SOUFIR Emma

Dépôt GitHub : https://github.com/esoufir/TM_detect

ASSIGNMENT AND DETECTION OF TRANSMEMBRANE SEGMENTS OF A PROTEIN

UE Programmation et Gestion de Projet

M2 BI – Université Paris Cité 2023/2024

Introduction

Transmembrane proteins are a crucial class of molecules that have essential roles in various cellular processes. Understanding the structures and functions of these proteins has central importance in biology and has many implications for drug discovery or disease understanding. However, the identification and classification of transmembrane proteins is a difficult process because the membrane cannot be resolved with the structure.

Here, we created a tool to identify the membrane limits of transmembrane proteins inspired by the method used in [1,2].

Material and Methods

Algorithm

The main goal of this algorithm is to find the best position of a membrane, represented by two planes, in a 3D space for a given protein. The best position is defined by the positions of the two planes which maximizes the relative hydrophobicity. The purpose is to maximize the number of hydrophobic exposed-to-solvent residues between the two planes. First, the correct orientation of the membrane is identified, then the membrane's width is adjusted.

To illustrate this problem, we used the OOP paradigm, which is quite suitable for encapsulating information together on a particular object (Fig.1).

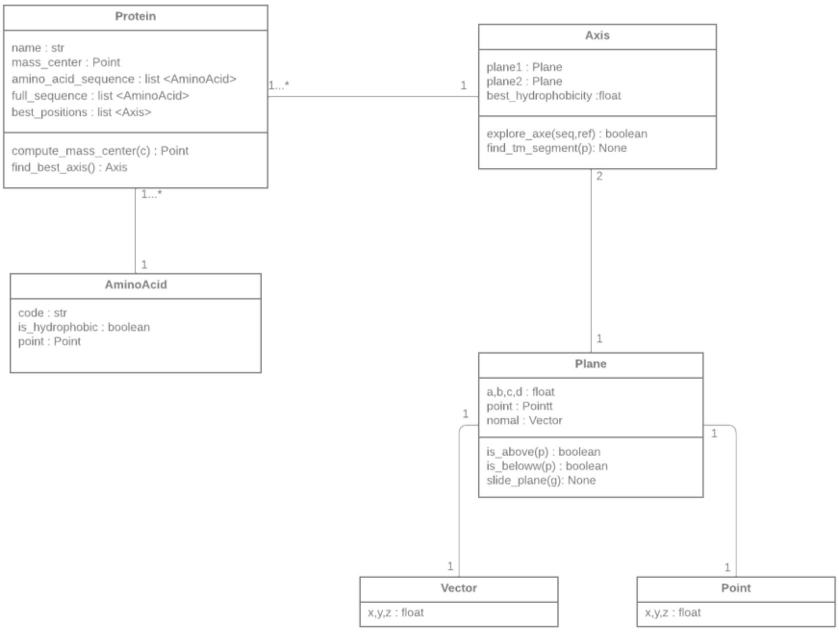

Figure 1: XML Diagram of the structure of TM_detect.

Firstly, the PDB file is checked and parsed to create and initialize the Protein object, using AminoAcid objects to encapsulate information on each residue. The solvent accessibility of each residue is then computed using the DSSP algorithm [3]. A residue is considered accessible if its DSSP solvent accessibility is above 30%. The coordinates of the Cα are also registered in each AminoAcid object. Finally, the hydrophobicity of the residue is registered. The residues considered as hydrophobic were the same as in the article [1,2].

Then, the center of mass of the chain A of the protein is calculated. From this center, N points equally distributed on a demi sphere using the Staff and Kuijlaars method [4] are generated. Therefore, there are N vectors generated between the center of mass and each point. For each of those vectors, two orthogonal planes are generated, separated by a certain gap (14 Angstroms by default). This information is stocked into an Axis object.

For each of those axes, the two planes slide firstly above and then below. After each slide (1 Angstroms by default), the relative hydrophobicity of the residues located between the two planes is calculated, using the formula :

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

If the relative hydrophobicity is higher than the best match yet, this axis is more advantageous. It is stocked in an object that has a role of reference. This step is repeated until there are no more atoms in between the two planes, updating the value of the more advantageous axis if necessary. This sliding is also performed in the other direction of the axis. In the end, for one axis, the best position for the planes is stocked with the associated relative hydrophobicity. This step is repeated for each point on the half-sphere. Then the axis with the best hydrophobicity ever is conserved.

Finally, the width of the membrane is adjusted by exploring the best axis step by step above and below for each plane.

Inputs

Since a protein is simplified here as a unique chain, we used only small and simple transmembrane proteins to evaluate our tool. We used some of the proteins mentioned in the article [1] as inputs for our program.

The user of the program can test different parameters. The number of initial points, the width of the initial membrane, and the size of the sliding window along the axis can be chosen by the user. The choice of the input parameters can affect the precision of the prediction of the tool.

Outputs

The program opens a PyMol [5] GUI with the protein and the predicted planes as heteroatom objects. The PyMol session containing the protein and all the objects representing the planes are saved into a “.pse” file. Another output file with the ‘.xyz’ extension which contains the coordinates of the membrane is also generated. This file can be opened with PyMol.

Results and Discussion

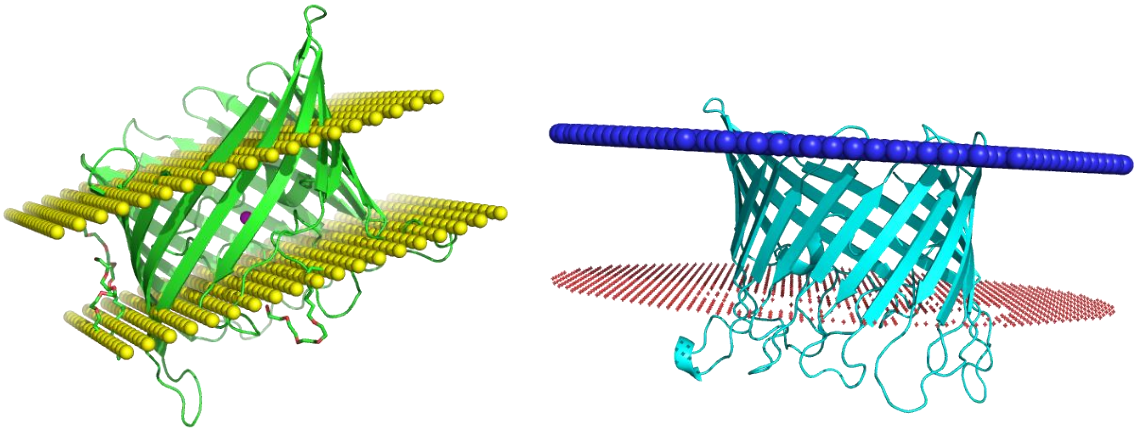

Figure 2: Comparison between the output of our program (left) and the PDB file given by the OPM databank (right).

To compare our results, we used the PDB files provided by the OPM data bank [6]. Here, we compared the results for the porin from Rhodopseudomonas blastica (PDB: 1prn) in Fig.2. The orientation of the planes seems coherent between the two programs. We can see that the width of the membrane is smaller in our results than the one proposed by OPM. The polar residues below seem well isolated but the prediction is less precise for the upper plane. However, the position of it seems coherent according to the databank.

To extend this tool, several leads could be investigated. Firstly, we could consider adding the possibility of detecting the membrane on a poly-chain protein to extend the possibilities of the tool. We should also consider adapting the way of calculating the relative hydrophobicity because the tool has issues when it comes to dealing with small transmembrane areas with large non-transmembrane areas. We could also take into account the different degrees of hydrophobicity of each residue. Finally, as mentioned above, we could improve the optimization of the membrane's width by being more tolerant in finding the best axis with the best planes. To solve this problem, we could also change the algorithm by starting with a quite large membrane and reducing it until the hydrophobicity is significantly lower.

References

[1] Tusnády GE, Dosztányi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. *Bioinformatics*. 2004;20(17):2964-2972. doi:10.1093/bioinformatics/bth340

[2] Tusnády GE, Dosztányi Z, Simon I. TMDET: web server for detecting transmembrane regions of proteins by using their 3D coordinates. Bioinformatics. 2005;21(7):1276-1277. doi:10.1093/bioinformatics/bti121

[3] DSSP algorithm: https://swift.cmbi.umcn.nl/gv/dssp/

[4] Saff B, Kuijlaar A.B.J. , Distributing Many Points on a Sphere, 9 Springer Verlag New York, Volume 19. Number 1, 1997

[5] PyMol Software: https://pymol.org/2/

[6] OPM databank : https://opm.phar.umich.edu/