ACS OMEGA

http://pubs.acs.org/journal/acsodf Article

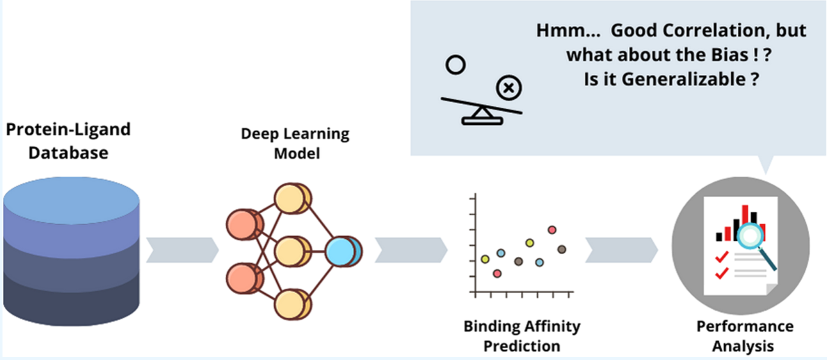
Latent Biases in Machine Learning Models for Predicting Binding Affinities Using Popular Data Sets

Ganesh Chandan Kanakala, Rishal Aggarwal, Divya Nayar, and U. Deva Priyakumar*

Cite This: *ACS Omega* 2023, 8, 2389−2397

Read Online

ACCESS | Metrics & More | Article Recommendations



ABSTRACT: Drug design involves the process of identifying and designing molecules that bind well to a given receptor. A vital computational component of this process is the protein−ligand interaction scoring functions that evaluate the binding ability of various molecules or ligands with a given protein receptor binding pocket reasonably accurately. With the publicly available protein−ligand binding affinity data sets in both sequential and structural forms, machine learning methods have gained traction as a top choice for developing such scoring functions. While the performance shown by these models is optimistic, there are several hidden biases present in these data sets themselves that affect the utility of such models for practical purposes such as virtual screening. In this work, we use published methods to systematically investigate several such factors or biases present in these data sets. In our analysis, we highlight the importance of considering sequence, protein−ligand interaction, and pocket structure similarity while constructing data splits and provide an explanation for good protein-only and ligand-only performances in some data sets. Through this study, we provide to the community several pointers for the design of binding affinity predictors and data sets for reliable applicability.

INTRODUCTION

Drug design is the process of developing and identifying small molecules or ligands that bind to a given protein molecule to modulate its function for therapeutic causes.[1] An integral requirement for such a process is a scoring function that can evaluate the strength of binding between the ligand and the protein.[2] Such a scoring function should be able to discern strong binders from nonbinders and is typically used to identify good hit candidates for a given receptor from a large library of molecules, a process more commonly known as virtual screening.

Machine learning and deep learning have been gaining traction as top choices for the development of these scoring functions due to their rapid success in technological domains such as computer vision[3] and natural language processing.[4] Protein−ligand scoring functions tend to be a natural extension of applying such algorithms as molecular information can easily be represented in the form of three-dimensional (3D) grids,[5] sequences, and graphs.[6,7] Although deep learning models are known to portray a very high accuracy, they are usually black-box functions that learn hidden features in the input data to make their predictions. Therefore, it is important to perform further post hoc analysis to ensure that these models are focusing on relevant features and are not predicting based on unintended biasing signals in the data. Ensuring the absence of biases in data sets enhances the practical utility of such algorithms, especially in fields such as health care and molecular informatics.

The database of useful decoys-enhanced (DUD-E)[8] and the maximum unbiased validation (MUV) data set[9] were initially

Received: October 20, 2022
Accepted: November 21, 2022
Published: January 5, 2023

developed to enable virtual scoring function development and benchmarking. These data sets contained large libraries of chemically similar active (binding) and decoy (poorly binding) molecules for a given set of protein structures. When deep learning models were trained to classify between actives and decoys, they showcased very high accuracy.[10−12] On further investigation, however, it was found that chemical descriptors of the small molecules themselves provided a sufficient signal to differentiate between the two classes.[13−15] This bias correlated very well with the high accuracy of deep learning models, thereby indicating that models trained on these data sets do not focus on protein−ligand interactions but rather on the features of the ligands themselves for classification. This leads to a situation where the models perform very well on the test data but fare poorly on future unseen data. Therefore, developing a scoring function based on an active vs decoy classification framework was found to be infeasible. The LIT-PCBA data set[16] was created to address such biases in virtual screening data sets, but its utility for training deep learning models needs to be explored further, especially since the data set contains only 15 targets.

An alternate approach for the scoring function is to use deep learning models to predict the binding affinity based on the 3D structures of protein−ligand complexes. This, in turn, could be used to rank small molecules based on favorable binding affinities. The PDBbind database[17] provides experimentally determined binding affinity values for protein−ligand cocrystal structures present in the Protein Data Bank (PDB).[18] This enables the development of deep learning model that takes the structure of the bound molecules for the prediction of binding affinity. 3D convolutional neural networks (CNNs),[19,20] graph neural networks (GNNs),[21] and more sophisticated deep learning architectures[22] when trained on this data set have shown good correlation between predicted and experimental values.

A recent series of studies, however, have shown that when such models are trained on protein-only or ligand-only information in this data set, they show similar performances as the full complex.[14,23] This again begs the question of whether these deep learning models focus on protein−ligand interactions when individual molecules (ligand-only and protein-only) themselves are enough to make accurate predictions on this data set. Another recent work also showed the efficacy of such deep learning-based protein−ligand scoring function on different virtual screening benchmarks.[24] The deep learning model showcased a slightly better enrichment factor over the traditional baseline method (Autodock Vina[25]) only for some of the receptors in the DUD-E and LIT-PCBA data sets. This performance improvement is still seemingly low for deep learning models compared to the excellent performance they showcase in drug design tasks like molecular docking.[26] Therefore, these data sets need to be studied in greater detail to improve the screening capabilities of deep learning scoring functions.

Several other binding affinity data sets provide protein−ligand data in the form of amino acid sequences and SMILES strings. These data sets are usually much larger than the data sets that provide protein−ligand cocrystal structures (PDBBind), which, in general, are difficult to resolve experimentally. In this study, we work with the Davis[27] and the KIBA[28] data sets that have been widely used by deep learning models for benchmarking their performances. These data sets also represent key situations in the virtual screening process where the structure of the target of interest has not been resolved and only the amino acid sequence of the protein is available. Therefore, accurate models on these data sets could have greater implications in virtual screening, especially when the structure of the target is difficult to elucidate. These data sets have been added to a collection of important therapeutics-related benchmarks called therapeutic data commons (TDC),[29] an open science initiative that aims to provide AI/ML-ready data sets and tools for health care and drug discovery method development. The DeepPurpose toolkit[30] was also created for easy development and usage of methods dependent on these data sets.

Most deep learning methods that have worked with the sequence-based data sets[31−33] so far have split the data randomly into cross-validation folds to benchmark the models. However, it is well known that similar protein sequences lead to similar 3D structures and therefore similar protein−ligand interactions. Random splits cause such similar sequences to be present in both training and test sets leading to data leakage in the benchmark. These models portray very low error rates on such a benchmark due to this internal bias that is present in random splits. In some ways, such a split would also reward overfitting as the test set would not be an indication of the model's generalizability.

In this work, we study the presence of various different types of biases present in protein−ligand data sets that affect deep learning performances in binding affinity prediction and virtual screening. We do this by drawing parallels between results obtained on sequence-based and structure-based data sets for multiple experiments. We highlight the effect of obtaining overoptimistic results due to random splitting on both types of data sets and show how this bias is controlled through well-constructed data splits that take sequence, pocket structure, or protein−ligand interaction similarity into account. Further, we explain the presence of protein-only and ligand-only biases in PDBbind data set by comparing the performances of the same experiments on both structure-based and sequence-based data sets. The idea behind this work is to study the influence of the splitting methods employed as they eventually determine the generalizability of the models developed on these data sets. Thus, in this work, we do not compare new methods on currently existing benchmarks but rather analyze the performance of previously published models across nontrivial splitting methodologies on the same popular data sets they were initially trained on.

Through this analysis, we provide reasoning as to why deep learning models do not perform as well in virtual screening experiments as expected. We also suggest solution that address biases in both data set construction and model design. With this work, we aim to provide the community several pointers on possible paths forward for the development of superior protein−ligand scoring functions.

■ METHODS

In this work, we use previously published methods and data sets designed for the protein−ligand binding affinity prediction to showcase confounding factors and biases in these deep learning methods. In the subsequent subsections, we describe the open-source data sets and methods used to obtain our results.

Data Sets and Preprocessing. The data sets used in this study have been predominantly utilized for binding affinity prediction. They provide binding affinity values for protein−

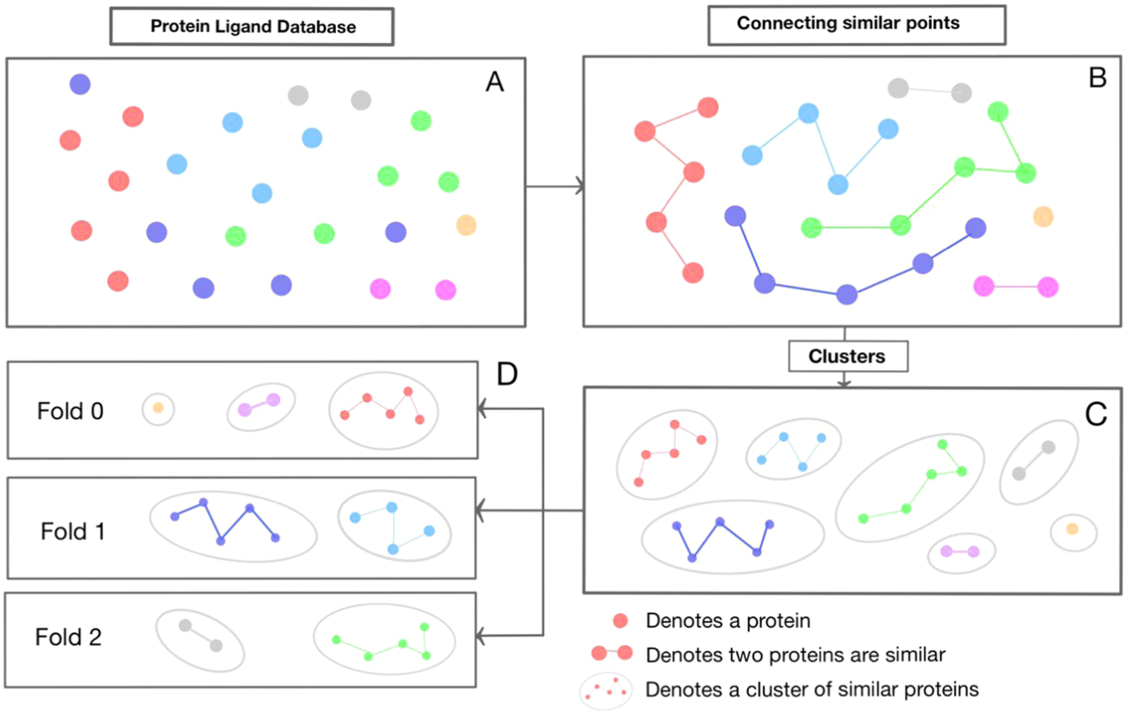

Protein Ligand Database
A
Connecting similar points
B
Clusters
C
D
Fold 0
Fold 1
Fold 2
Denotes a protein
Denotes two proteins are similar
Denotes a cluster of similar proteins

Figure 1. Schematic representation of a typical clustered cross-validation split. (A) Representation of the entire data set, where similar proteins or data points are of the same color. (B) Links are established between similar proteins or data points to form connected components. (C) All of the connected components are grouped together to form clusters of similar data points. (D) After all of the clusters are formed, they are allocated to one of three different folds while trying to ensure that each fold has a similar number of data points.

ligand complexes along with information on the protein and ligand—either in the form of one-dimensional (1D) sequences (amino acid sequences and SMILES strings) or in their cocrystal 3D structures:

- **PDBbind v.2019**[17] provides binding affinity values for protein−ligand cocrystal structures present in the PDB database. The 2019 version, PDBbind contains a total of 17,652 complexes. These complexes are split into three subsets, the general set, the refined set, and the core set. The refined set contains a subset of high-quality complexes with a resolution better than 2.5 Å and reported K_d or K_i binding affinity values. The core set is a subset of the refined set that was introduced as part of the CASF competitions to benchmark new protein−ligand scoring functions. Any complex not part of the refined set is a part of the general set. We only used protein−ligand pairs in this data set where the molecular weight of the ligand is <1000 Dalton for our analysis.

  Most machine learning binding affinity predictors on the PDBbind data set have been trained on only the refined set or on both the general and refined sets and benchmarked on the 2016 version of the CASF benchmark set. However, to take advantage of all of the available complexes in the data set, we construct the core set by taking a union of the 2007, 2013, and 2016 versions of the CASF benchmarks resulting in a core set of size 540. We benchmarked the Pafnucy model[34] on this core set for our analysis.

  We also analyze model performances for methods that were designed to work with protein residue sequences and ligand SMILES strings. These methods were benchmarked on the following data sets:

- **Davis:** the Davis kinase binding affinity data set[27] provides dissociation constant (K_d) values for all protein−ligand pairs formed by pairing 68 kinase inhibitors with 442 kinases.
- **KIBA:** the kinase inhibitor bioactivity (KIBA) data set[28] consists of 52,498 ligands and 467 target proteins. These proteins and ligands form a total of 246,088 protein−ligand pairs. The data set provides KIBA scores that are created to maintain consistency between different measures of binding affinity (K_D, inhibition constant (K_I), and half-maximal inhibitory concentration (IC_{50})) for each protein−ligand pair.

Most machine learning studies use these data sets, trained and tested models, by randomly splitting the data set into different cross-validation folds. To reproduce the performance of the methods used in this study, we train the models on a random 80:20 train−test split of the data sets.

Cross-Validation Folds. For all of the data sets (PDBbind and sequence-based data sets), we construct cross-validation folds that control the level of similarity between test and train sets, later referred to as clustered cross-validation (CCV) folds, and compare the performance on these folds to that of performance on core set (for PDBbind) and random splits (for sequence-based methods).

We create clustered cross-validation splits by following similar methodologies as Francoeur et al.[23] In general, clustered cross-validation data splits are created by first splitting the data into several disjoint sets or simply clusters based on some measure of similarity and then allocating these clusters into k folds (Figure 1). The clusters are allocated sequentially, i.e., we iterate through the clusters, and during each iteration the cluster in question is allocated to the fold with the lowest number of data points at that time. Once these folds are created, we perform *k*-fold cross validation where *k*-1 folds are used for training and 1-fold is used for testing the model. The method employed in the study is summarized in Figure 1. For PDBbind, we create splits based on sequence, pocket structure, and protein−ligand interaction similarities, while for the rest of the data sets we create splits based on sequence similarity.

For all of the data set splits, created using sequence similarity, we ensure that the protein similarity between any two data points of different folds is lesser than 50%. To implement this, we conduct pairwise global sequence alignment at a 100% sequence coverage and take the resulting sequence similarity into account. However, proteins with very different sequences may still have the same interaction mechanisms with the same ligand due to common substructures present in the proteins. Therefore, to control such a bias further, we apply an additional constraint that if any data points of two different folds have ligand Tanimoto similarity greater than 90%, then their protein similarity is kept below 40%. For the Davis data set, we create the folds only based on 40% protein similarity with no ligand similarity constraint as all ligands are paired with all proteins in the Davis data set. Notice that the fold sizes for KIBA are slightly disproportionate. Fold 0 is larger than the other folds as it contains a very large cluster of similar data points. However, we also notice that each fold has enough number of data points for robust cross validation and therefore we conduct our experiments with the created folds.

For PDBbind, we also create splits based on pocket structure and protein−ligand interaction similarity. To measure pairwise pocket structure similarity, we utilize the PocketMatch software[35] that has been designed to compare and identify identical binding sites on protein structures. We use a cutoff of 0.7 on the PocketMatch P-max score as a similarity threshold for our splits. For protein−ligand interactions, we first calculate the protein−ligand extended connectivity (PLEC).[36] Fingerprints of size 16,384 and a Tanimoto similarity of 0.25 as a threshold of pairwise similarity were used. The number of protein−ligand complexes per fold for each data set is reported in Table 1.

Table 1. Number of Protein−Ligand Pairs per Fold for Each Data Set[a]

data set	type of similarity	fold 0	fold 1	fold 2
PDBbind	protein sequence similarity	5536	5536	5536
PDBbind	PLEC fingerprint similarity	5536	5536	5536
PDBbind	PocketMatch similarity	5711	5394	5485
Davis	protein sequence similarity	10,648	10,648	9996
KIBA	protein and ligand sequence similarities	50,137	34,065	34,042

[a] For PocketMatch similarity, a few data points have been removed due to errors that arose while calculating similarity using PocketMatch.

Models. We analyzed the performance of the following models on the created data splits:

- **DeepDTA**[31] is a deep learning model, which uses only the sequence information of both proteins and ligands to predict the binding affinity of the interactions. The model uses separate 1D CNN blocks to extract ligand and protein features from the SMILES notation of ligands and the residue sequence of the proteins, respectively. It subsequently concatenates the extracted information and feeds it to a fully connected network for the prediction of binding affinity. DeepDTA was implemented using the open-source drug discovery library DeepPurpose.[30]
- **SimCNN-DTA**[33] uses a 2D CNN architecture for binding affinity prediction. First, drug−drug similarity and target−target pairwise similarity matrices are constructed using the Tanimoto coefficient and normalized Smith−Waterman algorithm with all of the data points in the training set. Then, for each training point, the outer product of the corresponding target and ligand similarity vectors (column vectors in the previously constructed matrices) are taken as input to a model consisting of a two-dimensional (2D) convolutional neural network. SimCNN-DTA showed a better ranking capability as compared to DeepDTA. Since SimCNN-DTA implementation is not open-sourced, we have reimplemented the method. We obtain similar results to the published values indicating a correct reimplementation of the method. The code is available and can be found at https://github.com/devalab/Protein-Ligand-Dataset-Bias/tree/master/sim-CNNDTA.
- **GraphDTA**[32] uses a 2D graph representation of the ligand molecule and the residue sequence of the protein. This study compares several graph neural network (GNN) architectures, i.e., graph convolution network (GCN), graph attention network (GAT), graph isomorphism network (GIN), and GAT−GCN for ligand input. A convolution neural network (CNN) is used to extract features from target sequence. The final ligand and target features from the GNN and CNN, respectively, are concatenated and used to predict the binding affinity via fully connected layers. For this study, we used the graph isomorphism network (GIN) proposed in the study,[32] as it showed better performance than DeepDTA for both the Davis and KIBA data sets in the original publication. The open-sourced code for this method was used to train the GIN model. The source code for the model is available at https://github.com/thinng/GraphDTA.
- **Pafnucy**[34] is a deep neural network that consists of 3D CNN blocks that extract the protein−ligand interaction information by using the 3D representation of the complex structure as the input. The protein−ligand complex is represented in a 3D grid with a 1 Å resolution. The input is represented as a four-dimensional (4D) tensor such that each point is represented by the three coordinates and a feature vector of 19 different atomic features like atom type, hybridization, partial charge, etc. The feature map of the input is obtained by a series of 3D CNNs, which is followed by a fully connected layer to make the final binding affinity prediction. The open-source code for Pafnucy available at https://gitlab.com/cheminfIBB/pafnucy was used to obtain results for the model.
- **Smina**[37] is a docking software that utilizes the Autodock Vina[38] scoring function. In this study, we use smina on PDBbind data set to function as baseline for comparison with deep learning-based scoring functions used in this study. The smina scores are converted to the P_k binding affinity score using

$$P_{\mathrm{k}} = -\log_{10}(e^{(\mathrm{smina}/T \times R)})$$

where $T = 295$ K is the temperature, $R = 1.9872 \times 10^{-3}$ kcal mol^{-1} K^{-1} is the ideal gas constant, and smina is the Smina score.

All of the data sets and models used in this study are summarized in Table 2 for quick reference. We also define ligand-only and protein-only models as those models that have been trained only ligand or only protein information. This is done by masking out either protein information (ligand-only) or ligand information (protein-only) in the input. For example, this is done in DeepDTA by simply masking out one of the two CNN blocks corresponding to either ligand or protein information.

Table 2. Data Sets and Various Models Used in This Study

type	data set	no. of complexes	models used
structure-based	PDBbind	16,608	Pafnucy Smina
sequence-based	Davis	30,056	DeepDTA GraphDTA SimCNN-DTA
	KIBA	118,254	DeepDTA GraphDTA SimCNN-DTA

All of the models used in this study have been retrained using the exact procedure, as suggested in the respective original studies. All models were also retrained for protein-only/ligand-only with relevant information masked out from the input.

RESULTS AND DISCUSSION

In this work, we conduct a series of experiments to study the various types of biases present in protein−ligand data sets that affect deep learning model's performances in binding affinity prediction and virtual screening. We describe our results in more detail in the subsequent sections.

Overoptimistic Performance in Random Splitting. Random splitting of the data set into training set and testing set is the most commonly used method to train and evaluate ML models. Random splitting tries to achieve the objective of distributing the data points in a data set into train and test sets while also maintaining the overall generality of the data points in both sets. This is done so that the data distribution is even in test and train sets and one can obtain a reasonable view of model generalizability. Hence, when such random splitting is employed for splitting biological data sets, it is certain that data points from similar groups of proteins are present in both train and test sets.

It is well known in biology that similar sequences lead to similar structures and therefore similar interaction mechanisms with druglike molecules. Due to this phenomenon, random splitting is not an ideal way of splitting the data as it leads to similar sequences or structures being present in both train and test sets, thereby causing a sort of data leakage between the two. In a way, this type of benchmarking could also be seen as awarding overfitting due to the high similarity in data points between the two sets. This data leakage or internal bias can be controlled by constructing cross-validation splits that handle the level of sequence similarity between the folds (referred to as clustered cross-validation splits). This provides a better outlook on the model's generalizability on unseen protein−ligand complexes.

In Table 3, we report the performance of the Pafnucy model on the PDBbind data set for testing on core vs clustered cross validation (CCV). The core set in PDBbind can be considered a special case of random splits as it was created by clustering sequences in the set and collecting representative sequences from each cluster. We created clustered cross-validation splits by following similar methodologies as Francoeur et al.[23] (see the Data Sets and Preprocessing section). We note that while similar studies have been done on PDBbind before,[14,23,39] we provide these results with the dual purpose of showcasing data splitting methods that should be adopted for all protein−ligand tasks (whether it be on structure-based or sequence-based data sets) and providing a comparison in performance drop between structure-based and sequence-based methods on CCV splits. As can be seen from values reported in Table 3, there is a significant reduction of at least 0.14 in Pearson R when we go from the core set to the CCV split for Pafnucy. The Smina Pearson R results also tend to reduce a bit for the clustered cross-validation splits due to the inclusion of IC_{50} values in the general set. We also noticed that for very few data points (11 in total), Smina predicted P_{k} values that were lower than −20. Since such outlier scores can arise due to problems in molecule files and lead to a misleading final average Pearson R scores, we have decided to not include them in our correlation calculations. For Pafnucy, the RMSE remains about the same for sequence-based clustering but increases on protein−ligand interaction and protein structure-based splits. On sequence-based splits, the model tends to predict values closer to the mean of the data set; however, it does not differentiate between different data points well, as evidenced by the reduction in Pearson correlation. In general, PLEC fingerprints and PocketMatch tend to form stricter data splits to evaluate model generalizability, with PocketMatch being the suggested method, especially for models that are independent of the residue sequence. Yet, the model still seems to learn protein−ligand interactions to some extent as it still has a positive correlation for all of the data split.

Table 3. Performance of Structure-Based Method on PDBbind for All Types of Splits Created in This Study

type	Pearson R^{P}	RMSE^{P}	Pearson R^{S}	RMSE^{S}
general-core split	0.73	1.53	0.38	2.65
sequence CCV split	0.59	1.51	0.35	2.52
PLEC CCV split	0.52	1.62	0.35	2.52
PocketMatch CCV split	0.50	1.59	0.29	2.52

^{P}Denotes the results for Pafnucy. ^{S}Denotes results for Smina.

Next, we compared the performance of popular sequence-based models like DeepDTA, SimCNN-DTA, and GraphDTA on random splits and CCV splits. The results are reported in Table 4. The performance measured by mean-squared error (MSE) and concordance index (CI) closely resembles the published results for random splits indicating correct reimplementations of all of the methods. From the reported values, it can be seen that there is a significant reduction in performance evidenced by poorer values for all metrics. The MSE increases by at least 2 times on going from random splitting to clustered cross validation. The Pearson correlation between the predicted affinity values and the true affinity values is also very low (sometimes even lower than 0.5),

https://doi.org/10.1021/acsomega.2c06781

Table 4. Comparison of Performances on Random vs CCV Splits for Sequence-Based Models

on Davis	RMSE^R	RMSE^C	CI^R	CI^C	Pearson R^R	Pearson R^C
DeepDTA	0.489	0.790	0.884	0.763	0.841	0.477
SimCNN-DTA	0.548	0.825	0.855	0.739	0.808	0.456
GraphDTA(GIN)	0.478	0.854	0.886	0.670	0.844	0.314
On KIBA	RMSE^R	RMSE^C	CI^R	CI^C	R^R	R^C
DeepDTA	0.423	0.764	0.860	0.673	0.864	0.497
SimCNN-DTA	0.484	0.729	0.827	0.657	0.822	0.422
GraphDTA(GIN)	0.383	0.721	0.883	0.658	0.885	0.511

^{R}Represents performance on random split. ^{C}Represents performance on clustered cross-validation split.

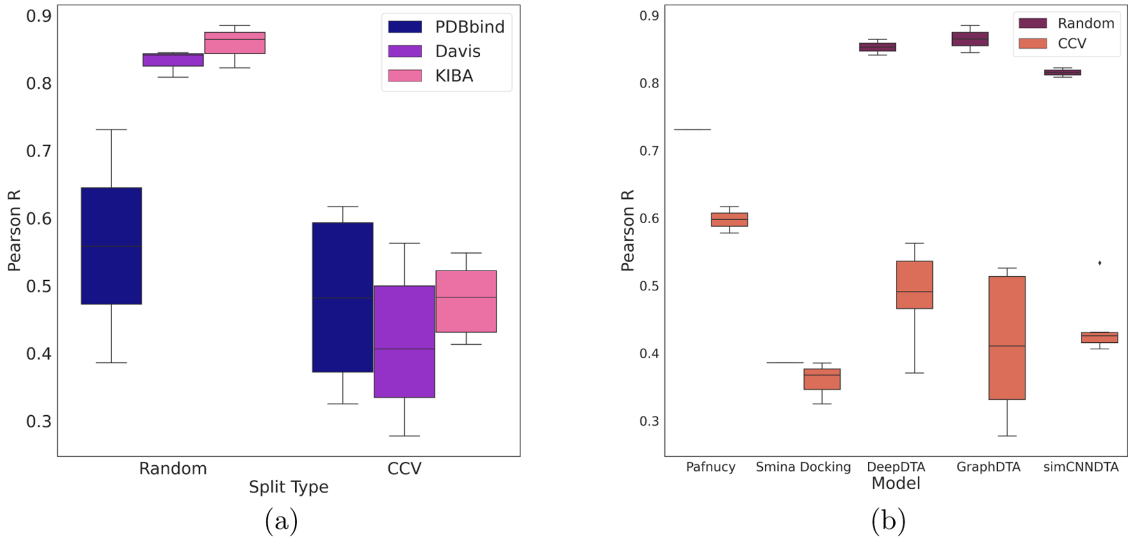


Figure 2. Random splitting vs sequence-based clustered cross-validation performance for all of the models across various data sets used in this study (left) and random splitting vs sequence similarity-based clustered cross-validation performance comparison for all methods (right).

indicating that there is not much of a linear correlation between the two values. This clearly indicates the level of optimism random splits provide for these data sets as compared to more practical situations when the model is exposed to unseen protein−ligand pairs.

In Figure 2, we showcase the results of random vs CCV for both structure-based (Pafnucy and Smina on PDBBind data set) and sequence-based methods (DeepDTA, GraphDTA, and SimCNN-DTA on Davis and KIBA data sets) for a visualization of overall performances. Figure 2a clearly showcases that there is a much greater drop in performance when switching from random splitting to sequence similarity CCV for models trained and tested on Davis and KIBA data sets (sequence-based) when compared to those trained and tested on PDBbind (structure-based). This clearly indicates that structure-based methods (Pafnucy) are more generalizable to novel protein−ligand targets evidenced by their overall performance in the CCV splits. This is also in agreement with the intuition that binding affinity depends on the 3D interactions between the two molecules and therefore structural information is required to make predictions for novel complexes. Figure 2b compares CCV and random splitting performance for all of the individual methods used in this study. It is noticeable that there is a significant drop in CCV for deep learning models when compared to their random splitting counterparts, while it remains largely the same for Smina. Since Smina is a non-deep learning scoring function, it remains unaffected by the type of splits used to evaluate the performance. On the other hand, you can clearly see the level of overoptimism deep learning functions on the random split.

From Figure 2b, we notice that Smina results almost remain unchanged and the obtained Pearson R is very low. This suggests that although deep learning models on 3D structures show overoptimistic performance in random splitting, their CCV performance is still better than a simple classical scoring function like Smina. It is also worth noting that Smina scores perform better on the core or refined sets as compared to the entire PDBbind data set as the general set also contains datapoint with IC$_{50}$ target values.

From the same figure, it is visible that DeepDTA has better performance on CCV splitting as compared to GraphDTA. Note that GraphDTA showcased the best result for random splitting but has done much worse on CCV splitting. As a result, the model seems to be rewarding overfitting due to similar protein−ligand pairs being present in the train and test sets. Finally, we see narrow distributions for Pafnucy and SimCNN-DTA on CCV splits, suggesting that the model performances are more consistent across different cross-validation folds.

While structure-based deep learning methods show quite a bit of promise for protein−ligand scoring, they are still limited with further biases present in the data set itself. Due to such biases, they may not perform well in virtual screening experiments. We discuss such aspects of the data sets in subsequent sections.

Presence of Only Stable Protein−Ligand Pairs Leads to Protein-Only and Ligand-Only Biases. A previously reported artifact associated with the structure-based protein–ligand data sets is the presence of protein-only and ligand-only biases[10,23] This kind of bias entails that models trained on ligand-only or protein-only information have comparable performance to that of a model trained on the full complex. This indicates that the model may not be focusing on learning appropriate protein–ligand interactions responsible for affecting the binding affinity of the complex to attain high-level performance. In this section, we draw parallels between sequence- and structure-based methods to infer the origins of this bias and discuss possible approaches to control it.

We first see the ligand-only and protein-only performances for the Pafnucy model on the general-refined-core split. The results are reported in Table 5.

Table 5. Ligand-Only and Protein-Only Performance Comparisons with the Full-Complex Performance on PDBbind by Pafnucy in General-Refined-Core Split

PDBbind	Pearson R	RMSE
full complex	0.73	1.53
ligand-only	0.62	1.75
protein-only	0.64	1.73

The results indicate that the ligand-only and protein-only performances are very comparable to the performance of the full complex, with the Pearson correlation dropping only by around 0.1. Both the ligand-only and protein-only Pearson R are above 0.6, showing that the model is able to attain such performance even without learning any information from the protein–ligand interaction. Hence, we can conclude that the model does not necessarily focus on protein–ligand interactions for making binding affinity predictions.

Next, we check the protein-only and ligand-only performances of sequence-based models on the Davis and KIBA data sets. The performance of the models is reported in Tables 6 and 7. As we can see, the performance for protein-only and ligand-only drops significantly, even for random splits. The performance is especially poor in the Davis set. The Pearson correlation for these models is below 0.5 for both protein-only and ligand-only, which shows that there is a very poor relationship between predicted and experimental values. This clearly indicates the absence of such a bias in these data sets.

Table 6. Ligand-Only and Protein-Only Performance Results of DeepDTA on Davis with Random Splitting Methods

Davis	Pearson R	CI	RMSE	MSE
full complex	0.84	0.88	0.48	0.23
protein-only	0.32	0.65	0.85	0.73
ligand-only	0.48	0.75	0.78	0.61

Table 7. Ligand-Only and Protein-Only Performances of DeepDTA on KIBA with Random Splitting

KIBA	Pearson R	CI	RMSE	MSE
full complex	0.86	0.86	0.42	0.17
protein-only	0.36	0.68	0.77	0.61
ligand-only	0.61	0.70	0.65	0.43

To explore this absence of bias in the sequence-based data sets further, we plot the correlation plots of predicted vs experimental values for the Davis set in Figure 3. We can see multiple dotted lines parallel to the x axis for both ligand-only and protein-only experiments. This is because while the model predicts only a single value for a protein or a ligand, the data set itself contains multiple experimental values for the same protein or ligand. These values are also spread across the *x* axis, indicating that the data set contains both strongly binding and poorly binding protein–ligand pairs.

This is in contrast to PDBbind where structures are only available for strongly binding protein–ligand pairs (stable) as cocrystal structure resolution is only possible for such pairs. Therefore, models trained on PDBbind are often not exposed to poor binding affinity values (unstable complexes) associated with a ligand or a protein. It can be inferred that this can be the primary cause of protein-only and ligand-only biases in this data set. Furthermore, this also points to a more consequential bias in the data set as models trained on it are never exposed to unstable protein–ligand pairs with poor binding affinities. This bias itself could cause low performance in virtual screening as the model is expected to extrapolate knowledge from stable protein–ligand pairs to unstable pairs, which is not entirely trivial. This hypothesis is further supported by relatively low performance in virtual screening (slightly better than traditional baseline) by deep learning models predicting binding affinity on the DUD-E and LIT-PCBA data sets.[24]

The solution to this problem seems simple, unstable or poorly binding protein–ligand pairs could be added to structure-based binding affinity data sets to get a better guarantee of model's performance in virtual screening. However, this is impractical as the structure resolution of nonbinding protein–ligand pairs is not trivial.

In an attempt to explore alternate solutions to eliminating protein-only and ligand-only biases, we start with checking Pafnucy performance on PDBbind cross-validation splits. The results of these experiments are reported in Table 8. From the results in the table, it is apparent that the ligand-only and protein-only performances drop (especially so for protein-only) with RMSE as high as 1.71 and 1.91 for each. It is possibly more prominent for protein-only because of the 50% similarity criteria we used for creating the CCV splits. Hence, CCV splitting can, in general, be a more robust technique that handles the protein-only and ligand-only biases to an extent and thus guarantees a more generalizable model.

One could also make architectural changes to ML models to enforce focusing on protein–ligand interactions. This would ensure a reduction in such a bias as the resulting output would be an outcome of the interactions and not the 3D structure of the molecules. OnionNet[22] employed such a method for the prediction of binding affinity by taking protein–ligand pairs at different distance thresholds. Jones et al.[21] also reported lower protein-only and ligand-only biases by taking a combination of graph-based and spatial features. However, while these solutions handle the ligand-only and protein-only biases to an extent, they are not guaranteed to do well in virtual screening as they are not exposed to nonbinding protein–ligand pairs, which they may face in virtual screening. This is an expected limitation for all data-driven methods.

Other techniques could also be utilized to handle this problem. Docking could be used to obtain more unstable protein–ligand structures. Alternatively, augmenting machine learning models with confidence also seems like a possible

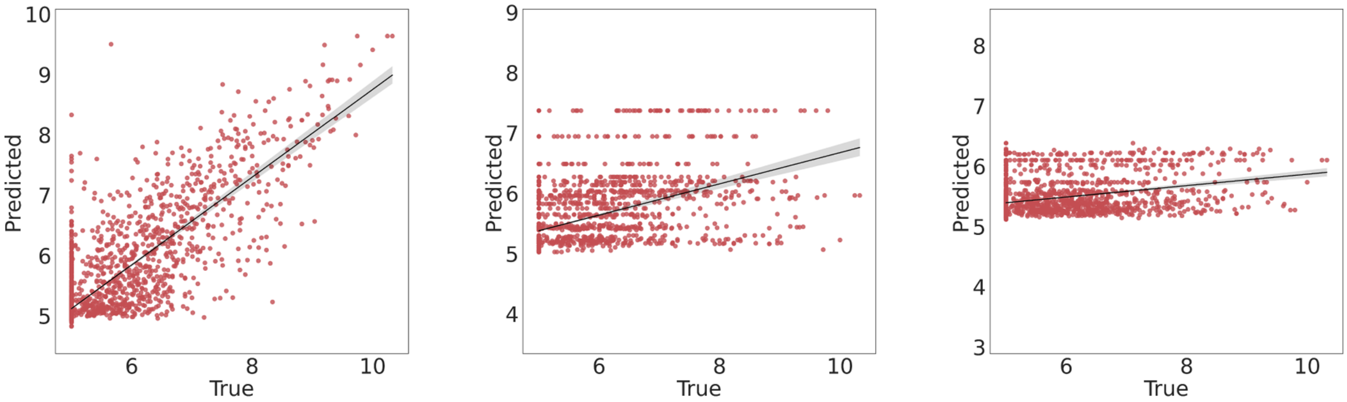


Figure 3. Correlation plots for predicting binding affinity using full complex, ligand-only, and protein-only on Davis data set for DeepDTA with random splitting.

Table 8. Ligand-Only and Protein-Only Performance Comparisons with Full Complex by Pafnucy on PDBbind with CCV Splitting

PDBbind	Pearson R	MAE	RMSE	SD
full complex	0.60	1.19	1.51	1.51
ligand-only	0.47	1.35	1.71	1.71
protein-only	0.34	1.52	1.91	1.91

route as poor confidence is usually well correlated with erroneous predictions. Finally, large-scale projects may be required to develop a bias-free protein−ligand data set that could be used by deep learning for binding affinity prediction. The final objective is to improve the overall virtual screening practices, regardless of how well these deep learning models are developed. Therefore, handling and designing the negative samples in these data sets is a key issue that needs to be addressed for the practical usage of such deep learning models.

CONCLUSIONS

The utility of deep learning models for biology and drug design is limited by their performance to unseen data distributions; therefore, it is important to get a good understanding of model generalizability by using appropriate data splits and tackling the different kinds of biases induced in practice.

We utilize popular and published methods to make our arguments and showcase results. We show a significant drop in the perceived performance of binding affinity prediction by deep learning architectures when controlling the levels of protein sequence similarity in the test and train sets, while non-deep learning scoring functions remain almost unaffected. Therefore, we advocate for the usage of the same splitting technique for benchmarking future machine learning models developed on these data sets. We showcase the effect of creating clustered cross-validation splits based on sequence, structure, and protein−ligand interaction similarity and conclude that the latter two methods provide much stricter data splits for assessing model generalizability. On comparing sequence-based methods and structure-based methods on sequence similarity clustered cross-validation splits, we conclude that structure-based methods have better generalizability. We discuss the presence of ligand-only and protein-only biases present in PDBbind and provide an explanation of how it is induced due to the absence of nonbinding protein−ligand pairs in the data set. We argue that this may indicate a more prominent bias of having only stable protein−ligand pairs in the data set, which, in turn, could have an effect on the ranking power of the developed deep learning scoring function for virtual screening.

In conclusion, we examined several sources of biases in protein−ligand data sets that affect deep learning performances in binding affinity prediction and virtual screening. Alongside, we suggest different machine learning techniques that could potentially be used to control such biases. We also point out the importance of community efforts for developing larger data sets that take into account such biases. With this, we encourage the design of deep learning methods that take into account such biases and could be incorporated into structure-based drug design pipelines for virtual screening.

AUTHOR INFORMATION

Corresponding Author

U. Deva Priyakumar − *International Institute of Information Technology, Hyderabad 500 032, India;* orcid.org/0000-0001-7114-3955; Email: deva@iiit.ac.in

Authors

Ganesh Chandan Kanakala − *International Institute of Information Technology, Hyderabad 500 032, India;* orcid.org/0000-0002-4469-1857

Rishal Aggarwal − *International Institute of Information Technology, Hyderabad 500 032, India*

Divya Nayar − *Department of Materials Science and Engineering, Indian Institute of Technology Delhi, New Delhi 110016, India;* orcid.org/0000-0001-8569-7633

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.2c06781

Notes

The authors declare no competing financial interest.
The code and the data set splits created can be accessed from the link https://github.com/devalab/Protein-Ligand-Dataset-Bias.

ACKNOWLEDGMENTS

The authors thank IHub-Data and KCIS, IIIT Hyderabad, and DST-SERB (Grant No. CRG/2021/008036) for the financial support. D.N. thanks DST-SERB for the Startup Research Grant (Grant No. SRG/2021/000804).

■ REFERENCES

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

(2) Lill, M.*In Silico Models for Drug Discovery*; Kortagere, S., Ed.; Humana Press: Totowa, NJ, 2013; pp 1−12.

(3) He, K.; Zhang, X.; Ren, S.; Sun, J. In *Deep Residual Learning for Image Recognition*, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Cornell University, 2016; pp 770−778.

(4) Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; Polosukhin, I.Attention Is All You Need2017https://arxiv.org/abs/1706.03762.

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

(6) Ramsundar, B.Molecular Machine Learning With DeepChem. Ph.D. thesis; Stanford University, 2018.

(7) Nguyen, D. D.; Wei, G.-W. AGL-Score: Algebraic Graph Learning Score for Protein−Ligand Binding Scoring, Ranking, Docking, and Screening. *J. Chem. Inf. Model.* **2019**, *59*, 3291−3304. PMID: 31257871.

(8) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *J. Med. Chem.* **2012**, *55*, 6582−6594.

(9) Rohrer, S. G.; Baumann, K. Maximum Unbiased Validation (MUV) Data Sets for Virtual Screening Based on Pubchem Bioactivity Data. *J. Chem. Inf. Model.* **2009**, *49*, 169−184.

(10) Zheng, S.; Li, Y.; Chen, S.; Xu, J.; Yang, Y. Predicting Drug-Protein Interaction Using Quasi-Visual Question Answering System. *Nat. Mach. Intell.* **2020**, *2*, 134−140.

(11) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-Ligand Scoring With Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57*, 942−957.

(12) Torng, W.; Altman, R. B. Graph Convolutional Neural Networks for Predicting Drug-Target Interactions. *J. Chem. Inf. Model.* **2019**, *59*, 4131−4149.

(13) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden Bias in the Dud-E Dataset Leads to Misleading Performance of Deep Learning in Structure-Based Virtual Screening. *PLoS One* **2019**, *14*, No. e0220113.

(14) Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Front. Pharmacol.* **2020**, *11*, No. 69.

(15) Sieg, J.; Flachsenberg, F.; Rarey, M. Need of Bias Control: Evaluating Chemical Data for Machine Learning in Structure-Based Virtual Screening. *J. Chem. Inf. Model.* **2019**, *59*, 947−961.

(16) Tran-Nguyen, V.-K.; Jacquemard, C.; Rognan, D. LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening. *J. Chem. Inf. Model.* **2020**, *60*, 4263−4273.

(17) Wang, R.; Fang, X.; Lu, Y.; Yang, C.-Y.; Wang, S. The PDBbind database: Methodologies and Updates. *J. Med. Chem.* **2005**, *48*, 4111−4119.

(18) Berman, H. M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T. N.; Weissig, H.; Shindyalov, I. N.; Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235−242.

(19) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: molecular docking with deep learning. *J. Cheminf.* **2021**, *13*, No. 43.

(20) Jiménez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G. K deep: Protein-Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.* **2018**, *58*, 287−296.

(21) Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. D.; Kirshner, D.; Wong, S. E.; Lightstone, F. C.; Allen, J. E. Improved Protein-Ligand Binding Affinity Prediction With Structure-Based Deep Fusion Inference. *J. Chem. Inf. Model.* **2021**, *61*, 1583−1592.

(22) Zheng, L.; Fan, J.; Mu, Y. OnionNet:A Multiple-Layer Intermolecular-Contact-Based Convolutional Neural Network for Protein-Ligand Binding Affinity Prediction. *ACS Omega* **2019**, *4*, 15956−15965.

(23) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 4200−4215.

(24) Sunseri, J.; Koes, D. R. Virtual Screening with Gnina 1.0. *Molecules* **2021**, *26*, No. 7369.

(25) Trott, O.; Olson, A. J. AutoDock Vina: Improving the Speed and Accuracy of Docking With a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* **2010**, *31*, 455−461.

(26) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: Molecular Docking With Deep Learning. *J. Cheminf.* **2021**, *13*, No. 43.

(27) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive Analysis of Kinase Inhibitor Selectivity. *Nat. Biotechnol.* **2011**, *29*, 1046−1051.

(28) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets: A Comparative and Integrative Analysis. *J. Chem. Inf. Model.* **2014**, *54*, 735−743. PMID: 24521231.

(29) Huang, K.; Fu, T.; Gao, W.; Zhao, Y.; Roohani, Y.; Leskovec, J.; Coley, C. W.; Xiao, C.; Sun, J.; Zitnik, M.Therapeutics Data Commons: Machine Learning Datasets and Tasks for Drug Discovery and Development2021https://arxiv.org/abs/2102.09548.

(30) Huang, K.; Fu, T.; Glass, L. M.; Zitnik, M.; Xiao, C.; Sun, J. DeepPurpose: ADeep Learning Library for Drug-Target Interaction Prediction. *Bioinformatics* **2021**, *36*, 5545−5547.

(31) Öztürk, H.; Ozkirimli, E.; Ozgur, A. DeeDTA: Deep Drug-Target Binding Affinity Prediction. *Bioinformatics* **2018**, *34*, i821−i829.

(32) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: Predicting Drug−Target Binding Affinity With Graph Neural Networks. *Bioinformatics* **2020**, *37*, 1140−1147.

(33) Shim, J.; Hong, Z.-Y.; Sohn, I.; Hwang, C. Prediction of Drug-Target Binding Affinity Using Similarity-Based Convolutional Neural Network. *Sci. Rep.* **2021**, *11*, No. 4416.

(34) Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P. Pafnucy−A Deep Neural Network for Structure-Based Drug Discovery. *Bioinformatics* **2017**, *1050*, 19.

(35) Nagarajan, D.; Chandra, N. In *PocketMatch (Version 2.0): A Parallel Algorithm for the Detection of Structural Similarities between Protein Ligand Binding-Sites*, 2013 National Conference on Parallel Computing Technologies (PARCOMPTECH); IEEE, 2013; pp 1−6.

(36) Wójcikowski, M.; Kukiełka, M.; Stepniewska-Dziubinska, M. M.; Siedlecki, P. Development of a Protein-Ligand Extended Connectivity (PlEC) Fingerprint and Its Application for Binding Affinity Predictions. *Bioinformatics* **2018**, *35*, 1334−1341.

(37) Koes, D. R.; Baumgartner, M. P.; Camacho, C. J. Lessons Learned in Empirical Scoring With Smina From the CSAR 2011 Benchmarking Exercise. *J. Chem. Inf. Model.* **2013**, *53*, 1893−1904. PMID: 23379370.

(38) Eberhardt, J.; Santos-Martins, D.; Tillack, A. F.; Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *J. Chem. Inf. Model.* **2021**, *61*, 3891−3898. PMID: 34278794.

(39) Su, M.; Feng, G.; Liu, Z.; Li, Y.; Wang, R. Tapping on the Black Box: How Is the Scoring Power of a Machine-Learning Scoring Function Dependent on the Training Set? *J. Chem. Inf. Model.* **2020**, *60*, 1122−1136.