

Assignation et détection des parties transmembranaires d'une protéine

Laura XÉNARD

[Lien GitHub](#)

1. Introduction

Au niveau génomique, les protéines membranaires intégrales représentent 20 à 30% de l'ensemble des protéines. Elles permettent le transport de diverses substances au travers de la membrane et jouent un rôle thérapeutique majeur en étant la cible de nombreux médicaments. Mais malgré leur importance, ces protéines sont sous-représentées au sein de la *Protein Data Bank*¹ (PDB). Cette sous-représentation s'explique d'une part par la difficulté à les cristalliser lorsqu'elles se trouvent en milieu aqueux, et d'autre part par leur grande taille rendant complexe la résolution de leur structure par résonance magnétique nucléaire (RMN).

Afin de pallier à cette sous-représentation, l'algorithme TMDet a été développé (Tusnády *et al.* 2004–2005[5]). Celui-ci permet non seulement de différencier les protéines transmembranaires des protéines globulaires mais aussi d'en déterminer les parties transmembranaires à partir de la seule information de structure. Grâce à cet algorithme, une base de données spécifique aux protéines transmembranaires a été mise en place : PBDTM² (Protein Data Bank of Transmembrane Proteins).

Depuis, d'autres méthodes plus rapides ou plus efficaces ont été proposées, notamment ANVIL (Assignment aNd Visualization of the Lipid bilayer) une approche par classification binaire ne s'appuyant que sur les coordonnées des carbones alpha par Postic *et al.* 2016 [2].

On propose ici une implémentation simplifiée de TMDet reprenant certains éléments de ANVIL, ainsi qu'une comparaison des résultats obtenus sur quelques protéines transmembranaires choisies. Le code est disponible sur [GitHub](#).

1. Voir <https://www.ebi.ac.uk/pdbe/>.
2. Voir <http://pdbtm.enzim.hu/>.

2. Matériel et méthodes

Le programme `membrane_plane` a été développé en Python 3.7 sous Linux et Windows et selon le paradigme de la programmation orientée objet. Il est composé de 3 modules interagissant tel que représenté ci-dessous en figure 1.

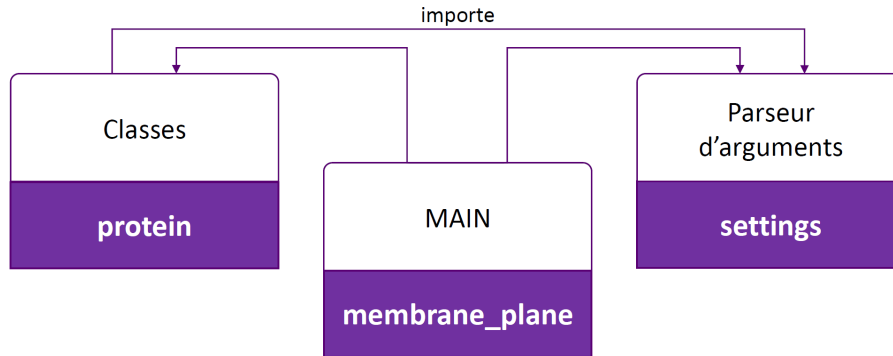


Figure 1. Architecture du programme `membrane_plane.py`.

Algorithme général

Ce programme identifie la position de la membrane en quadrillant la protéine et en attribuant un score à chaque tranche de celle-ci. La tranche présentant le meilleur score est définie comme étant la position de la membrane. Les étapes principales de l'algorithme sont les suivantes :

- lecture du fichier PDB d'entrée et parsing de la structure de la protéine ;
- sélection des résidus exposés à la membrane ou au solvant (les résidus enfouis sont définitivement ignorés) ;
- approximation de la position des résidus par la position de leur carbone alpha ;
- centrage du repère sur le centre de masse de la protéine et échantillonnage de l'espace selon N directions [1] ;
- pour chacune de ces directions, détermination de la tranche de 14Å centrée en (0, 0, 0) puis translation de cette tranche par pas de 1Å le long des directions ;
- pour chacune des tranches identifiées, calcul de son score ;
- épaississement progressif de la tranche présentant le meilleur score de manière à le maximiser ;
- sauvegarde de la protéine ainsi que des parois de sa membrane (i.e. plans parallèles définissant la meilleure tranche) dans un fichier PDB.

Calcul du score

Deux méthodes ont été implémentées pour le calcul du score des tranches. Dans les deux cas, seul les résidus exposés au solvant sont pris en compte. Pour la méthode dite

'simple', le score est calculé par l'expression :

$$\frac{1 * nb_res_hydrophobes_tranche - 0.5 * nb_res_hydrophiles_tranche}{nb_res_hydrophobes_proteine}$$

Pour la méthode dite 'ASA', ce n'est pas le nombre de résidus qui est pris en compte mais leur surface accessible au solvant :

$$\frac{1 * sum(ASA_res_hydrophobes_tranche) - 0.5 * sum(ASA_res_hydrophiles_tranche)}{sum(ASA_res_hydrophobes_proteine)}$$

Pour les deux méthodes, un score élevé est donc associé à une plus grande probabilité que la tranche en question fasse partie de la zone transmembranaire.

Visualisation des résultats

Lors du débogage, les résultats intermédiaires ont été visualisés grâce au package de visualisation 3D Mayavi [3] (voir en Annexes figure 3 et figure 4 à gauche).

Les fichiers PDB de sortie (et incluant donc la membrane) ont été visualisés avec le logiciel PyMol [4].

3. Résultats et discussion

Les résultats obtenus pour la petite protéine transmembranaire 2n90³ sont incorrects mais encourageants. En effet, même si l'axe membranaire n'est pas correctement identifié (voir figure 2 en page suivante), les résidus internes à la membrane l'ont plutôt bien été. Le programme délimite la membrane comme allant du résidu 8 (phénylalanine) au résidu 35 (glycine) tandis que la base PDBTM⁴ la délimite du résidu 12 (valine) à 34 (alanine).

Cependant, le fait que la membrane soit détectée par le programme comme étant perpendiculaire à la position donnée par PDBTM interpelle. Ce comportement ne se retrouve pas systématiquement sur toutes les protéines mais il est tout de même possible qu'une erreur au sein du programme soit la cause de cette mauvaise détection.

Pour les quelques autres protéines testées, notamment 2lor⁵, les résultats obtenus sont incorrects. Il est nécessaire de revoir l'algorithme et potentiellement la fonction de scoring si l'on veut obtenir des résultats qui se rapprochent de la réalité.

En terme de performance, les temps d'exécution sont courts, de l'ordre de la dizaine de seconde (inférieur à 3s pour 2n90 avec un nombre de directions par défaut et pas d'affichage 3D). Le temps d'exécution augmente linéairement ou quasi-linéairement en fonction du nombre de directions choisi pour l'échantillonnage de l'espace.

3. 2n90 sur PDB : <https://www.ebi.ac.uk/pdbe/entry/pdb/2n90>

4. 2n90 sur PDBTM : <http://pdbtm.enzim.hu/?=/pdbtm/2n90>

5. 2lor sur PDBTM : <http://pdbtm.enzim.hu/?=/pdbtm/2lor>

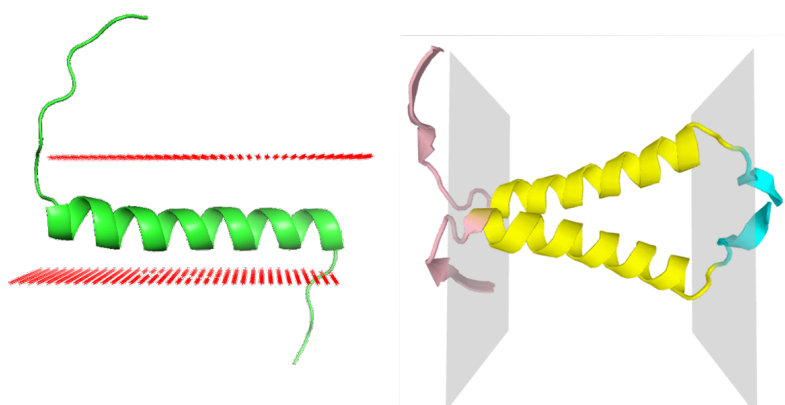


FIGURE 2. Protéine 2n90, comparaison des résultats. A gauche, la visualisation via PyMol de la sortie PDB du programme montrant la chaîne A ainsi que les parois de la membrane, en rouge. A droite, les deux chaînes A et B telles qu’affichées sur PDBTM ainsi que les parois de la membrane, en gris.

4. Conclusion

Pour ce projet, l’accent a été mis sur la programmation orientée objet et la qualité du code. Il en ressort que le programme fonctionne vite et sans erreur à condition que le fichier PDB d’entrée ne contienne pas d’hétéro-atomes ni de positions alternatives. La visualisation des étapes intermédiaires par Mayavi apporte une certaine confiance dans la correcte exécution des différentes étapes.

Faute de temps, les résultats obtenus n’ont cependant pas pu être approfondis. Les résultats préliminaires montrent que le programme ne permet pas de déterminer à coup sûr les résidus intra-membranaires ni même l’axe membranaire. Une amélioration de l’algorithme général et/ou de la fonction de scoring sont probablement nécessaires.

Bibliographie

- [1] Markus Deserno. [How to generate equidistributed points on the surface of a sphere.](#) page 1.
- [2] Guillaume Postic, Yassine Ghouzam, Vincent Guiraud, and Jean-Christophe Gelly. [Membrane positioning for high- and low-resolution protein structures through a binary classification approach.](#) *Protein Engineering, Design and Selection*, 29(3) :87–92, Mar 2016.
- [3] P. Ramachandran and G. Varoquaux. [Mayavi: 3D Visualization of Scientific Data.](#) *Computing in Science Engineering*, 13(2) :40–51, 2011.
- [4] Schrödinger, LLC. [The PyMOL Molecular Graphics System, Version 1.8.](#) November 2015.
- [5] Gábor E. Tusnády, Zsuzsanna Dosztányi, and István Simon. [Transmembrane proteins in the Protein Data Bank: identification and classification.](#) *Bioinformatics*, 20(17) :2964–2972, Nov 2004.

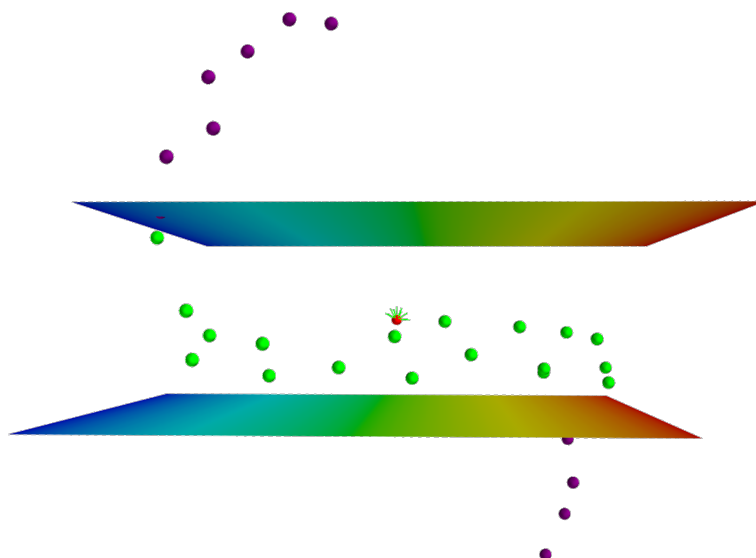


FIGURE 3. Protéine 2n90, model 1, chaîne A par Mayavi. En vert, les résidus internes à la tranche. En violet, les résidus externes à la tranche. En rouge et vert au centre, le centre de masse de la chaîne et les vecteurs quadrillant l'espace ($N = 20$). Seuls les résidus exposés au solvant sont affichés.

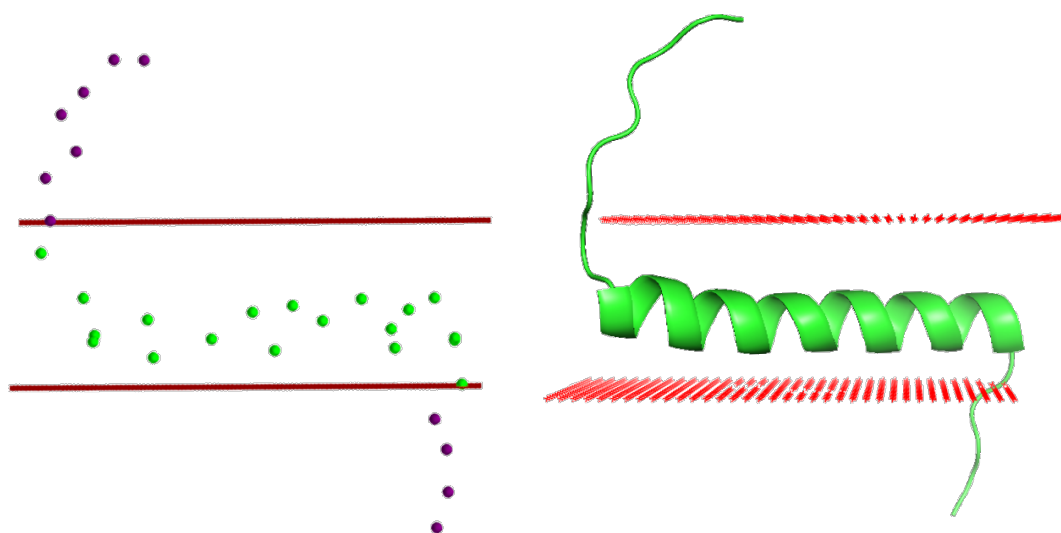


FIGURE 4. Protéine 2n90, model 1, chaîne A par Mayavi (gauche) et PyMol (droite). En rouge, les parois de la membrane. Pour Mayavi, les résidus internes (resp. externes) à la tranche sont représentés en vert (resp. violet) et seuls les résidus exposés au solvant sont affichés.