ADAM Gabriel

MEYVAERT Corentin

TENA Elise

**Rapport du Projet Python - 2023**



Formation: Génie Biologique GB4 – Polytech Nice Sophia

Professeurs: Mr. Gautier et Mme Robbe

Année scolaire : 2023-2024



Table des matières

[I- Introduction 3](#_Toc155261390)

[II- Présentation de notre programme 3](#_Toc155261391)

[A. Le programme dans son ensemble 3](#_Toc155261392)

[B. Avant de lancer le programme 5](#_Toc155261393)

[C. Détail de chaque fonction créée 5](#_Toc155261394)

[III- Stratégie mise en place / organisation 9](#_Toc155261395)

[A. Corentin 10](#_Toc155261396)

[B. Elise 10](#_Toc155261397)

[C. Gabriel 10](#_Toc155261398)

[IV- Commentaires sur le projet et les résultats 11](#_Toc155261399)

[A. Corentin 11](#_Toc155261400)

[B. Elise 11](#_Toc155261401)

[C. Gabriel 11](#_Toc155261402)

[V- Axes d’améliorations envisagées 11](#_Toc155261403)

# Introduction

L'étude des sciences biologiques présente un défi continu en raison de la richesse complexe et de la diversité des informations qui s'y trouvent. En effet, les domaines tels que la génomique, la protéomique, la métabolomique et d'autres branches des sciences biologiques révèlent des informations souvent entremêlées, ce qui nécessite une approche intégrée de bio-informatique afin d’extraire les connaissances et de les rassembler.

Ces données sont variées car chaque élément, qu'il s'agisse de gènes, de protéines, de voies métaboliques ou d'interactions cellulaires revêtent d’une importance cruciale dans le développement de nouvelles avancées, que ce soit dans le domaine pharmaceutique, industriel ou environnemental. C’est pourquoi pour relever ce défi, la bio-informatique s'est imposée comme une discipline essentielle, offrant une approche basée sur l'utilisation de la statistique et de l'informatique pour analyser et modéliser des éléments tels que les séquences génomiques, les protéines et leurs structures tridimensionnelles. Ces informations, une fois extraites, se transforment en connaissances essentielles pour résoudre des problématiques variées, que ce soit dans divers domaines comme par exemple la médecine personnalisée, le développement de médicaments, la compréhension de maladies génétiques, ou encore même au niveau de l'optimisation de cultures agricoles.

C’est pourquoi un projet comme celui que nous avons mené se montre très enrichissant. En effet, dans le cadre de ce projet collaboratif, notre objectif était d'explorer des données relatives à des protéines stockées dans la base de données PDB (Protein Data Bank). Notre travail avait pour but de créer un programme qui permet d’extraire un maximum d'informations en rapport avec notre séquence protéique, sa structure tridimensionnelle, ainsi que ses propriétés physico-chimiques.

À cet effet, nous avons utilisé Python ainsi que différents modules dans Python pour concevoir des fonctions dédiées à l'analyse de protéine, englobant des aspects tels que la récupération de données PDB, la construction d’une séquence Fasta, l’analyse de la composition en acides aminés de la protéine, le calcul du profil d'hydrophobicité, la détection des ponts disulfures et le calcul de la matrice de contact.

# Présentation de notre programme

## Le programme dans son ensemble

Dans cette partie, nous présenterons le programme principal, puis nous détaillerons les fonctions que nous avons utilisées dans les parties suivantes.

Tout d’abord, nous avons choisi de créer plusieurs modules afin de répondre à chaque question du projet, pour un souci de clarté et de meilleure compréhension du programme. Notre programme principal effectue une série d'analyses répartis en module sur la protéine "1CRN" dans notre exemple, car nous avons fait le choix de tester notre programme avec cette protéine en premier lieu. Voici un résumé des étapes effectuées par notre programme :

Le module 1 permet l’importation des données de la protéine en question à partir du site web PDB ou à partir de l’ordinateur en utilisant le code "1CRN". Le module 2 est utilisé pour rechercher et afficher des informations importantes sur la protéine, telles que sa description, sa résolution, et d'autres détails. Le module 3 est employé pour rechercher et afficher la liste des acides aminés présents dans la protéine, ainsi que pour construire une séquence au format fasta. Ensuite, l'utilisateur est invité à choisir entre enregistrer ou visualiser la séquence au format fasta, ou peut également choisir de faire les deux. En fonction de la réponse de l'utilisateur, le programme effectue donc différentes actions. Nous avons créé le module 4 pour qu’il calcule la fréquence des acides aminés dans la protéine et qu’il affiche les résultats. Le programme utilise également le module 4 pour générer un graphique illustrant la fréquence des acides aminés dans la protéine. Le module suivant (module 5) est utilisé pour calculer le profil d'hydrophobicité de la séquence Fasta de notre protéine. Ensuite, le programme affiche les valeurs du profil d'hydrophobicité et les sauvegarde dans un fichier Excel nommé "hydrophobicity\_profile.xlsx". Le module 6 affiche les possibles pont-disulfures de la protéine (en tenant compte de la distance entre les CYS et de la présence d’hélice alpha). Le module 7 permet de créer un nouveau fichier PDB où les B-factor représentent l’hydrophobicité du résidu. Le module 8 permet de créer une matrice de contact de la distance entre les acides aminés.

En résumé, notre programme combine plusieurs modules pour fournir une analyse approfondie de la protéine choisie, en mettant l'accent sur des aspects tels que la composition en acides aminés, la fréquence, le profil d'hydrophobicité, et offre à l'utilisateur des options interactives pour explorer les résultats.

Une image contenant capture d’écran, Tracé, diagramme, ligne

Description générée automatiquement

Figure 1 : Histogramme de la fréquence d’acides aminés dans notre protéine

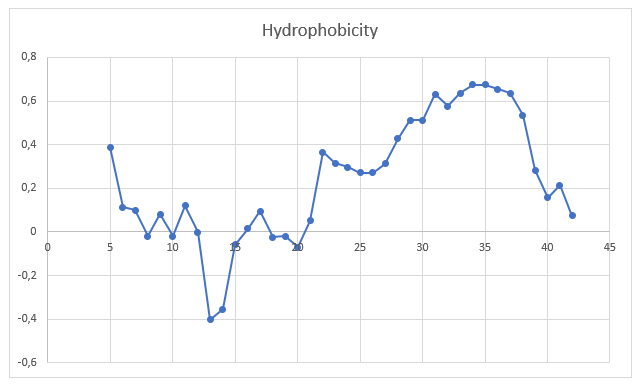


Figure 2 : Profil d’hydrophobicité de la protéine 1CRN

## Avant de lancer le programme

Afin de pouvoir utiliser convenablement notre programme Python, il faudra lire le « readme » servant à installer tous les packages nécessaires à l’utilisation de notre programme.

## Détail de chaque fonction créée

Voici chacune des fonctions que nous avons créées, accompagnées de leur rôle dans le programme :

**« importation\_online » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Récupérer le contenu d'un fichier PDB à partir d'une source en ligne. La fonction utilise le module urllib.request pour ouvrir une connexion avec l'URL du fichier PDB associé au code fourni. | Son argument est « code ». Il s'agit du code associé à une protéine dans la base de données PDB. Ce code est utilisé pour construire l'URL à partir de laquelle le fichier PDB sera récupéré. | « Fichier » est l’output de cette fonction : il s'agit d'une chaîne de caractères (str) contenant le contenu du fichier PDB. Ce fichier est obtenu en récupérant les lignes du fichier qu'elle concatène ensuite en une seule chaîne. En cas d'échec de la connexion ou de la récupération du fichier PDB, la fonction renvoie un message d'erreur indiquant le problème (gestion d’erreur dans la fonction). |

**« rech\_desc2 » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Rechercher et extraire des informations spécifiques liées à la description d'un fichier PDB à partir du contenu brut du fichier. | Chaîne de caractères « fichier\_brut » qui représente l'ensemble des informations du fichier PDB. | « descf » : il s'agit d'une chaîne de caractères qui représente la description du fichier PDB. Cette description est extraite à partir des lignes du fichier PDB qui commencent par "TITLE" ou "HEADER", etc.... La sortie de la fonction inclut donc la mention "Description du fichier PDB:" suivie des lignes de description correspondantes. |

**« rech\_AA2 » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Rechercher et extraire les codes à trois lettres des acides aminés de la protéine à partir du contenu brut d'un fichier PDB. | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations du fichier. | « desc » : il s'agit d'une liste contenant les codes à trois lettres des acides aminés de la protéine, extraits du fichier PDB. |

**« fasta » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Convertir une liste d'acides aminés de code à 3 lettres à une liste de code à une lettre en utilisant un dictionnaire préalablement défini (acides\_amines2). | La fonction prend en argument une liste « res\_rech\_AA » qui contient les codes à trois lettres des acides aminés de la protéine. | « resf » : il s'agit d'une chaîne de caractères représentant la séquence fasta générée à partir de la liste d'acides aminés fournie en entrée. |

**« creation\_fasta » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Créer un fichier au format Fasta dans l'ordinateur de l’utilisateur. Le fichier Fasta contiendra le header, la séquence fasta et sera nommé en fonction du code PDB de la protéine (ex : 1CRN). | « fasta »: il s'agit de la séquence au format FASTA que l’utilisateur veut écrire dans le fichier.  « desc »: il s'agit de la description du fichier PDB associé à la protéine, qui sera utilisée pour créer le header du fichier Fasta.  « code\_prot »: il s'agit du code PDB de la protéine, qui sera inclus dans le nom du fichier Fasta. | L’output est la phrase : "Un fichier fasta a été créé dans votre dossier python". Il s'agit d'une chaîne de caractères indiquant que le processus de création du fichier FASTA a été effectué avec succès. |

**« affich\_fasta » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Le but est de générer et afficher la séquence au format Fasta dans la console, utilisant la séquence fournie (« fasta ») et la description du fichier PDB associé à la protéine (« desc »). | Il y a 2 arguments :  « fasta » : il s'agit de la séquence au format Fasta que vous souhaitez afficher.  « desc » : description du fichier PDB associé à la protéine. | "Voici la séquence de votre protéine au format fasta:", fich\_fasta\_3: Il s'agit d'un tuple constitué d'une chaîne de caractères indiquant que la séquence de la protéine au format Fasta est affichée. |

**« cpt\_AA » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Compter le nombre de chaque acide aminé (AA) dans une séquence d'acides aminés fournie (seq\_AA). Elle utilise un dictionnaire (dico\_AA) pour stocker ces comptages. | La fonction prend en argument une séquence d'acides aminés (« seq\_AA »). | « dico\_AA » : il s'agit d'un dictionnaire où les clés sont les codes à trois lettres des acides aminés, et les valeurs associées sont le nombre de fois que chaque acide aminé apparaît dans la séquence fournie. |

**« freq\_AA » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Calculer la fréquence de chaque acide aminé dans une séquence d'acides aminés fournie (seq\_AA). Elle utilise la fonction cpt\_AA pour compter le nombre de chaque acide aminé, puis normalise ces compteurs pour obtenir des fréquences. | Séquence d'acides aminés (« seq\_AA »). | « dico\_AA » |

**« graph\_frq » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Le but est de créer un histogramme de fréquence d'acides aminés en utilisant deux dictionnaires de fréquences. | La fonction prend en argument un dictionnaire « dico\_AA » contenant les fréquences d'acides aminés obtenues à partir de la séquence fournie. | L’output est l’affichage d'un histogramme de fréquence d'acides aminés, avec des barres pour les fréquences obtenues à partir de la séquence fournie (« dico\_AA ») et des barres pour les fréquences de référence (« dicoref\_AA »). La légende indique quelle couleur correspond à chaque dictionnaire. |

**« calculate\_hydrophobicity » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| La fonction « calculate\_hydrophobicity » a pour objectif de calculer le profil d'hydrophobicité d'une séquence d'acides aminés au moyen d'une fenêtre glissante de 9 acides aminés.  La fonction utilise une fenêtre glissante de taille 9 (définie par la fonction « window\_size ») pour parcourir la séquence d'acides aminés.  Pour chaque position dans la séquence (jusqu'à la longueur de la séquence moins la taille de la fenêtre plus un), elle extrait donc une fenêtre de 9 acides aminés. | « seq\_fasta » : il s'agit de la séquence d'acides aminés au format Fasta pour lequel le profil d'hydrophobicité doit être calculé. | L’output est « hydrophobicity\_values » : il s'agit d'une liste contenant les valeurs d'hydrophobicité calculées pour chaque position de la séquence. |

**« save\_to\_excel » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| La fonction « save\_to\_excel » a pour objectif d'enregistrer les valeurs d'hydrophobicité dans un fichier Excel.  La fonction utilise le module Pandas pour créer un DataFrame à partir du dictionnaire data.  Elle utilise la méthode « to\_excel » de Pandas pour enregistrer le DataFrame dans un fichier Excel spécifié par « file\_name ». Le nom de la feuille Excel est défini par « protein\_name ». | Cette fonction prend 3 arguments : « file\_name » : c'est le nom du fichier Excel dans lequel les données d'hydrophobicité seront enregistrées.  « protein\_name » : c'est le nom de la protéine associée aux données d'hydrophobicité.  « hydrophobicity\_values » : c’est la liste des valeurs d'hydrophobicité à enregistrer dans le fichier Excel. | Cette fonction enregistre les valeurs d'hydrophobicité dans un fichier Excel spécifié par l'utilisateur. |

**« rech\_pondi » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Permet de chercher toutes les lignes du fichier ayant des sulfures | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations du fichier. | Donne les lignes du fichier qui contiennent des sulfures.  Donne aussi le nombre de cystéine dans la protéine. |

**« rech\_pos » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Argument | Output |
| Permet de n'avoir que les positions des sulfures dans une liste. | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations du fichier. | Une liste des positions des sulfures. |

**« calc\_distance » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Calculer les distances entre chaque sulfure et ne garder que les sulfures qui ont une distance entre eux inférieure à 3 Angstrom, et qui ne sont pas dans la même hélice alpha. | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations du fichier. | Une liste avec à la suite les sulfures concernés puis la distance les séparant. |

**« rech\_pos\_hel » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Permet d’avoir les positions des hélices alpha dans une liste. | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations | Une liste de liste des différentes positions des hélices alpha. |

**« rech\_bfactor »**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Permet de modifier le fichier PDB en remplaçant la valeur du B\_factor par des valeurs d’hydrophobicité des résidus | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations | Une chaîne de caractère qui correspond au fichier PDB modifié. |

**« create\_pdb »:**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Crée un nouveau fichier PDB avec la nouvelle valeur de B factor | Une chaîne de caractère qui correspond au fichier PDB modifié. (Le résultat de rech\_bfactor) et le code de la protéine | Une chaine de caractère : « Nouveau fichier PDB crée avec succès. » et un fichier .pdb |

**« creation\_matrice » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Créer une matrice de contact des distances entre les différents résidus de la protéine | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations | Une matrice numpy correspondant à la matrice de contact. |

**« heatmap » :**

|  |  |  |
| --- | --- | --- |
| But de la fonction | Arguments | Output |
| Créer une heatmap représentant la matrice de contact | « fichier\_brut »: il s'agit du contenu brut du fichier PDB. C'est une chaîne de caractères(str) qui représente l'ensemble des informations | Retourne une chaîne de caractère vide et un graphique. |

# Stratégie mise en place / organisation

Dès l’annonce du projet, nous nous sommes regroupés tous les trois afin de se répartir le travail. Nous nous sommes donc répartis les modules afin d’avancer chacun de notre côté puis au fur et à mesure de nos avancées, nous mettions notre travail en commun ce qui nous permettez d’améliorer les fonctions des uns et des autres. Une fois que nous avions mis en commun nos idées concernant l’ensemble des fonctions à créer et la façon dont nous voulions présenter nos résultats, nous avons créé un google drive afin de partager nos codes au fur et à mesure.

Pendant les vacances de Noël, nous avons terminé tous les codes puis avons rédigé le rapport grâce à un Word partagé afin que tout le monde avance simultanément. Nous avons fini de nous expliquer les codes entre nous et avons fait un point sur les difficultés rencontrées lors de ce projet.

Pour finir, nous nous sommes réunis une dernière fois afin de se mettre d’accord sur le fond et la forme de notre future présentation orale.

Voici ce que chacun d’entre nous à réaliser :

## Corentin

J’ai réalisé le module permettant de créer la séquence fasta de la protéine, ainsi que celui permettant de chercher les possibles ponts disulfures et le bonus. Pour le reste des modules nous avons travaillé ensemble. J’ai appris à installer de nouveaux packages sur l’ordinateur, afin de pouvoir créer des fonctions les plus optimisées possibles. J’ai aussi appris à manipuler un fichier PDB, à comprendre comment il était organisé, et j’ai appris la manière de rechercher des informations dans le fichier. Je me suis occupé de faire l’interface utilisateur et de faire la gestion des erreurs.

J’ai eu du mal à gérer les différences entre les fichiers PDB. Par exemple, pour le module qui permet de créer un nouveau fichier PDB en changeant le B-factor, la variabilité des positions des éléments de chaque ligne fait que la fonction n’est pas capable de traiter tous les fichiers PDB (certains ne pourront pas être colorés dans pymol). Sinon le module pour la recherche de ponts disulfures était aussi complexe à réaliser étant donné qu’il fallait prendre plusieurs conditions en compte comme la distance entre les sulfures et la présence des hélices alpha.

## Elise

Dans le projet, j’ai eu pour mission principale de créer les fonctions concernant le profil d’hydrophobicité et les fonctions concernant la fréquence des acides aminés dans la protéine. J’ai notamment utilisé les valeurs de l’échelle Fauchere et Pliska de 1983 et la formule de l’hydrophobicité moyenne pour établir le profil d’hydrophobicité de la protéine. J’ai également appris à installer de nouveaux packages sur l’ordinateur, afin de pouvoir créer des fonctions les plus optimisées possibles.

J’ai également élaboré le graphique du comparatif des fréquences des acides aminés de la séquence avec les fréquences moyennes sur Swiss-Prot. En ce qui concerne la fonction « graph\_frq », la difficulté a été de faire en sorte que les dictionnaires dico\_AA et dicoref\_AA contiennent respectivement les fréquences observées et les fréquences de référence des acides aminés et de créer l’histogramme avec les 2 ensembles de barre.

Pour l’affichage du profil d’hydrophobicité, la véritable difficulté que j’ai rencontrée a été d’apprendre à se servir du module openpyxl, mais j’ai rapidement compris quels outils il apportait et j’ai donc pu les utiliser. D’autre part, les fonctions n'incluent pas une gestion robuste des exceptions. Par exemple, si le fichier Excel est en cours d'utilisation, cela provoque une erreur lors de la sauvegarde, et j’ai donc eu affaire à ce problème que j’ai par la suite compris.

## Gabriel

Pendant ce projet, j’ai pu travailler sur différents modules en collaboration avec mon groupe mais mon rôle s’est principalement porté sur le module 2, visant à rechercher et afficher des informations importantes sur la protéine, telles que son nom, sa description, sa longueur, la technique expérimentale utilisée pour déterminer sa structure, les résidus de la chaîne polypeptidique, les hélices, les feuillets bêta et les ponts disulfure présents dans cette protéine. La seule réelle difficulté de cette partie a été d’extraire seulement les informations voulues. Par exemple, pour afficher la résolution, beaucoup de lignes étaient disponibles, mais il a fallu les trier afin de ne conserver que les plus importantes et les plus claires.

# Commentaires sur le projet et les résultats

Collectivement, nous partageons l'opinion selon laquelle la répartition des tâches au sein de notre équipe s'est déroulée de manière judicieuse, en prenant en considération les compétences individuelles et les orientations spécifiques de chacun. Cette approche a non seulement permis une utilisation efficace de nos ressources, mais a également contribué à la diversité et à la complémentarité des contributions de chaque membre.

À mesure que nous approchions de la conclusion du projet, nous avons su rapidement trouver un consensus sur la structure et le contenu de notre présentation orale, démontrant ainsi notre capacité à travailler de manière organisée. Nous avons tous voulu montrer notre engagement envers la qualité et la cohérence de nos résultats et démonstrations.

## Corentin

J’ai trouvé ce projet intéressant dans son ensemble. Etant donné que je veux poursuivre en option BIMB j’ai apprécié ce projet car il m’a permis d’avoir un bon entraînement en programmation. En effet, il n’est pas toujours simple de s’entraîner à coder avec des projets à sa portée. J’ai bien aimé la manière dont les différents problèmes à résoudre étaient liés et donc comment on pouvait réutiliser chaque fonction. J’ai pu renforcer mes compétences dans l’utilisation de plusieurs modules différents tels que matplotlib, math, pandas ou encore numpy. J’ai aussi bien aimé le fait d’organiser un dossier pour le projet avec un main et plusieurs modules, ce qui fait penser à l’organisation d’un vrai logiciel. J’ai hâte de poursuivre en filière BIMB et de pouvoir continuer à m’améliorer en programmation.

Je suis satisfait du travail rendu et de l’organisation générale du groupe.

## Elise

Ce projet m’a permis de me familiariser avec la manipulation de fichiers protéiques provenant de bases de données telles que PDB. Lors de la création du module concernant le profil d’hydrophobicité, j’ai appris à utiliser et à comprendre le module « openpyxl », ce qui pourra me servir pour la suite notamment pour des stages car ce module est très utile pour relier les données de python et les importer dans un fichier Excel. Ce travail m’a permis de me familiariser davantage avec Python et à en faire une utilisation plus fluide.

Je suis satisfaite vis-à-vis de mes idées d'affichage sur Excel, lesquelles non seulement respectent les consignes établies, mais fournissent également des résultats que je trouve qualitatifs. Cette expérience m'a offert l'opportunité de perfectionner mes compétences dans la représentation visuelle des données sur Excel ainsi que sur Python. Pour finir, je peux dire que la composition de notre groupe a contribué positivement à la réussite de ce projet. En somme, cette expérience a été enrichissante tout autant sur le plan technique avec l’approfondissement et la meilleure compréhension de Python, que sur le plan relationnel avec des membres de l’équipe très impliqués.

## Gabriel

Ce projet m'a tout d’abord permis d'apprendre énormément sur la mécanique de création de modules qui sont ensuite importés via un fichier principal. Compte tenu de la diversité des questions posées et donc des modules créés, j'ai pu découvrir et apprendre à utiliser certains packages. Ce travail de groupe m'a aussi permis de me familiariser avec Python en découvrant continuellement de nouvelles façons de coder, tout en partageant mes idées et mes connaissances avec les autres membres du groupe. En somme, ce projet m'a été très bénéfique, que ce soit pour partager nos connaissances ou pour découvrir de nouvelles fonctionnalités, tout en réfléchissant à la manière d'optimiser au mieux notre code.

Je suis satisfait du travail rendu, de l'organisation de mon groupe et de l'efficacité de mes collègues.Haut du formulaire

1. Axes d’améliorations envisagées

En conclusion, nous sommes tous les trois très satisfaits des résultats obtenus avec notre programme, mais nous reconnaissons qu'il existe très probablement des opportunités d'amélioration significatives, tel que la création d’une interface graphique, mais cela aurait nécessité de nombreuses connaissances et savoir-faire supplémentaires que nous n’avons pas aujourd’hui avec les délais impartis. Nous n’avons pas non plus eu le temps d’optimiser pleinement le module 7 qui crée un fichier PDB car dans certain fichier les positions de B\_factor varient et donc notre module n’arrive pas à modifier les valeurs de B\_factor.

Nous sommes également satisfaits du fait que nous ayons réussi à créer la matrice de contact qui était un bonus, tout en nous étant assurés de la robustesse et de la fiabilité de nos premières fonctions essentielles.