



UFR SCIENCES DU VIVANT

MASTER DE BIO-INFORMATIQUE

Rapport Projet court
Assignation et détection des parties
transmembranaires d'une protéine

Maya Zygodlo

Encadrant : Tatiana Galochkina

Co-encadrant : Jean-Christophe Gelly

Maya ZYGADLO

14 Septembre 2022

Table des matières

1	Introduction	1
1.1	Informations manquantes lors de la cristallisation d'une protéine transmembranaire .	1
1.2	Différentier les protéines membranaires et globulaires	1
1.3	Objectif	1
2	Matériel et Méthode	1
2.1	Molécules tests	1
2.2	Techniques / Technologie	2
2.3	Conception de l'algorithme	2
3	Résultats	3
4	Discussion	3

1 Introduction

1.1 Informations manquantes lors de la cristallisation d'une protéine transmembranaire

Les protéines transmembranaires sont sous-représentées dans les bases de données structurales comme la Protein Data Bank (PDB). Ceci est dû au fait qu'il est difficile de les cristalliser comme des protéines solubles. En effet, lors de la cristallisation, ces protéines ne sont pas dans leur milieu naturel, la membrane, et peuvent donc présenter des conformations différentes de celle qu'elles adoptent naturellement. Pour compenser ceci, elles sont mis en contact avec des détergents [1]. Cependant, cette solution n'est pas parfaite car, bien que les détergent soient des espèces amphiphiles formant des micelles et des membranes, celles-ci présentent des architectures différentes.

De plus, la position de la membrane n'est souvent pas indiquée et laisse alors une partie importante des informations sur ces protéines de côté [2].

1.2 Différentier les protéines membranaires et globulaires

Pourtant ces protéines sont essentielles au bon fonctionnement d'un organisme. En effet, elles représentent entre 20 à 30 % des séquences protéiques [3] et ont un rôle clef dans la communication cellulaire et le transport de molécules. Pour cela, ce sont des cibles préférentielles pour la recherche de nouveaux médicaments pharmaceutiques. Il est donc important d'avoir des informations sur la position de la membrane.

Cependant, avec seulement une information de séquence, il est difficile de distinguer les protéines transmembranaires des protéines solubles, permettant de placer des membranes sur les bonnes protéines [4]. En effet, en ayant que la polarité des acides aminés d'une protéine, nous ne pouvons distinguer une protéines solubles d'une protéine transmembranaire car les protéines solubles possèdent des acides aminés apolaire et les protéines transmembranaires possèdent des parties polaires permettant des liaisons au ligand ou étant le siège d'un site enzymatique.

1.3 Objectif

Il est donc intéressant de créer un algorithme capable, à partir d'une information de séquence trouvable dans un fichier PDB, de trouver la position la plus probable de la membrane.

Cet algorithme a déjà été conçu par une équipe de chercheurs (Tusnády et al. [5]) qui ont ensuite créé une base de donnée répertoriant les protéines avec l'information du placement de la membrane. Nous allons ici tenté de produire notre propre algorithme afin de comparer les résultats obtenus avec ceux présent dans la base de données OPM.

2 Matériel et Méthode

2.1 Molécules tests

Pour cela, nous avons effectuer nos tests sur 3 protéines, deux transmembranaire et une globulaire. Les deux protéines transmembranaires varient par rapport à l'endroit où la membrane est positionnée.

Nous avons ainsi travaillé avec le segment transmembranaire de l'intégrine α IIB humaine, trouvable sur le site PDB sous l'identifiant 2K1A [6]. Cette protéine de 42 acides aminés est présente dans les membranes plasmiques eucaryotes et a été résolue par RMN et ne présente aucune mutations.

Et, nous avons travaillé avec la protéine globulaire Hémoglobine A2 humaine, trouvable sur le site PDB sous l'identifiant 1SI4 [8]. Cette protéine présente 2 paires de chaînes de 141 et 146 acides aminés qui ont été résolues par rayon X avec une résolution de 2.20 Å et ne présente pas de mutations.

2.2 Techniques / Technologie

L'algorithme créé est disponible sur GitHub à l'adresse [suivante](https://github.com/CNOV0/Projet_Management) :

`https://github.com/CNOV0/Projet_Management`

Il a été écrit à l'aide du langage de programmation Python3 et du gestionnaire d'environnement Conda. Un README est disponible sur la page GitHub pour vous permettre de le tester.

L'environnement Conda comporte les modules suivants :

BioPython, Pandas, NumPy, matplotlib

Il requière également le logiciel DSSP [9] qui sera installé par salilab depuis l'environnement Conda et utilisé par le module BioPython.

2.3 Conception de l'algorithme

Cet algorithme prend en entrée le fichier PDB d'une protéine et rend un graphique sous forme PNG de la protéine avec l'endroit où se situe les deux feuillets de la membrane.

Pour cela, l'algorithme va lire le fichier PDB et le passer au logiciel DSSP depuis le module BioPython pour obtenir un score d'accessibilité au solvant.

Une fois cela fait, nous sélectionnons les carbones α de la protéine qui sont accessibles au solvant (ayant un score supérieur à 0.5). Ces carbones seront sauvegardés dans un dataframe Pandas avec leurs scores, le nom des résidus auxquels ils appartiennent et leurs coordonnées.

Dans un second temps, nous centrons notre protéines en (0, 0, 0) en calculant son centre de masse et en le soustrayant aux coordonnées des carbones α .

Dans un troisième temps, nous générons une hémisphère de rayon 1 Å. Présentant à sa surface des points équidistant. Cette hémisphère suit ainsi l'algorithme de Fibonacci pour placer les points uniformément. Cette partie de l'algorithme a été inspiré d'un existant et trouvé sur le site suivant [10]. Nous avons choisi de créer une hémisphère de 10 points car nous n'avons pas pu paralléliser l'étape suivante.

Dans un quatrième temps, à partir de chaque point de l'hémisphère, nous traçons une droite qui part de ce point vers le centre et plaçons deux plans perpendiculaires à la droite et parallèles entre eux de 22 Å. Le choix de 22 Å vient directement de l'article de Tusnady et al. [5] qui conseillent une membrane de 22 Å pour une protéine composée seulement de carbone α . Pour calculer ces plans nous utilisons le calcul suivant :

$d = x_sphere * x_point + y_sphere * y_point + z_sphere * z_point$

Sachant que pour séparer de 22 Å nous augmentons les coordonnées de point avec le calcul :

$x_point = x_point + x_sphere * 22$

Ces deux plans parallèles parcourent ensuite la droite du plus petit point (xmin, ymin, zmin) au plus grand (xmax, ymax, zmax) par pas de 1 Å. Pour chaque tranche est calculé le score d'hydrophobicité, c'est-à-dire le ratio entre le nombre d'acide aminés hydrophobes sur le nombre d'acide aminés. Cela est fait en comparant directement les résultats **d_carbone** avec **d_plan1** et **d_plan2**.

Le meilleur score est ainsi stocké dans un dictionnaire avec les coordonnées de ces deux plans.

Finalement, un graphique est tracé et sauvegarder dans le fichier *results* pour visualiser notre protéine et la membrane.

3 Résultats

Suite au lancement de l'algorithme, nous obtenons des graphiques permettant de visualiser notre protéine et où se situe chacun des feuillets de la membrane.

Pour nous assurer de nos résultats, nous les comparons à ceux obtenus dans OPM, une base de données regroupant les protéines transmembranaires et la position de la membrane par rapport à celles-ci. Ainsi, nous obtenons les résultats suivants :

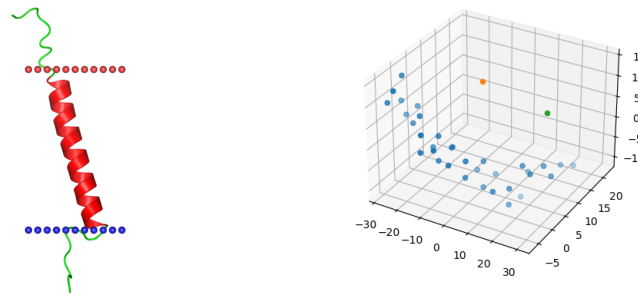


FIGURE 1 – Position de la protéine 2K1A par rapport à la membrane obtenus dans OPM [11]. Position de la protéine 2K1A (points bleu) par rapport à la position de la membrane (point orange : feuillet bas, vert : feuillet haut) obtenu par notre algorithme.

Les feuillets de la membrane sont perpendiculaires à la droite passant chacun des deux points. Ainsi, La figure 1 B démontre une membrane passant latéralement au centre de la protéine.

Nous obtenons alors un résultat proche de celui trouvé dans la base de données OPM. Cela confirme alors nos résultats pour une protéine transmembranaire.

Nous nous intéressons à présent à une protéine soluble. Nous nous attendons à ce que la membrane ne passe pas par cette protéine.

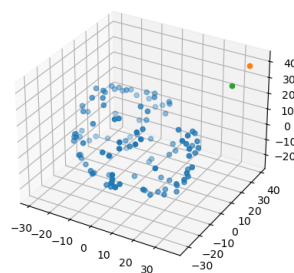


FIGURE 2 – Position de la protéine 1SI4 (points bleu) par rapport à la membrane (point orange : feuillet bas, vert : feuillet haut). Calculé à partir de notre algorithme.

La figure 2 démontre une membrane ne passant pas pas la protéine. Cela est cohérent avec les résultats attendus car une protéine non transmembranaire ne rentre pas en contact avec le membrane.

4 Discussion

Nous avons pu constater que notre algorithme produisait des résultats satisfaisants pour les protéines transmembranaires et solubles. Ainsi nous avons pu construire un algorithme capable de prédire la

position de la membrane avec une certaine précision.

Toutefois, il aurait été intéressant de pouvoir faire varier la taille de la membrane afin d'être davantage précis.

De plus, il aurait été bien de pouvoir paralléliser les étapes où l'on scanne la protéine avec notre membrane pour ainsi ajouter plus de possibilité. En effet, nous sommes, pour l'instant, restreint à 10 points sur l'hémisphère alors que nous aurions pu en rajouter davantage si la parallélisation était implémentée.

Références

- [1] Ostermeier C, Michel H. Crystallization of membrane proteins. *Curr Opin Struct Biol.* 1997 Oct;7(5) :697-701. doi : 10.1016/s0959-440x(97)80080-2. PMID : 9345629.
- [2] Anders Krogh, Björn Larsson, Gunnar von Heijne, Erik L.L Sonnhammer, Predicting transmembrane protein topology with a hidden markov model : application to complete genomes¹¹ Edited by F. Cohen, *Journal of Molecular Biology*, Volume 305, Issue 3, 2001, Pages 567-580, <https://doi.org/10.1006/jmbi.2000.4315>.
- [3] Wallin E, von Heijne G. Genome-wide analysis of integral membrane proteins from eubacterial, archaean, and eukaryotic organisms. *Protein Sci.* 1998 Apr;7(4) :1029-38. doi : 10.1002/pro.5560070420. PMID : 9568909 ; PMCID : PMC2143985.
- [4] Peter Tompa, Yasufumi Emori, Hiroyuki Sorimachi, Koichi Suzuki, Peter Friedrich, Domain III of Calpain Is a Ca²⁺-Regulated Phospholipid-Binding Domain, *Biochemical and Biophysical Research Communications*, Volume 280, Issue 5, 2001, Pages 1333-1339, <https://doi.org/10.1006/bbrc.2001.4279>.
- [5] Tusnády GE, Dosztányi Z, Simon I. Transmembrane proteins in the Protein Data Bank : identification and classification. *Bioinformatics.* 2004 Nov 22;20(17) :2964-72. doi : 10.1093/bioinformatics/bth340. Epub 2004 Jun 4. PMID : 15180935.
- [6] Bank, R. P. D. (s.d.). RCSB PDB - 2K1A : Bicelle-embedded integrin alpha(IIB) transmembrane segment. <https://www.rcsb.org/structure/2k1a> [\[lien\]](#).
- [7] Bank, R. P. D. RCSB PDB - 7M67 : NMR Structure of Schistocin-1 antimicrobial peptide in presence of DPC-d38 micelles. <https://www.rcsb.org/structure/7M67> [\[lien\]](#).
- [8] Bank, R. P. D. RCSB PDB - 1SI4 : Crystal structure of Human hemoglobin A2 (in R2 state) at 2.2 Å resolution. <https://www.rcsb.org/structure/1SI4> [\[lien\]](#).
- [9] Bio.PDB.DSSP module — Biopython 1.76 documentation. <https://biopython.org/docs/1.76/api/Bio.PDB.DSSP.html> [\[lien\]](#).
- [10] Evenly distributing n points on a sphere. (2012, 7 mars). Stack Overflow. <https://stackoverflow.com/questions/9600801/evenly-distributing-n-points-on-a-sphere> [\[lien\]](#).
- [11] OPM de la protéine 2K1A. <https://opm.phar.umich.edu/proteins/301> [\[lien\]](#).
- [12] OPM de la protéine 7M67. <https://opm.phar.umich.edu/proteins/7799> [\[lien\]](#).

Annexes

Difficulté

La plus grande difficulté rencontrée lors de ce projet a été de comprendre comment scanner la protéine avec la membrane.

Il a fallu retourner lire les cours d'algèbre lycée afin de se souvenir comment former un plan perpendiculaire à un droite.

Structure prog

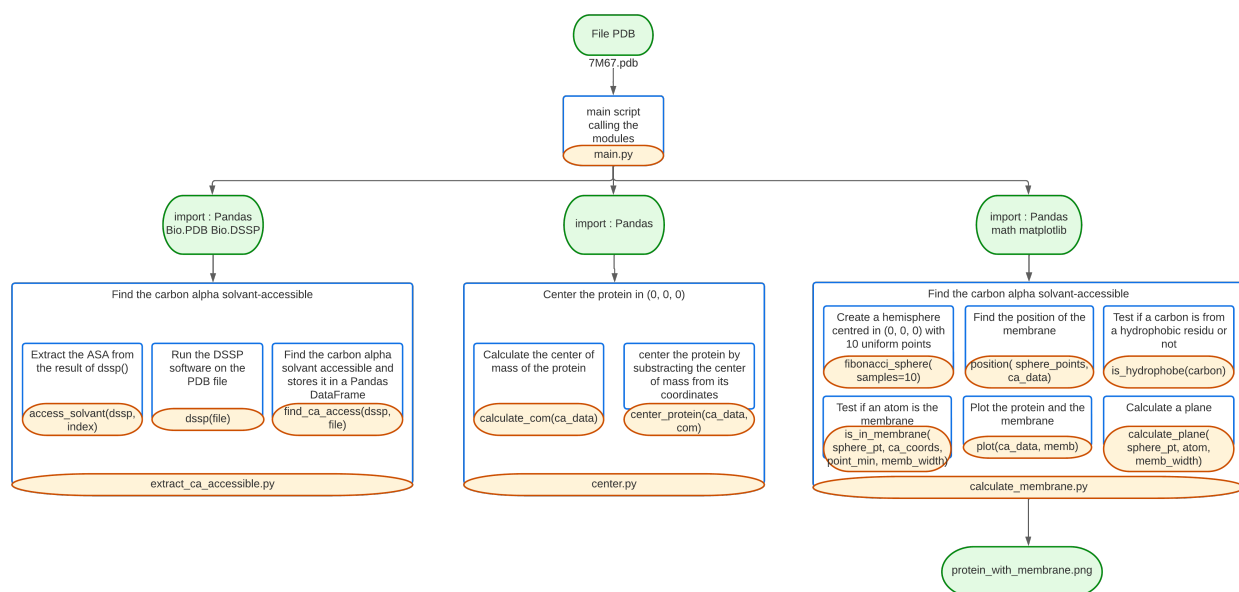


FIGURE 3 – Graphique de la structure de l'algorithme.

Exemple

Pour utiliser ce programme, nous devons lancer la commande suivante :

```
python3 src/main.py data/[File]
```

En faisant ceci nous lançons le script principal qui appellera 3 modules permettant d'obtenir un graphique. Si nous lançons notre algorithme sur le fichier 7m67.pdb, nous obtiendrons le graphique suivant :

Ici nous l'avons l'exemple d'un peptide transmembranaire. Ce peptide antimicrobien Schistocin-1 de l'espèce *Schistosoma mansoni*, trouvable sur le site avec le code 7M67 [7] mesure 21 acide aminés de long et a été résolu par RMN. Il présente une amidation sur son 21ème acide aminé. Il a la particularité d'être sécrétée par *Schistosoma mansoni* et est ainsi capable de traverser les membranes sans pour autant y être localisé. Cela rend ce peptide intéressant car, bien qu'il soit transmembranaire, il ne se trouvera pas dans la membrane.

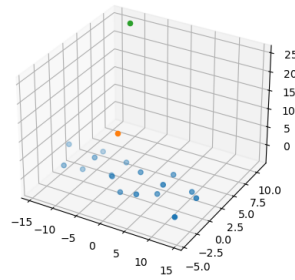


FIGURE 4 – Position de la protéine 7M67 (points bleu) par rapport à la position de la membrane (point orange : feuillet bas, vert : feuillet haut) obtenu par notre algorithme.

Ensuite, nous comparons nos résultats à ceux attendus pour les protéine transmembranaire :

+ + + + + + + + + + + + + + +

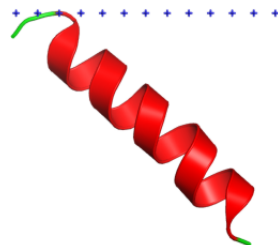


FIGURE 5 – Position de la protéine 7M67 dans la base de donnée OPM [12]

Nous observons que la membrane se situe bien à une extrémité du peptide comme obtenu avec notre algorithme. Ainsi, nous pouvons conclure que celui-ci fonctionne sur les protéine transmembranaires.