Université Paris Cité – UFR Sciences du Vivant

Meriem YOUSSEF

Dépôt GitHub : https://github.com/Meriemyssf/Projet_Court

ASSIGNATION ET DETECTION DES PARTIES TRANSMEMBRANAIRES D'UNE PROTEINE

UE : Programmation avancée et gestion de projets informatiques

—

Référents : **Monsieur Jean-Christophe Gelly**
Madame Tatiana Galochkina

2024-2025

1. Introduction :

Les protéines transmembranaires (TMPs) sont localisées dans les bicouches lipidiques des membranes plasmiques et jouent un rôle crucial dans divers processus biologiques, tels que le transport d'ions, la signalisation cellulaire et le maintien de l'équilibre intracellulaire/extracellulaire. Ces protéines font partie des trois grandes classes de protéines : fibreuses, globulaires et membranaires. Leur identification demeure cependant complexe, en raison des difficultés rencontrées pour résoudre leur structure dans les membranes. Pour répondre à ce défi, G.E. Tusnády et al. ont développé, en 2004, l'algorithme TMDET, conçu pour identifier et classifier les TMPs à partir de la Protein Data Bank (PDB) (1) (2)

Le présent projet vise à implémenter un outil inspiré de cet algorithme TMDET, capable de déterminer les régions transmembranaires d'une protéine en se basant sur son aspect géométrique et son hydrophobicité en maximisant le nombre de résidus hydrophobes et exposés au solvant entre deux plans parallèles représentant la membrane. Nous nous concentrerons sur l'identification de la position probable de la membrane, sans aborder les aspects structuraux de la fonction objective de l'article original.

2. Matériels et méthodes :

Algorithme :

L'objectif principal de cet algorithme est de déterminer la position optimale d'une membrane, représentée par deux plans parallèles, dans un espace tridimensionnel pour une protéine donnée. Cette position optimale est définie comme celle où l'agencement des deux plans maximise l'hydrophobicité relative. En d'autres termes, il s'agit de maximiser le nombre de résidus hydrophobes exposés au solvant entre les deux plans.

Pour atteindre cet objectif, plusieurs étapes ont été suivies :

1. **Importation des bibliothèques nécessaires :** Utilisation d'argparse pour la gestion des arguments en ligne de commande, Bio.PDB pour le parsing et l'analyse des fichiers PDB, numpy et math pour les calculs numériques, pymol pour la visualisation 3D des structures, ainsi que copy, sys, et os pour la gestion des objets, des systèmes et des fichiers.

2. **Validation et Extraction des données :** Les fichiers PDB sont vérifiés pour s'assurer qu'ils contiennent la chaîne 'A'. Les fichiers valides sont ensuite analysés pour extraire les coordonnées des atomes de carbone alpha de chaque acide aminé. Les acides aminés sont ensuite classifiés en fonction de leur hydrophobicité. Les résidus considérés comme hydrophobes sont les mêmes que dans l'article (1). Ces informations sont stockées et organisées pour faciliter et simplifier l'exécution du programme.

3. **Calcul de la surface accessible au solvant :** La surface accessible relative (ASA) de chaque acide aminé est déterminée à l'aide de la fonction DSSP du module Biopython(3)(4). Seuls les résidus ayant une ASA supérieure à 30% sont retenus pour les analyses ultérieures.

4. **Calcul du Centre de Masse :** Le centre de masse de la chaîne « A » est calculé à l'aide de la méthode "center_of_mass". À partir de ce point, N points sont générés uniformément sur une demi-sphère de rayon 1, avec N déterminé par l'utilisateur (par défaut : 20), en utilisant l'algorithme de Staff and Kuijlaars (5). Ces points

servent à créer des vecteurs entre le centre de masse et les points de la demi-sphère.

5. **Détermination des Plans :** Pour chaque vecteur, deux plans orthogonaux sont générés (6), séparés par un écart prédéfini (15 Angstroms par défaut) représentant la membrane. Ces informations sont stockées dans « axes ». Pour chacun de ces axes, les deux plans glissent au-dessus et en dessous. Après chaque glissement (1 Angström par défaut), l'hydrophobicité relative des résidus entre les deux plans est calculée en utilisant la formule suivante :

$$hydrophobicity = \frac{n\ polar\ out\ of\ planes}{n\ polar\ residues} + \frac{n\ hydrophobic\ between\ planes}{n\ hydrophobic\ residues}$$

6. **Optimisation :** Si l'hydrophobicité relative est supérieure à celle du meilleur axe trouvé jusqu'à présent, cet axe cet axe devient la nouvelle référence. Cette procédure est répétée pour chaque direction de la demi-sphère jusqu'à ce qu'il n'y ait plus de résidus entre les deux plans, en mettant à jour la valeur du meilleur axe si nécessaire. Ce glissement est également effectué dans l'autre direction de l'axe. À la fin, pour chaque axe, la meilleure position des plans est conservée avec l'hydrophobicité relative associée. Enfin, la largeur de la membrane est ajustée en en déplaçant les plans au-dessus et en dessous pour chaque position pour maximiser l'hydrophobicité relative jusqu'à ce que tous les résidus soient pris en compte.
7. **Visualisation avec PyMol :** Les résultats finaux, incluant la position des plans transmembranaires et la structure de la protéine, sont visualisés en 3D à l'aide de PyMol.

Jeu de données :

Le programme a été testé sur les six protéines suivantes : 2l34, 2llm, 2lly, 1qjp, 1prn, 4grv comportant respectivement 33, 43, 137, 171, 289, 510 résidus. Ces protéines sont considérées comme des protéines membranaires de référence.

Données d'entrée :

Étant donné qu'une protéine est simplifiée ici comme une chaîne unique, nous avons utilisé uniquement des protéines transmembranaires petites et simples pour évaluer notre outil.

L'utilisateur du programme peut définir plusieurs paramètres :

- **Nombre de points initiaux** : Le nombre de points uniformément répartis sur une demi-sphère pour générer les vecteurs de direction. La valeur par défaut est 20 Å.
- **Largeur de la membrane initiale** : L'épaisseur des deux plans parallèles qui définissent la membrane. La valeur par défaut est 15 Å.
- **Taille de la fenêtre de glissement le long de l'axe** : L'écart par lequel les plans sont déplacés lors de l'exploration de l'axe. La valeur par défaut est 1 Å.

Ces paramètres permettent à l'utilisateur de moduler la précision et la performance du programme en fonction de ses besoins spécifiques.

Données de sortie :

Le programme ouvre PyMol (7) avec la protéine, les plans prédits, la sphère et son centre, ainsi que le centre de masse de la protéine, tous représentés comme objets

hétéroatoms. La session PyMol contenant ces éléments est sauvegardée dans un fichier avec l'extension ".pse".

3. Résultats et discussion :

Le temps d'exécution de l'algorithme varie selon la taille de la protéine, allant de quelques minutes pour les petites protéines à plusieurs minutes pour les plus grandes.

La première analyse des résultats consiste en la visualisation des régions transmembranaires à l'aide de PyMOL. Le programme parvient à afficher la protéine avec son centre de masse, les deux plans ainsi que la sphère avec son centre, prouvant que le programme est fonctionnel.

Pour vérifier l'exactitude de l'orientation et de la position des membranes prédites, nous avons comparé les résultats obtenus avec les fichiers PDB issus de la base de données Orientations of Proteins in Membranes (OPM) (8).

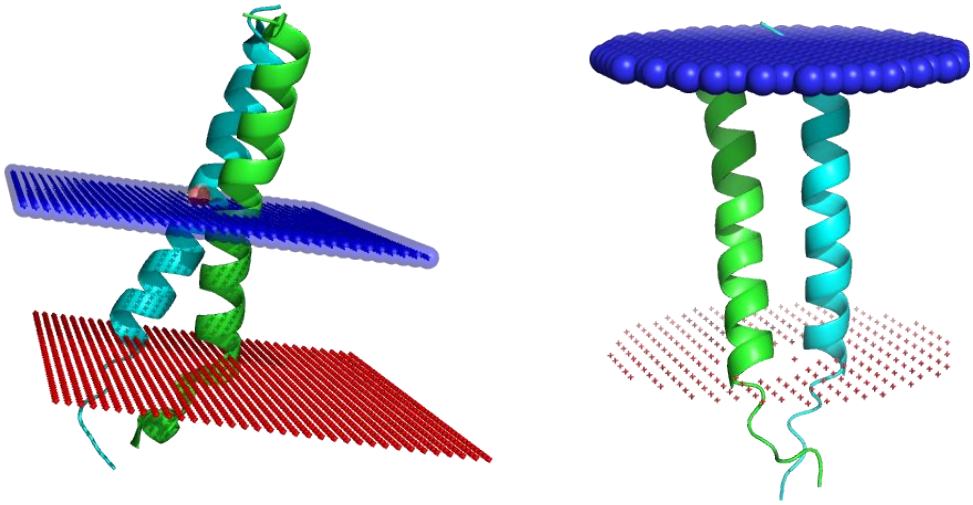

Figure.1 : Comparaison entre les résultats de notre programme (à gauche) et le fichier PDB fourni par la base de données OPM (à droite) pour la protéine 2l34

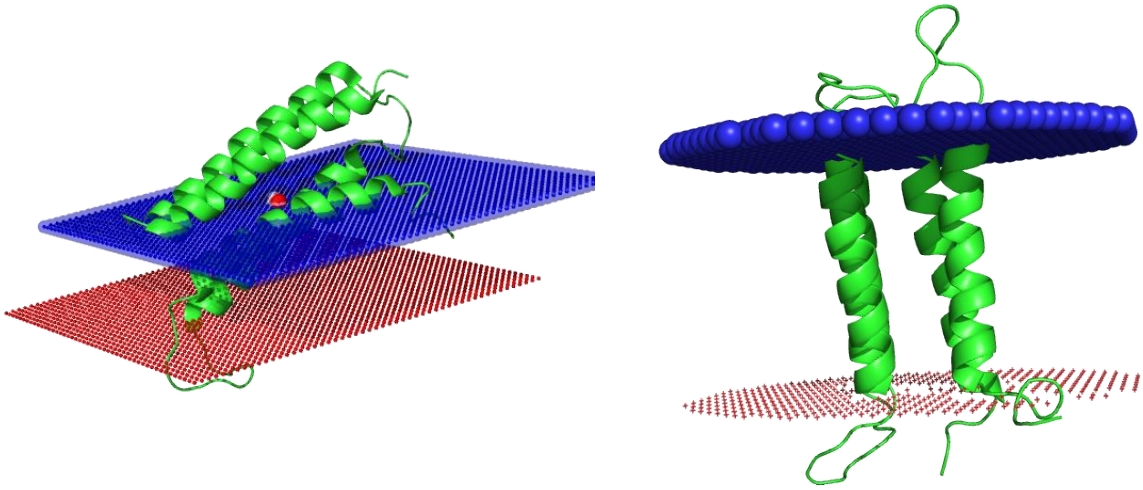

Figure.2 : Comparaison entre les résultats de notre programme (à gauche) et le fichier PDB fourni par la base de données OPM (à droite) pour la protéine 2lly

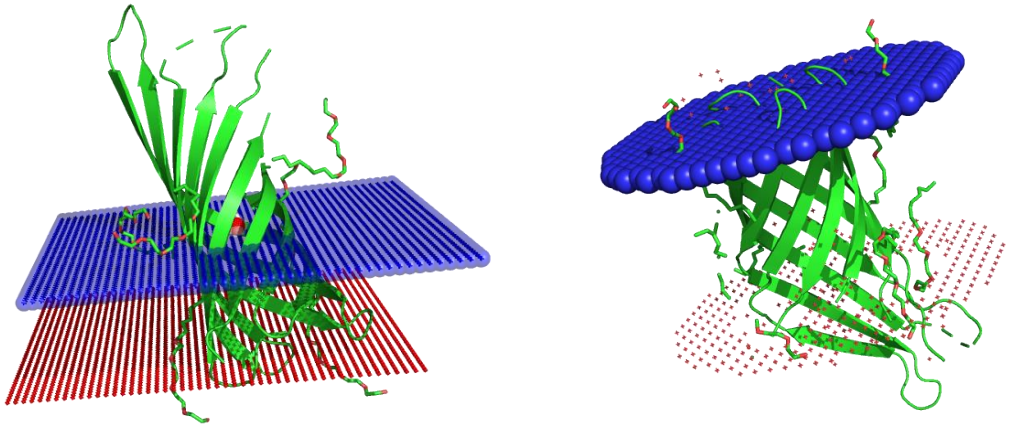

Figure.3 : Comparaison entre les résultats de notre programme (à gauche) et le fichier PDB fourni par la base de données OPM (à droite) pour la protéine 1qjp

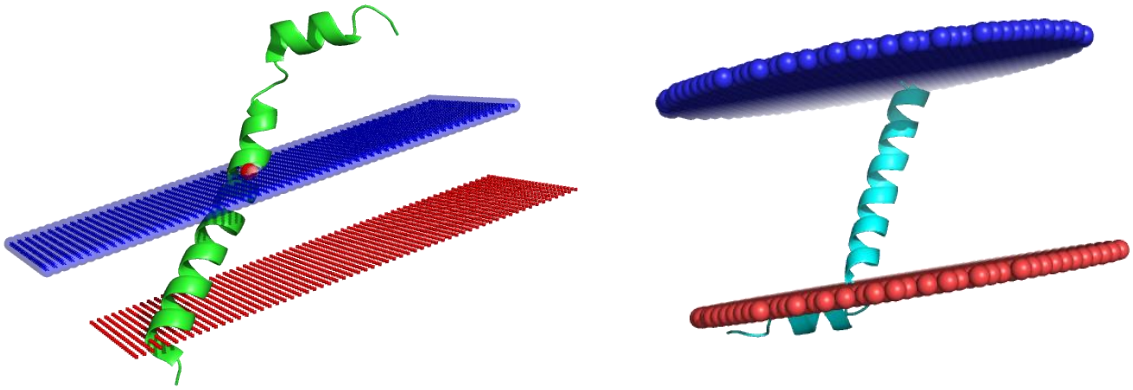

Figure.4 : Comparaison entre les résultats de notre programme (à gauche) et le fichier PDB fourni par la base de données OPM (à droite) pour la protéine 2llm

Nous avons comparé les résultats pour plusieurs protéines : TYRO protein tyrosine kinase-binding protein (PDB : 2l34) en Figure.1, Neuronal acetylcholine receptor subunit alpha-4 (PDB : 2lly) en Figure.2, outer membrane protein A (PDB : 1qjp) en Figure.3 et Amyloid beta A4 protein (PDB : 2llm) en Figure.4.

Les résultats obtenus présentent des incohérences par rapport aux prédictions fournies par la base de données OPM notamment concernant l'orientation et le positionnement des membranes Le plan inférieur (en rouge) semble isoler de manière relativement correcte les résidus situés en dessous, bien que ce ne soit pas aussi précis que les résultats de la base OPM. En revanche, le plan supérieur (en bleu) montre des prédictions totalement erronées, avec une mauvaise délimitation des résidus hydrophobes et un positionnement significativement éloigné des résultats attendus.

Plusieurs facteurs peuvent expliquer ces divergences. Premièrement, l'algorithme générant les vecteurs directeurs à partir du centre de masse de la protéine pourrait comporter des erreurs ou des approximations, affectant ainsi l'échantillonnage des directions et la qualité des prédictions. En outre, le calcul de l'hydrophobicité des résidus pourrait ne pas être optimal, influençant la détection des résidus transmembranaires. De plus, les paramètres d'entrée, tels que le nombre de points, la largeur de la membrane et la taille de la fenêtre de glissement, influencent directement la précision des résultats. Des ajustements inadéquats de ces paramètres pourraient expliquer les écarts constatés.

En outre, le manque de temps pour effectuer des tests approfondis et optimiser ces paramètres a limité la capacité à affiner les prédictions. Une vérification complète du code, des formules et de l'algorithme de génération des points sur la sphère est

nécessaire. Avec une optimisation plus rigoureuse de ces éléments, il serait possible de réduire les écarts observés et d'améliorer la précision globale des résultats.

- **Difficultés rencontrées :**

- Complexité du sujet, avec un temps limité pour approfondir tous les aspects.

- Utilisation de la géométrie dans l'espace, nécessitant une révision des concepts de lycée (détermination d'un vecteur à partir de deux points, équation cartésienne d'une droite, calcul de la distance entre deux plans).

- Positionnement des points sur la sphère, où il a fallu trouver et adapter un algorithme existant (l'algorithme de Staff and Kuijlaars).

- Affichage des résultats et utilisation de PyMOL pour la visualisation des structures en 3D.

4. Conclusion :

En conclusion, ce projet a permis de développer un outil capable d'identifier les zones transmembranaires d'une protéine en se basant sur son hydrophobicité et sa géométrie, inspiré de l'algorithme TMDET de Tusnády et al. Malgré l'exécution du programme, des divergences ont été observées par rapport à la base de données OPM, notamment concernant l'orientation et le positionnement des membranes. Ces incohérences pourraient être dues à des approximations dans l'algorithme de génération des points ainsi qu'à des ajustements insuffisants des paramètres d'entrée.

La complexité de la géométrie dans l'espace, associée au manque de temps pour des tests approfondis et l'optimisation des paramètres, a limité la précision des résultats. Une révision approfondie de l'algorithme, ainsi qu'une meilleure gestion des paramètres, pourraient permettre d'améliorer la qualité des prédictions et les aligner davantage avec les données de référence d'OPM. Ce travail ouvre ainsi la voie à des optimisations futures pour mieux comprendre et prédire les structures des protéines transmembranaires.

Références :

1. Tusnády GE, Dosztányi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics. 22 nov 2004;20(17):2964-72.

2. Tusnády GE, Dosztányi Z, Simon I. TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates. Bioinformatics. 1 avr 2005;21(7):1276-7.

3. Bio.PDB.DSSP.DSSP – biopython v1.71.0 [Internet]. [cité 12 sept 2024]. Disponible sur: https://homolog.us/Biopython/Bio.PDB.DSSP.DSSP.html

4. DSSP [Internet]. [cité 12 sept 2024]. Disponible sur: https://swift.cmbi.umcn.nl/gv/dssp/

5. Saff EB, Kuijlaars ABJ. Distributing many points on a sphere. Math Intell. déc 1997;19(1):5-11.

6. Weisstein EW. Plane [Internet]. Wolfram Research, Inc.; [cité 12 sept 2024]. Disponible sur: https://mathworld.wolfram.com/

7. PyMOL | pymol.org [Internet]. [cité 12 sept 2024]. Disponible sur: https://www.pymol.org/

8. OPM [Internet]. [cité 12 sept 2024]. Disponible sur: https://opm.phar.umich.edu/