Predicting transmembrane protein topology from 3D structure

02456 Deep Learning

Sitong Chen (s230027) & Xiaopeng Mao (s194408)
Supervised by Felix Teufel

Technical University of Denmark

Goal

There are hundreds of millions of proteins, but we barely know the topology of a few of them. This project aims to improve the performance of identifying the protein parts that are located inside (I), between (P), or outside a cell membrane (O) (the so-called topological prediction). If we can develop more accurate models for prediction, it might greatly benefit the creation of new medicines and even advance the field of bioinformatics. **Figure 1** shows the aforementioned topologies using the proteins α helices and β barrel as an example.

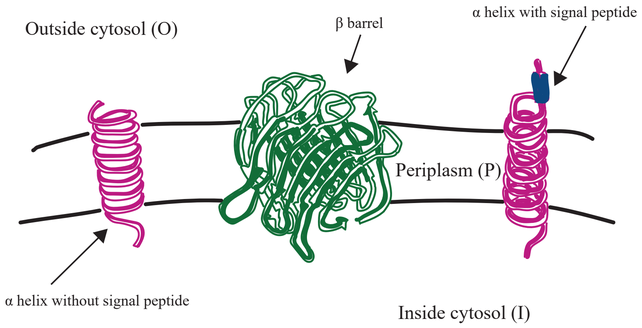

Figure 1: *Simplified illustration of the topological characters of proteins. Inspired by [1].*

Methodology

The methods used in this project are illustrated in **Figure 2**.

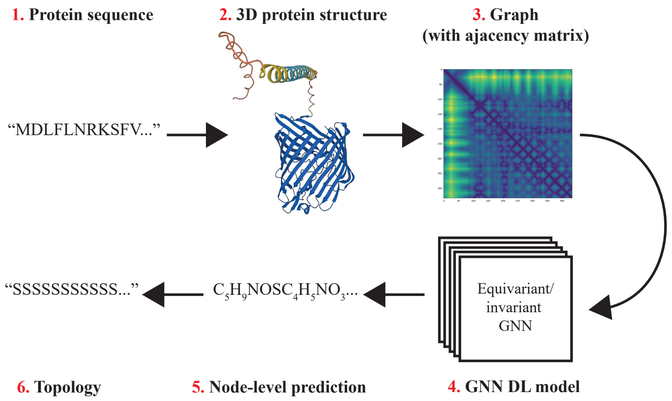

Figure 2: *Methodology throughout the project. All stages are numbered for later reference. The 3D protein structure is from AlphaDB.*

Protein sequence (1)

DeepTMHMM is a state-of-the-art Deep Learning (DL) model that uses the protein sequence information directly [2]. This project also uses the same dataset for topological prediction. Please notice that a small number of proteins are excluded because their corresponding 3D structure cannot be found. **Table 1** shows the protein types and numbers in the dataset.

Table 1: *Number and name of each protein type used in the* DeepTMHMM *study. TM: Transmembrane, SP: Signal Peptide. The numbers without parentheses indicate sequences with both 1D and 3D structures available.*

Protein type	alpha TM	alpha SP+TM	beta barrel	SP+Globular	Globular
Amount	382 (387)	102 (106)	81 (81)	1,982 (2,000)	994 (1,000)

3D protein structure & Graph (2 & 3)

In this project, the molecular 3D structures of the protein sequences are used. The 3D structures are predicted by Alphafold and are available in AlphaFold Protein Structure Database (AlphaDB) [3,4]. These 3D structures are converted into graphs in `.pdb` format, so they can be processed by dedicated Graph Neural Networks (GNNs).

GNN DL model (4)

The main reason why a GNN DL model has to be used for this purpose is simply because the other DL models are not suited for graph learning. For example, a Convolutional Neural Network (CNN) requires the same input dimension for all the inputs. With biological variation, some proteins are larger than others. Hence, the graphs also have different sizes. The key aspects of GNN are an AGGREGATE and an UPDATE function, which constitute the GNN message passing [5]:

$$m_{\mathcal{N}(u)} = \text{AGGREGATE}^{(k)}(\{\mathbf{h}_v^{(k)}, \forall v \in \mathcal{N}(u)\}), \tag{1}$$

$$\text{UPDATE}(\mathbf{h}_u, \mathbf{m}_{\mathcal{N}(u)}) = \sigma\left(\mathbf{W}_{\text{self}}\mathbf{h}_u + \mathbf{W}_{\text{neigh}}\mathbf{m}_{\mathcal{N}(u)}\right). \tag{2}$$

where u is the current node being processed, $\mathcal{N}(u)$ is the neighborhood of u. v is a neighbor node to u an thus belongs to $\mathcal{N}(u)$, $\mathbf{h}_u$ is a so-called hidden node embedding while $\mathbf{W}_{\text{self}}$ and $\mathbf{W}_{\text{neigh}}$ are trainable parameters (see **Figure 3**) [5].

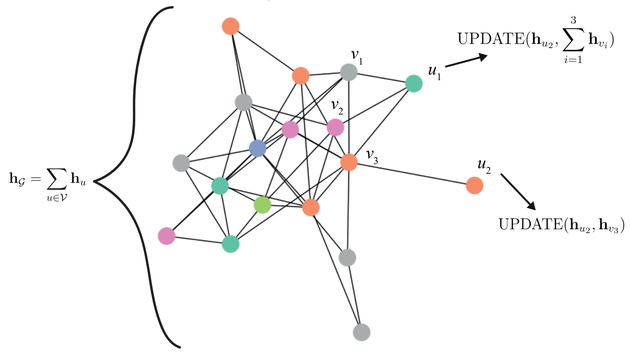

Figure 3: *Example graph with notations explaining the message passing functions UPDATE and AGGREGATION (with summation sign). The graph is generated from random data.*

To put it simply, the AGGREGATE function would aggregate the node embeddings for the nodes that are in the neighborhood of node u while the UPDATE function combines the output from AGGREGATE with the previous node embedding to give an updated node embedding [5]. When the GNN message passing is performed over all nodes and edges, it results in a graph embedding that can reveal more general structure information about the whole graph. GNNs with this message passing are *equidistant* GNNs, meaning the GNN output has the same order as its input. As for *invariant* GNNs, the output order is independent of the input order.

While it depends on the task at hand, GNNs generally use fewer layers compared to e.g. CNNs. This is due to the fact that the neighbor information becomes unclear when a multi-layer abstraction is applied to it.

Node-level prediction & Topology (5 & 6)

This project focuses on the node-level classification. The greatest difficulty in this task is that the length of a protein sequence is different compared to its 3D structure. This is because the protein sequence consists of the residues (bound amino acids) while the 3D structure contains the actual atoms. Hence, a translation is needed to align the node-level predictions with the sequence labels. **Figure 4** illustrates how the model output has been translated.

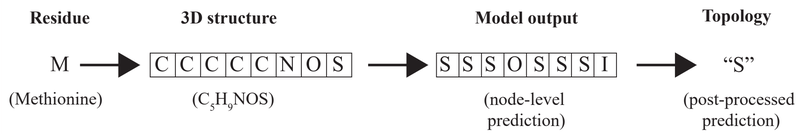

Figure 4: *Simple diagram illustrating the label translation from the atom-level to the residue-level.*

As for the actual GNN training, the popular invariant *SchNet* has been chosen as a starting point. Hold-out method has been used, where *Globular* and *SP+Globular* are used as the training and validation sets while *alpha TM*, *alpha SP+TM* and *beta barrel* all are used as the test set.

Results

This section presents the current available results. **Figure 5** shows the atom-level and residue-level validation obtained from the *SchNet* model training. It can be seen that both the atom-level and the residue-level validations are quite high, indicating the fact that the translation from the atoms to the residues is effective for the GNN to learn the residue-level representation. On the other hand, no correct topology has been predicted (see **Table 2**). This might be because the protein types *alpha TM*, *alpha SP+TM* and *beta barrel* are not even present in the training and validation sets, and they are very different compared to *SP Globular* and *Globular*. Hence, the next step is to use five-fold cross-validation to train and evaluate the model as used by the DeepTMHMM study [2].

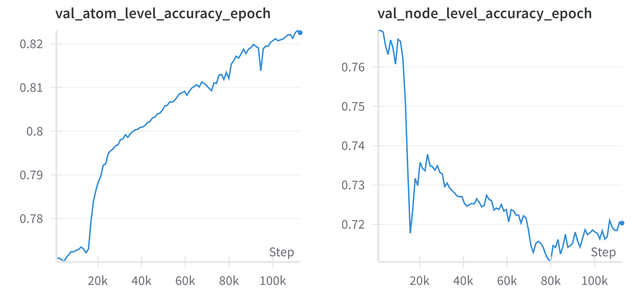

Figure 5: *Left trace: atom-level validation accuracy. Right trace: protein residue-level accuracy. The residue-level accuracy is obtained by comparing the full prediction to the label.*

Table 2: *Test results from* SchNet *trained upon* Globular *and* SP + Globular. *Overall, no correct topology has been predicted even though many residues are precisely located.*

Protein type	alpha TM	alpha SP+TM	beta barrel
Correct topology (%)	0.0	0.0	0.0
Correct residue positions (%)	43.6	32.7	16.7
Correct atom positions (%)	49.2	46.5	18.0

References

[1] Alberts, Bruce, *et al. Essential cell biology*. Garland Science, 2015.
[2] Hallgren, Jeppe, *et al.* "DeepTMHMM predicts alpha and beta transmembrane proteins using deep neural networks." *BioRxiv*, pp. 2022–04, 2022.
[3] Jumper, J *et al.* "Highly accurate protein structure prediction with AlphaFold." *Nature*, 2021.
[4] Varadi, M *et al.* "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models." *Nucleic Acids Research*, 2022.
[5] Hamilton, William L. *Graph representation learning*. Morgan & Claypool Publishers, 2020.