SHORT PROJECT - ASSIGNATION AND DETECTION OF PROTEIN TRANSMEMBRANE REGIONS

Giulia Di Gennaro
M2-BI, Paris-Cité University

September 12th 2024

---

1. INTRODUCTION

Transmembrane proteins play crucial roles in various cellular functions, including molecular transport and signal transduction, yet their structural analysis remains challenging. Despite comprising a significant portion of genomic proteins, they are underrepresented in databases like the Protein Data Bank due to difficulties in crystallizing them without their natural lipid bilayers. This project aims to address these challenges by developing a computational tool that detects and assigns transmembrane regions in proteins based solely on their atomic coordinates. By leveraging structural data such as hydrophobicity and solvent-accessible surface areas, the tool will provide an efficient method to identify membrane boundaries and distinguish transmembrane from globular proteins.

Following the method proposed by Tusnády, Dosztányi, and Simon (2004), the project will implement an algorithm to detect transmembrane regions by analyzing protein structures. The tool will calculate the most probable location of membrane planes relative to atomic coordinates, providing a means to automate the classification of membrane proteins. This will enhance the accuracy of protein structure databases, facilitating future research.

2. METHODS

Protein structure input and preprocessing

The tool takes a PDB file containing the 3D structure of the target protein as input. Using Biopython's PDBParser class, the protein structure is parsed, and the atomic coordinates are extracted for further analysis. The first preprocessing step involves running the DSSP (Dictionary of Secondary Structure of Proteins) algorithm on the PDB file to calculate the solvent accessibility of each residue, ensuring only surface-exposed residues are considered for membrane interaction analysis. DSSP data is filtered to include only accessible residues, which are then stored in a dictionary with associated metadata such as residue name and 3D coordinates.

Spherical sampling and axis generation

To effectively analyze potential membrane positions, the protein is analyzed along multiple axes. The tool generates a uniform distribution of points on a hemisphere using the Fibonacci lattice method (figure 1), ensuring coverage of all possible orientations for membrane interaction. These points on the hemisphere (figure 2) represent axes along which the protein will be sliced. Each axis is defined by a normal vector extending from the center of mass of the protein to the corresponding point on the hemisphere.

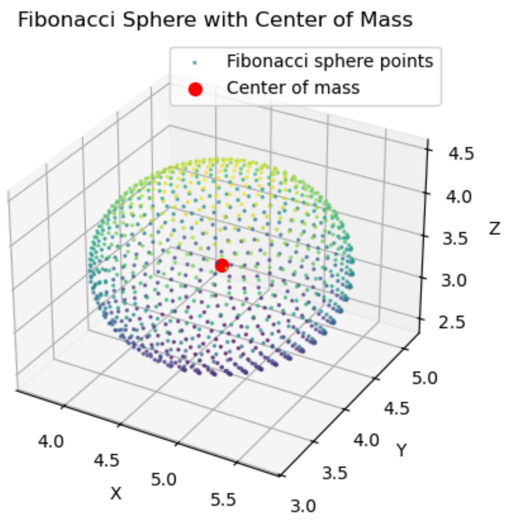


Figure 1 : Fibonacci sphere with centre of mass aligned with the centre of the sphere

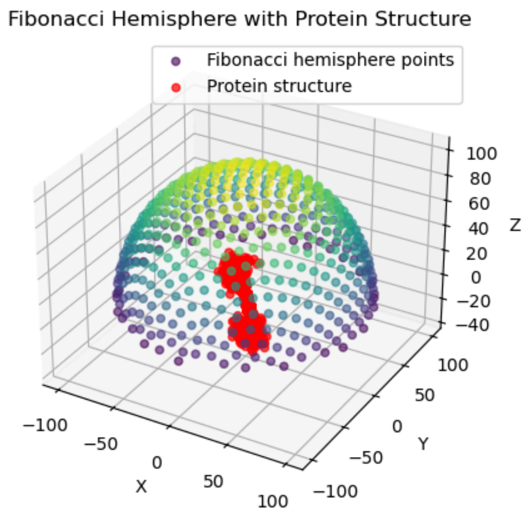


Figure 2 : Fibonacci hemisphere with its centre aligned with the protein's centre of mass

Hydrophobicity profiling and membrane detection

For each axis, the protein is sliced into 1 Å thick sections. These slices are analyzed for their hydrophobic content by evaluating the residue hydrophobicity using a predefined scale. Residues with high hydrophobicity scores that are located near the membrane surface are of particular interest, as these are most likely interacting with the lipid bilayer. The tool calculates a hydrophobicity ratio: the ratio of hydrophobic residues within the membrane-exposed regions to those in solvent-exposed regions. This ratio is maximized across different axes, allowing the tool to identify the most likely transmembrane region. In order to process many axes at the same time, we will make use of the ***multiprocessing*** library from python in order to calculate the hydrophobic ratio for the multiple axes at the same time in parallel of each other.

Membrane plane construction

Once the optimal axis for membrane interaction is determined, two membrane planes are generated perpendicular to the axis, positioned 15 Å above and below the center of mass, resulting in a total membrane thickness of 30 Å. This simulates the lipid bilayer. The tool then recalculates the hydrophobicity ratio across different membrane thicknesses to determine the optimal membrane boundaries, ensuring maximum interaction between the protein's hydrophobic residues and the membrane.

Visualization

The final output includes a modified PDB file with dummy atoms representing the membrane planes, as well as a PyMOL script that visualizes the membrane planes, the protein structure, and the best axis of interaction. This visualization helps researchers easily identify membrane regions and understand the protein's orientation within a membrane environment.

Pseudocode

The tool will follow the pipeline below :

- ➢ Loads and parses the PDB file: Ensures that the PDB file is valid and handles multiple models appropriately.
- ➢ Runs DSSP and filters accessible residues: Confirms that only relevant residues are processed based on solvent accessibility.
- ➢ Builds a comprehensive protein dictionary: Accurately maps residues with their coordinates, accessibility, and names.
- ➢ Generates slicing planes: Utilizes parallel processing to handle multiple slicing directions concurrently, enhancing performance.
- ➢ Identifies the best membrane axis: Employs hydrophobicity analysis to pinpoint the most relevant membrane region.
- ➢ Writes PyMol scrips: Ensures that visualization files adhere to standards, facilitating accurate representation in PyMol.
- ➢ Displays informative results: Provides users with clear and concise information about the best axis, hydrophobicity factor, and runtime.

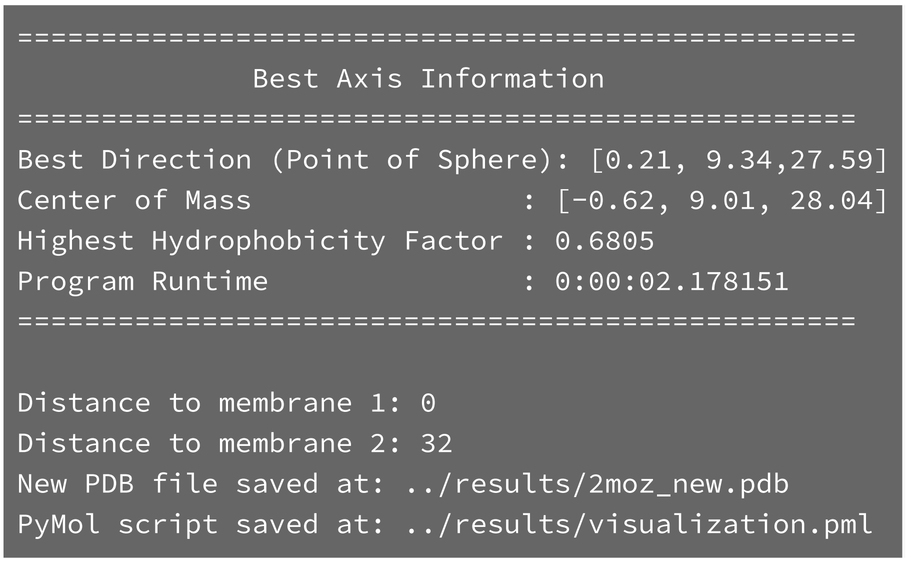
3. RESULTS

We tested our tool on the protein 2MOZ, successfully generating uniform points (figure 2) around the protein, extracting different axes, and calculating the hydrophobicity ratio for each. The program accurately identified the point on the Fibonacci hemisphere that creates the optimal axis for membrane generation.

As observed in figure 3, the program performs efficiently on this small protein, with a runtime of just over two seconds.

However, for the membrane generation step, the two planes representing the membrane are incorrectly positioned, showing an 32 Å distance between them instead of the expected 30 Å.

Additionally, while we successfully created a PyMol (.pml) file for visualization, the membrane planes do not align with the expected results (figures 4 and 5). The membrane axis (axis with the best hydrophobicity ratio), seems to be relatively accurate since the membranes would be perpendicular to this axis, corresponding to the geometry of the 2moz reference (figure 5).



```
==================================================
              Best Axis Information
==================================================
Best Direction (Point of Sphere): [0.21, 9.34,27.59]
Center of Mass                  : [-0.62, 9.01, 28.04]
Highest Hydrophobicity Factor : 0.6805
Program Runtime                 : 0:00:02.178151
==================================================

Distance to membrane 1: 0
Distance to membrane 2: 32
New PDB file saved at: ../results/2moz_new.pdb
PyMol script saved at: ../results/visualization.pml
```

Figure 3 : Program output

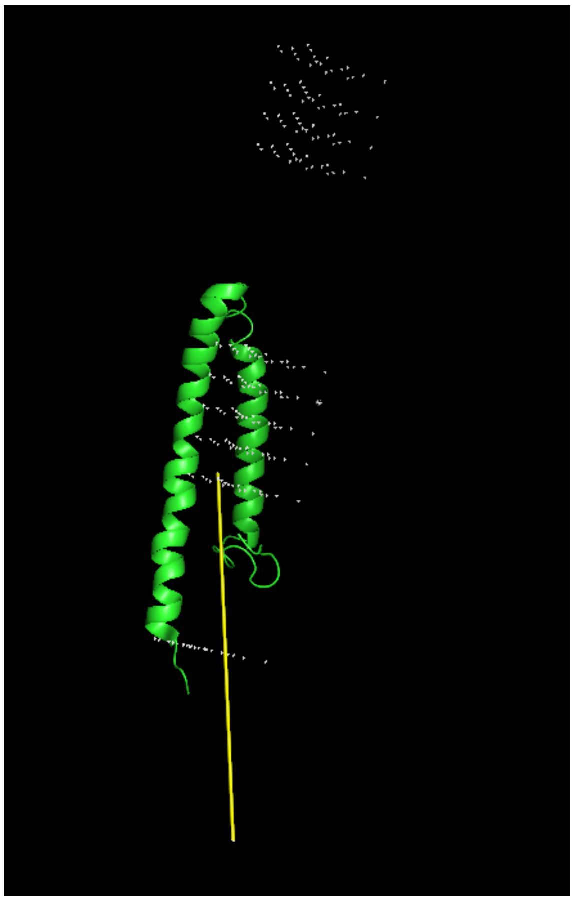


Figure 4 : PyMol representation of 2moz.pdb (green) with the membrane axis (yellow) as well as multiple membranes (white dots)

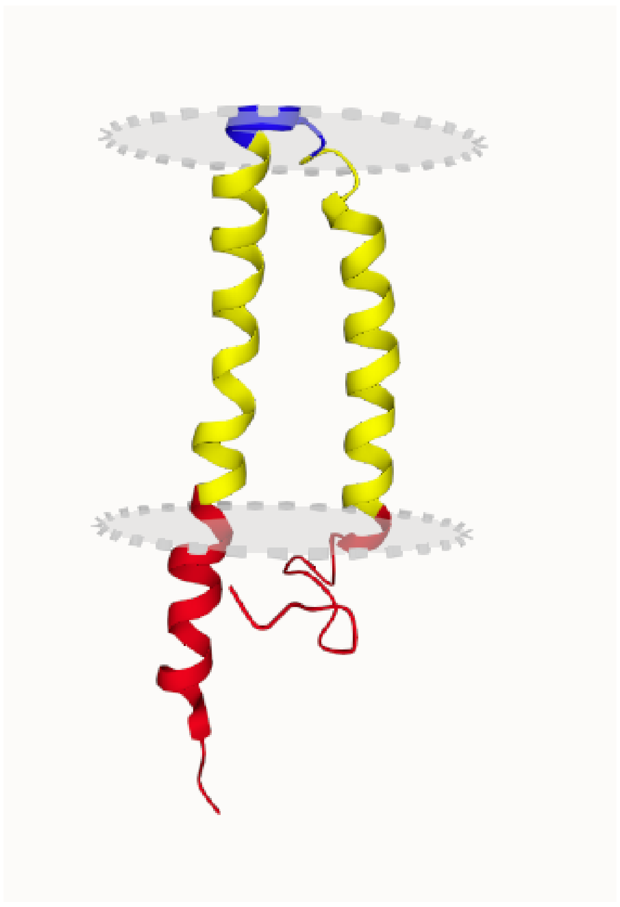


Figure 5 : TM_PDB representation of 2moz.pdb with the membrane planes represented as discs

4. CONCLUSION

In conclusion, the program successfully identifies the membrane axis related to the protein with high efficiency. By incorporating multiprocessing, the program enhances the accuracy of axis identification by enabling simultaneous analysis of multiple axes.

However, the primary goal of accurately identifying, positioning, and orienting the membrane planes relative to the protein was not fully achieved. Future research should focus on developing a more robust method for generating membrane planes. Additionally, implementing a mechanism to adjust membrane thickness would be beneficial in determining the optimal thickness for proper protein embedding.

5. REFERENCES

Tusnády GE, Dosztányi Z, Simon I. Transmembrane proteins in the Protein Data Bank: identification and classification. Bioinformatics. 2004 Nov 22;20(17):2964-72. Epub 2004 Jun 4.

Zucic, D. and Juretic, D. (2004) in Croatica Chemica Acta, 77, pp. 397–401.

https://docs.python.org/3/library/multiprocessing.html

https://observablehq.com/@meetamit/fibonacci-lattices

https://biopython.org/docs/1.75/api/Bio.PDB.DSSP.html