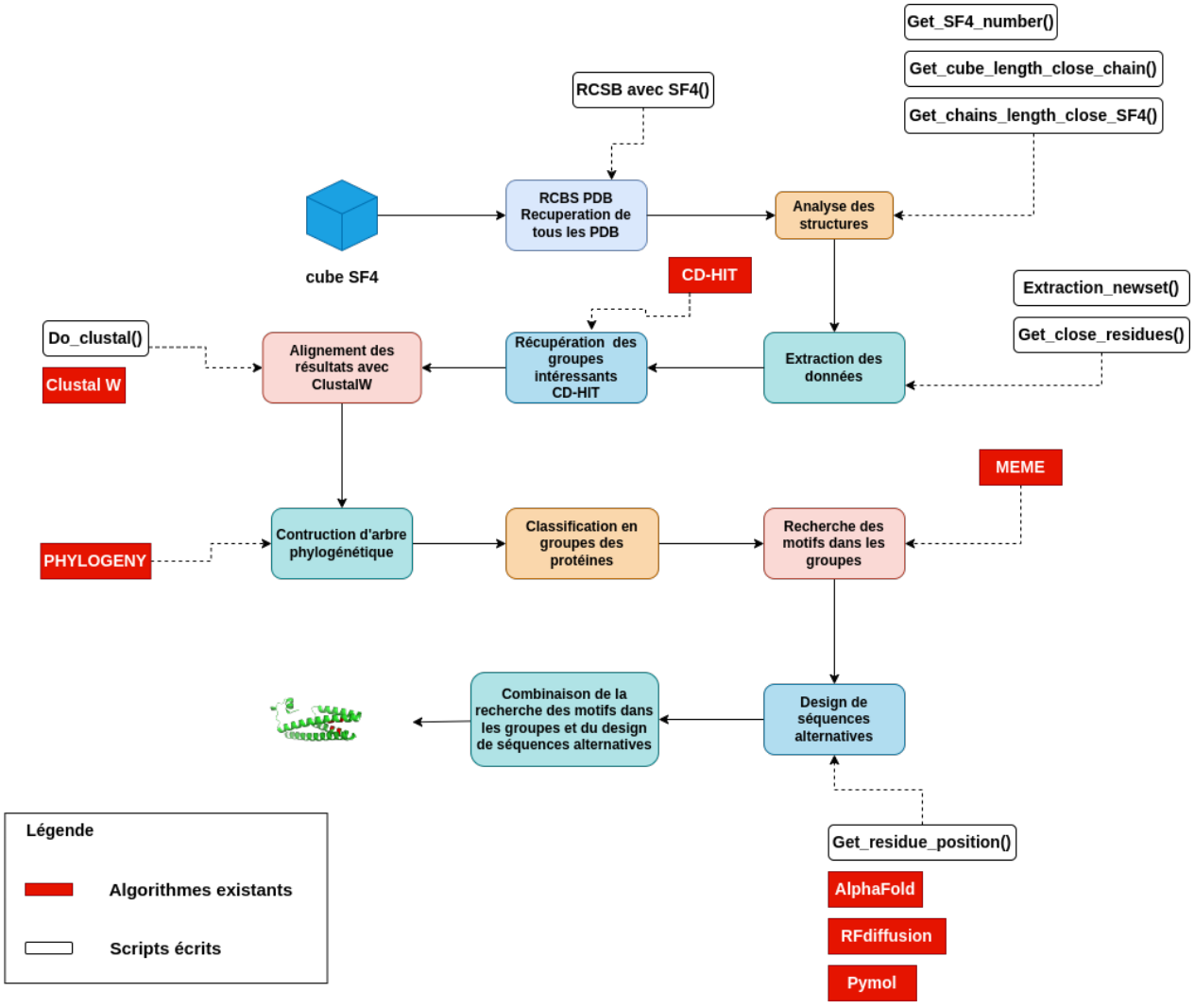

Figure 2.1 Figure montrant les différents processus suivis dans notre travail.

2.2 Trouver et récupérer les protéines

Le jeu de données que nous avons utilisé pour ce travail a été construit à partir de structures de protéines de la banque de données des protéines de la Research Collaboratory for Structural Bioinformatics RCSB (Burley *et al.*, 2025). Cette banque de données contient des fichiers (données) depuis son origine sous format PDB et par la suite des fichiers sous format mmCIF. Aussi 6000 à 7000 structures s'y ajoutent chaque année.
Dans cette banque de données, la recherche a été faite sur base de toutes les structures contenant le cluster SF4 et dont le nom scientifique de l'organisme source est Escherichia coli. Ces structures ont par la suite été téléchargées et empaquetées dans un dossier commun.
La commande qui nous a permis de faire la sélection est la suivante :

```
QUERY: Full Text = "sf4" AND Scientific Name of the Source Organism =
"Escherichia coli".
```

Cependant, nous avons un script qui nous permet de faire ce travail plus facilement. Le script se base sur la requête précédemment utilisée et télécharge tous les résultats issus de la requête.

Script 1: Download CIF Files Based on Text Query

```
Procedure Main(path_cif) :
    q1 ← TextQuery("sf4")
    q2 ← attrs.rcsb_entity_source_organism.ncbi_scientific_name == "Escherichiacoli"
    query ← q1&q2
    uids ← list(set(x[ :4] for x in query("assembly")))
    foreach uid in uids do
        DwldCif(uid, path_cif)
Function DwldCif(uid, PATH_CIF) :
    out_path ← PATH_CIF
    out_path.mkdir(parents = True, exist_ok = True)
    out_path ← out_path/(uid +' .cif')
    if out_path.is_file() then
        return
    while True do
        Try {
        out ← requests.get("https ://files.rcsb.org/download/" + uid + ".cif")
        with out_path.open('w')as f
        f.write(out.content.decode('utf − 8'))
        Break;
        }
        except{
        requests.exceptions.ReadTimeout as e sleep(1)
        print(e)
        }
        except{
        requests.exceptions.ConnectionError as e sleep(1)
        print(e)
        }
Main(Path('../Data'))
```

2.3 Analyse du dataset

Nous avons commencé par afficher la structure d'une protéine du point de vue de ses différentes composantes et nous avons vu qu'une structure est composée de modèles, les modèles eux-mêmes composés de chaînes, les chaînes à leur tour composées de résidus et les résidus composés d'atomes suivant la hiérarchie suivante SMCRA(Structure/Model/Chain/Residue/Atom). Le

terme **structure** fait référence à la protéine entière, **Model** à la conformation 3D. Il y en a souvent un seul. Le terme **Chain** fait allusion aux chaînes polypeptidiques contenues dans la structure et le terme **Residu** correspond aux acides aminés contenus dans les chaînes polypeptidiques. Finalement le terme **Atom** indique les atomes contenus dans les acides aminés.
Cette partie permet de se familiariser avec le type de données sur lequel le travail est effectué et de voir les différentes relations de la PDB. Cette relation permet de voir les différentes proportions de chaînes polypeptidiques et d'acides aminés autour du cluster sf4.L'accent alors sera mis dans cette partie sur les différentes analyses faites sur notre dataset, les scripts qui ont été écrits sur ces derniers ainsi que les résultats obtenus.

2.3.1 Nombre de cube SF4

Nous voudrions comprendre un processus compliqué, celui de comment intégrer le cluster SF4 à la protéine. Nous avons donc estimé important de commencer avec des exemples simples. Nous voudrions nous concentrer sur les structures de protéines contenants 1 seul cluster SF4. Le cluster SF4 dans la structure de protéine de la PDB est noté dans le fichier de structure comme étant un résidu. Il était important de voir le nombre de clusters SF4 contenus dans chacune les structures de protéines. Pour ce faire, un script a été écrit dans ce sens pour pouvoir voir le nombre de clusters SF4 dans chacune des structures de protéines présentes dans le dataset. Tout cela est décrit par le script ci-dessous.

Script 2: get_sf4_number

Data: structure

```
1  models ← structure.get_models();
2  chains_length, chains, residues, sf4chain, residuesf4, sf4_count ← [];
3  for model in structure do
4      for chain in model do
5          chains.append(chain.id);
6          chains_length.append(len(chain));
7  atoms ← structure.get_atoms();
8  residues ← structure.get_residues();
9  for residue in model.get_residues() do
10     if residue.resname == "SF4" then
11         ligand ← residue;
12         residuesf4.append(ligand);
13         chain ← ligand.get_parent();
14         sf4chain.append(chain.id);
15 sf4_count ← count_to_dict(sf4chain);
16 return list(sf4_count.values());
```

Un premier traitement a été fait sur les structures de protéines par cet algorithme dans le but de savoir exactement quelle était la proportion du nombre de cluster cluster SF4 contenus dans chaque chaîne polypeptidique de chacune des structures de protéines. La réponse est traduite par la figure 2.2 ci-après.

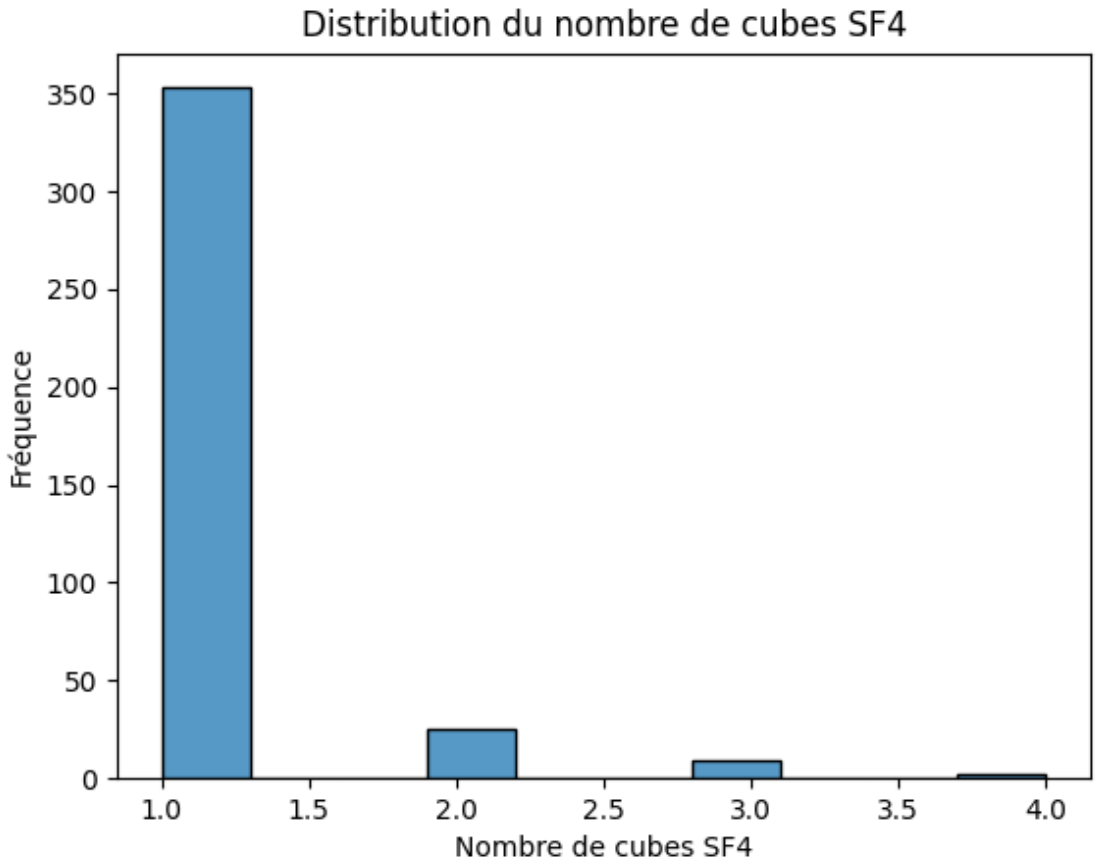

Figure 2.2 Diagramme montrant la proportion du nombre de cubes dans chaque structure de protéine

Dans cette figure, nous voyons que la plupart des chaînes contiennent un seul cluster SF4 , peu en contiennent 2, parfois 3 clusters SF4 et rares sont ceux qui hébergent 4 cubes à l'intérieur d'elles.

2.3.2 La proportion de la taille des chaines autour du cube SF4.

Connaissant la proportion du nombre de clusters SF4 contenus dans les chaînes des structures, la seconde préoccupation était de savoir quelle était la taille des chaines qui étaient proche du clusters, c'est-à-dire dans le voisinage du clusters, le voisinage étant défini comme le nombre de chaînes dans un rayon de 9 angströms ce qui équivaut à 9×10^{-10} mètres autour du cluster. Le script qui nous a permis de répondre à cette préoccupation est celui de *get_chains_lengths_close_sf4*.

Script 3: get_chains_lengths_close_sf4

```
Data: structure
cutoff_distance ← 9;
i, o ← 0;
neighbor_chains ← [];
sf4chain, residuesf4, sf4_count, chains_length, chains ← [];
models ← structure.get_models();
for model in structure do
    for chain in model do
        chains.append(chain.id);
        chains_length.append(len(chain));
atoms ← structure.get_atoms();
residues ← structure.get_resid();
for residue in model.get_residues() do
    if residue.resname == "SF4" then
        ligand ← residue;
        residuesf4.append(ligand);
        chain← ligand.get_parent sf4chain.append(chain.id);
        atm_coord_list←[];
        neighbors ← -8;
        for sf4atom in ligand do
            sf4atom_coord ← sf4atom.get_coord();
            i ← i+1;
            neighbor ← neighborsearch.search(sf4atom_coord, cutoff_distance);
            residue_list ← Selection.unfold_entities(neighbor, 'R') for residuen in residue_list
             do
                o ← o+1;
                neighbor_chains.append(residuen.get_parent().id);
chain_of_sf4res = count_to_dict(neighbor_chains);
return list(chain_of_sf4res.values());
```

L'algorithme nommé *get_chains_lengths_close_sf4* 4 extrait de la structure toutes les chaînes contenant le cluster SF4 ainsi que le nombre de ces cubes présents dans chaque chaîne. Ces informations sont ensuite organisées dans un dictionnaire. Cependant, dans ce contexte, l'accent est mis spécifiquement sur la taille des chaînes, en particulier le nombre d'acides aminés situés à proximité du cluster SF4. Le résultat est décrit par la figure 2.3 qui suit.

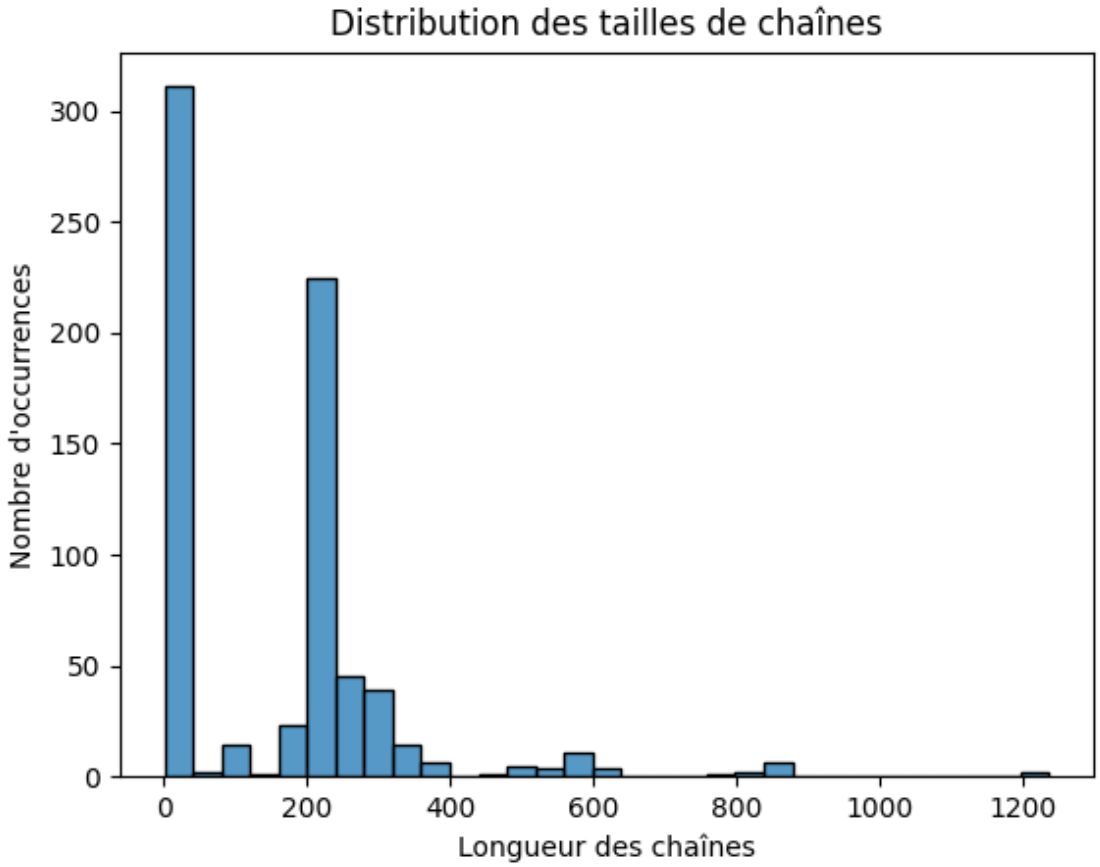

Figure 2.3 Diagramme montrant la proportion de la taille des chaines (nombre de résidus) autour du cube SF4

Dans la figure 2.3, nous pouvons voir qu'il y a une forte proportion (plus de 250) de chaînes contenant entre 1 et 25 acides aminés, une assez forte proportion contenant entre 200 et 400 acides aminés, une faible proportion contenant entre 500 et 900 acides aminés et de très rares qui contiennent 1200 et plus résidus.

2.3.3 La proportion du nombre de cubes qui sont proches des chaînes contenues dans les structures de protéines

Nous voudrions déterminer pour chaque structure de protéine spécifique, le nombre de clusters SF4 que nous pouvons trouver dans son voisinage. Ce voisinage est défini comme une zone de 9 angströms autour de la chaîne polypetidique contenant le cluster SF4.

L'algorithme *get_cube_length_close_chain* a été conçu pour répondre à cette question.

Script 4: get_cube_length_close_chain

```
Data: structure
1  i, o ← 0;
2  neighbors ← -8;
3  cutoff_distance ← 9;
4  all_sf4_atoms, neighbor_chains, close_cube, nb_cube ← [];
5  models ← structure.get_models();
6  atoms ← structure.get_atoms();
7  for model in structure do
8      for chain in model do
9          for residue in chain do
10             if residue.resname == "SF4" then
11                 ligand ← residue;
12                 ligand_chain ← residue.get_parent();
13                 ligand_chain_atoms ← ligand_chain.get_atoms();
14                 all_sf4_atoms ← list(ligand.get_atoms());
15                 neighbor_search ← NeighborSearch(list(all_sf4_atoms));
16                 for atom in ligand_chain_atoms do
17                     atom_coord ← atom.get_coord();
18                     i ← i+1;
19                     neighbor ← neighbor_search.search( atom_coord, cutoff_distance);
20                     neighbors+=len(neighbor);
21                     residue_list ← Selection.unfold_entities(neighbor, 'R');
22                 nb_cube.append(len(residue_list));
23 return nb_cube;
```

Le résultat est décrit par la figure 2.4 ci-dessous :

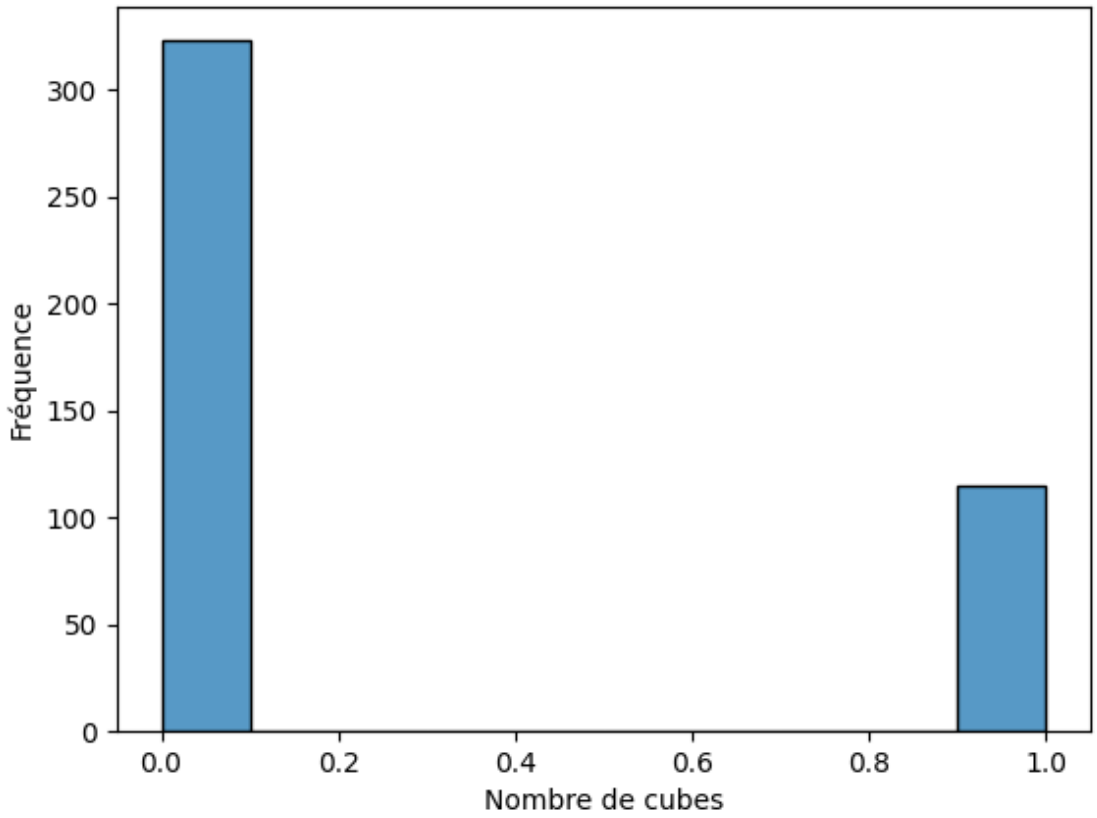

Figure 2.4 Diagramme montrant la proportion du nombre de cube qui sont proches des chaines contenues dans les structures de protéine.

Comme nous pouvons le voir dans la figure ci-dessous près du quart des protéines contenant des clusters SF4 sont proches des chaînes polypeptidiques de certaines structures de protéines c'est-à-dire dans un rayon de 9 Armstrong autour de la chaîne.

2.4 Préparation du jeu de données

Après avoir analyser les structures de protéines ainsi que leur interaction avec le cluster SF4, l'étape suivante a consisté à convertir les données en divers formats spécifiques. Cette démarche vise à permettre la réalisation de différentes expérimentations.

2.4.1 Les résidus proches et loins du cube

Dans un premier temps pour toutes les structures de protéines extraites constituant notre jeu de données, l'intérêt a été porté sur les chaînes à proximité de SF4. La question ici est de savoir quels étaient les résidus proches du cluster ou non. L'image, ci-dessous, illustre bien ce que nous voulons avoir avec nos séquences.

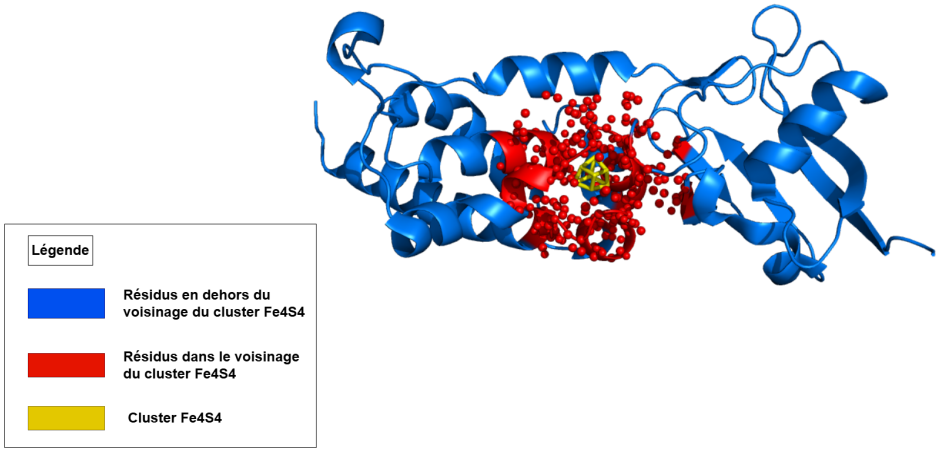

Figure 2.5 Figure montrant le voisinage de la chaine B de la structure 1L0V. Le cluster SF4 est mis en jaune, les acides aminés qui sont dans le voisinage soit dans un rayon de 9 Armstrong autour du cluster sont mis en rouge et les acides aminés qui ne sont pas le voisinage du cluster sont mis en bleu.

Dans la figure 2.5, les acides aminés éloignés du cluster SF4 sont représentés en bleu, tandis que ceux qui en sont proches apparaissent en rouge. L'objectif est donc d'extraire la séquence de chaque chaîne et, pour chaque acide aminé, de déterminer s'il est proche du cluster ou non.. Nous avons écrit à cet effet un algorithme qui nous permet de faire l'extraction. Nous l'avons nommé *extraction_newSet*.

Script 5: Extraction du jeu de données à partir d'un résidu SF4

```
Function extractNewSet(structure) :
    i ← 0; cutoff ← 9; neigh ← −8
    for model ∈ structure.get_models() do
        for chain ∈ model do
            if len(chain) < 300 then
                for res ∈ chain do
                    if res.id[0] =′ H_SF4′ then
                        atoms ← chain.get_atoms()
                        search ← NeighborSearch(list(atoms))
                        sf4 ← list(res.get_atoms())
                        for a ∈ sf4 do
                            neigh+ = len(search.search(a.get_coord(), cutoff))
                            reslist ←
                              Selection.unfold_entities(search.search(a.get_coord(), cutoff),′ R′)
                            reslist_s ← sorted(reslist)
                        for r ∈ chain do
                            if r.id[0] ∉ {′H_SF4′,′ HEM′,′ W′} and r.id[0][0 : 2] ≠′ H_′ and
                              r.id[0] ≠″ and r ∈ reslist then
                                i++; reslist_s.append(r)
                        for r ∈ chain do
                            if reslist_s[0].id[1] ≤ r.id[1] ≤ reslist_s[−1].id[1] then
                                if r ∈ reslist then
                                    seq+ = Polypeptide.three_to_one(r.get_resname())
                                    nei+ =′ 1′
                                else if r.id[0] ∉ {′H_SF4′,′ HEM′,′ W′} and r.id[0][0 : 2] ≠′ H_′
                                  and r.id[0] ≠″ then
                                    seq+ = Polypeptide.three_to_one(r.get_resname())
                                    nei+ =′ 0′
                        results ← [structure.id + chain.id + str(len(chain)), seq, nei]
                        if file exists then
                            for r ∈ results do
                                file.write(r + '\n')
                            file.open("filename.txt", "r"); print(file.read())
                        seq, nei, status ←″,″ ,″
```

Cet algorithme parcourt toutes les chaînes polypeptidiques et pour chaque acide aminé de la chaîne nous retourne le chiffre 1 si l'acide aminé est dans le voisinage du cluster SF4 et 0 si l'acide aminé ne se situe pas dans le voisinage du cluster comme illustré dans la figure 2.5. Les acides aminés que nous choisissons sont ceux compris entre le premier et le dernier acide aminé voisin. À la fin de l'extraction le jeu de données se présente comme dans la figure 2.6 ci-dessous.

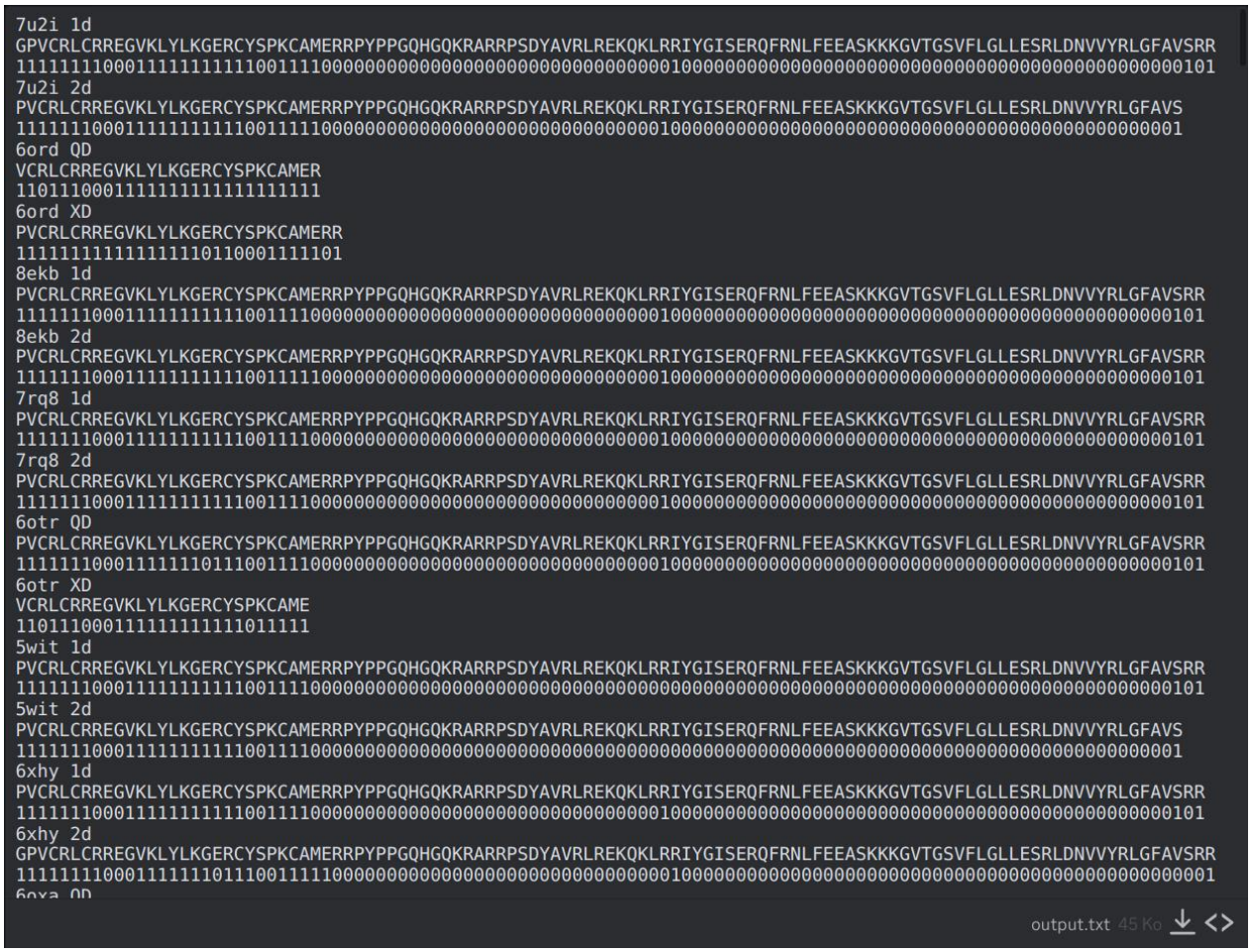

Figure 2.6 Figure montrant le résultat issu de la préparation du jeu de données.

Il faut noter que le nombre total de séquences récupérées est de 232.

Après récupération dans les chaînes des différentes structures de tous les acides aminés dans le voisinage du cluster nous récupérons dans les chaînes la proportion des résidus qui n'étaient pas

du tout dans le voisinage du cube. La figure 2.7 nous illustre clairement ce dont nous parlons.

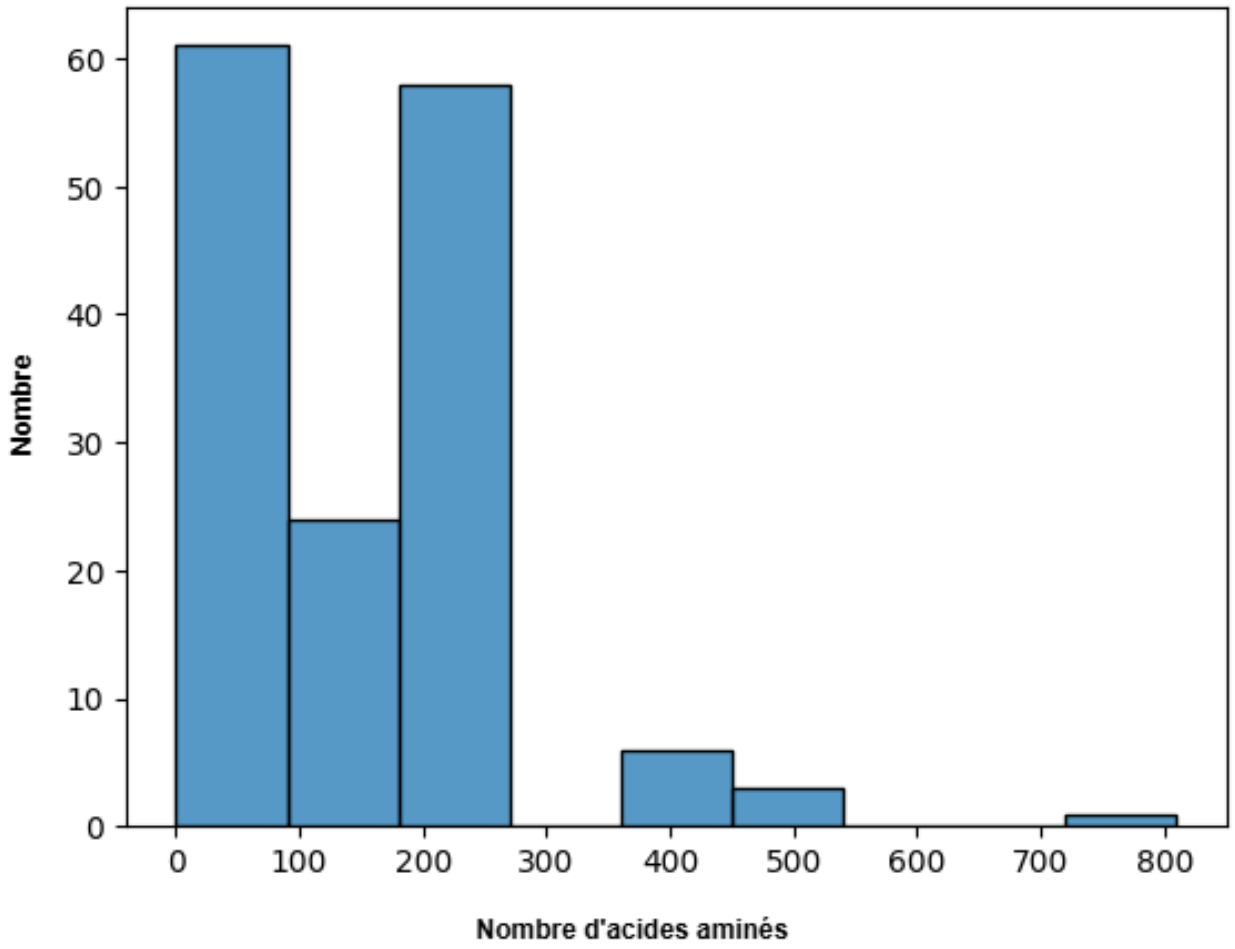

Figure 2.7 Figure montrant la proportion des acides aminés qui ne sont pas dans le voisinage du voisinage du cluter SF4, c'est-à dire à plus de 9 Angstrom autour du cluster SF4.

2.5 Récupération des groupes intéressants

Dans cette partie, l'objectif est d'accélérer les analyses en conservant les séquences représentatives par groupes et en évitant les redondances. Pour ce faire, nous avons eu recours à CD-HIT pour clusteriser les séquences issues de l'extraction faite précédemment sur les acides aminés dans le voisinage ou non du cluster partant du premierau dernier acide aminé dans le voisinage. Cet outil permet de diminuer la redondance en organisant les séquences en groupes ou clusters, selon un

seuil de similarité spécifié.

Avant tout, il est important de rappeler que CD-HIT est un outil conçu pour organiser de grandes bases de données de séquences afin de diminuer les redondances. Il utilise une méthode de clustering basée sur la similarité pour regrouper les séquences similaires selon un seuil prédéfini. Ainsi, cette approche réduit le volume de données à analyser, ce qui représente un avantage considérable pour la tâche à accomplir.

La commande qui nous a permis de pouvoir réaliser cette tache est :

```
./cd-hit -i /home/ades/Bureau/SF4ligand/readyforClustal.fasta
-o /home/ades/Bureau/SF4ligand/cluster.fasta -n 5 -c 0.99 -S 30
```

Celà étant fait, il convient néanmoins de définir les différents paramètres de cette commande pour pouvoir mieux comprendre ce que nous demandons à CD-HIT de faire.

`./cd-hit -i /home/ades/Bureau/SF4ligand/readyforClustal.fasta` : cette partie spécifie le fichier d'entrée et le chemin vers le fichier que nous apssons en entrée à CD-HIT en l'occurence ***readyforClustal.fasta***.

`-o /home/ades/Bureau/SF4ligand/cluster.fasta` : cette partie de la commande spécifie le fichier de sortie et le chemin vers le fichier de sortie en question. Ici le fichier de sortie qui est le fichier qui contient les résultats notamment ***cluster.fasta***.

`-n 5` : spécifie la taille du mot pour la comparaison des séquences. Dans notre cas, la taille du mot est défini à 5.

`-c 0.99` : sert à définir le seuil d'identité entre les séquences pour les regrouper dans le même

cluster. La valeur de 0,99 signifie que les séquences doivent être identiques à 99% pour être regroupées ensemble.

`-S 30` : spécifie le seuil de longueur d'alignement avec lequel les séquences seront regroupées. Seules les séquences avec au moins 30% de leur longueur alignée seront considérées pour le clustering.

Pour ce travail, nous avons passé en entrée 242 séquences, séquences que nous avons obtenues de notre extraction précédente. Après clustering nous obtenons 4 séquences qui sont représentatives et que nous utilisons dans la suite de ce travail.

```
>2WS3B
MRLEFSIYRYNPDVDDAPRMQDYTLEADEGRDMMLLDALIQLKEKDPSLSFRRSCREGVCGSDGLNMNGKNGLACITPISALN
QPGKKIVIRPLPGLPVIRDLVVDMGQFYAQYEKIKPYLLNNGQNPPAREHLQMPEQREKLDGLYECILCACCSTSCPSFWWNP
DKFIGPAGLLAAYRFLIDSRDTETDSRLDGLSDAFSVFRCHSIMNCVSVCPKGLNPTRAIGHIKSMLLQRNA

>6GSK32
GRYIGPVCRLCRREGVKLYLKGERCYSPKCAMERRPYPPGQHGQKRARRPSDYAVRLREKQKLRRIYGISERQFRNLFEEASK
KKGVTGSVFLGLLESRLDNVVYRLGFAVSRRQARQLVRHGHITVNGRRVDLPSYRVRPGDEIAVAEKSRNLELIRQNLEAMKG
RKVGPWLSLDVEGMKGKFLRLPDREDLALPVNEQLVIEFYSR

>1L0VB
AEMKNLKIEVVRYNPEVDTAPHSAFYEVPYDATTSLLDALGYIKDNLAPDLSYRWSCRMAICGSCGMMVNNVPKLACKTFLR
DYTDGMKVEALANFPIERDLVVDMTHFIESLEAIKPYIIGNSRTADQGTNIQTPAQMAKYHQFSGCINCGLCYAACPQFGLN
PEFIGPAAITLAHRYNEDSRDHGKKERMAQLNSQNGVWSCTFVGYCSEVCPKHVDPAAAIQQGKVESSKDFLIATLKPR

>3CB8A
```

VIGRIHSFESCGTVDGPGIRFITFFQGCLMRCLYCHNRDTWDTHGGKEVTVEDLMKEVVTYRHFMNASGGGVTASGGEAILQ
AEFVRDWFRACKKEGIHTCLDTNGFVRRYDPVIDELLEVTDLVMLDLKQMNDEIHQNLVGVSNHRTLEFAKYLANKNVKVWI
RYVVVPGWSDDDDSAHRLGEFTRDMGNVEKIELLPYHELGKHKWVAMGEEYKLDGVKPPKKETMERVKGILEQYGHKVMF

2.6 Classification en groupes des protéines avec les cubes

Cette partie décrit le processus de classification des protéines que nous avons en groupes. Une fois le fichier ci-dessus reconstitué, il a été soumis au logiciel ClustalW. En effet, ClustalW est un outil de bio-informatique utilisé pour l'alignement de séquences multiples de protéines, d'ARN et d'ADN. C'est un programme parmi l'ensemble des programmes d'alignements Clustal. Il est capable d'aligner des séquences rapidement et avec précision en utilisant des arbres guides ensemencés et des techniques de profil-profil HMM (Hidden Markov Models), le profil HMM étant un modèle probabiliste qui représente les variations d'un alignement multiple (insertions, délétions, substitutions) via une machine de Markov cachée.

Ainsi, nous comparons et analysons nos séquences et ce dans le but de pouvoir comprendre les relations entre elles.

En ce qui concerne l'alignement et l'arbre phylogénétique, nous avons un script qui nous permet d'effectuer cette tâche de manière assez rapide et efficace. Nous l'avons nommé **Clustal Alignment and Tree Generation**.

Cet algorithme s'occupe de recevoir en entrée les séquences formatées et d'en ressortir un alignement multiple de séquences et un arbre phylogénétique. Les lignes suivantes présentent l'alignement de séquences 2.9 ainsi que l'arbre phylogénétique 2.8 qui découlent de cet algorithme.

Script 6: Clustal Alignment and Tree Generation

```
Procedure Main() :
    input_file ← '../clustered_containers.txt'
    output_alignment ← '../alignseq.aln'
    output_tree ← '../alignseqtree.ph'
    DoClustal(input_file, output_alignment, output_tree)
Function DoClustal(input_file, output_alignment, output_tree) :
    command ←
      ['clustalw', '-infile=' + input_file, '-tree', '-align', '-matrix=BLOSUM', '-newtree=' +
      output_tree, '-outfile=' + output_alignment]
    run(command)
Main()
```

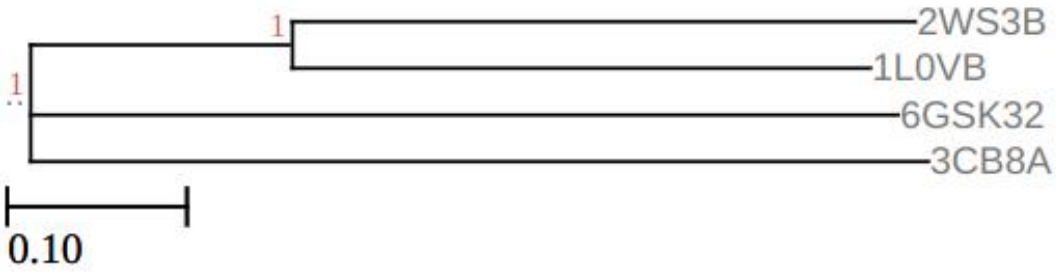

Figure 2.8 Figure montrant l'arbre phylogénétique associé aux séquences résultant de CD-HIT.

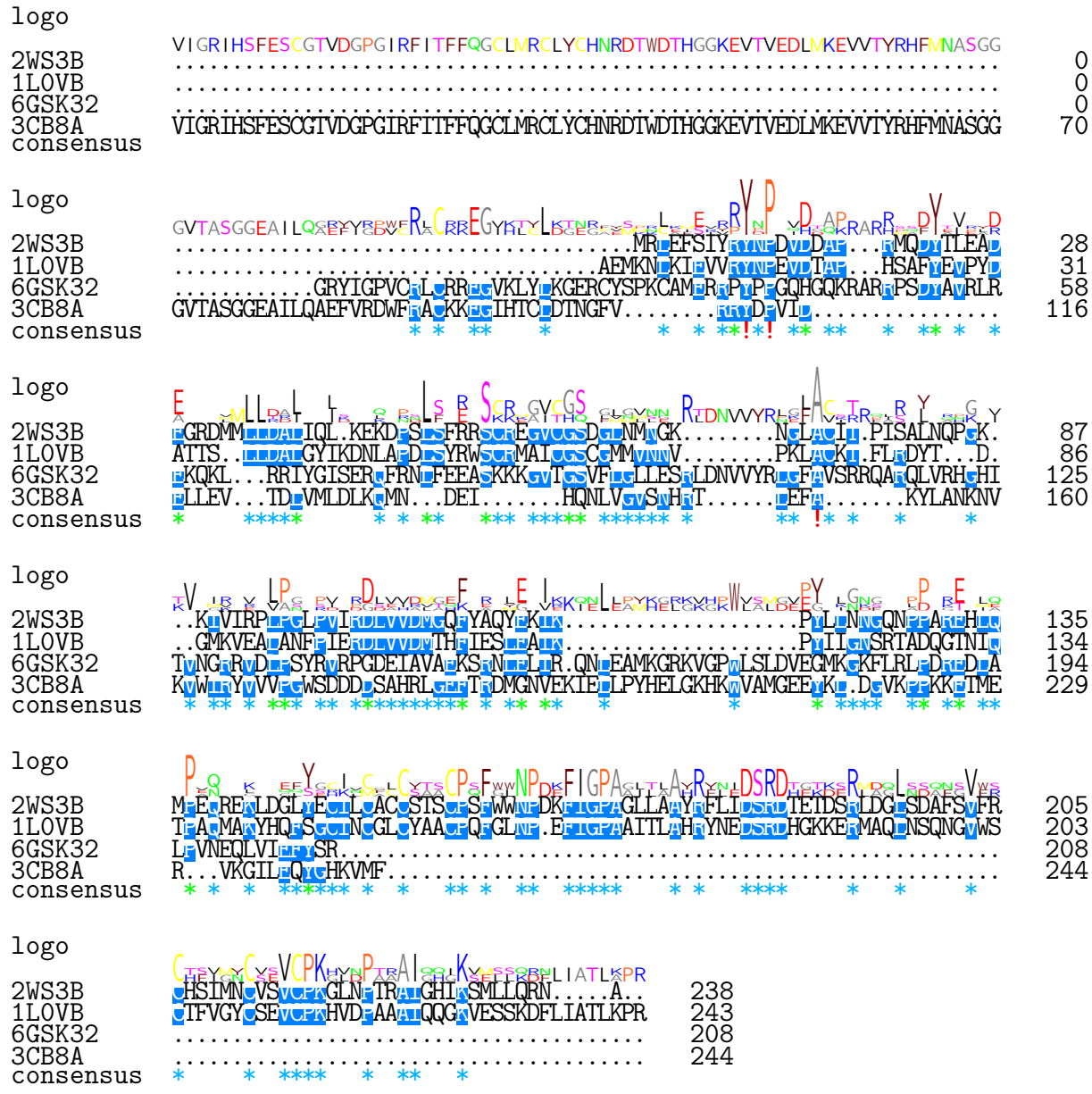

Figure 2.9 Figure montrant l'alignement multiple des séquences résultant de CD-HIT

2.7 Recherche des motifs dans les groupes

Dans la section précédente, nous avons analysé les quatre séquences obtenues à l'aide de CD-HIT pour regrouper les séquences et ainsi éviter les redondances. Les résultats de cette analyse ont été ensuite utilisés comme entrée pour MEME, dans le but de découvrir les motifs récurrents

parmi les quatre séquences. Nous partons du principe que le motif qui capture le cluster doit être proche du cluster. L'objectif ici est donc de repérer les motifs pertinents parmi ces séquences afin de distinguer les acides aminés proches du cluster de ceux qui ne le sont pas, similaire à ce qui a été fait dans la section 2.3. Cette démarche vise à associer les parties des motifs proches du cluster à nos protéines d'intérê. Le résultat est défini par la figure 2.10 ci-après :

Figure 2.10 Figure montrant la combinaison de l'alignement des séquences de ClustalW et la recherche des motifs.

Après avoir concaténé l'alignement des séquences obtenu via CLUSTAL avec les résultats de recherche de motifs conservés générés par MEME, un motif distinctif a pu être clairement identifié. Ce motif est présent dans la chaîne B de la structure 2WS3 (Ruprecht *et al.*, 2009), où il correspond à la succinate déshydrogénase, ainsi que dans la chaîne B de la structure 1L0V, correspondant cette fois à la fumarate réductase (Iverson *et al.*, 2002).

La succinate déshydrogénase (SDH), aussi appelée succinate :quinone oxydoréductase (SQR), est une enzyme clé du métabolisme cellulaire, jouant un rôle à la fois dans le cycle de Krebs et dans la chaîne respiratoire aérobie, où elle est exprimée en présence d'oxygène.

De son côté, la fumarate réductase (QFR), ou quinol :fumarate oxydoréductase, est une enzyme essentielle du métabolisme anaérobie. Elle intervient dans la chaîne de transport des électrons en catalysant la réduction du fumarate en succinate, avec le fumarate agissant comme accepteur final d'électrons en absence d'oxygène.

Ainsi, la SDH est majoritairement exprimée en milieu aérobie, tandis que la QFR est exprimée en milieu anaérobie, jouant un rôle vital dans la production d'énergie chez les organismes anaérobies.

Dans le cadre des expériences de chimie, il est préférable d'utiliser la fumarate réductase — en particulier celle présente dans la chaîne B de la structure 1L0V car, contrairement à la SDH, elle ne réagit pas avec l'oxygène, ce qui constitue un avantage pour la réalisation d'expériences en conditions anaérobies contrôlées.

2.8 Design de séquences alternatives

Cette section constitue une étape de validation in silico visant à évaluer le comportement de notre protéine d'intérêt au sein d'un pipeline de modélisation structurale. L'objectif est de tester la compatibilité et la robustesse des outils AlphaFold et RFdiffusion sur cette séquence protéique, avant toute analyse comparative ou design rationnel ultérieur. Nous disposons d'une séquence d'acides aminés ainsi que d'une structure de référence, et nous commencerons par soumettre cette séquence à AlphaFold afin de générer une prédiction tridimensionnelle. Cette structure prédite sera

ensuite utilisée comme entrée dans RFdiffusion, dans le but de produire des suggestions de design protéique tout en vérifiant que les propriétés structurales clés sont conservées. Un aspect technique spécifique consiste à contraindre la prédiction d'AlphaFold de façon à maintenir fixe les acides aminés de Cystéine noté C. Pour cela, nous avons développé un script *get_Residue_Position* permettant d'identifier les résidus concernés, afin de passer leurs positions comme contraintes explicites lors de la modélisation. Cette approche permet de préserver l'intégrité structurale de la région d'intérêt au cours du processus de prédiction.

La fonction qui nous permet donc de faire réaliser cette tâche est get_Residue_Position.

Script 7: get_Residue_Position

Data: sequence

```
1 cys ← "C";
2 for i in range(0, len(sequence) do
3     if sequence[i] == cys then
4         res ← i+1 ;
5         print("Residue  is present at ".format(c, str(res)));
6     if res == NONE then
7         print("No residue CYS found");
```

Cet algorithme permet juste de fouiller dans la séquence de la protéine d'intérêt donnée et de récupérer la position de tous les acides aminés de cystéine "C" contenus dans la séquence pour ensuite pouvoir empêcher la mutation lors du Design avec RFdiffusion.

La séquence de notre protéine d'intérêt est la suivante :

QELQRIAEAAERCWRQCQQLSEKTSNPEKKQALQESAEESLRFAQKGSVSPQEFVEEARACAQRCQRLSEQTSNPEKKQSLER
EAQESQAFAQSLEQAAAAALEHHHHHH

2.9 Combinaison de la recherche des motifs dans les groupes et du design de séquences alternatives

Cette section fait suite aux analyses présentées dans les parties 2.7 et 2.8, et propose de combiner les approches développées précédemment afin d'explorer de nouvelles configurations structurales. Maintenant que la validité de notre structure d'intérêt a été établie à l'aide d'AlphaFold et de RFdiffusion, nous cherchons à tirer parti des résultats obtenus dans ces deux étapes.

Pour ce faire, nous allons concaténer les motifs identifiés lors du clustering présenté en section 2.7 avec la séquence de notre protéine d'intérêt, dans le but d'analyser les effets de cette fusion sur la structure finale.

L'examen des motifs extraits montre que ceux-ci contiennent des résidus positionnés à des distances variables par rapport au cluster SF4. Pour cette étape, nous choisissons de nous concentrer uniquement sur les résidus situés à proximité du cluster métallique, en excluant ceux qui s'en éloignent. Concrètement, nous conservons les résidus du motif jusqu'au premier résidu identifié comme étant spatialement éloigné du cluster.

Il est également important de préciser que le motif utilisé pour cette concaténation provient de la chaîne B de la **quinol-fumarate réductase (structure PDB 1L0V)**, qui sert ici de structure de référence pour l'extraction des éléments à intégrer.

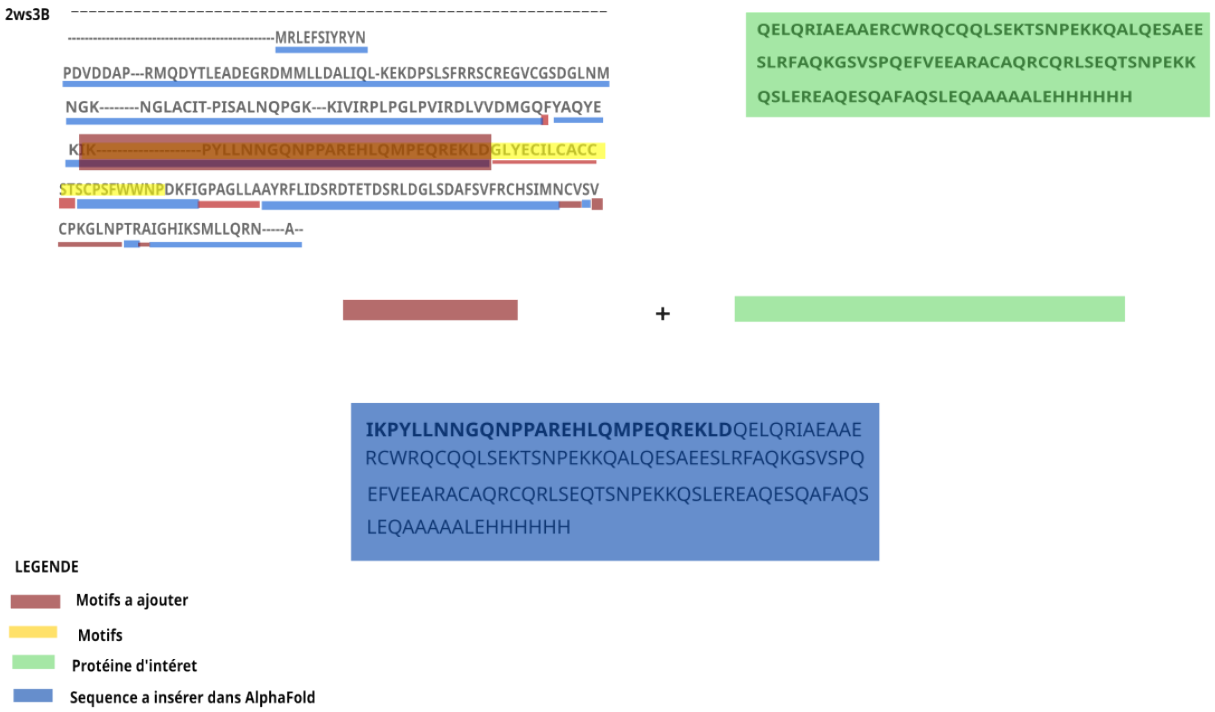

Figure 2.11 Figure montrant la concaténation de la protéine d'intérêt avec le motif.

Cette figure illustre l'intégration du motif à séquence de la protéine CCIS1. Cette combinaison de séquence ainsi obtenue est ensuite passée dans AlphaFold2 pour la prédiction.

CHAPITRE 3
RÉSULTATS

Dans cette partie nous discuterons des résultats obtenus suite aux expérimentations des séquences faites avec les algorithmes d'intelligence artificielle de la partie précédente. Nous mettrons l'accent sur l'analyse des résultats et aussi sur ce à quoi nous aboutissons notamment ce sur quoi les résultats se sont avérés concluants ou pas. Notons que ces résultats sont obtenus après avoir passé en entrée nos séquences à deux algorithmes d'intelligence artificielle : l'un pour la prédiction de structure de protéine nommé AlphaFold et l'autre pour le design de structure nommé RFDiffusion. Un des grands défis du problème auquel nous faisons face c'est que notre protéine d'intérêt n'a pas de vrai modèle expérimental de sa structure. Ce qui est pour le moins rassurant c'est que nous avons un modèle informatique validé par des experts. (Grzyb *et al.*, 2010). Nous voulons voir si le modèle informatique sur lequel nous nous basons répond à ce que nous observons sur la figure de l'article. Pour ce faire, nous allons soumettre la séquence de la protéine d'intérêt à AlphaFold.

3.1 Prédiction de la protéine d'intérêt avec AlphaFold2

Pour rappel dans cette partie, nous avons utilisé le CCIS1 pour voir si AlphaFold prédit correctement la structure à partir de sa séquence. Nous pouvons le dire avec certitude que AlphaFold marche correctement dessus et que si nous avons la séquence de la protéine d'intérêt nous retrouvons bel et bien la structure et les propriétés qui y sont liées.
Dans cette partie, nous allons analyser les résultats de AlphaFold, ensuite comparer les modèles que nous avons avec le modèle informatique que nous avons tout en ayant une attention particulière sur les résidus de cystéines noté en C.

AlphaFold prédit 5 structures (figure 3.1) lorsque des séquences lui sont soumises.

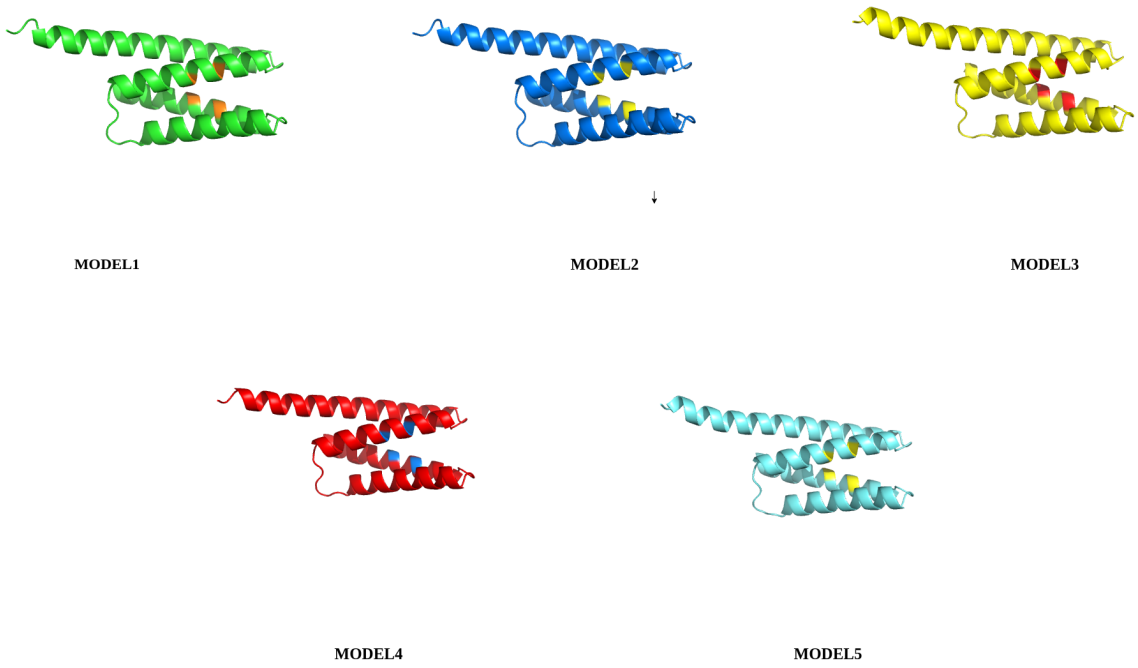

Figure 3.1 Figure montrant les résultats AlphaFold du CCIS1.

Les figures 3.1 et 3.2 correspond à la prédiction de la structure tridimensionnelle de la protéine d'intérêt CCIS1 à l'aide de AlphaFold. Cette structure prédite sert ensuite d'entrée pour l'outil RF-diffusion, dans le cadre d'un processus de conception de nouvelles structures protéiques. C'est à cette étape que les résidus de cystéine (C) sont fixés, de manière à en préserver la position et le rôle fonctionnel au cours de la génération des structures.

RFdiffusion propose initialement un ensemble de huit structures candidates issues du processus de design. Parmi ces structures, une sélection est effectuée en s'appuyant sur plusieurs critères d'évaluation, tels que le score énergétique, la valeur de RMSD, les résultats de simulations de dynamique moléculaire, ainsi que, le cas échéant, des éléments de validation expérimentale. La structure présentant les performances les plus satisfaisantes selon l'ensemble de ces critères est alors retenue comme conception finale (figure 3.3), et constitue le résultat principal de cette étape d'optimisation in silico.

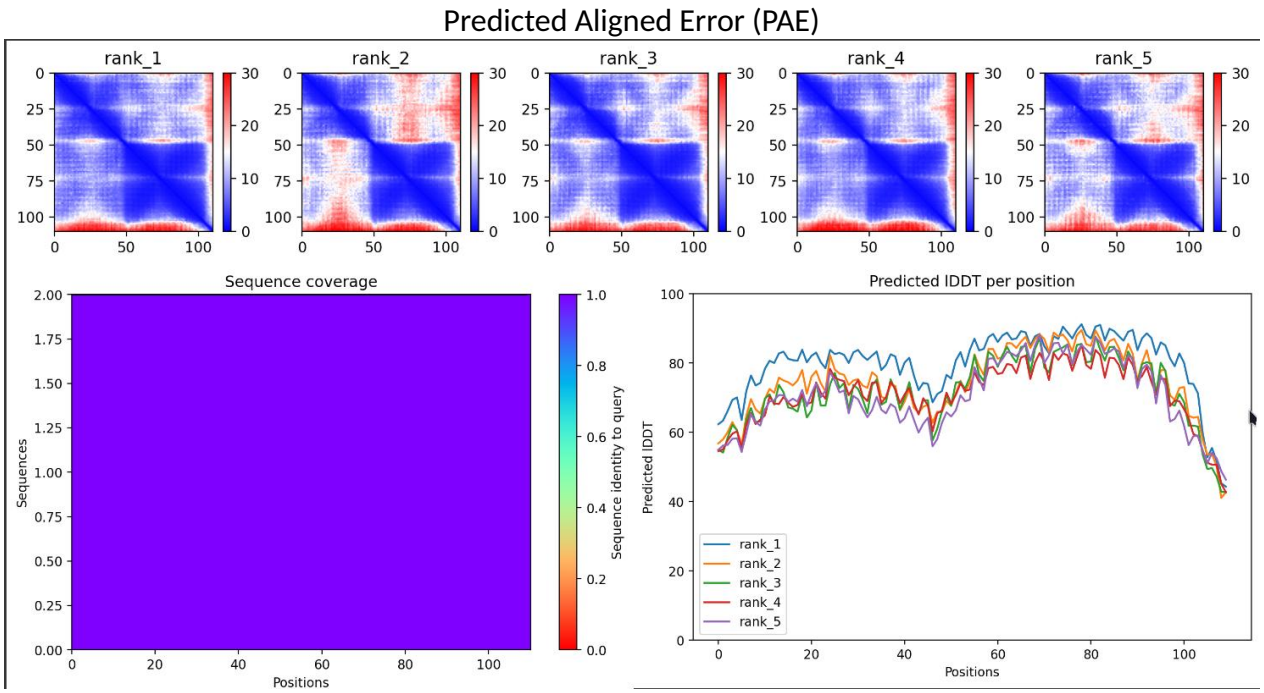

Figure 3.2 Figure montrant le score pLDDT, la couverture séquentielle ainsi que l'erreur d'alignement prédite d'AlphaFold2 pour le CCIS1

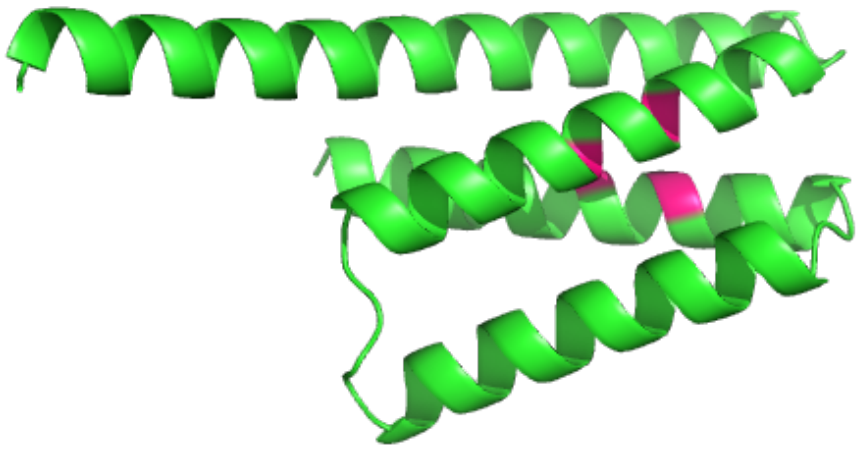

Figure 3.3 Figure montrant le meilleur résultat de RFdiffusion pour le CCIS1.

3.1.1 Les différentes métriques utilisées

3.1.1.1 Prédictions avec AlphaFold

Les métriques produites par AlphaFold revêtent une importance essentielle pour l'évaluation de la qualité des structures protéiques prédites, en fournissant des indicateurs quantitatifs permettant d'interpréter la fiabilité et la pertinence des modèles structuraux générés.

Le score pLDDT (Predicted Local Distance Difference Test) :Les scores pLDDT prédits montrent

comment la prédiction de la struture de protéines varie en fonction des positions.
L'erreur d'alignement prédite (PAE - Predicted Aligned Error) : les matrices PAE offrent une représentation visuelle de la confiance dans les alignements résidu-résidu pour diverses structures protéiques.
La couverture séquentielle : La couverture séquentielle se définit par la portion de la séquence d'acides aminés pour laquelle AlphaFold a pu faire une prédiction structurale.

3.1.1.2 Design avec RFdiffusion

Les métriques intégrées à RFdiffusion servent à encadrer et valider la génération de structures protéiques en vérifiant qu'elles respectent les contraintes biologiques et physicochimiques, tout en garantissant leur cohérence structurale, leur diversité et leur capacité à se replier correctement.

ProteinMPNN : Valeur du score du réseau neuronal de protéines (ProteinMPNN score). ProteinMPNN (Dauparas *et al.*, 2022), est une approche novatrice utilisant l'apprentissage profond pour concevoir des séquences de protéines.
Elle permet de créer des séquences fiables et exactes, capables d'adopter des structures tridimensionnelles spécifiques, offrant ainsi de nouvelles opportunités pour des applications dans les domaines de la biotechnologie et de la biomédecine. Une valeur plus basse indique une meilleure performance.
pLDDT : (Predicted Local Distance Difference Test), cette méthode est utilisée pour évaluer la précision des modèles de structure protéique prédits. Elle mesure les différences de distance entre les positions des acides aminés dans la structure prédite par rapport à une structure expérimentale. Plus la valeur de cette métrique est faible, plus la précision du modèle est grande.
PTM : (predicted TM-score) (Xu et Zhang, 2010), mesure la similarité entre la prédiction et la structure expérimentale. Avec cette métrique plus le score est élevé et meilleur c'est.
PAE :Erreur de distance moyenne prédite (predicted aligned error), plus elle est basse, mieux c'est.
RMSD :moyenne quadratique de déviation (root-mean-square deviation), mesurant la différence entre la position des atomes dans la structure prédite et la structure de référence. Plus cette valeur est basse, plus la prédiction est précise.

3.1.2 Analyse du résultat (score LDDT, figure 3.2)

Dans la figure la partie inférieure droite indique le score LDDT (initial distance deviation test) prédit par position des cinq modèles de structures de protéines différentes classées rank_1 à rank_5. Ainsi pour pour le rank_1 nous avons le modèle 3, pour rank_2 nous avons le modèle 4, pour rank_3 nous avons le modèle 2, pour rank_4 nous avons le modèle 5 et finalement pour rank_4 nous avons le modèle 1. Voici ce que nous observons sur cette figure 3.2 :

- Chaque position sur l'axe X correspond à un résidu spécifique dans la séquence de la protéine.
- L'axe des Y représente le score LDDT prédit, sur une échelle de 0 à 100. Le score LDDT est une mesure de la confiance dans la prédiction de la structure pour chaque résidu, avec des valeurs plus élevées indiquant une plus grande confiance.
- rank_1 à rank_5 : Les courbes de différentes couleurs notamment bleu, orange, vert, rouge et violet correspondent aux structures classées de la première à la cinquième en termes de score de confiance global.

3.1.3 Interprétation du résultat(score LDDT figure 3.2)

- Les scores sont généralement plus élevés (autour de 70-90) pour la majorité des résidus, ce qui indique une confiance relativement élevée dans les prédictions de ces régions. Les scores baissent relativement (entre 40 et 50), ce qui indique une incertitude relativement moyenne dans la prédiction des structures.

  Les scores chutent aux extrémités (positions proches de 0 et >100), suggérant une plus grande incertitude dans la prédiction des structures aux extrémités de la séquence.
- **Les rangs** : Rank_1 notifié en bleu présente les scores LDDT les plus élevés suivant toutes les positions de la séquence de protéine, ce qui indique un meilleur degré de fiabilité par rapport aux autres structures prédites.
  Rank_2 à rank_5 : Ces courbes affichent des scores légèrement inférieurs à ceux de "rank_1", mais suivent une tendance globale similaire. Cela pourrait indiquer que les structures sont assez semblables, avec seulement quelques variations mineures.

La partie supérieure présente la matrice de l'erreur d'alignement prédite (PAE - Predicted Aligned Error) des structures prédites classées rank_1 à rank_5. Les axes X et Y représentent les positions des résidus sur toute la structure de protéine et chaque pixel représente l'erreur prédite pour une paire de résidus (i,j). En outre cette figure contient une échelle de couleur allant de 0 à 30Å avec la couleur bleue pour 0Å ce qui indique qu'il n'y a pas d'erreur dans l'alignement des résidus et la couleur rouge pour 30Å ce qui indique qu'il y a eu beaucoup d'erreur dans l'alignement des résidus.

3.1.4 Analyse de la figure (Erreur d'alignement prédite PAE)

D'un point de vue général, on constate que les diagonales sont bleues, ce qui est normal étant donné que les résidus sont alignés avec eux-mêmes.

Rank_1 : La majorité de la matrice est bleue, ce qui veut dire que l'erreur prédite est donc faible et alors que les résidus sont bien alignés. Néamoins quelques zones rouges aux extrémités nous

dénote d'une incertitude plus élevée dans ces zones.

Rank_2 à Rank_5 : Ces matrices montrent que la zone diagonale est en blue et qu'il y a des différences subtiles de rouges surtout en ce qui concerne les parties non diagonales. Cela est signe de légères différences en ce qui concerne la prédiction des alignements entre les structures.

3.1.5 Interprétation du résultat (Erreur d'alignement prédite PAE)

Nous voyons clairement que la structure fiable rank_1 qui est celle avec la plus faible erreur prédite. Ce qui confirme les résultats de la figure antérieure (celle avec les scores LDDT) où nous avons observé que son score LDDT était plus élevé par rapport aux 4 autres.

rank_2 à rank_5 sont toutes aussi bien prédites seulement avec relativement plus d'erreurs les uns les autres par rapport à rank_1.

Si nous combinons les différents résultats de cette figure de rank_1 à rank_5, nous sommes à même de voir de par les subtiles différences de couleurs aux niveaux des structures prédites certaines de ces régions où la prédiction d'une structure est faible par rapport à une autre.

La structure de rang 1 se distingue comme étant la plus fiable, suivie de près par les autres, avec de légères variations dans les prédictions d'alignement. En combinant ces données avec les scores LDDT, on peut effectuer une évaluation exhaustive de la qualité des structures prédites générées.

La figure de la partie inférieure gauche quant à elle fait référence à une carte de couverture des séquences qui montre l'identité des séquences alignées par rapport à la séquence de référence. Les couleurs indiquent l'identité de la séquence par rapport à la séquence de référence sur une échelle de 0 à 1 avec pour 0 la couleur rouge pour indiquer que les séquences sont différentes et 1 pour indiquer qu'elles sont identiques.

3.1.6 Analyse et Interprétation de la couverture séquentielle

Le tableau 3.1 est un tableau comparatif du modèle informatique de la protéine d'intérêt et des résultats obtenus au cours de la prédiction avec AlphaFold.

Protéine	Longueur helice	Helice 2	Helice 3	Helice 4	position C bonne place
Modèle CCIS1	21	19	19	33	oui
Rank 1 (Model 3)	21	18	20	34	oui
Rank 2 (Model 4)	21	18	20	33	oui
Rank 3 (Model 2)	21	18	20	32	oui
Rank 4 (Model 5)	21	18	20	34	oui
Rank 5 (Model 1)	21	18	20	32	oui

Table 3.1 Tableau comparatif de la protéine d'intéret et des modèles prédits par AlphaFold

Nous voyons clairement dans la figure 3.2 que la matrice faisant office de couverture séquentielle est coloriée en violet en ce qui concerne toutes les positions de la séquence de protéine, ce qui voudrait dire que toutes les séquences alignées sont identiques ou presque identiques à la séquence de référence.

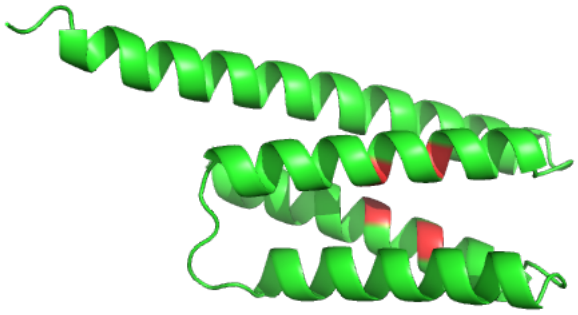

Figure 3.4 Figure montrant la meilleure prédiction dans AlphaFold2 : Model3

Comme dit précédemment, le Model3 est celui prédit avec le plus de précision, et ce dans tous les métriques que nous avons eu à voir.

3.2 RFdiffusion produit de nouvelles séquences avec la protéine d'intérêt

Dans cette partie nous devrions partir d'une structure de protéine pour commencer les expérimentations. Il n'y a aucune expérience existante pour la protéine d'intérêt.
Nous avons obtenu les résultats de AlphaFold de la protéine d'intérêt. Nous avons pris ensuite le résultat de la meilleure prédiction dans AlphaFold et nous l'avons passé en entrée à RFdiffusion pour le design de la protéine. Notons aussi qu'en passant ledit résultat dans RFdiffusion nous avons eu à empêcher la mutation des résidus de Cystéine "C". La commande qui nous a permis de faire cette manipulation est la suivante :

```
Contigs: 'A13':'A17':'A61':'A65'
```

RFdiffusion nous produit en retour 8 design parmi lesquels il choisit le meilleur sur base de certaines mesures de qualité et de confiance définies dans le tableau et la figure ci-après :

Modélisation N°	MPNN	pLDDT	PTM	PAE	RMSD	Séquence prédite
0	1.254	0.876	0.666	5.493	1.610	Data
1	1.159	0.903	0.718	4.698	1.008	Data
2	1.156	0.883	0.713	5.315	1.873	Data
3	1.227	0.889	0.735	4.923	1.546	Data
4	1.168	0.880	0.711	5.070	1.614	Data
5	1.194	0.891	0.736	4.773	1.842	Data
6	1.171	0.859	0.661	6.127	1.614	Data
7	1.157	0.922	0.758	3.720	1.421	Data

Table 3.2 Tableau des différents design de la séquence alternative dans RFdiffusion

Le tableau suivi des séquences font référence aux résultats que nous obtenons lors du passage en entrée du meilleur résultat obtenu de AlphaFold dans RFdiffusion.

3.2.0.1 Analyse du résultat

Nous avons eu dans ce cas à générer plusieurs séquences pour un seul design, le design 0. Nous essaierons d'analyser et interpréter ces résultats.

Pour les 8 séquences du Design que nous avons, nous voyons que la séquence N°7 semble être la meilleure séquence compte tenu des meilleurs scores de PLDDT, PTM et PAE, quoique la séquence N°1 soit notable pour avoir eu le meilleur score RMSD. Le résultat du meilleur design est présenté par la figure 3.6 ci-après :

>Séquence DesignN°0

SMREKLLRALKKCKELCEKALKVEKNPEVKARLQEAIEKIQALLDDPSATLEEYIEALKKCKELCEELREELLKVLSKEEIDK

LIEEIEELIEEALEAQRREAERAAALA

>Séquence DesignN°1

SLREELLAALKECAELCKKALKVEKDKEIKEELEKLIEKLEAVLADPNATEEELIAALKECAELCKKYKKEFLKVLSEEEINA

LIERLEALIERAERERREERRRLEEELE

>Séquence DesignN°2

HMREKLIEALKKCRELCEKALEKEENEEIKKKLKELIEEIEKLLEDPNATLEEYIEALKKCRDLCKELREELLVVLSQEEIDAL

IEELEELIEEAEKELKERKELAEKLA

>Séquence DesignN°3

HMREKLLEALEKCRDLCEKALEKETNPEIKARLQAEIERLTALLADPDATLEELIEALEKCRALCEELAEELLVVISQEEIDEL

IEELEELIEEARKKIEEEEKAREEAR

>Séquence DesignN°4

SMREKLLKALKECKELCEEALKEETNEEIKKKLQEKIESITKLLEDPNATLEELIKALKECKDLCEELKEEFKKVKSEEEIEKL

IEEIEKLIEEAEREQAERAAAAAAAA

>Séquence DesignN°5

SMMEELREALEECRDLCEKALEKETDPEVKAKLQALIEKITALLEDPNATLEEAIEALEECRELCEELKEELKVVLSEEEIEEL

IEEIEELIEEAKEELEREKREAEELA

>Séquence DesignN°6

SLEEELKKALQECKELCEKALKKEKNKEIKKELEELIEKIKKLLEDKNATLEEYIKALKECKELCKKYKEEFLVVLSKEEIEAL

IKELEELIKKAEEEREKREKELKELA

>Séquence DesignN°7

SMREKLKEALKKCRELCEKALKEEEDEEVKKELEELIEEIEALLADPNATLEEYIEALKKCRDLCEKHKEELLKVLSKEEIDEL

IEEISELIEEAEEAKAAAAAAAAAAA

Figure 3.5 Liste des séquences obtenu dans RFdiffusion du meilleur résultat de AlphaFold.

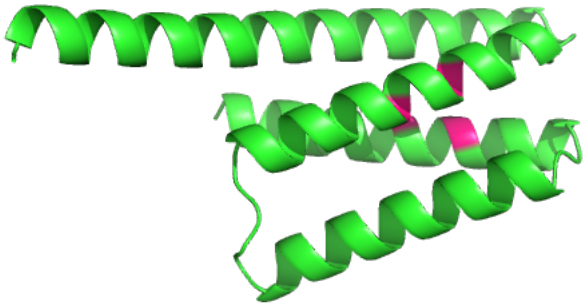

Figure 3.6 Figure montrant le meilleur design de la meilleure prédiction de la combinaison de la séquence alternative et des motifs introduite dans RFdiffusion.

En somme à la fin de ce test nous voyons que le design de la protéine d'intérêt que nous essayons d'étudier marche parfaitement avec la séquence que nous avons comme dans l'article (Grzyb *et al.*, 2010) dans lequel il a été tiré. Cela voudrait dire que tout fonctionne correctement donc nous pourrons ensuite combiner les motifs que nous avons trouvés dans la partie 2.8 avec notre séquence de protéine d'intérêt et voir les résultats qui en découlent.

3.3 Résultats obtenus de la combinaison de la recherche des motifs dans les groupes et du design de séquences alternatives

Nous voulons dans cette section garder les éléments de la structure 3D de la séquence tout en concaténant le motif avec la séquence de la protéine d'intérêt. Vérifions !

3.3.1 Résultats de AlphaFold

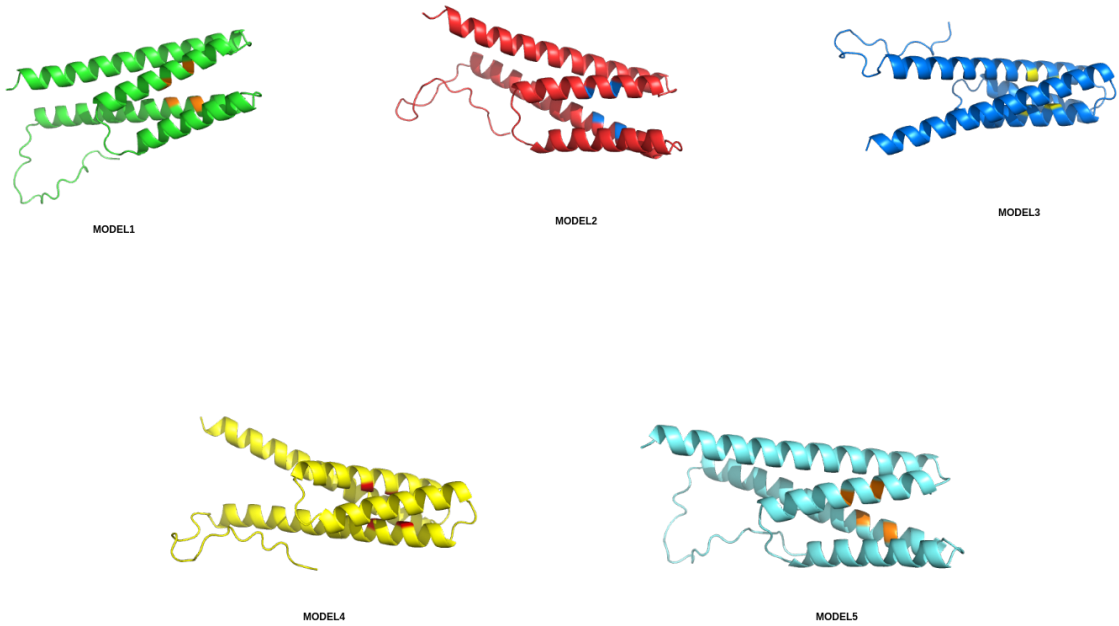

Figure 3.7 Figure montrant les résultats d'AlphaFold pour la protéine d'intérêt concaténée avec le motif.

Ci-dessus nous avons les résultats issus du passage en entrée de la séquence obtenue précédemment par association de motif. Ces résultats les uns après les autres seront placés ensuite en entrée dans RFdiffusion comme fait préalablement, et ce dans le but de faire la conception de cette structure.

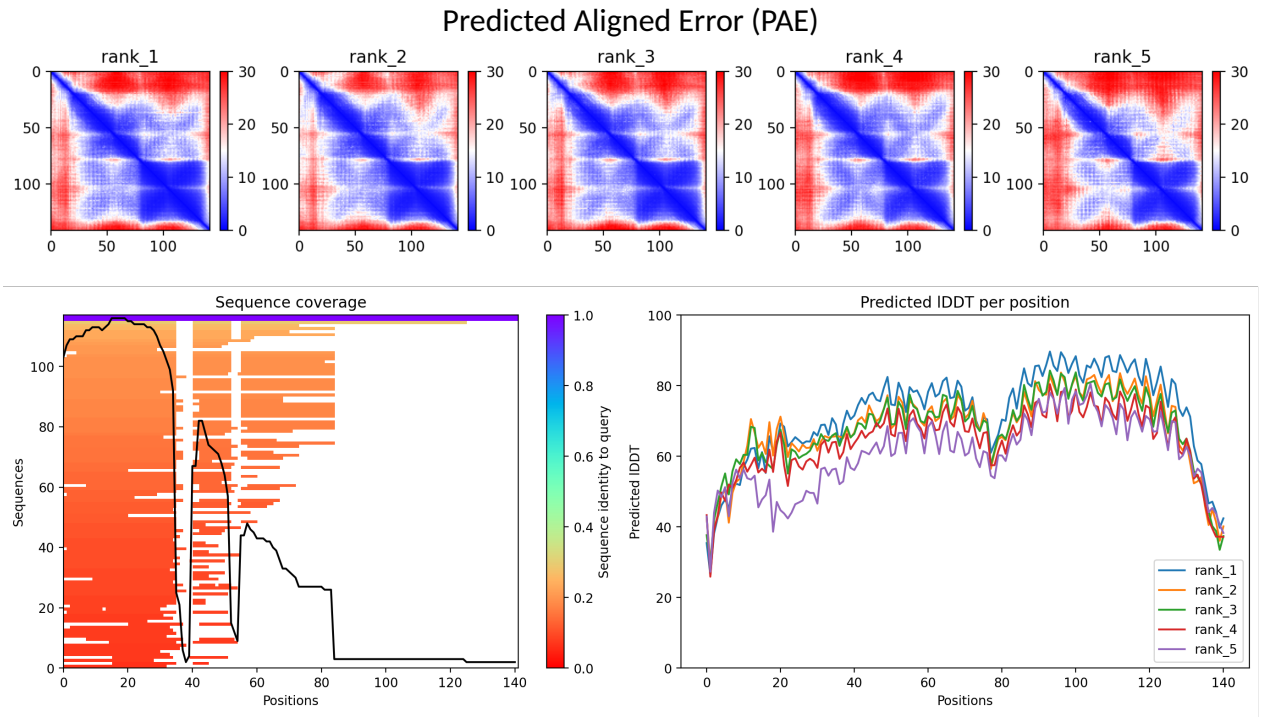

Figure 3.8 Figure montrant les résultats obtenus de la combinaison de la recherche des motifs dans les groupes et du design de séquences alternatives

3.3.2 Analyse du résultat

La figure de la partie inférieure droite nous montre le graphique des valeurs LDDT prédites par position pour 5 classements différents avec sur l'axe X la position des acides aminés le long de la séquence de protéine et sur l'axe Y le score LDDT prédit pour chaque position.

Nous pouvons voir dans un premier temps que les 5 courbes (rank 1 à rank 5) ont une tendance similaire, ce qui nous laisse à suggérer une cohérence en ce qui concerne la confiance dans la précision de la modélisation de la structure.

Nous pouvons aussi voir de légères différences entre les courbes, ce qui nous montre que suivant les différents classements il y a de légères variations dans la confiance de la prédiction.

En outre, dû au fait que la tendance est pratiquement la même et ce pour les différents classements nous pouvons voir qu'au début de la séquence (de la position 0 à la position 20) le score LDDT est faible, ce qui indique une faible confiance dans la prédiction de cette partie.

À partir de la position 20 nous remarquons une augmentation progressive du score LDDT par po-

sition, ce qui indique une augmentation progressive de la confiance de la prédiction dans ces régions.
Partant de la position 120 à la position 140 qui montrent la fin de la séquence, nous observons une diminution progressive du score LDDT, ce qui nous suggère une diminution progressive de la confiance dans la prédiction de cette région.

La figure de la partie inférieure gauche représente la couverture des séquences alignées par rapport à la séquence de référence avec sur l'axe des abscisses les positions des acides aminés le long de la séquence de référence et sur l'axe des ordonnées les séquences qui sont alignées sur la séquence de référence.
La barre colorée comme dans les résultats précédents indique l'identité des séquences alignés par rapport à la séquence de référence avec 0 pour le rouge indiquant une faible identité et 1 pour le violet indiquant une forte identité.
La courbe en noir indique les régions où la séquence est alignée avec la séquence de référence.
La position 0 à 40 indique une couverture élevée avec une forte identité. Cela sous-tend que cette région est bien conservée par rapport aux séquences alignées.
De la position 40 à 60 nous observons une réduction de la couverture des séquences et les identités des séquences variées. Cela suggère une plus grande variabilité ou une région moins conservée parmi les séquences alignées.
Passé la position 60 nous observons une forte diminution de la couverture de la séquence, ce qui nous indique que peu de séquences s'alignent avec la séquence de référence dans cette région.
Au-delà de la position 120 nous observons qu'il n'y a plus de séquences alignées dans cette région et que les séquences ne soient pas conservées.

La partie supérieure présente les cartes PAE où chaque sous-figure est étiquetée avec "rank_X" où X est le rang du modèle parmi les cinq prédits et les axes X et Y représentent les positions des résidus dans la séquence de protéine.
Les couleurs vont du bleu (erreur faible) au rouge (erreur élevée), avec des valeurs allant de 0 à

30 Å.

Les zones bleues le long de la diagonale montrent que les résidus prédits sont très proches de leur position réelle, ce qui indique une haute précision dans ces régions.

Les zones rouges indiquent des erreurs de prédiction plus élevées, où les positions des résidus sont moins précises.

Les cinq modèles montrent des motifs similaires avec des zones bleues concentrées le long de la diagonale et des zones rouges aux extrémités. Cela suggère une cohérence relative des modèles prédits entre eux.

Les motifs bleus et rouges apparaissent de manière récurrente dans toutes les cartes PAE, montrant que certaines parties de la protéine sont prédites avec plus de précision que d'autres. Les zones bleues sont probablement des domaines structurés et bien conservés, tandis que les zones rouges pourraient correspondre à des boucles ou à des régions non structurées.

3.3.3 Résultats de RFdiffusion

Dans cette partie nous avons pris les 5 modèles obtenus comme résultats dans AlphaFold que nous avons soumis de nouveau, les uns après les autres à RFdiffusion. Il faudra comme fait précédemment, avec la protéine d'intérêt fixer les résidus Cystéine "C" suivant les positions auxquelles ils se trouvent. La commande pour le faire est la suivante :

```
Contigs: 'A44':'A48':'A92':'A96'
```

RFdiffusion nous produit en entrée 8 séquences parmi lesquelles, le meilleur design sera choisi sur la base des différentes métriques. Le meilleur design de tous les 5 modèles est donc choisi et consigné dans le tableau ci-après :

Meilleur Modélisation Model N°	MPNN	pLDDT	PTM	PAE	RMSD	Séquence prédite
1	1.17	0.88	0.74	8.094	2.494	Data
2	1.19	0.84	0.68	7.939	1.128	Data
3	1.176	0.829	0.631	9.962	3.729	Data
4	1.190	0.837	0.673	8.460	1.954	Data
5	1.11	0.850	0.679	8.192	1.336	Data

Table 3.3 Tableau des différents design de RFdiffusion

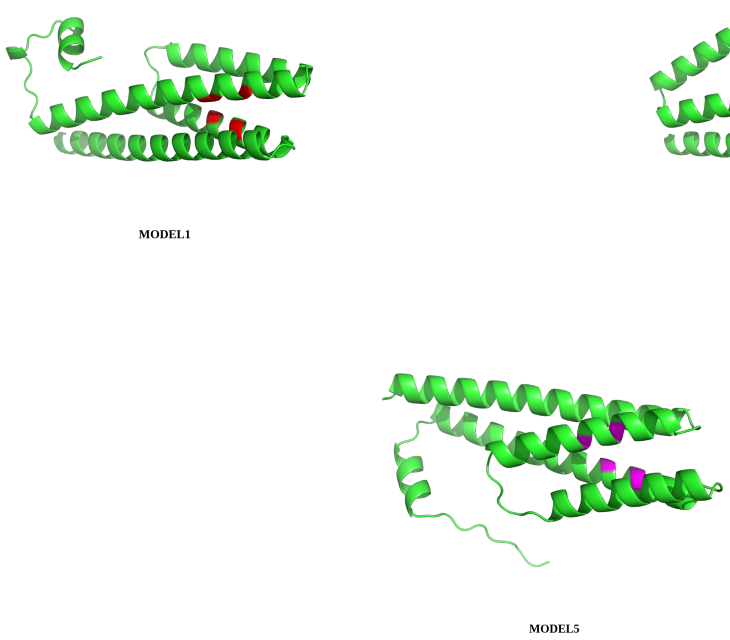

Figure 3.9 Figure montrant les résultats de RFdiffusion considérés positifs obtenus pour la combinaison de la recherche des motifs dans les groupes et de la protéine d'intéret

Sur les cinq modèles dont nous avons prédit les design et choisi les meilleurs, nous remarquons qu'il y en a certains qui marchent avec au départ une petite hélice et ensuite des hélices de plus en plus longues. Nous pouvons clairement voir que les design issus des modèles 1, 2 et 5 contiennent la petite hélice. La taille des petites hélices est énumérée dans le tableau ci-après :

Modèle N°	Petite hélice
1	7
2	14
5	8

Table 3.4 Tableau de la taille de la petite hélice des différents design bien alignés dans RFdiffusion

>best Design Model N°1
MKKRKVEFEEIRLGPPKEEPEELKKLKEKLEEKINELVELIEECIELCKEKAKKEKDEEEKKKILEVREKMEEMLAKEGLTDE
ELEELAELCIEACEYLAKKEKDEEEKKKWEELLEKLKETREEIKKVKEELEKLKKELR

>best Design Model N°2
SLREELLAALKECAELCKKALKVEKDKEIKEELEKLIEKLEAVLADPNATEEELIAALKECAELCKKYKKEFLKVLSEEEINA
LIERLEALIERAERE REERRRLEEELE

>best Design Model N°3
HMREKLIEALKKCRELCEKALEKEENEEIKKKLKELIEEIEKLLEDPNATLEEYIEALKKCRDLCKELREELLVVLSQEEIDAL
IEELEELIEEAEKELKERKELAEKLA

>best Design Model N°4
HMREKLLEALEKCRDLCEKALEKETNPEIKARLQAEIERLTALLADPDATLEELIEALEKCRALCEELAEELLVVISQEEIDEL
IEELEELIEEARKKIEEEEKAREEAR

>best Design Model N°5
SMREKLLKALKECKELCEEALKEETNEEIKKKLQEKIESITKLLEDPNATLEELIKALKECKDLCEELKEEFKKVKSEEEIEKL
IEEIEKLIEEAEREQAERAAAAAAAA

À côté de design qui semblent avoir bien été réalisés, nous avons quelques-uns qui ne semblent pas avoir bien marché compte tenu de l'absence de la petite hélice et du décalage dans les alignements. La figure 3.10 suivante illustre les résultats obtenus pour les modèles 3 et 4.

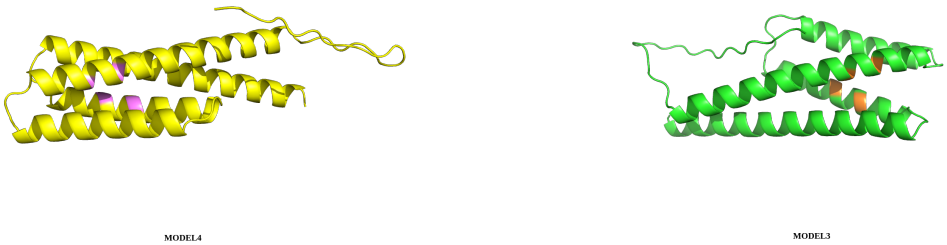

Figure 3.10 Figure montrant les résultats considérés négatifs obtenus de la combinaison de la les motifs dans les groupes et du design de séquences alternatives.

CONCLUSION

3.4 Vue d'ensemble sur l'étude

Somme toute, les protéines incorporant des clusters fer-soufre ([Fe-S]) jouent un rôle essentiel dans une large variété de processus biologiques, et ce, dans l'ensemble du vivant. Parmi ces cofacteurs, le cluster [4Fe-4S] se distingue par sa complexité structurale et par les mécanismes sophistiqués impliqués dans son insertion dans les matrices protéiques. Cette complexité a suscité un intérêt considérable au sein de la communauté scientifique, donnant lieu à de nombreuses recherches visant à comprendre et reproduire les processus de biogenèse des clusters métalliques.

C'est dans ce contexte que s'inscrit notre travail, dont l'objectif principal est de concevoir artificiellement, par des approches informatiques, des protéines capables d'incorporer un cluster [4Fe-4S] de type SF4. Plus précisément, nous avons cherché à détourner, de manière contrôlée, les mécanismes cellulaires naturels d'assemblage en concevant des séquences protéiques optimisées in silico, aptes à favoriser l'intégration spontanée du cofacteur.

Dans un premier temps, un travail de collecte, de traitement et d'analyse des données structurales a été mené afin de comprendre l'organisation hiérarchique des protéines au voisinage du cluster SF4. Cette phase exploratoire a mis en évidence que la structure protéique peut être décrite comme une hiérarchie imbriquée de modèles, de chaînes polypeptidiques, de résidus d'acides aminés, et d'atomes. L'analyse fine de ces composantes nous a permis de déterminer les proportions et les positions relatives des chaînes et acides aminés entourant le cluster, fournissant ainsi les fondements nécessaires pour la phase de conception.

Cette base structurale a ensuite été exploitée pour identifier des motifs conservés, sélectionner des régions d'intérêt, et orienter les étapes suivantes du design assisté par des outils de modélisation structurale tels que AlphaFold, pour la prédiction tridimensionnelle, et RFdiffusion, pour la génération de séquences respectant les contraintes spatiales imposées. Ces outils ont permis de proposer des prototypes de protéines hybrides plausibles, dans le cadre d'une approche exploratoire visant à ouvrir de nouvelles perspectives en bio-ingénierie et biologie synthétique.

3.5 Contributions de l'étude

Ce mémoire propose une contribution originale à la conception rationnelle de systèmes moléculaires hybrides, en s'appuyant sur une stratégie combinatoire d'identification et d'intégration de motifs fonctionnels partagés, notamment ceux impliqués dans l'interaction avec des clusters métalliques de type SF4. L'objectif est de générer des entités hybrides combinant, au sein d'une seule structure, des éléments fonctionnels issus de plusieurs contextes moléculaires.

La démarche repose sur l'identification de motifs conservés via des analyses comparatives, leur intégration dans des architectures cibles, et la génération de séquences compatibles avec les contraintes structurales et de séquentielles attendues. La validation des entités conçues est assurée à l'aide d'outils de modélisation structurelle, tels que AlphaFold pour la prédiction tridimensionnelle de la conformation finale, et RFdiffusion pour l'exploration guidée de séquences respectant la topologie souhaitée. Ce couplage permet d'évaluer la plausibilité structurale, la stabilité du repliement, ainsi que la compatibilité des modules greffés, ouvrant la voie à la création de protéines artificielles fonctionnelles.

3.6 Perspectives et améliorations portant sur l'étude

Bien que les résultats obtenus reposent exclusivement sur des approches in silico, plusieurs axes d'amélioration peuvent être envisagés pour renforcer la portée de ce travail. Un prolongement naturel consisterait à soumettre les molécules hybrides générées à une validation expérimentale. Cela permettrait notamment d'évaluer la stabilité réelle des structures prédictivement favorables, ainsi que leur capacité à interagir efficacement avec les partenaires biologiques ciblés. À ce jour, ces aspects dépassent le cadre de ce mémoire, qui se concentre sur les aspects computationnels.

Par ailleurs, l'approche développée pourrait être généralisée à d'autres contextes biologiques, notamment les interactions protéine-protéine, qui représentent un enjeu majeur en biologie structurale. Lorsqu'une protéine est connue pour interagir avec plusieurs partenaires sans que les mécanismes précis soient élucidés, le système mis en place ici pourrait être mobilisé pour générer des variants ou des interfaces artificielles capables de mimer ou d'explorer ces interactions poten-

tielles.

Ainsi, l'apport de ce projet réside non seulement dans la conception d'un pipeline pour l'identification et l'intégration de motifs fonctionnels, mais aussi dans la mise à disposition d'un cadre réutilisable pour générer, par diffusion structurale dirigée, des molécules hybrides adaptées à divers systèmes biologiques complexes.

ANNEXE A

ANNEXE

Tous les algorithmes utilisés sont dans le dépôt github : https://gitlab.info.uqam.ca /adekoudjo.adedayo_nassir/sf4ligand.

SCRIPT

BIBLIOGRAPHIE

Abramson, J., Adler, J., Dunger, J., Evans, R., Green, T., Pritzel, A., Ronneberger, O., Willmore, L., Ballard, A. J., Bambrick, J. *et al.* (2024). Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, *630*(8016), 493–500.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W. et Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*, *215*(3), 403–410.

Anand, N. et Achim, T. (2022). Protein structure and sequence generation with equivariant denoising diffusion probabilistic models. *arXiv preprint arXiv :2205.15019*.

Baek, M., DiMaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G. R., Wang, J., Cong, Q., Kinch, L. N., Schaeffer, R. D. *et al.* (2021). Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, *373*(6557), 871–876.

Bailey, T. L., Elkan, C. *et al.* (1994). Fitting a mixture model by expectation maximization to discover motifs in bipolymers.

Bailey, T. L., Johnson, J., Grant, C. E. et Noble, W. S. (2015). The meme suite. *Nucleic acids research*, *43*(W1), W39–W49.

Beitz, E. (2000). Texshade : shading and labeling of multiple sequence alignments using latex2e. *Bioinformatics*, *16*(2), 135–139.

Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. et Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, *28*(1), 235–242.

Blau, T., Ganz, R., Kawar, B., Bronstein, A. et Elad, M. (2022). Threat model-agnostic adversarial defense using diffusion models. *arXiv preprint arXiv :2207.08089*.

Boyd, E. S., Thomas, K. M., Dai, Y., Boyd, J. M. et Outten, F. W. (2014). Interplay between oxygen and fe–s cluster biogenesis : insights from the suf pathway. *Biochemistry*, *53*(37), 5834–5847.

Burley, S. K., Bhatt, R., Bhikadiya, C., Bi, C., Biester, A., Biswas, P., Bittrich, S., Blaumann, S., Brown, R., Chao, H. *et al.* (2025). Updated resources for exploring experimentally-determined pdb structures and computed structure models at the rcsb protein data bank. *Nucleic acids research*, *53*(D1), D564–D574.

Buxbaum, E. *et al.* (2007). Protein structure. In *Fundamentals of protein structure and function* chapitre 2, 15–43. Springer.

Chenna, R., Sugawara, H., Koike, T., Lopez, R., Gibson, T. J., Higgins, D. G. et Thompson, J. D. (2003). Multiple sequence alignment with the clustal series of programs. *Nucleic acids research*, *31*(13), 3497–3500.

Cock, P. J., Antao, T., Chang, J. T., Chapman, B. A., Cox, C. J., Dalke, A., Friedberg, I., Hamelryck, T., Kauff, F., Wilczynski, B. *et al.* (2009). Biopython : freely available python tools for computational molecular biology and bioinformatics. *Bioinformatics*, *25*(11), 1422.

Croitoru, F.-A., Hondru, V., Ionescu, R. T. et Shah, M. (2023). Diffusion models in vision : A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(9), 10850–10869.

Dauparas, J., Anishchenko, I., Bennett, N., Bai, H., Ragotte, R. J., Milles, L. F., Wicky, B. I., Courbet, A., de Haas, R. J., Bethel, N. *et al.* (2022). Robust deep learning–based protein sequence design using proteinmpnn. *Science*, *378*(6615), 49–56.

Dempster, A. P., Laird, N. M. et Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the royal statistical society : series B (methodological)*, *39*(1), 1–22.

Grzyb, J., Xu, F., Nanda, V., Łuczkowska, R., Reijerse, E., Lubitz, W. et Noy, D. (2012). Empirical and computational design of iron-sulfur cluster proteins. *Biochimica et Biophysica Acta (BBA)-Bioenergetics*, *1817*(8), 1256–1262.

Grzyb, J., Xu, F., Weiner, L., Reijerse, E. J., Lubitz, W., Nanda, V. et Noy, D. (2010). De novo design of a non-natural fold for an iron–sulfur protein : Alpha-helical coiled-coil with a four-iron four-sulfur cluster binding site in its central core. *Biochimica et Biophysica Acta (BBA)-Bioenergetics*, *1797*(3), 406–413.

Guzzo, A. V. (1965). The influence of amino acid sequence on protein structure. *Biophysical journal*, *5*(6), 809–822.

Han, G. W., Yang, X.-L., McMullan, D., Chong, Y. E., Krishna, S., Rife, C. L., Weekes, D., Brittain, S. M., Abdubek, P., Ambing, E. *et al.* (2010). Structure of a tryptophanyl-trna synthetase containing an iron–sulfur cluster. *Structural Biology and Crystallization Communications*, *66*(10), 1326–1334.

Hou, J., Wu, T., Cao, R. et Cheng, J. (2019). Protein tertiary structure modeling driven by deep learning and contact distance prediction in casp13. *Proteins : Structure, Function, and Bioinformatics*, *87*(12), 1165–1178.

Iverson, T. M., Luna-Chavez, C., Croal, L. R., Cecchini, G. et Rees, D. C. (2002). Crystallographic studies of the escherichia coli quinol-fumarate reductase with inhibitors bound to the quinol-binding site. *Journal of Biological Chemistry*, *277*(18), 16124–16130.

Jagilinki, B. P., Ilic, S., Trncik, C., Tyryshkin, A. M., Pike, D. H., Lubitz, W., Bill, E., Einsle, O., Birrell, J. A., Akabayov, B. *et al.* (2020). In vivo biogenesis of a de novo designed iron–sulfur protein. *ACS Synthetic Biology*, *9*(12), 3400–3407.

Johnson, D. C., Dean, D. R., Smith, A. D. et Johnson, M. K. (2005). Structure, function, and formation of biological iron-sulfur clusters. *Annu. Rev. Biochem.*, *74*(1), 247–281.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A. *et al.* (2021a). Applying and improving alphafold at casp14. *Proteins : Structure, Function, and Bioinformatics*, *89*(12), 1711–1721.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A. *et al.* (2021b). Highly accurate protein structure prediction with alphafold. *nature*, *596*(7873), 583–589.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. et Moult, J. (2019). Critical assessment of methods of protein structure prediction (casp)—round xiii. *Proteins : Structure, Function, and Bioinformatics*, *87*(12), 1011–1020.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. et Moult, J. (2021). Critical assessment of methods of protein structure prediction (casp)—round xiv. *Proteins : Structure, Function, and Bioinformatics*, *89*(12), 1607–1617.

Li, W., Jaroszewski, L. et Godzik, A. (2001). Clustering of highly homologous sequences to reduce the size of large protein databases. *Bioinformatics*, *17*(3), 282–283.

Lill, R. (2009). Function and biogenesis of iron–sulphur proteins. *Nature*, *460*(7257), 831–838.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y. *et al.* (2023). Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, *379*(6637), 1123–1130.

Moult, J., Pedersen, J. T., Judson, R. et Fidelis, K. (1995). A large-scale experiment to assess protein structure prediction methods.

Needleman, S. B. et Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, *48*(3), 443–453.

Rees, D. C. (2002). Great metalloclusters in enzymology. *Annual review of biochemistry*, *71*(1), 221–246.

Ruprecht, J., Yankovskaya, V., Maklashina, E., Iwata, S. et Cecchini, G. (2009). Structure of escherichia coli succinate : quinone oxidoreductase with an occupied and empty quinone-binding site. *Journal of Biological Chemistry*, *284*(43), 29836–29846.

Saitou, N. et Nei, M. (1987). The neighbor-joining method : a new method for reconstructing phylogenetic trees. *Molecular biology and evolution*, *4*(4), 406–425.

Schrödinger, LLC (2015). The PyMOL molecular graphics system, version 1.8.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A. *et al.* (2019). Protein structure prediction using multiple deep neural networks in the 13th critical assessment of protein structure prediction (casp13).

Proteins : structure, function, and bioinformatics, *87*(12), 1141–1148.

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W., Bridgland, A. *et al.* (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, *577*(7792), 706–710.

Simons, K. T., Kooperberg, C., Huang, E. et Baker, D. (1997). Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and bayesian scoring functions. *Journal of molecular biology*, *268*(1), 209–225.

Sokal, R. R. et Michener, C. D. (1958). A statiscal method for evaluating systematic relationships. *Univ Kans sci bull*, *38*, 1409–1438.

Thompson, J. D., Higgins, D. G. et Gibson, T. J. (1994). Clustal w : improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic acids research*, *22*(22), 4673–4680.

Trippe, B. L., Yim, J., Tischer, D., Baker, D., Broderick, T., Barzilay, R. et Jaakkola, T. (2022). Diffusion probabilistic modeling of protein backbones in 3d for the motif-scaffolding problem. *arXiv preprint arXiv :2206.04119*.

Watson, J. L., Juergens, D., Bennett, N. R., Trippe, B. L., Yim, J., Eisenach, H. E., Ahern, W., Borst, A. J., Ragotte, R. J., Milles, L. F. *et al.* (2023). De novo design of protein structure and function with rfdiffusion. *Nature*, *620*(7976), 1089–1100.

Wu, R., Ding, F., Wang, R., Shen, R., Zhang, X., Luo, S., Su, C., Wu, Z., Xie, Q., Berger, B. *et al.* (2022). High-resolution de novo structure prediction from primary sequence. *BioRxiv*, 2022–07.

Xu, J. et Zhang, Y. (2010). How significant is a protein structure similarity with TM-score = 0.5 ? *Bioinformatics*, *26*(7), 889–895. `http://dx.doi.org/10.1093/bioinformatics/btq066`. Récupéré de `https://doi.org/10.1093/bioinformatics/btq066`

Xu, Q. et Dunbrack Jr, R. L. (2023). The protein common assembly database (protcad)—a comprehensive structural resource of protein complexes. *Nucleic Acids Research*, *51*(D1), D466–D478.

Zheng, W., Li, Y., Zhang, C., Pearce, R., Mortuza, S. et Zhang, Y. (2019). Deep-learning contact-map guided protein structure prediction in casp13. *Proteins : Structure, Function, and Bioinformatics*, *87*(12), 1149–1164.