DeepPocket: Ligand Binding Site Detection and Segmentation using 3D Convolutional Neural Networks

Rishal Aggarwal, Akash Gupta, Vineeth Chelur, C.V. Jawahar, and U. Deva Priyakumar*

International Institute of Information Technology, Hyderabad 500 032, India

E-mail: deva@iiit.ac.in

Abstract

A structure-based drug design pipeline involves the development of potential drug molecules or ligands that form stable complexes with a given receptor at its binding site. A prerequisite to this is finding druggable and functionally relevant binding sites on the 3D structure of the protein. Although several methods for detecting binding sites have been developed beforehand, a majority of them surprisingly fail in the identification and ranking of binding sites accurately. The rapid adoption and success of deep learning algorithms in various sections of structural biology beckons the usage of such algorithms for accurate binding site detection. As a combination of geometry based software and deep learning, we report a novel framework, DeepPocket that utilises 3D convolutional neural networks for the rescoring of pockets identified by Fpocket and further segments these identified cavities on the protein surface. Apart from this, we also propose another dataset SC6K containing protein structures submitted in the Protein Data Bank (PDB) from 1st January, 2018 till 28th February, 2020 for ligand binding site (LBS) detection. DeepPocket's results on various binding site datasets and SC6K highlights its better

performance over current state-of-the-art methods and good generalization ability over novel structures.

Introduction

An essential step in the structure-based drug design (SBDD) pipeline is to identify and validate the receptor target.[1] Once the receptor is identified, small molecules are designed such that they can bind well to these targets and exhibit desired pharmocological effect. However, for the rational design of such drug molecules, we need to locate the binding sites of such molecules on the protein structure that are druggable and are functionally relevant.[2] Furthermore, most docking and virtual screening techniques are efficient and are known to perform better with prior knowledge of the binding site.[3] Therefore, predicting locations on the structure of protein where a ligand molecule can bind forms an indispensable step in the drug design process. This requires development of highly accurate in-silico algorithms that can detect ligand binding sites from a given 3D structure of the receptor.

Classical methods that utilise the 3D structure of the protein extract either geometry-based or probe-energy-based features to detect binding sites.[2] Since most ligand binding sites occur in cavities on the 3D structure, geometry-based methods[4–12] are designed to identify these hollow spaces and then rank them based on their binding ability. Fpocket[5] is a widely used geometry-based tool that works with Voronoi tessellation. It uses alpha spheres to detect local curvatures on the protein surface. It then follows a 3-stage process consisting of finding clusters of alpha spheres and ranking them according to their binding site score which is calculated using properties of residual atoms present in each of the pockets. The scoring functions of classical geometry-based methods are usually dependent on custom featurization based on knowledge of binding site properties and therefore are limited in their scoring capacity.

Probe-based methods[13–17] on the other hand place small molecule like probes across the

surface of the structure to identify locations with good binding ability on the structure. FTSite,[18] for example, spreads 16 different probes across specific positions on the grid which are determined by empirical free energy functions. The probe molecules are then clustered according to their types and clusters with favorable interaction energies with the protein residues are predicted as binding sites. Q-SiteFinder is another very successful probe energy method that uses methyl probes and Van der Waals energy to detect binding sites.[19] SILCS[20] uses a fragment based approach by generating probability maps of fragment binding using MD simulations after which Ligand Free energies (LGFEs) are calculated. These methods are highly dependant on the choice of probes, energy functions as well as the state of the protein structure. Such methods may not work well in scenarios where energetically stable sites are not sufficient enough to give accurate predictions.[21] Furthermore, it is difficult to design a set of probes and energy functions that simulate chemical properties that can cover large amounts of small molecules and ligands.[22] Therefore, the design of such methods may lead to biases that could result in inaccurate predictions of binding sites on protein structures.

Template-based methods[23–25] are another class of binding site detection methods that take advantage of the significantly large published databases for protein structures. The basic algorithm involves searching for a similar protein in the database and mapping its binding site to the query protein. FINDSITE[26] is one of the earliest examples of these methods. It identifies a template protein binding to the ligand from the PDB database and overlays the template with the target protein. The binding sites on the template are then ranked for predictions. However, such methods require a large database of protein templates with annotated binding sites and therefore fail if such templates are not available for new protein structures.

In recent years, with the advancement in computer technology and increase in practicable data, the use of data-intensive techniques like machine learning and deep learning has burgeoned in numerous domains.[27–29] Consequently, this has also influenced the field of bio and

cheminformatics greatly by providing solutions to a multitude of problems including binding site prediction.[30–34] The improvement in performance with an increase in accuracy of binding site detection forms substantial evidence to the increasing adoption of ML methods for this problem.

One such method that uses a conventional machine learning algorithm like random-forest(RF) is PRANK.[33] PRANK utilizes Fpocket and Concavity[35] to select pocket points and label them according to their physicochemical properties of the local neighborhood. Then, the RF algorithm assigns a "ligandibility" score to each pocket point that evaluates the pocket's binding ability to a ligand. These scores are then merged into a final pocket score for ranking. PRANK showed that replacing the conventional Fpocket scoring function with their machine learning function led to much better accuracy. P2Rank,[36] a better implementation of PRANK was then developed to serve as a standalone tool.

With the further advancements in artificial intelligence, deep learning has proven to surpass other statistical methods in almost every domain. It allows building such intricate relations from data that are infeasible for traditional machine learning algorithms. Deep learning models stack layers of interconnected neurons based on the principle of hierarchy of concepts which states that complex concepts are learned by building them from simpler ones.[37] These algorithms have been shown to make great strides in computer vision[38,39] and natural language processing.[40] Convolutional neural networks (CNNs),[41] for example, have shown state-of-the-art of performance in image recognition.[38]

Binding site detection can be modelled as a computer vision problem through the voxelization of 3D protein structures. This enables the usage of these CNNs for the same task. DeeplyTough[42] is a method that uses CNN-based siamese networks[43] to compare pockets using euclidean distances by encoding them into descriptor vectors. DeepSite[44] follows a similar approach to P2Rank as they use a CNN to score all points on the protein surface and clusters all points with high scores to generate candidate binding pockets. Kalasanty,[45] on the other hand, passes the entire protein structure through a CNN-based segmentation

model inspired by the U-Net[46] to generate the predicted binding sites in one step. It assigns a probability to each voxel of being part of a pocket. It is shown to perform better than DeepSite in binding site detection.

In this work, we propose rescoring of the pockets detected by a geometry-based software called Fpocket with a CNN. We follow this approach as geometry-based softwares work sufficiently well in identifying cavities on the protein surface. However, their scoring functions usually perform worse than most modern methods at ranking these cavities. Therefore, we supplement Fpocket with a CNN scoring function and show that it outperforms all state-of-the-art methods at identifying and ranking binding sites on the protein structure. Furthermore, we implement a CNN-based segmentation algorithm at high resolution to better indicate sub-locations within a binding pocket that can be targeted for rational drug design. This is further extended to predicting relevant binding residues present at the binding site by taking residues within a certain distance threshold from the predicted mask.

Methods

Our approach

We propose DeepPocket, a novel and comprehensive framework for the efficient detection of binding sites in a 3D structure of a protein. We follow a multi-step approach to get the final pocket location and 3D shape prediction from the input protein structure. We first clean the input structure by removing all hetero-atoms and solvent molecules from the protein structure using the Biopython[47] library. Then, we run Fpocket across the structure and calculate the barycenter of each predicted pocket. These become the candidate pocket centers that need to be ranked by the CNN scoring function. Therefore, constant-sized grids are placed at each barycenter followed by scoring using the CNN. The top-ranked centers are then sent through a CNN segmentation model to get the final pocket structure. The pipeline for our approach is given in Figure 1. More details of the approach are given in the

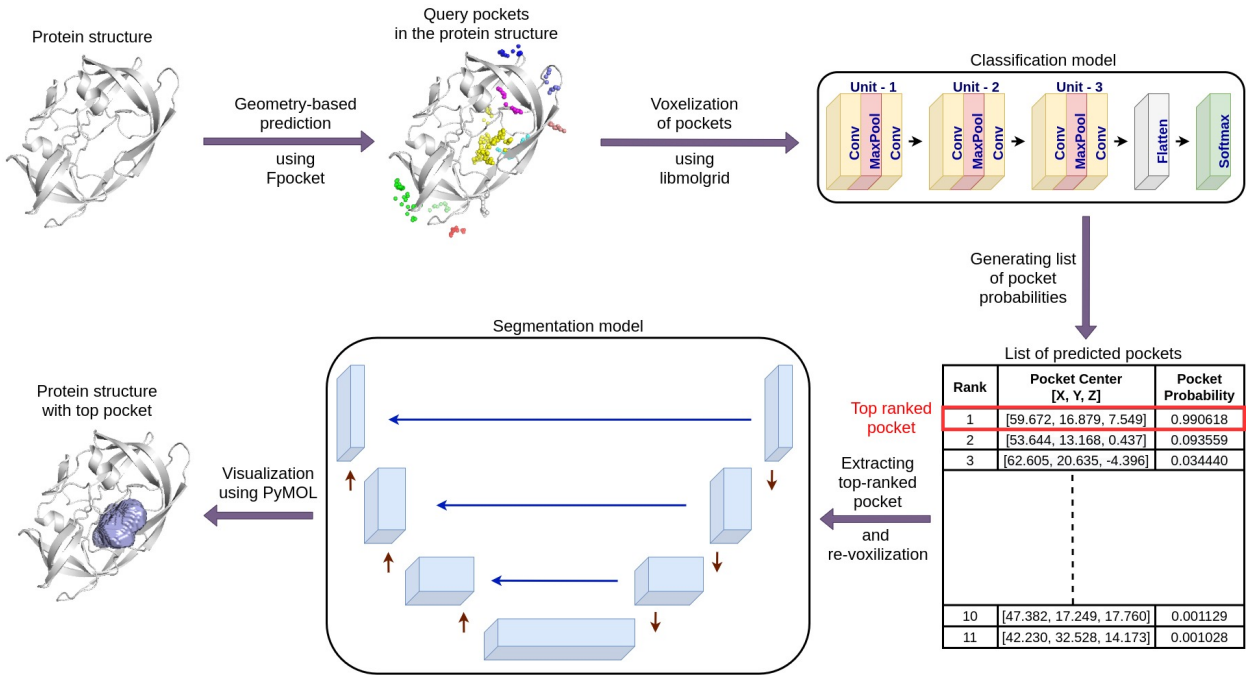


Figure 1: **DeepPocket: Ligand binding site detection using 3D CNNs**: A protein structure taken from Protein Data Bank is passed to the geometry-based software Fpocket, which detects candidate pockets. DeepPocket then uses 3D CNNs to learn the pocket structure and outputs a decreasing list of pockets probabilities with the most druggable pocket at the top. The Top-n pockets are extracted and passed to a CNN-based segmentation module for pocket shape estimation. Protein is shown in "grey", top pocket in "purple" and other colors represent query pockets. Protein and pocket visualizations are generated using PyMOL[48]

following sections.

Datasets and preprocessing

In this work, we used the scPDB v.2017 database[49] for training and cross-validation of the model. It contains 17594 binding sites, which corresponds to 16612 proteins and 5540 UniProt IDs. Different versions of the scPDB dataset have been used previously by other binding site detection methods such as Kalasanty and DeepSite. The proteins, ligands, and 3D shape of binding site structures in the dataset were generated by Volsite.[50] Volsite projects the protein on a 3D grid lattice of resolution 2 Å, after which, each accessible voxel is assigned a pharmacophoric property based on the nearest protein atom. Since each voxel has

a fixed volume, cavities are estimated by counting these pharmacophore-annotated voxels.[51] The dataset was cleaned and split in a similar way as Kalasanty, into 10 cross-validation sets based on their Uniprot IDs to avoid data leakage. We adopted the same data splits for our model training and validation. In this case, 10 separate models are trained where for each model, 9 folds are kept for training and 1 fold for validation. This ensures that we are able to evaluate the model performance on structures not present in the training sets, thereby avoiding data leakage.

To evaluate our model and compare it against existing baselines, we used 4 different external datasets. Two of the datasets, COACH420 and HOLO4k, were first introduced by P2Rank.[36] Alongside, we developed a new dataset called SC6K to test the model on structures recently added to the Protein Data Bank (PDB) database.[52] In addition, we also consider the problem of structured sub-pockets[53] and benchmark the depth of coverage of the binding site provided by our segmentation model on the Refined subset of the PDBbind dataset.[54]

COACH420 and HOLO4k were first used by P2Rank to test their models against multiple state-of-the-art methods. They also curated subsets (called Mlig) of each dataset that they considered contained relevant ligands for binding site detection. We used this subset for our model evaluation. Furthermore, we noticed that in multiple cases the dataset reported some of the standard 20 amino acids as ligands. However, on visualisation of these structures, we realised they were reported as ligands due to poor preparation of the PDB files. Therefore we ignored such ligands in the dataset. The proteins and ligands were separated from the corresponding structure files using the Biopython library[47] and converted into Mol2 format using Openbabel.[55] After separation, any ligands that could not be parsed by RDKit[56] or BioPandas[57] were also removed. The resultant datasets had 291 protein structures and 359 ligands, 3413 protein structures and 4288 ligands for COACH420 and HOLO4k respectively.

To prepare the test sets for segmentation we ran Volsite across the COACH420 and HOLO4k datasets. Due to high buriedness of the ligand molecules in the binding site of some

proteins, Volsite was only able to detect cavities in 207 out of 291 proteins (71.13%) and 2752 out of 3413 proteins (80.63%) for the COACH420 and HOLO4k datasets respectively. This translates to a total of 248 cavities in the COACH420 dataset and 3449 cavities in the HOLO4k dataset. We believe these sets are sufficiently large and diverse to evaluate our segmentation model.

We developed a new dataset, SC6K by taking all the protein structures submitted to the Protein Data Bank from 1st January, 2018 till 28th February, 2020 that contained at least one ligand. These were run through the pdbconv tool from the IChem Toolkit [51] to get a dataset with the same filters and site selection algorithm as scPDB. Furthermore, we removed any protein for which a ligand could not be parsed by RDKit or BioPandas. [57] The final dataset contained 2378 proteins and 6139 structure-ligand pairs.

Refined set is constructed by taking the union of the Refined sets of v2007, v2013, v2015 and v2016 from PDBbind. The dataset consists of 4414 protein-ligand pairs. On running Volsite across this dataset we obtain ground truth segmentation masks for 2793 (63.28%) protein-ligand complexes. These masks were used to obtain the best performing model checkpoint from training for the substructure benchmark.

We used stringent conditions on external datasets to avoid data leakage. Concretely, we removed all proteins from the training set that had either sequence identity greater than 50% or ligand similarity greater than 0.9 and sequence identity greater than 30% to any of the structures in the test set. This ensured that the proteins in the training and test sets neither had similar global structure nor similar binding sites. This resulted in the removal of 2418 structures for COACH420, 7951 structures for HOLO4k, 6285 structures for SC6K and 5801 structures for the Refined set from the scPDB training set.

We ran Fpocket across all datasets to generate data for the classification and segmentation models. Fpocket version 3.0 was used to extract candidate pocket centers for all protein structures. We used Fpocket as it is a geometry-based software that can detect pocket curvatures in most protein structures with high accuracy and therefore has good recall.

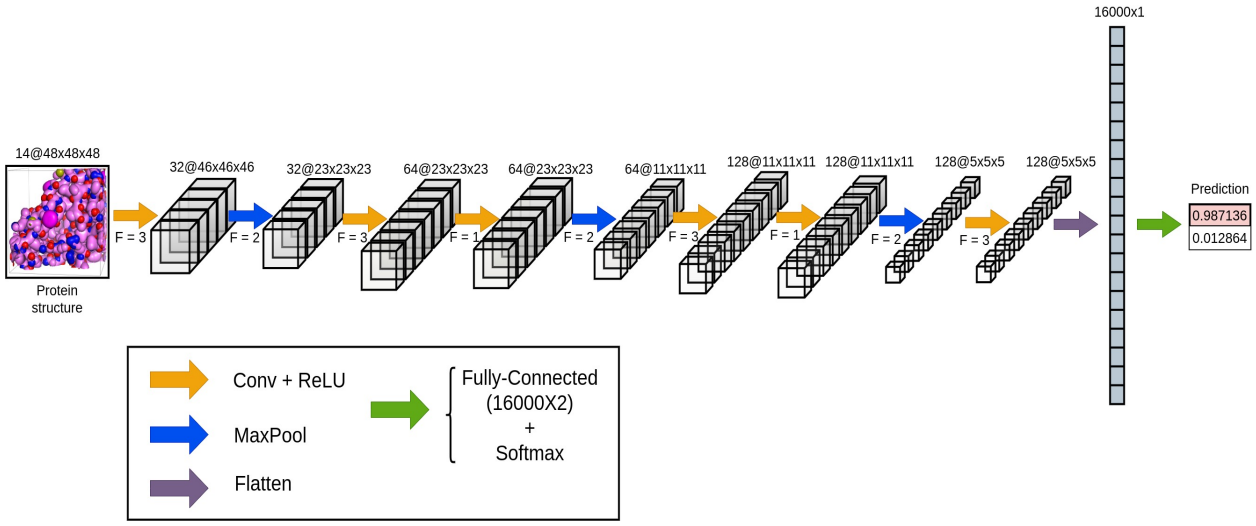


Figure 2: The classification architecture based on 3D convolutional neural networks. The network takes a voxelized 4D image of a pocket present in the protein structure of shape $C@L \times W \times H$ ($C = 14$ is the number of atom-type channels and $L = 48, W = 48, H = 48$ are the dimensions of the bounding box) as input. F represents the kernel size of the operation. Different colors in the input represent atom types like carbon as "pink", nitrogen as "blue", oxygen as "red", and so on. Intermediary shapes (shown in "light grey") follow the same notation as input.

Specifically, Fpocket found pocket centers that were within 4 Å of any ligand atom for 96.4% of ligands in scPDB, 80.78% of ligands in COACH420, 87.62% of ligands in HOLO4k, 91.64% of ligands in SC6K and 95.89% of ligands in Refined set.

Classification of candidate pockets into binding sites

For this task, we processed the dataset using the pocket centers predicted by Fpocket. Any center that was within 4 Å of any ligand atom was marked as positive data point and the rest were marked as negative. Due to the presence of redundant proteins in the scPDB database, we processed the dataset to take account of all the binding sites for a redundant protein and labelled the candidate pocket centers from Fpocket according to the information from all of these binding sites. We had a total of 5,18,460 data points across the cross-validation set, out of which, 22,030 (4.25%) were positive and the rest negative. This data imbalance is handled by oversampling the positive examples such that each batch contains equal amounts

of positive and negative samples while training the model. To voxelize the 3D structure around a pocket center onto a PyTorch tensor we used libmolgrid[58] to generate cubic grids of side 23.5 Å and resolution 0.5 Å. The libmolgrid default receptor atom types were used as grid channels and are stated in the Table S1 of the Supporting Information

We used a simple CNN model for the classification and rescoring of candidate pocket centers. The model is depicted in Figure 2 and the architecture is given in more detail in Table S2 of the Supporting Information. The models were trained using the Adam optimizer with a base learning rate of 0.001 and weight decay of 0.001. We trained the models on a NVIDIA 2080 Ti GPU hardware for 200,000 iterations, with rotational augmentation, and tested the model at every 1000 iterations. Each iteration contained a batch of 50 samples that contained an equal number of positive and negative samples through oversampling. Furthermore, we used the `stratify_receptor` option in Libmolgrid to ensure equivalent sampling of all the receptors during training. The Pytorch learning rate scheduler `ReduceLROnPlateau` was also implemented to reduce the learning rate by a factor of 10 if model performance did not improve after 15 contiguous test intervals.

Segmenting shapes of top ranked binding sites

Positive data points of the classification dataset were used to train the segmentation CNN models. However, the cavities provided by Volsite are at a 2 Å resolution and therefore had to be upsampled to a 0.5 Å resolution. We used Libmolgrid for this by placing binary atoms of radius 1 Å at each cavity point followed by morphological binary dilation.

To train the model we used cubic grids of 0.5 Å resolution with sides of size 32 Å. Our segmentation model is inspired by the U-Net and thus contains encoder and decoder modules with cross-connections as shown in Figure 3. The segmentation model architecture is described in detail in Table S6 of the Supporting Information. For reference, we also visualize an example of input and ground-truth tensor used in training of the segmentation model in Figure S3 and Figure S4 of the Supporting Information respectively. The model was trained

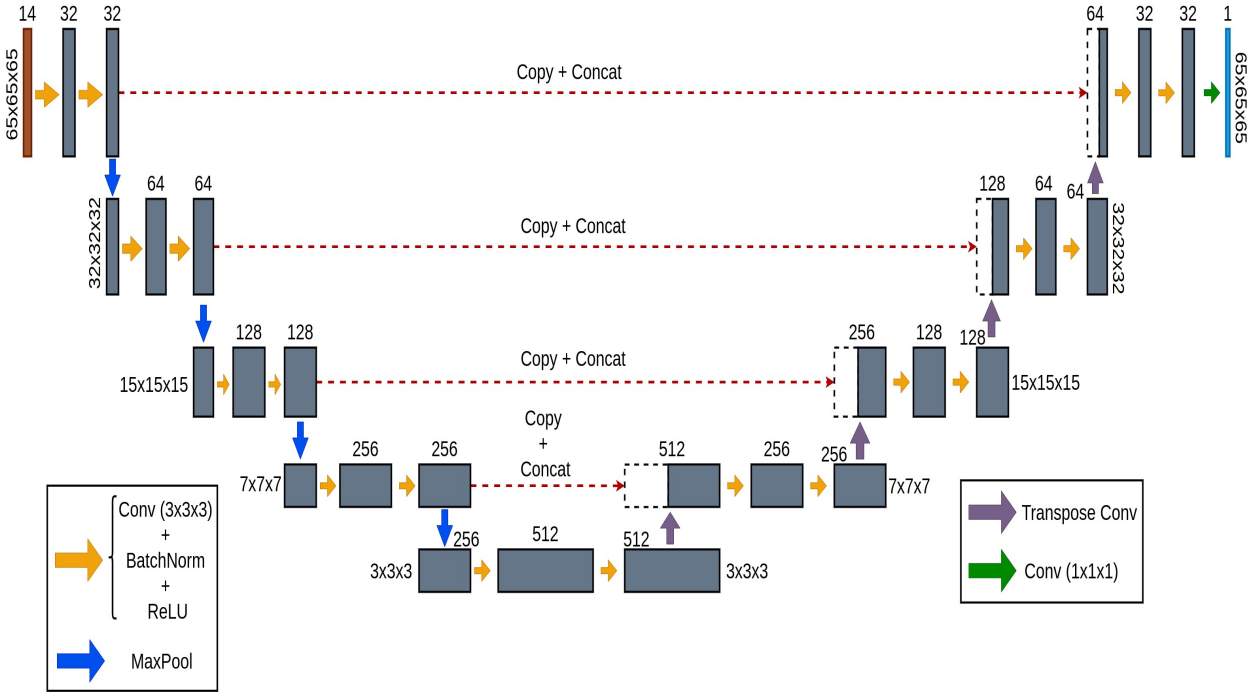


Figure 3: An illustration of the segmentation architecture similar to U-net based on 3D convolutional neural networks. The input (shown in "brown") is a re-voxelized 4D image (with 0.5 Å resolution) of a pocket of shape $14@65 \times 65 \times 65$ which is passed through 4 down and 4 up convolutional units to get a segmentation mask (shown in "cyan") of shape $1@65 \times 65 \times 65$. Intermediary shapes are shown in "grey".

on the NVIDIA 2080 Ti GPU hardware for 200 epochs with rotational and translational augmentation using the same optimizers and hyperparameters as the classification model.

Metrics and evaluation

There are mainly three metrics used to evaluate binding site detection algorithms. These metrics check the algorithm's ability to detect the location and shape of the binding site. The metrics are:

- **Distance to any atom of the ligand (DCA/PPC)** - It is the shortest distance between the predicted center and any atom of the ligand. Predictions with DCA lesser than 4 Å are considered successful.

- **Distance to the center of the binding site (DCC)** - It is the distance between

the predicted center and actual center of the pocket. Predictions with DCC lesser than 4 Å are considered successful.

- **Discretized volume overlap (DVO)** - It is the ratio of the volume of intersection of the predicted and actual shapes to the union of their volumes. We calculate it using the Jaccard index formula

$$DVO = \frac{\#|V_r \cap V_p|}{\#|V_r \cup V_p|} \tag{1}$$

where V_r and V_p are the sets of voxels that fall inside the volume of real and predicted binding pockets respectively.

Classification models were tested by comparing their ranking performance with different baselines according to the DCA criterion which evaluates the model's ability to find the location of the binding site. To evaluate our CNN-based scoring system we followed the ligand-centric approach as this seems to be more suitable in evaluating predictions with more than one ligand binding site (LBS) in a protein.[33] The CNN models output probabilities or scores for all the candidate pockets which are sorted in decreasing order. We then evaluate the ranking capability of the model based on the success rates that the model achieves when we take the top ranked pockets. Top-n corresponds to the success rate score when we take the top "n" ranked unique pockets for a proteins where "n" is the number of annotated binding pockets for that protein. Similarly, Top-(n+1) corresponds to the success rate score when take the top "n+1" ranked unique pockets for a proteins where "n" is the number of annotated binding pockets for that protein. In a similar manner we take the other scores (Top-(n+2),Top-(n+3) ... Top(n+7)) to evaluate the scoring capability of the CNN model. We calculate the success rate % using the below formula:

$$Success\_rate\% = \frac{No.\,of\,correctly\,identified\,pockets}{Total\,number\,of\,pockets} \tag{2}$$

On the other hand, segmentation models were used to elucidate the shape of the binding

pockets from the top-ranked centers. The DCC criterion was used to check if the predicted shape was centered around the true center and the DVO criterion was used to check the level of overlap between the predicted and true shapes. The DVO criterion is only reported for the prediction that pass the DCC criterion (true center is within 4 Å of the center of the predicted shape).

Zhao et. al.[53] designed a metric to evaluate a method's ability in predicting binding site residues given the pocket location. First, the ratio between the intersection of predicted and known binding site residues and the known binding site residues is taken for each pocket. Known binding site residues can be defined as the set of all residues that are within a distance threshold from any ligand atom. The ratio can be represented mathematically as:

$$Ratio = \frac{Predicted\,binding\,residues \cap Known\,binding\,residues}{Known\,binding\,residues} \tag{3}$$

Then, the success rate that a method achieves in getting ratios higher than a given ratio threshold across the dataset is taken. The segmentation model's ability in capturing relevant binding site residues is tested for different distance and ratio thresholds on this benchmark.

Results and Discussion

In order to perform a comprehensive evaluation and test DeepPocket's generalization ability, a 10-fold cross-validation of the classification and segmentation models on the scPDB v.2017 database was performed, followed by testing on the COACH420, HOLO4k, and SC6K datasets. Testing the segmentation models focused on comparing with Kalasanty which is the current state-of-art method in predicting 3D shapes of pockets. Finally, we validate DeepPocket's ability in identifying binding site residues on an established benchmark using the Refined set.

For cross-validation classification experiments returned an average validation accuracy of 0.943 and AUC-ROC of 0.966 from the 10 classification models that were trained on their

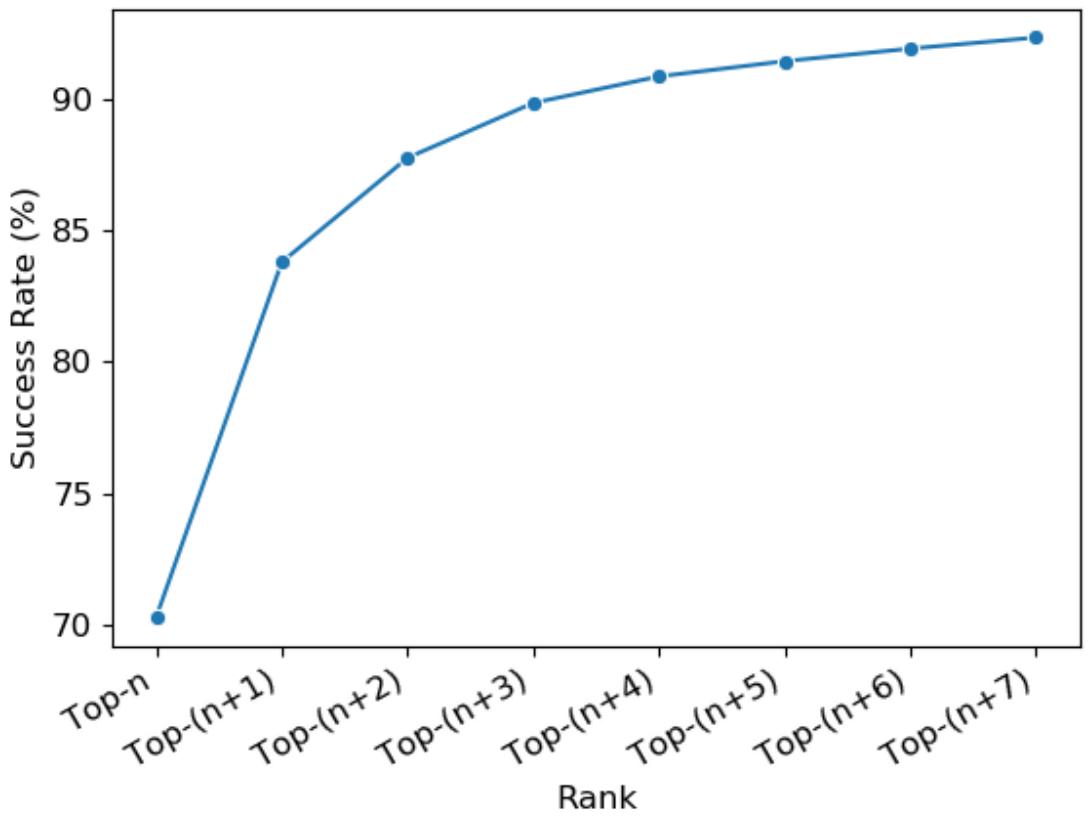
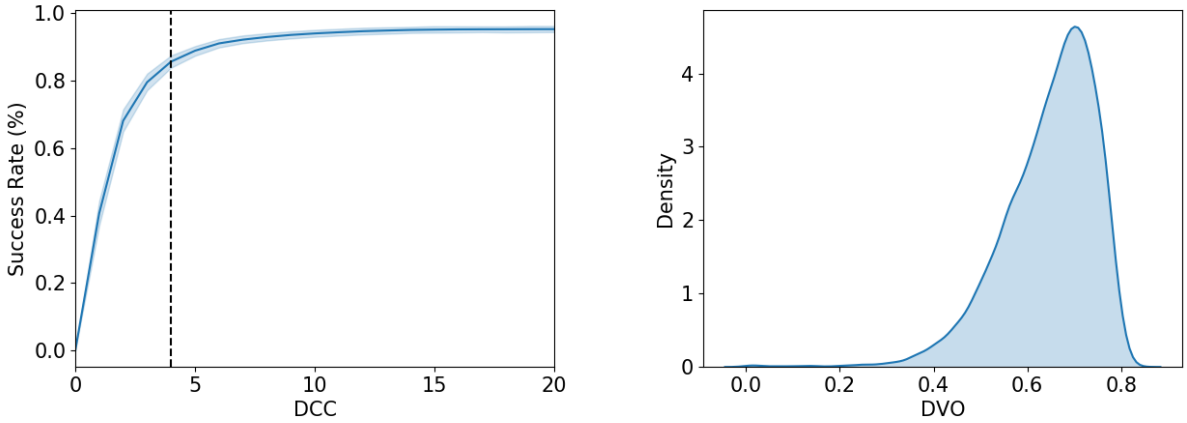


Figure 4: Average success rate on scPDB v.2017 dataset of correctly identifying pockets for ranks ranging from Top-n to Top-(n+7), where n is the number of pockets for a protein

corresponding cross-validation splits. The accuracy and AUC-ROC for each of the folds are reported in Table S3 along with their training graphs in Figure S1 of the Supporting Information. Success/recall rates for all 10 models on their corresponding validation sets obtained an average of 70.27% for Top-n and 87.77% for Top-(n+2). Success rates for all the folds have also been provided in the Table S4 of the Supporting Information. We also plot the success rate from Top-n to Top-(n+7) across the validation set in Figure 4. We believe the big jump of 17% in success rate from Top-n to Top-(n+2) could be an indication of the presence of putative or cryptic binding sites[2] that have not been annotated in the dataset. We also see that most of the pockets in the dataset have been predicted in the Top-(n+7) ranks itself.



(a) (b)

Figure 5: Segmentation results on scPDB v.2017 dataset, (a) success rate vs distance in DCC, (b) kernel density estimate (KDE) vs DVO

Similarly, we followed a 10-fold cross-validation and trained 10 separate segmentation models. To get the segmented pockets, the predicted density was converted into a binary tensor based on a threshold followed by morphological closing and border clearing. Only the largest connected pocket density was retained by erasing the smaller pocket densities in the prediction. We checked threshold values from 0.5 to 0.8 and obtained similar performances, therefore we decided to maintain the threshold at a value of 0.5. The models achieved an average Dice Coefficient of 0.740 and Intersection over union (IOU) of 0.596 across the dataset. The Dice coefficient and IOU for each of the folds are reported in Table S7 along with their training graphs in Figure S2 in the Supporting Information.

We extracted only the correctly predicted pockets in the Top-(n+2) ranks to evaluate the model's performance on the DCC and DVO metrics. If no predicted pocket shapes were returned by DeepPocket for a data point, then values of maximum DCC and minimum DVO were allocated to that prediction. The model returned an average DCC success rate of 85.2% at 4 Å and an average DVO of 0.644 across all cross-validation folds. DCC success rates and average DVO for each fold are provided in the Table S8 of the Supporting Information. Success rate vs distance for DCC and the distribution of DVO across the dataset is depicted in Figure 5. The success rate is plotted along with a 0.95 confidence interval region across

all validation fold. The narrowness of this region indicates that the model performed almost equivalently on all the folds. The DVO curve peaks at a value around 0.65-0.7, which, we believe is a good enough estimate of the shape of the pocket.

Next, we compare the performance of DeepPocket in identifying binding cavities with other state-of-the-art methods on the three test datasets - COACH420, HOLO4k and SC6K. The classification model is compared to the other methods according to the DCA criterion. We note that while we did extensive data leakage removal to train our models for testing, we took pre-trained models for other machine learning and deep learning methods that were trained on datasets that may have some overlap with the test sets. For DeepSite, the results provided by P2Rank[36] for COACH420 and HOLO4k were used. SC6K results were obtained via a python script that submits the PDB files to the `playmolecule` web-service[1], which is a online version of DeepSite. On the other hand, we used the publicly released binary package for P2Rank. For Kalasanty, pocket centers were calculated from predicted densities.

Table 1: DCA results comparison

	COACH420		HOLO4K		SC6K	
	Top-n	Top-(n+2)	Top-n	Top-(n+2)	Top-n	Top-(n+2)
Fpocket	35.09%	51.25%	36.34%	51.53%	23.99%	37.23%
Deepsite	53.07%	53.07%	51.65%	51.67%	52.94%	65.41%
Kalasanty	63.51%	65.18%	61.21%	62.63%	61.75%	61.75%
P2Rank	**68.24%**	75.48%	70.6%	80.05%	62.9%	75.74%
DeepPocket	67.96%	**79.94%**	**73.36%**	**82.97%**	**64.58%**	**83.01%**

The success rate results for Top-n and Top-(n+2) are reported in Table 1. We also report accuracy and AUC-ROC on these test sets in Table S5 of the Supporting Information. In case a method failed to predict pockets for a protein, the pockets for that protein were given a value of maximum DCA to avoid erroneous calculation of the success rate. DeepPocket outperforms all other state-of-the-art methods in all the datasets except in the Top-n score for COACH420, where P2Rank detects only 1 extra pocket therefore beating DeepPocket by just 0.28%. DeepPocket is also the only deep learning method that does not fail to

[1] `https://playmolecule.org/deepsite`

provide pocket locations on any of the proteins in the dataset. It is important to note here that the performance of DeepPocket also depends on the performance of Fpocket. In the COACH420 dataset, Fpocket found pocket centers within 4 Å of any ligand heavy atom for only 80.78% of the binding sites which is much lesser than the other test sets (87.62% for HOLO4k and 91.64% for SC6K). Despite that, DeepPocket has successfully ranked 85% of the Fpocket detected binding sites in the Top-n ranks. We believe this to be the underlying reason behind slightly higher performance of P2Rank in COACH420 Top-n.

Table 2: No. of proteins where the method fails

	COACH420	HOLO4K	SC6K
Fpocket	0	0	0
Deepsite	1	18	206
Kalasanty	12	340	611
P2Rank	0	0	70
DeepPocket	0	0	0

The number of proteins where each method failed to predict pockets for the three datasets are provided in Table 2. We note that the online version of DeepSite has a limitation of only processing protein structures with ≤ 1000 amino acids and therefore failed for larger proteins. P2Rank, on the other hand, failed in parsing some protein PDB files. However, this error only occurs for approximately 1% of our dataset and therefore does not greatly affect the results. Kalasanty, mostly failed due to an absence of predicted pocket densities. This conclusively indicates that DeepPocket can be considered a more accurate and robust method than the rest for binding site detection.

The segmentation algorithm is designed to predict the shape of top-ranked pockets retrieved from the classification scores. Therefore, only the correctly identified pockets in the Top-(n+2) predictions of the three test datasets are used to evaluate the segmentation models. The major difference between DeepPocket and Kalasanty is that while Kalasanty tries to segment out the pockets by taking the entire protein as input, DeepPocket identifies pockets by taking sub-locations around predicted pocket centers as input. This enables DeepPocket to function at a finer resolution of 0.5 Å. Like before, we used the provided pre-trained

Kalasanty model to get its predictions. The final pocket shape and coordinates are obtained from the `pockets.cmap` file that Kalasanty outputs.

Table 3: DCC and DVO results comparison

	COACH420		HOLO4K		SC6K	
	DCC	DVO	DCC	DVO	DCC	DVO
Kalasanty	78.33%	0.43	57.12%	0.43	74.33%	0.35
DeepPocket	**81.31%**	**0.64**	**65.79%**	**0.64**	**80.24%**	**0.62**

The DCC success rate for a 4 Å threshold and average DVO values for both the methods are reported in Table 3. We also report IOU and Dice-Coefficient values on these test sets in Table S9 of the Supporting Information. DeepPocket is seen to have a better performance than Kalasanty for both DCC and DVO criterion. We believe that the major reason for this is the difference in resolutions at which the models operate. A resolution of 0.5 Å, as compared to 2 Å, allows for more fine-grained computation and predictions which in turn leads to better segmentation predictions. DeepPocket achieves an overwhelming DCC success rate of 81.31% on COACH420 and 80.24% on SC6K showing that most of the predicted pocket shapes are centered near the true center of the pocket whereas on the HOLO4K dataset, the success rate is lower (65.79%) but still relatively better. DeepPocket also returns mean DVOs greater than 0.6 when tested on the three datasets indicating good segmentation accuracy.

The success rate curves and DVO distribution plots for all the three datasets are depicted in Figure 6. The success rate plots of DeepPocket on the COACH420 and SC6K datasets are very similar to the success rate plots in the cross-validation experiments indicating good generalisation. For HOLO4k, DeepPocket has a slightly lower success rate performance as compared to the cross-validation set. We deduce that the reason for this is because training on the scPDB dataset does not generalize the DeepPocket segmentation model well to the HOLO4k dataset. This is also evidenced by the lower performance of the pretrained kalasanty model. Furthermore, it is also worthwhile to note that we had to remove 7951 structures from the training set in order to avoid data leakage while testing on the HOLO4k dataset. This lead to a significant reduction in training dataset size for the segmentation model. The

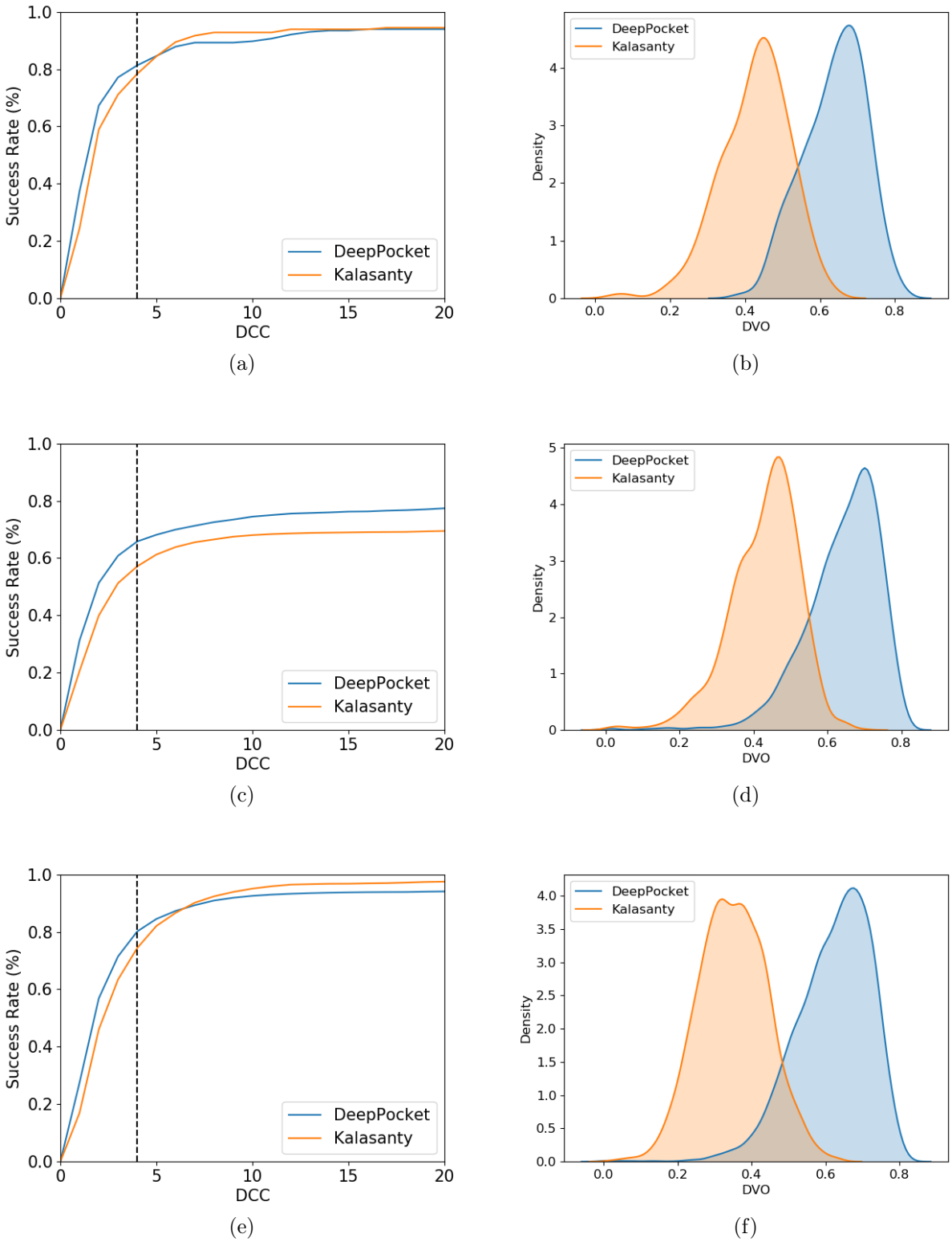


Figure 6: Segmentation results of DeepPocket and Kalasanty on test datasets. Success rate and DVO distributions for (a),(b) COACH420, (c),(d) HOLO4K, and (e),(f) SC6K

returned DVO distributions of the test sets by DeepPocket peak around a value of 0.65. This could be considered a good approximation of the shape of the pocket at a 0.5 Å resolution.

Kalasanty, on the other hand, returns lower DCC success rates at a 4 Å threshold. However, the success rate improves at a good pace with increasing values of the distance threshold. DVO distributions on all the three datasets for Kalasanty peak at values close to 0.35-0.4, indicating that DeepPocket performs better at shape prediction. These results lead to the conclusion that while the Kalasanty method generates densities in the locality of the binding site (Table 1), DeepPocket's segmentation algorithm functions relatively better for elucidating the entire binding pocket shapes.

Finally, we validate the segmentation model's ability in identifying binding site residues on the Refined test set. Zhao et. al.[53] established a benchmark for the identification of functional residues given a binding site. We test our model against this benchamrk, thereby giving a better picture of the depth of coverage of our segmentation model predictions. However, the segementation model is trained to predict a mask present in the void space of a binding site. Therefore, in order to get predicted binding site residues we take all residues within a distance threshold of the predicted mask.

We conduct further experiments to decide a distance threshold. Taking too large of a distance threshold may result in detecting residues not involved in the binding site, whereas, taking too small a distance threshold may result in insufficient coverage of the binding site. Therefore, we take sets of binding site residues that are within different distance thresholds of ligand atoms as well as different distance thresholds of the predicted mask. We then report the intersection over union (IOU) for these sets to reach an ideal value of the mask distance threshold. To get the predicted mask, the candidate Fpocket center closest to the ligand is taken as input for the segmentation model. The average IOU values over the Refined set are shown in Table 4. Ligand distance threshold is represented as ld and mask distance threshold is represented as md.

From Table 4 it is clear that a mask distance threshold of 2.5 Å would give the best IOU

Table 4: Intersection over Union of residues within ligand distance threshold (ld) and mask distance threshold (md)

	md					
ld	1 Å	1.5 Å	2 Å	2.5 Å	3 Å	3.5 Å
3 Å	0.31	0.38	**0.40**	0.39	0.37	0.34
4 Å	0.34	0.42	**0.47**	**0.47**	0.45	0.42
5 Å	0.32	0.41	0.46	**0.48**	**0.48**	0.47

over it's predictions of the binding site. However, at mask distance threshold of 3.5 Å the IOU drops by only 0.01 at ligand distance threshold of 5 Å. Since 3.5 Å mask distance threshold would also give the maximum coverage over the binding site, we benchmark DeepPocket at both mask distance thresholds of 2.5 Å and 3.5 Å. The model success rates at different ligand distance thresholds, mask distance thresholds and ratio cutoffs (Equation 3) across the Refined test set are given in Table 5.

Table 5: Success rate at different ligand distance thresholds (ld), mask distance thresholds of 2.5 Å and 3.5 Å and different ratio cutoffs (r)

	$r = 0.25$			$r = 0.5$			$r = 0.75$		
ld	2.5 Å	3.5 Å	Zhao et. al.	2.5 Å	3.5 Å	Zhao et. al.	2.5 Å	3.5 Å	Zhao et. al.
3 Å	0.91	**0.93**	0.91	0.84	**0.88**	0.86	0.69	**0.79**	0.78
4 Å	0.91	**0.93**	0.91	0.83	**0.88**	0.86	0.63	**0.76**	**0.76**
5 Å	0.9	**0.93**	0.91	0.79	**0.86**	**0.86**	0.5	**0.68**	**0.68**

From Table 5, it is clear that DeepPocket with mask distance threshold of 3.5 Å slightly outperforms the baseline. We believe the main reason for this to be that the model can predict binding site residues on multiple chains as opposed to the baseline. It is important to note, however, that while DeepPocket performs better than the baseline at identifying the residues, it does not provide hierarchical information of the pocket substructure like the baseline.

In Figure 7, we show examples (PDB IDs: 1K2C, 1SQN) of our predicted binding pockets, where the correct center was top-ranked by the classification model. These visualizations are generated by open source molecular visualization system, PyMOL. [48]

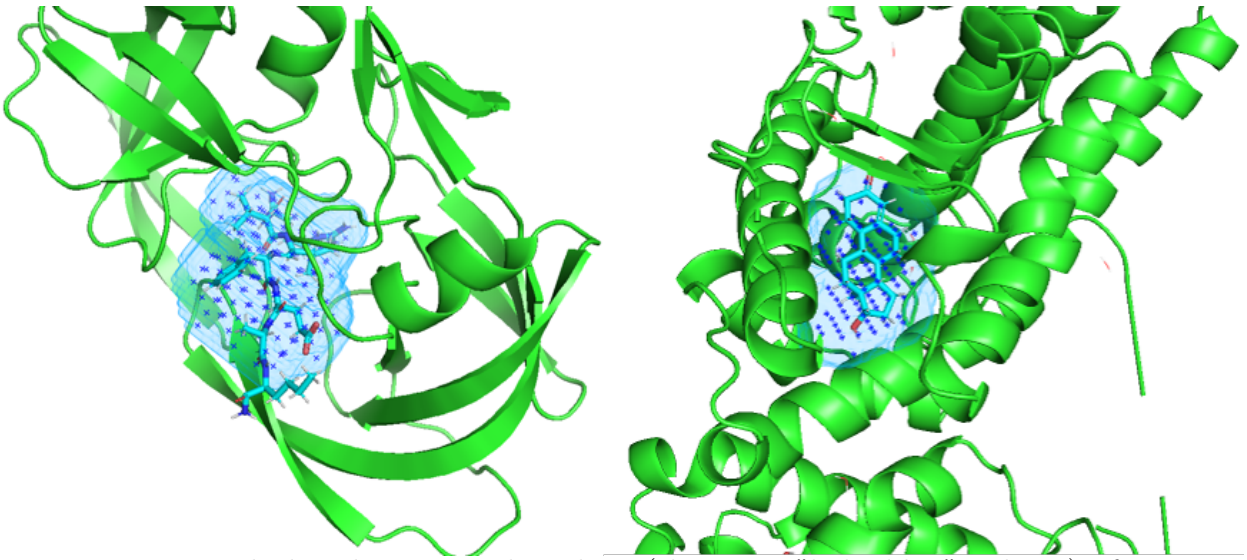


Figure 7: Top-ranked and segmented pockets (shown in "light blue" volume) of proteins HIV-1 Protease (PDB ID: 1K2C) and Progesterone ligand binding domain (PDB ID: 1SQN) by DeepPocket. True pocket annotations are shown as "dark blue" points. Corresponding proteins are show in "green"

Conclusion

In this work, we designed a method DeepPocket that follows a multi-step approach to identify ligand binding sites on the 3D structures of proteins. This mainly involves three steps, extracting candidate pocket centers, ranking them using a classification model, and finally elucidating shapes for the top-ranked centers using a segmentation model. This modular design enables both, the usage of any of these parts independently and their replacement by other methods that may be developed in the future. The classification model uses 3D CNNs on the candidate pockets generated by Fpocket and predicts accurate binding sites. The segmentation model again uses 3D CNNs in a U-net like architecture to elucidate pocket shapes of the predicted binding sites. We made the design choice of working at a fine-grained resolution of 0.5 Å to ensure better performance as evidenced by the DCA, DCC, DVO and Zhao at. al. benchmark results. DeepPocket also has the added advantage of not failing on any of the provided protein structures. Therefore, we believe it would be advantageous to incorporate DeepPocket into structural bioinformatics and drug design pipelines where identification of binding cavities is required. We believe this would especially be useful in

cases where template methods fail to provide the required binding site.

Data and Software Availability

The source code of the method, data and pretrained models are available at `https://github.com/devalab/DeepPocket`

Acknowledgement

The authors thank Manasa Kondamadugu, Bhuvanesh Sridharan and Manan Goel for their comments during the preparation of the manuscript. We thank IHub-Data, International Institute of Information and Technology, Hyderabad for financial support.

Supporting Information Available

Supporting Information contains training results and model architectures of the classification and segmentation models. Furthermore, atom types channels of CNN models and representative figures of segmentation input and ground truth are also provided.

References

(1) Anderson, A. C. The process of structure-based drug design. *Cell Chem. Biol.* **2003**, *10*, 787–797.

(2) Zhao, J.; Cao, Y.; Zhang, L. Exploring the computational methods for protein-ligand binding site prediction. *Comput. Struct. Biotechnol. J.* **2020**, *18*, 417 – 426.

(3) Hassan, N. M.; Alhossary, A. A.; Mu, Y.; Kwoh, C.-K. Protein-ligand blind docking using QuickVina-W with inter-process spatio-temporal integration. *Sci. Rep.* **2017**, *7*, 1–13.

(4) Huang, B.; Schroeder, M. LIGSITEcsc: predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.* **2006**, *6*, 19.

(5) Le Guilloux, V.; Schmidtke, P.; Tuffery, P. Fpocket: An open source platform for ligand pocket detection. *BMC Bioinf.* **2009**, *10*, 168.

(6) Laskowski, R. A. SURFNET: A program for visualizing molecular surfaces, cavities, and intermolecular interactions. *J. Mol. Graphics* **1995**, *13*, 323 – 330.

(7) Hendlich, M.; Rippmann, F.; Barnickel, G. LIGSITE: automatic and efficient detection of potential small molecule-binding sites in proteins. *J. Mol. Graphics Modell.* **1997**, *15*, 359 – 363.

(8) Xie, Z.-R.; Hwang, M. Ligand-binding site prediction using ligand-interacting and binding site-enriched protein triangles. *Bioinformatics* **2012**, *28*, 1579–1585.

(9) Zhu, X.; Xiong, Y.; Kihara, D. Large-scale binding ligand prediction by improved patch-based method Patch-Surfer2.0. *Bioinformatics* **2015**, *31*, 707–713, 25359888[pmid].

(10) Sael, L.; Kihara, D. Binding ligand prediction for proteins using partial matching of local surface patches. *Int. J. Mol. Sci.* **2010**, *11*, 5009–5026, 21614188[pmid].

(11) Sael, L.; Kihara, D. Detecting local ligand-binding site similarity in nonhomologous proteins by surface patch comparison. *Proteins: Struct., Funct., Bioinf.* **2012**, *80*, 1177–1195, 22275074[pmid].

(12) 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 Pharmacol. Sin.* **2020**, *41*, 138–144.

(13) Laurie, A. T. R.; Jackson, R. M. Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. *Bioinformatics* **2005**, *21*, 1908–1916.

(14) Amari, S.; Aizawa, M.; Zhang, J.; Fukuzawa, K.; Mochizuki, Y.; Iwasawa, Y.; Nakata, K.; Chuman, H.; Nakano, T. VISCANA: Visualized Cluster Analysis of Protein Ligand Interaction Based on the ab Initio Fragment Molecular Orbital Method for Virtual Ligand Screening. *J. Chem. Inf. Model.* **2006**, *46*, 221–230.

(15) Lin, Y.; Yoo, S.; Sanchez, R. SiteComp: a server for ligand binding site analysis in protein structures. *Bioinformatics* **2012**, *28*, 1172–1173.

(16) Ghersi, D.; Sanchez, R. Improving accuracy and efficiency of blind protein-ligand docking by focusing on predicted binding sites. *Proteins: Struct., Funct., Bioinf.* **2009**, *74*, 417–424, 18636505[pmid].

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

(18) Ngan, C.-H.; Hall, D. R.; Zerbe, B.; Grove, L. E.; Kozakov, D.; Vajda, S. FTSite: high accuracy detection of ligand binding sites on unbound protein structures. *Bioinformatics* **2011**, *28*, 286–287.

(19) Laurie, A. T.; Jackson, R. M. Q-SiteFinder: an energy-based method for the prediction of protein–ligand binding sites. *Bioinformatics* **2005**, *21*, 1908–1916.

(20) Faller, C. E.; Raman, E. P.; MacKerell, A. D.; Guvench, O. *Fragment-Based Methods in Drug Discovery*; Springer, 2015; pp 75–87.

(21) Tsujikawa, H.; Sato, K.; Wei, C.; Saad, G.; Sumikoshi, K.; Nakamura, S.; Terada, T.; Shimizu, K. Development of a protein–ligand-binding site prediction method based on interaction energy and sequence conservation. *J. Struct. Funct. Genomics* **2016**, *17*, 39–49.

(22) Xie, Z.-R.; Hwang, M.-J. Ligand-binding site prediction using ligand-interacting and binding site-enriched protein triangles. *Bioinformatics* **2012**, *28*, 1579–1585.

(23) Wass, M. N.; Kelley, L. A.; Sternberg, M. J. E. 3DLigandSite: predicting ligand-binding sites using similar structures. *Nucleic Acids Res.* **2010**, *38*, W469–W473.

(24) 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.

(25) Skolnick, J.; Kihara, D.; Zhang, Y. Development and large scale benchmark testing of the PROSPECTOR_3 threading algorithm. *Proteins: Struct., Funct., Bioinf.* **2004**, *56*, 502–518.

(26) Brylinski, M.; Skolnick, J. A threading-based method (FINDSITE) for ligand-binding site prediction and functional annotation. *Proc. Natl. Acad. Sci. U. S. A.* **2008**, *105*, 129–134.

(27) Mater, A. C.; Coote, M. L. Deep Learning in Chemistry. *J. Chem. Inf. Model.* **2019**, *59*, 2545–2559.

(28) Pathak, Y.; Mehta, S.; Priyakumar, U. D. Learning Atomic Interactions through Solvation Free Energy Prediction Using Graph Neural Networks. *J. Chem. Inf. Model.* **2021**, *61*, 689–698, PMID: 33546556.

(29) Pattnaik, P.; Raghunathan, S.; Kalluri, T.; Bhimalapuram, P.; Jawahar, C. V.; Priyakumar, U. D. Machine Learning for Accurate Force Calculations in Molecular Dynamics Simulations. *J. Phys. Chem. A* **2020**, *124*, 6954–6967, PMID: 32786995.

(30) Chauhan, J. S.; Mishra, N. K.; Raghava, G. P. Identification of ATP binding residues of a protein from its primary sequence. *BMC Bioinf.* **2009**, *10*, 434.

(31) Chen, K.; Mizianty, M. J.; Kurgan, L. Prediction and analysis of nucleotide-binding residues using sequence and sequence-derived structural descriptors. *Bioinformatics* **2011**, *28*, 331–341.

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

(33) Krivák, R.; Hoksza, D. Improving protein-ligand binding site prediction accuracy by classification of inner pocket points using local features. *J. Cheminf.* **2015**, *7*, 12.

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

(35) Chen, K.; Mizianty, M.; Gao, J.; Kurgan, L. A Critical Comparative Assessment of Predictions of Protein-Binding Sites for Biologically Relevant Organic Compounds. *Structure* **2011**, *19*, 613–621.

(36) Krivák, R.; Hoksza, D. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *J. Cheminf.* **2018**, *10*, 1–12.

(37) Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press, 2016; `http://www.deeplearningbook.org`.

(38) 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.

(39) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems* **2014**, *27*.

(40) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. Advances in neural information processing systems. 2017; pp 5998–6008.

(41) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems. 2012; pp 1097–1105.

(42) Simonovsky, M.; Meyers, J. DeeplyTough: learning structural comparison of protein binding sites. *J. Chem. Inf. Model.* **2020**, *60*, 2356–2366.

(43) Koch, G. R. Siamese Neural Networks for One-Shot Image Recognition. 2015.

(44) 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.

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

(46) Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. Cham, 2015; pp 234–241.

(47) Cock, P. J.; Antao, T.; Chang, J. T.; Chapman, B. A.; Cox, C. J.; Dalke, A.; Friedberg, I.; Hamelryck, T.; Kauff, F.; Wilczynski, B.; Hoon, M. J. L. d. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **2009**, *25*, 1422–1423.

(48) DeLano, W. L. Pymol: An open-source molecular graphics tool. *CCP4 Newsletter on protein crystallography* **2002**, *40*, 82–92.

(49) 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–D404.

(50) Desaphy, J.; Azdimousa, K.; Kellenberger, E.; Rognan, D. Comparison and Druggability Prediction of Protein–Ligand Binding Sites from Pharmacophore-Annotated Cavity Shapes. *J. Chem. Inf. Model.* **2012**, *52*, 2287–2299.

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

(52) Bernstein, F. C.; Koetzle, T. F.; Williams, G. J.; Meyer Jr, E. F.; Brice, M. D.; Rodgers, J. R.; Kennard, O.; Shimanouchi, T.; Tasumi, M. The Protein Data Bank: a computer-based archival file for macromolecular structures. *J Mol Biol.* **1977**, *112*, 535–542.

(53) Zhao, R.; Cang, Z.; Tong, Y.; Wei, G.-W. Protein pocket detection via convex hull surface evolution and associated Reeb graph. *Bioinformatics* **2018**, *34*, i830–i837.

(54) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind database: methodologies and updates. *J. Med. Chem.* **2005**, *48*, 4111–4119.

(55) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An open chemical toolbox. *J. Cheminf.* **2011**, *3*, 1–14.

(56) Landrum, G. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling. 2013.

(57) Raschka, S. BioPandas: Working with molecular structures in pandas DataFrames. *Journal of Open Source Software* **2017**, *2*, 279.

(58) Sunseri, J.; Koes, D. R. libmolgrid: Graphics Processing Unit Accelerated Molecular Gridding for Deep Learning Applications. *J. Chem. Inf. Model.* **2020**, *60*, 1079–1084.

Graphical TOC Entry