The Hong Kong University of Science and Technology

COMP5212 Machine Learning

Term Project
Group 13

Title: Bio-macromolecules structure and function prediction

Roles and responsibilities

Lau Cheuk Nam:	Tensorflow and Pytorch model training, video recording
Leung Pak Him:	Data analysis, background research
Simona Pepe:	Pytorch model building and training
Wong Yuk Ming:	Tensorflow model building and training, 3D reconstruction UI

Introduction

Protein is one of the basic building blocks for organisms. It can bring numerous functions like catalytic function. It can construct structures like hair and nails. It can be a material transporter. It can be used as a converter to activate stimuli into electrical signals. It can also be used as a tag to label foreign pathogens, and then signalize our immune system to confirm the intrusive cells to attack. Protein is the fundamental functional unit of biological process, and the characteristic functions of many proteins rely significantly on their three-dimensional structures. Therefore, being able to understand the structure of protein is crucial for scientists to predict the protein function. By understanding the proteins in viruses like COVID-19, scientists can learn better its weakness, and hence helpful to develop new drugs.

The structure of a protein can be categorized into 4 layers, Figure.1, which are primary structure, secondary structure, tertiary structure, and quaternary structure. Primary structure is constructed from amino acid sequences. Amino acid is the basic unit of the sequence and there are 20 types of them. The second layer is a secondary structure, where different neighbor amino acids interact with each other and form different spatial patterns, such as alpha helix, and beta-sheet. Tertiary structure defines the 3D-shape of protein by indicating the relative orientation, measured by Psi and Phi angles, of the individual local structure in a single amino acid sequence. Quaternary Structure tells the interaction among different tertiary structures from different amino acid sequences.

However, protein structure is extremely hard to obtain. It requires either NMR [1], X-ray crystallography [2] or cryo-EM [3] to measure. All these processes require purification of protein from the cell, yet purification is time consuming, costly and often with low yield .Due to the difficulty in measuring protein structure, many proteins remain structurally unsolved and a more advanced structure prediction of protein is required to facilitate studies on this paramount topic. Recently, some algorithms have been developed to predict the secondary structure of protein, such as the famous Chou Fasman method [4] . However, it is an empirical method, where sometimes not reliable under some circumstances.

Getting inspired by some modern protein structure prediction methods, such as I-TASSER [5] and AlphaFold [6], which utilize the power deep neural network, we would like to build a simple deep network to investigate this meaningful prediction task. Therefore, we tried to implement deep learning architectures and predict an amino acid sequence up to its tertiary structure. Then we tried to build a user interface for visualizing the corresponding predicted 3D main chain structure of the amino acid sequence. Therefore, users can simply input the requested amino acid sequence into the proposed model, then the user interface will show the predicted 3D structure automatically in the pipeline.

This paper tried to utilize two different architectures — Long short-term memory (LSTM) and Bidirectional LSTM to carry out the prediction. The decision of the two models is based on the fact that the previously built protein will affect the structures of the newly built protein, so the memory in LSTM will be crucial in order to correctly predict the protein structure. We compared the results from the two models and both of the two models have their own advantages. Although the prediction accuracy does not out-perform other research from the past, which also achieved around 60-80% in accuracy[7] [8] [9], our resulting model has around 69% in accuracy and it provides a simple solution that only requires the amino acid sequence to give a preliminary 3D shape of an unknown protein.

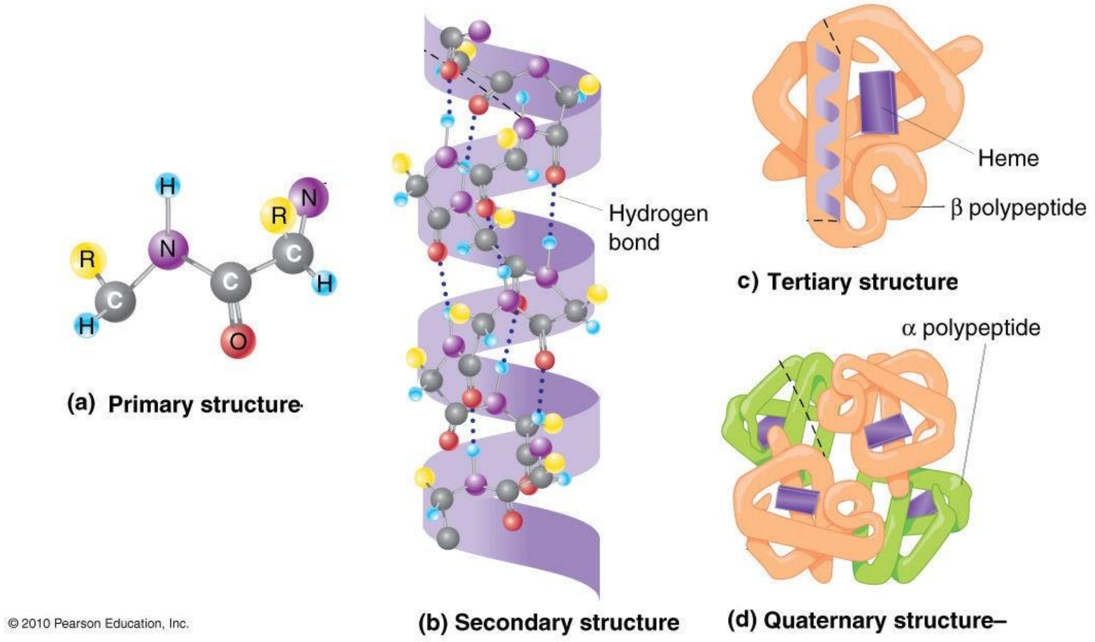


Fig.1 Structure of protein []

Methodology

Data Set

The protein sequence dataset used in this paper was obtained from Protein Data Bank (PDB) [11] in late April 2020. We can get the protein sequence stored in their protein library, with the tags of the secondary structure. We can also download the 3D model of the protein in PDB so that we can compare it with the 3D model generated by the predicted result of their Phi and Psi angles. Besides, the removal of redundant protein structures ensures a clear dataset for training in our machine learning. Redundant protein structures are the protein structure from some proteins which have the same or very similar sequence with very similar structure. In our dataset, these redundant protein structures are removed due to the low value of repeated training. Besides, there are some cases where the sequences are the same or very similar but result in a very different structure. These sequences will be included in the dataset with choosing one of the resulted structures as the predicted output.

Different proteins can have significant differences in sequence length. Some of them can have sequence length of 3660 while the average is just around 100. Loading the protein with long length is a waste of efficiency since preprocessing will likely remove the extended part of it compared to other inputs. Considering the limited computation power we own, we further reduced the dataset size by only keeping those PDB protein sequences that are between the size of 200kb to 250kb, resulting in a dataset with 2265 of protein sequences.

Amino Acid Sequence Vectorization

The dataset obtained from PDB is not suitable to be fed into the model directly, and therefore, we have to preprocess the data using our “PDBVectorizeSX9” function. It will vectorize the PDB dataset to a usable format in the following manner:

- Input Vector, x

To perform the prediction of secondary and tertiary structure from primary structure of protein, the amino acid sequence is the input. However, the characters in the amino acid sequence do not provide much information for training. Therefore, pre-processing of digitalization on the 20 amino acid types is carried out before the input process. We have prepared 9 properties that contribute different types of amino acid the most [12], and each of the 20 amino acid has their own unique values on these 9 properties. The 9 properties are:

Average Mass
Hydrogen Bond

Hydrophilicity
Hydrophobicity
Net Charge Index of Side Chains
Polarity
Polarizability
Solvent Accessible Surface Area
Volume of Side Chains

To feed the vectorized protein sequence to the neural network, we need to ensure the sequential feature of it since the position and neighbor amino acids type will also affect the whole 3D structure. Therefore, to define the primary structure of a protein in a vector x and feed into our model, we turn each protein sequence data into a <N_sequence x 9> vector, where N_sequence is the length of the protein sequence, which is also known as the number of amino acids to build up the complete protein structure.

- Output Vector, y

There are 4 types of secondary structure to describe a polypeptide:

Alpha Helix
Beta Sheets
Turn
Others

And 2 properties for tertiary structure, see Figure.2 for the visualization of the spatial relationship:

Psi angle: dihedral angle formed by the current amino acid N atom, Cα atom, C atom and the next amino acid N atom

Phi angle: dihedral angle formed by the previous amino acid C atom, and the current amino acid N atom, Cα atom and C atom

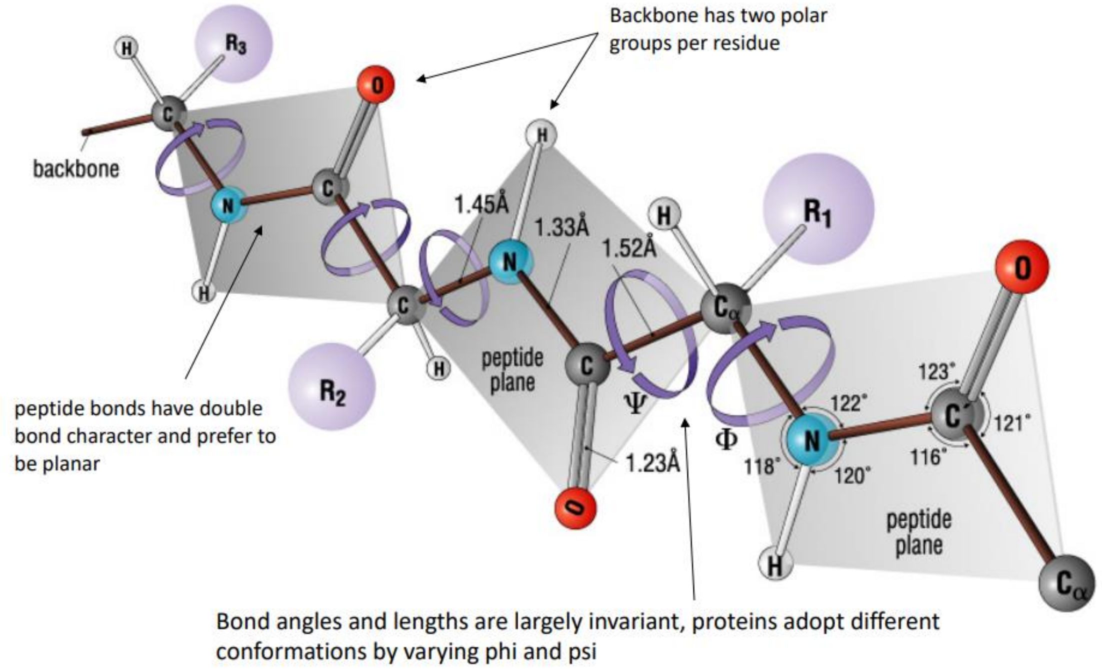
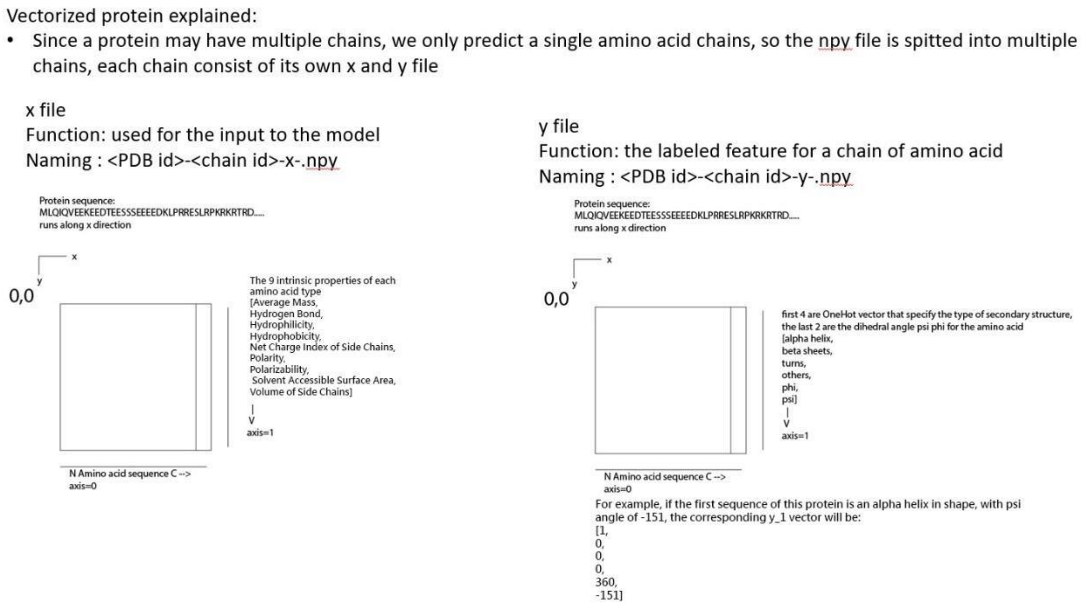

Backbone has two polar groups per residue
R3
H
C
O
backbone
N
C
H
1.45Å
H
1.33Å
H
1.52Å
R1
N
C
Cα
O
R2
H
peptide plane
C
Ψ
Φ
123°
122°
C
121°
N
118°
120°
116°
1.23Å
O
H
peptide plane
Cα
peptide bonds have double bond character and prefer to be planar
Bond angles and lengths are largely invariant, proteins adopt different conformations by varying phi and psi

Fig.2 Visualization of tertiary structure in detail[13]

The goal of the model is to predict the secondary and tertiary structure of polypeptides and therefore, we vectorized the above 6 properties into our output vector y, which is a <N_sequence x 6> vector. N_sequence is the length of the protein. While in each row of output vector y, the first 4 values build up a one-hot vector of the 4 types of secondary structure. Each column represents the type of secondary structure of the current position in the amino acid sequence. The remaining 2 values represent the corresponding Phi and Psi angles. Note that, the Phi angle in the first amino acid sequence and the Psi angle of the last amino acid sequence are indicated as 360° while for all the other angles in between will be in the range of [-180°, 180°]. See Figure.3.



Vectorized protein explained:

- Since a protein may have multiple chains, we only predict a single amino acid chains, so the npy file is spitted into multiple chains, each chain consist of its own x and y file

x file
Function: used for the input to the model
Naming : <PDB id>-<chain id>-x-.npy

Protein sequence:
MLQIQVEEKEEDTEESSSEEEEDKLPRRESLRPKRKRTRD.....
runs along x direction

The 9 intrinsic properties of each amino acid type
[Average Mass,
Hydrogen Bond,
Hydrophilicity,
Hydrophobicity,
Net Charge Index of Side Chains,
Polarity,
Polarizability,
Solvent Accessible Surface Area,
Volume of Side Chains]

y file
Function: the labeled feature for a chain of amino acid
Naming : <PDB id>-<chain id>-y-.npy

Protein sequence:
MLQIQVEEKEEDTEESSSEEEEDKLPRRESLRPKRKRTRD.....
runs along x direction

first 4 are OneHot vector that specify the type of secondary structure, the last 2 are the dihedral angle psi phi for the amino acid
[alpha helix,
beta sheets,
turns,
others,
phi,
psi]

For example, if the first sequence of this protein is an alpha helix in shape, with psi angle of -151, the corresponding y_1 vector will be:
[1,
0,
0,
0,
360,
-151]

Fig**.3.** Visualization of the X, Y vector to represent a protein and its 3D shape properties.

We randomly selected 100 training data from the dataset in each batch, while 80 of them were used for training and 20% were used for validation. Test data are selected from the group of samples that do not take part in the training session.

Baseline LSTM Model

The model we used to retrieve the proteins structures is based on classification method and LSTM architecture (refer to Figure.4). The same model can be substituted with equivalent RNN or GRU networks. Although, for our protein model, LSTM represents better the idea behind protein synthesis, in practical after some comparison of performance in the early stage of our study.

We tested a basic model architecture, composed of one LSTM layer, a linear layer and a softmax layer. To start with, we used the online learning technique. The batch size was set to one where the input vectors x were fed to the model one by one to speed up the calculation, which most likely introduced more instability since the weights were updated each time. Each input vector has the shape of <N_sequence x 9> (for details, please refer to the Amino Acid Sequence Vectorization section). Since N_sequence varies from the amino acid sequence length while the neural network needs to have a fixed-sized input, padding of zeros is used along the first dimension if the vector has a N_sequence < 100. Otherwise, if N_sequence is > 100, the sequence will be trimmed to exact 100 length.

For each amino acid of the sequence, we want the model to compute the probability of the secondary structure type in the current protein sequence position. Hence for each x in input, we retrieved the corresponding y output of shape <N_sequence x 4>. After that, a slight varied model is built for predicting the Psi and Phi angles. The difference of the two models are the removal of the softmax layer and the y output is in shape of <N_sequence x 2>.

To further improve the performance of our LSTM model, we adjusted two model parameters, which are the size of hidden layers and the number of hidden layers. We considered the optimum as the model that determined the lowest validation loss. Where the losses are defined as the Cross-Entropy-Losses between the prediction and the true y values for the structure prediction part and as the MSE-Losses for the angles prediction. The following sets of parameters were validated:

hidden_dim = [50, 100, 200],

num_layers = [1, 2, 4]

Among all the combinations, the optimized model was the one with 4 hidden layers and 50 hidden dimensions per layer. After these, the batch size was set to 100 for training. Since overfitting remains significant in the first few training, early stop was added in our training model and the optimal number of iterations were determined in order to prevent overfitting. No advantages were detected in using the dropout technique. Details of the model architecture are provided in Figure.5. (n_term -> c_term)

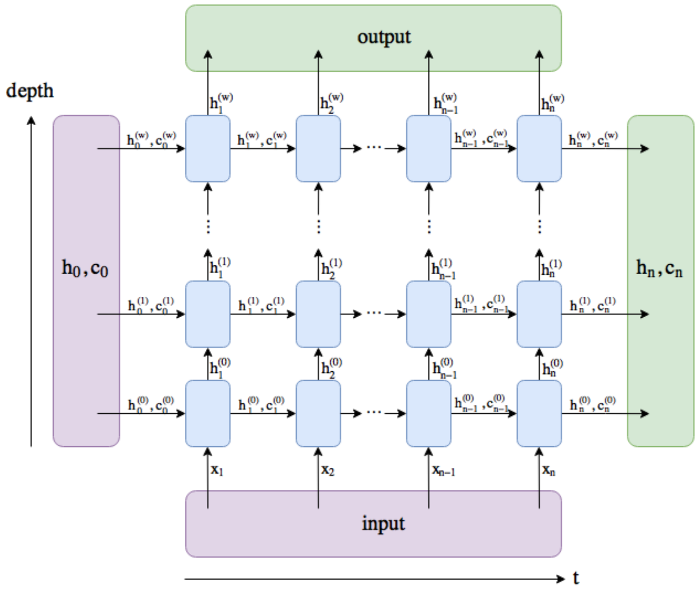
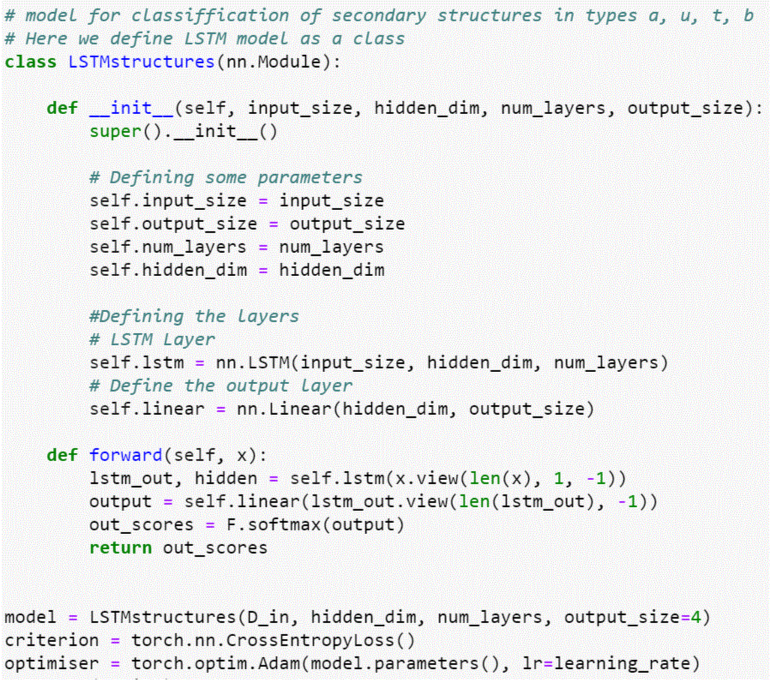


Fig.4. Schematics of a general LSTM architecture



```python
# model for classiffication of secondary structures in types a, u, t, b
# Here we define LSTM model as a class
class LSTMstructures(nn.Module):

    def __init__(self, input_size, hidden_dim, num_layers, output_size):
        super().__init__()

        # Defining some parameters
        self.input_size = input_size
        self.output_size = output_size
        self.num_layers = num_layers
        self.hidden_dim = hidden_dim

        #Defining the layers
        # LSTM Layer
        self.lstm = nn.LSTM(input_size, hidden_dim, num_layers)
        # Define the output layer
        self.linear = nn.Linear(hidden_dim, output_size)

    def forward(self, x):
        lstm_out, hidden = self.lstm(x.view(len(x), 1, -1))
        output = self.linear(lstm_out.view(len(lstm_out), -1))
        out_scores = F.softmax(output)
        return out_scores


model = LSTMstructures(D_in, hidden_dim, num_layers, output_size=4)
criterion = torch.nn.CrossEntropyLoss()
optimiser = torch.optim.Adam(model.parameters(), lr=learning_rate)
```

Fig.5. Class function that defines the model

Bidirectional LSTM model with Sliding Window

After we have further studied some protein prediction papers, we consider to apply some of their assumptions into our model. Hence, the second model we built is a bidirectional LSTM model. The reason for adopting bidirectional model is based on an assumption that the future built polypeptide may affect the protein structure in the previously built structures. By applying bidirectional model, the new model enables the memories both from the previous sequence and the future sequence.

Moreover, the concept of sliding window is introduced to the preprocessing part of the input data. According to a study from [14] , better predictions can be computed through screening the current amino acid in the sequence with a sliding window with size of 35 residues. Although bidirectional LSTM can carry previous and future memories which shares slightly similar function as sliding window, we would like to inform the neural network with a more straightforward way, which is directly feeding the network with a 35-length amino acid sequence as input. Therefore, further preprocessing is necessary for the sliding window screening function. Instead of padding zeros or cutting the redundancy from long sequence, the input samples with S sequence length are cut to (S - 34) 35-length amino acid sequences, then they are fed to the network as input. Apart from using 35-length sliding window, 18-length sliding window was also tested due to ensuring the balance between training speed and accuracy.

3D Visualization Platform

Figure.6 shows the webpage user interface for visualizing the protein structure we built. It allows users to quickly run through our model and visualize the predicted protein. Users can input the amino acid sequence of a protein in “FASTA” format. Figure.7 shows an example of a protein sequence in FASTA format, which is obtained from PDB. And after choosing which models to be implemented to carry out the prediction. Then the UI will show the predicted 3D model of this input protein. Where the button “tf-18” being the bidirectional LSTM with sliding window size of 18 and “tf-35” being the same model with sliding window size of 35. Users can also download the predicted output by clicking the bottom most button. It will output a .npy format.   Figure.8 shows the actual structure window, and the user can directly upload the actual structure obtained from the experiment and compare it with the prediction result side by side. Figure.9 shows a result of the UI.

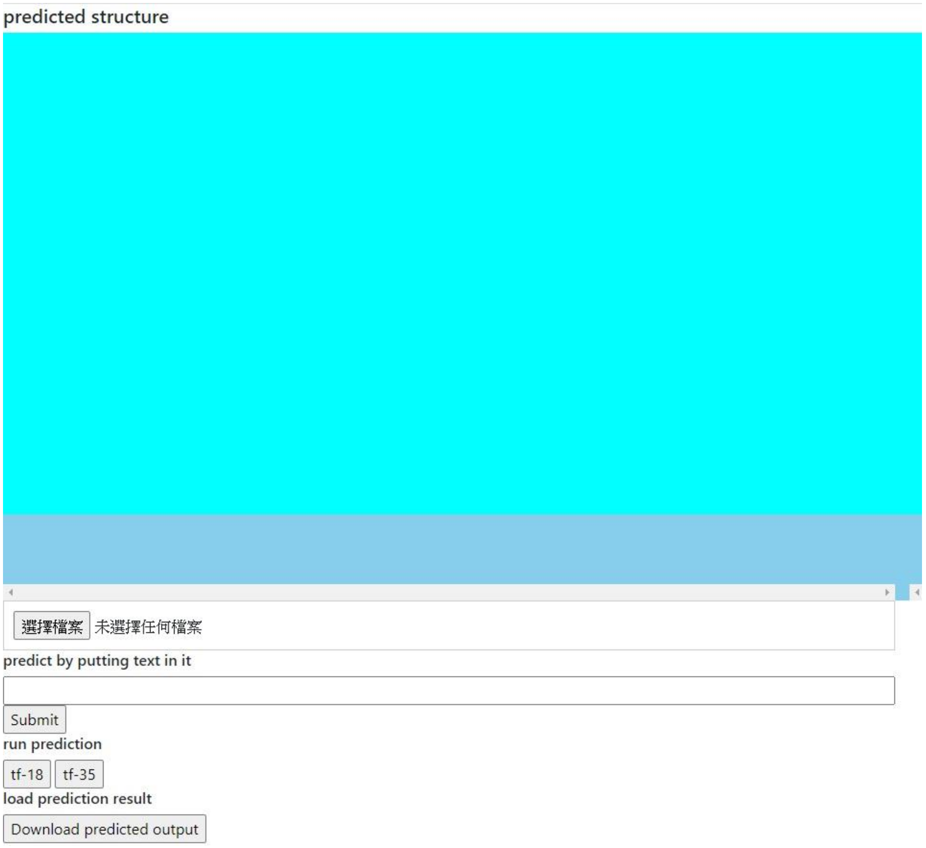
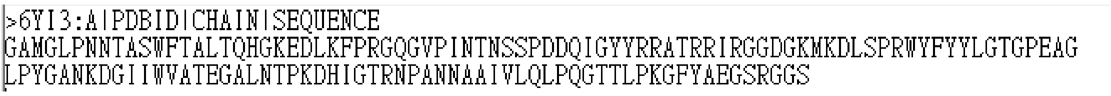


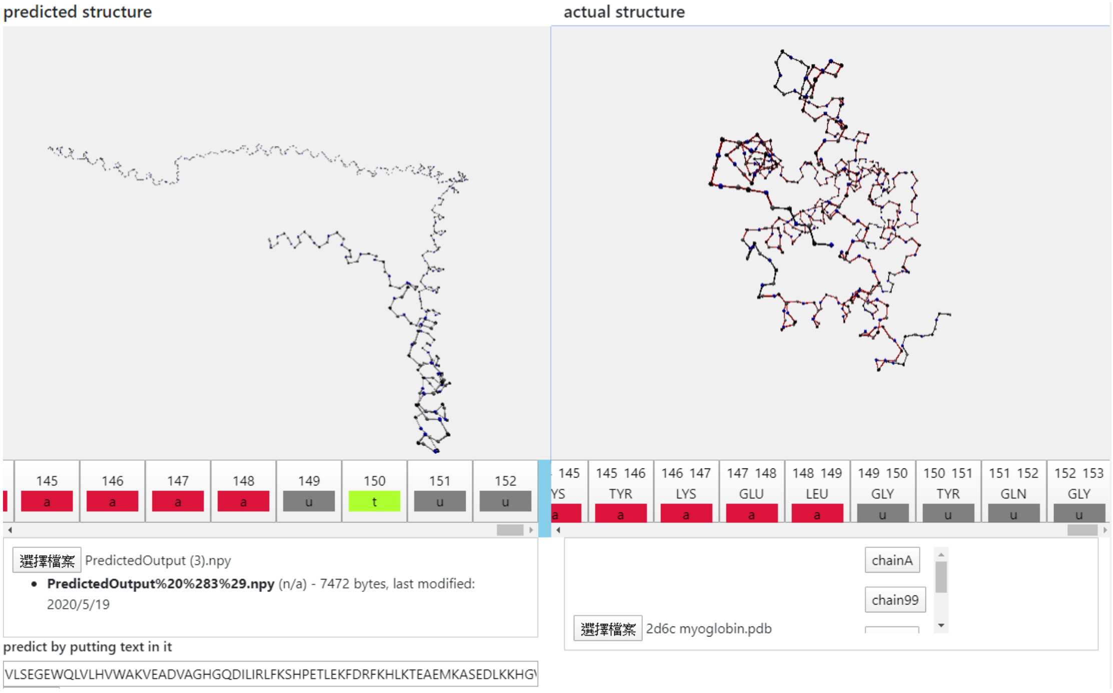
Fig.6. Shows the UI for users to upload an unknown protein sequence.



```
>6YI3:A|PDBID|CHAIN|SEQUENCE
GAMGLPNNTASWFTALTQHGKEDLKFPRGQGVPINTNSSPDDQIGYYRRATRRIRGGDGKMKDLSPRWYFYYLGTGPEAG
LPYGANKDGIIWVATEGALNTPKDHIGTRNPANNAAIVLQLPQGTTLPKGFYAEGSRGGS
```

Fig.7. The protein sequence of the N-terminal RNA-binding domain of the SARS-CoV-2 nucleocapsid phosphoprotein (6YI3) in FASTA format.

Fig.8. Allow users to upload and visualize the actual structure of a unfolded protein



PredictedOutput (3).npy

- **PredictedOutput%20%283%29.npy** (n/a) - 7472 bytes, last modified: 2020/5/19

2d6c myoglobin.pdb

VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKHG\

Fig.9. On the left, it is our prediction result only based on the amino acid sequence input. On the right is the actual experiment result obtained by unfolding the same protein.

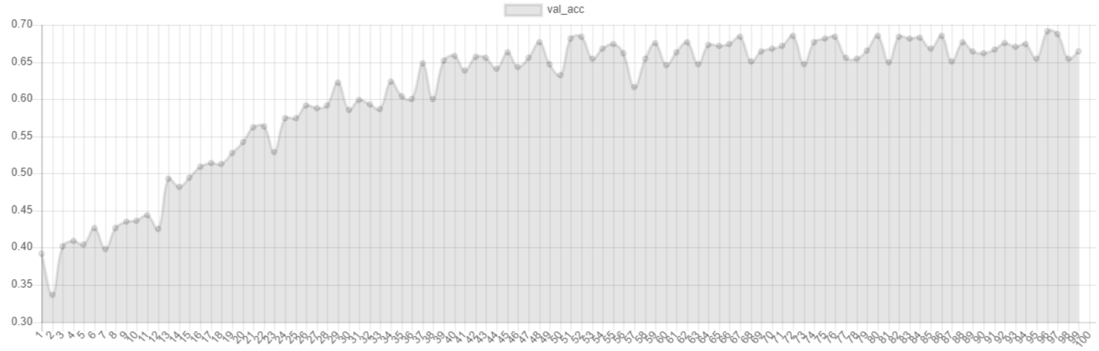
Result

Numerical Result

Both models were trained with the same filtered database. The accuracies of the two retrieved models were tested from the same test dataset. Accuracy for the secondary structure classification was determined by the number of correct classified classes over the total amount of test data; while the accuracy for the Psi and Phi angles prediction was determined by the mean square error between the predicted result and the ground truth angle.

For the baseline LSTM, we determined 0.9466 of accuracy on the training data and 0.6915 on the testing data. The results were then compared with the second model developed, the bidirectional LSTM.

The best bidirectional LSTM model on predicting the secondary structure performed quite similar with the baseline LSTM. Figure.10 shows that the best test accuracy of this model reaches 0.6908. The configurations of the best bidirectional LSTM model are as follows, sliding window size of 35, 3 Hidden Layers with dimensions of 500, 200, 50 respectively, and tanh as their activation function. This result reveals that our assumption on future polypeptides affecting the previously built polypeptide may not be significant while there may exist other factors that are more important to the protein structure, such as environmental factors from the protein localization.



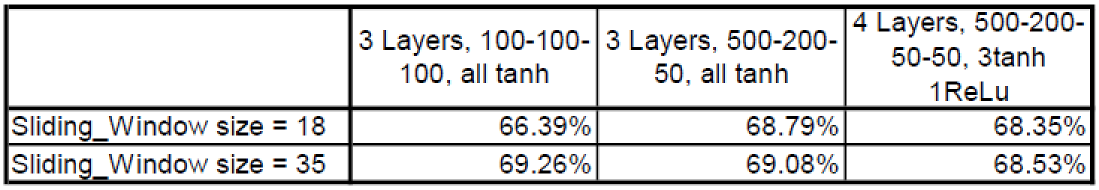
max:0.6908 ,min:0.3361

Fig.10. The test accuracy curve of the bidirectional LSTM model

Sliding Window Size

We also did some experiments to identify the effect on different sliding window sizes when we were building the bidirectional LSTM model.

When predicting the secondary structure, we compared the performance of 6 models. 3 of them were built with sliding windows size of 18 and the remaining 3 were built with sliding window size of 35; while keeping all other configurations the same. Table1. shows the result of these 6 models and we can see that generally, the model with larger sliding window size gives better performance. But the increment is subtle, with around 1% enhancement in average.



	3 Layers, 100-100-100, all tanh	3 Layers, 500-200-50, all tanh	4 Layers, 500-200-50-50, 3tanh 1ReLu
Sliding_Window size = 18	66.39%	68.79%	68.35%
Sliding_Window size = 35	69.26%	69.08%	68.53%

Table1. Validation accuracy of different models in secondary structure prediction

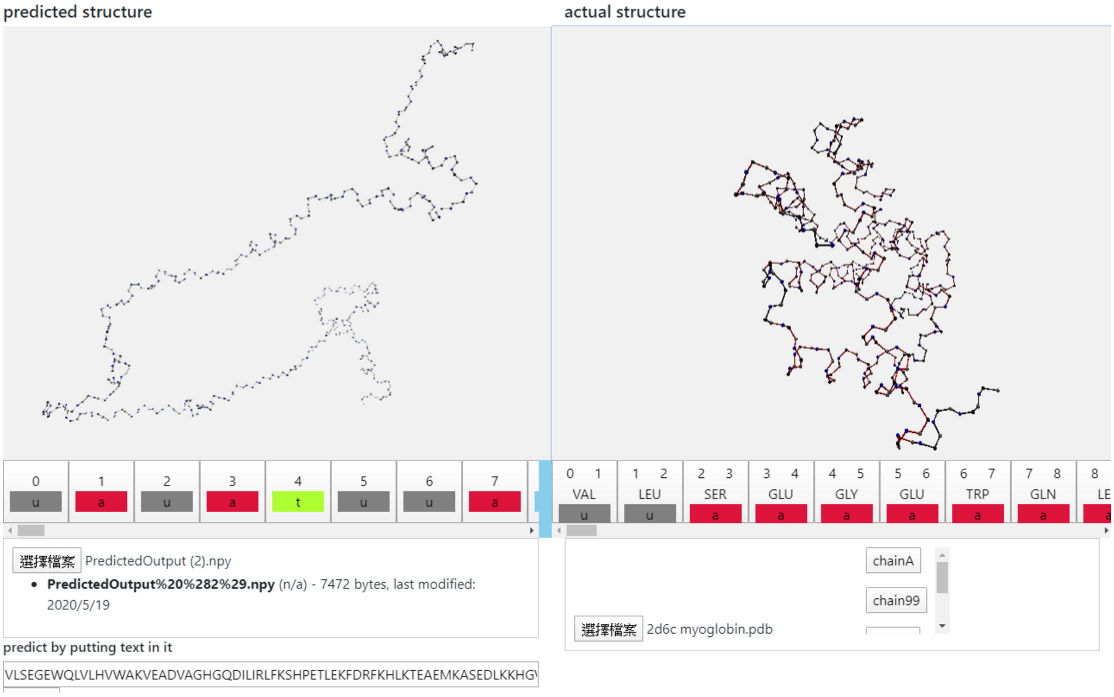
In tertiary structure prediction, the same experiment has also been carried out to see the effect of sliding window size. Table2. shows the performance of the 6 models which are built for psi, phi angle prediction in terms of its validation loss, the larger the value, the worse a model is. We can see the performance does not gain any improvement, instead, get deteriorated when extending the sliding window size to 35. The increase of sliding window size is negatively affecting the tertiary structure prediction. It may be due to the regularization power of sliding window size. When the sliding window includes more amino acid residues, it may tend to regularize the whole chain to a linear chain, weakening the effect of local interaction.

	3 Layers, 100-100-100, all tanh	3 Layers, 500-200-50, all tanh	4 Layers, 500-200-50-50, 3tanh 1ReLu
Sliding_Window size = 18	5112	5125	4676
Sliding_Window size = 35	5606	5780	5645

Table2. The mean square error between the predicted result and the ground truth angle in tertiary structure prediction

Visualization

This is to illustrate the actual output of our model using the webpage user interface. Figure.11 shows the comparison between the predicted myoglobin and the actual protein model using our bidirectional LSTM model with sliding window size of 18. And Figure.12 shows the comparison between the predicted myoglobin and the actual protein model using our bidirectional LSTM model with sliding window size of 35. We can see that the predicted model may look quite different from the actual one, but that may be caused by the environmental factor that is not considered in our model. And the model can predict the secondary structure quite accurately, especially the one shown in Figure.12.



PredictedOutput (2).npy

- **PredictedOutput%20%282%29.npy** (n/a) - 7472 bytes, last modified: 2020/5/19

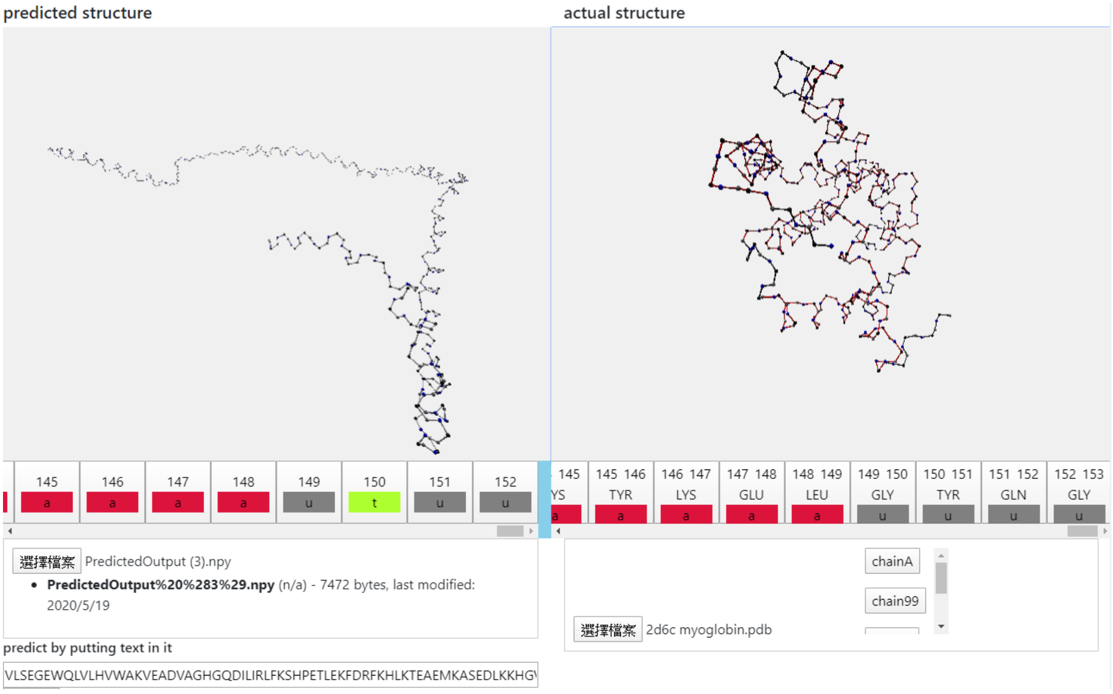
2d6c myoglobin.pdb

predict by putting text in it

VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKHG\

Fig.11. myoglobin tf-18 prediction

Prediction using sliding window of size 18 on human myoglobin (PDBid: 2d6c) tf-18 consist of 2 models namely secondary structure model and angle model that have the highest accuracy and lowest loss respectively and accept vectors with sliding window of 18.



predicted structure

actual structure

145	146	147	148	149	150	151	152
a	a	a	a	u	t	u	u

145	145 146	146 147	147 148	148 149	149 150	150 151	151 152	152 153
YS	TYR	LYS	GLU	LEU	GLY	TYR	GLN	GLY
a	a	a	a	a	u	u	u	u

選擇檔案 PredictedOutput (3).npy

- **PredictedOutput%20%283%29.npy** (n/a) - 7472 bytes, last modified: 2020/5/19

predict by putting text in it

VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFKHLKTEAEMKASEDLKKHG\

選擇檔案 2d6c myoglobin.pdb

Fig.12. myoglobin tf-35 prediction

Prediction using sliding window of size 35 on human myoglobin (PDBid: 2d6c)

tf-35 consist of 2 models namely secondary structure model and angle model that have the highest accuracy and lowest loss respectively and accept vectors with sliding window of 35.

Discussion and Limitation

As shown in the Numerical Result section, adopting the bidirectional LSTM model, which seems to represent the actual protein synthesis better, cannot bring much improvement to the prediction accuracy. This may reveal one limitation that our model can only predict the 3D shape of the protein right after syntheses, but we cannot predict the shape after transporting to the designated part of the body. Since this model does not take the environmental factors, such as pH, ion concentration, solvent accessible surface area, into account

Another possible assumption for insignificant performance from bidirectional LSTM model is due to the complexity of bidirectional LSTM model making it harder to train, given the limited computational power we have. Since bidirectional feature increases the computational cost for training and the sliding window increases the input dimension, the action for parameter adjustment and modification of structure is limited as the model is harder to train. Besides obtaining better graphic cards for faster training, more machine learning techniques and architectures can be tried to increase the training performance while maintaining the sliding window and bi-directional features.

Although we use the secondary structure model and the angle model to predict both secondary structure and tertiary structure of a given amino acid sequence. 3D reconstruction is primarily based on the result of the angle model that gives the dihedral angle psi and phi of each amino acid. From the result, most of the predicted structures are having an extended chain structure despite their original structure being globular. This is because the loss remains stable at a very large value for the angle model, and further training will lead to overfitting, such that the model generalizes the dihedral angles for each amino acid to the form that an extended chain is generated in the 3D reconstruction. However both tf18 and tf35 reconstruction are capable of seeing some structure that resemble the secondary structure of the protein. This suggests that there is a correlation between dihedral angle and secondary structure and is matched with previous studied results.

Conclusion

This paper proposes a simple method that utilizes the power of modern deep neural networks to perform the highly complex, yet important protein shape prediction task. We tested the performance of LSTM and Bidirectional LSTM; it is found that the bi-directional LSTM performs on par with the baseline LSTM, with accuracy of around 69%. We have also tested the network with different sliding windows size, and experiment showed that the sliding window size does bring improvement in predicting the secondary structure, but not when predicting the tertiary structure of the protein. We would recommend increasing the sliding window size during secondary structure prediction, but not in tertiary structure prediction, in order to maintain balance between prediction performance and computational cost.

Our method only requires the input of the primary structure, which is the amino acid sequence, feeding into the neuron network. It is simple, yet able to predict the secondary and the tertiary structure of the protein. Scientists can then get a preliminary 3D shape from an uncertain protein and guess the function of it. And hence, the efficiency of developing drugs which counter the known functions of the protein can be further increased.

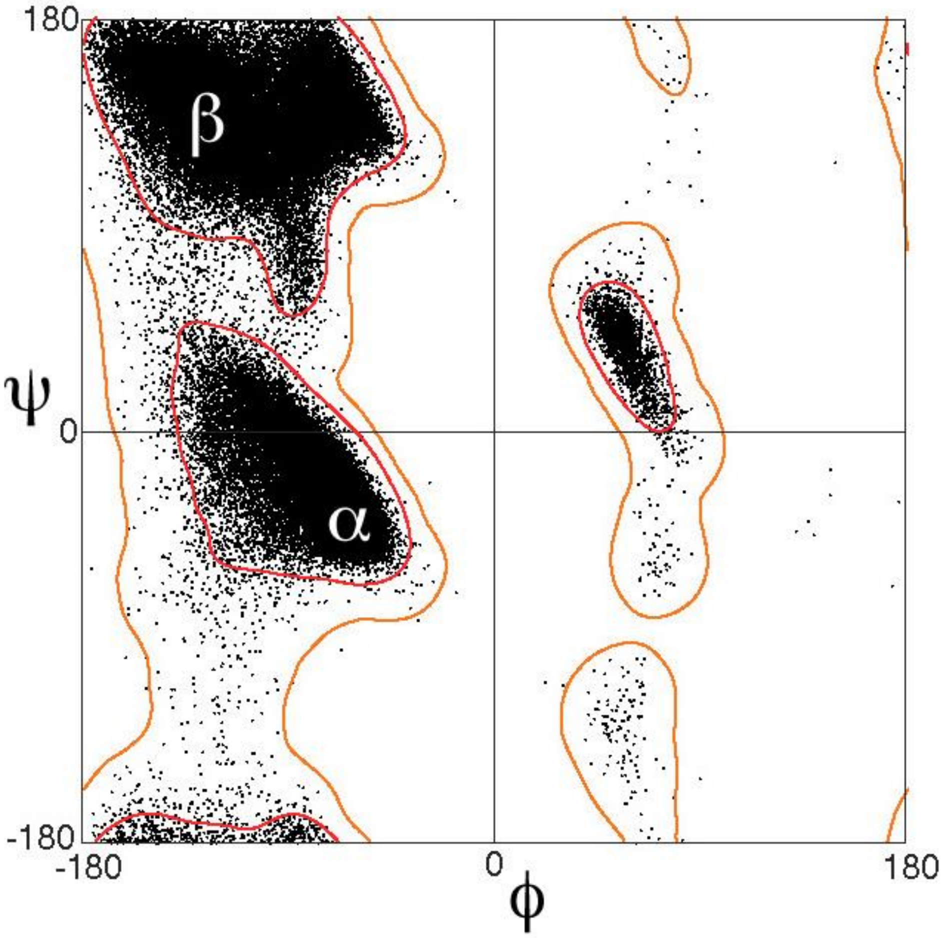


Fig.13. Ramachandran plot psi phi angle correlates secondary structure

The Ramachandran plot shows the correlation between secondary structure and the psi phi dihedral angle. such correlation is also observable on our 3D reconstruction

Reference

[1] Christopher A, Waudby, Hélène L, Lisa D.C, John C “Protein folding on the ribosome studied using NMR spectroscopy” Progress in Nuclear Magnetic Resonance Spectroscopy Vol 74, Oct 2013, P. 57-75

[2] Parker M.W. “Protein Structure from X-Ray Diffraction.” J. Boil. Phys. 2003; 29:341–362.

[3] Jonic S, and Vénien B.C. “Protein structure determination by electron cryo-microscopy.” Current Opinion in Pharmacology. 2009: 9. 636-642.

[4] Chou PY, Fasman GD. "Prediction of protein conformation". Biochemistry.1974; 13 (2): 222–245.

[5] Roy A, Kucukural A, Zhang Y (2010). "I-TASSER: a unified platform for automated protein structure and function prediction". Nature Protocols. 5 (4): 725–738.

[6] Sample, Ian. Dec 2018. "Google's DeepMind predicts 3D shapes of proteins". The Guardian. Retrieved 11 April 2020

[7] Garnier J, Gibrat J-F, Robson B. “GOR method for predicting protein secondary structure from amino acid sequence.” Methods in enzymology. 1995; 266:540–553.

[8] Pollastri G, Przybylski D, Rost B, Baldi P. “Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles.” Proteins: Structure, Function, and Bioinformatics. 2002;47:228–235.

[9] Dor O, Zhou Y. “Achieving 80% ten-fold cross-validated accuracy for secondary structure prediction by large-scale training.” Proteins: Structure, Function, and Bioinformatics. 2007; 66:838–845.

[10] P. J. Russell, iGenetics: A Molecular Approach, Third ed., PEARSON, 2009

[11] Berman H, Westbrook J, Feng Z, Gilliland G, Bhat T, Weissig H, Shindyalov I, Bourne P. The protein data bank. Nucleic Acids Res. 2000;28:235–42.

[12] Li Z, et al. “Identification of 14-3-3 proteins phosphopeptide-binding specificity using an affinity-based computational approach.” PLoS One. 2016

[13] Frank D, Class Lecture, Topic: “Lecture 1: Protein Structure.” Structural Bioinformatics/Genome 54. Department of Genome Sciences, University of Washington, Spring, 2020

[14] Li, H., Hou, J., Adhikari, B. et al. “Deep learning methods for protein torsion angle prediction.” BMC Bioinformatics 2017; 18, 417