# Assignation et détection des parties transmembranaires d'une protéine

**PETIT Ferdinand** Université de Paris - Étudiant Master 2 BI-IPFB **KERMARREC Maxime** Université de Paris - Étudiant Master 2 BI-IPFB

Ce projet de bioinformatique a été réalisé dans le cadre de l'UE Programmation et Gestion de Projets de la deuxième année du Master BI-IPFB de l'université de Paris.

#### Introduction

Les protéines transmembranaires jouent un rôle important dans la vie cellulaire. Elle peuvent avoir un rôle dans: la transduction de signal, la structure de la membrane ou encore en temps que canal ionique. Cependant, une caractéristique lie les protéines transmembranaires, en effet la partie en contact avec la membrane est composée en majotitée d'acides aminés hydrophobes et exposés.

Lorsqu'une protéine est identifiée comme étant transmembranaire, l'emplacement de la bicouche lipidique n'est pas indiqué dans la *protein data bank* (PBD) car ces protéines sont cristallisées sans leur bicouche lipidique naturelle, et actuellement il existe des méthodes permettant de détecter le plan membranaire en utilisant les coordonnées atomiques des protéines membranaires. Dans ce projet, nous nous sommes proposés de remplir l'objectif d'établir un outil permettant de déterminer les zones transmembranaires d'une protéine.



#### Matériels et méthodes

Nous avons listées et résumées toutes nos classes et fonctions en annexe, ici nous expliquerons la logique derrière notre programme.

Dans un premier temps, le programme lit un fichier PDB donné un argument de commande afin d'en extraire les coordonnées des carbones alpha mais aussi le fichier amino\_acid.csv répertoriant les caractéristiques des 20 acides aminés protéinogènes. Par la suite, il va calculer le centre de masse de la protéine. Toutefois, suite au calcul de la surface accessible au solvant de chaque acide aminé avec l'algorithme DSSP, les carbones alpha qui ne sont n'y hydrophobe n'y ayant un score au DSSP supérieur à 0.3 (par défaut) sont exclus. Grâce à un algorithme de fibonacci nous pouvons répartir des points de manière uniforme sur une sphère ayant pour centre le centre de masse. Chaque point sera à l'origine, avec le centre de masse, d'un vecteur normal de deux plans cartésiens. Le programme générera donc deux plans cartésiens parallèle et non confondue permettant de délimiter un espace de 15 angstroms et calculera le nombre de carbone alpha à l'intérieur. Les deux plans se déplaceront d'un angstromse du début jusqu'à la fin de la protéine. L'espace ayant eu le plus grand score sera considérés comme étant celui qui définit au mieu l'interaction entre la protéine et la membrane.

Détails techniques du programme :

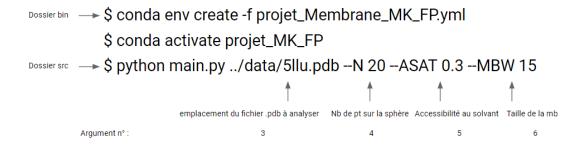


Figure 1: Ligne de commande à entrer

Input:

### Obligatoire:

Fichier pdb d'une molécule a une seule chaine, stockée dans le répertoire data

#### Optionnel:

–N nombre (par défault 20) indiquant le nombre de points générés et répartis de facon homogène sur une sphère. Plus le nombre est élevé plus la précision augmente. −ASAT nombre (compris entre 0 et 1, par défault 0.3) indiquant le critère de sélection des acides aminés exposés au solvant. A noté qu'il semble y avoir un bug a partir de 0.6 (leproramme ne smble pas s'arreter. −MBW nombre (en angstöm, par défault 15) indiquant la taille de la membrane.

#### Output:

Un fichier text nomproteine.txt contenant : -un rappel des différents arguments utilisés -les coordonnées du centre de masse de la proteine (x,y,z), -le score ainsi que les facteur (a,b,c,d1,d2) des coordonées cartésiennes du plan tilt pour résoudre l'équation cartésiennes ax+by+cz+d=0 -La séquence des acides aminés présant dans la membrane Un fichier pdb nomproteine\_ca.pdb contenant tous les carbonnes alpha de la proteine. Toutes les fichiers de sortis se trouvent dans le répertoire results

#### Résultats

Nous avons soumis la protéine de transport EAAT1 (code PDB : 5llu) à notre programme.

PDB file: ../data/5llu.pdb

Gravity center: [7.447830808080808, 47.44442929292934, 0.12914646464646493]

Best tilt: score=145 a=-0.778899864655898, b=0.5384615384615401, c=0.32151854135750446,

d1=-25.4777751902465, d2=-10.47777519024649

Sequences: 0(35),1(39),2(42-43),3(45-46),4(49-50),5(52-53),6(56-57),7(60),8(64),9(68),10(71-72),11(76),12(80-81),13(83-84),14(87-88),15(91-92),16(95-96),17(98-100),18(102-103),19(175),20(178-179),21(182),22(185-186),23(188-189),24(192-193),25(195-202),26(207-208),27(210-211),28(214-215),29(245-246),30(248-267),31(272-273),32(275-285),33(288-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),34(315),35(318-319),36(322),37(327),38(328-304),38 358),41(360-362),42(364-383),43(386-387),44(390)

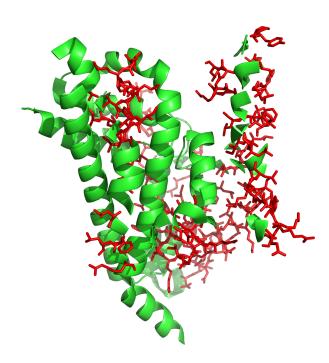


Figure 2: La protéine EAAT1. Les résidus dont le carbone alpha est compris dans l'espace définis par les plans cartésien sont en rouge, le reste est en vert.

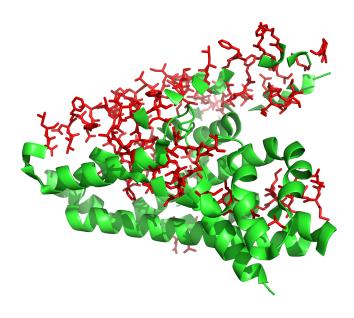


Figure 3: La protéine EAAT1 sous une vue différente. Les résidus dont le carbone alpha est compris dans l'espace définis par les plans cartésien sont en rouge, le reste est en vert.

Nous pouvons remarquer qu'une face de la protéine EAAT1 est entièrement recouverte d'acids aminés en contact avec la membrane, ainsi la partis en verte est soit cytosolique, soit extra-cellulaire. En effet, la protéine EAAT1 étant une protéine de fusion. Nous pouvons donc en déduire raisonnablement que la partie verte est à l'extérieur ou à l'intérieur de la cellule en atttendant de créer un complexe avec une autre protéine avant de s'immerger entiérement dans la membrane lipidique afin d'atteindre le compartiment opposé.

#### Discussion

Nous avons conçus un programme permettant d'estimer l'emplacement d'une membrane biologique vis-à-vis d'une protéine transmembranaire. Celui-ci fonctionne rapidement (2s pour 400 acides aminés) et semble cohérent. Toutefois, de petite extensions donnerait énormément de modularité dans un contexte scientifique. En effet, nous pourions faire en sorte que le programme puisse effectuer les mêmes calculs sur des valeurs d'hydrophobicté spécifique à chaque acide aminé, ou encore que l'on puisse spécifier sur quelle caractéristique des acides aminés nous voulons travailler (aliphatic, petit, etc...). Finalement une représentation visuelle de la membrane en interaction la protéine générée est la prochaine amélioration que nous voulons intégrer à ce projet.

#### Annexe

Ce projet fut court, intense et gratifiant. Nous avons été stimulé par ce dernier et aurions aimé aller plus loin . Toutefois, nous soumettons ce projet en ayant le sentiment de l'avoir mené à bien.

## Bibliographie

Modules utilisé:

Librairie

argparse, numpy as np, pandas as pd, csv, from Bio.PDB import PDBParser, from Bio.PDB.DSSP import DSSP, sys

Nos propre modules

import amino\_acid as aa

Classes et fonction

Fichier main.py

class sphere(object): def init(self, samples=20): def str(self): def translation(self, center): class proteine(object): def init(self, file, aacDF): def trouve\_calpha(self, file): def center(self): def calc\_ASA(self, file): def hydrophobicity(self, aacDF): def str(self):

Fichier calpha.py

def trouve\_calpha(pdbfile, out):

Fichier amino\_acid.py

def amino\_acid\_caracteristics():

# Bibliographie

Simon. 2014. *Transmembrane Proteins in the Protein Data Bank: Identification and Classification*. Bioinformatics, Volume 20, Issue 17. doi:10.1093/bioinformatics/bth340.