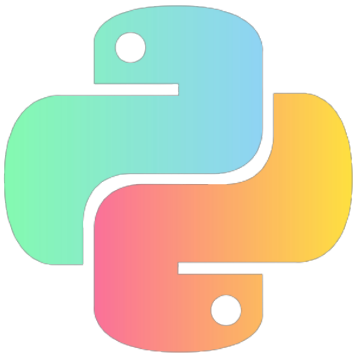
UNIVERSITÉ CÔTE D'AZUR

POLYTECH NICE-SOPHIA

## Projet de Groupe

# Programmation Python et extraction d'informations biologiques à partir d'un fichier PDB

**Romain Gautier et Karine Robbe-Sermesant**



RCSB PDB
PROTEIN DATA BANK

Année Scolaire 2023-2024

Groupe 12
Maëna Degoul - Sarah Ung - Angélique Vella

## Sommaire

# 1. Introduction

Les progrès dans les technologies expérimentales, notamment la cristallographie aux rayons X et la résonance magnétique nucléaire, ont considérablement amélioré la résolution des structures tridimensionnelles des protéines. En parallèle, les bases de données telles que la Protein Data Bank (PDB) et les outils de visualisation moléculaire comme Pymol ont élargi notre capacité à comprendre la structure 3D et le comportement des macromolécules biologiques. Ces avancées ont conduit à la collecte d'une grande quantité d'informations, rendant ainsi l'étude des fichiers PDB complexe et créant le besoin d'approches avancées de traitement de données.

C’est dans ce contexte que nous entreprenons ce projet d'analyse de fichiers PDB. Notre objectif est de développer une méthode pour extraire des connaissances significatives à partir de fichiers de données complexes et d’offrir des analyses de ces données pour faciliter la compréhension des mécanismes biologiques. Notre programme vous offre la possibilité d'obtenir une analyse complète de la composition en acides aminés d'une protéine, son profil d'hydrophobicité, sa capacité à créer des ponts disulfures, ainsi que l'analyse de ses propriétés physicochimiques par une visualisation sur le logiciel PyMol.

En résumé, nous vous proposons une interface utilisable par tous qui génère une visualisation des différentes analyses et permet de récupérer l’ensemble de ces résultats dans un fichier. Nous sommes convaincus que ces efforts contribueront à une meilleure compréhension de la complexité moléculaire et faciliteront la conception de nouvelles stratégies thérapeutiques.

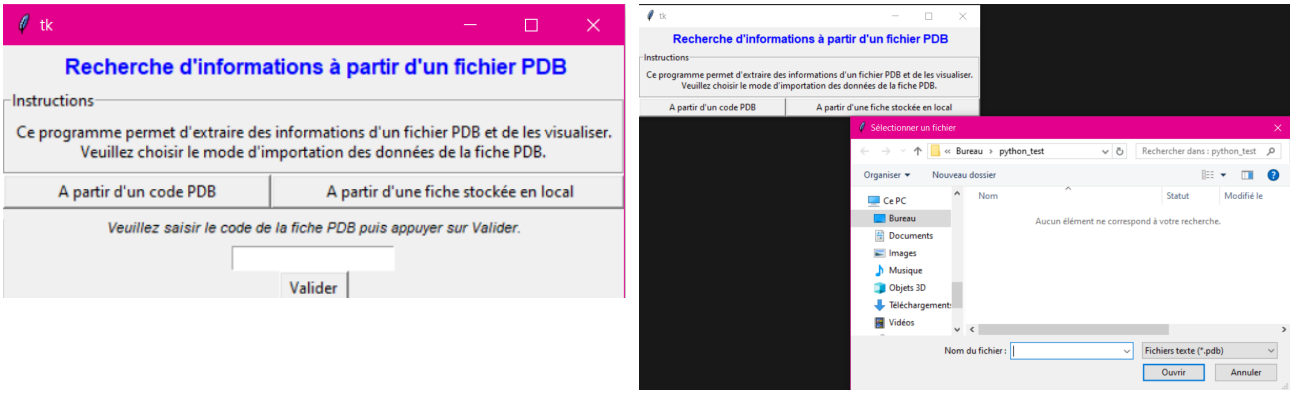
## 2. Programme

*Outre l'objectif purement biologique de notre projet, nous souhaitions également créer un outil rapide et simple d'utilisation pour tout biologiste. Ainsi, nous avons créé une interface graphique avec Tkinter permettant de minimiser les erreurs lors d'entrées d'informations par l'utilisateur et donc de simplifier l'utilisation par l'usage de bouton numérique à clic.*
***Afin de faciliter l'explication de notre programme et de vous présenter son utilité dans l'interprétation biologique de la structure 3D d'une protéine, nous avons choisi de présenter les résultats obtenus à partir du code PDB: 1CRN.***

### a) Récupération des fichiers

Nous avons décidé de donner à l'utilisateur la possibilité de choisir son mode de récupération de fichier. Dans la première étape, il peut choisir d'importer le fichier PDB depuis un navigateur web. Dans ce cas, il devra entrer le code PDB en minuscule ou en majuscule. Alternativement, il peut choisir d'utiliser directement un fichier local. Pour éviter toute erreur de chemin potentielle, compte tenu des difficultés liées aux différences entre les systèmes d'exploitation, nous avons opté pour la sélection directe du fichier grâce à une interface graphique créée avec la bibliothèque Python Tkinter.

Figure 1 : Présentation de la visualisation du gestionnaire de recherche grâce à l'interface tkinter à partir d'un code PDB: à partir d'un fichier local:




A l'issue de ces fonctions, nous obtenons deux listes. La première, nommée **"PDB_fichier"**, est composée de chaque ligne du fichier PDB sans les espaces anciennement présents en début et fin de chaque ligne. La seconde, nommée **"data"**, est une liste de listes. Chaque sous-liste est une ligne du fichier PDB et chaque élément d'une sous liste est un mot de cette ligne (le séparateur correspondant au caractère espace). Nous avons choisi ces formats de sortie dans le but de simplifier la récupération des informations que nous jugeons importantes à partir du fichier et de faciliter la reconstruction d'une fiche PDB modifiée, que le logiciel Pymol peut ouvrir.
Ainsi, ces deux listes nous ont servi de base pour extraire :

- Le **titre** de la fiche
- Le **nom de la protéine**
- L'**espèce** d'où provient les informations de la protéine
- La **méthode expérimentale** utilisée avec l'éventuelle **résolution** associée
- La **longueur de la séquence** en acides aminés

Pour faciliter la lecture par l'utilisateur, nous avons présenté des onglets contenant les informations de chaque résultat. Voici comment les informations essentielles relatives à notre protéine sont présentées:

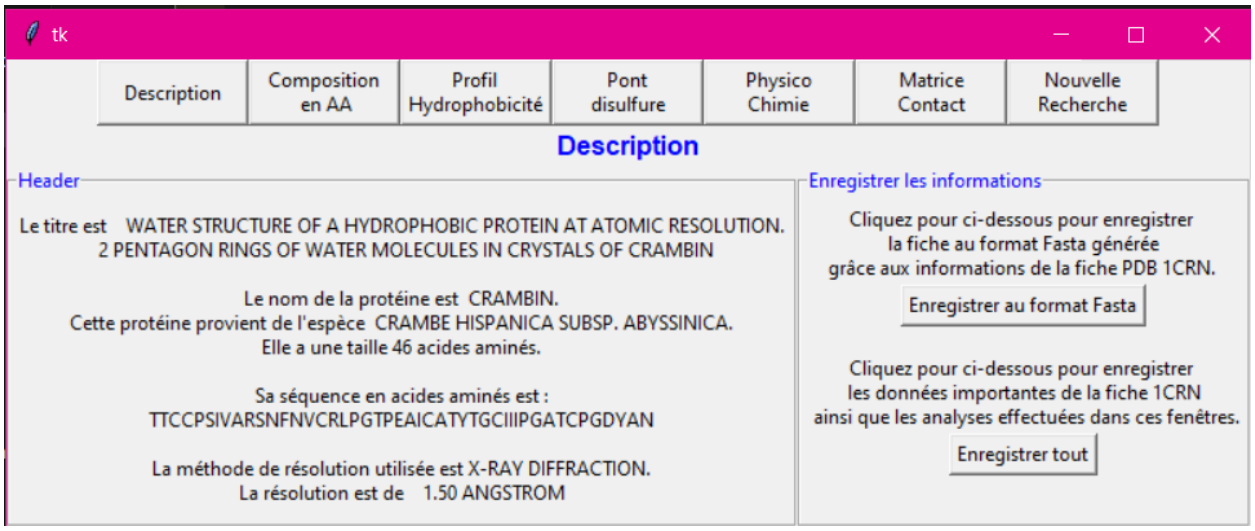



Figure 2 : Présentation de la visualisation du descriptif de la protéine Crambine

### b) Construction d'un fichier contenant la séquence Fasta

L'exploration de la structure 3D d'une protéine peut impliquer l'utilisation d'outils de prédiction, qui requièrent fréquemment une séquence d'acides aminés au format FASTA en tant qu'entrée. Dans cette optique, nous offrons à nos utilisateurs la possibilité de télécharger un fichier contenant cette séquence pour faciliter ce processus.

Les fichiers PDB contiennent des informations sur chaque atome constituant les résidus d'une protéine. Afin d'extraire la séquence en acides aminés sous forme du code à une lettre, nous avons commencé par récupérer les lignes contenant un CAlpha à partir de la liste data. Le CAlpha représente un atome unique pour chaque protéine, facilitant ainsi la discrimination entre les différents acides aminés. Pour chacune des lignes, nous avons extrait le code à trois lettres (3L) de chaque acide aminé afin de créer une liste tout en prenant en compte que le format de 3 lettres n'était pas toujours respecté. À l'aide d'un dictionnaire ayant pour clé le code à trois lettres (3L) et pour valeur le code à une lettre (1L), nous avons traduit cette liste en code à une lettre (1L). Enfin, cette liste a été transformée en chaîne de caractères et écrite dans un nouveau fichier. Le résultat est une première ligne d'en-tête descriptive (header) commençant par un chevron, suivie de la séquence en acides aminés présentée par ligne de 80 résidus.

Le header est composé dans l'ordre de lecture de : L'identifiant de la molécule (code PDB), les identifiants des chaînes décrites (A,B,C …), le nom de la molécule, le nom de l'organisme scientifique à partir de laquelle la protéine est issue, ainsi que son identifiant entre parenthèses. Ces mêmes informations sont retrouvées dans les remarques descriptives résumées du document. Nous les avons choisies, car elles sont utilisées régulièrement dans le header des fichiers fasta.

```
>1CRN 1|Chain A| CRAMBIN|Crambe hispanica subsp. abyssinica (3721)
TTCCPSIVARSNFNVCRLPGTPEAICATYTGCIIIPGATCPGDYAN
```

Figure 3 : Présentation du fichier texte contenant la séquence de la protéine Crambine au format Fasta

### c) Analyse de la composition en acides aminés de la séquence

A partir de la séquence en code à 1 L, nous comparons la fréquence observée de chaque acide aminé à la fréquence moyenne de chaque acide aminé écrite dans Swissprot.

Nous avons récupéré manuellement les données de fréquence de référence dans un dictionnaire (la clé représentant l'acide aminé et la valeur étant la fréquence moyenne d'apparition de celui-ci dans une séquence). Nous avons également construit un dictionnaire similaire pour les fréquences observées dans notre séquence. Nous avons choisi de représenter les résultats sous forme d'un histogramme et d'un tableau de valeurs. Ceci permet à l'utilisateur de visualiser clairement et rapidement les différences notables entre la séquence étudiée et les valeurs de références, tout en facilitant la récupération des données brutes.

En observant l'histogramme, on remarque que certains acides aminés sont très fréquents dans cette séquence par rapport aux autres acides aminés et par rapport aux moyennes de références (notamment l'alanine, la cystéine, l'isoleucine, la proline, et la thréonine). Au contraire, certains acides aminés comme le tryptophane ou l'histidine sont absents de notre protéine.

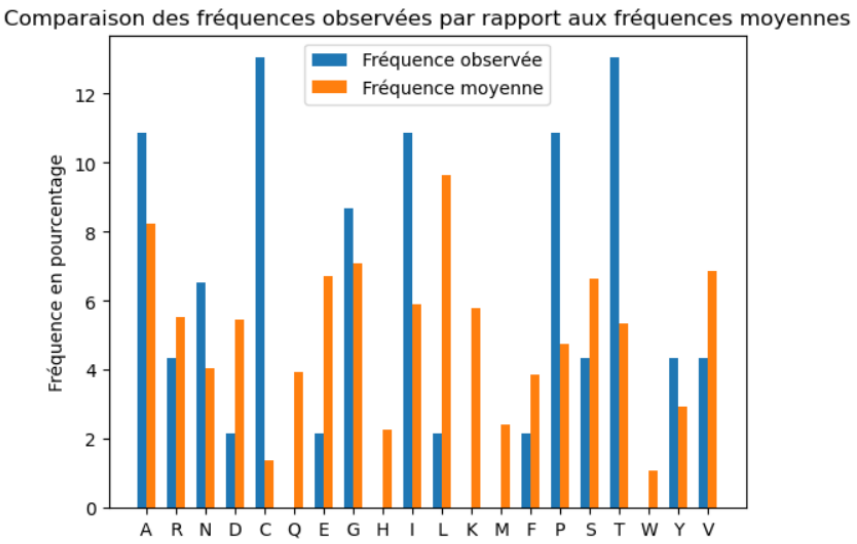



Figure 4 : Analyse de la composition en acide aminé de la protéine Crambine et comparaison avec les fréquences moyennes des acides aminés dans une protéine (Swissprot)

Certains acides aminés ont des caractéristiques utiles dans l'interprétation des structures tertiaires.

Notre protéine par exemple, a un **nombre significativement élevé de prolines** par rapport à la moyenne répertoriée dans Swissprot. Cela suggère la présence potentielle d'un nombre plus élevé de coudes, pouvant induire un changement de forme de notre protéine.

La cystéine est également un acide aminé important pour le repliement des protéines. Elle est connue pour sa capacité à former des ponts disulfures, qui sont des liaisons covalentes fortes. Ces ponts disulfures peuvent jouer un rôle crucial dans la stabilisation de la structure de certaines protéines, en verrouillant leur structure et en maintenant les liaisons entre différentes chaînes ou sous-unités. Nous pouvons constater une **fréquence significativement plus élevée de cystéines par rapport aux moyennes de référence**. Cette observation suggère que notre protéine est susceptible d'être très stable.

De plus, nous n'avons **pas de tryptophane dans notre protéine**. Etant donné que c'est un acide aminé qui encombrant, il peut entraîner un défaut de repliement. On peut émettre l'hypothèse que notre protéine a de faibles chances d'avoir un défaut de repliement.

### d) Localisation des ponts disulfures

Comme expliqué précédemment, les ponts disulfures sont des liaisons covalentes fortes entre deux cystéines, qui participent au repliement des protéines ou aux interactions protéine-protéine. On considère qu'un pont disulfure peut voir le jour entre deux cystéines si leur souffre se situe à une distance inférieure à 3 angströms.

Afin de déterminer les possibles ponts disulfures qui pourraient intervenir dans le repliement de la protéine, nous avons créé un dictionnaire avec pour clés, les positions des cystéines dans la séquence en acide aminé et pour valeurs, les coordonnés en x, y, z du soufre de celles-ci. Le parcours de ce dictionnaire permet de calculer la distance euclidienne entre deux atomes de soufre à partir de la formule suivante :

$$AB=\sqrt{((xB - xA)^2 + (yB - yA)^2 + (zB - zA)^2)}$$

En prenant en compte le fait qu'une cystéine ne puisse former qu'un seul pont disulfure, nous avons créé une liste contenant les cystéines impliquées dans les ponts disulfures, c'est-à-dire seulement si la distance entre celle-ci et un second soufre est inférieure à 3. L'utilisateur peut ainsi visualiser les positions des deux cystéines possiblement liées par un pont disulfure et connaître la distance qui les sépare.

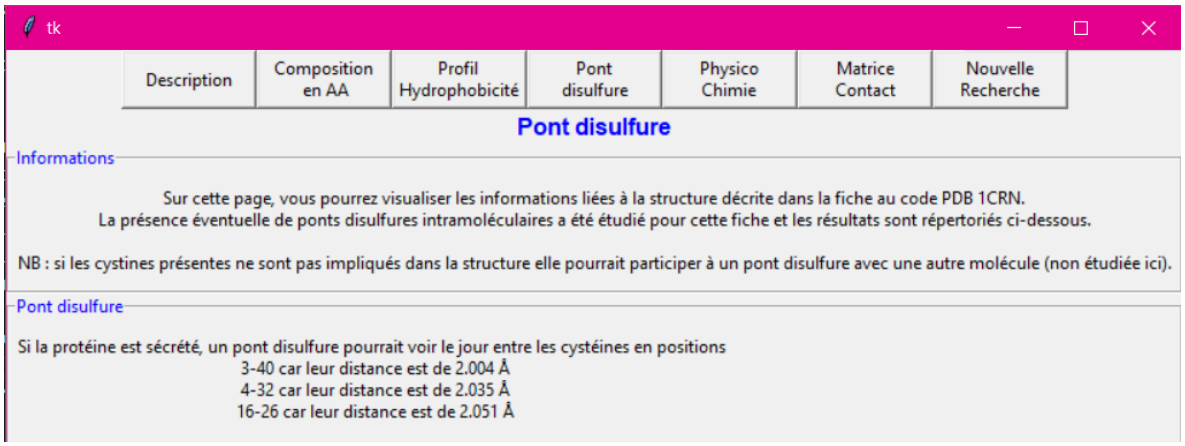



Figure 5 : Présentation de la visualisation des ponts disulfure de la protéine Crambine

Ici, si la protéine était sécrétée, **3 ponts disulfures** pourraient voir le jour intra-moléculairement et participeraient donc à un fort repliement de la protéine.

Les ponts disulfures pouvant également être construits entre deux protéines, nous avons choisi de présenter également la liste des positions des cystéines qui ne sont pas impliquées dans un pont disulfure (seulement s'il en existe).

Pour cela, nous parcourons le dictionnaire contenant l'ensemble des positions des atomes de soufres et listons celles dont les soufres ne sont pas impliqués dans un pont disulfure interne à la molécule. Dans l'exemple de la protéine 1CRN, il n'y avait pas de cystéines qui n'étaient pas impliquées dans un pont disulfure interne.

### e) Construction du profil d'hydrophobicité

Le profil d'hydrophobicité permet de cibler les zones hydrophobes et hydrophiles le long de la séquence d'acides aminés. Dans notre contexte, ces régions peuvent nous donner des informations sur la localisation de la protéine (membranaire, transmembranaire, cytosolique, etc…). Les profils d'hydrophobicité aident également à comprendre les interactions protéine-protéine et la formation de structures tertiaires.

Le calcul du profil d'hydrophobicité implique l'utilisation d'une échelle d'hydrophobicité pour attribuer une valeur à chaque acide aminé en fonction de sa tendance à être hydrophobe ou hydrophile. Nous avons pris pour référence l'**échelle de Fauchere et Pliska** (Eur. J. Med. Chem. 18:369- 375(1983)). La méthode consiste à calculer la moyenne des valeurs des acides aminés environnants en utilisant une fenêtre glissante de 9 acides aminés. Nous offrons l'opportunité à nos utilisateurs de changer la taille de cette fenêtre glissante s'ils en ont le besoin. Afin d'éviter toute erreur d'entrée, nous proposons un curseur placé par défaut sur 9. Le réglage par défaut permet aux utilisateurs ignorant les détails du calcul d'un profil d'hydrophobicité d'obtenir tout de même un résultat. Ce curseur est réglé avec un maximum égale à la taille de la séquence étudiée afin d'éviter toute erreur de dimension. A chaque déplacement du curseur, la page s'actualise et affiche le profil correspondant.

Ainsi, pour obtenir ce profil, nous avons parcouru la séquence d'acides aminés (codée avec le code à 1L) grâce à une fenêtre glissante associant chaque indice à un acide aminé et chaque acide aminé à une valeur d'hydrophobicité. La moyenne de ces valeurs est ensuite attribuée à la médiane de cette fenêtre par le calcul de la moyenne des indices de positions. En prenant l'entier supérieur pour l'indice, nous prenons en compte que la fenêtre glissante peut être une valeur pair ou impaire. Grâce aux listes de sorties de cette fonction, nous pouvons associer les positions du centre des fenêtres aux moyennes d'hydrophobicité et ainsi créer le tableau de valeur (que l'utilisateur peut télécharger grâce au bouton de l'interface sous le format xlsx). Ainsi, l'utilisateur peut directement visualiser le profil d'hydrophobicité et le reproduire à partir d'un tableau de valeur s'il en a le besoin.

Ce profil indique que notre peptide à un **indice d'hydrophobicité globalement élevé**. En effet, nous ne trouvons pas de valeur négative pouvant indiquer des régions hydrophile. Nous pouvons donc supposer que cette protéine est hydrophobe.

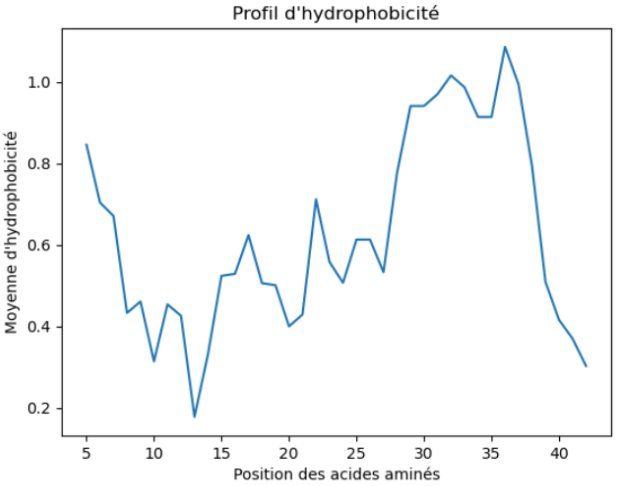



Figure 6 : Profil d'hydrophobicité de la protéine Crambine

De plus, on observe un grand pic entre les **positions 25 et 39** indiquant que les **acides aminés les plus hydrophobes** sont dans cette région. On aurait pu supposer que cette région correspond à une hélice alpha transmembranaire. Cependant, cela ne peut pas être le cas, car cette région contient moins de 20 acides aminés.

### f) Visualisation des propriétés physico-chimiques via le logiciel Pymol

Pymol est un logiciel de visualisation moléculaire largement utilisé dans le domaine de la biologie structurale. Il permet à partir d'un fichier PDB, d'analyser des structures moléculaires en 3D telles que des protéines, et de visualiser par gradient de coloration les atomes selon des indicateurs tel que le Bfactor. Le Bfactor est une valeur comprise entre 0 et 999 contenue dans la fiche PDB, qui rend compte de l'agitation thermique des atomes. C'est un indicateur de la mobilité des atomes dans une structure cristallographique. Le principe de notre manoeuvre a été ici d'utiliser la donnée du Bfactor dans le fichier PDB; de modifier sa valeur selon d'autres échelles pour visualiser d'autres caractéristiques des protéines comme les propriétés physico-chimiques des acides aminés, leur fréquence ou encore leur poids moléculaire. L'échelle de coloration va de **0,00 (bleu) à 999,99 (rouge)**. Afin d'optimiser la coloration de la molécule, nous avons étalé les valeurs de fréquences et du poids moléculaires sur l'échelle de coloration. Pour ce faire, nous avons multiplié la fréquence de chaque acide aminé par 50 et le poids moléculaire par 4,5. Pour la visualisation selon leur propriété physico-chimique, nous avons attribué une valeur à l'acide aminé selon les catégories suivantes :

1 = polaires_non_charges = SER, THR, ASN, GLN, CYS
200 = polaires_acides = ASP, GLU
400 = polaires_basiques = LYS, ARG, HIS
600 = apolaires_non_aromatiques = GLY, ALA, VAL, LEU, ILE, PRO, MET
800 = apolaires_aromatiques = PHE, TYR, TRP

L'attribution des classes a été réalisée selon la composition des acides aminés (en fonction des groupements présents sur la chaîne latérale, leur hydrophobicité, la possibilité de faire des liaisons avec une molécule d'eau ou encore la présence d'un cycle aromatique ou non).

Pour chaque acide aminé, nous avons donc attribué une nouvelle valeur au B-factor selon sa classification et modéliser sur Pymol la nouvelle coloration de la molécule.

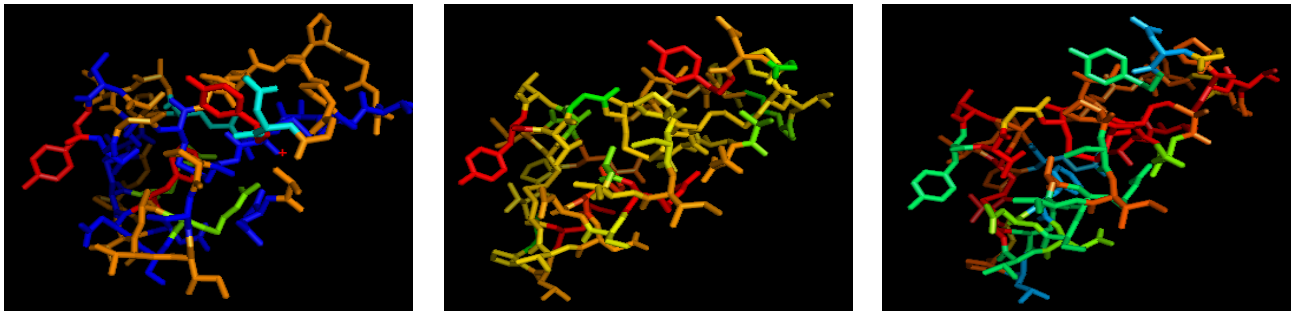
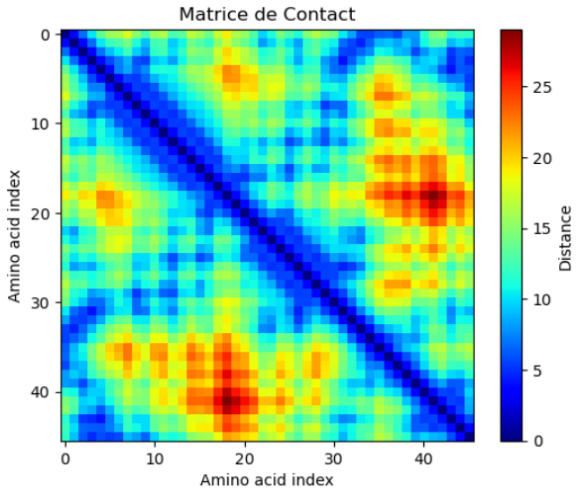


Figure 7 : Visualisation respectivement de la polarité, du poids moléculaire et de la fréquence de chaque acide aminé de la protéine Crambine avec le logiciel Pymol

### g) Hypothèse de repliement grâce à la matrice de contact

Dans la poursuite de l'étude de la structure d'une protéine et principalement de ces replis, la matrice de contact peut être d'une grande utilité. En effet, une matrice de contact est une représentation matricielle des interactions entre les résidus d'une protéine tridimensionnelle qui permet de rendre compte de la distance entre les différents acides aminés d'une protéine grâce à une échelle de couleur. Celle-ci va du bleu (acides aminés très proches entre eux) au rouge (acides aminés très éloignés entre eux). En analysant les distances entre les acides aminés et en associant ces données aux informations déjà collectées, il est possible de déduire les interactions potentielles (telles que les interactions de Van der Waals, les liaisons hydrogène, etc.) entre différents acides aminés et ainsi, d'en apprendre davantage sur le repliement de la protéine.

Pour obtenir la matrice de contact, nous avons calculé la distance entre tous les carbones alpha de notre protéine (en prenant en compte les coordonnées dans l'espace, c'est-à-dire selon les axes x, y et z). Les distances ont été calculées avec la même formule que celle utilisée pour les ponts disulfure. Nous avons ensuite résumé toutes ces distances dans un tableau et avons représenté ce tableau à l'aide de la fonction imshow().




On observe une grande zone bleue traduisant un **rapprochement des régions 1- 5 et 30-46**.
D'après les informations collectées, un pont disulfure pourrait voir le jour entre les cystéines en positions 3 et 40, et un autre entre les cystéines 4 et 32, ce qui pourrait expliquer le rapprochement spatial de ces régions.

Une zone bleue plus petite est présente entre les régions **22-28 et 12-16**. Cela pourrait être expliqué par le dernier pont disulfure possiblement présent dans cette protéine entre les cystéines 16-26.

Figure 8 : Matrice de contact de la protéine Crambine

On peut aussi remarquer que **les acides aminés 5 à 7 sont éloignés des acides aminés 17 à 21** (zone rouge). Pareillement, **les acides aminés 15 à 21 et 35 à 46, sont très éloignés entre eux**. Il est difficile d'interpréter cet éloignement. On peut simplement conclure que ces zones sont sûrement éloignées à cause des différents coudes dans la molécule, causées par les prolines 5, 19, 22, 36 et 41 (nous avions vu précedemment à l'aide de l'histogramme que notre protéine contenait beaucoup de proline), ainsi que des ponts disulfures, qui ont donné une structure tertiaire spécifique à cette protéine.

L'ensemble des résultats d'analyse et des figures peuvent être enregistrés dans un fichier de sortie téléchargeable au format Word (pas simplement avec un fichier texte), permettant ainsi de visualiser les graphiques. Le nom du fichier utilisé par défaut est "Analyse_{Code_PDB}.docx".

## 3. Organisation du travail

Pour mener à bien un projet d'une telle envergure, une organisation minutieuse est indispensable afin d'optimiser notre temps.

Dès le début du projet, nous avons créé un document collaboratif (Google Colab). Ce document partagé recensait les fonctions sur lesquelles nous travaillions ou celles que nous avions terminées. Cette approche s'est avérée pratique, car elle nous permettait de nous adapter à nos emplois du temps respectifs, qui nous empêchaient de nous réunir régulièrement. Chacun pouvait contribuer au projet lorsqu'il disposait de temps libre. De plus, lors de la rédaction des fonctions, nous avons pris l'habitude d'ajouter de nombreux commentaires liés aux erreurs ou des remarques sur le code, facilitant ainsi la relecture et limitant la perte de temps liée à la compréhension de ce qui avait déjà été réalisé.

Une fois le projet bien avancé, nous avons pu nous répartir les dernières fonctions à créer ou à améliorer. Et, en fin de projet, nous avons créé l'interface.

Le **point fort** de notre groupe a été la **communication**. Les visioconférences nous ont permis de discuter de l'avancement du projet, de l'aspect biologique des résultats, et surtout d'expliquer aux autres membres du groupe ce que nous avions déjà réalisé concernant les dernières fonctions réparties. Grâce aux visioconférences, nous étions autonomes et avions tous une compréhension globale du projet. Cette approche nous a permis de rédiger rapidement le rapport et le diaporama pour la présentation orale.

## 4. Gestions des Erreurs

L'utilisation de l'interface a permis d'éviter certains problèmes, tels que les **erreurs liées à l'entrée d'informations de l'utilisateur et aux chemins d'accès des fichiers**. En effet, il est facile de commettre une faute de frappe sans s'en rendre compte. L'ouverture d'un gestionnaire de fichiers pour récupérer le chemin d'accès évite tout problème d'input (faute de frappes ou dossier introuvable). Le gestionnaire impose aussi le **type de fichier sélectionnable** (.pdb). Pour le chargement en ligne, seuls les codes pdb sont pris en compte. La fonction .upper() permet de ne pas prendre en compte la casse de l'input. Une fenêtre d'erreur s'affiche si le code entré ne correspond pas à une page ou si la connexion internet n'est pas bonne. Cette fenêtre ne bloque pas l'utilisation du code et une nouvelle recherche peut être lancée par la suite. Enfin, si cette étape est réussie, le bloc try, except, else permet d'exécuter le reste du code que.

Dans le codage des fonctions, nous avons également pris en compte que le **titre du fichier pouvait être écrit sur deux lignes**, que les **acides aminés pouvaient être écrits avec plus de 3 lettres** et qu'il pouvait y avoir des **informations manquantes pour l'écriture du header** dans le fichier fasta. Nous avons également vérifié que l'absence d'un acide aminé ne gênait pas l'analyse de la séquence. Par exemple, l'absence d'un acide aminé comme le tryptophane pour cette protéine ne bloque par la comparaison des fréquences d'acides aminés.

Enfin, lorsque l'on lance une nouvelle recherche, il n'y a **pas de fermetures automatiques de la page de gestion**. Toutefois, cette action était nécessaire afin d'écraser les valeurs des variables globales. Dans le cas échéant, nous n'avions pas pas d'affichage des fenêtres comprenant des figures ("Composition en AA" et "Profil d'Hydrophobicité").

## 5. Difficultés rencontrées

Dans un premier temps, la **récupération des données** a été compliquée. Nous avons dû étudier attentivement la structure d'un fichier PDB pour extraire les informations attendues. Parfois, l'output attendu n'était pas clair, ce qui nous a amenés à demander des précisions lors des séances en classe.

La **création de l'interface** nous a permis d'effectuer au mieux la gestion des erreurs. Cependant, l'utilisation de modules inconnus pour nous, nous a retardé dans l'écriture du script. En effet, nous avons rencontré de nombreuses erreurs de codage basiques dues à notre méconnaissance du module Tkinter. Bien que ces erreurs aient diminué au fur et à mesure de notre utilisation du module, elles ont pris une part non négligeable de notre temps sur le projet.

Nous avons rédigé tout notre code sur un seul document, ce qui a nécessité une gestion des variables locales et globales minutieuse. Certains problèmes étaient liés à la **définition des variables.**

Enfin, une part non négligeable du temps perdu lors de la réalisation de ce projet a été liée aux **versions de python utilisées, à l'environnement de travail utilisé, ainsi qu'à l'installation de module**. Les erreurs sont survenues lors de l'exécution du code en raison de versions différentes de Python sur nos ordinateurs respectifs. Malgré l'installation des versions les plus récentes de Python, l'installation des modules a posé de nombreux problèmes, avec des messages d'erreur constants lors de l'utilisation de la fonction "pip install". De plus, selon que nous utilisions IDLE, Jupiter Lab ou autre, certaines bibliothèques comme numpy ont généré des messages d'erreur. La résolution de l'ensemble de ces difficultés a été chronophage.

## 6. Conclusion et pistes d'amélioration du code

En résumé, ce programme permet à l'utilisateur d'accéder plus facilement et plus rapidement aux caractéristiques structurales essentielles d'une protéine. La mise en relation de l'ensemble des données fournies par ce programme permet d'avoir une première idée des éventuels comportements de la protéine dans un milieu biologique (repliement, localisation,...).

Pour améliorer le code, il serait utile d'ajouter une fonction permettant d'extraire les informations sur la **sécrétion d'une protéine,** à partir d'une fiche uniprot. Cette fonction pourrait être reliée à notre fonction sur les ponts disulfures afin de préciser à l'utilisateur si les ponts disulfures prédit peuvent réellement exister. Une autre fonction pourrait être ajoutée afin de **récupérer les données de fréquence moyenne de chaque acide aminé directement depuis internet**. Cette fonction permettrait ainsi d'avoir une mise à jour constante des fréquences moyennes de référence. Nous pourrions également améliorer ce programme en **ouvrant automatiquement Pymol lorsque l'utilisateur demande de visualiser la molécule**. Par ailleurs, nous imposons les données recherchées et donc les résultats obtenus. Offrir à l'utilisateur l'opportunité de choisir les analyses qu'il souhaite réaliser pourrait être une amélioration envisageable. Enfin, ce projet se concentre sur l'analyse de la structure tertiaire d'une protéine à partir d'un fichier PDB. Toutefois, la **structure secondaire, dont nous ne parlons pas ici, peut** donner des informations sur le repliement de la protéine.

Pour conclure, ce projet a renforcé nos compétences en Python et en biologie structurale, grâce à la découverte de nouveaux modules, d'une nouvelle manière de présenter les résultats et une compréhension plus claire et enrichissante de la relation entre les analyses biologiques.