Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

ECEN 766 Final Project

Cross Modality Learning on Proteins

Shaowen Zhu

Department of Electrical and Computer Engineering, Texas A&M University, College Station, TX 77843, United States.

Abstract

Motivation: Discovering sequence-structure-function relationship on protein molecules remains a great challenge for bioinformatics scientists. While the protein functions are highly related to the structures, the structures are also folded by protein sequences of 20 kinds of amino acids following some complex and intractable physical principles. With the quickly accumulating data on proteins sequences and structures, more and more people began to focus on data-driven methods and apply artificial intelligence models to learn the relationship automatically. On the other hand, to represent proteins there are also various modalities that reflect the information on different aspects, so a question this project is addressing is: will the information contained in other modalities help to learn the sequence-structure relationship?
Results: This project focused on the fold prediction problem which is to learn the structure category a protein would belong to. The fold prediction model was based on the state-of-the-art model DeepSF but with less input features. At the same time, another modality of the proteins, the secondary structure (SS) is also considered as its related prediction model was constructed and combined to the fold prediction model in order to get a better performance. The Top-1 accuracy of the DeepSF with only protein sequences as the input was 0.122, while it reached 0.502 when the SS was also taken as the input. The best SS prediction model attained an accuracy of 0.833, and when it was combined to the DeepSF model the Top-1 accuracy was improved to 0.131.
Availability: The codes for this project are available at `https://github.com/Shaowen1994/ECEN766_CourseProject`.
Contact: shaowen1994@tamu.edu

1 Introduction

Protein is a kind of essential biomolecules that plays various significant roles in our living cycle. While a protein molecule is simply made of one or multiple sequences of 20 kinds of amino acids, it can fold into a complex 3D structure which will also result in its certain function. Therefore there are various modalities for proteins that contain information on different aspects. For example, the protein sequences, secondary structures (SS) and solvent accessibility (SA) are 1-D modalities since they are of sequence format, and they contain the information of ordered amino acid, local structure and the ability to touch the water of each residue respectively; for 1-D modalities there are distance map and contact map that are matrices and reflect the pairwise distance between the residues; the structure of the proteins is a 3-D modality as each atom is assigned a 3-D coordinate. Besides, during the evolution of living things, one protein could easily mutate into another one, but proteins with similar amino acid (AA) sequences are likely to have similar structures and functions, so multiple sequence alignment (Carrillo and Lipman, 1988) is also applied for protein analysis, and then there are also modalities containing such information like the position-specific scoring matrices (PSSM). Despite the different modalities are not on the same aspect, they are not totally separated either as there can be shared part of the information, like the 3-D structure data also contains the information in the SS and SA. Therefore learning the relationship between different modalities can be a research topic as we would like to extract or predict one from another, and related models have been developed which will be introduced in Section 2. In many cases, for one protein not all of the modalities are available, and for some modalities the data can be rather scarce, so the model performance may be limited by the data size. Then here comes a question that whether the learned relationship between two modalities could help to learn that between another pairs, which refers to the cross modality learning problem? Since cross modality learning has gotten some breakthrough in computer vision and natural language process, whether we can learn the relationship between two different modalities, and applied the learned information to improve other modality learning problems, which is finally a cross modality learning problem on proteins.

This project focused on three kinds of protein modalities, the AA sequence, the SS and the structure, and the aimed at predicting the structure of a given sequence, which is to learn the sequence-structure relationship. Due to the usefulness and the intractability, the sequence-structure-function remains a fundamental question in protein science (Alberts *et al.*, 2015). And since the protein function is highly related to the structure, learn the sequence-structure relationship can be helpful on many applications like antibody detection, enzyme discovery and drug design. Besides, as the data size of the protein sequence keeps increasing, there is limited structure data because of the high cost of the experiments and it is even not feasible for some protein. As a result, there are much more unlabeled protein sequences (proteins that their structures are unknown) than the labeled ones (Consortium, 2019). Therefore, learning the relationship between the protein sequence and its structure becomes rather important and helpful. At the beginning such models were mainly driven by principle which is limited by our current interpretation on biology and the computational cost on exhausted search; as the data accumulates, some data-driven approach came out, but the performance was also impeded by the limited labeled data. In this project the structure data was not taken as the 3-D format but represented by the fold name since the structures that belong to the the same fold are similar, so the goal is to predict the structure category that the given sequence would fold into and it was a multi-label classification problem. The model was built based on the state-of-the-art model DeepSF (Hou *et al.*, 2018a) but only took the sequence as the input, while the original one concatenate the AA sequence, the PSSM, the SS and the SA together as the input and it would be time and resource consuming to get such features. As the fold labels would reflect the global structure of proteins, the SS contains the local structure information and may help to discover the global structure. Inspired by the cross modality learning idea, the Seq-SS relationship was also taken into consideration and efficient models have been built to learn the relationship. Experiments were firstly done to find out whether concatenate the SS information with the sequence would increase the DeepSF accuracy. While the SS prediction models were available, the predicted SS were applied instead of the real ones in order to find out whether this method would by effective when only sequence data were available.

2 Related Work

For different modalities, multiple programs have been developed to discover the relationships between them. While the protein structure data with known sequence contains almost all the information for other modalities and we can directly extract the SS and contact map from it, and we can simply detect the second structure from the contact map, we need some principle or data driven method to discover some other relationships. For example, SCRATCHCheng *et al.* (2005) provides softwares to predict SS from a given sequence with an accuracy higher than 90% and PSI-BLAST (Bhagwat and Aravind, 2007) can calculate the PSSM (or sequence profile) for a given sequence, but both of the methods are time-consuming (minutes just one sequences). There are also some contact map predictions programsDi Lena *et al.* (2012) and structure reconstruction programs based on contact map like Pietal *et al.* (2015) and Vassura *et al.* (2008). For these methods, the accuracy varies depending on the modality complexity, but the results from one model may benefit another, like the SS predictor can detect the local structure and is likely to help determine the global structure.

For the sequence-structure relationship, various related research topics emerged such as structures prediction on protein sequences(Baker and Sali, 2001) and sequences design for certain structures (Pabo, 1983; Street and Mayo, 1999). The forward problem of protein structure prediction, especially *ab initio* prediction without templates, has often been solved by energy minimization. For data-driven approaches, DeepSF Hou *et al.* (2018b) applied a 1-D convolutional neural network (CNN) to predict protein fold labels on sequences. There are also inverse programs of protein sequence design. cVAEGreener *et al.* (2018) and gcWGANKarimi *et al.* (2019) constructed generative models to generate novel protein sequences given a desired fold.

3 Materials and Methods

3.1 Data

For the labeled data this project applied the ASTRAL SCOPeO'maille *et al.* (2002) 2.07 dataset (genetic domain sequence subsets, based on PDB SEQRES records) since the proteins are well hierarchical classified and their labels (proteins folds) are provided. While there are subsets with different redundancy, the subset with less than 40 percent identity was considered due to the limited time and computational source. There are 14,665 sequences of 14,323 proteins in all, covering 1,232 folds of 7 protein classes (from a to g). The lengths of the sequences vary from 6 to 1,664. Table 1 shows the statistics of the data size for different classes and Figure 1 shows the sequence distribution of different lengths.

protein classes	a	b	c	d	e	f	g
# of sequences	2,540	2,895	4,160	3,462	279	272	715
# of proteins	2,555	2,983	4,337	3,502	285	282	722
# of folds	1,232	1,232	1,232	1,232	1,232	1,232	1,232

Table 1. Data size statistics for different classes (from a to g).

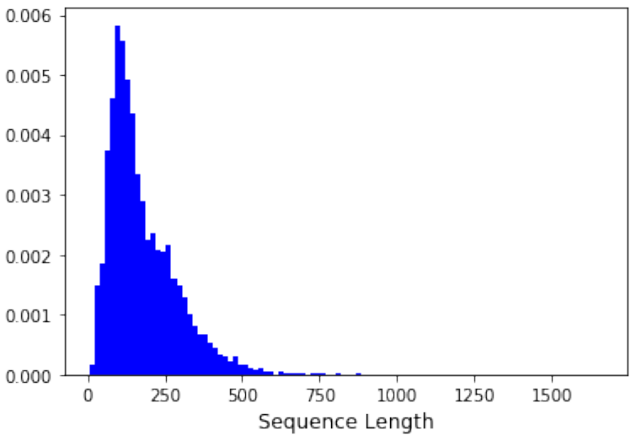

Fig. 1: Sequence length distribution.

3.1.1 Pre-processing

For simplicity, only the proteins with single chains are considered, so that the proteins and the sequences are of one-to-one relationship and for each sequence there will be one fold label. According to Figure 1, the majority sequences are of lengths shorter than 500, so this project only took the sequence shorter than 512 (or 2^9) for the following steps. After that there were 13,814 sequences (proteins) and 1,206 folds left. The lengths of the sequences are between 20 and 512.

3.1.2 Protein Segments for Secondary Structure Prediction

The structure data of the proteins were downloaded from the Protein Data Bank (PDB) and DSSP was applied to extract the SS information. While

the output SS of DSSP are of 8 classes, this project only consider the 3-class SS (helix, strand and coil) and map the 8-class form into that of 3. Since there are missing residues or abnormal ones, and sometimes there can be residues that DSSP cannot recognize, I simply removed such cases to make sure the sequence data and the SS are completely paired. After that there are 8845 sequences remaining and they were split into training, validation and test sets according to the ratio of 7:1:2. The statistics are shown in Table 2.

	Training	Validation	Test
# of sequences	7,076	885	1,769
# of residues	975,134	138,952	269,936

Table 2. Data size statistics for training, validation and test datasets.

The SS prediction was a residue-wise program. The data directly applied were the odd-length protein segments split from the original sequences with the SS of the middle residue to be the label (add padding for the residues at the beginning or the end). According to Table 2 we can see there are tens of thousands residues which refers to the data size. For simplicity I randomly selected 10 percent of them and finally got 94,759 segments for training and 13,516 for validation. The test data were not split into segments cause the trained models were finally directly applied to the whole sequences like a sliding window to get a complete predicted SS sequence, and compare it with the real one.

3.1.3 Data for Fold Prediction

Figure 2 shows the sequence amount of different folds, and we can see that for some folds there are only few sequences. For fold prediction we need all the folds to appear in the training set, and would like they all also appeared in the test set. Therefore I removed the folds with less than 3 sequences and then randomly split the data into training, validation and test sets by 7:1:2 but ensure that if there are only 3 sequences for a fold then they will separately assigned to the three sets. Finally I got 5,498 sequences for training, 856 for validation and 1,778 for test, covering 420 folds. The data were transformed into the input format of DeepSF accordingly.

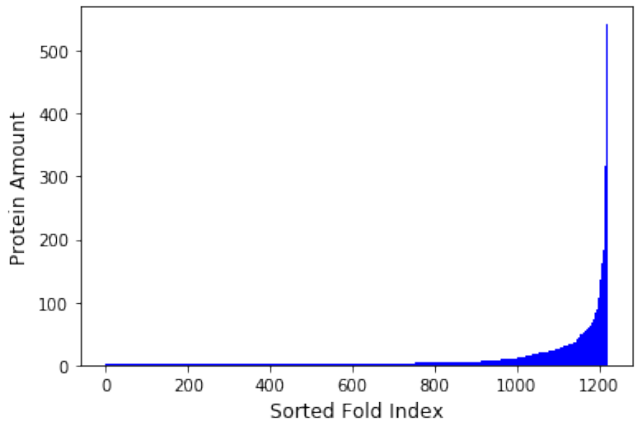

Fig. 2: Sequence distribution of different folds.

3.2 Secondary Structure Prediction

Since SS captures the local structure of proteins, ir would be more likely influenced by the local residues. Then here comes a simple idea that we could split the protein sequences into small segments as the features, and take the SS of the middle residue as the label. Then applying some simple algorithms we could develop a efficient model to make quick SS prediction. In the project I tried a shallow model, support vector machine (SVM), with different kernels and a 4-layer fully connected neural network (FCNN) for the experiments. Different length of the segments have also been tried.

3.3 Fold Prediction

The fold prediction model was built based on DeepSF, but only took the one-hot encoding sequences or sequences with SS as the input, since he original DeepSF concatenates the one-hot encoding sequences, PSSM, SS and SA to be the input, and the other three features can be hard to acquire due to the time and resource limit. The original DeepSF applied a 1-D CNN as the kernel would only slide along the direction of the sequence. A 30-max pooling layer that select the 30 most important hidden row vectors was applied to fix the various sequence length issue. Since in the original DeepSF input SS can help the prediction, the idea is that if we have a quick SS prediction model, then we can simply concatenate the model with DeepSF and also take the predicted SS as the input so as to increase the model accuracy. Figure 3 illustrates the idea of this modified DeepSF.

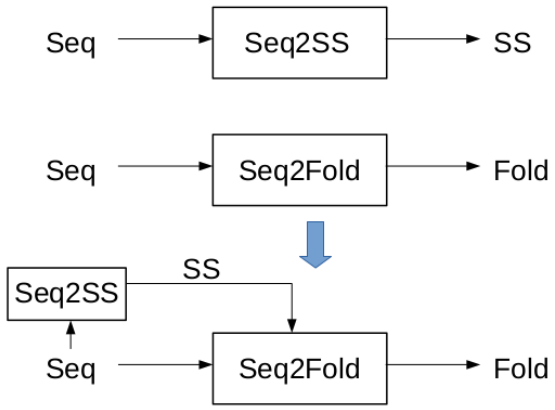

Fig. 3: DeepSF concatenated with the quick SS prediction model.

4 Experiments and Results

4.1 Secondary Structure Prediction

4.1.1 Support Vector Machine

For the SS prediction the segment lengths of 3,5 and 7 were tried for the experiments. Linear SVM and kernel SVM with radial basis function (RBF) kernel were implemented and trained on the protein segments. Table 3 shows the performance results of different SVM models. While the validation accuracy was also calculated based on sequence segments, the test accuracy was based on the complete sequences as the model slide from the beginning of the sequences (with padding) to the end, and then get a complete SS sequence prediction (so did the FCNN in Section 4.1.2). The residue-wise accuracy was then calculated by comparing the predicted SS sequences to the real ones. In this way it can also be interpreted as the test process was also on the protein segments but for all the residues rather than the randomly selected ones.

From Table 3 we can see that the the longer the segments are, the better performance we would finally get, but it would cost more time for the training and test process. However, compared to SCRATCH, the inference time is significantly shorter since we can get the prediction of thousands of sequences (1,769 test sequences in all) in just several seconds. The best SVM model was the RBF kernel SVM with a test accuracy of 0.802.

	Linear SVM				RBF Kernel			
Segment Length	Accu-Vali	Accu-Test	Train-Time	Infer-Time	Accu-Vali	Accu-Test	Train-Time	Infer-Time
3	0.539	0.722	1246.3	0.772	0.539	0.722	1511.9	1.043
5	0.585	0.762	2950.0	1.326	0.587	0.770	1806.1	1.984
7	0.616	0.794	5812.0	1.639	**0.616**	**0.802**	2248.3	2.010

Table 3. SS prediction results on SVM models.

4.1.2 Fully Connected Neural Network

Similar to the process mentioned above, a fully connected neural network with 3 hidden layers (100 nodes each layer) was trained on the protein segments of length 3, 5 and 7 as well. The results are shown in Table 4. Compared to SVM the FCNN performed better since it got higher accuracy with costing much less training and inference time. The best FCNN model was also based on the segments of length 9, and it got a test accuracy of 0.833 and could make predictions for thousands of sequences in less than one second. Then this model was applied to predict the SS of all the protein data for the fold prediction process.

Segment Length	Accu-Vali	Accu-Test	Train-Time	Infer-Time
3	0.553	0.706	221.5	0.045
5	0.591	0.738	226.0	0.032
7	**0.611**	**0.833**	230.1	0.057

Table 4. SS prediction results on FCNN models.

4.2 Fold Prediction

4.2.1 DeepSF with Different Inputs

For the fold prediction, firstly the original DeepSF was modified to only take the one-hot encoding sequences as the input and this model was called DeepSF (AA). In order to show that taking SS as input as well can help improve the performance, the second version of DeepSF was developed that both the sequence and the SS are concatenated together to make a 23-dimensional feature vector for each residue. This version was called DeepSF (AA + SS). The results in Table 5 shows that with SS the accuracy was significantly improved, so the SS can help the model to perform better. Then the third version, DeepSF (AA + SS_{pre}), was built that combine the DeepSF with the previous best SS prediction model (FCNN on 9-residue segments) and take the AA sequence and the predicted SS as the input. The illustration of the three versions of DeepSF are shown in Figure 4 and the results are shown in Table 5. All the DeepSF models were trained for 100 epochs. With the predicted SS the performance was also improved but not too much, even though the SS prediction accuracy can reach 0.833 in the previous experiments.

Accuracy	DeepSF (AA)	DeepSF (AA + SS_{pre})	DeepSF (AA + SS)
Top 1	0.122	0.131	0.502
Top 5	0.307	0.317	0.757
Top 10	0.422	0.438	0.830
Top 15	0.503	0.507	0.879
Top 20	0.558	0.569	0.903

Table 5. Fold prediction results with different inputs.

4.2.2 DeepSF on Mixed Data

Considering the previous results that the real SS can significantly increase the accuracy of DeepSF but the predicted SS only made a small improvement, another experiment was developed to discover the influence on the origin of the SS (real or predicted) input. The experiment was done by mixing the real data and the predicted SS together according to a certain ratio, and then repeat the DeepSF training and test process to compare the results. The experiment outcomes are shown in Table 6. With the portion of real SS got larger, the performance of also got better for the accuracy was getting higher. From 0% to 75% of real SS, the accuracy increased relatively slow, but when the portion was changed from 75% to 100% (all the SS were from the real set), there was a huge improvement that the Top-1 accuracy became 2 times more than before. Therefore the DeepSF model with SS is sensitive to the purity of the SS data. Even a small portion of predicted or fake data could seriously effect the performance.

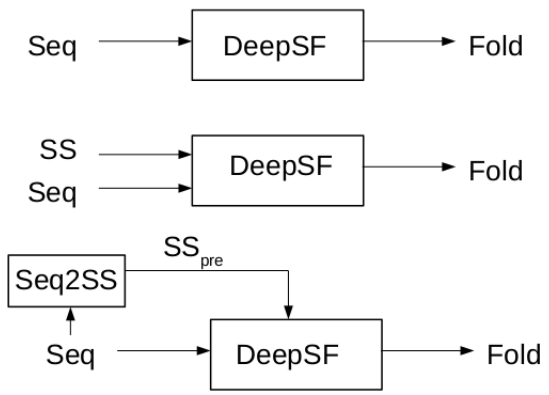

Fig. 4: Illustration of the 3 cases to train the DeepSF model.

Accuracy	0% (All Predicted SS)	25%	50%	75%	100% (All Real SS)
Top 1	0.131	0.173	0.218	0.236	0.502
Top 5	0.317	0.394	0.454	0.498	0.757
Top 10	0.438	0.512	0.578	0.612	0.830
Top 15	0.507	0.588	0.643	0.682	0.879
Top 20	0.569	0.641	0.692	0.729	0.903

Table 6. Fold prediction results with different portion of real SS as the input.

5 Conclusions and discussion

In this project 2 relationships across 3 protein modalities were learned. The sequence-SS relationship was learned by several SVM models and a fully connected neural network based on protein sequence segments. The best model reached a test accuracy of 0.833 and could do the inference of more than a hundred sequences in less than a second. Based on longer segments, the prediction accuracy would also be higher. For the sequence-structure relationship, the structures were represented by the fold categories. DeepSF models were developed to predict the folds by taking merely the sequence or the combination of the sequence and the SS as the input. With the SS the performance can be improved, and the improvement would be much larger with real SS than the predicted ones, and the DeepSF model is sensitive the purity of the real SS sequences. As a result, to further improve the performance based on the cross modality learning idea, we can try to improve the accuracy SS prediction to make the fake data more close to the real ones. We can also try to combine other modalities like the PSSM, the SA and the contact map with the current input. By developing different prediction models concatenated with the DeepSF, hopefully the model can learn more information and get a better performance.

6 Future Work

Other models can be tried for the SS prediction, such as sequence transformer (Vaswani *et al.*, 2017) and some language translation models in the natural language processing (NLP) domain, as these models may also capture the global information to make the prediction. More experiments can be done to discover the phenomena about the performance of the SS prediction models and the DeepSF, like maybe SS prediction was bad at some certain segments and it may also effect the fold prediction. Due to the limited time and computational resource, only a relatively small dataset was applied for the project, which might the performance variance to higher and the deep model could be overfitting. Therefore larger datasets can be applied in the future, and even unlabeled data could be used to improve the performance.

Acknowledgements

Thank Dr. Yang Shen from Electrical and Computer Engineering Department of Texas A&M University for the instruction and Texas A&M High Performance Research Computing for GPU allocations.

Contribution

This work was accomplished by Shaowen Zhu and instructed by Dr. Yang Shen.

References

Alberts, B. *et al.* (2015). *Essential cell biology*. Garland Science.

Baker, D. and Sali, A. (2001). Protein structure prediction and structural genomics. *Science*, **294**(5540), 93–96.

Bhagwat, M. and Aravind, L. (2007). Psi-blast tutorial. In *Comparative genomics*, pages 177–186. Springer.

Carrillo, H. and Lipman, D. (1988). The multiple sequence alignment problem in biology. *SIAM journal on applied mathematics*, **48**(5), 1073–1082.

Cheng, J. *et al.* (2005). Scratch: a protein structure and structural feature prediction server. *Nucleic acids research*, **33**(suppl_2), W72–W76.

Consortium, U. (2019). Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, **47**(D1), D506–D515.

Di Lena, P. *et al.* (2012). Deep architectures for protein contact map prediction. *Bioinformatics*, **28**(19), 2449–2457.

Greener, J. G. *et al.* (2018). Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, **8**(1), 1–12.

Hou, J. *et al.* (2018a). DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, **34**(8), 1295–1303.

Hou, J. *et al.* (2018b). Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, **34**(8), 1295–1303.

Karimi, M. *et al.* (2019). De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks (gcwgan). *bioRxiv*, page 769919.

O'maille, P. E. *et al.* (2002). Structure-based combinatorial protein engineering (scope). *Journal of molecular biology*, **321**(4), 677–691.

Pabo, C. (1983). Molecular technology: designing proteins and peptides. *Nature*, **301**(5897), 200–200.

Pietal, M. J. *et al.* (2015). Gdfuzz3d: a method for protein 3d structure reconstruction from contact maps, based on a non-euclidean distance function. *Bioinformatics*, **31**(21), 3499–3505.

Street, A. G. and Mayo, S. L. (1999). Computational protein design. *Structure*, **7**(5), R105–R109.

Vassura, M. *et al.* (2008). Reconstruction of 3d structures from protein contact maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, **5**(3), 357–367.

Vaswani, A. *et al.* (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.