Original Article

# A GU-Net-Based Architecture Predicting Ligand–Protein-Binding Atoms

**Fatemeh Nazem[1,2], Fahimeh Ghasemi[2], Afshin Fassihi[3], Reza Rasti[4], Alireza Mehri Dehnavi[1,5]**

*Departments of [1]Bioelectrics and Biomedical Engineering and [2]Bioinformatics and Systems Biology, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, [3]Department of Medicinal Chemistry, School of Pharmacology and Pharmaceutical Sciences, Isfahan University of Medical Sciences, [4]Department of Biomedical Engineering, Faculty of Engineering, University of Isfahan, [5]Medical Image and Signal Processing Research Center, Isfahan University of Medical Sciences, Isfahan, Iran*

## Abstract

**Background:** The first step in developing new drugs is to find binding sites for a protein structure that can be used as a starting point to design new antagonists and inhibitors. The methods relying on convolutional neural network for the prediction of binding sites have attracted much attention. This study focuses on the use of optimized neural network for three-dimensional (3D) non-Euclidean data. **Methods:** A graph, which is made from 3D protein structure, is fed to the proposed GU-Net model based on graph convolutional operation. The features of each atom are considered as attributes of each node. The results of the proposed GU-Net are compared with a classifier based on random forest (RF). A new data exhibition is used as the input of RF classifier. **Results:** The performance of our model is also examined through extensive experiments on various datasets from other sources. GU-Net could predict the more number of pockets with accurate shape than RF. **Conclusions:** This study will enable future works on a better modeling of protein structures that will enhance knowledge of proteomics and offer deeper insight into drug design process.

**Keywords:** *Graph convolutional neural network, point cloud semantic segmentation, protein–ligand-binding sites, three-dimensional U-Net model*

**Submitted:** 30-Jul-2021 **Revised:** 23-Aug-2021 **Accepted:** 28-Oct-2021 **Published:** 27-Mar-2023

## Introduction

Drug design is generally defined as designing or discovering new drugs based on the available information about protein targets. The proteins play a significant role in the organic life. The role they play is not only functional but also structural. Within organisms, proteins have a key impact on metabolic processes and activities. To function in living organisms, they need to bind to other biomolecules, such as nucleic acids or small molecules. These small molecules, which are known as ligands, are aimed at enhancing or inhibiting the protein function in metabolites. Ligand-binding sites are amino acid residues at specific portions of the protein that participate in the interaction between the protein and the ligand.[1]

Localization of pockets is central to the structure-based drug design process as it can be utilized to design and develop new treatments. The binding sites of proteins contain important information about their biological function. In cases where a particular function and a specific binding site of a protein are associated with a disease, such as cancer, the binding site can be as a potential target for treatment.[2] Laboratory methods might require valuable tools and time, but they have also some bottlenecks such as sample preparation and data interpretation. Therefore, they require a high level of expertise and experience. The other method is based on computation of protein–ligand docking, which scanned the whole surface of the protein to identify potential hotspots for ligand interactions. MolSite and BINDSURF are the efficient blind methods that dock the ligands on the protein to predict pockets.[3,4] They require the potential suitable ligands of structures to do docking.

Computational prediction of binding sites has attracted much attention as an alternative to the experimental methods.[5] These methods can be broadly classified into geometric, energetic, evolutionary, and machine learning. In the first two methods, the three-dimensional (3D) structure information of the proteins is used, while in the remaining two, the sequence information or 3D structure information of proteins or both is used.[6] In this study,



**How to cite this article:** Nazem F, Ghasemi F, Fassihi A, Rasti R, Dehnavi AM. A GU-Net-based architecture predicting ligand–Protein-binding atoms. J Med Signals Sens 2023;13:1-10.

***Address for correspondence:***
*Dr. Alireza Mehri Dehnavi, Department of Bioelectrics and Biomedical Engineering, School of Advanced Technologies in Medicine, Isfahan University of Medical Sciences, Isfahan, Iran. Medical Image and Signal Processing Research Center, Isfahan University of Medical Sciences, Isfahan, Iran.*
*E-mail: mehri@med.mui.ac.ir*

**Access this article online**

**Website:** www.jmssjournal.net

**DOI:** 10.4103/jmss.jmss_142_21

**Quick Response Code:**

machine learning algorithm based on the 3D structure of the protein is used to predict the binding sites.

P2Rank is one of the recent machine learning methods utilizing the random forest (RF) algorithm to classify the protein surface atoms as binding or nonbinding sites. The ligand ability of the solvent accessible surface (SAS) points is scored using RF based on the atomic feature vectors of neighboring atoms. The SAS points are described by physicochemical, geometric, and statistical properties derived from their neighbors in the 3D surrounding sphere.[7]

The availability of a large number of crystallography structures in recent years and the-state-of-the art performance of deep models in a variety of tasks have accelerated the use of deep models in drug design fields such as binding site prediction.[8-11] The recent use of deep models in pocket prediction is given in Table 1.

In the models given in Table 1, a grid box is located around the protein and different features are computed for each grid box voxel as the input. In convolutional neural network CNN-based classification models, the protein is discretized to subgrids, and a binding site score is predicted for each subgrid.[8,12-14] The whole grid box with all information is fed to semantic segmentation and object detection models.[15-17] The semantic segmentation models are based on the U-Net architecture.[18] They predict a class label for every voxel of inputs unlike CNN, which predicts a single class label for all voxels of inputs. U-Net could improve the localization of the predictions. Given the sparsity of protein atoms, the U-Net model based on submanifold sparse convolution is applied on input grid box at the PointSite.[19]

Actually, the standard implementation of convolution operations is dense. The operations are optimized to be implemented on regular and Euclidean data. That is, while in real world, a large number of data such as social networks, biological networks, atoms of protein structures, or 3D point cloud data have graph or non-Euclidean structure. To use the advantages of CNNs on these data, 3D point clouds are usually mapped to a 3D occupancy regular grid. An alternative method to work on 3D data without destroying geometric information is based on the graph concepts.[20]

Like in CNN, the convolution operation in graph convolution network (GCN) learns the features exploited from neighboring nodes. CNN operates on regular data, but GCN operates on irregular grid with disorder nodes and a variable number of connections. There are different convolution methods in the graph network based on spectral or spatial information such as GCN,[21] GraphSAGE,[22] ARMA convolutions,[23] and graph convolution skip layer.[24]

One of the other important aspects of generalizing CNN on graph data in the addition of the defining convolutional layer is the definition of pooling layers. Different methods proposed to do pooling and unpooling operations are based on graph concepts. Some of these methods are minCUT pooling,[24] differentiable pooling,[25] self-attention graph pooling, and TopKPool.[26]

The point cloud semantic segmentation models are also done using graph neural network.[27] The architecture of these methods is based on U-Net model to classify every input point. Graph U-net is also used for node classification and graph classification task under transductive and inductive learning setting, respectively.[26]

GCNs have shown great performance in predicting the binding affinity,[28] protein function,[29] and the quantitative structure–activity relationship model.[30]

We used GCN to predict binding sites on the 3D non-Euclidean structure of the protein atoms. A graph is made from a 3D protein structure and its atoms are applied to the network, irrespective of the voxel regularity in the grid box. The features of each atom are signals on the graph. In the proposed model, GCN is extended to do point cloud semantic segmentation based on the U-Net architecture. To evaluate the result of the proposed model named GU-Net, the RF model is used as an atom-based model. The training of the model is done by using scPDB database, and the test of the model has been conducted on the test data by different sources and using various evaluation metrics.

**Table 1: Deep learning based methods in the prediction of pockets**

| Method | Network type | Approach | Descriptors | Year |
|---|---|---|---|---|
| Deepsite[8] | CNN | Classification | Pharmacological descriptors from HTMD | 2017 |
| Hybrid descriptors[14] | CNN | Classification | Combination of geometry and energetic descriptors | 2019 |
| FRSite[12] | Fast-RCNN | Object detection | Pharmacological descriptors from HTMD | 2019 |
| Pointsite[15] | U-Net based on submanifold sparse CNN | Semantic segmentation | The atom and residue type with coordinates of the atoms | 2022 |
| Kalasanty[16] | U-Net | Semantic segmentation | Pafnucy features | 2020 |
| Voxel based U-Net[17] | U-Net | Semantic segmentation | Pharmacological descriptors from HTMD | 2021 |
| DeepSurf[13] | Res-Net | Classification | Pafnucy features on the protein surface atoms | 2021 |

CNN: Convolutional Neural Network, RCNN: Regions with Convolutional Neural Networks, HTMD: High Throughput-Molecular-Dynamics

## Material

### Atom descriptors and label computing

The descriptors defined in Pafnucy[9] are computed for each atom; therefore, they are used for the proposed graph network as the feature vectors of the nodes. They are listed in Table 2.[9]

To define the true pockets used in evaluating the models, the distance of each atom from any atom of the ligand is computed; if this distance is smaller than 4.5 Angstroms, the atom is considered as the binding atom. The number of binding atoms in comparison with nonbinding atoms is very low, thereby causing to severe unbalancing in data.

## Datasets

Training of the model is done on scPDB that contains 4782 proteins with about 17k of high-quality binding sites.[31] To prevent the leakage of the same protein structure in training or testing the model, the splitting of data is done based on UniProt ID. Each UniProt ID contains PDB IDs of the same protein structure. The fivefold cross-validation is applied on 90% of UniProt IDs. In each fold, one group is used as validation and the others are used to train the model. The remaining 10% of UniProt IDs are used as test datasets. The datasets from other sources are also used to evaluate the proposed model. Chen11,[32] B210,[33] and DT198[34] are well-known datasets used in testing the binding site prediction methods. The B210 and DT198 datasets containing 210 and 198 structures are benchmark datasets from Ligsite-csc and MetaPocket, respectively. Chen11 dataset containing 251 structures is also used to evaluate some of the previous binding site prediction methods. Since binding a ligand to a protein may change the conformation of the protein, the proposed model is also evaluated on protein structures without bounded ligands. U48 and B48, which contain 48 corresponding proteins in unbound and bound state, respectively, are also utilized to assess the model.[33] In unbound structures, the ligands of B48 proteins are used as the ligand of corresponding unbound proteins in U48 to have a comparison with the true binding sites. Coach-420 dataset[35] containing 420 protein structures is also employed to assess the models.

The annotation and filtering of PDB IDs were done in scPDB datasets completely. The coordinates of PDB files are precisely assessed to produce standardized files in the scPDB database. The relevant ligands of PDB IDs are also specified in scPDB data. In the test datasets, irrelevant ligands are removed according to LIG Tool rules.[15] The relevant ligands have five or more atoms. Polynucleotide ligands, metal ions, and irrelevant ligands according to scPDB database are removed from test datasets.

## Graph convolutional network architecture

We employ the GCN model to work directly on the protein atoms, use local geometry information, and inherent non-Euclidean structure of data. The protein structure and its corresponding point cloud structure are shown in Figure 1. Here, the main concepts of the graph convolutional layer and the pooling layers of GCN are briefly described, and then, the proposed architecture model is introduced.

In this method, the protein is considered as an undirected graph, G = (V, E, X), where V = {1,…, N} is the set of atoms as nodes, E ∈ V × V is the set of edges, and X ∈ $R^{(N \times F)}$ where $F$ shows the dimension of each node attributes. The 18 atomic features introduced by Pafnucy and the spatial coordinate of atoms are considered as the feature vector for each atom ($F = 21$). A set of nearest neighboring atoms is used as the $k$-nearest neighborhood. The neighboring number is limited to $K = 20$ based on validation results. Internode edges are weighted using Gaussian-based filter as given in Eq. 1.[36] These weights are used to compute the adjacency matrix.

$$W_{i,j} = \begin{cases} exp\left(-\frac{\| x_i - x_j \|^2}{\sigma^2}\right) & if\ j \in N_k(i) \\ 0 & otherwise \end{cases} \quad (1)$$

where $x_i$ is the coordinates of the atom and $\mathcal{N}_k(i)$ is the set of $K$-nearest neighbors of node $i$. Therefore, there is nonzero weight between neighboring atoms and zero weight between nonneighboring atoms.

The graph convolution operation used in this work is based on the first-order spectral filter, which is defined as multiplication of a signal x ∈ $R^N$ with a diagonal filter $g_\theta$ in the Fourier domain as given in Eq. 2.[37]

$$g_\theta * x = U g_\theta U^T x \quad (2)$$

**Table 2: Atom descriptors as feature vectors of nodes[9]**

| Type of feature | Description |
|---|---|
| Atom type | 9 bits (one-hot coding) correspond to atom type (B, C, N, O, P, S, Se, halogen, and metal) |
| Hyb | 1 integer to show hybridization of an atom |
| Heavy_valence | 1 integer to count the number of bonds with other heavy atoms |
| Hetro_valence | 1 integer to count the numbers of bonds with other hetero atoms |
| SMARTS patterns | 5 bits defined 1 when the properties defined in SMARTS patterns present (hydrophobic, aromatic, acceptor, donor, and ring) |
| Partial charge | A float value |

SMARTS: SMiles ARbitrary target specification pattern

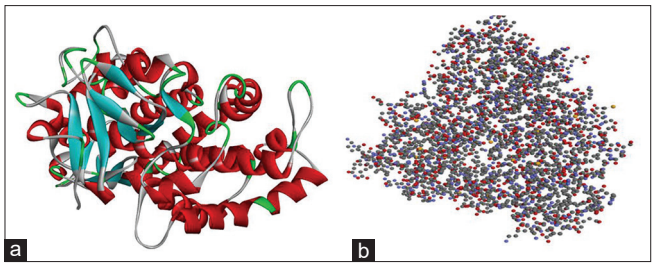


**Figure1: a) The Protein structure b) Corresponding point-cloud structure of the protein. The figures are produced in Autodock4**

If the normalized graph, Laplacian is defined as $L = I_N - \tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}} = U\,U^T$, $U$, is the eigenvector matrix. This formula can be approximated by the Chebyshev polynomial with $k = 1$ as defined in Eq. 3:[38]

$$\bar{X} = \sigma(\tilde{D}^{\frac{-1}{2}}\tilde{A}\tilde{D}^{\frac{-1}{2}}XW^l) \tag{3}$$

where $\tilde{A} = A+I$ is the graph adjacency matrix with added self-loop and $\widetilde{D} = \sum_j \tilde{A}_{ij}$ is the degree matrix and $W^l$ is the trainable weight matrix.

The initial features of the nodes are also added to the graph convolution operation through skip connection as given in Eq. 4. It does not require the added self-loop and is known as the graph convolution skip.[24]

$$\bar{X} = \sigma(\tilde{D}^{\frac{-1}{2}}A\tilde{D}^{\frac{-1}{2}}XW_1^l + XW_2^l) = \sigma(\tilde{A}XW_1^l + XW_2^l) \tag{4}$$

where $\sigma$ is ReLU, and there is an alternate definition for $\tilde{A} = \tilde{D}^{\frac{-1}{2}}A\tilde{D}^{\frac{-1}{2}}$, which is a symmetric normalized version of $A$ to avoid feature vector distribution changes.

In GCN, the convolutional layer is also followed by the pooling layer. The pooling layers are defined in a way to cover the concepts of the graphs. minCUT pooling is one of the recent proposed graph poolings that can be used in semantic segmentation. The nodes $\mathcal{V}$ of graph $\mathcal{G}$ is partitioned to $K$ disjoint subgraphs using the minCUT problem. The problem is defined as maximizing Eq. 5.

$$\frac{1}{K}\sum_{k=1}^{K}\frac{links(V_k)}{degree(V_k)} = \frac{1}{K}\sum_{k=1}^{K}\frac{\sum_{i,j\in V_k}\varepsilon_{i,j}}{\sum_{i\in V_k, j\in IV_k}\varepsilon_{i,j}} \tag{5}$$

The numerator is the sum of the edges within each cluster and the denominator is the sum of the edges between cluster nodes. Therefore, it improves similarity within clusters and dissimilarity between clusters. By defining a cluster assignment matrix $C \in \{0,1\}^{N\times K}$, where $C_{i,j}$=1 if node $i$ is in the cluster and otherwise, Eq. 5 can be rewritten as Eq. 6. Finding an optimal solution for this problem is an NP-hard problem.

$$maximize\frac{1}{K}\sum_{k=1}^{K}\frac{(C_k^T AC_k)}{(C_k^T DC_k)} \tag{6}$$

The multilayer perceptron (MLP) could be used to solve this Np-hard spectral clustering problem. MLP with softmax function output is used to compute a continuous cluster assignment for each node. The soft clustering assignment matrix is $S$ computed as Eq. 7.

$$S = MLP(\bar{X}) = softmax(ReLU(\bar{X}W_1)W_2) \tag{7}$$

The loss function consists of two auxiliary terms which are designed to optimize the trainable parameters: minCUT loss $L_C$ and the orthogonality loss $L_O$ as shown in Eq. 8.

$$L_u = L_c + L_o = -\frac{Tr(S^T\tilde{A}S)}{Tr(S^T\tilde{D}S)} + \left\| \frac{S^TS}{\|S^TS_F\|} - \frac{I_K}{\sqrt{K}} \right\|_F \tag{8}$$

where $\tilde{D}$ is the degree matrix of $\tilde{A}$, and $\| \|_F$ is the Frobenius norm. By minimizing the numerator of $L_C$, the strongly connected nodes are clustered together while the denominator assesses the size of clusters to prevent small clusters. The orthogonality loss prevents an assignment of all nodes to the same cluster or prevents the assignment of all nodes to all clusters equally. It helps find clusters which are orthogonal. Finally, by optimizing the parameters, the graph is reduced to Eq. 9.

$$A^{pool} = S^T\tilde{A}S; \tag{9}$$

$$X^{pool} = S^T\bar{X}$$

where $A^{pool} \in R^{K\times K}$ is symmetric matrix, and $X^{pool} \in R^{K\times F}$. Each $x_{i,j}^{pool}$ in $X^{pool}$ is the sum of features $j$ of the nodes in the cluster.

Since the trace of is maximized with the loss function Eq. 8, the diagonal elements of $A^{pool}$ are much larger than other elements. The final graph has a very strong self-loop due to node self-adjustment. To resolve this problem, the diagonal of $A^{pool}$ is removed, and then, it is normalized using its degree matrix. The new adjacency matrix $\tilde{A}^{pool}$ is computed by Eq. 10.[24]

$$\hat{A} = A^{pool} - I_k.diag(A^{pool}); \tilde{A}^{pool} = \hat{D}^{\frac{-1}{2}}\hat{A}\hat{D}^{\frac{-1}{2}} \tag{10}$$

In the proposed model, GCN is used to predict the binding sites. It is extended to do point cloud semantic segmentation based on the U-Net architecture. The architecture of the proposed graph binding site prediction is shown in Figure 2.

The encoder and decoder path of GU-Net contains four steps. Each step has a convolution block compromising two sets of graph convolutional skip with batch normalization and ReLU activation function. The minCUT pooling is applied at the end of each step. In the decoder path like in encoder, convolution blocks are applied after each unpooling layer.

We used the same soft clustering assignment matrix $S$ as given in Eq. 11 to unpooling the graph from the preceding layer. In the computed feature map $X^{unpool}$, the features of nodes in a cluster are similar.

$$X^{unpool} = SX^{pool} \tag{11}$$

$$A^{unpool} = SA^{pool}S^T$$

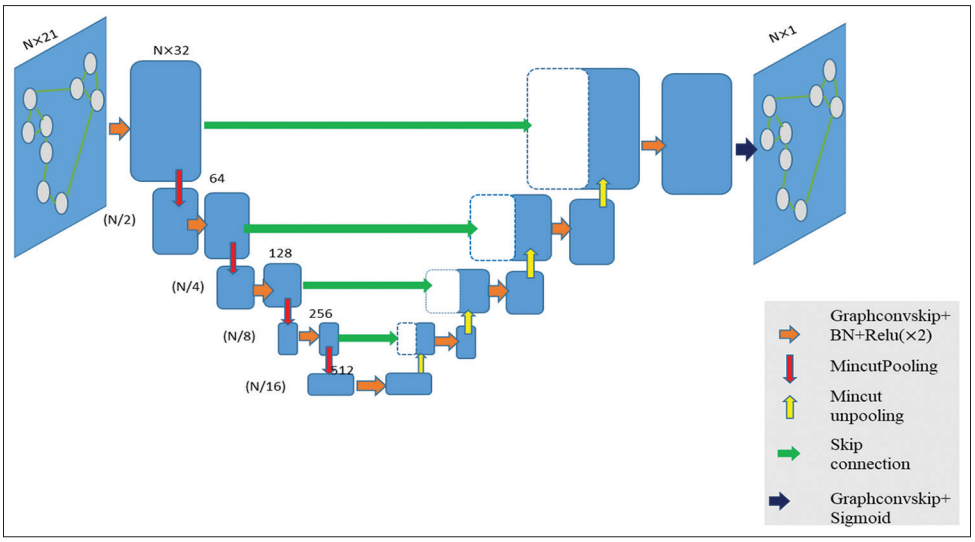


**Figure 2: The GU-Net architecture; N indicates the number of nodes and the number of feature maps is shown at the top of the box**

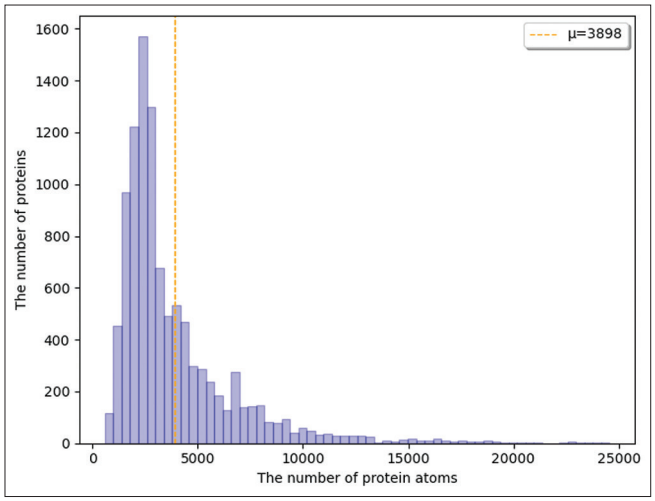


**Figure 3: The distribution of the number of proteins based on the number of atoms**

The dice loss is used as supervised loss function for the optimization network. This loss function can alleviate the severe unbalancing of data due to a large number of nonbinding atoms compared to binding atoms. Our model was implemented in Spektral with Keras and TensorFlow programming interface.[39]

To define the number of nodes to represent the number of protein atoms, the histogram of the number of the proteins based on the number of heavy atom in the proteins is shown in Figure 3. The number of nodes is assigned to cover the most number of atoms without increasing the computational complexity.

The median and mean of the data are 2825 and 3898, respectively. To have an input of power of two and a reasonable computational complexity, $N = 4096$ is chosen as the number of nodes. It is near the mean and could cover most part of the large numbers of proteins. 4096 heavy atoms around the center of the protein are selected as the nodes of the model and feature vectors are computed for them. In the test time, if the number of input atoms is more than 4096, the input may be chosen from different parts of the proteins based on the spatial shape of the protein.

We also used the RF classifier to evaluate deep models. To improve the RF classification performance, the local information is used instead of single atom information. The feature vector is made using local information of $k = 25$ nearest neighbor atoms. The 9-bit (one-hot coding) showing the type of the origin atom is converted to one categorical feature and 9 remaining features of the origin atom are added with nine correspond features for all 25 neighboring atoms. The size of resulting feature vector is 10, which is concatenated to the coordinates of the origin atom.

RF is trained on the protein structures of scPDB. Since the number of PDB IDs is very large, one protein is randomly selected from each UniProt ID. RF with 150 trees, 13 features, and unlimited depth is tried on these data. The unbalancing of data is too high; therefore, the nonbinding site atoms are downsampled with the ratio of 6:1.

## Evaluation metrics

The complementary metrics are used to evaluate the predicted pockets. These metrics could consider the size of ligands and the shape of the predicted pockets: They are listed as follows:

- Matthews correlation coefficient (MCC): MCC as a reliable statistical metrics is used to evaluate the performance of the binary prediction model
- Success rate of DCC: The distance of the binding site center from the ligand center is computed. If this distance is smaller than a given threshold, the prediction is considered as a successful. The number of successful predicted pockets on the total number of pockets is defined as the success rate of DCC. In this work,

different thresholds ranging from 2 to 10 Angstrom are used to assess the models

- Success rate of DCA: In this metric, the distance of the binding site center from any atoms of the ligand is used to define the successful predicted pocket. Its ratio is similar to DCC
- Discretized volumetric overlap (DVO): The DVO between true binding atoms and predicted is computed to consider the shape of predicted pockets. It is defined as Eq. 12.

$$J = \frac{\#\left|V_t \cap V_p\right|}{\#\left|V_t \cup V_p\right|} \tag{12}$$

where $V_t$ is the true binding atoms and $V_P$ is the predicted binding atoms.

The results of our model are compared with DeepSite as the classification and Kalasanty as the segmentation deep model. The atom-based P2Rank model using RF classifier is also compared with the proposed models.

## Results and Discussion

In these methods to compute the evaluation metrics, a label is assigned to each atom of the proteins. To cluster the atoms and arrange them as binding pockets, the mean shift method is applied on the output label map.[40] This fast algorithm clusters the predicted binding atoms as separated binding sites. The clusters containing 30 atoms or less are removed. The predicted pockets are ranked based on the binding site probability.

The results of cross-validation on RF are used to find the best number of trees and number of neighboring atoms. The results are shown in Figure 4. Based on the validation results, 150 is selected for the number of trees and 25 neighboring atoms are used to make the feature vectors.

The fivefold cross validation is performed on the train datasets to evaluate the models. One PDB Ids from each UniProt ID test set is randomly selected. The evaluation metric results of GU-Net are averaged over fivefold for each protein. The performance of RF and proposed GU-Net is evaluated on the identical test data from scPDB and the mean and standard deviation of each metric on all proteins of test data are represented in Table 3.

The ROC curve of GU-Net and RF on test data from scPDB are also shown in Figure 5. As shown in Figure 5, the area under ROC curve of GU-Net is higher than RF.

The performance of the proposed GU-Net is compared with RF on independent test datasets. The MCC metric is a good measure for unbalanced data. To show this, in addition to MCC, the precision and sensitivity of the methods are compared in this experiment as shown in Figure 6. The low precision is corresponding to high false positive and low sensitivity corresponds to high false negative. MCC is considered to overcome this contradiction to gain reliable results.

Given Figure 6, although the precisions of RF on test datasets are higher than the on the proposed graph model, the sensitivity and MCC of our model are higher than RFs.

The DCC metric of the methods is computed with different thresholds on test datasets as shown in Figure 7.

The DCC success rates of GU-Net outperform RF on all independent test datasets. For example, the distance between the predicted pockets and true pockets by GU-Net and RF is compared on DT198 test data in Figure 8. Each point shows the DCC value obtained by GU-Net and RF models. If the model could not predict a pocket, this distance is considered as 120 Å as a maximum value.

As shown in Figure 8, more DCC value points are below the diagonal line, which shows the lower DCC of predicted pockets by the GU-Net model in comparison with RF.

The predicted pockets are saved in the.mol2 format that can be used in other molecular software. The predicted pocket for one of the protein structures of DT198 (PDB ID: 1lpb. pdb) as an example is shown in Figure 9a. The overlap

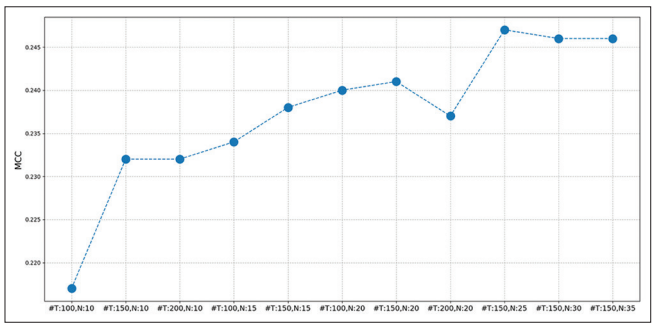


**Figure 4: The grid search results on different hyper parameters. T shows the number of tree and N shows the number of neighbors used to make the feature vector**

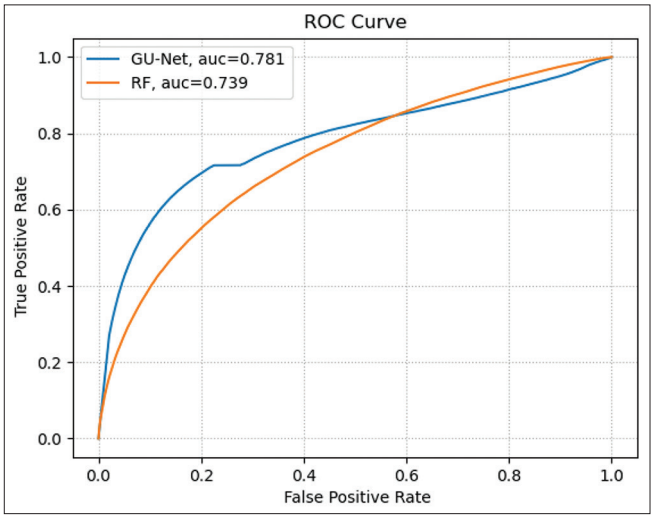


**Figure 5: ROC curve of GU-Net and RF on test data from scPDB**

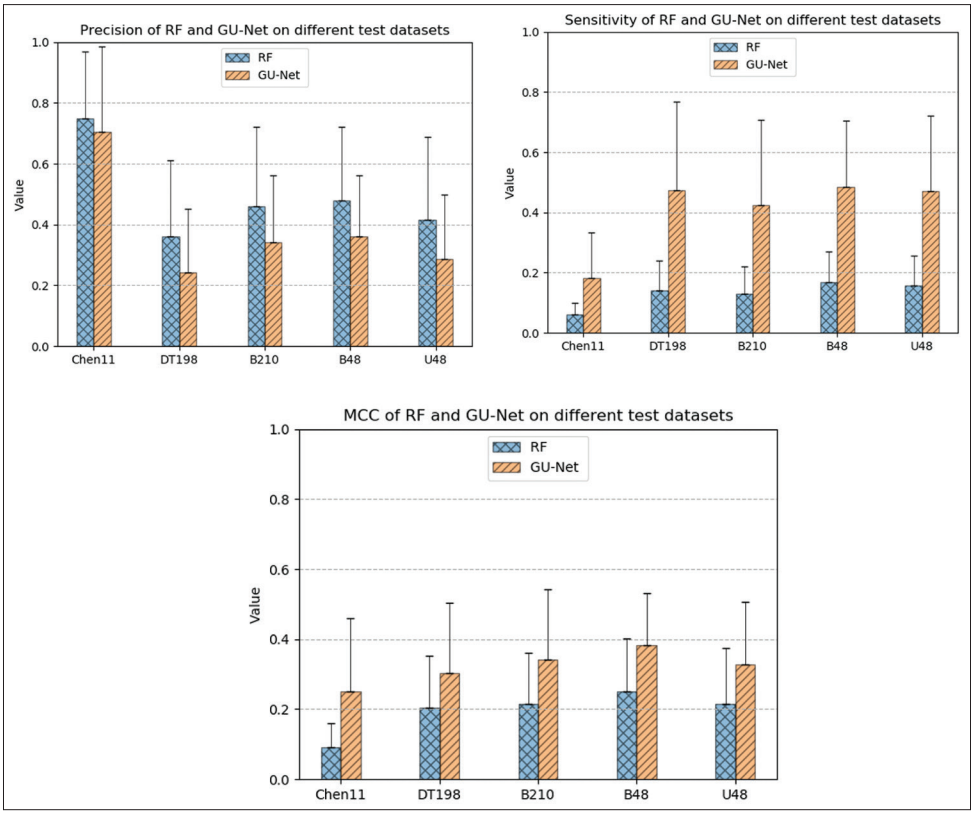


**Figure 6: Precision, sensitivity, and MCC evaluation of RF and GU-Net on test data. Bar charts show the mean of the metrics and error bars show +standard deviation**

**Table 3: Evaluation of random forest and GU-Net on test data from scPDB**

| Method | MCC | DVO | Success rate of DCA | Success rate of DCC |
|---|---|---|---|---|
| RF method | 0.201±0.15 | 0.123±0.09 | 37.4 | 8.81 |
| GU-Net | 0.352±0.224 | 0.275±0.156 | 46.45 | 17.31 |

RF: Random forest, MCC: Matthews correlation coefficient, DVO: Discretized volumetric overlap, DCA: Distance of the binding site center from any atoms of the ligand, DCC: Distance of the binding site center to ligand center, scPDB: SCreening Protein Data Bank

of the predicted pocket and true pockets is also shown in Figure 9b. The predicated pocket could cover most part of true pockets and its DVO value is 0.61.

The results of GU-Net and RF are compared with DeepSite and Kalasanty as shown in Table 4. The threshold of 4 Å is considered for computing DCA and DCC.

As shown in Table 4, the metric results of GU-Net are superior to RF in all test datasets. RF could not predict the shape of pockets correctly. The MCC and DVO results of Kalasanty and GU-Net as the segmentation models are bigger than DeepSite as the classification model. Although the number of correctly predicted pockets by GU-Net is less than DeepSite and Kalasanty, the MCC and DVO results of GU-Net model are more accurate than Kalasanty. The results of GU-Net model on unbound protein structures are close to the bound structures. The model could predict pockets for unbound protein structures with a reasonable performance.

At the end of this part, the results of the methods are also compared with P2rank based on RF. The results of the models on Coach-420 as another test dataset are given in Table 5.

The variety number of features provided for each SAS point in P2rank helps the model to predict the binding site with better performance than our RF. The DCC and DCA success rate of the Kalasanty outperform than other models on Coach-420. GU-Net could predict shape of the pockets with more overlap to true pockets than other models.

## Conclusion

In our study, we concentrate on the atoms of the protein without any preprocessing and proposing any grid box around them. Our method uses graph convolution operation based on U-Net models on atom proteins to predict the binding atoms. To overcome the severe unbalancing problem of the data, we used dice coefficient as loss function and MCC is used as the evaluation metric. As seen in the RF model results in Figure 6, the unbalancing in data leads to high precision and low sensitivity. RF predicts a

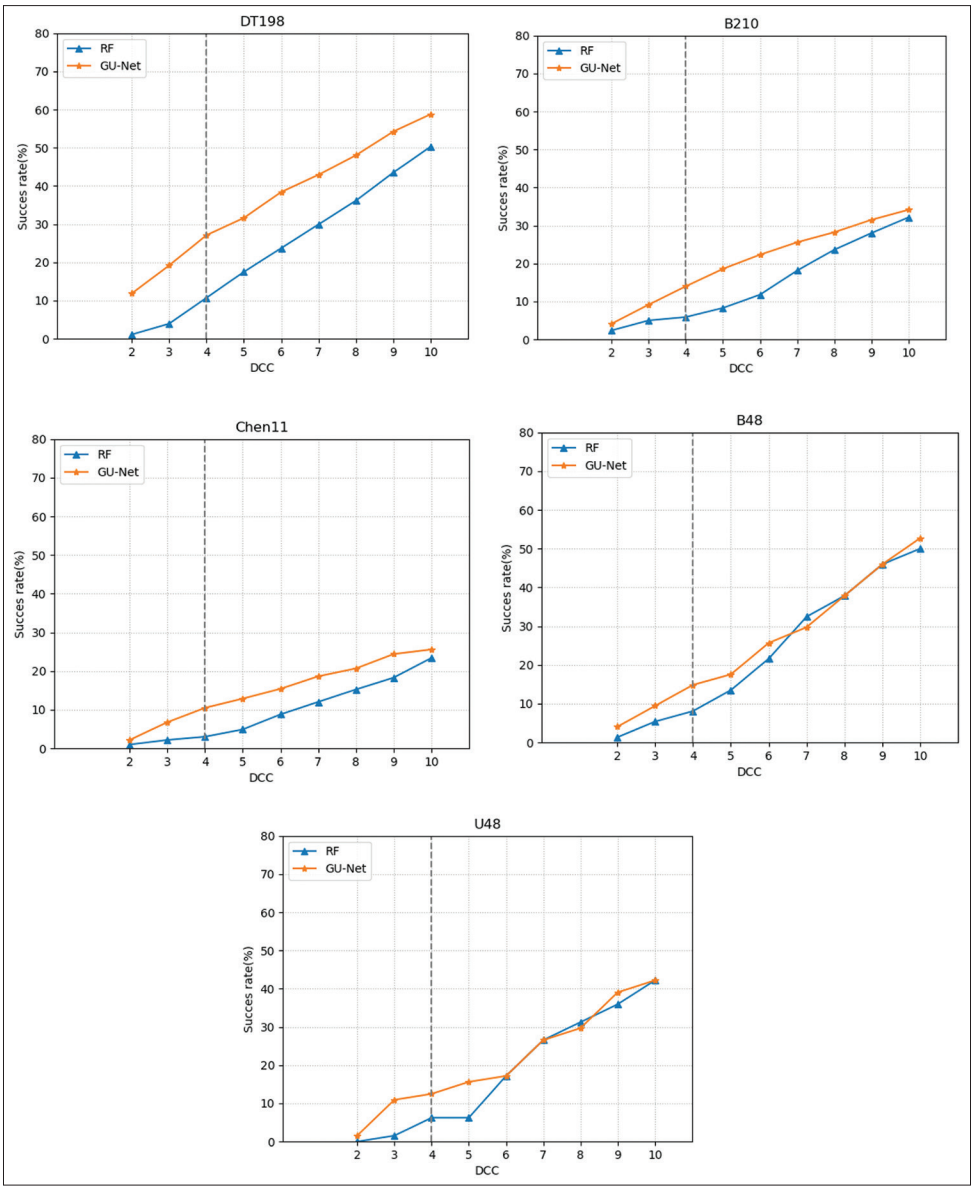


**Figure 7: Success rate of DCC in different thresholds**

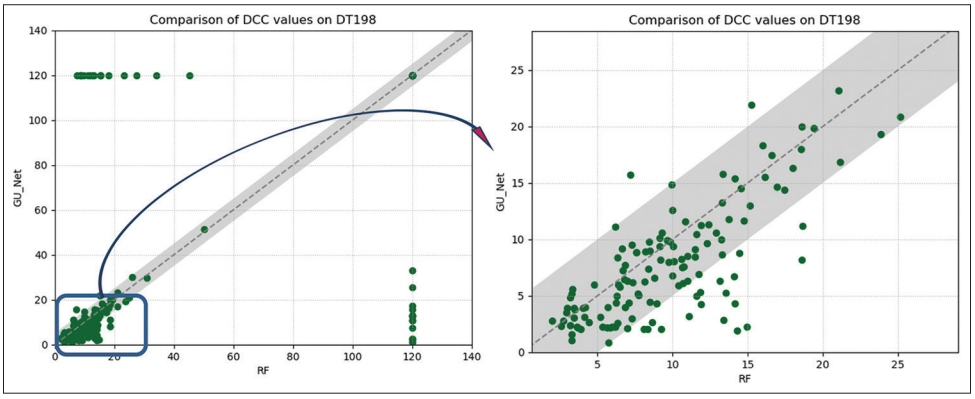


**Figure 8: Comparison of DCC values of GU-Net and RF. Zoom to show with high quality. Each point shows the DCC values of the GU-Net and RF on DT198. The shaded area shows the 5Ao difference from the diagonal line**

fewer number of true binding pockets atoms than GU-Net. GU-Net using dice loss can achieve better tradeoff between precision and recall. Therefore, the MCC values of the proposed method are better than RF as shown in Figure 6.

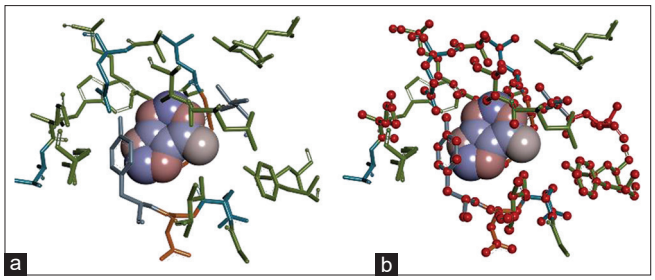


**Figure 9: The predicted pocket for 1lpb.pdb PDB ID from DT198 dataset. The ligand is bubble shaped. a) The predicted binding atoms covering the annotated ligand. b) The overlap between predicted pockets and true pockets shown in ball and stick format**

**Table 4: Comparison of DeepSite, Kalasanty, GU-Net, and random forest on test data**

| Data | Method | MCC | DVO | Success rate of DCA | Success rate of DCC |
|---|---|---|---|---|---|
| Chen11 | DeepSite | 0.267±0.19 | 0.157±0.14 | 28.73 | 19.82 |
| | Kalasanty | 0.295±0.23 | 0.173±0.14 | 28.1 | 18.62 |
| | GU-Net | 0.250±0.21 | 0.194±0.17 | 19.7 | 10.8 |
| | RF method | 0.06±0.07 | 0.088±0.03 | 12.71 | 7.4 |
| DT198 | DeepSite | 0.254±0.17 | 0.129±0.11 | 52.87 | 38.50 |
| | Kalasanty | 0.283±0.21 | 0.138±0.12 | 59.1 | 40.22 |
| | GU-Net | 0.303±0.20 | 0.191±0.17 | 47.45 | 27.12 |
| | RF method | 0.173±0.15 | 0.11±0.09 | 25.9 | 12.7 |
| B210 | DeepSite | 0.292±0.184 | 0.169±0.12 | 31.93 | 23.95 |
| | Kalasanty | 0.335±0.22 | 0.192±0.14 | 33.2 | 26.2 |
| | GU-Net | 0.341±0.20 | 0.231±0.18 | 27.35 | 14.0 |
| | RF method | 0.184±0.146 | 0.117±0.08 | 15.90 | 8.90 |
| B48 | DeepSite | 0.314±0.17 | 0.187±0.12 | 43.52 | 32.94 |
| | Kalasanty | 0.351±0.21 | 0.201±0.14 | 43.53 | 31.76 |
| | GU-Net | 0.382±0.15 | 0.237±0.12 | 33.78 | 14.86 |
| | RF method | 0.221±0.15 | 0.153±0.09 | 28.37 | 11.10 |
| U48 | DeepSite | 0.282±0.2 | 0.165±0.13 | 35.13 | 22.97 |
| | Kalasanty | 0.236±0.22 | 0.127±0.13 | 35.13 | 23.13 |
| | GU-Net | 0.327±0.18 | 0.216±0.12 | 26.56 | 12.5 |
| | RF method | 0.185±0.16 | 0.14±0.09 | 18.75 | 8.25 |

RF: Random forest, MCC: Matthews Correlation Coefficient, DVO: Discretized volumetric overlap, DCA: Distance of the binding site center from any atoms of the ligand, DCC: Distance of the binding site center to ligand center

**Table 5: Comparison of random forest, P2rank, GU-Net, and Kalasanty on Coach42**

| Method | MCC | DVO | Success rate of DCA | Success rate of DCC |
|---|---|---|---|---|
| RF | 0.24±0.12 | 0.13±0.08 | 29.6 | 13.8 |
| P2rank | 0.33±0.25 | 0.19±0.17 | 50.6 | 32.1 |
| Kalasanty | 0.37±0.21 | 0.22±0.13 | 52.8 | 36.4 |
| GU-Net | 0.40±0.23 | 0.25±0.12 | 39.3 | 22.6 |

RF: Random forest, MCC: Matthews correlation coefficient, DVO: Discretized volumetric overlap, DCA: Distance of the binding site center from any atoms of the ligand, DCC: Distance of the binding site center to ligand center

The proposed model compared to RF could significantly improve the number of correctly predicted pockets with more overlap to true binding atoms, as Table 4 shows. We used DeepSite and Kalasanty as CNN-based models to evaluate the performance of the proposed graph model in Table 4. Based on the results, the segmentation models could predict the shape of the pockets more accurate than classification model. The proposed model has bigger MCC and DVO than other models.

This study can be viewed as a practical example of how deep graph methods can be applied to other topics in the structural drug design.

## Financial support and sponsorship

None.

## Conflicts of interest

There are no conflicts of interest.

## References

1. Zhao J, Cao Y, Zhang L. Exploring the computational methods for protein-ligand binding site prediction. Comput Struct Biotechnol J 2020;18:417-26.
2. Krivák R, Hoksza D. P2Rank: Machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. J Cheminform 2018;10:39.
3. Fukunishi Y, Nakamura H. Prediction of ligand-binding sites of proteins by molecular docking calculation for a random ligand library. Protein Sci 2011;20:95-106.
4. Sánchez-Linares I, Pérez-Sánchez H, Cecilia JM, García JM. High-throughput parallel blind virtual screening using BINDSURF. BMC Bioinformatics 2012;13 Suppl 1:S13.
5. Vajda S, Guarnieri F. Characterization of protein-ligand interaction sites using experimental and computational methods. Curr Opin Drug Discov Devel 2006;9:354-62.
6. Roche DB, Brackenridge DA, McGuffin LJ. Proteins and their interacting partners: An introduction to protein-ligand binding site prediction methods. Int J Mol Sci 2015;16:29829-42.
7. Krivák R, Hoksza D. P2Rank: Machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. J Cheminform 2018;10:39.
8. Jiménez J, Doerr S, Martínez-Rosell G, Rose AS, de Fabritiis G. DeepSite: Protein-binding site predictor using 3D-convolutional neural networks. Bioinformatics 2017;33:3036-42.
9. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Development and evaluation of a deep learning model for protein-ligand binding affinity prediction. Bioinformatics 2018;34:3666-74.
10. Liu Q, Wang PS, Zhu C, Gaines BB, Zhu T, Bi J, *et al*. OctSurf: Efficient hierarchical voxel-based molecular surface representation for protein-ligand affinity prediction. J Mol Graph Model 2021;105:107865.
11. Skalic M, Varela-Rial A, Jiménez J, Martínez-Rosell G, de Fabritiis G. LigVoxel: Inpainting binding pockets using 3D-convolutional neural networks. Bioinformatics 2018;35:243-50.
12. Jiang M, Wei Z, Zhang S, Wang S, Wang X, Li Z. FRSite: Protein drug binding site prediction based on faster R-CNN. J Mol Graph Model 2019;93:107454.
13. Mylonas SK, Axenopoulos A, Daras P. DeepSurf: a surface-based

deep learning approach for the prediction of ligand binding sites on proteins. Bioinformatics 2021;37:1681-90.
14. Jiang M, Li Z, Bian Y, Wei Z. A novel protein descriptor for the prediction of drug binding sites. BMC Bioinformatics 2019;20:478.
15. Yan X, Lu Y, Li Z, Wei Q, Gao X. PointSite: A Point Cloud Segmentation Tool for Identification of Protein Ligand Binding Atoms Journal of Chemical Information and Modeling 2022;62:2835-45. DOI: 10.1021/acs.jcim.1c01512.
16. Stepniewska-Dziubinska MM, Zielenkiewicz P, Siedlecki P. Improving detection of protein-ligand binding sites with 3D segmentation. Sci Rep 2020;10:39.
17. Nazem F, Ghasemi F, Fassihi A, Mehri Dehnavi A. 3D U-net: A voxel-based method in binding site prediction of protein structure. J Bioinform Comput Biol 2021;19:2150006.
18. Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation. InInternational Conference on Medical image computing and computer-assisted intervention; Springer, Cham 2015; pp. 234-241.
19. Graham B, van der Maaten L. Submanifold Sparse Convolutional Networks; 2017. p. 1-10.
20. Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering. Adv Neural Inf Process Syst 2016;59:3844-52.
21. Kipf TN, Welling M. Semi-supervised Classification with Graph Convolutional Networks. In: 5th The International Conference on Learning Representations ICLR 2017-Conference Track Proceedings; 2017. p. 1-14.
22. Hamilton W, Ying Z, Leskovec J. Inductive representation learning on large graphs. Adv Neural Inf Process Syst 2017;30:1025-35.
23. Bianchi FM, Grattarola D, Livi L, Alippi C. Graph Neural Networks with Convolutional ARMA Filters. In: IEEE Transactions on Pattern Analysis and Machine Intelligence; 2021.
24. Bianchi FM, Grattarola D, Alippi C. Spectral clustering with graph neural networks for graph pooling. InInternational Conference on Machine Learning 2020 Nov 21 (pp. 874-883). PMLR.
25. Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J. Hierarchical graph representation learning with differentiable pooling. Adv Neural Inf Process Syst 2018;31:4800-10.
26. Gao H, Ji S. Graph U-Nets. 36th International Conference on Machine Learning ICML; 2019. p. 3651-60.
27. Wang L, Huang Y, Hou Y, Zhang S, Shan J. Graph Attention Convolution for Point Cloud Semantic Segmentation. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2019. p. 10288-97.
28. Wu Y, Gao M, Zeng M, Zhang J, Li M. BridgeDPI: a novel Graph Neural Network for predicting drug–protein interactions. Bioinformatics 2022;38:2571-8.
29. Gligorijević V, Renfrew PD, Kosciolek T, Leman JK, Berenberg D, Vatanen T, *et al*. Structure-based protein function prediction using graph convolutional networks. Nat Commun 2021;12:3168.
30. Duvenaud DK, Maclaurin D, Iparraguirre J, Bombarell R, Hirzel T, Aspuru-Guzik A, Adams RP. Convolutional networks on graphs for learning molecular fingerprints. Adv Neural Inf Process Syst 2015;28:1-9.
31. Desaphy J, Bret G, Rognan D, Kellenberger E. Sc-PDB: A 3D-database of ligandable binding sites-10 years on. Nucleic Acids Res 2015;43:D399-404.
32. Chen K, Mizianty MJ, Gao J, Kurgan L. A critical comparative assessment of predictions of protein-binding sites for biologically relevant organic compounds. Structure 2011;19:613-21.
33. Huang B, Schroeder M. LIGSITEcsc: Predicting ligand binding sites using the connolly surface and degree of conservation. BMC Struct Biol 2006;6:19.
34. Zhang Z, Li Y, Lin B, Schroeder M, Huang B. Identification of cavities on protein surface using multiple computational approaches for drug binding site prediction. Bioinformatics 2011;27:2083-8.
35. Roy A, Yang J, Zhang Y. COFACTOR: An accurate comparative algorithm for structure-based protein function annotation. Nucleic Acids Res 2012;40:471-7.
36. Toomer D. Predicting protein functional sites through deep graph convolutional neural networks on atomic point-clouds. 2020. Available online: http://cs230.stanford.edu/projects_winter_2020/reports/32610279.pdf.
37. Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A Comprehensive Survey on Graph Neural Networks. Vol. 32. In: EEE Transactions on Neural Networks and Learning Systems 2021. p. 4-24.
38. Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907. 2016.
39. Grattarola D, Alippi C. Graph Neural Networks in TensorFlow and Keras with Spektral [Application Notes]. IEEE Comput Intell Mag 2021;16:99-106.
40. Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis. IEEE Trans Pattern Anal Mach Intell 2002;24:603-19.