DEEP-POCKET

Deep-learning based predictor for ligand sites

Allal el Hommad Zarkik, Daniel Pérez Artiles, Javier Herranz Delcerro

MSc Bioinformatics for Health Sciences

University of Pompeu Fabra

Theory background and scientific explanation

Proteins participate in various essential processes in vivo via interactions with other molecules. Identifying the residues that participate in these interactions provides biological insights of protein function and also has great significance for drug discoveries. Therefore, predicting protein binding sites has long been an area of interest in the fields of bioinformatics and computer aided drug discovery.

Since the 1990s, numerous computational methods have constantly been proposed to identify protein binding sites. Earlier methods relied on manually crafted spatial geometry or energy features to identify large hollows or cavities within protein structures where interactions often occur. However, since protein binding sites are also influenced by other bio-physicochemical properties specific to different ligands and protein types, these methods often result in high false positive rates (Xia et al., 2024). In recent years, databases such as the Protein Data Bank (PDB), BioLip, and DrugBank have enabled large-scale dataset-based methods, including template-based, traditional machine learning-based, and deep learning-based methods.

Deep learning is a complex machine learning technique that simulates the learning mechanism of the human brain by building and simulating the neural networks in the human brain and uses this mechanism to interpret data (Zhao et al., 2020). Deep learning has enabled us to make sense of massive amounts of complex data sets where the ability of the model to identify intrinsic patterns in a complex plane of data is the strength of the approach. We chose to use this approach because we found it to be the most popular in the bibliography in recent years, although we used a feedforward neural network instead of more complex implementations like a convolutional neural network for simplicity.

The architecture of a feedforward neural network consists of three types of layers: the input layer, hidden layers, and the output layer (DeepAI, 2020). Each layer is made up of units known as neurons, and the layers are interconnected by weights.

- **Input Layer:** This layer consists of neurons that receive inputs and pass them on to the next layer. The number of neurons in the input layer is determined by the dimensions of the input data.
- **Hidden Layers:** These layers are not exposed to the input or output and can be considered as the computational engine of the neural network. Each hidden layer's neurons take the weighted sum of the outputs from the previous layer, apply an activation function, and pass the result to the next layer. The network can have zero or more hidden layers.
- **Output Layer:** The final layer that produces the output for the given inputs. The number of neurons in the output layer depends on the number of possible outputs the network is designed to produce.

Each neuron in one layer is connected to every neuron in the next layer, making this a fully connected network. The strength of the connection between neurons is represented by weights, and learning in a neural network involves updating these weights based on the error of the output.

Although deep learning approaches in binding site prediction have been used and applied in the past 5 years, there are still some problems and deficiencies to this method. A key problem is that deep learning algorithms often require extremely high training costs (expensive computing resources,

huge training sets, etc.) compared with traditional machine learning algorithms (LeCun et al., 2015). Overfitting is another common issue where the network learns the training data too well, including the noise, and performs poorly on new, unseen data.

A loss function, also known as a cost function or objective function, is a measure of how well a machine learning model performs on a dataset. It quantifies the difference between the predicted output of the model and the actual target values in the dataset. The goal of training a deep learning model is to minimize this loss function, which means making the model's predictions as close to the actual targets as possible. Deep-pocket's model chosen loss function is Binary Cross-Entropy Loss, which combines a Sigmoid activation function and the binary cross-entropy loss, which is suitable for training neural networks to perform binary classification.

The chosen parameters to train the model were the following ones:

Table 1. model_100.pth parameters

Parameter	Value
Learning rate	0.001
Number of epochs	100
Step Size	10
Gamma	0.5

The loss function evolution of the training and validation datasets was also assessed:

Figure 1. Training and validation loss function for the deep-pocket model trained with 100 epochs.

To assess the model we use ROC curve is a graphical plot that illustrates the diagnostic ability of a binary classification model across various thresholds. It is a commonly used tool for assessing the performance of machine learning models that perform binary classification.

$$\text{True Positive Rate (TPR)} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{False Positive Rate (FPR)} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Figure 2. TPR and FPR formulas

The ROC curve is created by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) at various threshold settings. The ROC area under the curve (AUC) quantifies the overall performance of the model across all possible thresholds. The closer the AUC is to 1 the better the model performs.

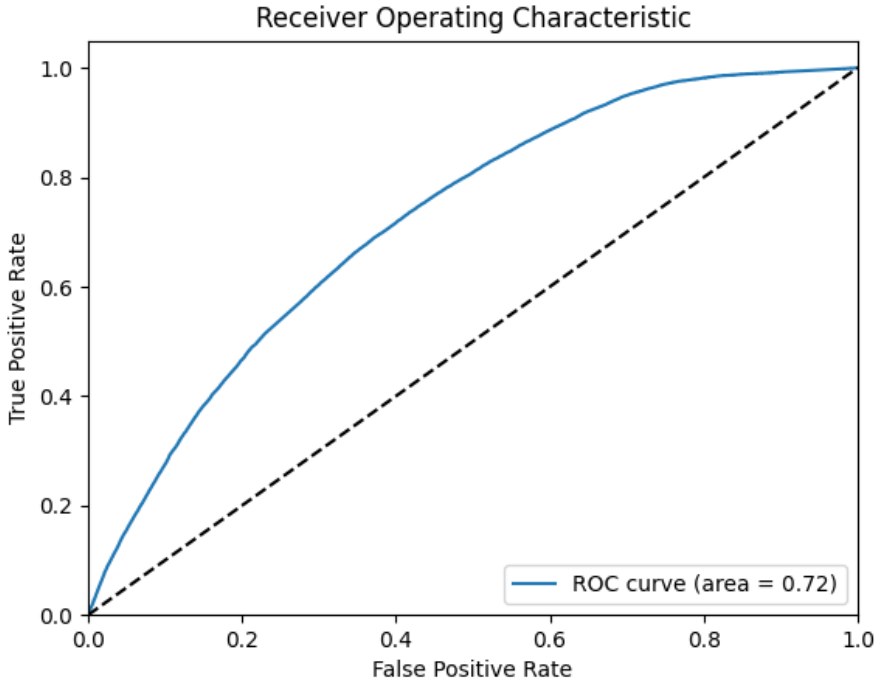

Figure 3. ROC curve and area under the curve (AUC = 0.72)

The model used for deep-pocket has an AUC value of 0.72, which is an acceptable value considering the amount of data the model had to train (less than 1000 PDB's). If more data was used to train the model this value would probably be higher, however the amount of time and computational resources in order to obtain huge datasets would increase considerably.

For feature extraction, given that the input is a PDB file, we looked for features used in other structure-based binding site prediction methods. We obtained protein PDB files with their corresponding binding pocket PDB files from the paper Ahmed et al. (2021). From this, we were able to obtain our first feature: whether a particular residue is on a binding site or not.

Then, we assigned secondary structure types to our protein structures using the program DSSP (Define Secondary Structure of Proteins) (Kabsch & Sander, 1983). The algorithm identifies

hydrogen bonds between main chain carbonyl and amide groups. Partial charges are applied to the amide and carbonyl bonds, and the C, O, N, and H atoms are assumed to be point charges (hence C has charge $+\rho 1$, O $-\rho 1$, N $-\rho 2$, and H $+\rho 2$). The electrostatic energy between these 4 atoms is calculated, and if it is smaller than -0.5 kcal/mol, a hydrogen bond exists. Helices and sheets are then identified where there are characteristic hydrogen bond patterns.

DSSP has 8 different secondary structure assignments, which are assigned by order of preference – HBEGITSC:

- G – 3_{10} helix
- H – α-helix
- I – π-helix
- E – β-sheet
- B – β-bridge
- T – helix turn
- S – bend (high curvature)
- C – coil (none of the above)

From the output of DSSP, we also obtained as features the Phi and Psi angles, which are the peptide backbone torsion angles as described in the IUPAC standard, as well as the solvent accessibility of each residue, which is the water exposed surface in Ångström^2 and essentially describes whether a given residue is buried or exposed to the solvent.

From the coordinates of the PDB file, we also calculated the total contacts of each residue. Residue–residue contacts (or simply "contacts") in protein 3-D structures are pairs of spatially close residues (Adhikari & Cheng, 2016). A pair of amino acids are in contact if the distance between their specific atoms (we considered the alpha-carbon) is less than a distance threshold (we used 5 Å as this threshold.)

References

Adhikari, B., & Cheng, J. (2016). Protein Residue Contacts and Prediction Methods. En *Methods in molecular biology (Clifton, N.J.)* (pp. 463-476). https://doi.org/10.1007/978-1-4939-3572-7_24

Ahmed, A., Mam, B., & Sowdhamini, R. (2021). DEELIG: A Deep Learning Approach to Predict Protein-Ligand Binding Affinity. *Bioinformatics And Biology Insights*, *15*, 117793222110303. https://doi.org/10.1177/11779322211030364

DeepAI. (2020, 25 junio). *Feed Forward neural network*. DeepAI. https://deepai.org/machine-learning-glossary-and-terms/feed-forward-neural-network

Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers (Print)*, *22*(12), 2577–2637. https://doi.org/10.1002/bip.360221211

LeCun, Y., Bengio, Y., & Hinton, G. E. (2015). Deep learning. *Nature (London)*, *521*(7553), 436-444. https://doi.org/10.1038/nature14539

Xia, Y., Pan, X., & Shen, H.-B. (2024). A comprehensive survey on protein-ligand binding site prediction. *Current Opinion in Structural Biology*, *86*, 102793. https://doi.org/10.1016/j.sbi.2024.102793

Zhao, J., Cao, Y., & Zhang, L. (2020). Exploring the computational methods for protein-ligand binding site prediction. *Computational And Structural Biotechnology Journal*, *18*, 417-426. https://doi.org/10.1016/j.csbj.2020.02.008