PocketAnchor: Learning Structure-based Pocket Representations for Protein-Ligand Interaction Prediction

Shuya Li
Institute for Interdisciplinary Information Sciences, Tsinghua University
Tingzhong Tian
Tsinghua University https://orcid.org/0000-0003-4899-0632
Ziting Zhang
Tsinghua University
Ziheng Zou
Silexon AI Technology
Dan Zhao
Tsinghua University https://orcid.org/0000-0003-0195-6031
Jianyang Zeng (zengjy321@tsinghua.edu.cn)
Tsinghua University https://orcid.org/0000-0003-0950-7716

Article

Keywords:

Posted Date: May 2nd, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1583468/v1

PocketAnchor: Learning Structure-based Pocket Representations for Protein-Ligand Interaction Prediction

Shuya Li[1,†], Tingzhong Tian[1,†], Ziting Zhang[2,3], Ziheng Zou[4], Dan Zhao[1,*] and Jianyang Zeng[1,*]

[1]Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China.
[2]Department of Automation, Tsinghua University, Beijing, China.
[3]MOE Key Laboratory of Bioinformatics, Tsinghua University, Beijing, China.
[4]Silexon AI Technology Co., Ltd., Nanjing, Jiangsu Province, China.
[†]These authors contributed equally.
[*]Corresponding authors: Dan Zhao, zhaodan2018@tsinghua.edu.cn,
and Jianyang Zeng, zengjy321@tsinghua.edu.cn.

Abstract

Modeling and predicting protein-ligand interactions have a wide range of applications in drug discovery and biological research. Appropriate and effective protein feature representations are of vital importance for developing computational approaches, especially data-driven methods, for predicting protein-ligand interactions. However, existing sequence-based protein representation methods often fail to explicitly learn the spatial features of proteins, while current structure-based methods do not fully investigate the ligand-occupying regions in protein pockets. In this work, we propose a novel structure-based protein representation method, named PocketAnchor, for capturing the local environmental and spatial features of protein pockets to facilitate protein-ligand interaction-related learning tasks. We define "anchors" as probe points reaching into the cavities and those located near the surface of proteins, and we design a specific message passing strategy for gathering local information from the atoms and surface neighboring these anchor points. Comprehensive evaluation of our method demonstrated that it can be successfully applied to detect the ligand binding sites on a protein surface and greatly outperform existing baseline methods. Our anchor-based model also achieved state-of-the-art performance in the protein-ligand binding affinity prediction task and exhibited great generalization ability for novel proteins. Further analyses illustrated that the anchor features learned by PocketAnchor can successfully capture the geometric and chemical properties of subpockets. In summary, our anchor-based approach can provide effective protein feature representations for developing computational methods to improve the prediction of protein-ligand interactions.

1 Introduction

Protein-ligand interactions are the molecular basis of many essential cellular activities, such as signal transduction, gene regulation, and metabolism [1]. Prediction and characterization of such interactions are important for understanding the biological functions of proteins and developing therapeutic agents against pathological protein targets [2, 3]. Despite the fact that many experimental techniques have been developed for measuring and analyzing protein-ligand interactions [4], there is a growing trend towards developing computational methods for solving this problem because of their advantages in terms of cost, speed, and scalability [5].

Although knowledge-based computer-aided drug design (CADD) approaches such as molecular docking methods have been applied to model protein-ligand interactions for decades [6, 7, 8, 9], emerging data-driven methods have also shown great advantages in solving such problems, mainly due to their high accuracy and speed [10]. In particular, an increasing number of machine learning and deep learning-based methods have been proposed to address different issues related to protein-ligand interactions, such as pocket detection (or binding site prediction) [11, 12, 13, 14], pocket classification [15], protein-ligand complex scoring [16, 17, 18, 19], binding affinity prediction (which is mainly used for virtual screening) [20, 21, 22, 23], and non-covalent interaction prediction [23]. For these computational methods, representing the molecular features appropriately is one of the

key steps towards obtaining satisfactory performance. Small-molecule ligands can be efficiently represented by Morgan fingerprints, simplified molecular-input line-entry system (SMILES) strings, or graphs [24]. In comparison, proteins generally have larger sizes and more complex spatial structures, which makes it more challenging to design effective feature representations.

Existing methods for representing protein features can be classified into two main categories, i.e., sequence-based and structure-based schemes. For sequence-based methods, the amino acid sequences of proteins are typically encoded by k-mers (i.e., fragments of length k), one-hot encodings, and matrices containing evolutionary information from the blocks substitution matrix (BLOSUM) or position-specific scoring matrix (PSSM) [25, 26, 27]. Machine learning techniques, including representation learning, convolutional neural networks (CNNs), recurrent neural networks (RNNs), and attention-based methods, can then be used to extract the intrinsic features of protein sequences [25, 26, 27]. Protein structures, on the other hand, contain more information about the spatial organization of the amino acid sequences, and thus are more directly associated with the corresponding biological functions and ligand binding properties. Structure-based methods often encode proteins as contact maps (or distance maps), surface meshes, three-dimensional (3D) voxels, 3D points, or graphs containing spatial information [28, 29, 30, 31]. Correspondingly, 2D and 3D CNNs, point cloud-based methods, and graph neural networks (GNNs) have been successfully applied to learn the structure-based feature embeddings of proteins [28, 29, 30, 31]. For example, DeeplyTough, a 3D CNN-based method for learning the feature embeddings of binding pockets, has been successfully applied in pocket matching [32]. MaSIF, which represents the protein surface as meshes of triangles and calculates the chemical and geometric features of the protein surface, has shown superior performance in protein-protein interaction prediction and binding site classification tasks [31].

Although these protein feature representation methods can learn useful embeddings for several prediction tasks, they still have certain limitations when applied to prediction tasks related to protein-ligand interactions. For example, the sequence-based representations generally fail to describe the ligand binding pockets explicitly and ignore the informative 3D protein structures. On the other hand, existing structure-based methods focusing on the spatial arrangements of either protein atoms or the surface generally do not explicitly profile the local spatial regions within the protein pockets, whose environmental properties can actually affect the ligand-binding behaviors of the proteins directly.

In this paper, we propose a novel 3D structure-based protein representation method, named PocketAnchor, for addressing protein-ligand interaction prediction problems. Our method employs anchor-based protein feature representations, in which "anchors" are defined to represent the locations and features of the potential ligand-occupying regions. This method for the *first* time learns the substructure-level feature representations of protein pockets in an end-to-end manner. We design a new information aggregation strategy for anchor-based protein feature representations, in which neighboring messages from both protein surface and atom features are integrated into environmental feature representations of protein pockets. Our method demonstrates superior performance and better generalization ability over state-of-the-art baseline methods in predicting ligand binding sites and protein-ligand binding affinities.

2 Results

2.1 PocketAnchor learns the subpocket-level features of protein pockets

A small-molecule ligand can bind to specific surface regions of its protein partners, which are called protein pockets or binding sites. The protein pockets generally have different characteristics, including different geometric and chemical properties, compared with other non-pocket regions of the protein surface. These properties, which are determined by local protein substructures, can impact ligand binding. For example, the hydrogen-bond donors/acceptors in the pockets can interact with the corresponding acceptor/donor partners in the ligands, while the hydrophobic regions of pockets tend to interact with the hydrophobic functional groups of ligands. Therefore, effectively representing the features of protein pockets is a critical and fundamental step in protein-ligand interaction prediction.

We use an example to intuitively compare several feature representation approaches for protein pockets. As shown in Figure 1**a**, a binding pocket in the ALK kinase domain interacts with the small-molecule ligand entrectinib (PDB ID: 5FTO). Note that the 3,5-difluorobenzyl moiety of the ligand interacts with two residues (i.e., 1127F and 1256L) of ALK [33], which are colored in blue and red, respectively. Sequence-based algorithms are generally used to model the amino acid

sequence of the protein [21, 22, 23], which in this case the distance from 1127F to 1256L covers 129 residues (Figure 1**b**). Capturing these kinds of long-range and indirect associations can be quite challenging when using sequence-based methods. Surface-based feature representation [31], on the other hand, focuses on the protein surface and usually converts the protein into meshes. Although the straight-line distance between 1127F and 1256L is only 8 Å, the geodesic distance along the protein surface is about 15 Å, as shown in Figure 1**c**. In such a scenario, it may also be difficult for the surface-based methods to directly model the collaborative effects of these two residues in the subpocket. Similarly, other structure-based feature representations (e.g., 3D grid or point cloud) [15, 16, 29, 30] may face the same challenge, as they do not represent the interspace regions of the subpocket explicitly, which contain information important for ligand binding.

To fill this gap, we propose using an anchor-based method for learning the feature representations of protein pockets, named PocketAnchor (Figure 1**d**), to better model the local 3D environments of protein pockets. More specifically, we introduce imaginary points named anchors to probe into every potential ligand binding region of proteins to directly bridge the components of proteins that are spatially associated, but remote for each other along the sequence or surface. In this example, the 1127F and 1256L residues can both contribute to the same nearby anchor, resulting in the anchor being an effective feature descriptor representing the properties of the subpocket region.

Next, we describe how to gather and represent the environmental information for each anchor point using our PocketAnchor module, an anchor-based protein feature encoder (Figure 1**e** and Methods). First, anchors are generated by sampling and clustering points in a specific region of the protein pocket or the nearby protein surface (more details can be found in Methods). Then, the features of the generated anchors are learned through the PocketAnchor module. In particular, each anchor receives messages from nearby protein atoms and surface vertices within a radius of 6 Å through three steps of message passing and aggregation. First, a typical message passing operation is performed among protein atoms, that is, each atom receives messages from its neighboring atoms to update its features. Here, two atoms are considered neighboring atoms if they are connected by a chemical bond. Then, the surface vertices containing geometric and chemical characteristics (calculated using MaSIF [31]) collect messages from adjacent vertices to update the corresponding vertex features. The adjacent vertices are linked by edges of the surface mesh, which is also calculated using MaSIF [31]. After several iterations of atom and vertex feature updating, the updated features are aggregated into nearby anchors and the features of these anchors are obtained. More details about the PocketAnchor module can be found in Methods. In the remaining part of this paper, we will introduce two applications of our anchor-based protein representation method, namely protein-ligand binding site prediction and binding affinity prediction.

2.2 The anchor-based model PocketAnchor-site accurately identifies ligand binding sites

Ligand binding sites or binding pockets are defined as the locations on the protein surface where ligands can bind to. Identifying the binding sites of a protein is essential for designing potential drugs that can activate or inhibit the functional activities of the protein. In this work, we propose using an anchor-based model, named PocketAnchor-site (Figure 2**a** and Methods), to accurately recognize the specific regions of binding pockets on the surface of a whole protein. More specifically, given a protein structure as shown in Figure 2**a**, anchors covering all the regions near the surface of the protein are generated. Then, the anchor features are extracted by the PocketAnchor module and scored using an extra ligand binding site prediction module (Methods). Finally, the predicted binding pockets are defined by clustering those anchors with high prediction scores. More details about the PocketAnchor-site model can be found in Methods.

Two benchmark datasets (i.e., COACH420 [34] and HOLO4k [35]) were used to evaluate the performances of our model and baseline methods on the binding site prediction task. The scPDB v2017 dataset [36] (the training data used in DeepSurf [13]), which consisted of 9,444 training samples after excluding the proteins homologous to those in the test set, was used as training data. We mainly used the DCC (i.e., distance from the predicted pocket center to its nearest ligand center) and DCA (i.e., distance from the predicted pocket center to its nearest ligand atom) as the evaluation metrics (Figure 2**b**); these metrics have been widely used for evaluating binding site prediction methods [12, 13, 14]. The DCC- or DCA-based success rate is then defined as the proportion of successfully predicted binding pockets (i.e., DCC or DCA < 4 Å) among all the true pockets. As in previous studies [12, 13, 14], the success rates based on the top-n and top-$(n+2)$ predicted binding pockets were both evaluated, where n is the number of true pockets in each sample.

We compared PocketAnchor-site with several state-of-the-art baseline methods. Two of these methods, DeepSite [12] and DeepSurf [13], are 3D-CNN-based models that are specifically designed for grid-based protein feature representation, while another, P2Rank [14], uses a random forest classifier that mainly takes the descriptors of the solvent accessible surface as input. As shown in Figure 2**c**, our PocketAnchor-site model achieved the best performance on both datasets according to the DCC-based success rates, and it achieved the best DCA-based success rates on the COACH420 dataset and the best comparative DCA-based results on the HOLO4k dataset. According to its definition, DCC is a stricter metric than DCA, and our model achieved 9.7% and 7.6% increases in the success rate defined by DCC-(n+2) over the best baseline method on the COACH420 and HOLO4k datasets, respectively.

To illustrate the contributions of the two sources of information employed by our model, i.e., the protein atom features and protein surface features, we conducted an ablation study in which the model performance was evaluated when using only one source of information. As shown in Figure 2**d**, the success rates dropped when removing the features from either protein atoms or surface vertices, indicating the importance of both sources for predicting binding sites using our PocketAnchor-site model.

We also noticed that not all pockets of a protein were occupied by ligands, resulting in potential missing labels in the benchmark datasets. Through a case analysis, we observed that our model found additional binding sites that were not labeled in the benchmark dataset. Figure 2**e** shows the prediction results for the ricin protein, in which two pockets (colored in red) were predicted by our PocketAnchor-site model. One pocket was the ligand binding site originally labeled in the COACH420 dataset (ligand colored in blue). Although the other pocket was not labeled, it is also a true binding site (ligand colored in green), as reported in [37]. Manual inspection found that this was not the only case in which a true binding site that was not originally labeled in the benchmark datasets was identified by PocketAnchor-site, indicating that the reported performance may underestimate the true success rate.

To examine the potential factors that may affect the performance of our model, we divided all the test samples in the HOLO4k dataset into two groups according to the DCC-(n+2) metric with a threshold of 4 Å, and compared the distributions of several protein- or ligand-related properties between those two groups of samples (Figure 2**f**). For each test protein, similarity to the training proteins did not have much effect on the performance, indicating that the model was not overfitted to similar proteins. As the true pockets in the two benchmark datasets were defined by the locations of observed ligands in the protein structures, we also analyzed the relationship between the ligand properties and model performance. Predictions of pockets with smaller ligands (molecular weight < 200) were more likely to fail (i.e., DCC-$(n+2) > 4$ Å), which suggested that the features of small and shallow pockets were more difficult to capture. Further more, the logP (the logarithm of octanol-water partition coefficient) of ligands seemed to have little effect on model performance. In addition, ligands in the successfully predicted pockets (i.e., those with DCC-$(n+2) < 4$ Å) tended to have smaller B factors in the crystal structures, which may indicate that the pockets with more stably bound ligands were easier to detect.

In conclusion, our anchor-based method achieved the best performance in detecting the ligand binding sites of novel proteins, and is thus a useful tool for identifying potential ligand-binding pockets in structure-based drug design, especially for protein targets without known protein-ligand complex structures.

2.3 The anchor-based model PocketAnchor-affinity generalizes well to novel proteins in protein-ligand binding affinity prediction

Predicting the binding affinities of protein-ligand pairs is a fundamental problem in drug discovery. In particular, binding affinities can be quantified by several affinity or activity measurements including dissociation constant (Kd), inhibition constant (Ki), and half-maximum inhibitory concentration (IC50). Structure-based molecular docking methods have been widely used to predict protein-ligand binding affinities [7, 8, 9]. However, they are limited by the accuracy of the underlying energy functions used for modeling and often require tremendous computational resources. Recent advances in deep learning techniques have enabled and promoted the development of protein-ligand binding affinity prediction models. Yet most of them require high-quality structures of protein-ligand co-complexes as input [16, 18, 19], thus limiting their application. On the contrary, a number of sequence-based deep learning methods taking the protein sequence and ligand structure as separate inputs have exhibited satisfactory performance [21, 22, 23, 38]. However, there is a significant drop in performance when the test proteins are not seen during training [23],

indicating that the current protein representation methods may not generalize well to novel proteins.

We speculate that our anchor-based representation method could directly extract the rich structural information from protein pockets, and thus help alleviate the current generalization issue. In this work, we design an anchor-based model, named PocketAnchor-affinity, for predicting protein-ligand binding affinities (Figure 3**a** and Methods). More specifically, given a protein pocket, anchors covering the potential ligand binding regions in the pocket were first generated. The anchor and ligand features were then extracted by the PocketAnchor module and a ligand encoder module, respectively. Finally, the protein-ligand binding affinities were predicted through a binding affinity prediction module (More details can be found in Methods).

To thoroughly evaluate the prediction performance as well as the generalization ability of our binding affinity prediction model, we designed three comprehensive evaluation scenarios with different train-test splitting schemes (Figure 3**b**). In the first splitting scheme, named original CASF split, the core set of PDBbind v2016 was used as the test set (the same test set as in the CASF-2016 benchmark [7]), which contained 285 compound-protein pairs related to 57 protein families, and the general set of PDBbind v2016 was used as the training set. The original CASF split cannot be applied to evaluate the generalization ability of the data-driven models, because proteins that were the same as or similar to those in the test set were also included in the training data, and we thus cannot examine the model performance on novel proteins. We employed the original CASF split because it has been widely applied for evaluating the machine learning-based methods on this task [16, 18, 19], and it can also serve as a baseline for comparing with the two additional splitting schemes. To design new evaluation scenarios especially for generalization ability evaluation, we employed a hierarchical clustering algorithm (the same one reported in [23]) to cluster all the proteins in the dataset, and the training proteins located in the same clusters as the test proteins were all grouped into a subset named "CASF-similar". Compared with the remaining training proteins, the "CASF-similar" subset exhibited significantly higher similarity with the test proteins (i.e., proteins in the CASF-2016 set) as shown in Figure 3**c**. Through visualization of the training and test proteins (Figure 3**d**), it was also obvious that the "CASF-similar" subset clustered with the test proteins, while the remaining training proteins were relatively well separated from both the test and "CASF-similar" proteins. Therefore, we introduced a new-protein split by removing the "CASF-similar" subset from the training data. The training data in this new-protein split were derived from the PDBbind v2020 general set. In addition, a third splitting scheme named expanded CASF was introduced by also including the CASF-similar subset in the test set to take full advantage of samples in the dataset. This expanded set contained 4916 test samples, which was much more than the 285 test samples in the original CASF and new-protein CASF splits. The three splitting schemes are illustrated in Figure 3**b**.

We compared PocketAnchor-affinity with several state-of-the-art baseline methods for predicting protein-ligand binding affinities (Figure 3**e**). These baseline methods, namely DeepDTA [21], GraphDTA [22], and MONN [23], mainly employ SMILES or graph representations for ligands and sequence-based representations for proteins. Using the original CASF split, which was expected to be the easiest in terms of making predictions, as most of the tested proteins were also in the training data, almost all the methods achieved relatively high Pearson's correlation coefficients (PCCs). GraphDTA, MONN, and PocketAnchor-affinity exhibited comparable results with PCCs above 0.7. However, we observed significant decreases in performance using the other two splits for all the prediction methods. Although MONN achieved the best PCC (0.781) on the original CASF split, its performances on the new-protein and expanded CASF splits were only 0.615 and 0.536, with a decrease of 0.166 and 0.245, respectively.

Making predictions using the new-protein and expanded CASF splits was generally more challenging compared with the original CASF split, and these two splits were better for evaluating the generalization ability of models required in practical scenarios. The best performances on the new-protein and expanded CASF splits were achieved by PocketAnchor-affinity, with PCCs of 0.675 and 0.588, respectively (Figure 3**e**). PocketAnchor-affinity also exhibited the smallest performance decrease from the original CASF split. To compare our method with the traditional molecular docking methods, we also tested the performances of docking scoring functions on the CASF-2016 test set from [7]. The PCCs achieved by these docking scoring functions ranged from 0.21 to 0.63 (except for Δ_{Vina}RF20, which was trained on data that included about 50% of the test samples), which were lower than those achieved using our method (Figure 3**e**). The ablation study demonstrated that removing features from either protein atoms or the surface slightly impaired the performance of our model (Figure 3**f**).

The results indicate that, compared with most data-driven protein-ligand affinity prediction models, which suffer from decreased performance when applied to novel proteins, our anchor-

based model has a much better generalization capacity in practical scenarios. This suggests that our model generalizes well and can be potentially applied in real-world drug discovery scenarios for first-in-class protein targets.

2.4 The subpocket-level representations learned by PocketAnchor are associated with protein and ligand properties

In the previous sections, we demonstrated that our anchor-based protein representation methods performed well on the tasks related to protein-ligand interaction prediction. We speculate that this performance may have benefited from the subpocket-level anchor features learned by our model, which encoded the ligand-binding properties of the corresponding subpocket regions. To provide evidence to support this hypothesis, we examined whether the learned anchor features were highly associated with the ligand-binding patterns of the surrounding biophysical environment.

We first visualized the relationship between anchor features and local characteristics of protein pockets (Figure 4**a**). All the anchors that were close to a high-affinity ligand in the PDBbind-v2020 dataset were included in the visualization (i.e., anchors with distances to ligand fragment centers < 1 Å and affinities ≤ 100 nM). Among these anchors, we found that the anchor features were associated with certain protein surface geometric (i.e., shape index, reflecting the curvature of the local protein surface) and chemical (i.e., hydrophobicity) properties. In addition, the anchor features exhibited certain patterns related to protein atom features, such as the charge and the existence of hydrogen bond donors and acceptors in the amino acids close to the corresponding anchors.

Next, we examined the feature distributions of the anchors that were occupied by different types of ligand fragments (Figures 4**b**). For those fragments that were widely found in the small-molecule ligands, including phenyl groups (SMILES: *c1ccccc1), methylene groups (SMILES: *C*), ether groups (SMILES: *O*), and amide groups (SMILES: *NC(*)=O), the features of anchors occupied by them exhibited a dispersed pattern, indicating that the preferential subpockets of these fragments are relatively universal. Phosphate groups (SMILES: *OP(=O)(O)O) tended to bind to the protein pockets in hydrophilic regions mainly because of their polarity. For certain fragments, such as sulfamine (SMILES: *S(N)(=O)=O) and amidine (SMILES: *C(N)=[NH2+]) groups, the corresponding anchor features exhibited an aggregated pattern (Figure 4**b**). This can potentially be explained by the fact that these fragments are selectively bound to protein subpocket regions with specific properties. For example, according to Figure 4**a** and 4**b**, we assumed that the anchors occupied by the amidine group are likely to have a concave shape and be surrounded with negatively charged amino acids. Such geometric and chemical properties are well suited to these ligand fragments, which have only one attachment point and positive charges. To confirm our assumption about the properties of amidine-occupied anchors, we picked out three examples and examined the local subpocket environments. As expected, these anchors were located in the inner sides of protein pockets with concave shapes, and there was also at least one negatively charged amino acid (i.e., aspartic or glutamic acid) close to these anchors (Figure 4**c**). We also noticed that these anchors, though exhibiting similar patterns, were actually from three distinct proteins (furin, anti-dabigatran antibody, and urokinase-type plasminogen activator). This indicated that our model can capture similar local features from diverse proteins, suggesting that the learned patterns can be generalized to novel proteins for protein-ligand binding prediction.

All these analyses demonstrated that our PocketAnchor method can effectively extract the subpocket-level anchor features and thus provide useful protein pocket representations for modeling protein-ligand interactions.

3 Discussion and conclusion

Selecting proper feature representation methods is crucial for developing machine learning and deep learning-based models for protein-ligand interaction-related learning tasks. The anchor-based representation proposed in this work can provide informative features for learning the intrinsic properties of substructures in protein pockets. We have demonstrated that such a feature representation approach can achieve outstanding performance in two prediction tasks related to protein-ligand interactions. Despite the progress achieved in this work, the current anchor-based representation scheme still has some limitations. Although our PocketAnchor-based models do not require co-complex structures of proteins and compounds as input, the limited number of proteins with solved structures may still narrow the application scope of current structure-based models. Nevertheless, this issue can be largely resolved with recent advances in the protein structure prediction

field [39]. In addition, our current anchor-based representation framework depends on MaSIF [31] to calculate the surface features, which is relatively slow and occasionally fails for certain proteins. The generation of this framework can possibly be improved in the future by using alternative surface feature extraction strategies, e.g., an end-to-end feature learning strategy [40] for generating surface vertices.

4 Methods

4.1 Anchor-based protein pocket representation

Here we introduce "anchors", which are points sampled in the three-dimensional (3D) space within the protein pockets to represent the surrounding subpocket environment. More specifically, given a protein structure, evenly distributed grid points with an interval of d_g are first sampled near the protein surface, and only those points with distances to the nearest protein atoms within the range of 2–4 Å (which are estimated according to the observed distribution of the distances between all pairs of ligand and protein atoms in the training dataset) are kept. Then, the remaining grid points are clustered using an agglomerative clustering algorithm [41] with the maximum linkage criterion and a distance threshold of d_a. Finally, the centers of all the clusters are defined as anchors. When predicting the ligand binding sites, the anchors are sampled to cover the full protein structures, and the distance parameters are set as $d_g = 2$ Å and $d_a = 6$ Å. When predicting the binding affinities, the anchors are sampled to cover only the pocket region, which is defined by starting from the ligand center and then expanding to a maximum of 800 grid points, regardless of non-connecting points with distances larger than 8 Å to the nearest points considered. In this task, we choose $d_g = 1$ Å and $d_a = 4$ Å to describe the pocket regions more precisely.

4.2 The PocketAnchor module

In this section, we describe how to obtain the anchor features using our PocketAnchor method. Basically, each anchor gathers the information from protein atoms and the surface within a sphere of radius 6 Å to represent the corresponding subpocket environment. More specifically, given a protein, let $a_i,\ i = 1, \cdots, n_a$ denote the anchors, $u_j,\ j = 1, \cdots, n_u$ denote the atoms of the protein, and $s_k,\ k = 1, \cdots, n_s$ denote the vertices of the surface mesh, where i, j, and k stand for the indices, and n_a, n_u, and n_s stand for the numbers of anchors, atoms, and surface vertices in the protein, respectively. Let F_a, F_u, and F_s denote the feature vectors of anchors, atoms, and vertices, respectively. The initial feature vector $F_{u_j}^{(0)} \in \mathbb{R}^{131}$ of an atom u_j is defined as a concatenated vector containing one-hot encodings of atom elements, residue types, secondary structure elements, and other properties, namely B factor, formal charge, Van der Waals radius, number of protons, geometric type, and valence, obtained using PyMOL [42]. The initial feature vector $F_{s_k}^{(0)} \in \mathbb{R}^5$ of a surface vertex s_k is defined as a vector containing its geometric and chemical properties, as in MaSIF [31]. These features are learned and updated with message passing neural networks (MPNNs). Specifically, the protein atom features F_{u_j} of atom u_j and the surface features F_{s_k} of vertex s_k are updated through MPNNs (also see Figure 1**e**), according to the following formulas:

$$F_{u_j}^{(h)} = \mathrm{MPNN}_u^h\left(F_{u_j}^{(h-1)}, \left\{F_{u_i}^{(h-1)}, u_i \in \mathrm{Nr}_u(u_j)\right\}\right),$$
$$F_{s_k}^{(h)} = \mathrm{MPNN}_s^h\left(F_{s_k}^{(h-1)}, \left\{F_{s_i}^{(h-1)}, s_i \in \mathrm{Nr}_s(s_k)\right\}\right),$$

where $\mathrm{MPNN}_u^h(\cdot)$ and $\mathrm{MPNN}_s^h(\cdot)$ stand for the MPNN layers for updating atom and surface features, respectively, the superscript $h = 1, \cdots, H$ stands for the layer number in the MPNNs, H stands for the number of message-passing iterations, and $\mathrm{Nr}(\cdot)$ stands for the set of neighboring atoms, vertices, or anchors. Then, the atom and surface features from all the iterations are combined, that is,

$$F_{u_j} = W_u \cdot \mathrm{Cat}\left(F_{u_j}^{(0)}, \cdots, F_{u_j}^{(H)}\right), \quad F_{s_k} = W_s \cdot \mathrm{Cat}\left(F_{s_k}^{(0)}, \cdots, F_{s_k}^{(H)}\right),$$

where W_u and W_s stand for the learnable parameters and $\mathrm{Cat}(\cdot, \cdot)$ stands for the concatenation operation. Finally, the anchor features F_{a_i} of anchor a_i are aggregated from both the protein atoms and surface, that is,

$$F_{a_i} = \mathrm{Cat}\Big(\sum_{u_j \in \mathrm{Nr}_u(a_i)} F_{u_j} \cdot w_{ij}, \sum_{s_k \in \mathrm{Nr}_s(a_i)} F_{s_k} \cdot w_{ik}\Big),$$

$$w_{ij} = \frac{\exp(6 - \mathrm{Dist}(a_i, u_j))}{\sum_{u_k \in \mathrm{Nr}_u(a_i)} \exp(6 - \mathrm{Dist}(a_i, u_k))}, \quad w_{ik} = \frac{\exp(6 - \mathrm{Dist}(a_i, s_k))}{\sum_{s_j \in \mathrm{Nr}_s(a_i)} \exp(6 - \mathrm{Dist}(a_i, s_j))},$$

where w_{ij} and w_{ik} stand for the normalized distances as weights, $\mathrm{Dist}(\cdot, \cdot)$ stands for the Euclidean distance function, and $\exp(\cdot)$ stands for the exponential function.

4.3 The ligand feature encoding module

The ligand features can be represented hierarchically at three levels, i.e., the global, fragment, and atom levels. Fragments can be generated by splitting the ligands and breaking the non-ring single bonds as in [43]. A ligand feature encoder module is adopted from the graph convolution module of MONN [23] and slightly modified in this work to learn the three levels of ligand features. More specifically, given a ligand, let $g_i,\ i = 1, \cdots, n_g$ denote its fragments, and $t_j,\ j = 1, \cdots, n_t$ denote the atoms in the ligand, where i and j stand for the indices, and n_g and n_t stand for the numbers of fragments and atoms in the ligand, respectively. Let F_c, F_g, and F_t denote the global, fragment, and atom-level feature vectors, respectively. The initial feature vector of an atom is defined as a concatenated vector containing one-hot encodings of atom elements, atom degrees, valence, and aromatic features. Then, the atom features F_{t_j} and global features F_c are extracted and updated through the graph warp modules as in [23]. The features F_{g_i} of fragment g_i are calculated by averaging over all the atoms within the fragment, that is,

$$F_{g_i} = \frac{1}{|\{t_j | t_j \in g_i\}|} \sum_{\{t_j | t_j \in g_i\}} F_{t_j}.$$

4.4 The ligand binding site prediction module

The ligand binding site prediction module takes the anchor features extracted by PocketAnchor as input. More specifically, given an anchor a_i, its features F_{a_i} are first converted into an embedding space through linear projection followed by a leaky ReLU layer, that is,

$$F_{a_i}^{(\mathrm{site})} = \mathrm{LeakyReLU}\Big(W_a^{(\mathrm{site})} \cdot F_{a_i} + b_a^{(\mathrm{site})}\Big),$$

where the superscript "site" stands for the notation of ligand binding site prediction, $\mathrm{LeakyReLU}(x) = \max(0.1x, x)$ stands for the leaky ReLU activation function, and $W_a^{(\mathrm{site})}$ and $b_a^{(\mathrm{site})}$ stand for the learnable parameters of the linear projection layer. The binding site score $\hat{s}_{a_i}$ is then predicted through a linear projection followed by a sigmoid function, that is,

$$\hat{s}_{a_i} = \sigma(W^{(\mathrm{site})} \cdot F_{a_i}^{(\mathrm{site})} + b^{(\mathrm{site})}),$$

where $W^{(\mathrm{site})}$ and $b^{(\mathrm{site})}$ stand for learnable parameters, and $\sigma(\cdot)$ stands for the sigmoid function.

4.5 The binding affinity prediction module

To predict the binding affinity between protein p and ligand compound c, the affinity prediction module utilizes the extracted features from both the atom level (i.e., protein and ligand atoms) and substructure level (i.e., protein anchors and ligand fragments). More specifically, at the atom level, the protein atom features F_{u_i}, the ligand atom features F_{t_j}, and the ligand global features F_c are first converted into the corresponding embedding spaces through linear projection followed by a leaky ReLU activation function, that is,

$$F_{u_i}^{(\mathrm{atom})} = \mathrm{LeakyReLU}\Big(W_u^{(\mathrm{atom})} \cdot F_{u_i} + b_u^{(\mathrm{atom})}\Big),$$

$$F_{t_j}^{(\mathrm{atom})} = \mathrm{LeakyReLU}\Big(W_t^{(\mathrm{atom})} \cdot F_{t_j} + b_t^{(\mathrm{atom})}\Big),$$

$$F_c^{(\mathrm{atom})} = \mathrm{LeakyReLU}\Big(W_c^{(\mathrm{atom})} \cdot F_c + b_c^{(\mathrm{atom})}\Big),$$

where the superscript "atom" stands for the notation of the atom-level features, $\mathrm{LeakyReLU}(x) = \max(0.1x, x)$ stands for the leaky ReLU activation function, and $W_u^{(\mathrm{atom})}$, $W_t^{(\mathrm{atom})}$, $W_c^{(\mathrm{atom})}$, $b_u^{(\mathrm{atom})}$, $b_t^{(\mathrm{atom})}$, and $b_c^{(\mathrm{atom})}$ stand for the learnable parameters of the linear projection layers. The atom features are then updated through a self-attention layer to account for the importance score of individual atoms, that is,

$$\hat{F}_u^{(\mathrm{atom})} = \sum_{u_i} F_{u_i}^{(\mathrm{atom})} \cdot w_i, \qquad \hat{F}_t^{(\mathrm{atom})} = \sum_{t_j} F_{t_j}^{(\mathrm{atom})} \cdot w_j,$$

where w_i and w_j stand for the weights for individual features, which are calculated as follows:

$$w_i = \mathrm{Softmax}\left(W_u^{(\mathrm{att})} \cdot F_{u_i}^{(\mathrm{atom})} + b_u^{(\mathrm{att})}\right), \qquad w_j = \mathrm{Softmax}\left(W_t^{(\mathrm{att})} \cdot F_{t_j}^{(\mathrm{atom})} + b_t^{(\mathrm{att})}\right),$$

where $\mathrm{Softmax}(x_i) = \exp(x_i)/\sum_j \exp(x_j)$, and $W_u^{(\mathrm{att})}$, $W_t^{(\mathrm{att})}$, $b_u^{(\mathrm{att})}$, and $b_t^{(\mathrm{att})}$ stand for the learnable parameters of the self attention layers. The atom-level features are then obtained through the outer product between protein atom features and ligand atom features, that is,

$$F^{(\mathrm{atom})} = \hat{F}_u^{(\mathrm{atom})} \cdot \mathrm{Cat}\left(\hat{F}_t^{(\mathrm{atom})}, F_c^{(\mathrm{atom})}\right)^\top.$$

The substructure-level features $F^{(\mathrm{sub})}$ are obtained in a similar way by using the protein anchor features F_{a_i}, the ligand fragment features F_{g_j}, and the ligand global features F_c. Finally, the binding affinity $\hat{a}$ is predicted through a linear projection of the above feature vectors:

$$\hat{a} = W^{(\mathrm{aff})} \cdot \mathrm{Cat}\left(F^{(\mathrm{atom})}, F^{(\mathrm{sub})}\right) + b^{(\mathrm{aff})},$$

where $W^{(\mathrm{aff})}$ and $b^{(\mathrm{aff})}$ stand for the learnable parameters.

4.6 Data processing and evaluation for the ligand-binding site prediction task

We used the scPDB v2017-derived dataset [44] to train our PocketAnchor-site model as was done for DeepSurf [13]. During the training process, the anchors within a radius of 4 Å from any ligand atom were assigned as positive training samples while the rest were assigned as negative ones. During the evaluation process, to determine the centers of binding pockets based on the predicted scores of anchors, we first selected the anchor points with prediction scores that were two standard deviations above the average score for each protein. Then, we used a greedy strategy to cluster the selected anchors. That is, we started from the anchor with the highest score and then expanded the cluster by including those selected anchors within 3 Å. The above process was repeated until no anchor was left. The averaged anchor coordinate of each cluster was marked as a pocket center, and all the pocket centers in a sample were then ranked according to the number of anchors within the corresponding clusters.

To evaluate the performance of PocketAnchor-site and baseline methods on the ligand-binding site prediction task, we used two benchmark datasets, COACH420 [34] and HOLO4k [35], as test sets, which contained 420 and 4,009 protein-ligand complexes, respectively. Homologous proteins in the benchmark test datasets were removed to prevent data leakage (20 and 475 samples were removed from COACH420 and HOLO4k, respectively). Two proteins were considered as homologous if their similarity score, calculated using the sequence alignment obtained by the Smith-Waterman algorithm [45], was greater than 0.9 [13]. The true pocket labels were defined based on the ligands provided from the original datasets [34, 35], and 181 samples that contained no ligand matching the list in HOLO4k were removed. The samples that failed to be processed and predicted by any method were also removed for fair comparison. Specifically, for the COACH420 dataset, PocketAnchor-site, P2RANK, DeepSurf, and DeepSite failed to generate prediction results for 0, 4, 7, and 3 samples, respectively; for the HOLO4k dataset, the numbers were 7, 1, 853, and 21, respectively. The final numbers of samples evaluated for COACH420 and HOLO4k were 391 (511 pockets) and 2,523 (5,150 pockets), respectively. The prediction results of DeepSite [12] and P2Rank [14] were obtained from P2Rank [14], and the prediction results of DeepSurf were obtained using the trained model provided in the GitHub repository [13].

4.7 Data processing and evaluation for the binding affinity prediction task

As described in the main text, we employed three train-test splitting schemes to evaluate the performance of PocketAnchor-affinity and baseline methods on the protein-ligand binding affinity prediction task. In the original CASF split, the PDBbind v2016 general set was used as the training set and the corresponding core-set was used as the test set [7]. The samples appearing in the test set were removed from the training set. For the new-protein and expanded CASF splits, the proteins in the PDBbind v2020 dataset [46] were clustered according to the similarity scores calculated using the Smith-Waterman sequence alignment algorithm [45]. Proteins with sequence similarities greater than or equal to 0.7 were assigned to the same cluster. The samples in the PDBbind v2020 dataset were used as the training set, and the proteins in the same clusters as those in CASF2016 were removed. The affinity label of a protein-ligand complex was normalized by $-\log_{10}(\text{affinity})[\text{mol/L}]$.

For each protein-ligand pair, the protein and the ligand were pre-processed separately. The ligand information was extracted from the PDBbind v2020 database [46]. For each ligand, the fragments were obtained using RDKit [47] following the same rules as in [43]. For proteins, the protein-ligand complexes were first downloaded in .pdb format from the Protein Data Bank (PDB, https://www.rcsb.org). Then, all the solvent molecules (e.g., water) and ligands in the structures were removed. For each protein, the nearest biological assembly to the ligand center was obtained, and the atom and surface features were extracted using PyMOL [42] and MaSIF [31], respectively. The biological assembly information was retrieved from the lines of the .pdb files starting with "REMARK 350".

For the baseline methods, we followed the same pre-processing protocols and recommended hyper-parameters as in the original papers. Note that since the protein sequences were not provided by the PDBbind database, the sequences retrieved using distinct schemes might be different. Here, we trained the sequence-based baseline models using protein sequences from either PDB or the Uniprot database separately and reported the best performance. More specifically, to extract a protein sequence from its structure file obtained from the PDB, we first selected a chain with the largest number of atoms within the 8 Å neighborhood of the ligand. Then the sequence of the chain was used as the PDB sequence, in which the non-standard residues were marked with "X". The sequences with non-standard residues making up more than 50% of the total length were considered abnormal and thus removed. We also adopted the mappings from the PDB IDs to UniProt IDs provided by PDBbind [46], and extracted the protein sequences from the Uniprot database [48]. Those samples that failed during the pre-processing procedure were removed.

4.8 Training and hyper-parameter selection

For the ligand binding site prediction task, cross-entropy loss was used for training. For the protein-ligand binding affinity prediction task, mean-square-error loss was employed. For each task, 20% of the training data were separated and used as a validation set in each repeat. The validation set was selected randomly for the original CASF split. For the new-protein and expanded CASF splits, the validation set was chosen based on the protein clusters, ensuring that the protein clusters were distinct from those in the training set. Because of a large number of hyper-parameters in our model, the hyper-parameters were selected empirically or based on the validation performance. In particular, the number of epochs was determined using an early stopping technique [49] with a patience parameter of 20 epochs on the validation set. In other words, the learning process would stop when the performance measured on the validation set was no longer improved after 20 epochs.

Author contributions S.L., T.T., D.Z., and J.Z. conceived the project. S.L. and T.T. designed the methodology and performed experiments. S.L., T.T., Z.Zhang, and Z.Zou analyzed results. S.L., T.T., and J.Z. wrote the paper. S.L., T.T., Z.Zhang, Z.Zou, D.Z., and J.Z. contributed to the revision of the manuscript.

Acknowledgement This work was supported in part by the National Natural Science Foundation of China (61872216, T2125007 to JZ, 31900862 to DZ), the National Key Research and Development Program of China (2021YFF1201300), the Turing AI Institute of Nanjing, and the Tsinghua-Toyota Joint Research Fund.

We thank Mr. Shengde Zhang and Mr. Lin Chen for the helpful discussions about this work.

Code availability The source code can be found in our GitHub repository (https://github.com/tiantz17/PocketAnchor).

Figures

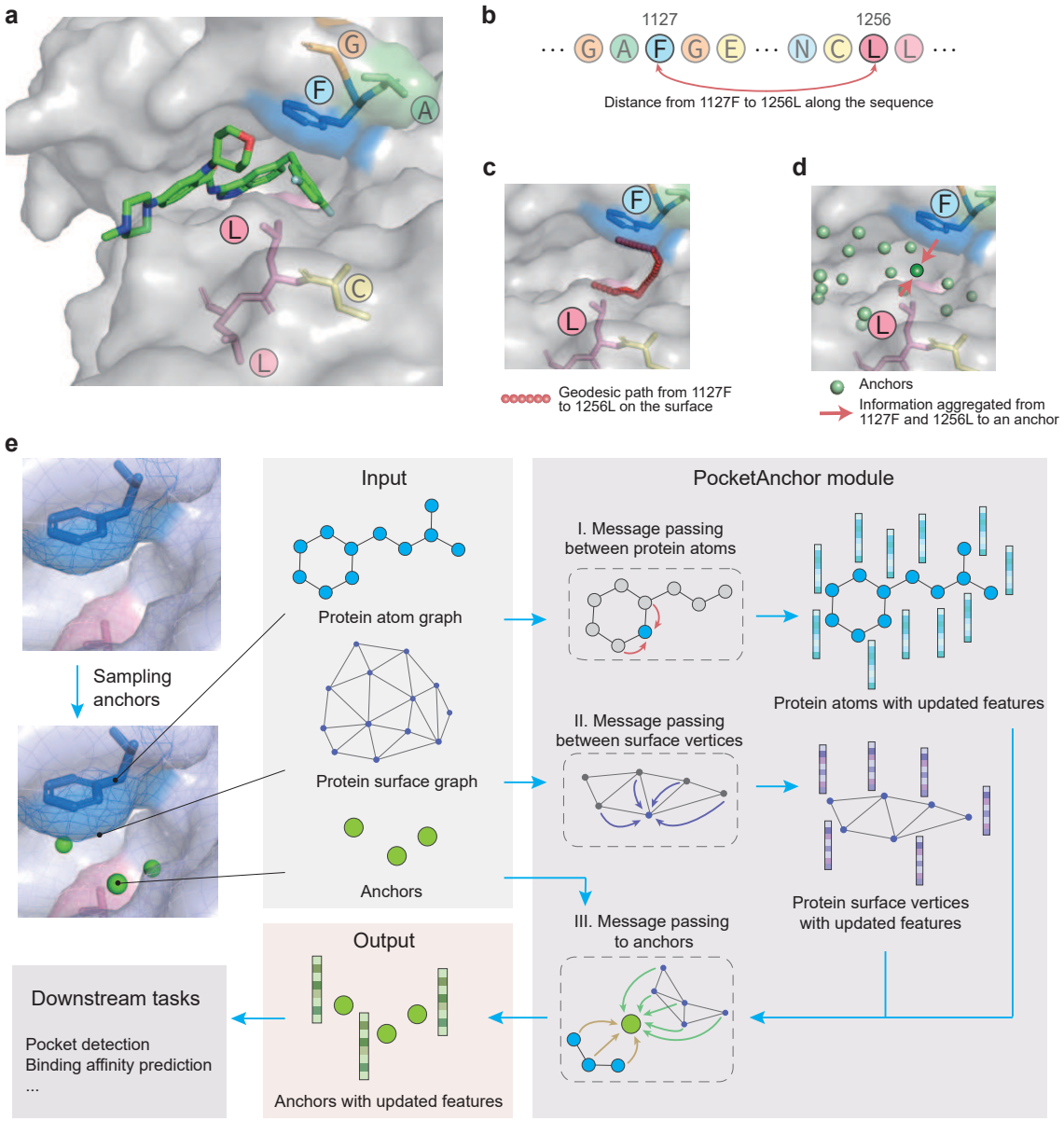

Figure 1: Illustrations of different feature representations of protein pockets and a description of the PocketAnchor module for obtaining anchor-based protein feature representations. **a**. An example of a protein-ligand complex (PDB ID: 5FTO). Residues close to the 3,5-difluorobenzyl moiety of the ligand are colored by amino acid type (the colors of amino acids are consistent in **a**–**e**). **b**. Protein feature representation based on the amino-acid sequence. **c**. Protein feature representation based on protein surface descriptors. The geodesic path from 1127F to 1256L along the protein surface is shown. **d**. Protein feature representation based on anchors (green balls), which are sampled from points within the protein pocket (see the main text for more details). Anchors can bridge the essential residues contributing to the properties of the local pocket regions. **e**. Deriving the anchor feature representations using the PocketAnchor module. The protein pocket is represented by anchors, whose features are aggregated from protein atoms and the surface in three steps.

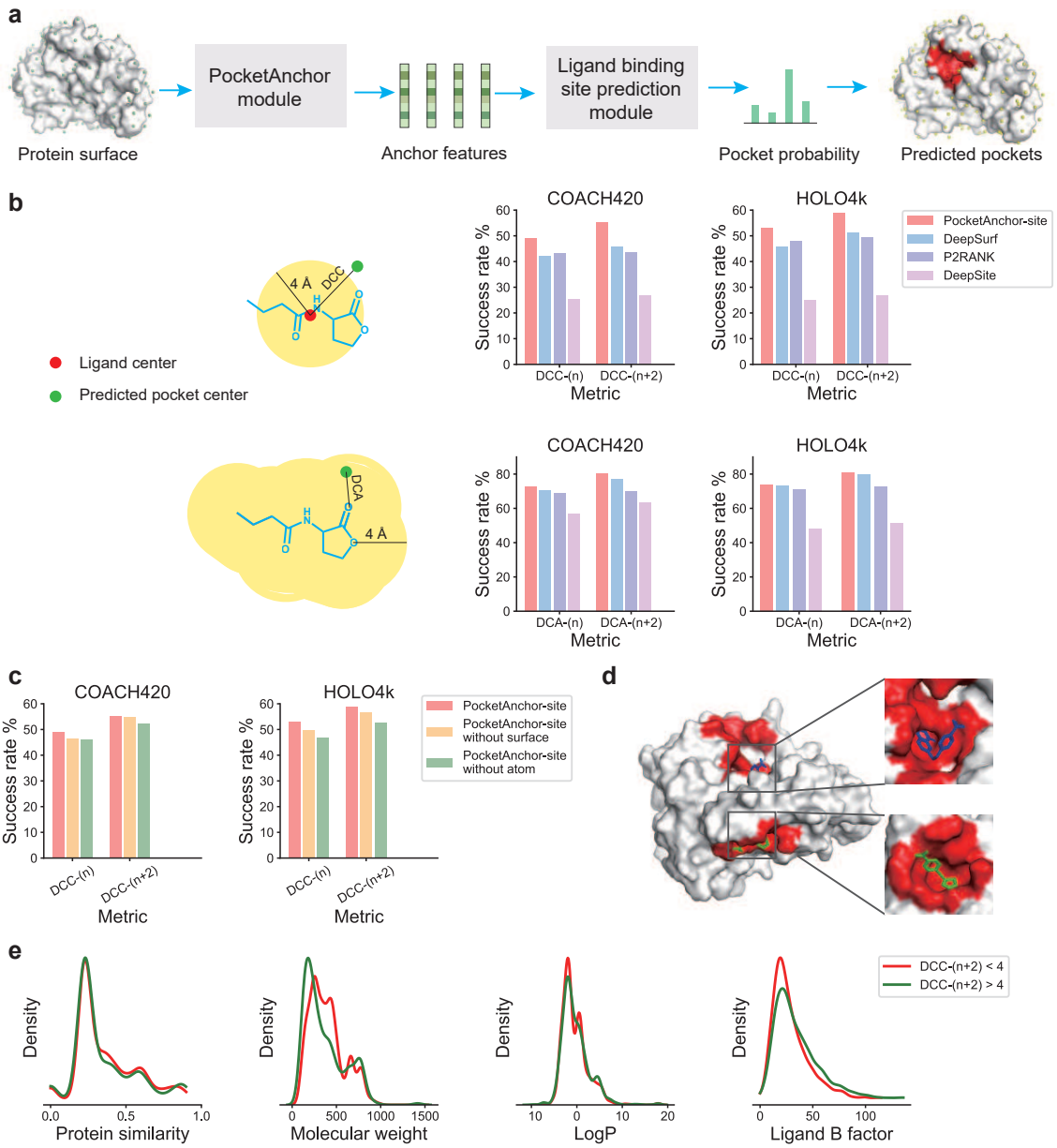

Figure 2: PocketAnchor-site accurately predicts the ligand binding sites of proteins. **a**. The architecture of the PocketAnchor-site model (see the main text and Methods for more details). **b**. Illustration of the DCC and DCA criteria. **c**. Performance of PocketAnchor-site and baseline methods on the binding site prediction task, evaluated in terms of the success rates determined according to the criterion DCC or DCA < 4 Å. DCC/DCA-(n) and DCC/DCA-($n+2$) stand for the DCC/DCA scores measured using the top-n and top-$(n+2)$ predicted binding pockets for each sample, respectively, where n stands for the number of pockets in the sample. **d**. Performance of the PocketAnchor-site model compared with models without information from either protein surface or atom features, evaluated in terms of the DCC-based success rates as in **c**. **e**. An example of a binding site prediction result. The ligand binding sites colored in red on the ricin protein (PDB ID: 1BR6) were predicted by the PocketAnchor-site model. The ground truth ligand from the COACH420 benchmark dataset is colored in blue. The other binding site predicted by the PocketAnchor-site model was previously reported to be the binding site for another ligand [37], which is colored in green (PDBID: 6URW). **f**. Distributions of protein or ligand properties for two groups of samples, divided according to the DCC-$(n+2)$ criterion with a threshold of 4 Å. The maximal similarity to the training proteins, ligand molecule weight, ligand logP, and ligand B factor are illustrated. The curves of the corresponding estimated probability distribution functions are shown.

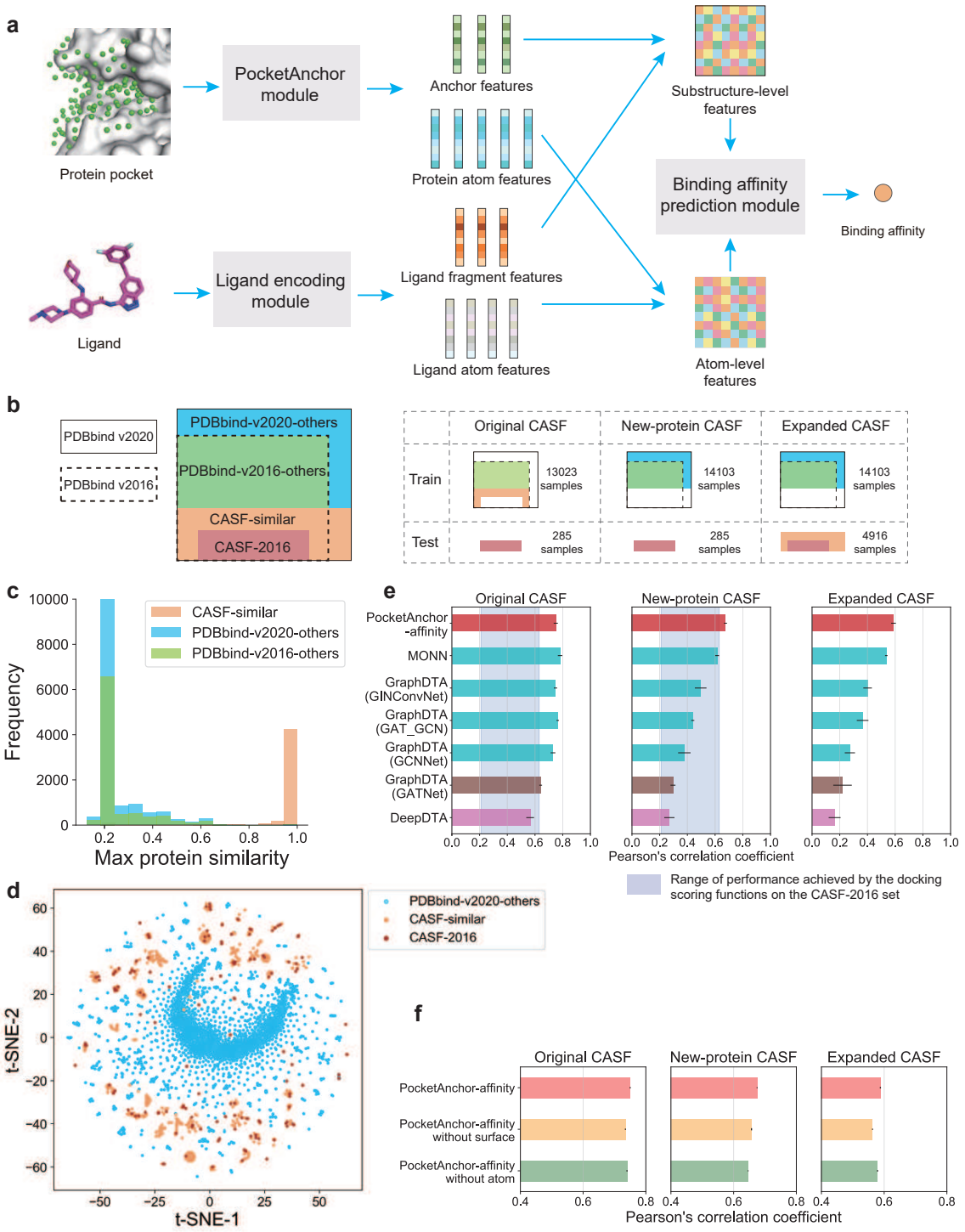

Figure 3: Performance evaluation of the protein-ligand binding affinity prediction task. **a**. The PocketAnchor-affinity model architecture (see the main text and Methods for more details). **b**. The definitions of the three train-test splitting schemes. **c**. The distribution of similarity scores for proteins in different subsets. For a protein, the corresponding similarity score is defined as its maximum sequence similarity with all the proteins in the CASF-2016 set. **d**. Visualization of different subsets of proteins in the PDBbind dataset using t-SNE. **e**. Performance of PocketAnchor-affinity and baseline methods on three splitting schemes for the protein-ligand binding affinity prediction task, measured in terms of Pearson's correlation coefficients (PCCs). The error bars indicate the standard deviations over five repeats. The shaded regions for the original CASF and new-protein CASF splits denote the range of performances achieved by the docking scoring functions obtained from [7]. **f**. Performance of the PocketAnchor-affinity model compared with the models without information from either protein surface or atom features, evaluated in terms of PCCs as in **e**.

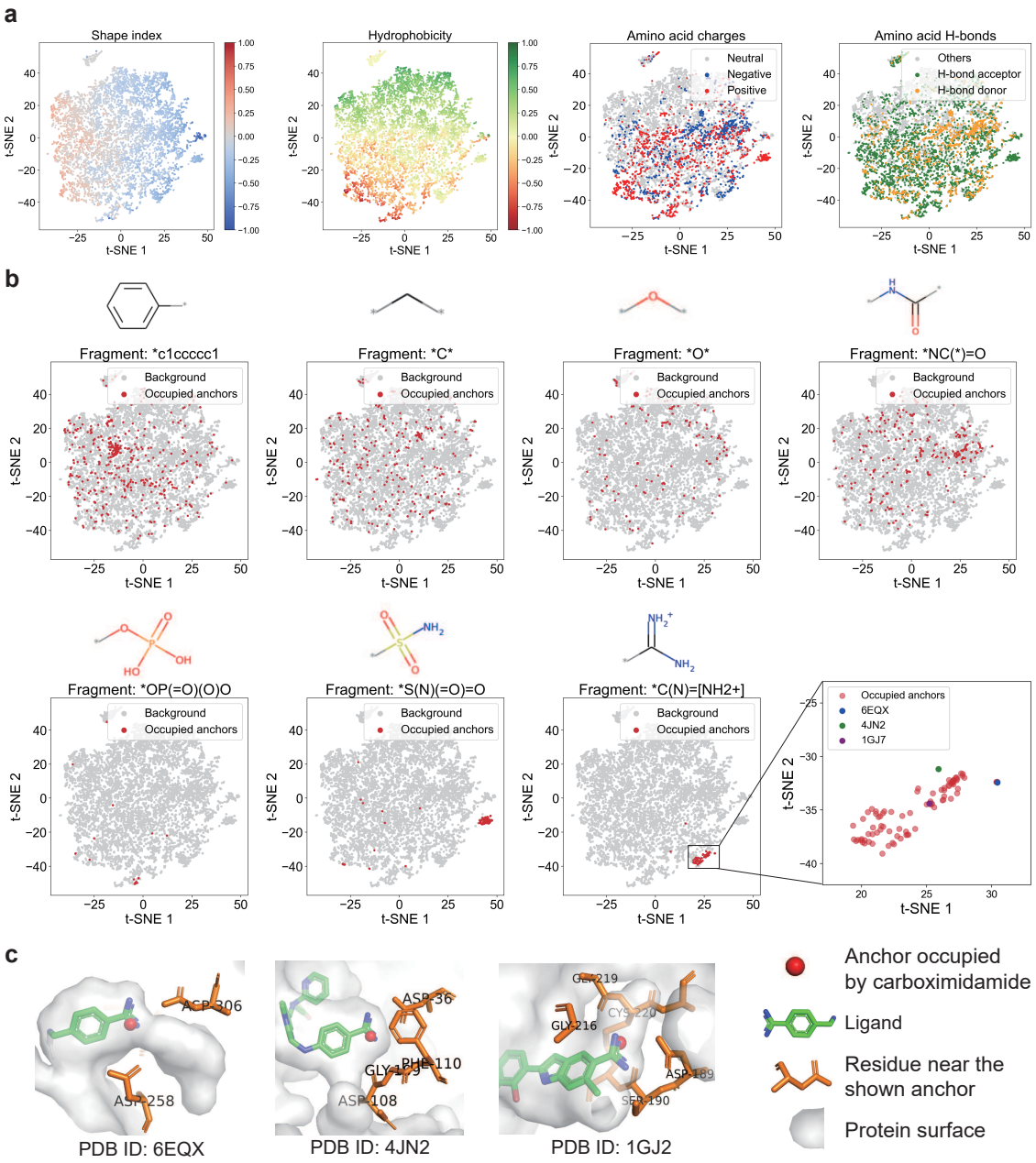

Figure 4: PocketAnchor can learn the ligand-binding characteristics of protein subpockets. **a**. Visualization of anchor features using t-SNE. Colors indicate the protein properties, namely the shape index of protein surface, hydrophobicity, amino acid charge types, and hydrogen bond types. The former two properties were obtained by averaging the corresponding properties of surface points within a 6 Å distance from the anchor, while the latter two were collected from the amino acid closest to each anchor. Here, the "H-bond acceptor" group represents the amino acids that can only serve as hydrogen bond acceptors and do not contain any hydrogen bond donor atoms. **b**. Visualization of anchor features using t-SNE, with those anchors occupied by specific types of ligand fragments colored in red. The diagram and SMILES strings of these ligand fragments are shown. The region covering the majority of the colored anchors in the last example is magnified, and three anchors from three distinct samples are marked in different colors. **c**. The three selected anchors from the zoomed-in panel in **b**, in which three anchors from different samples were occupied by a specific ligand fragment. Only the corresponding ligands and the residues located within 4 Å of the selected anchors in the protein pockets are shown.

References

[1] Xing Du, Yi Li, Yuan-Ling Xia, Shi-Meng Ai, Jing Liang, Peng Sang, Xing-Lai Ji, and Shu-Qun Liu. Insights into protein–ligand interactions: mechanisms, models, and methods. *International Journal of Molecular Sciences*, 17(2):144, 2016.

[2] Tony Pawson and Piers Nash. Assembly of cell regulatory systems through protein interaction domains. *Science*, 300(5618):445–452, 2003.

[3] Duncan E Scott, Andrew R Bayly, Chris Abell, and John Skidmore. Small molecules, big targets: drug discovery faces the protein–protein interaction challenge. *Nature Reviews Drug Discovery*, 15(8):533–550, 2016.

[4] Inga Jarmoskaite, Ishraq AlSadhan, Pavanapuresan P Vaidyanathan, and Daniel Herschlag. How to measure and evaluate binding affinities. *Elife*, 9:e57264, 2020.

[5] Sangsoo Lim, Yijingxiu Lu, Chang Yun Cho, Inyoung Sung, Jungwoo Kim, Youngkuk Kim, Sungjoon Park, and Sun Kim. A review on compound-protein interaction prediction methods: Data, format, representation and model. *Computational and Structural Biotechnology Journal*, 19:1541, 2021.

[6] CR Beddell, PJ Goodford, FE Norrington, S Wilkinson, and R Wootton. Compounds designed to fit a site of known structure in human haemoglobin. *British Journal of Pharmacology*, 57(2):201–209, 1976.

[7] Minyi Su, Qifan Yang, Yu Du, Guoqin Feng, Zhihai Liu, Yan Li, and Renxiao Wang. Comparative assessment of scoring functions: the CASF-2016 update. *Journal of Chemical Information and Modeling*, 59(2):895–913, 2018.

[8] Oleg Trott and Arthur J Olson. AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of Computational Chemistry*, 31(2):455–461, 2010.

[9] David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. Lessons learned in empirical scoring with smina from the CSAR 2011 benchmarking exercise. *Journal of Chemical Information and Modeling*, 53(8):1893–1904, 2013.

[10] Xin Yang, Yifei Wang, Ryan Byrne, Gisbert Schneider, and Shengyong Yang. Concepts of artificial intelligence for computer-assisted drug discovery. *Chemical Reviews*, 119(18):10520–10594, 2019.

[11] Vincent Le Guilloux, Peter Schmidtke, and Pierre Tuffery. Fpocket: an open source platform for ligand pocket detection. *BMC Bioinformatics*, 10(1):1–11, 2009.

[12] José Jiménez, Stefan Doerr, Gerard Martínez-Rosell, Alexander S Rose, and Gianni De Fabritiis. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics*, 33(19):3036–3042, 2017.

[13] Stelios K Mylonas, Apostolos Axenopoulos, and Petros Daras. DeepSurf: a surface-based deep learning approach for the prediction of ligand binding sites on proteins. *Bioinformatics*, 37(12):1681–1690, 2021.

[14] Radoslav Krivák and David Hoksza. P2Rank: machine learning based tool for rapid and accurate prediction of ligand binding sites from protein structure. *Journal of Cheminformatics*, 10(1):1–12, 2018.

[15] Limeng Pu, Rajiv Gandhi Govindaraj, Jeffrey Mitchell Lemoine, Hsiao-Chun Wu, and Michal Brylinski. DeepDrug3D: Classification of ligand-binding pockets in proteins with a convolutional neural network. *PLoS Computational Biology*, 15(2):e1006718, 2019.

[16] Marta M Stepniewska-Dziubinska, Piotr Zielenkiewicz, and Pawel Siedlecki. Development and evaluation of a deep learning model for protein–ligand binding affinity prediction. *Bioinformatics*, 34(21):3666–3674, 2018.

[17] Matthew Ragoza, Joshua Hochuli, Elisa Idrobo, Jocelyn Sunseri, and David Ryan Koes. Protein–ligand scoring with convolutional neural networks. *Journal of Chemical Information and Modeling*, 57(4):942–957, 2017.

[18] Duc Duy Nguyen and Guo-Wei Wei. AGL-Score: Algebraic graph learning score for protein–ligand binding scoring, ranking, docking, and screening. *Journal of Chemical Information and Modeling*, 59(7):3291–3304, 2019.

[19] Liangzhen Zheng, Jingrong Fan, and Yuguang Mu. Onionnet: a multiple-layer intermolecular-contact-based convolutional neural network for protein–ligand binding affinity prediction. *ACS omega*, 4(14):15956–15965, 2019.

[20] Fangping Wan, Yue Zhu, Hailin Hu, Antao Dai, Xiaoqing Cai, Ligong Chen, Haipeng Gong, Tian Xia, Dehua Yang, Ming-Wei Wang, et al. DeepCPI: a deep learning-based framework for large-scale in silico drug screening. *Genomics, Proteomics & Bioinformatics*, 17(5):478–495, 2019.

[21] Hakime Öztürk, Arzucan Özgür, and Elif Ozkirimli. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics*, 34(17):i821–i829, 2018.

[22] Thin Nguyen, Hang Le, Thomas P Quinn, Tri Nguyen, Thuc Duy Le, and Svetha Venkatesh. GraphDTA: Predicting drug–target binding affinity with graph neural networks. *Bioinformatics*, 37(8):1140–1147, 2021.

[23] Shuya Li, Fangping Wan, Hantao Shu, Tao Jiang, Dan Zhao, and Jianyang Zeng. MONN: a multi-objective neural network for predicting compound-protein interactions and affinities. *Cell Systems*, 10(4):308–322, 2020.

[24] Youjun Xu, Chenjing Cai, Shiwei Wang, Luhua Lai, and Jianfeng Pei. Efficient molecular encoders for virtual screening. *Drug Discovery Today: Technologies*, 32:19–27, 2019.

[25] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.

[26] Amelia Villegas-Morcillo, Stavros Makrodimitris, Roeland CHJ van Ham, Angel M Gomez, Victoria Sanchez, and Marcel JT Reinders. Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, 37(2):162–170, 2021.

[27] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.

[28] Yuning You and Yang Shen. Cross-modality protein embedding for compound-protein affinity and contact prediction. *arXiv preprint arXiv:2012.00651*, 2020.

[29] Yeji Wang, Shuo Wu, Yanwen Duan, and Yong Huang. A point cloud-based deep learning strategy for protein-ligand binding affinity prediction. *arXiv preprint arXiv:2107.04340*, 2021.

[30] Zhen Li, Xu Yan, Qing Wei, Xin Gao, Sheng Wang, and Shuguang Cui. PointSite: a point cloud segmentation tool for identification of protein ligand binding atoms. 2019.

[31] Pablo Gainza, Freyr Sverrisson, Frederico Monti, Emanuele Rodola, D Boscaini, MM Bronstein, and BE Correia. Deciphering interaction fingerprints from protein molecular surfaces using geometric deep learning. *Nature Methods*, 17(2):184–192, 2020.

[32] Martin Simonovsky and Joshua Meyers. DeeplyTough: learning structural comparison of protein binding sites. *Journal of Chemical Information and Modeling*, 60(4):2356–2366, 2020.

[33] Maria Menichincheri, Elena Ardini, Paola Magnaghi, Nilla Avanzi, Patrizia Banfi, Roberto Bossi, Laura Buffa, Giulia Canevari, Lucio Ceriani, Maristella Colombo, et al. Discovery of entrectinib: a new 3-aminoindazole as a potent anaplastic lymphoma kinase (ALK), c-ros oncogene 1 kinase (ROS1), and pan-tropomyosin receptor kinases (Pan-TRKs) inhibitor. *Journal of Medicinal Chemistry*, 59(7):3392–3408, 2016.

[34] Jianyi Yang, Ambrish Roy, and Yang Zhang. Protein–ligand binding site recognition using complementary binding-specific substructure comparison and sequence profile alignment. *Bioinformatics*, 29(20):2588–2595, 2013.

[35] Peter Schmidtke, Catherine Souaille, Frédéric Estienne, Nicolas Baurin, and Romano T Kroemer. Large-scale comparison of four binding site detection algorithms. *Journal of Chemical Information and Modeling*, 50(12):2191–2200, 2010.

[36] Zhihai Liu, Yan Li, Li Han, Jie Li, Jie Liu, Zhixiong Zhao, Wei Nie, Yuchen Liu, and Renxiao Wang. PDB-wide collection of binding data: current status of the PDBbind database. *Bioinformatics*, 31(3):405–412, 2015.

[37] Xiao-Ping Li, Rajesh K Harijan, Jennifer N Kahn, Vern L Schramm, and Nilgun E Tumer. Small molecule inhibitors targeting the interaction of ricin toxin A subunit with ribosomes. *ACS Infectious Diseases*, 6(7):1894–1905, 2020.

[38] Shuangjia Zheng, Yongjian Li, Sheng Chen, Jun Xu, and Yuedong Yang. Predicting drug–protein interaction using quasi-visual question answering system. *Nature Machine Intelligence*, 2(2):134–140, 2020.

[39] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[40] Freyr Sverrisson, Jean Feydy, Bruno E Correia, and Michael M Bronstein. Fast end-to-end learning on protein surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15272–15281, 2021.

[41] Stephen C Johnson. Hierarchical clustering schemes. *Psychometrika*, 32(3):241–254, 1967.

[42] Warren L DeLano et al. Pymol: An open-source molecular graphics tool. *CCP4 Newsl. Protein Crystallogr*, 40(1):82–92, 2002.

[43] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Hierarchical generation of molecular graphs using structural motifs. In *International Conference on Machine Learning*, pages 4839–4848. PMLR, 2020.

[44] Jamel Meslamani, Didier Rognan, and Esther Kellenberger. sc-PDB: a database for identifying variations and multiplicity of 'druggable'binding sites in proteins. *Bioinformatics*, 27(9):1324–1326, 2011.

[45] Mengyao Zhao, Wan-Ping Lee, Erik P Garrison, and Gabor T Marth. SSW library: an SIMD Smith-Waterman C/C++ library for use in genomic applications. *PLoS One*, 8(12):e82138, 2013.

[46] Renxiao Wang, Xueliang Fang, Yipin Lu, Chao-Yie Yang, and Shaomeng Wang. The PDBbind database: methodologies and updates. *Journal of Medicinal Chemistry*, 48(12):4111–4119, 2005.

[47] Greg Landrum. RDKit: A software suite for cheminformatics, computational chemistry, and predictive modeling, 2013.

[48] UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 2019.

[49] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.