A Robust and Explainable Model for Protein-Peptide Binding Site Prediction

Irtesam Mahmud Khan
ID: 0421052039
Department of CSE, BUET
Dhaka, Bangladesh
irtesam.m.khan@gmail.com

***Abstract*—Protein-peptide binding sites are crucial to our understanding of several cellular processes. Due to the lack of experimental data, especially information related to protein structure, this is a tricky problem to conquer. Recently, many machine learning models have been developed to tackle this issue, but have not performed very well without structural information. We propose a deep learning based technique that takes only protein sequence as input and predicts binding sites. We have leveraged pretrained language models and undersampling techniques to make the model more robust. We also proposed an attention based mechanism to increase the explainability of the model.**

***Index Terms*—protein, peptide, binding sites, data imbalance**

I. Introduction

Proteins are large molecules or macromolecules that comprise one or more long chains of amino acid sequence. Peptides are small(< 30) amino acid sequences. Protein-peptide interaction plays a vital role in drug design as they are involved in various cellular processes such as DNA repair, replication, gene expression, and metabolism. In fact, 15-20% of all protein–protein interactions are mediated by small peptides [8].Recently, new functional roles of protein–peptide interaction were described and investigated. These interactions were implicated in human diseases especially in cancers and viral infections. Given the importance of this type of interaction, it is essential to identify peptide binding sites in a protein.

Experimental data, especially solved structure, is scarce for this problem. For this reason, modeling protein-peptide binding sites computationally is very critical for achieving molecular insight into how several cellular processes work. But solving this problem has been far from simple. Although billions of protein sequences are available now, there is a lack of protein structure information available generally. Extracting features from this sequence needs a lot of domain knowledge. Another important obstacle is the class imbalance in the data. Very few regions in the protein chain are actually peptide-binding sites. This is a very big issue if any machine learning model needs to be trained.

For solving the above mentioned issues, we propose a robust machine learning model that tackles the issue of feature extraction and data imbalance. We used ProtBERT [4], a language model pretrained on billions of amino acid sequences. By using pretrained language model for feature extraction we reduce our dependency on domain knowledge. This also makes the model more robust, as we can find features that are not yet discovered by the experts. Next, we used an undersampling technique to solve the data imbalance problem during training. Finally, we trained a deep learning model comprised of both RNN and CNNs. The CNN model achieved accuracy comparable to the state-of-the-art models.

So our contribution to this paper can be stated as follows:

- Using pretrained language model for feature extraction that reduces the dependency on domain knowledge
- Undersampling techniques and class weights to solve imbalanced data
- Using attention mechanism for increasing explainability of the model

II. Motivation

As was discussed in the Introduction section, identifying protein-peptide binding sites remain a critical task for understanding several cellular processes. Roughly 10^4 human proteins contain at least one peptide recognition module(PRM) [2]. Like other protein–biomolecular interactions, peptides bind in the conserved region of the target protein. In addition, peptides use hydrogen bonds to form interactions with their protein partner. The peptide-binding regions in proteins appear to be dominated by large and flatter pockets. Although we have this bit of domain knowledge about peptide binding sites, it remains a challenging task. Because of small peptide size [12], weak binding affinity [3] and peptide flexibility experimental methods find it difficult to identify peptide binding sites. So this remains an open problem. As a further matter, the robustness and explainability of the current models are not up to the mark. Despite all of these methods, none of the models achieve remarkable accuracy(more precisely MCC). Most of the recent methods take into account protein structure information. So we propose a sequence only method that can predict protein-peptide binding sites only from protein sequences.

III. Related Works

Several dry lab methods have been proposed to identify peptide binding sites. Direct docking of peptides onto protein structures can predict binding residues by predicting protein–peptide complex structure. However, docking methods are less feasible for docking typical peptides of lengths between

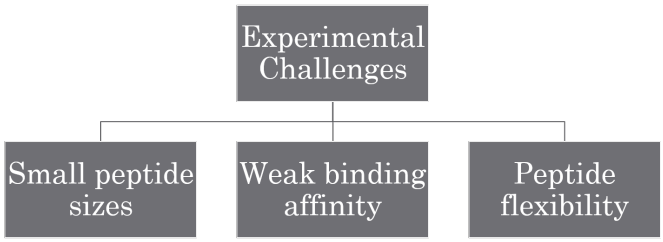

Fig. 1. Experimental Challenges

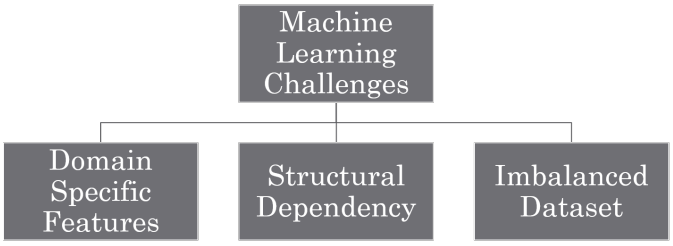

Fig. 2. ML Challenges

5 and 10 residues onto proteins with unknown binding sites because of the large search space for flexible peptide conformation.

GalaxyPepDock [6], on the other hand, builds peptide-binding sites based on known docked structures by structural similarity search. FoldX [9] attempts to infer peptide-binding sites by employing interacting backbone fragment pairs. Due to several limitations of docking based methods different strategies have been developed to predict binding sites. Pepsite employs spatial position specific scoring matrix (PSSM) derived from known protein–peptide complex structures to locate hot-spots on protein surfaces and determine binding sites based on distance constraints. However, the above methods are limited due to their requirement of binding peptide sequences that are not always known. To solve this problem, Peptimap maps and clusters potential binding sites by docking small molecule probes. Because of the limitations of above methods,*Ghazaleh Taherzadeh et al.,* proposed a sequence based method called SPRINT [10]. They also proposed a structure based method in a later work, namely SPRINT-str [11]. Another model PepNN [1] was proposed, which is a deep attention model that leverages transfer learning to compensate the scarcity of structural information. This study uses both only sequence based and combination of sequence and structure based features. However, PepNN needs both protein sequence and peptide sequence to identify the binding sites. This is not an ideal scenario for finding novel peptide binding sites. Because we do not know the peptide sequence beforehand. Kozlovskii et al., [5] propose a 3D convolutional model to predict peptide binding site using protein structure as a 3D image.DELPHI [7] is a sequence based method but it predicts protein-protein interaction which can not exclusively predict protien-peptide binding sites. Most of the previous studies rely on Position Specific Scoring Matrix(PSSM) and Hidden Markov Models(HMM) for feature extraction. But these techniques are computationally expensive. Moreover, these techniques were developed using a lot of domain knowledge. So any chance of extracting features that we do not yet know is not possible. Some recent studies have used language models to extract information from protein sequences. For example, PepNN also used the same language models for feature extractions. Most studies have shown that these language models outperform PSSM/HMM for feature extraction.

IV. Methods

A. Dataset

The dataset used in this study is available publicly. This dataset was released as part of the materials by the authors of SPRINT-str. In total, the Train set contains 1116 no. of protein chains and the Test set contains 125 protein chains. The average length of each protein chain is around 476 residues. But the longest chain contains 2835 amino acid residues. The detailed distribution of Binding and Non-Binding Residues in the dataset can be found in Table I. As we can see, the dataset is pretty imbalanced. The number of negative samples is around 16 times the number of positive samples. Also because of the large size of the sequence high computational resources are needed.

	Binding Residues	Non-Binding Residues	P/N Ratio
Train	14959	251769	0.059
Test	1716	29154	0.0588

TABLE I
DATASET: DISTRIBUTION OF BINDING AND NON-BINDING RESIDUES

There are some other available datasets related to this work. We plan to explore them in the future. For now, all of the studies conducted are on the above mentioned dataset.

B. Feature Extraction

A language model was used to extract features from protein sequences only. ProtTrans is a collection of transformer based models that are trained on protein sequences. These models were trained on 493 billion amino acid residues. We used ProtBert to extract features from the protein sequences. We used an embedding size of 1024. So for each residue, the model generates a 1024-size vector that is used as the input feature. Previously other studies have found that using language models instead of PSSM and HMM for feature generation has two distinct advantages. More often than not, results have improved. But the main advantage is the speed of feature generation. ProtBert can generate features for the whole dataset within 15-20 minutes, whereas PSSM may take hours or even days to generate the similar quantity of features.

We use a sliding window for capturing the neighborhood information. For each residue, we take a window of size 31(15 residues before the target one and 15 residues after it). The features of all of these 31 residues are concatenated together and this is used as the input features for predicting the target residue. The beginning and ending part of the protein chain is padded with zero vectors. In this way, we have also augmented our dataset. Instead of having only 1116 sequences, we now have around 266000 residues as a data point.

Fig. 3. Feature Extraction

C. Model

We wanted to capture both the neighboring information and the long-term dependency among the residues. So we propose an ensemble of a CNN and RNN model to capture both types of information(Fig. 4). For our initial study, only CNN model was implemented. The plan was to create a CNN model with state-of-the-art computer vision techniques. On the other hand, we wanted to create a bidirectional LSTM model. The output of both of these models can be fused together and passed on to a fully connected layer to produce the output of the model. The RNN model unfortunately was not producing desired results. A simple CNN model was built that comprised three convolutional layers. The output of the final layer was passed on to a sigmoid activation function to produce the output.

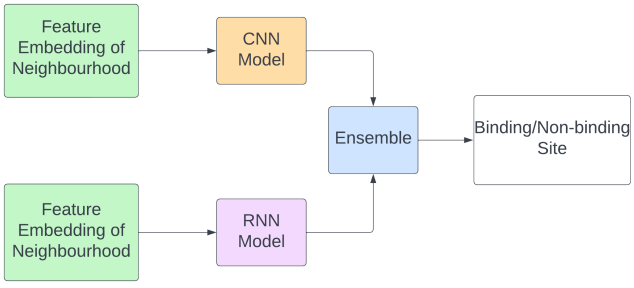

Fig. 4. Model Overview

D. Solving Data Imbalance

One of the main focuses of this study is to solve the data imbalance problem. To solve that, we used undersampling in a different way. If we only perform undersampling, we must lose some information. So we created multiple subsets to reduce the information loss as much as possible. Initially, we created several subsets of the original dataset by taking only 20% of the negative samples and all positive samples. Each subset now has 1:3 positive-to-negative ratio. Now, we trained our model on each of these subsets with a weighted binary cross entropy loss function. Next, all of these separately trained models are used to create an ensemble model that is trained on the whole training dataset. The ensemble model only contained two fully connected layers that takes input from the subset models.

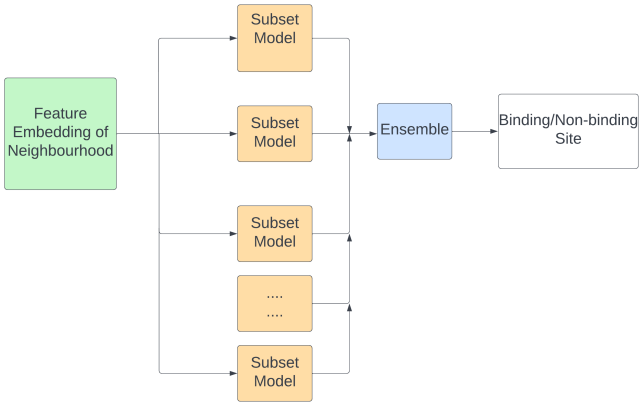

Fig. 5. Undersample and ensemble method

E. Training and Evaluation

We trained our models with different numbers of feature channels and kernel sizes for the CNN model. We performed experiments with and without the undersampling technique discussed in the previous section. Weighted Binary Cross Entropy(WBCE) loss was used for training the model. During training, we used Dropout between convolutional layers, Early Stopping and Learning Rate scheduling to avoid overfitting. We used accuracy and Matthews correlation coefficient (MCC) as our evaluation criteria.

F. Data and Code

We used python language for implementation. PyTorch framework was used for all deep learning based tasks. We performed all our experiments on Google Colab. All the data and code are available in the following Google Drive folder.

V. Experimental Evaluation

We wanted to combine both CNN and RNN. But our implementation of a bidirectional LSTM network performed extremely poorly. So we had to only experiment with CNN based models. We also did not use any structurul information as structure information for most protein is not available as of this moment.

A. Results from Previous Works

Some previous studies are compared in Table II. This comparison was obtained from the authors of PepNN. In this table, we can check how different methods have been performed on the test dataset we are also using(TS125). These

Test dataset	Training dataset size	Model	ROC AUC	MCC
TS125	956	PepNN-Struct	**0.885**	0.39
		PepNN-Seq	0.794	0.259
		BiteNet	0.882	0.435
	640	PepBind	0.793	0.372
	1156	SPRINT-Str	0.78	0.29
	1199	SPRINT-Seq	0.68	0.2
	1004	Visual	0.73	0.17
	–	AlphaFold-Multimer	–	**0.576**
	–	AlphaFold-Gap	–	0.44

TABLE II
COMPARISON AMONG PREVIOUS STUDIES

Undersampling?	Parameters		Validation		Test	
	Kernel Size	Feature Channels	F1	MCC	F1	MCC
No	3	128	0.36	0.36	0.26	0.23
	5	128	0.39	0.38	0.27	0.24
	5	512	0.38	0.38	0.28	0.24
Yes	5	512	0.37	0.36	0.27	0.24
	5	128	0.50	0.43	0.30	0.27

TABLE III
EFFECT OF HYPERPARAMETERS ON ACCURACY AND MCC

models are trained with a different number of sequences. Some are only sequence-based models, others take into account both sequence and structure-based information. We can see that PepNN performs best in terms of ROC AUC and AlphaFold Multimer performs best in terms of MCC. We have previously discussed some limitations of PepNN. AlphaFold-Multimer is a structure prediction tool that takes into account both protein and peptide sequences. But we want to predict novel peptide binding sites i.e., without knowing the peptide sequence beforehand. Except for AlphaFold-Multimer all of the other methods perform considerably poorly in terms of MCC.

B. Analysis on Extracted Features

The feature embedding size is pretty large. So we performed PCA to visualize the data space. An example plot of a protein is shown in Figure 6. The Yellow colored ones are the binding residues. Apparently, only feature embeddings are not enough to differentiate between Binding and Non-Binding residues. But a very important thing was observed. Only the top 20 dimensions after the PCA analysis can capture 98% of the variance. This points to a very good possibility of reducing the feature dimensions without degrading the results.

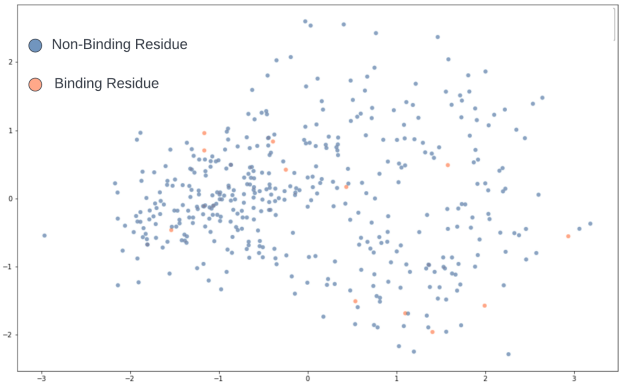

Fig. 6. PCA of a single protein sequence

C. Results

Results from our study are presented in Table III. This table describes the results on different experimental settings. We can see that the best performing model is the model with *kernel size = 5* and *feature channels = 128*. We can also see the effect of our undersampling method on the results. Without undersampling, using a weighted BCELoss function does not perform as well. So our technique to solve the imbalanced data problem has worked to some extent.

Although the results presented here may seem poor, if it is compared with other sequence-only models mentioned in the previous section, the results are comparable. Even the latest model, PepNN-seq has an MCC Score of 0.26, which is similar to our model. This observation also points to the fact that only sequence related information may not be enough to solve this problem. That is why we can see a significant jump in accuracy and MCC, when structures are used.

As our LSTM model was not performing well, we could not implement attention mechanism to increase the explainability of the model.

VI. CONCLUSION

In this study, we proposed a deep learning based robust and explainable model for predicting protein-peptide binding sites. So far the results from only sequences have been promising. One way forward can be using a structure prediction model like AlphaFold to predict structure from the sequences and then use that information to create a basic Graph Neural Network(GNN). This will preserve the sequence-only nature of our input but can learn from the structure related information. We have trained and tested it on only one dataset. Three other similar datasets can be obtained publicly. We plan to test our model on them in the future.

REFERENCES

[1] Osama Abdin, Satra Nim, Han Wen, and Philip M. Kim. PepNN: a deep attention model for the identification of peptide binding sites. *Communications Biology*, 5(1), May 2022.

[2] Joseph M. Cunningham, Grigoriy Koytiger, Peter K. Sorger, and Mohammed AlQuraishi. Biophysical prediction of protein–peptide interactions and signaling networks using machine learning. *Nature Methods*, 17(2):175–183, January 2020.

[3] H. Jane Dyson and Peter E. Wright. Intrinsically unstructured proteins and their functions. *Nature Reviews Molecular Cell Biology*, 6(3):197–208, March 2005.

[4] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Wang Yu, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[5] Igor Kozlovskii and Petr Popov. Protein–peptide binding site detection using 3d convolutional neural networks. *Journal of Chemical Information and Modeling*, 61(8):3814–3823, July 2021.

[6] Hasup Lee, Lim Heo, Myeong Sup Lee, and Chaok Seok. GalaxyPepDock: a protein–peptide docking tool based on interaction similarity and energy optimization. *Nucleic Acids Research*, 43(W1):W431–W435, May 2015.

[7] Yiwei Li, G Brian Golding, and Lucian Ilie. DELPHI: accurate deep ensemble model for protein interaction sites prediction. *Bioinformatics*, 37(7):896–904, August 2020.

[8] Victor Neduva, Rune Linding, Isabelle Su-Angrand, Alexander Stark, Federico de Masi, Toby J Gibson, Joe Lewis, Luis Serrano, and Robert B Russell. Systematic discovery of new recognition peptides mediating protein interaction networks. *PLoS Biology*, 3(12):e405, November 2005.

[9] J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano. The FoldX web server: an online force field. *Nucleic Acids Research*, 33(Web Server):W382–W388, July 2005.

[10] Ghazaleh Taherzadeh, Yuedong Yang, Tuo Zhang, Alan Wee-Chung Liew, and Yaoqi Zhou. Sequence-based prediction of protein–peptide binding sites using support vector machine. *Journal of Computational Chemistry*, 37(13):1223–1229, February 2016.

[11] Ghazaleh Taherzadeh, Yaoqi Zhou, Alan Wee-Chung Liew, and Yuedong Yang. Structure-based prediction of protein– peptide binding regions using random forest. *Bioinformatics*, 34(3):477–484, September 2017.

[12] Patrick Vlieghe, Vincent Lisowski, Jean Martinez, and Michel Khrestchatisky. Synthetic therapeutic peptides: science and market. *Drug Discovery Today*, 15(1-2):40–56, January 2010.