Thesis for Bachelor’s Degree

Protein Secondary Structure Prediction using Protein Sequence Information and Deep Learning Method

Giyeon Shin

Electrical Engineering and Computer Science Concentration

Gwangju Institute of Science and Technology

2018

Protein Secondary Structure Prediction using Protein Sequence Information and Deep Learning Method

단백질 서열 정보와 딥 러닝 방법론을 이용한 단백질 이차 구조 예측

Protein Secondary Structure Prediction using Protein Sequence Information and Deep Learning Method

Advisor : Professor Hojung Nam

By

Giyeon Shin

Undergraduate: Electrical Engineering and Computer Science Concentration

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Bachelor of science of in the Electrical Engineering and Computer Science Concentration.

Gwangju, Republic of Korea

2017. 12. 29.

Approved by

Hojung Nam

Professor Hojung Nam

Committee Chair

Protein Secondary Structure Prediction using Protein Sequence Information and Deep Learning Method

Giyeon Shin

Accepted in partial fulfillment of the requirements of the degree of Bachelor of Science

12. 29. 2017.

Committee Chair Hojung Nam

Prof. Hojung Nam

Committee Member

Prof. Jong Won Shin

BS/EC

20145076

Giyeon Shin(신기연). Protein Secondary Structure Prediction using Protein Sequence Information and Deep Learning Method (단백질 서열 정보와 딥 러닝 방법론을 이용한 단백질 이차 구조 예측).
Electrical Engineering and Computer Science Concentration. 2017.
Advisor Prof. Hojung Nam

Abstract

Verifying structure of proteins will be helpful in chemistry and biology field. Because it is important to know which features the proteins have in researching new drug development or body mechanisms. These features of proteins is especially determined in protein secondary structure level. However, since the number of protein data having the protein secondary structure label is small, it is very hard to clearly understand the features of the protein via the protein secondary structure. In this study, we predicted protein secondary structure with protein primary sequence. In protein primary sequence, the protein fragment cut into n-grams was embedded to predict the protein secondary structure. This embedded protein fragment existed as a mathematical vector (protein vector). We then made the new prediction models through the embedded protein fragments and deep learning method, as well compared the performance of the models. Deep learning methodologies such as convolutional neural network and recurrent neural network were used to extract the characteristic of the local and whole parts in protein during the prediction modeling process. Finally, we examined the effect of the embedded protein fragments on the prediction of protein secondary structure.

List of Contents

List of Tables

List of Figures

1. Introduction

1.1 Motivation

Verifying secondary structures of proteins is important to develop new drugs or research body mechanisms. Since secondary structures of proteins are maintained by some chemistry bonds between the backbone chain and the side chain structures that make up the protein, the features of protein are determined in the secondary structures. For instance, there are hydrogen bonds, hydrophobic bonds, ionic bonds and disulfide bonds in secondary structure of protein. The stability from these bonds plays a role in determining significant features such as active site of protein.

Name	3- Class(Q3)	8-Class(Q8)
$\alpha - \text{helix}$	H	H
$\beta - \text{strand}$	E	E
loop or irregular	C	L
$\beta - \text{turn}$	C	T
bend	C	S
$3_{10} - \text{helix}$	H	G
$\beta - \text{bridge}$	E	B
$\pi - \text{helix}$	C	I

Table 1. Description of protein secondary structure Q3 and Q8 classes in this study dataset.[1] Q3 classes of protein secondary structure can be further classified into Q8 classes of protein secondary structure.[1]

However, there are few protein data having their secondary structure label. There are two kinds of protein secondary structure classification, Q3 prediction and Q8 prediction (Table 1). Q3 prediction classify protein secondary sequence into helices, sheet and coils. In Q8 prediction, there is eight protein secondary sequences. [2] Especially, since the number of Q8 secondary structure data of protein is small, it is difficult to use information of features in the Q8 secondary structure. As there is less information on currently known protein secondary structure, the extent to which it can be practically used is limited. The data is small that protein secondary structure cannot be used even though it determines the features of the protein. If we directly analyze protein secondary structure, the job will take a lot of time and money. So we decided to create a model to predict protein secondary structure.

[1] http://www.princeton.edu/~jzthree/datasets/ICML2014/

1.2 Related works

Many papers have been reported to predict protein secondary structures using machine learning and deep learning method. Sujun Hua *et al*. used support vector machine for protein secondary structure prediction [3]. They dramatically increased the accuracy of protein secondary structure prediction in that era, in a machine learning method called support vector machine, which maximizes the margin function for classification. Sheng Wang *et al*. reported a paper about feature extraction of protein secondary structure using convolutional neural network(CNN) [4]. They proposed 2D convolutional neural network for extracting the features of protein secondary structure. Sønderby *et al*. reported a paper about a method of protein secondary structure prediction with long short term memory networks. They use long short term memory in recurrent neural networks for treating protein sequences. When a long short term memory is operated on recurrent neural network(RNN), it is more noticeable in spatial sequences because it recognizes and memorizes the importance of information rather than the importance of time. [1] Zhen Li *et al*. user cascaded convolutional and recurrent neural networks for protein secondary structure prediction. They used convolution layers of convolutional neural networks, because the determination of protein secondary structure in a protein primary residue is affected by the surrounding residues. They also embed the input data in one-hot state and recombine it with one-hot input data to recreate the input data. Then, they extracted the feature map from the reconstructed input data through the convolutional layer, insert the extracted feature map into BGRU layers, and predict protein secondary structure through full-connected layers [5, 6].

These papers have generally methods of extracting features between protein sequences and using that feature to predict the secondary structure of protein. Taking these deep-learning methods into considerations, we examined accuracy of protein secondary structure prediction in deep-learning networks. In this paper, we checked how input data is affected by the performance of the model when it is cut in n-gram units without using the input data as itself in the process of embedding. We determined how the protein primary sequence affects the prediction of secondary structure when treated as a natural language.

1.3 Overview

For this study, we constructed deep-learning models for protein secondary structure predictions (Figure 1). At first, protein primary sequence data are taken from *Troyanskay*a 2014 [7]. The dataset consists of protein primary sequences labeled with secondary structure. We used Cullpdb-6133-filtered dataset and CB513 dataset for this study. There is no duplicate data between them. After that, we divided the data into n-gram units and learn it with embedding vector. We then converted the sequence data into learned embedding vector. The original data, one-hot vector, is recombined with both side one-hot vector and the learned embedding vector, which in turn predicts the final output via Convolutional layers, BiRNN layers, and Full-Connected layers. To understand the characteristics of protein vector, we refer to Zhen Li *et al*, 2016 [5] as an architecture and parameter of the overall deep learning.

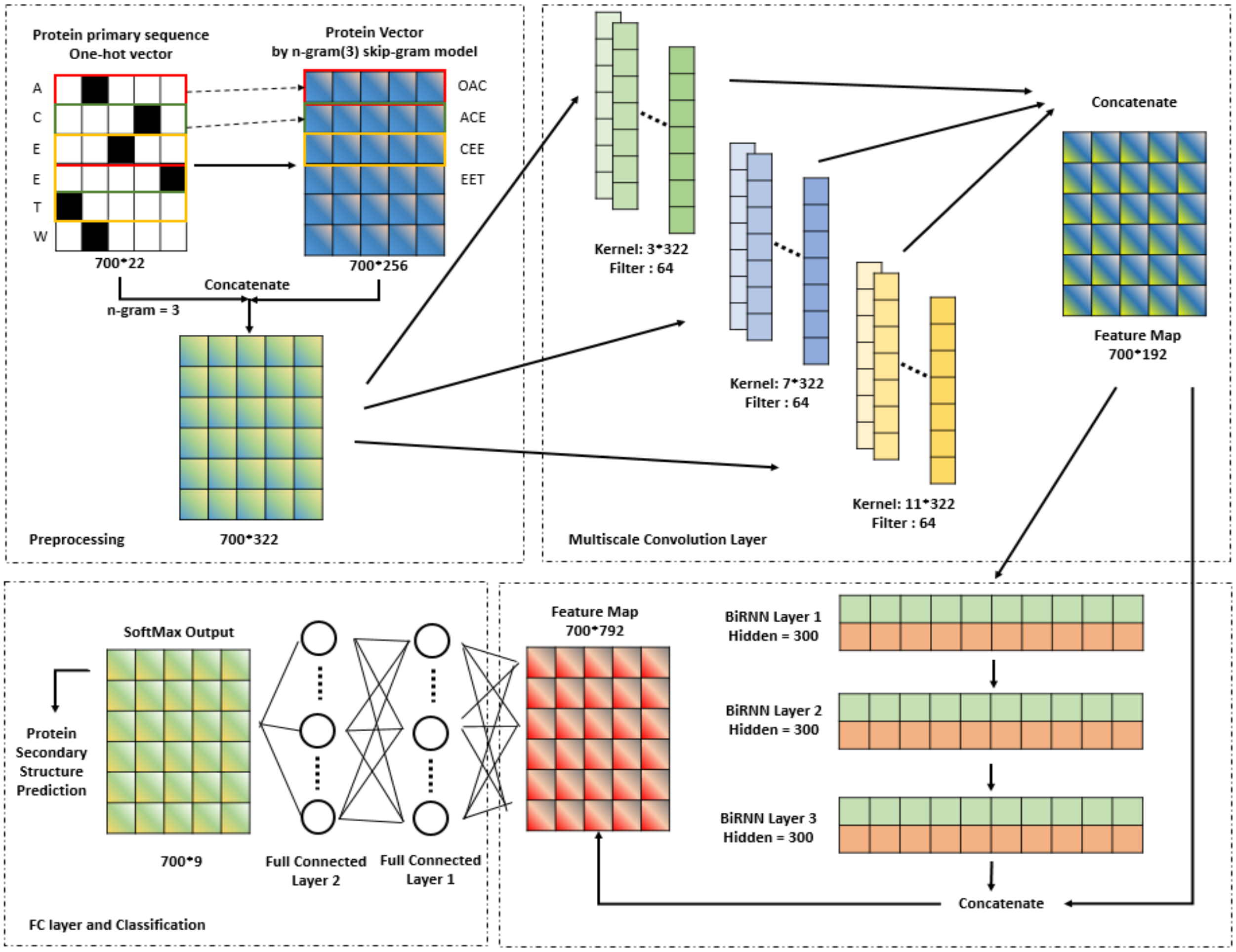

Figure 1. The flow chart of this study.

2. Materials and Method

2.1 Materials

For this study, we used protein primary sequence labeled with secondary structure. The protein primary sequence data are taken from *Troyanskay*a 2014 [7]. We used Cullpdb-6133-filtered dataset and CB513 dataset for this study and there are no duplicate datasets between them. We used Cullpdb-6133-filtered dataset for training set and CB513 dataset for test set. The maximum length of sequences in the datasets is 700. Since the length of the protein in the datasets is not fixed, it has undergone a padding process to fix the length to 700. The number of data in Culllpdb-6133-filtered is 5534, and the number of data in CB513 used for test is 514. The frequency of the protein secondary structure in the training data set and the test data set is as follows.

Name	Q8 classes	Training Set Frequency	Test Set Frequency
$\alpha - \text{helix}$	H	0.3454	0.3086
$\beta - \text{strand}$	E	0.2178	0.2125
loop or irregular	L	0.1919	0.2114
$\beta - \text{turn}$	T	0.1128	0.1181
bend	S	0.0826	0.0981
$3_{10} - \text{helix}$	G	0.0391	0.0369
$\beta - \text{bridge}$	B	0.0103	0.0139
$\pi - \text{helix}$	I	0.0002	0.0004

Table 2. The frequency of the protein secondary structure in the datasets.

2.2 Word2Vec

In nature language processing (NLP), there are two ways to learn words in contexts. One of them is to learn words by one-hot vector, and other thing is to learn words by quantified vector. In one-hot vector, one of vector dimension refers to a word itself. So, there is no relationship about the similarity between words because one dimension of vector represents the word itself. Eventually, one-hot vector means which number the word has on the dictionary. In quantified vector, the dimension of it refers a hidden feature of the word. This quantified vector is called word embedding vector and the similarity between words is learned in this vector [8]. The method of learning a word as an embedding vector is called Word2Vec [9].

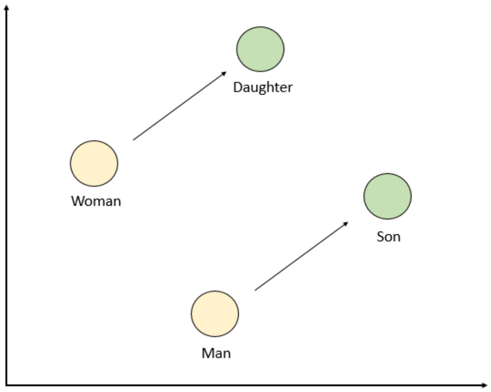

Figure 2. Visualization of concept about Word2Vec model.

As we mentioned earlier, Word2Vec model learns about the relationship between words, so we can easily relate these words to the other words. For example, in Figure 2, the following equation can be established (Figure 2):

$$\overrightarrow{Woman} - \overrightarrow{Daughter} = \overrightarrow{Man} - \overrightarrow{Son}$$

2.3 Protein Vector [10]

This word embedding vector method can also be applied in protein units. For examples, sequences of DNA, RNA and proteins can be described as a language. By recognizing proteins as one language and acting like NLP, we may be able to get hidden features in proteins. Obtaining the hidden feature is very important biologically and chemically, since it means getting the relationship between the surrounding protein fragments. For this purpose, the protein characteristic that is recognized as a word and converted into an embedding vector is called a protein vector.

To learn a protein as a protein vector, it is necessary to make the protein into words and context like natural language. Then, the context becomes a protein primary sequence and the word becomes a subsequence of the sequence. The technique of making subsequence in proteins to learn sequences involves fixed-length overlapping n-grams (Figure 3). These split n-gram units of protein are learned as a protein vector with a sliding window technique in one protein. At this time, the sliding window technique is the same as the method used in Word2Vec. In this study, the training sample is extracted using the skip-gram model among the methodologies used in Word2Vec (Figure 4).

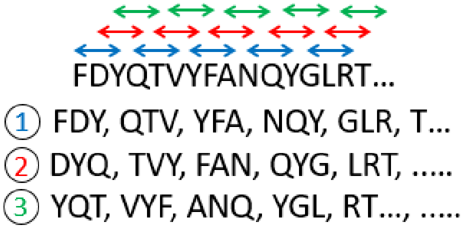
FDYQTVYFANQYGLRT...
1 FDY, QTV, YFA, NQY, GLR, T...
2 DYQ, TVY, FAN, QYG, LRT, ...
3 YQT, VYF, ANQ, YGL, RT..., ...

Figure 3. Visualization of splitting by n-gram (3-gram).

In the skip-gram model, it is an architecture to infer the appearance of the word appearing around the present given word. It is an architecture that puts the current input word into the input layer as a one-hot vector, projects it from the hidden layer, multiplies the weight again at the output layer, and finally predicts the surrounding word [11].

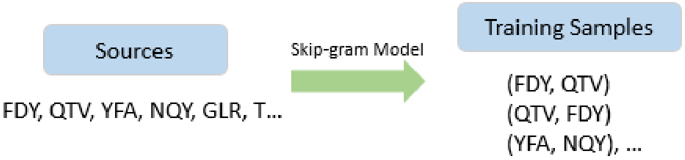

Figure 4. Training samples for skip-gram model in Protein Vector.

Even when learning the protein vector, simply convert the n-gram protein unit into a one-hot vector and learn it with Word2Vec.

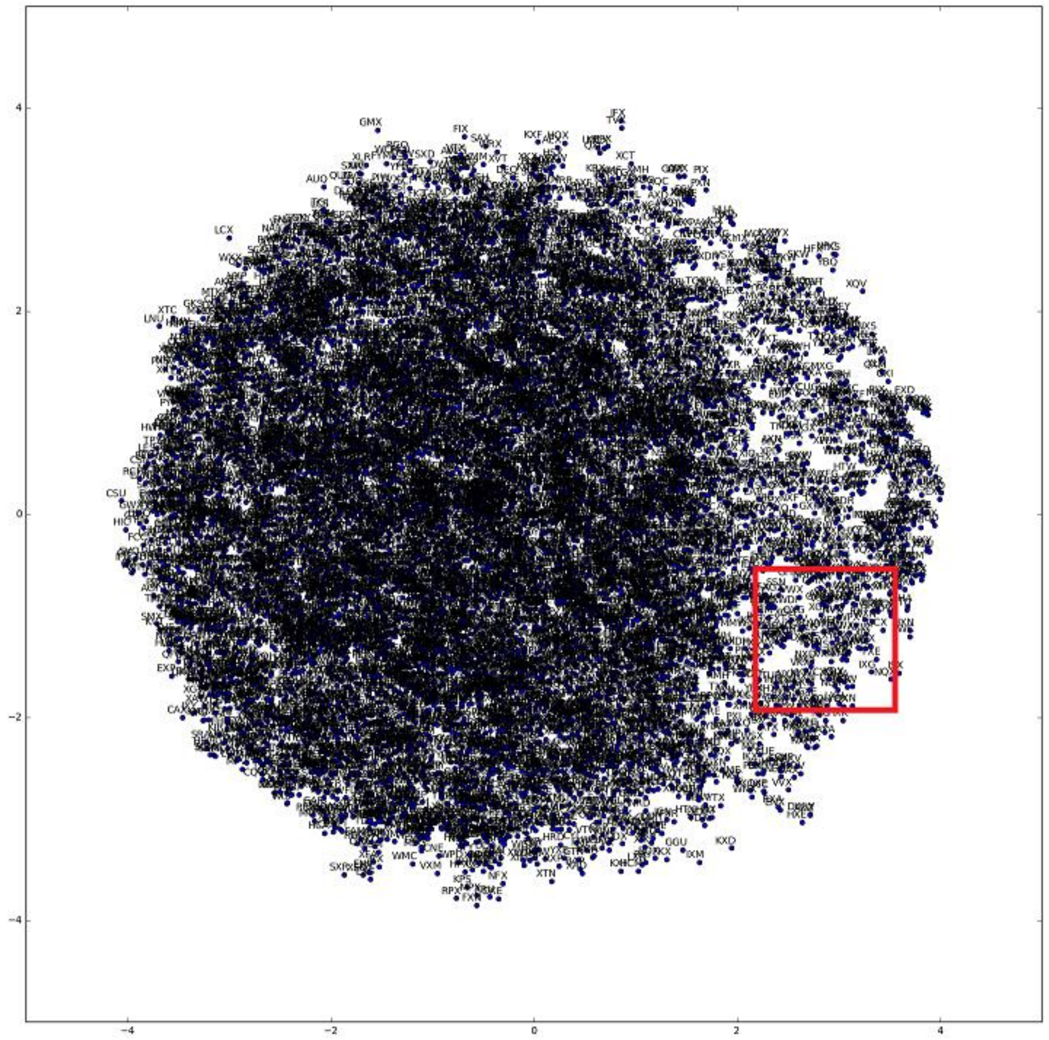

Figure 5. Visualization of Protein Vector.

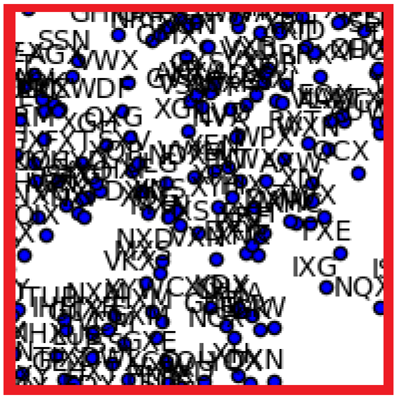

Figure 6. Magnified visualization of Protein Vector.

2.4 Convolutional layer

Convolutional layer is a technique commonly used in protein secondary structure prediction. In particular, techniques such as the deep-convolutional neural network can learn a very complex relationship between input data and label data [12]. The convolutional neural network is a deep learning method that emphasizes the image part differently from the recurrent neural network because it has the advantage of extracting the features of the local part in the convolutional layer as a filter. In the case of protein secondary structure prediction, the closest residues in one residue have the greatest effect, and the conformational layer can well characterize such effects.

In the convolutional layer, the window technique is used to extract features from different size filters. At this time, the size of the output from the convolutional layer is as follow: $sequence\ length - filter\ size + 1$. However, since the number of labels is equal to the sequence length, we use padding method in the convolutional layer to match the number of labels and output size of convolutional layer.

2.5 Gated Recurrent Unit [13]

Gated Recurrent Unit(GRU) is a type of cell used in recurrent neural networks. A GRU cell remembers a value for either long or short time period. This is the biggest feature of GRU, and it is the biggest advantage that can appear in protein secondary structure prediction. In the protein secondary structure, three-dimensional spatial properties of proteins exist. That is, residues far from one residue can have a large impact even in the primary sequence of a protein. GRU is optimized for this property. The GRU consists of an input gate, an output gate, update gate and reset gate, and these gates usually specify activation using a logistic function

Recent papers have shown good performance in protein secondary structure prediction using GRU layers. We will also make a deep-learning model for protein secondary structure using a similar bidirectional GRU layers. Bidirectional RNN learns RNN cells in both directions unlike normal RNN, which prevents the loss of spatial features for sequences coming in time sequentially [6].

2.6 Experimental Setup [5]

The model of the overall deep learning method is similar to Zhen Li et al, 2016 [5]. Their combination model of convolutional neural network and recurrent neural model is used in this study. We made and compared a total of two models. One is with input data of one-hot vector, and the other one is with input data of 3-gram embedding vector and 3-gram one-hot vector.

In the protein vector architecture, the parameter of batch size in the model is 256, and number of steps is 250,000 times, making training data set sufficiently learned. The embedding size of protein vector is 50, number of skips is 2, and skip of window is 1, so that when one n-gram is drawn, both n-grams on both sides can be learned. At this time, the n-gram was set to size 3.

After that, the input data goes through convolutional layers. In this case, the window sizes are 3, 7, and 11, respectively. Since each filter number is 64, a total of 192 features are displayed per residue. The data through the convolutional layers goes through Bidirectional GRU layers, where the layer is 3 and the number of hidden units in the RNN cell is set to 300.

In the combined architecture of CNN and RNN, we used AdamOptimizer for optimizing, the learning rate was 0.0003, the size of batch was 48, the number of epochs was 4000, and the dropout_keep_probability parameter was 0.5 which was applied to CNN and RNN. The L2 loss parameter was applied only to the weight of the full-connected layer and the convolutional layer, and the lambda value of the L2 loss was set to 0.001.

3. Results

3.1 Preprocessing: One-hot vector input data

The Q8 accuracy of the model was 0.540 when the protein vector was not added to the input data and only used as one-hot vector information. When we trained the deep learning model, Cullpdb-6133-filtered dataset was used to train the model, and validation set was used by dividing part of the dataset in validation process during training. The final Q8 accuracy used the CB513 dataset and, as we mentioned earlier, there was no overlapping protein between training data set and test data set.

Name	Q8 classes	Test Set Frequency	Sensitivity	Specificity	Precision	Accuracy
$\alpha - \text{helix}$	H	0.3086	0.902	0.707	0.579	0.767
$\beta - \text{strand}$	E	0.2125	0.603	0.891	0.601	0.830
loop or irregular	L	0.2114	0.472	0.847	0.452	0.767
$\beta - \text{turn}$	T	0.1181	0.282	0.944	0.401	0.865
bend	S	0.0981	0.009	0.999	0.520	0.902
$3_{10} - \text{helix}$	G	0.0369	0.004	1.000	0.341	0.963
$\beta - \text{bridge}$	B	0.0139	0	1		0.986
$\pi - \text{helix}$	I	0.0004	0	1		1.000

Table 3. The result in one-hot vector input data

In this table, sensitivity means the true positive rate, which is how similar the actual prediction results to the protein secondary structure. Sensitivity results were the same as the order of magnitude of the test set frequency. Unusual is the sensitivity of the test set frequency which is similar to E and L of Q8 class.

3.2 Preprocessing: Input data with 3-gram one-hot vector

The Q8 accuracy of the model was 0.530 when the protein vector was not added to the input data and used as 3-gram one-hot vector. If the 3-gram one-hot vector is placed in the dimension of 21*21*21, the size is too large and the calculation becomes too large. To solve this problem, the concentration was processed in the dimension of 21+21+21.

Name	Q8 classes	Test Set Frequency	Sensitivity	Specificity	Precision	Accuracy
$\alpha - \mathrm{helix}$	H	0.3086	0.851	0.755	0.607	0.784
$\beta - \mathrm{strand}$	E	0.2125	0.666	0.840	0.530	0.803
loop or irregular	L	0.2114	0.414	0.864	0.449	0.769
$\beta - \mathrm{turn}$	T	0.1181	0.319	0.924	0.360	0.853
bend	S	0.0981	0.004	0.999	0.444	0.902
$3_{10} - \mathrm{helix}$	G	0.0369	0.003	0.999	0.125	0.963
$\beta - \mathrm{bridge}$	B	0.0139	0	1		0.986
$\pi - \mathrm{helix}$	I	0.0004	0	1		1.000

Table 4. The result in input data of 3-gram one-hot vector

The input data set was constructed by binding three consecutive residues and selecting the label as the secondary structure of the residue located in the middle of the three consecutive residues. The performance of the overall model was shown to be lower than that of the previous experiment, but the trend of overfitting in $\alpha - \mathrm{helix}$ that appeared in the previous model showed a decline.

3.3 Preprocessing: Input data with 3-gram one-hot vector and protein vector information

The Q8 accuracy of the model was 0.556 when the protein vector was added to the input data and also used as 3-gram one-hot vector information. Both the test data set, the validation set and the test set used the same data set as the previous experiments and showed slightly better Q8 accuracy then the previous experiments. The 3-gram one-hot vector created here was also made the same method as the second experiment, but we combined the three consecutive residues and the protein vector of these three consecutive residues to form an input data.

Name	Q8 classes	Test Set Frequency	Sensitivity	Specificity	Precision	Accuracy
$\alpha - \mathrm{helix}$	H	0.3086	0.845	0.810	0.665	0.821
$\beta - \mathrm{strand}$	E	0.2125	0.672	0.872	0.587	0.830
loop or irregular	L	0.2114	0.509	0.828	0.442	0.760
$\beta - \mathrm{turn}$	T	0.1181	0.371	0.919	0.379	0.854

bend	S	0.0981	0.018	0.997	0.415	0.901
$3_{10} - \text{helix}$	G	0.0369	0.011	0.999	0.24	0.962
$\beta - \text{bridge}$	B	0.0139	0	1		0.986
$\pi - \text{helix}$	I	0.0004	0	1		1.000

Table 5. The result in input data of 3-gram one-hot vector and protein vector

Compared with the previous experiments, $\alpha - \text{helix}$ with the maximum frequency in the test data set and the training data set showed a lower sensitivity, but higher specificity, precision, and accuracy. The performance of the model is more uniform compared to the previous model.

4. Conclusion

Compared with the first experiment and the second experiment, we showed slightly higher Q8 accuracy in the third experiment. It can be seen that the protein vector feature is different from the one-hot vector that can be obtained when working on CNN or RNN.

Especially in the first experiment, as shown in [Table 2], overfitting tended to be due to the frequency of the training data set. The training set frequency of $\alpha - \text{helix}$ is 0.3454, which corresponds to about 1.12 times of the test set frequency. This phenomenon eventually increases only the sensitivity of the alpha helix and decreases in specificity, precision, and accuracy.

On the other hand, when approaching with n-gram method, such overfitting tendency was relaxed and specificity, precision and accuracy tended to increase. In this sense, approaches to n-grams tend to reduce overfitting, which affects dataset frequency.

In addition, comparing the second experiment with the third experiment shows that the third experiment has an overall upward performance. From the overall Q8 accuracy to the overfitting problem that can be caused by the frequency, the overall performance result is better. This is thought to be due to the protein vector generalizing n-gram protein units and quantifying them. Generalized protein units prevent frequency-induced overfitting and extract features that are not found in CNN or RNN in protein vector.

5. Discussion

Although protein vectors may have an impact on protein secondary structure prediction, overall model performance is poor compared to current studies. One of the reasons is related to the padding method of input data. To use CNN and RNN, the sequence length of the input data mush be fixed. To do so, we used the padding method, which was to put the padded input data as a zero vector on a one-hot vector, but there is a severe bias in the prediction process when we inserted the label padding as a zero vector. On the other hand, if the padding is designated as a kind of one-hot vector without putting it as a zero vector, the phenomenon of bias disappears. The reason for this phenomenon is unknown, but if this phenomenon is resolved, it is expected to rise to the current paper-level accuracy. If a padding is specified as a kind of one-hot vector, there is a process of calculating the padding one-hot vector, and since the model needs to learn, the calculation of the padding is included in the model. If these calculations are removed from the model, the model will be able to compute without considering padding calculation, which will eventually improve the performance of the model.

The disadvantage of the protein vector appears in the test data set. We cut the training data set into n-gram units to make a dictionary for learning protein vector, and matched the n-gram unit with protein vector. However, if the n-gram unit appearing in the test data set is not in the dictionary, a problem arises. We used regular expression method to solve this problem [14]. At first, if the n-gram appearing in the test data set appeared in the dictionary, we used the corresponding embedding vector (protein vector). However, if the n-gram units that appeared in the test data set is not in the dictionary, referred to the temporary dictionary. The temporary dictionary which gathered n-gram units in the test data set, matched the n-grams that do not appear in the training data set with protein vectors. In this case, the method of matching n-gram and protein vector in the temporary dictionary is to consider how many same residues locate in same position. If the n-gram units have the same number of residues, the method consider how many n-grams the training data set have. However, this method has limitations in reflecting the characteristics of the protein n-gram unit of the test data set. If the architecture uses the similar protein vector, it has a limitation in reflecting the actual protein characteristics of the n-gram unit, and there is also a question that the n-gram unit in training data set and the n-gram unit in the temporary dictionary of the test data set are similar.

In addition, the lack of correlation between data in the training data set and the validation data set is also a factor that causes the influence of protein vector to fall in the deep learning model. Protein vectors have great significance in generalizing n-gram units in protein sequences. However, the data used, Cullpdb-6133-filtered and CB513, are just the data that the protein secondary structure is labeled, and there is no information about what kind of protein it belongs to. This is a generalization of the proteins found in data set, which means that the protein vector extracts only the common characteristics of the protein from the training data set. If the target of this deep-learning model is to find the secondary structure of a particular protein class, it may have a greater advantage in predicting it, since it can generalize certain protein class features to protein vectors.

6. Summary

단백질 서열과 딥러닝 방법론을 이용한 단백질 이차 구조 예측

단백질의 특성은 단백질을 매개체로 하는 딥러닝 연구 분야에서 매우 중요한 역할을 한다. 단백질의 특성은 단백질의 이차구조에서 대부분 결정되기 때문에 단백질의 이차 구조를 예측하는 일은 상당히 중요하다. 허나, 단백질의 이차 구조가 밝혀진 단백질 일차 서열의 데이터 수는 현저히 적다. 본 연구에서는 이러한 문제를 해결하고자 단백질 이차 구조를 단백질 서열과 딥러닝 방법론을 이용해 예측하고자 한다. 단백질 이차 구조가 알려진 단백질 일차 서열 데이터를 n-gram 으로 잘라 protein vector 로 만들었다. 이 때 자연어 처리 방식 중 하나인 Word2Vec 기법을 사용하였으며, protein vector 정보를 다시 n-gram one-hot vector 와 결합해 CNN, RNN, 그리고 FC layer 를 거쳐 단백질의 이차 구조를 예측하였다. Protein vector information 이 포함된 모델이 그렇지 않은 모델보다 약 1~2% 가량 높은 Q8 accuracy 를 가졌으며, Data 의 frequency 에 의해 나타나는 overfitting 현상을 줄여주는 경향을 보였다.

7. Reference

1. Sønderby S.K., Winther, W., "Protein Secondary Structure Prediction with Long Short Term Memory Networks", *arXiv preprint arXiv:1412.7828*, 2014.
2. Y. Liu, Y. Chen and J. Cheng, "Feature extraction of protein secondary structure using 2D convolutional neural network", *2016 9th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, Datong, 2016, pp. 1771-1775.
3. Hua, S., Sun, Z., "A novel method of protein secondary structure prediction with high segment overlap measure: support vector machine approach", Edited by B. Holland, *Journal of Molecular Biology*, Volume 308, Issue 2, 2001, pp. 397-407.
4. Zhang, C.T., Zhang, R., "A Refined Accuracy Index to Evaluate Algorithms of Protein Secondary Structure Prediction", *PROTEINS: Structure, Function, and Genetics*, Volume 43, Issue 4, 2001, pp. 520-522.
5. Li, Z., Yu, Y., "Protein Secondary Structure Prediction Using Cascaded Convolutional and Recurrent Neural Networks", *arXiv preprint arXiv:1604.07176*, 2016.
6. Schuster, M., Paliwal, K.K., "Bidirectional Recurrent Neural Networks", *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, Volume 45, No. 11, 1997, pp. 2673-2681.
7. Zhou, J., Troyanskaya, O.G., "Deep Supervised and Convolutional Generative Stochastic Network for Protein Secondary Structure Prediction", *arXiv preprint arXiv:1403.1347*, 2014.
8. Bengio, Y., Ducharme, R., Vincent, P., Jauvin C., "A Neural Probabilistic Language Model", *Journal of Machine Learning Research*, 3:1137-1155, 2003.
9. Mikolov, T., Chen, K., Corrado, G., Dean, J., "Efficient Estimation of Word Representations in Vector Space", *arXiv preprint arXiv:1301.3781*, 2013.
10. Asgari, E., Mofrad, M. R. K., "Continuous Distributed Representation of Biological Sequences for Deep Protomics and Genomics", *PLUS ONE 10(11): e0141287*, 2015.
11. Rong, X., "word2vec Parameter Learning Explained", *arXiv preprint arXiv:1411.2738*, 2016.
12. Wang, S., Peng, J., Ma, J., Xu, J., "Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields", *Scientific Reports*, Volume 6, 2016.
13. Chung, J., Gulcehre, C., Cho, K., Bengio, Y., "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling", *arXiv preprint arXiv:1412.3555*, 2014.
14. Conway, Damian, “Regular Expressions, End of String”, *Perl Best Practices*, p. 240, 2005.