ProSBiRD - Protein Sequence-based Binding Residue Detection using Deep Resnets

Vineeth R. Chelur and U. Deva Priyakumar*

Center for Computational Natural Sciences & Bioinformatics, IIIT-H, Hyderabad

E-mail: deva@iiit.ac.in

Abstract

Motivation: Protein-drug interactions play important roles in many biological processes. The prediction of the active binding site of a protein helps discover such interactions. The tertiary structure of a protein determines the binding sites available to the drug molecule. However, the methods for structure determination are labour-intensive and time-consuming. Hence it becomes important to make predictions using the sequence alone. Deep Learning has been used in a variety of biochemical tasks and has been hugely successful. In this paper, a residual neural network is implemented to predict a protein's most active binding site using features extracted from just the sequence.

Results: The model achieves an MCC of 0.52 and an accuracy of 92.2% on the validation sets averaging across 10-folds. On the test set, the prediction of 10 models aggregate gives an MCC of 0.55 and an accuracy of 94.1%.

Implementation: https://github.com/crvineeth97/protein-binding-site-prediction

Introduction

The rapid speed of sequencing attained with modern DNA sequencing technology has been instrumental in the sequencing of complete DNA sequences, which leads to faster indirect sequencing of proteins. Although there have been improvements in the determination of the three-dimensional protein structure by techniques such as X-ray crystallography and NMR Spectroscopy, the gap between the number of known sequences (333,201,385 as of July 2020) and the number of known structures (154,706 as of July 2020) is increasing rapidly. Proteins perform a vast array of functions within organisms, and the tertiary structure of a protein can provide important clues about these functions.

In the process of drug design, new medications are found based on the knowledge of a biological target such as a protein. The identification of the potential active binding site of a protein is an essential step in drug design. Predicting the binding site of a protein, based on sequence alone, becomes critical when the three-dimensional structure of the protein is not available.

Ligand binding site prediction methods can be clustered into four groups: 3D structure-based, Template similarity-based, Traditional machine learning-based and Deep learning-based prediction methods.

3D structure-based methods are based on the concept that most small ligand bindings occur in hollows or cavities on protein surfaces because large interfaces have high affinity. Hence, these methods locate the binding site by searching for spatial geometry or energy features (by placing probes) in protein structures. SITEHOUND[1] uses a carbon and phosphate probe inside a grid that covers the entire protein. The grid points with higher interaction energies are clustered and determine the binding residues. CURPOCKET[2] is a recent spatial geometric measurement method that computes the curvature distribution of the protein surface and identifies the clusters of concave regions. While 3D structure-based methods have been widely used, they depend on the state of the provided protein 3D structure.

Template similarity-based methods do not consider proteins as independent entities but as evolved from structurally, functionally or sequentially similar proteins. S-SITE[3] employs the Needleman-Wunsch algorithm to align the query protein to each of the proteins in the BioLip[4] database and selects similar sequences according to the alignment. The binding residues of the aligned proteins which occur more frequently are selected for predicting the binding site.

Traditional machine learning-based methods promote the application of artificial intelligence in Biochemistry. 3D structure-based and template similarity-based methods complement each other well. Machine learning is used to integrate the information of both methods and apply mathematical functions to improve prediction accuracy. P2RANK[5] uses a random forest algorithm to predict ligandibility scores across the entire protein surface. The points with high scores are then clustered into a single binding pocket.

Deep Learning is a subfield of machine learning based on artificial neural networks with feature learning. When a deep learning model is fed large amounts of data, it can automatically discover the representations needed for feature detection or classification. Deep learning has been hugely successful in all fields of Natural Sciences such as binding affinity predictions, protein contact map predictions, and protein-structure predictions.

Deep learning-based methods like DeepSite,[6] and Kalasanty[7] model binding site prediction as an image processing problem. They voxelize the protein 3D structure into small grids and calculate specific properties of each grid. These values are then used to train a deep convolutional neural network which predicts whether a grid belongs to a binding site.

DeepCSeqSite[8] is a template-based method that uses seven characteristics (position-specific scoring matrix, relative solvent accessibility, secondary structure, dihedral angle, conservation scores, residue type and positional embeddings) of each residue to create a feature map, which is then used as an input to a convolutional neural network.

In this paper, a deep residual neural network is trained to make the binary prediction of whether an amino acid residue of the sequence belongs to the primary binding site or not.

Dataset

For the training and validation of the model, the sc-PDB[9] dataset (v 2017) is used. The database consists of druggable binding sites of the Protein Data Bank along with prepared protein structures. Thus each sample in the dataset contains the three-dimensional structure of one ligand, one protein, and one site, all stored in mol2 format.

Typically, the complete structure of a protein is unavailable due to missing residues, and hence the full sequence of the protein is obtained from RCSB[10] website*. A one-to-one mapping of the amino acids in the sequence to the one in the protein mol2 file is required, to know which amino acid belongs to a binding site. This mapping is done by first extracting the protein sequence from the 3D structure. Next, the Needleman-Wunsch dynamic programming algorithm is used to align the sequence extracted from the structure file to the downloaded RCSB sequence. This algorithm is implemented using a modified version of Zhanglab's NW-Align program.[11] The protein structure file is then reindexed, based on this alignment, to match the indexing of the RCSB sequence. This way, the specific binding residues can be labelled in the RCSB sequence.

The training set consists of 17,594 PDB structures with 28,959 sequences (9519 unique sequences), originating from 1240 organisms, the most abundant being humans(28.26%). The dataset was diverse and contained proteins from 1996 different PFAM families and 856 PFAM clans.

Table 1: Summary of the datasets

	N_{prot}	N_{br}	N_{nbr}	$P_{br}(\%)$
Train	15,860	589,329	8,725,043	6.33
Test	2,464	86,230	1,345,646	6.02

N_{prot} - Number of proteins
N_{br} - Total number of binding residues
N_{nbr} - Total number of non-binding residues
$P_{br}(\%)$ - Percentage of binding residues

*Some PDBs in the dataset were obsoleted, and hence the sequences were manually tracked on RCSB, and the corresponding sequences were used. List of obsoleted PDBs is provided in Supporting information

This data is split into 10-folds (each containing 1586 structures), based on Uniprot ID, exactly like how Stepniewska-Dziubinska et al. did in their paper.[7] This split ensures that there is no data leakage between the validation and training set by putting all structures of a single protein in the same fold.

The test set is constructed using all PDBs from 2018 onwards, till 28th February 2020. All PDBs available during this period and having at least one ligand are considered. These are then run through the pdbconv tool from the IChem Toolkit[12] to generate a dataset with the same filters and site selection as the sc-PDB training set.

The test set consists of 2,274 PDB structures with 3,434 sequences (1889 unique sequences), originating from 548 organisms, the most abundant being human(23.76%). The test set contained proteins from 882 PFAM families and 452 PFAM clans.

Methods

MSA Generation

Collections of multiple homologous sequences (called Multiple Sequence Alignments or MSAs) can provide critical information for the modelling of the structure and function of unknown proteins. DeepMSA[13] is an open-source method for sensitive MSA construction, which has homologous sequences and alignments created from multiple sources of databases through complementary hidden Markov model algorithms.

The search is done in 2 stages. In stage 1, the query sequence is searched against the UniClust30[14] database using HHBlits from HH-suite[15] (v2.0.16). If the number of effective sequences is < 128, Stage 2 is performed where the query sequence is searched against the Uniref50[16] database using JackHMMER from HMMER[17] (v3.1b2). Full-length sequences are extracted from the JackHMMER raw hits and converted into a custom HHBlits format database. HHBlits is applied to jump-start the search from Stage 1 sequence MSA against this custom database.

Features

There are 9519 unique protein sequences in the sc-PDB dataset and 1889 unique protein sequences in the test set. The MSAs are generated for these sequences using the method described above and stored in PSICOV[18] .aln format.

One-hot encoding and Positional embeddings are extracted from the sequence alone. Position Specific Scoring Matrix, Information Content, Secondary Structure and Solvent Accessibility are extracted from the generated MSAs.

One-hot encoding and Positional embeddings

There are 20 standard amino acids, and the non-standard amino acids are all labelled as an additional dummy residue, X. This labelling leads to 21 amino acids in our vocabulary. The 20 standard amino acids are used in alphabetical order, numbered from 1 to 20, and the dummy residue X is given the position 0. The one-hot encoding (OHE) of an amino acid will be a vector of zeroes of length 21, where the position of the amino acid in the vocabulary is marked with a one. OHE is used to help the model to differentiate between the different types of amino acids

Positional Embeddings (PE) carry information about the absolute position of the amino acids in the sequence. A simple method of embedding is used where the position of the i^{th} amino acid is represented by $PE_i = \frac{i}{L}$, where L is the length of the protein sequence.

Position Specific Scoring Matrix and Information Content

Position Specific Scoring Matrix (PSSM) is a commonly used representation of patterns in biological sequences. PSSMs are derived from MSAs using Easel[19] and Heinikoff position-based weights so that similar sequences collectively contributed less to PSSM probabilities than diverse sequences.

The information content (IC) of a PSSM gives an idea about how different the PSSM is from a uniform distribution. IC is also derived using Easel.

Secondary Structure and Solvent Accessibility

The secondary structure is defined by the pattern of hydrogen bonds formed between the amino hydrogen and carboxyl oxygen atoms in the peptide backbone. It gives an idea of the three-dimensional structure of the protein. The secondary structural elements are alpha helices, beta sheets and turns. PSIPRED (v4.0)[20] is used to predict the probability of each state of the 3-state secondary structure (SS3) for every amino acid in the sequence.

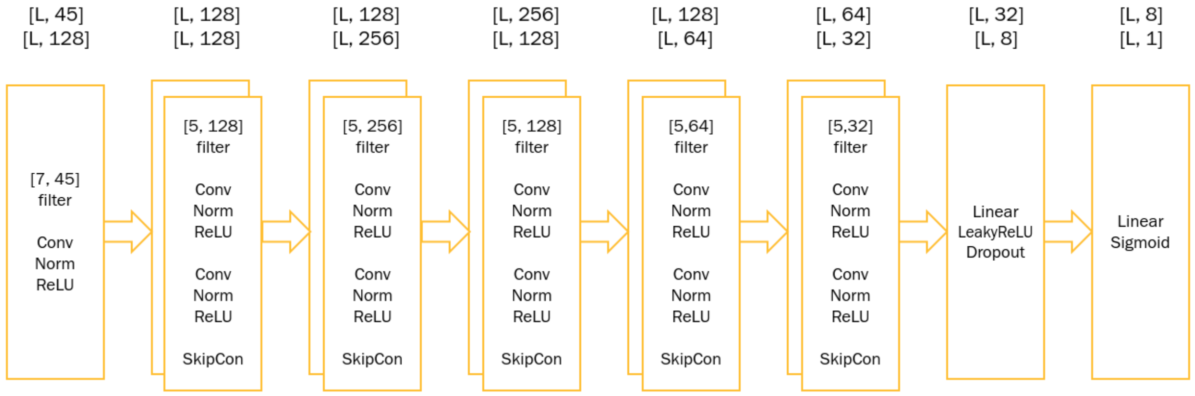
The solvent-accessible surface area is the surface area of a biomolecule that is accessible to a solvent. SOLVPRED from MetaPSICOV 2.0[21] is used to predict the relative solvent accessibility (RSA) of every amino acid in the sequence. RSA can be calculated as $RSA = ASA/MaxASA$, where ASA is the solvent-accessible surface area, and MaxASA is the maximum possible solvent accessible surface area for the amino acid residue.

Deep Learning Model

Residual neural networks[22] are immensely popular in image recognition and have been gaining traction in the field of computational Natural Sciences as well. Deep neural networks are difficult to train, but with the introduction of skip connections (shortcuts to jump over some layers) in Convolutional Neural Networks, the vanishing gradient problem is avoided. Convolutional Neural Networks with skip connections are known as Residual neural networks or ResNets. ResNets use representation learning to extract the most important features for classification. They can also model long-range interactions very well and hence have been very successful in the field of Computational Natural Sciences. The architecture of the deep Residual Neural Network is shown in Figure 1.

Each sample protein in the dataset consists of one or more protein sequences. Let the length of the sequences be $l_1, ..., l_n$. Features are generated for each sequence in the protein (ordered by chain ID in PDB), leading to multiple vectors of shape $[l_i, 45]$ for the i^{th} sequence. These generated features are combined through simple concatenation, giving a final feature vector of shape $[L, 45]$ as input to the model, where $L = l_1 + ... + l_n$.

Figure 1: Architecture of the deep learning model



The feature vector is passed through the first level which consists of a 1D convolutional layer with 128 filters, each filter size being 7, a batch normalization layer and the ReLU (Rectified Linear Unit) activation function. The input is padded with zeroes to ensure that the length of the output vector remains the same. The filters of the convolution layer stride across the length of the protein, considering the features of the three amino acids before, the three amino acids after and the current amino acid (totalling 7). This stride along the input allows for the extraction of the required information of the current amino acid based on the features of nearby amino acids.

The next few levels consist of 2 blocks called BasicBlocks. A BasicBlock consists of a 1D convolutional layer, a batch normalization layer, a ReLU activation function, a second 1D convolutional layer, a second batch normalization layer, and a final ReLU activation. The skip connection is made after the final ReLU activation, where the initial input to the BasicBlock is added to the output of the final ReLU activation. Usually, the input and output size of the first BasicBlock do not match, and hence there is an up/down-sampling layer that ensures that the input has the same shape as that of the output. The output from the previous level passes through the first BasicBlock, which has an up/down-sampling layer and then goes through the second BasicBlock. In the proposed architecture, the number of filters used at each level goes from 128 → 256 → 128 → 64 → 32, with the filter size being 5 for every convolution.

The last two levels contain a simple, linear, fully-connected artificial neural network. The last but one level has a LeakyReLU activation function along with a dropout as well. The last level has a Sigmoid function at the end to ensure that the output of the model is between $[0, 1]$. The final output of the model is a vector of size L (length of the protein), denoting the probabilities of a residue being a part of the binding site.

Loss Function

For model training, the loss function is given by $L(\hat{y}, y) = -(\alpha \hat{y} \log(y) + (1 - \hat{y}) \log(1 - y))$, where $\hat{y}$ is the vector of true labels of whether an amino acid belongs to the binding site or not, y is the model output of probabilities of a residue belonging to a binding site, α is the weight that is assigned to the rare class.

The main problem in this classification task is the substantial imbalance in the two classes of binding and non-binding residues. As shown in table 1, the percentage of binding residues is only around 6%. Hence, α is used to penalize the model more heavily if it incorrectly predicts binding residues. α is calculated on the fly for every batch of inputs as $\alpha = \frac{n_{nbr}}{n_{br}}$, where n_{nbr} is the total number of non-binding residues in the batch and n_{br} is the total number of binding residues in the batch.

Implementation

The model is implemented using PyTorch Lightning,[23] which is a wrapper on the popular open-source deep learning library, PyTorch.[24] The implementation can be found at https://github.com/crvineeth97/protein-binding-site-prediction.

Results

The sc-PDB dataset was split into ten folds, and ten models with the same architecture were trained. One fold formed the validation set, and the remaining folds formed the training set for each of the models. The validation results are provided in table 2, along with the confusion matrix in figure 2

For testing, the ten trained models are run on the test set. For consensus, an amino acid belongs to the binding site if five or more models predict the same. The test results are also provided in table 2, along with the confusion matrix in figure 3

Evaluation Metrics

Confusion Matrix

A confusion matrix is a table that allows for the visualization of the performance of a supervised learning algorithm. In the case of binary classification of a residue as a binding residue (BR) or non-binding residue (NBR), the following terminologies can be defined.

- True Positive (TP): Number of BRs predicted correctly as BRs.
- True Negative (TN): Number of NBRs predicted correctly as NBRs.
- False Positive (FP): Number of NBRs predicted incorrectly as BRs.
- False Negative (FN): Number of BRs predicted incorrectly as NBRs.

The following metrics can be derived from the confusion matrix

Accuracy: $ACC = \frac{TP+TN}{TP+TN+FP+FN}$

Precision: $PPV = \frac{TP}{TP+FP}$

Recall: $TPR = \frac{TP}{TP+FN}$

F1 score: $F_1 = \frac{2TP}{2TP+FP+FN}$

Matthews Correlation Coefficient: $MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$

Table 2: Validation results of all 10 trained models and test results

Dataset	ACC(%)	PPV(%)	TPR(%)	F1(%)	MCC(%)
Fold 1	92.58	48.64	70.65	57.62	54.83
Fold 2	92.18	46.80	67.29	55.20	52.08
Fold 3	92.94	44.85	69.37	54.48	52.27
Fold 4	91.28	39.62	65.73	49.44	46.72
Fold 5	91.74	46.33	73.11	56.72	54.07
Fold 6	92.19	47.16	69.70	56.25	53.34
Fold 7	91.90	45.45	69.55	54.98	52.13
Fold 8	92.52	47.58	68.16	56.04	53.10
Fold 9	92.06	41.86	69.69	52.31	50.14
Fold 10	92.08	44.54	68.88	54.10	51.41
Test	94.05	50.46	67.45	57.73	55.27

Figure 2: Sum of confusion matrices of the 10 models on their corresponding validation set

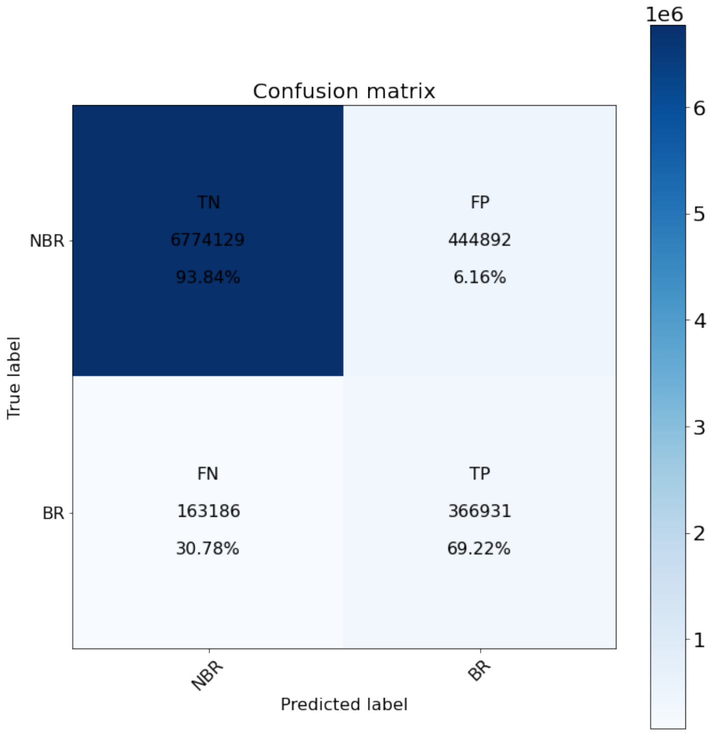
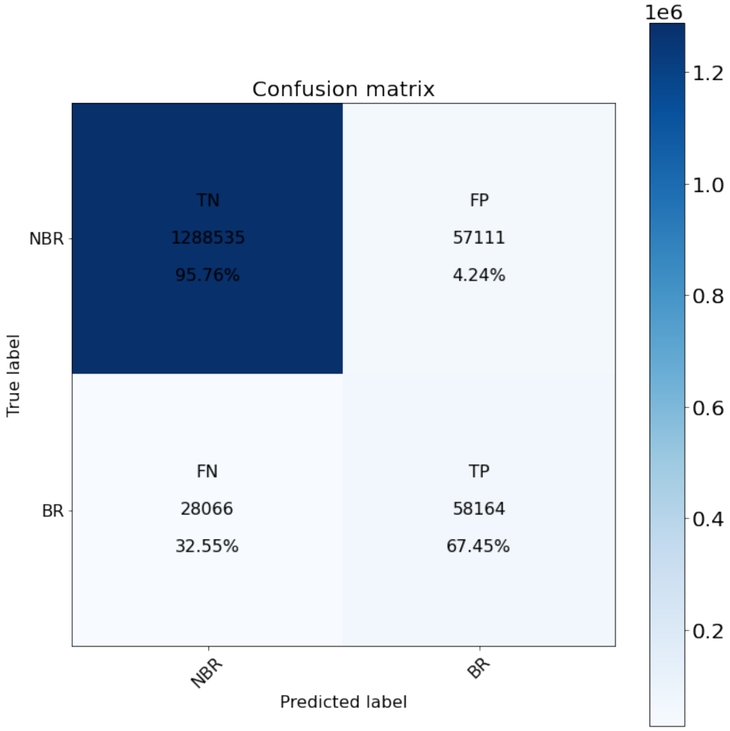


Figure 3: Confusion matrix on the test set after averaging the predictions of the 10 models



DCC

DCC is the distance between the centre of the predicted binding pocket and the centre of the actual binding pocket. It is commonly used for evaluating 3D-structure based models. The success rate of DCC is defined as the fraction of predictions below a given threshold. Predicted pockets with DCC below 4Å are considered to be correctly located. The model predictions were mapped back to the available 3D structures of proteins for the calculation of DCC.

Figure 4 denotes the cross-validation results. The deep learning model is the same across all ten splits of training and validation datasets. The success rate of the models varies based on the fold that is used for validation. It ranges from 33% to 49% success rate when the DCC threshold is less than 4Å. Figure 5 denotes the test result. The predictions have a 40% success rate when the DCC threshold is less than 4Å.

Figure 4: Success rate plot for various DCC thresholds of the 10 models on their corresponding validation set

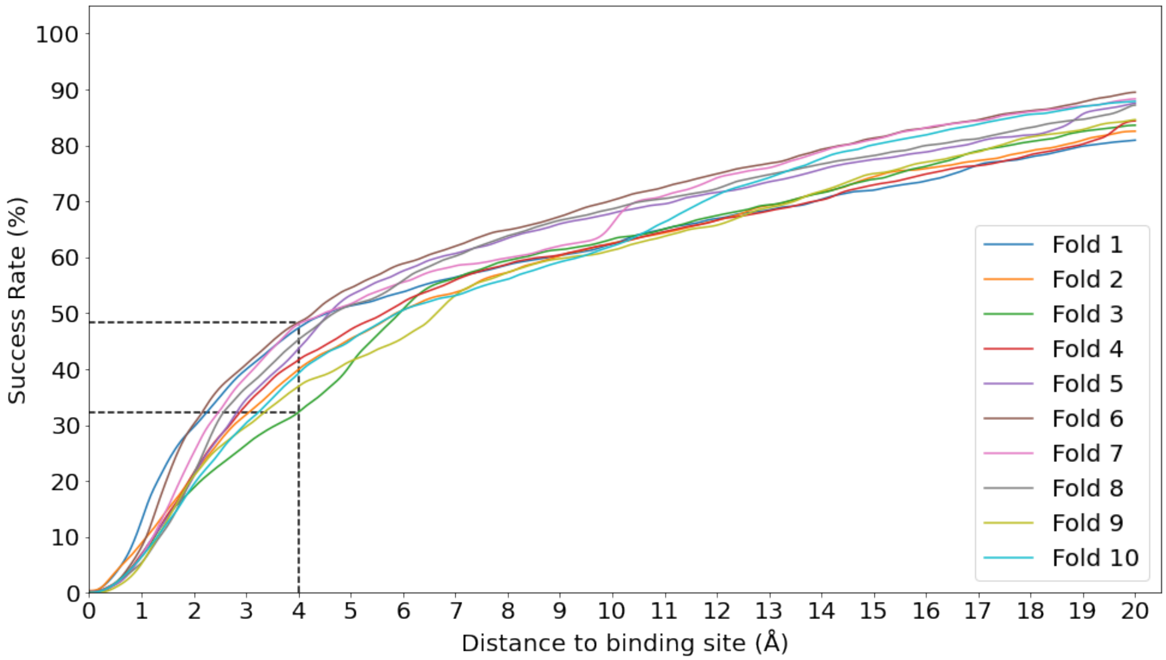
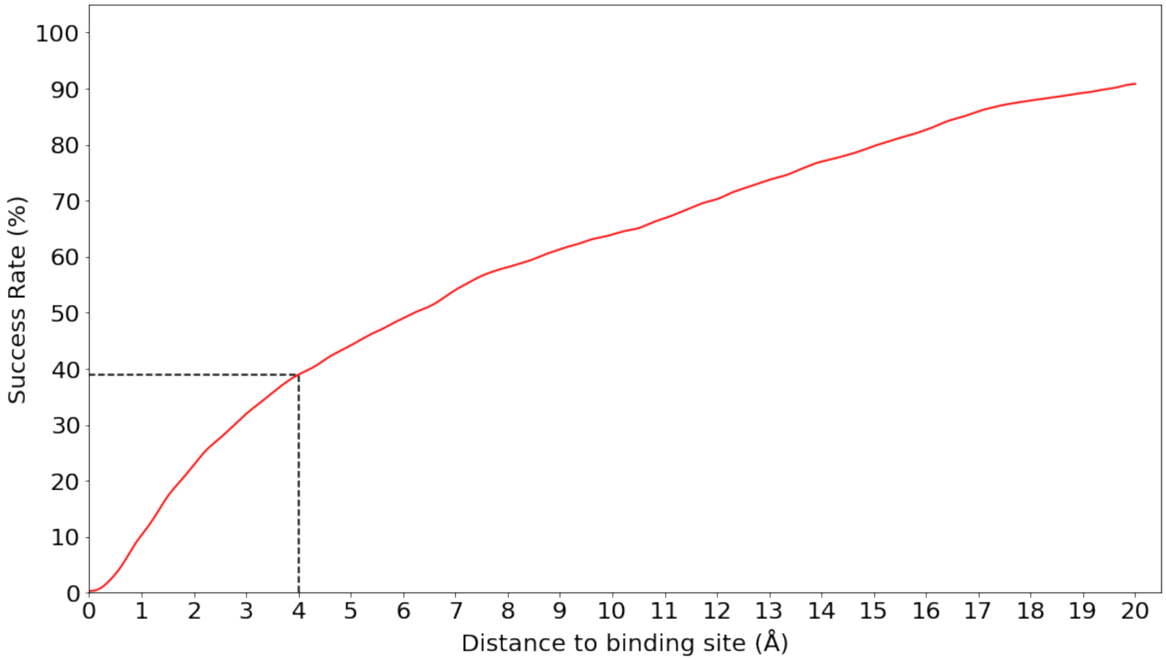


Figure 5: Success rate plot for various DCC thresholds on the test set after averaging the predictions of the 10 models



Discussion

MCC

The Matthew's Correlation varies from $[-1, +1]$, with $+1$ representing a perfect prediction, 0 representing no better than a random prediction and -1 representing total disagreement between the prediction and the observation. The MCC on the test set seems to be ranging from -0.11 to +1, which may seem surprising at first but shows the effectiveness of the model.

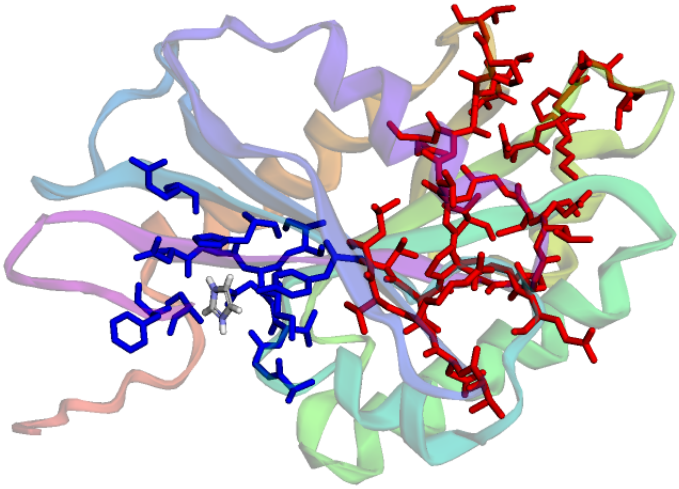
Case Studies

The aggregated predictions of the models on the test set were mapped back to the three-dimensional structure of the protein-ligand complex to see how good the predictions are. In the following examples, the colour red indicates an incorrect prediction of the amino acid as a binding residue, blue indicates an amino acid that is a binding residue but was not predicted as binding by the model and green indicates an amino acid that was correctly predicted as binding.

5YMX

In figure 6, it looks like the model is mispredicting everything, but, it is predicting another binding site of the protein! The sc-PDB dataset was generated through a series of filters, and the residues surrounding the most buried ligand was selected to be the most ligandable binding site. This selection, unfortunately, is a flaw of the dataset and the method used for predictions. There is no right way to cover cases like these where the model needs to be penalized less when it predicts a binding site that is not the most ligandable binding site. Hence, the evaluation metrics used will generally give an abysmal score for such cases.

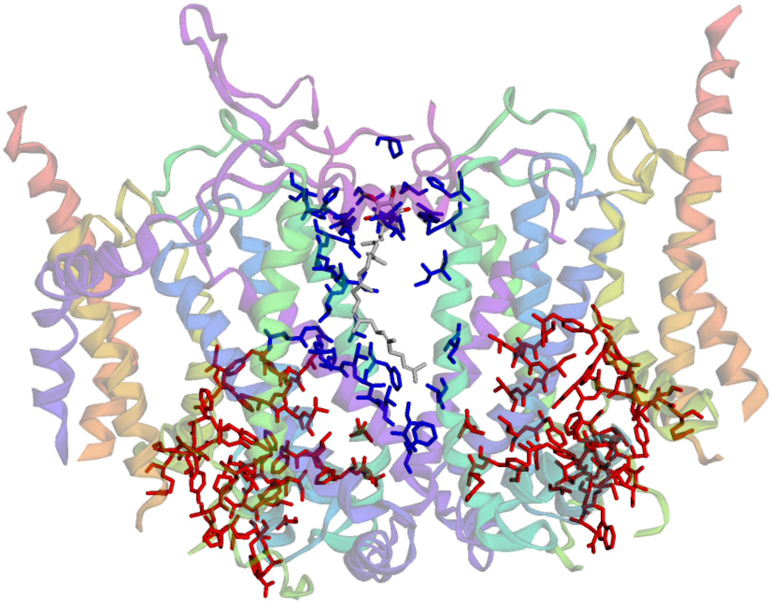
Figure 6: 5YMX



6HU9

Figure 7 shows an example where the model predicts individual binding sites of 2 proteins with the same sequence, but, it finds it difficult to predict the binding site created due to the interaction between the 2 proteins.

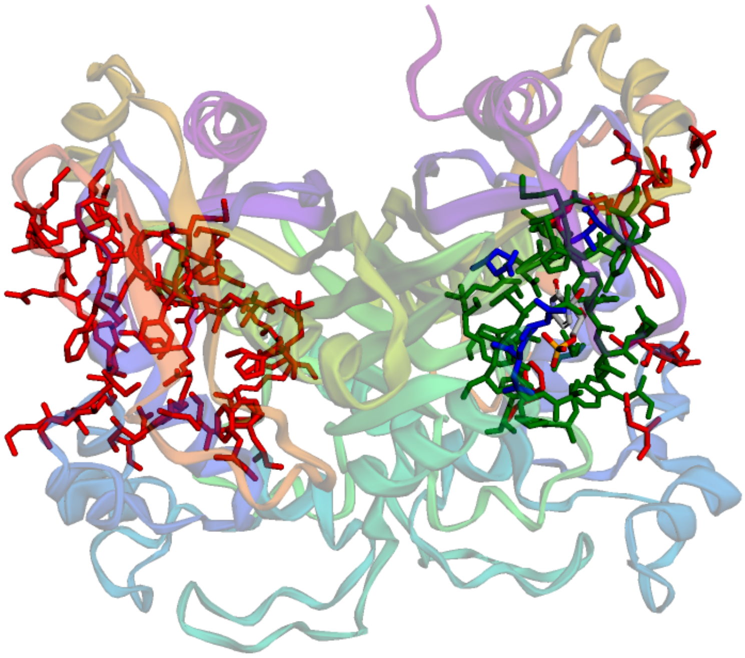
Figure 7: 6HU9



6PF6

Figure 8 shows an example where the model predicts the binding site with a good accuracy but since the two chains of the protein have the same sequence, it predicts the binding site of the other chain as well. Again, since scPDB selects only one binding site, the metrics do not do justice to these type of predictions.

Figure 8: 6PF6



DeepCSeqSite Comparison

All the available methods for predicting binding site of a protein based on sequence, predict the site for only a specific set of ligands, while our model predicts the most ligandable binding site irrespective of the ligand. Hence, there is no available method of comparison for the sequence-based prediction of binding residues on the scPDB dataset.

As a simple way to test the effectiveness of our model against DeepCSeqSite's model, we followed two approaches:

1. Run the trained DeepCSeqSite model on our test set
2. Use DeepCSeqSite model architecture, train on our dataset, and then test

Both approaches failed to provide good results, with approach 1 providing an MCC score of 0.05 and approach 2 providing an MCC score of 0.1

Areas of Improvement

- The training data was split into multiple folds by UniprotIDs, which means that similar structures went into the same fold. However, this does not completely guarantee that proteins with similar sequences go into the same fold as well.
- A variety of deep learning models such as Complementary GANs, stacked BiLSTMs, and TAPE protein embeddings, were used for training. None of them provided any significant improvement over Vanilla ResNets.

Acknowledgement

The author thanks his mentor, Dr Deva Priyakumar, for presenting the opportunity to work on the project. The author also offers thanks to his family for providing their support, especially during the pandemic. The author specially thanks Yashaswi Pathak for being a fruitful part of the project discussions.

References

(1) Hernandez, M.; Ghersi, D.; Sanchez, R. SITEHOUND-web: a server for ligand binding site identification in protein structures. *Nucleic acids research* **2009**, *37*, W413–W416.

(2) Liu, Y.; Grimm, M.; Dai, W.-t.; Hou, M.-c.; Xiao, Z.-X.; Cao, Y. CB-Dock: a web server for cavity detection-guided protein–ligand blind docking. *Acta Pharmacologica Sinica* **2020**, *41*, 138–144.

(3) Yang, J.; Roy, A.; Zhang, Y. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics* **2013**, *29*, 2588–2595.

(4) Yang, J.; Roy, A.; Zhang, Y. BioLiP: a semi-manually curated database for biologically relevant ligand–protein interactions. *Nucleic acids research* **2012**, *41*, D1096–D1103.

(5) Krivák, R.; Hoksza, D. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of cheminformatics* **2018**, *10*, 39.

(6) Jiménez, J.; Doerr, S.; Martínez-Rosell, G.; Rose, A. S.; De Fabritiis, G. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* **2017**, *33*, 3036–3042.

(7) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Improving detection of protein-ligand binding sites with 3D segmentation. *Scientific reports* **2020**, *10*, 1–9.

(8) Cui, Y.; Dong, Q.; Hong, D.; Wang, X. Predicting protein-ligand binding residues with deep convolutional neural networks. *BMC bioinformatics* **2019**, *20*, 93.

(9) Desaphy, J.; Bret, G.; Rognan, D.; Kellenberger, E. sc-PDB: a 3D-database of ligandable binding sites—10 years on. *Nucleic acids research* **2015**, *43*, D399–D404.

(10) Burley, S. K.; Berman, H. M.; Bhikadiya, C.; Bi, C.; Chen, L.; Di Costanzo, L.; Christie, C.; Dalenberg, K.; Duarte, J. M.; Dutta, S., et al. RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic acids research* **2019**, *47*, D464–D474.

(11) Zhang, Y. `http://zhanglab.ccmb.med.umich.edu/NW-align`.

(12) Da Silva, F.; Desaphy, J.; Rognan, D. IChem: A Versatile Toolkit for Detecting, Comparing, and Predicting Protein–Ligand Interactions. *ChemMedChem* **2018**, *13*, 507–510.

(13) Zhang, C.; Zheng, W.; Mortuza, S.; Li, Y.; Zhang, Y. DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics* **2020**, *36*, 2105–2112.

(14) Mirdita, M.; von den Driesch, L.; Galiez, C.; Martin, M. J.; Söding, J.; Steinegger, M. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic acids research* **2017**, *45*, D170–D176.

(15) Remmert, M.; Biegert, A.; Hauser, A.; Söding, J. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods* **2012**, *9*, 173–175.

(16) Suzek, B. E.; Wang, Y.; Huang, H.; McGarvey, P. B.; Wu, C. H.; Consortium, U. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **2015**, *31*, 926–932.

(17) Johnson, L. S.; Eddy, S. R.; Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC bioinformatics* **2010**, *11*, 431.

(18) Jones, D. T.; Buchan, D. W.; Cozzetto, D.; Pontil, M. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics* **2012**, *28*, 184–190.

(19) Potter, S. C.; Luciani, A.; Eddy, S. R.; Park, Y.; Lopez, R.; Finn, R. D. HMMER web server: 2018 update. *Nucleic acids research* **2018**, *46*, W200–W204.

(20) Jones, D. T. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology* **1999**, *292*, 195–202.

(21) Jones, D. T.; Singh, T.; Kosciolek, T.; Tetchner, S. MetaPSICOV: combining coevolution methods for accurate prediction of contacts and long range hydrogen bonding in proteins. *Bioinformatics* **2015**, *31*, 999–1006.

(22) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. 2016; pp 770–778.

(23) Falcon, W. PyTorch Lightning. *GitHub. Note: https://github.com/PyTorchLightning/pytorch-lightning Cited by* **2019**, *3*.

(24) Paszke, A. et al. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc., 2019; pp 8024–8035.