LIFE SCIENCE UFR

MASTER OF BIO-INFORMATIC

Report short project
Assignation and detection of the transmembrane parts of a protein

Maya Zygadlo

Supervisor : Tatiana Galochkina

Co-supervisor : Jean-Christophe Gelly

Maya ZYGADLO

September, 14 2022

Table des matières

1 Introduction

1.1 Missing information during the crystallisation of a transmembrane protein

Transmembrane proteins are underrepresented in structural databases such as the Protein Data Bank (PDB). This is due to the fact that it is difficult to crystallise them as soluble proteins. Indeed, during the crystallisation process, these proteins are not in their natural environment, the membrane, and can therefore adopt different conformations than the one they naturally adopt. To compensate this, they are put in detergent membrane [1]. However, this solution is not perfect because, although detergents are amphiphilic molecules forming micelles and membranes, they can also form different architectures.

Moreover, the position of the membrane is often not indicated and thus leaves an important part of the information on these proteins aside [2].

1.2 Differentiate between membrane and globular proteins

Yet these proteins are essential for the good functioning of an organism. Indeed, they represent between 20 to 30 % of the protein sequences [3] and have a key role in cellular communication and in the transportation of molecules. Therefore, they are preferential targets for the search of new pharmaceutical drugs. It is therefore important to have information on the position of the membrane.

However, with only sequence information, it is difficult to distinguish transmembrane proteins from soluble proteins (globular protein), allowing to place membranes on the right proteins [4]. Indeed, by having only the polarity of the amino acids of a protein, we cannot distinguish a soluble protein from a transmembrane one because soluble proteins have apolar amino acids and transmembrane proteins have polar parts allowing ligand binding or enzymatic reaction.

1.3 Objectif

Therefore, it is interesting to create an algorithm that is able to find the most probable position of the membrane from sequence information found in a PDB file.
This algorithm has already been designed by a team of researchers (Tusnády et al. [5]) who then created a database listing all the transmembrane proteins with their membrane location information. Here we will try to produce our own algorithm to compare the results obtained with those in the OPM database.

2 Material and Method

2.1 Test Molecules

To do this, we performed our tests on 2 proteins, a transmembrane and a globular one.

We thus worked with the transmembrane segment of the human α integrin IIB, found on the PDB site under the identifier 2K1A [6]. This 42 amino acid long protein is present in eukaryotic plasma membranes and has been resolved by NMR. No mutations are present in this protein.

We also worked with the human globular protein Hemoglobin A2, which can be found on the PDB under the identifier 1SI4 [8]. This protein has 2 pairs of chains of 141 and 146 amino acids each which were resolved by X-ray diffraction with a resolution of 2.20 Åand does not have any mutations.

2.2 Tools

The created algorithm is available on GitHub at the follow address :

https ://github.com/CNOV0/Projet_Management

It was written using the Python3 programming language and the Conda environment manager. A README is available on the GitHub page to test it out.
The Conda envrionment contains the following modules :

BioPython, Pandas, NumPy, matplotlib

It also requires the DSSP software [9] which will be installed by salilab from the Conda environment and used by the BioPython module.

2.3 Algorithm conception

This algorithm takes in input the PDB file of a protein and returns a PNG graphic of the protein with the location of the two membrane leaflets.

To do this, the algorithm will read the PDB file and pass it to the DSSP software from the BioPython module in order to obtain a solvent accessibility score for each residu.
Once it is done, we select the α carbons of the protein that are accessible to the solvent (having a score higher than 0.5). These carbons will be saved in a Pandas dataframe with their scores, the name of the residues to which they belong and their coordinates.

As a second part, we center our protein in (0, 0, 0) by calculating its center of mass and subtracting it from the α carbon coordinates.

As a third part, we generate a hemisphere of radius 1Åwith equidistant points on its surface. This hemisphere follows the Fibonacci algorithm to place the points uniformly. This part of the algorithm was inspired by an existing one found on the following website [10]. We chose to create a 10 point hemisphere because we could not parallelize the next step.

As a fourth part, starting from each point of the hemisphere, we draw a straight line to the center and place two planes perpendicular to the line and parallel to each other of 22 Å. The choice of using a membrane width of 22 Åcomes directly from the article by Tusnády et al. [5] who advise a 22 Åmembrane for a protein composed only of α carbon. To calculate these planes we use the following calculation :
d = x_sphere * x_point + y_sphere * y_point + z_sphere * z_point
Given that in order to separate each plane of 22 Å, we increase the point coordinates with the calculation :
x_point = x_point + x_sphere * 22
These two parallel planes then move along the line from the minimum point (xmin, ymin, zmin) to the maximum one (xmax, ymax, zmax) in steps of 1 Å. For each slice, the hydrophobicity score is calculated, i.e. the ratio between the number of hydrophobic amino acids and the number of amino acids. This is done by directly comparing the results *d_carbon* with *d_plane1* and *d_plane2*.
The best score is thus stored in a dictionary with the coordinates of these two planes.

Finally, a graph is drawn and saved in the *results* directory to visualize our protein and the membrane.

3 Results

Following the launching of the algorithm, we obtain graphs allowing us to visualize our protein and where each of the membrane leaflet is located.

To be sure of our results, we compare them to those obtained in OPM, a database of transmembrane proteins containing the position of the membrane in relation to them. Thus, we obtain the following results :

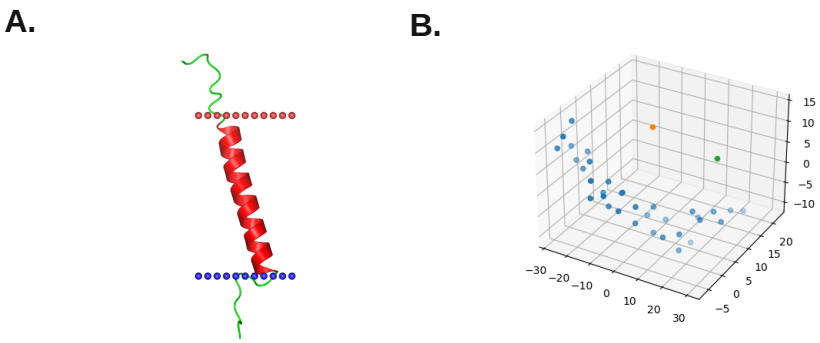

FIGURE 1 – **A**. Position of the 2K1A protein in relation to the membrane obtained in OPM [11]. **B**. Position of the 2K1A protein (blue dots) relative to the position of the membrane (orange dot : lower leaflet, green : upper leaflet) obtained by our algorithm.

The membrane leaflets are perpendicular to the line passing though each of the two points. Thus, Figure 1 B demonstrates a membrane passing laterally through the center of the protein.

We then obtain a result close to the one found in the OPM database. This confirms our results for a transmembrane protein.

We now consider the soluble protein. The expected result is that the membrane does not pass through this protein.

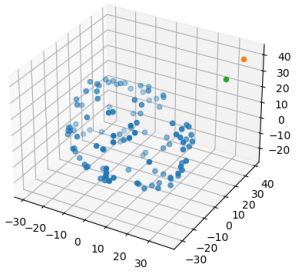

FIGURE 2 – Position of the 1SI4 protein (blue dots) relative to the position of the membrane (orange dot : lower leaflet, green : upper leaflet) obtained by our algorithm.

The figure 2 shows a membrane that does not pass through the protein. This is consistent with the expected results because a non-transmembrane protein isn't in contact with the membrane.

4 Discussion

We could see that our algorithm produced satisfactory results for transmembrane and soluble proteins. Thus we were able to build an algorithm able to predict the membrane position with a certain accuracy.

However, it would have been interesting to be able to vary the size of the membrane in order to be more precise.

Moreover, it would have been nice to be able to parallelise the steps where we scan the protein with our membrane to add more possibilities. Indeed, we are, for the moment, restricted to 10 points on the hemisphere whereas we could have added more if the parallelisation was implemented.

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 genomes11Edited 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 A 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

Difficulties

The biggest challenge in this project was to understand how to scan the protein with the membrane.

We had to go back and read the high school algebra courses to remember how to form a plane perpendicular to a line.

Program structure

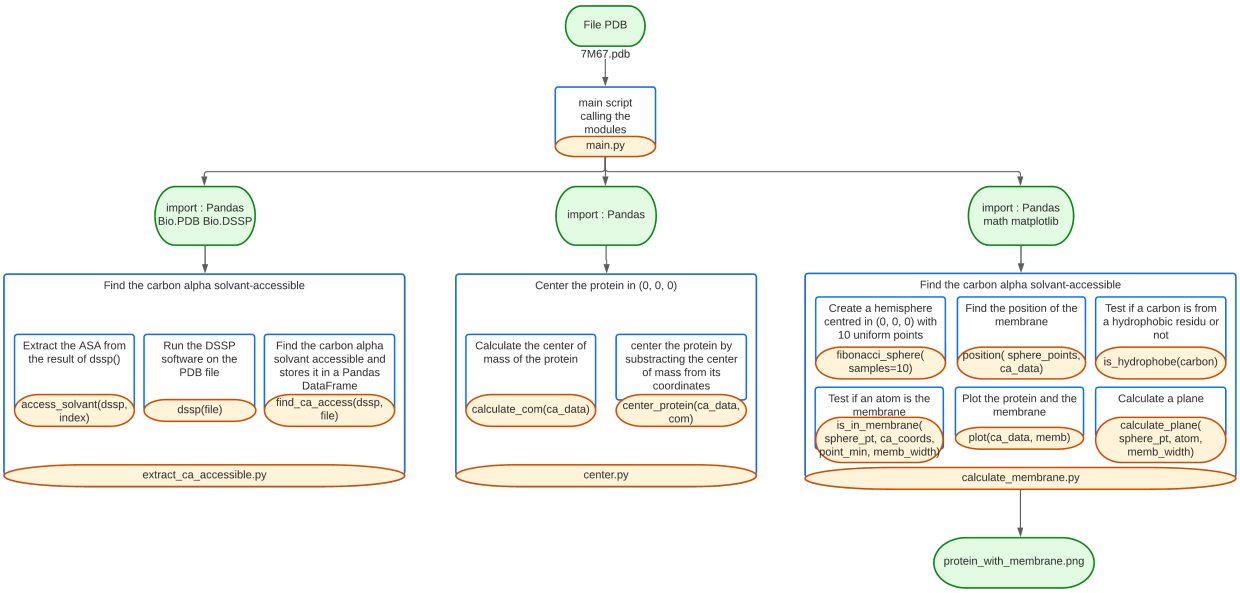

FIGURE 3 – Graphe of the algorithm structure.

Example

To use this program, we need to run the following command :

python3 src/main.py data/[File]

By doing this we launch the main script which will call 3 modules to obtain a graph.

In this example we will work with the 7m67.pdb file. Here we have a transmembrane peptide. This antimicrobial peptide Schistocin-1 from the species Schistosoma mansoni, found on the PDB website with the code 7M67 [7] is 21 amino acids long and was resolved by NMR. It presents an amidation on its 21st amino acid. It has the particularity of being secreted by Schistisoma mansoni and is thus able to cross membranes without being localised there. This makes this peptide interesting because, although it is a transmembrane peptide, it will not be found in the membrane.

TO run the algorithm, the command will be :

python3 src/main.py data/7m67.pdb

If we run our algorithm on this file, we will get the following graph :

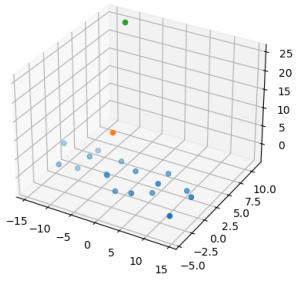

FIGURE 4 – Position of the 7M67 protein (blue dots) relative to the position of the membrane (orange dot : lower leaflet, green : upper leaflet) obtained by our algorithm.

Then, we compare our results with those expected for the trasmembrane proteins :

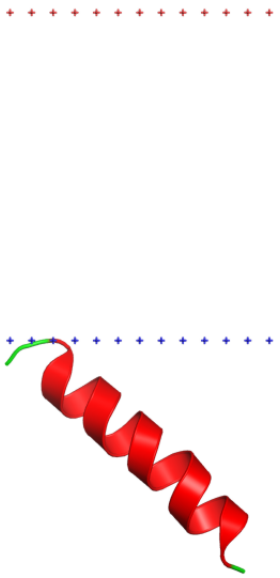

FIGURE 5 – Position of the 7M67 protein from the OPM database [12].

We can see that the membrane is indeed located at one end of the peptide as obtained with Tusnady's et al. algorithm. Thus, we can conclude that it works on transmembrane proteins.