# On the frustration to predict binding affinities from protein-ligand structures with deep neural networks

Mikhail Volkov, Joseph-André Turk, Nicolas Drizard, Nicolas Martin, Brice Hoffmann, Yann Gaston-Mathé, Didier Rognan

# On the frustration to predict binding affinities from protein-ligand structures with deep neural networks.

Mikhail Volkov,[†] Joseph-André Turk,[≠] Nicolas Drizard,[≠] Nicolas Martin,[≠] Brice Hoffmann,[≠] Yann Gaston-Mathé[≠] and Didier Rognan[†]*

† Laboratoire d'innovation thérapeutique, UMR7200 CNRS-Université de Strasbourg, 74 route du Rhin, 67400 Illkirch, France.

≠ Iktos, 65 rue de Prony, 75017 Paris, France.

* To whom correspondence should be addressed (phone: +33 3 68 85 42 35, fax: +33 3 68 85 43 10, email: rognan@unistra.fr)

**ABSTRACT**

Accurate prediction of binding affinities from protein-ligand atomic coordinates remains a major challenge in early stages of drug discovery. Using modular message passing graph neural networks describing both the ligand and the protein in their free and bound states, we unambiguously evidence that explicit description of protein-ligand non-covalent interactions does not provide any advantage with respect to ligand or protein descriptors. Simple models, inferring binding affinities of test samples from that of the closest ligands or proteins in the training set, already exhibit good performances suggesting that memorization largely dominates true learning in the deep neural networks. The current study suggests considering only non-covalent interactions while omitting their protein and ligand atomic environments. Removing all hidden biases probably require much denser protein-ligand training matrices and a coordinated effort of the drug design community to solve the necessary protein-ligand structures.

## INTRODUCTION

Predicting absolute binding free energies (affinities) from three-dimensional atomic coordinates of protein-ligand complexes remains one of the grand challenges of computational chemistry.[1] For example, drug discovery would immediately benefit from key advances in this topic, by better triaging potentially interesting molecules among virtual screening hits[2-3] and proposing viable analogs in emerging ultra-large chemical spaces[4] for hit to lead optimization. With the ever increasing amount of high-resolution experimentally-determined protein-ligand structures,[5] binding affinity prediction algorithms have switched from physics-based[6] to empirical scoring functions,[7] and in the last years to machine learning[8] and deep learning methods.[9-10] The latter category of descriptor-based scoring functions has notably led to numerous protein-ligand affinity models [11-38] (see a non-exhaustive list **Table S1**), notably because deep learning does not require explicit descriptor engineering and is ideally suited to find hidden non-linear relationships between three-dimensional protein-ligand structures and binding affinity. The first deep neural networks (DNNs) to predict binding affinities were convolutional neural networks (CNN) reading a protein-ligand complex as an ensemble of grid-based voxels with multiple channels corresponding to pharmacophoric properties.[12, 15-16] The CNN architecture is relatively inefficient from a computational point of view since most of the voxels do not carry any relevant information. Moreover, the search for the best possible hyperparameters is very demanding with respect to memory usage and cpu time. Last, the same object must be presented in multiple orientations in a three-dimensional (3D) grid to remove the dependency to the initial atomic coordinates. To overcome these issues and speed-up the training process, most recently developed DNNs reads inputs in the form of a molecular graph[39] where nodes are represented by atoms and edges by bonds and/or non-covalent intra- and intermolecular interactions. Atoms and edges are embedded with user-defined atomic and/or pharmacophoric properties, enabling all graph components to be updated according to their surroundings all along the network during the training phase.

A gold standard dataset to probe DNN models is the PDBbind database, developed by Wang et al.[40] and updated on a regular basis.[41] In its last version (v.2020), it stores 19443 protein-ligand X-ray structures of known binding affinity expressed as either inhibition constant ($K_i$), dissociation constant ($K_d$) or half-maximum inhibition concentrations ($IC_{50}$). The general set which encompasses all data is further split in a refined set (5316 entries in the v.2020 release) containing high-quality X-ray structures and the most reliable affinity data ($K_i$ and $K_d$ only), and a core set (290 entries) made of a set of 58 proteins co-crystallized with five different ligands of various affinity. Despite several warnings on the composition[29] and completeness[42] of the PDBbind archive, it remains the largest resource to train machine learning models for structure-based prediction of binding affinities. Many graph neural networks (GNN), used as end-to-end standalone architecture,[14, 19-20, 24, 38] in cascade[37] or in combination with CNNs,[33] have been described recently. None of them significantly outperforms first-generation CNNs, most models presenting rather similar accuracies (Pearson correlation coefficient in the 0.80-0.85 range; root-mean square error around 1.2-1.3 pK unit) in predicting affinities for the PDBbind core set (**Table S1**) but significantly lower accuracies for true external test sets.[31, 33, 35]

Despite the strong commitment of data scientists, we believe that drug discovery has not really benefited from the already described models for the major reasons that machine (deep) learning scoring functions still generalize poorly and are not readily applicable to virtual screening of large compound libraries.[32] This major discrepancy does not prevent computer scientists to propose novel deep learning models, almost on a monthly basis, usually focusing on the novelty of the deep neural network architecture but often omitting to answer three questions: (i) is the apparent performance biased by either the chosen descriptors,[43-44] or the protein-ligand training space? [29, 45] (ii) does the model generalize well to external test sets? (iii) has the model captured the physics of intermolecular interactions and does it achieve good predictions for meaningful reasons?

A first warning has been raised by several groups noticing that CNNs trained on voxelized protein-ligand complexes or graphs do not really learn the physics of protein-ligand recognition because ligand-

only or protein-only models exhibit performances quite similar to those reached by protein-ligand reading models.[14, 46, 44, 29, 35] Comparison of the performance of 24 recently-published DNNs[11-38] reveals that the model accuracy is independent on the size of the training set (e.g. PDBbind general vs. refined set; **Table S1**), contradicting the general idea that more high-quality input protein-ligand structures are required to generate better models. Data augmentation strategies consisting in adding high-quality docking poses to PDBbind X-ray structures also leads to contradictory results.[22, 28, 33, 38] Although very few attempts to predict a true thermodynamic cycle, considering proteins and ligands in their free and bound states have been reported;[12, 29, 35] it remains counter-intuitive that the best models are not obtained with architectures explicitly taking into account the three bound/unbound species. Moreover, there is no relationship between the complexity of protein (sequence vs. structure) and ligand (SMILES strings vs. 2D graphs vs. 3D structures) descriptors and the accuracy of the resulting DNN models.[47-48, 39] Simple models even omitting to consider the protein-ligand bound state are equally good at predicting binding affinities.[47, 31, 48, 42] It is therefore tempting to speculate that DNNs just memorize hidden patterns in either the ligand or protein spaces on which the models have been trained. As a consequence, modifications of protocols used to split input data into training, validation and test sets have a major impact on the accuracy and applicability domain of obtained models.[12, 29]

Since the publicly available training set is limited to the world of PDBbind protein-ligand complexes, there is a need of better identifying still hidden biases in the PDBbind archive, as well as to remove probable redundancies in the choice of descriptors. In the current study, we present a critical evaluation of a modular message passing graph neural network architecture to predict binding affinities from three independent graphs describing proteins, ligands and their complexes. The modularity of the DNN architecture enables to depict the true contribution of each state (free vs. bound) of the two partners and to clearly evidence serious biases in both the ligand and protein composition of the PDBbind space. The current study suggests that descriptors focusing on non-covalent interactions with no ligand/protein additional information are the most suited to unbiased learning.

## RESULTS AND DISCUSSION.

**Describing ligands, proteins and protein-ligand complexes as graphs.** Ligand graphs were generated from PDBbind mol2 input files, defining atoms as nodes and bonds as edges. Each node was annotated by the corresponding atom element, whereas each edge was annotated by the corresponding bond length (**Figure 1A**).

Protein graphs were described from ligand-binding sites, defined as any amino acid, ion or water molecule for which one heavy atom is less than 4 Å away from any ligand heavy atom (**Figure 1B**). In the protein graph, nodes correspond to protein pseudoatoms (PPA), as previously defined by Schmitt et al.,[49] and placed at key main chain/side chain positions and annotated by the molecular interaction properties of the corresponding residue (**Figure S1**). A total of six properties were used to annotate protein nodes with the following labels and interaction properties: CA, Aliphatic (hydrophobic interactions); O, hydrogen-bond acceptor (hydrogen bond); CZ, aromatic (π-π interaction); OG, hydrogen-bond acceptor and donor (hydrogen bond); N, hydrogen-bond donor (hydrogen bond); ZN, metal (metal chelation). To avoid keeping protein residues whose side chains are pointing outwards the ligand-binding cavity, a residue-based filtering was done based on the angle between the ligand center of mass, the residue c-alpha atom and all residue-specific PPAs. PPAs of amino acid side chains, for which the corresponding angle was higher than 90 deg. were removed from the binding site definition. Last, edges were added between final protein nodes distant by less than 4.0 Å and further annotated according to the distance between the corresponding PPAs.

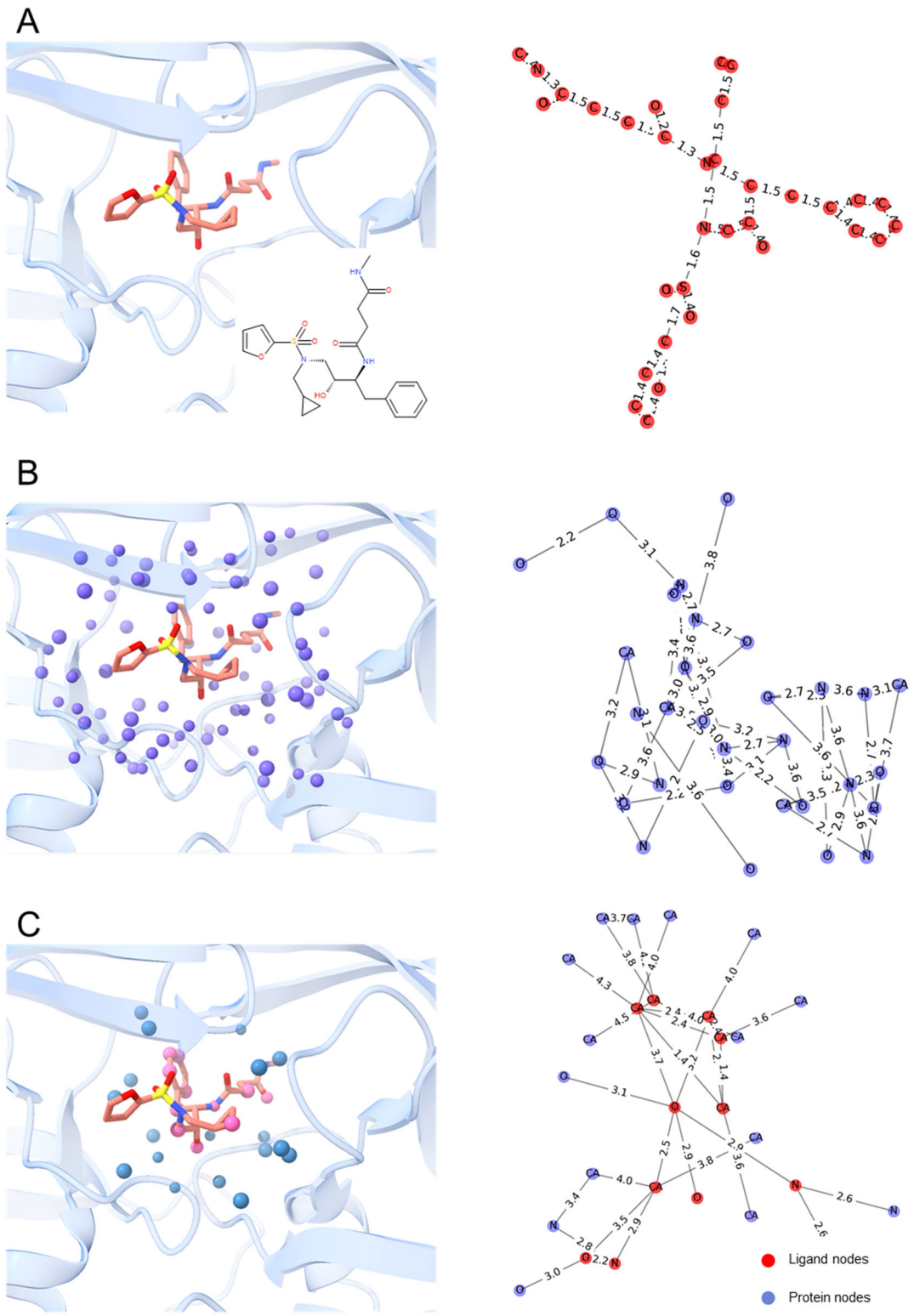


***Figure 1.*** *Encoding protein, ligand and protein-ligand structures (PDB ID 2PSV) in graphs.* ***A)*** *Nodes are set at ligand atomic coordinates (2D sketch in insert), and labelled by atomic element. Edges represent bonds, annotated by bond length.* ***B)*** *Proteins are represented by ligand-binding site pseudoatoms (slate blue spheres) placed at amino acid-specific positions. Nodes are set at protein pseudoatom coordinates and annotated by pharmacophoric properties. Edges link two nodes distant by less than 4.0 Å.* ***C)*** *Protein-ligand interactions are represented by interaction pseudoatoms (pink and blue spheres) set at protein and ligand-interacting atoms. Edges are placed between two nodes (protein, blue; ligand, red) in direct interaction, or between protein or ligand notes if distant by less than 4.0 Å. Each edge is annotated by the distance between the corresponding nodes.*

Non-covalent interactions (hydrophobic, aromatic, hydrogen bonds, ionic bonds, metal chelation; see details in **Table S2**) between protein and ligands were computed on the fly with the GRIM routine of the IChem v5.2.9 package.[50]

For each interaction, interaction pseudoatoms (IPA) are placed at the two atoms of the interacting pair (**Figure 1C**). The resulting representation was converted to a graph where nodes represent either protein or ligand-interacting atoms. Edges between nodes were added in two consecutive steps. First, the principal edges were added between interacting IPAs. Then, secondary edges were added between non-interacting IPAs at the conditions that the corresponding IPAs originate from the same molecule (protein or ligand) and that their distance is less than 4 Å. Each node was annotated by one of the following label, according to the nature of the corresponding non-covalent interaction: CA, hydrophobic; NZ; ionic (Interacting protein atom is positively charged); N, hydrogen-bond (interacting protein atom is donor); OG, hydrogen-bond (interacting protein atom is both acceptor and donor); O, hydrogen-bond (interacting protein atom is acceptor); CZ, aromatic; OD1, ionic (interacting protein atom is negatively charged); ZN: metal coordination. An additional binary label was added to nodes to account for their belonging to either the protein or the ligand. The only edge feature is the distance between pseudoatoms corresponding to the graph nodes (edge length). Therefore, the information on the spatial structure of the binding site was partially preserved, while the representation remained invariant to binding site rotations and node numbering

**Deep neural network architecture**. We used a graph convolutional neural network architecture that belongs to the family of message passing neural networks (MPNN), recently shown to exhibit excellent performance in predicting quantum chemical properties.[51] The MPNN is here applied to an undirected graph $G$ with node features $x_v$and edge features $e_{vw}$. In a MPNN, each node $v$ in the graph has a hidden state $h_v^t$ (feature vector). For each node $v$, a function of hidden states and edges of all neighboring nodes is aggregated. The hidden state of the node Vt is then updated with the obtained

message $m_v^{t+1}$and its previous hidden state. Three main equations characterize the MPNN on graphs. First the message $m_v^{t+1}$ obtained from all neighboring nodes N(v) is given by equation 1:

$$m_v^{t+1} = \sum_{w \in N(v)} M_t(h_v^t, h_w^t, e_{vw}) \qquad (1)$$

where $M_t$ is the aggregation function applied at step t,

$h_v^t$ the hidden state of node $v$,

$h_w^t$ the hidden state of the neighboring node $w$,

$e_{vw}$ is the feature of edge between $v$ and $w$.

The hidden state $h_v^{t+1}$ of the node $v$ is then updated according to equation 2:

$$h_v^{t+1} = U_t(h_v^t, m_v^{t+1}) \quad (2)$$

where $U_t$, the update function is another neural network used to update the hidden state by taking into account both the sum of all previous messages and the previous hidden state.

The message passing algorithm is repeated a user-defined number of times until the readout phase generates a final feature vector $\hat{y}$ describing the entire graph $G$ according to equation 3:

$$\hat{y} = R(\{h_v^T \mid v \in G\}) \quad (3)$$

where $R$ is the readout function

$T$ is the number time steps

The message functions $M_t$ , node update function $U_t$ and readout function R are all learned differentiable functions. The complete architecture of the graph convolutional network (**Figure 2A**)

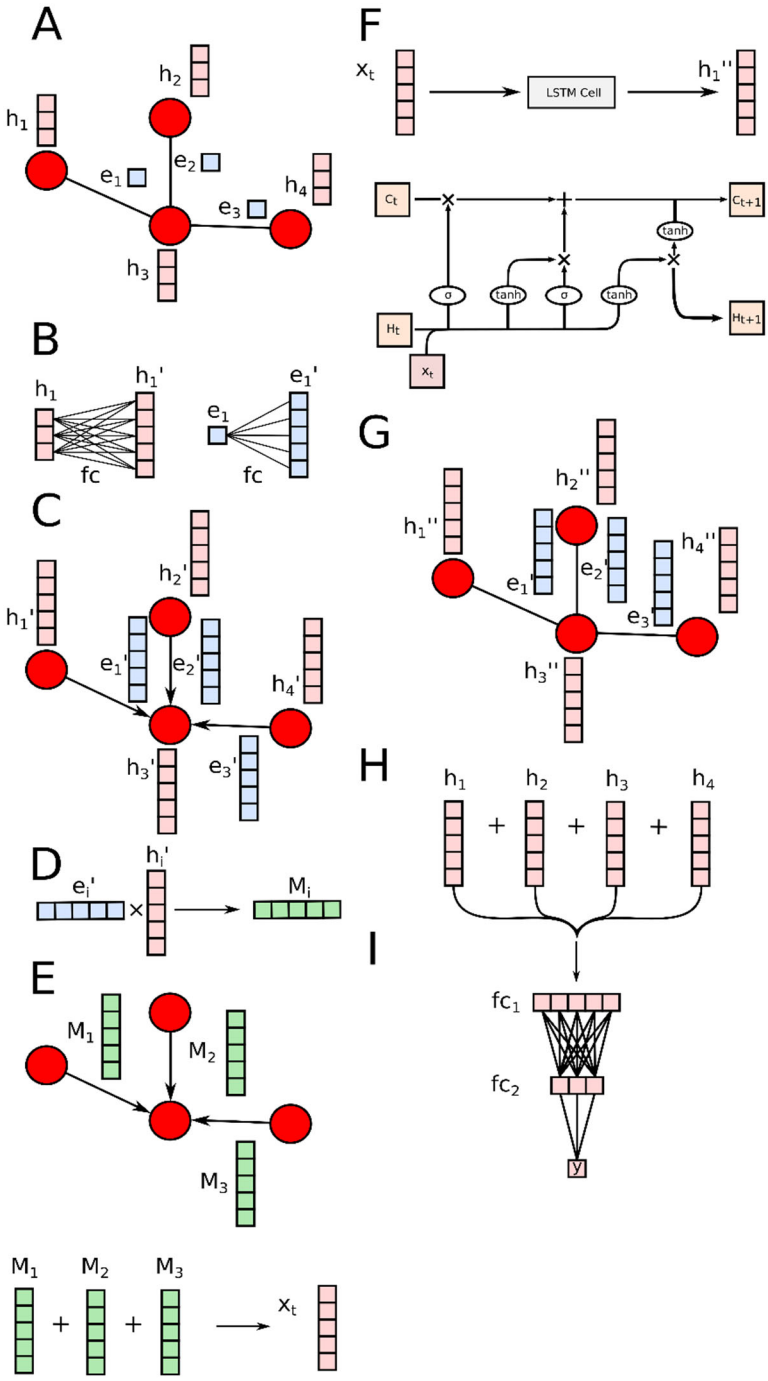


***Figure 2.*** *General architecture of a MPNN with two message passing steps.* ***A)*** *Initial graph with node and edge labels.* ***B)*** *Transformation of node and edge feature vectors with fully connected layers (fc)* ***C)*** *Application of linear layers to node and edge feature vectors.* ***D)*** *Message generation.* ***E)*** *Message passing.* ***F)*** *Node features update using a standard LSTM cell architecture.* ***G)*** *Graph with updated node features.* ***H)*** *Readout.* ***I)*** *Fully connected (fc) layers.*

includes an MPNN module with a customizable hidden size and a two-layer dense module with a top layer size of *hidden size / 4.* The invariance of the MPNN readout function to node and edge re-

enumeration enables applying MPNNs to a merged input consisting of multiple disconnected graphs describing protein, ligand, and protein-ligand interactions without modifying the network architecture. In the current study, MPNN models have been derived from graphs describing the two molecular species (protein, ligand) in both their liganded and unliganded states, thereby enabling to evaluate the exact contribution of each state. To ascertain the fairest possible comparison, all models were trained on the same training/validation set using exactly the same input graphs.

**DNN models are heavily biased by ligand and protein features.** Starting from three possible input graphs describing the protein, the ligand and their non-covalent interactions, seven combinations (one graph, two graphs, three graphs) were first tested as baselines with two objectives: (i) benchmark the performance of MPNN in predicting binding affinities with respect to other DNN architectures,[11-31, 33-38] (ii) analyze the contribution of each input graph and assess their potential synergistic use (**Table 1**).

Despite our customized protocol to process PDBbind entries, we were able to reproduce the performance of the native Pafnucy model,[16] estimated by the Pearson's correlation coefficient R*p* in predicting experimentally-derived affinities for samples of the PDBBind 2016 core set (R*p*= 0.777; **Table 1**). Our seven MPNN models exhibit various performances with Rp values ranging from 0.687 to 0.813. Intuitively, one would have expected that a model trained on protein-ligand interactions (I model) achieves better performance than models trained solely on either the ligands (L model) or the proteins (P model). However, the P and L models exhibit a better performance than the I model (**Table 1**). Out of the one-component models, the ligand-based model is clearly the one leading to the best results (R*p* = 0.749, RMSE=1.567). Combining two graph inputs increases the accuracy of the corresponding predictions, with a clear advantage to the PL model (R*p* = 0.812, RMSE=1.553) omitting protein-ligand interaction features. The most sophisticated model, taking into account the three graph inputs (PLI model), does not provide any clear advantage compared to the PL model, suggesting that explicitly-defined molecular interactions are not required to predict binding affinities of the core set sample.

Applying the models to a much larger (n=3386) and more difficult hold-out set, obtained by temporal splitting of the PDBbind dataset (hold-out 2019 set) illustrates a moderate generalization capacity, with the R*p* value decreasing by ca. 0.15 unit for all models (**Table 1**).

**Table 1.** Performance of modular MPNN models in predicting affinities for the external 2016 core set and the 2019 hold-out set.

| Model[a] | 2016 core set | | 2019 hold-out set | |
|---|---|---|---|---|
| | R*p* | RMSE[b] | R*p* | RMSE |
| P | 0.725 | 1.569 | 0.570 | 1.528 |
| L | 0.749 | 1.567 | 0.611 | 1.455 |
| I | 0.687 | 1.605 | 0.538 | 1.563 |
| PL | 0.812 | 1.553 | 0.645 | 1.512 |
| PI | 0.777 | 1.462 | 0.613 | 1.485 |
| LI | 0.780 | 1.477 | 0.630 | 1.425 |
| PLI | 0.813 | 1.511 | 0.652 | 1.481 |
| Pafnucy[c] | 0.773 | 1.429 | 0.456[d] | 1.642 |

[a] P: protein graph, L: ligand graph, I: interaction graph; PL: merged protein and ligand graphs, PI: merged protein and interaction graph; LI: merged ligand and interaction graph; PLI: merged protein, ligand and interaction graph.

[b] root-mean square error, in pK unit.

[C] in-house Pafnucy prediction (R*p*=0.78 in the original paper)[16]

[d] predictions failed for 29 entries.

From a pure statistical point of view, the performance of four out of the seven MPNN models is superior to that achieved with the CNN Pafnucy model, when applied to the 2016 external core set (**Table 1**). Extending predictions to the challenging 2019 hold-out set suggests that all models

outperform Pafnucy. Assuming that a Pearson R*p* threshold value of 0.600 is commonly used in pharmaceutical industrial settings to qualify a good predictive QSAR model,[52] five out of the seven MPNN models could be considered as satisfactory. However, these models remain enigmatic from a physicochemical point of view since ligand-only and protein-only models still outperform the interaction model. Moreover, the impact on model predictive performance of the explicit consideration of protein-ligand interactions in the two or three-component models remains very limited (**Table 1**). Noteworthy, focusing the analysis on three target classes for which enough samples are present in the hold-out set (GPCRs, 47 samples; kinases, 572 samples; nuclear receptors, 106 samples) did not change the above conclusions (**Figure S2**).

Several conclusions can be drawn from these results. First, the herein implemented MPNN architecture provides a lower accuracy to previously reported CNN and GNN models, when just protein-ligand interactions are taken as input. Pafnucy, used here as a state-of-the-art CNN achieves a better accuracy than the MPNN I model (**Table 1**). Second, protein-ligand binding affinities of the 2016 core set can apparently be predicted from sole protein or ligand structures. Third, the explicit description of protein-ligand interactions does not provide any clear advantage compared to the corresponding interaction-agnostic models (e.g. compare P to PI, L to LI, and PL to PLI models, **Table 1**). Fourth, all models exhibit a decreased accuracy when applied to a hold-out set of newly described complexes, suggesting a probable overtraining. Most of these observations are counter-intuitive and cannot been rationally explained by first-principle physics. They evidence, to our viewpoint, potential biases in the composition of the PDBbind training/test sets suggesting that the derived models have partly memorized input data but not learned the physics of protein-ligand non-covalent interactions. This phenomenon has already been described for many ligand-based machine learning models and frequently happens when training and test sets exhibit significant redundancies.[53] Another alert, that we already mentioned for both machine learning and deep neural networks,[43, 54] is their propensity to predict binding affinities with apparently satisfactory performance metrics (R*p*, RMSE), but where the predicted values are in fact contained within a very tiny range centered on the mean value of training

samples. This tendency is again observed for the current predictions of all MPNN models, whatever the chosen input graph(s) and external test set (**Figure 3**).

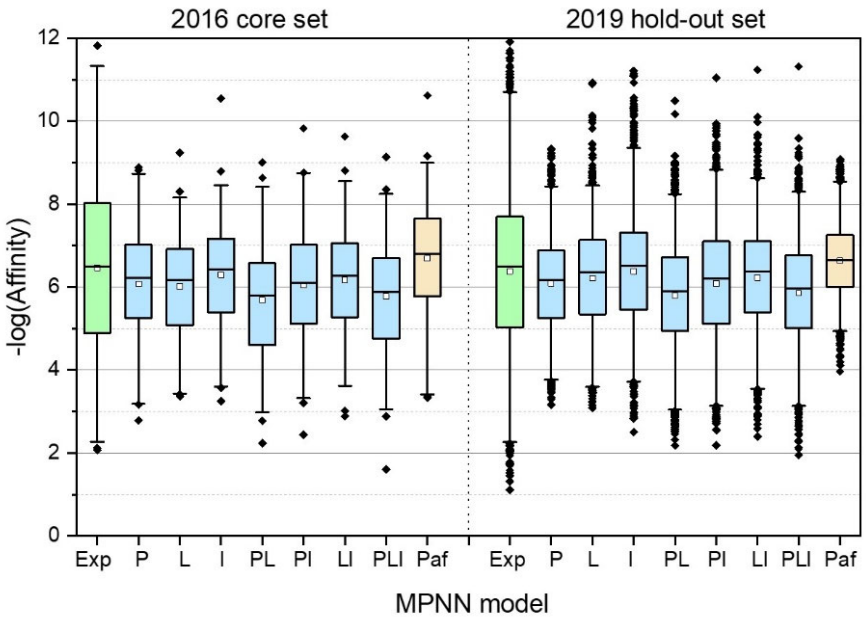


***Figure 3.*** *Distribution of experimental and predicted affinities for the 2016 core set (n=257) and the 2019 hold-out set (n=3386). Exp: experimental affinity; P, L, I, PL, PI, LI, PLI: predicted by MPNN models using protein(P), ligand(L) and protein-ligand interaction (I) graphs used alone or in combinations; Paf: predicted by the Pafnucy model. The boxes delimit the 25$^{th}$ and 75$^{th}$ percentiles, and the whiskers delimit the 1$^{st}$ and 99$^{th}$ percentiles. The median and mean values are indicated by a horizontal line and a filled square in the box, respectively. Outliers are indicated by a diamond.*

Whereas experimental affinities of the two external test sets are spread over 10 pk units, MPNN and Pafnucy predictions are restricted to ca. 6 pk units. Considering only the 25$^{th}$ and 75$^{th}$ percentiles of the distributions (boxes in **Figure 3**), 50% of the predicted data are centered on a mean value ± 1.5 pk unit, Pafnucy predictions lying even in a narrower range for 2019 hold-out set predictions (**Figure 3**). The prediction error is statistically minored if the output value is close to the mean of trained samples. This may be a reason why machine learning models tend to yield narrow distribution of predicted values. This phenomenon might be even amplified in machine learning models for which the loss function aims at minimizing the root-mean-square error. Altogether, we suspect significant biases in the ligand and protein composition of the PDBbind archive which, to our viewpoint, should prevent the blind usage of DNN models in prospective applications.

**Simple memorization models suggest that ligand and protein neighborhoods contribute massively to MPNN predictions.** To estimate the relative contribution of simple memorization vs. true learning when applying MPNNs to predict affinities for PDBbind samples, we generated simple memorization baseline models in which the predicted affinity of a test sample was just inferred by ligand or protein similarity to the five closest training samples (**Table 2**). Of course, such memorization models are meaningless and just define baselines to quantify the amount of biases in the training dataset.

**Table 2.** Performance of simple memorizing models in predicting affinities for the external 2016 core set and the 2019 hold-out set.

| Model | 2016 core set | | 2019 hold-out set | |
|---|---|---|---|---|
| | R*p* | RMSE | R*p* | RMSE |
| PLI MPNN[a] | 0.813 | 1.511 | 0.652 | 1.481 |
| Ligand similarity[b] | 0.663 | 1.624 | 0.509 | 1.641 |
| Protein similarity[c] | 0.547 | 1.765 | 0.310 | 1.794 |

[a] three –component (protein, ligand, protein-ligand interactions) MPNN model of Table 1

[b] prediction is equal to the average affinity of the five training samples with the most similar ligands, similarity being expressed by a Tanimoto coeeficient on ECFP4 circular fingerprints (see Experimental section).

[c] prediction is equal to the average affinity of the five training samples with the most similar proteins, similarity being expressed by an Euclidean distance on protein cavity fingerprints (see Experimental section)

Given its simplicity, the ligand memorization model performs remarkably well on the two external test sets (**Table 2**) and is almost equivalent in accuracy to the protein-ligand interaction MPNN model (I model, **Table 1**). The protein similarity model exhibits a decreased but still noticeable performance. The observed dependency was relatively insensitive to the number of closest training samples (ligands, proteins) used to infer average affinity values for prediction (**Figure S3**). We can therefore conclude that simple memorization probably accounts for a large part of the excellent performance of the MPNN model using ligand, protein and protein-ligand interaction graphs as input (**Table 2**).

**Undersampling the training set does not remove ligand and protein biases.** The goal of this procedure was to reduce the bias originating from the sampling of proteins and ligands present in the PDBBind dataset. So, we undersampled the PDBbind training set by removing progressively the protein-ligand pairs which are easily predictable if we rely solely on protein or ligand graphs, while ignoring the interaction graphs. Intuitively, those are probably the most biased datapoints. As a first approach to remove potential ligand and protein biases in the training set, we filtered out all training samples whose affinities were easily predicted by ligand-only or protein-only five-fold cross-validation MPNN models. The protocol was repeated for batches of 50 samples to get a good tradeoff between speed and precision of the unbiasing algorithm.

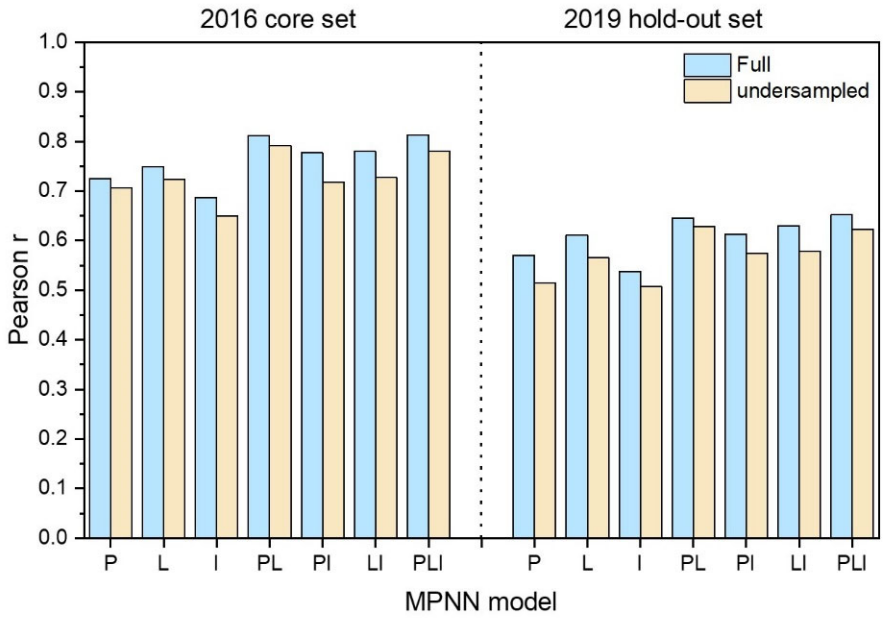


***Figure 4.*** *Effect of undersampling the PDBbind training set on the on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. Default models were trained on the full set (9662 entries) whereas undersampled models were trained only on 4635 samples. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.*

Undersampling reduced the size of the training set from 9662 to 4635 samples, but marginally affected the accuracy of all MPNN models, whatever the graphs used as inputs (**Figure 4**). Interestingly, decreasing the size of the training set by 50% did not alter the quality of the predictions for both

external sets. However, the same obvious biases (good performance of ligand-only and protein-only models, no benefit of explicitly considering protein-ligand interactions) were found again, suggesting that the hidden biases reported above are still present in the undersampled training set.

**Influence of ligand buriedness.** In a second approach, we looked whether the buriedness of the protein-bound ligands in the training and external sets may be a source of potential biases. Indeed, a fully buried ligand would generate quite complementary protein and ligand graphs that implicitly encode all possible non-covalent protein-ligand interactions. In such cases, it might be conceivable to predict albeit with a moderate accuracy the binding affinity of the corresponding complex from sole ligand or protein graphs.

Computing the buried surface area of all PDBbind ligands in their bound state shows a similar distribution for the three sets (training, 2016 core set, 2019 hold-out set) centered on a mean value close to 60-65% (**Figure 5A**).

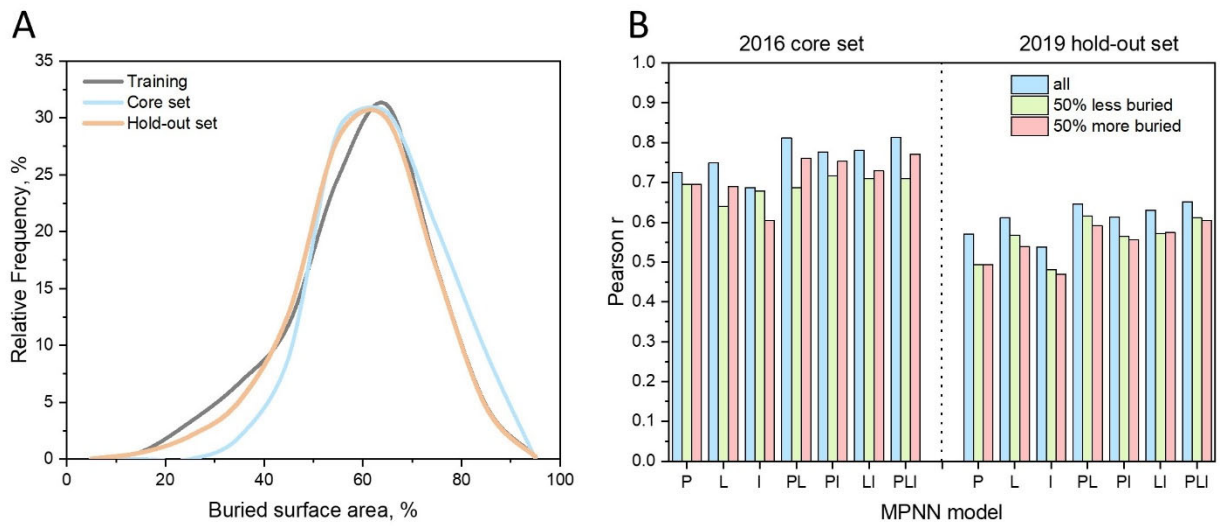


***Figure 5.*** *Effect of ligand buriedness on MPNN predictions.* ***(A)*** *Distribution of the buried surface area of protein-bound PDBbind ligands. (**B**) Influence of the protein-bound ligand buriedness on the scoring power of MPNN models in predicting binding affinities for the core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model ; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.*

We then trained novel MPNN models on two subsets of the PDBbind training set defined by ligand buriedness. The first subset contained the samples with the 50% less buried ligands, whereas the second subset encompassed complexes with the 50% more buried ligands. Using these new MPNN models to predict the binding affinities of samples from the two external sets gave disappointing results (**Figure 5B**). First, all new models were less accurate that the former models trained on the full training set. Second, neither the ligand nor the protein dependency was removed in the new models since novel ligand-only (I models) and protein-only models (P models) were still able to predict binding affinities of both external test samples (**Figure 5B**). We can therefore safely conclude that ligand buriedness is not the cause of protein and ligand biases in the PDBbind dataset.

**Complexity of the protein-ligand interaction descriptors.** As a third approach, we made the hypothesis that the importance of protein and ligand descriptors with respect to the interaction descriptors may originate from the different complexity level of the input graphs. Indeed, interactions graphs computed in IChem are far simpler than the cognate protein and ligand graphs, when considering the number of nodes, edges and the graph density. By default, protein-ligand interactions have been computed using strict geometrical rules (distances, angles),[55] notably interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π- π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å), leading to relative simple graphs with respect to the number of nodes and edges (**Figure 6**). To increase the importance of protein-ligand interactions in our MPNN models, we therefore increased the complexity of interaction graphs by registering non-covalent interactions up to of 6.0 Å. The new interaction graphs ("int6" label) contain much more nodes and edges, are definitely denser and are now comparable with protein and ligand graphs (**Figure 6**).

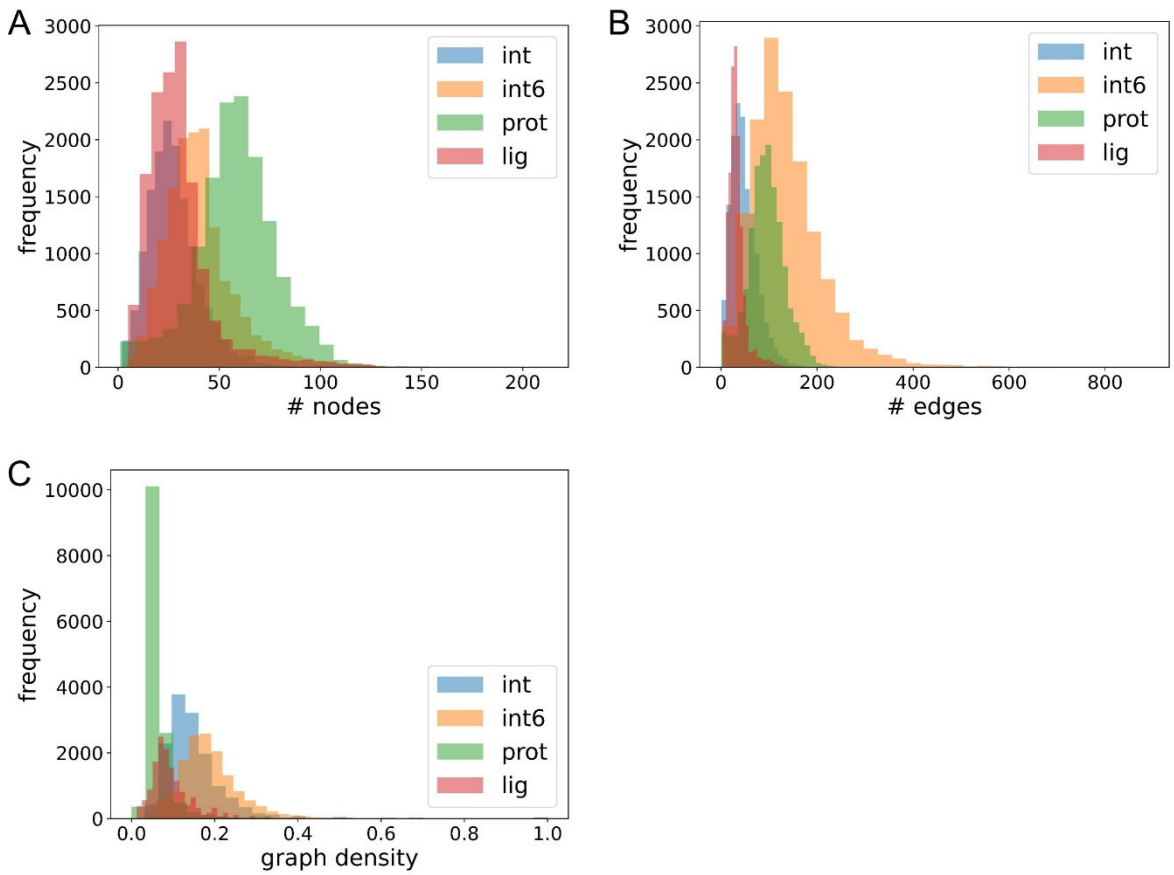


***Figure 6.*** *Distribution of the number of nodes (**A**), number of edges (**B**) and density (**C**) for interaction (int), protein (prot) and ligand graphs derived from PDBbind protein-ligand complexes (n=14 215). The graph density is defined as* $Density = \frac{N_{edges}}{N_{nodes}(N_{nodes}-1)}$ *where* $N_{edges}$ *is the number of edges and* $N_{nodes}$ *is the number of nodes. By default, protein-ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π- π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å). In the extended mode (int6), a larger distance cut-off of 6.0 Å is applied to all non-covalent interactions.*

Using the new interaction graphs as input to MPNN models increased significantly the scoring power of the interaction-only I model for the two external test sets (core set, R*p*= 0.728; hold-out set, R*p*=0.607; **Figure 7)**. Interestingly, this modification did not increase the accuracy of two-component and three-component models (**Figure 7**). Given the marginal benefit of combining the new interaction graph with either protein and/or ligand graphs, using the single new interaction graph definition appears as the best possible compromise between prediction accuracy, model applicability and lower risk of memorization effects.

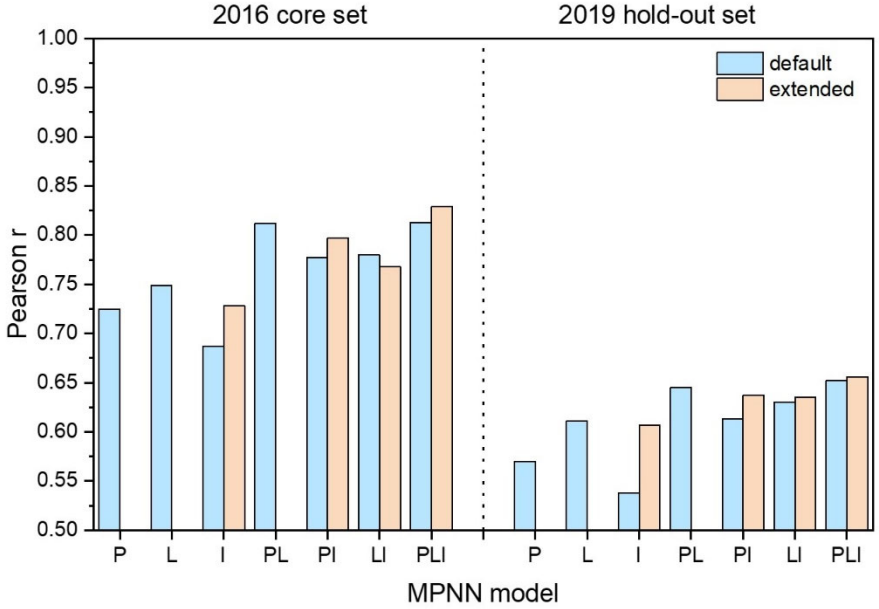


***Figure 7.*** *Influence of the interaction graph complexity on the on the scoring power of MPNN models in predicting binding affinities for the 2016 core set and the 2019 hold-out set. By default, (blue bars), protein-ligand interactions are computed using interaction-specific upper distance thresholds (hydrogen bond: 3.5 Å, aromatic π- π interactions: 4.0 Å, ionic bonds: 4.0 Å, hydrophobic interactions: 4.5 Å). In the extended mode (tan bars), a larger distance cut-off of 6.0 Å is applied to all non-covalent interactions. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.*

**Sparsity of the training protein-ligand matrix.**

Despite a regular increase in the number of entries in PDBbind (**Fig. 8A**), the accuracy of machine learning models in predicting binding affinities has reached a plateau (R*p* = 0.80 ± 0.05), whatever the DNN architecture, the chosen descriptors and the size of the training set (**Table 1**, **Table S1**). Higher accuracies are not necessarily required, given the experimental error associated with heterogeneous binding assays use to collect PDBbind affinities. However, better models are still desirable, notably to achieve accurate and stable predictions when applied to external test sets. Looking at the yearly increase in the number of PDBbind samples, it appears that the number of unique complexes grows faster that the number of unique proteins, the latter increasing faster than the number of unique ligands (**Fig. 8A**).

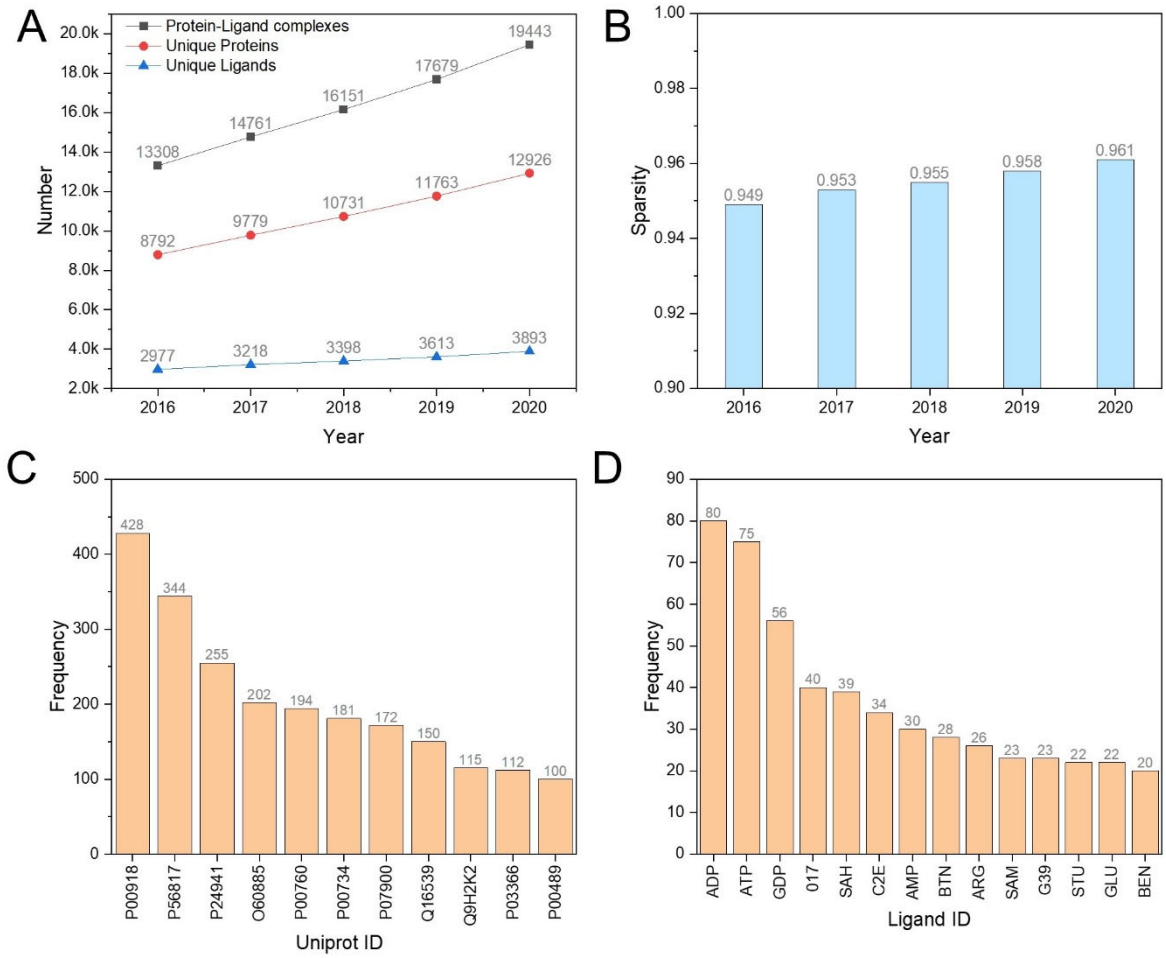


***Figure 8**. Yearly evolution of the PDBbind dataset. **A)** Number of unique entries (protein-ligand complexes, proteins, ligands), **B)** Sparsity of the protein-ligand matrix, **C)** Ten most frequent proteins (PDBbind 2020 release) labelled by their UniProt identifier, **D)** Ten most frequent ligands (2020 release) labelled by their PDB ligand identifier.*

Considering a matrix of $x$ proteins, $y$ ligands and $z$ protein-ligand complexes of known structure, the sparsity $S$ of the PDBbind matrix is defined by the following equation:

$$S = 1 - \frac{z}{x.y} \qquad (5)$$

In other words, the sparsity index describes the fraction of the overall matrix with a missing value (here a protein-ligand complex of known structure and binding affinity). The sparsity $S$ value is very high for the PDBbind dataset (ca. 0.95) and even tends to slightly increases with time (**Figure 8B**). By comparison with high-performance QSAR models, that rely on a minimal number of compound annotations per assay (usually > 200), and now reach the accuracy of four-concentration $IC_{50}$

determinations,[52] the sparsity of the corresponding protein-ligand matrices may reach values as low as 0.65. [56-57, 52]

The PDBbind matrix contains very few targets annotated by multiple ligands (**Figure 8C**). The number of single ligands annotated by multiple proteins is even lower and mostly concerns target-permissive cofactors and nucleotides (e.g. ATP, ADP, AMP, SAM; **Figure 8D**). To check the influence of the training matrix sparsity, we selected the 2030 PDBbind entries from the ten most frequent proteins (**Figure 8D**) to design novel training (n=1505), validation (n=147), and external test sets (core 2016, n=49; hold-out 2019, n=329). Importantly, the set membership (training, evaluation, core, hold-out) of selected entries was kept unchanged, as well as the distribution of experimental affinities (**Figure S4**). The previously described extended interaction model (int6) was here used to describe non-covalent interactions. Altogether, the new subset contains only ten unique proteins and 1777 unique ligands, thereby achieves a lower sparsity (S=0.885) with respect to the full PDBbind 2019 dataset (S=0.958).

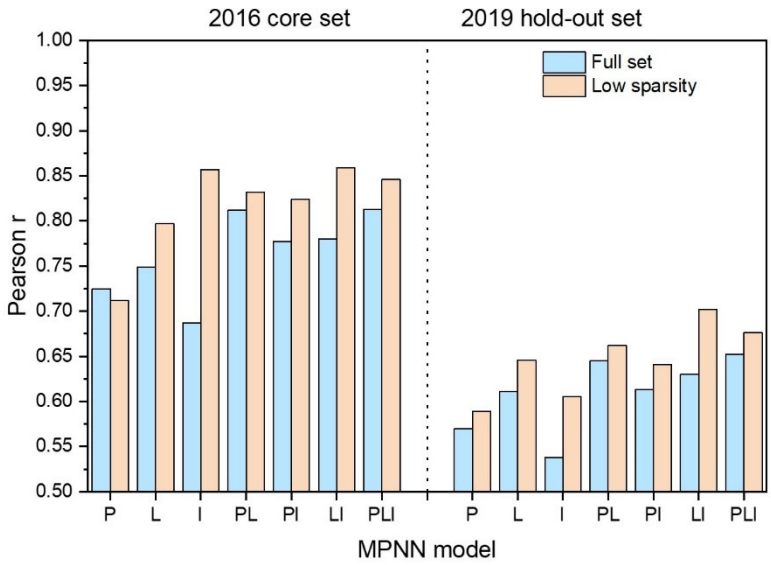


***Figure 9**. Increasing the density of training protein-ligand matrices to predict binding affinities for the 2016 core set and the 2019 hold-out set. P: protein graph model, L: ligand graph model, I: interaction graph model; PL: merged protein and ligand graphs model, PI: merged protein and interaction graphs model; LI: merged ligand and interaction graphs model; PLI: merged protein, ligand and interaction graphs model.*

The performance of the MPNN models on the new subset is higher than that obtained on the full set (**Figure 9**). Unfortunately, neither protein nor ligand dependencies have been removed when predicting affinities for the two external test sets still focusing of the 10 most frequent proteins. The protein-only and ligand-only models remain very accurate, notably for predicting affinities of core set samples. Interestingly, the interaction model is the only one for which the performance is significantly increased for the two external test sets (core set, R*p*=0.852, RMSE=1.256; hold-out set, R*p*=0.605, RMSE=1.363; **Figure 9**). The I model appears again as a reasonable choice for predicting affinities of novel protein-ligand complexes. The current study suggests that increasing the density of the training protein-ligand matrix is an attractive path to increase the accuracy of affinity prediction models. From a practical point of view, it will necessitate a coordinated effort from the drug design community and research financing agencies to solve a wide array of protein-ligand structures in which the same target is repeatedly pictured with different ligands of various affinities, and vice-versa.

**CONCLUSIONS**

Predicting binding affinities of protein-ligand complexes by considering both the corresponding free and bound states appears frustrating because the explicit description of non-covalent intermolecular interactions does not provide any statistical advantage with respect to simpler approximations omitting fine details of protein-ligand interactions. The current study confirms the protein and ligand biases already observed in several studies using DUD-E and PDBbind datasets as sources of three-dimensional information.[12-14, 46, 44, 29, 32-33, 35] However, important controversies still remain regarding the interpretation of these observations. On one side, many computer scientists are not alerted and keep focusing on a pure metrics-based analysis which usually shows that adding descriptors of protein-ligand interactions indeed produce prediction models with slightly better performance metrics (Pearson R correlation, RMSE).[12-14, 32, 35] On the other side, several warnings have been raised by a few groups[46, 44, 29] arguing that a machine learning model must be interpretable from a physicochemical ground. We totally agree with the latter studies, but we were unable to find obvious ways to remove

hidden protein and ligand biases in the PDBbind archive of protein-ligand complexes. Neither undersampling, nor considering ligand buriedness and sparsity of the protein-ligand training matrix could remove the observed tendency of deep neural models to accurately predict binding affinities from sole ligand or protein descriptors. The approach proposed by Yang et al.[29] to split the dataset according to ligand scaffold and protein sequence/structure similarity is efficient in reducing protein and ligand biases but remains artificial and not satisfactory for daily practice where affinity data have to be predicted for new proteins bound to "old ligands" (repurposing), "old proteins" bound to new ligands (hit to lead optimization) and new proteins bound to new ligands (virtual screening). In the current study, we therefore privileged a temporal splitting protocol in which affinities for novel protein-ligand complexes are predicted from a model trained on past structural data. The sparsity of the protein-ligand training matrix appears to be the most important parameter, notably for models trained only on protein-ligand interactions. To avoid building models relying on ligand-specific and protein-specific features, we disfavor annotating the non-covalent interactions with explicit ligand and protein descriptors, as often seen in graph neural networks with attention procedures to annotate graph nodes with ligand and binding pocket connectivity atomic tables.[14, 19, 26, 32] As a conclusion, we recommend training DNN models on pure interaction descriptors in order to reduce the risk of overfitting. Only the latter models appear robust enough to be used for prospective applications.

## EXPERIMENTAL SECTION

**Dataset preparation.** The index files of the PDBbind 2019 release were downloaded from the PDBbind website.[41] For each registered protein-ligand complex, the corresponding atomic coordinates (PDB format) were retrieved from the RCSB Protein Data Bank[58] and processed with Protoss v.4.0[59] to generate atomic coordinates of hydrogen atoms while optimizing the protonation and ionisable states of both ligand and protein amino acids. Each structure was then post-processed using an in-house script to keep only water molecules exhibiting , according to IChem[50] rules, at least two hydrogen bonds to either protein or ligand atoms. Entries with covalently-bound ligands were excluded. Remaining protonated ligand and protein (including all remaining bound water molecules, co-factors, prosthetic groups and ions) were saved separately in mol2 file format. A curated set of 14215 complexes, for which graphs generation succeeded without any failure, was further split in two parts according to the release date (part 1: until 2016-12-31, part 2: after 2017-01-01). Part 1 complexes, corresponding to the general and refined 2016 sets, were divided into training (9662 entries), validation (903 entries) and test (257 entries) as previously described.[16] Part 2 (3386 entries) was saved as an external hold out set, mimicking a real temporal split scenario in which binding affinities for newly released structures are predicted by a model trained on past structural data. Analyzing the distribution of pairwise ligand similarities evidence a large scaffold diversity of each set (training set, 2016 core set, 2019 hold-out set) as well as the absence of obvious similarity biases when comparing the training set to the two external sets. The pairwise similarity and UMAP[60] plots of all PDBbind ligands are provided as supplementary information (**Figure S5**).

**Molecular descriptors.** Proteins, ligands, and protein-ligand interactions were represented as graphs using in-house scripts and the IChem package.[50] The graph processing pipeline was implemented using the Networkx framework v.2.5.[61]

**Message passing neural networks.** The neural network models were implemented using PyTorch v.1.6.0[62] and PyTorch Lightning v.1.5.1.[63] The graph convolution procedure was implemented with a Deep Graph Library framework v.0.5.0.[64]

Two approaches were tested in order to consider the three molecular graphs. In the 'merged approach' the feature vectors of the three input graphs are simply merged. An alternative architecture (parallel approach) was tested, that included separate MPNNs for each input, yielding parallel hidden vectors, which were concatenated before applying fully connected layers to them. Preliminary trials indicated that the parallel architecture had higher memory requirements and demanded longer computational time, while having an accuracy close to that obtained with the merged approach. Thus, the graph merging was selected as the preferable procedure of multiple graph inputs. The parameter optimization aimed to increase the determination coefficient $R^2$ in predicting binding affinities using a stochastic gradient descent (SGD) approach with the ADAM optimizer. The learning rate (lr) was changed over time by the factor of 0.9 after 20 epochs with no improvement for the first lr modification and after 40 epochs for the subsequent lr modifications. The weight decay and dropout rate were set to values of 0.001 and 0.2, respectively. Other hyperparameters (batch size, size of hidden layers, number of message passing steps) were systematically optimized by a grid search as follows:

Batch size: search space [32 , 64, 128, 256 ], final value 256

size of hidden layers: search space [256 , 512, 1024, 2054], final value 2054

message passing steps: search space [1, 2], final value 1

**Data undersampling.** Data undersampling was performed using an iterative five-fold cross-validation approach on the whole PDBbind 2016 training set. At each iteration, ligand-only and protein-only MPNN models were trained using one fold as a test set and the remaining folds as a training set. Binding affinity was predicted for all test complexes with both models. At each iteration, training

samples with the lowest sum of binding affinity prediction errors given by the two protein and ligand models were removed from the dataset. 100 iterations of undersampling were performed and 50 complexes were removed at each iteration. The final undersampled training set contains 4635 protein-ligand complexes.

**Prediction of binding affinities with Pafnucy**.[16] The package was downloaded from the Pafnucy website.[65] In a first step, 3D grids were prepared for each protein-ligand complex in mol2 file format, to create an HDF file with atoms' coordinates and features. In the second step, the recommended model (batch5-2017-06-05T07:58:47-best) was used to rescore each protein-ligand complex, expressing results in $pK_d$ unit.

**Estimation of ligand buriedness.** Ligand buriedness was computed with IChem v5.2.9[50] using bound states of protein and ligand in separate mol2 files.

**Ligand and protein pairwise similarity.** Pairwise ligand similarities were computed from circular ECFP4 fingerprints[66] determined in PipelinePilot v.2019 (Dassault Systèmes Biovia Corp., San Diego, U.S.A). Protein similarities were estimated from the Euclidean distance of 89 cavity descriptors generated by IChem v5.2.9.[50]

**Evaluation metrics.** The scoring power of the different DNN models was evaluated using the Pearson's correlation coefficient (R*p;* equation 4) and the root-mean square error metric (RMSE, equation 5).

$$Rp = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \qquad (4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_i - Y_i)^2}{n}} \quad (5)$$

**SUPPORTING INFORMATION**

Location and pharmacophoric properties of protein pseudoatoms; Performance of modular MPNN models in predicting affinities for specific target classes of the 2019 hold-out set; Influence of the number of closest ligands or proteins used to average binding affinities in the performance of simple memorization models; PDBbind low sparsity subset; Chemical diversity of PDBbind ligands; Structure-based deep neural networks to predict protein-ligand binding affinities; Geometric rules to define protein-ligand non-covalent interactions (PDF).

This material is available free of charge via the Internet at http://pubs.acs.org

**DATA AVAILABILITY**

*Data.* Input files (curated mol2 input files for PDBbind samples; ligand, protein and interaction graphs; training, validation and test set membership) are freely available at http://bioinfo-pharma.u-strasbg.fr/labwebsite/downloads/pdbbind.tgz.

*Software.* Pafnucy version 1.0 was downloaded from https://gitlab.com/cheminfIBB/pafnucy, and used with default settings. Rescoring was performed using the recommended model batch5-2017-06-05T07:58:47-best. IChem (version 5.2.9) was downloaded from http://bioinfo-pharma.u-strasbg.fr/labwebsite/download.html. IChem is freely available for non-profit academic research and subjected to moderate license fees for companies.

**ACKNOWLEDGMENTS**

The Calculation Center of the IN2P3 (CNRS, Villeurbanne, France) is acknowledged for the allocation of computing time and excellent support. We sincerely thank Prof. M. Rarey (University of Hamburg, Germany) for providing an executable version of Protoss. This study was funded by a PhD grant to M.V from Iktos SAS.

**ABBREVIATIONS**

2D, two-dimensional; 3D, three-dimensional; ADP, adenosine diphosphate; AMP, adenosine monophosphate; ATP, adenosine triphosphate; CNN, convolutional neural network; CPU, central processing unit; DNN, deep neural network; ECFP, extended connectivity fingerprint ; IC50, half maximal inhibitory concentration; IPA, interacting pseudoatom; kd, dissociation contant; Ki, inhibition constant; MPNN, message passing neural network; PDB, protein data bank; PPA, protein pseudoatom; Rp, Pearson correlation coefficient; RMSE, root-mean-square error; SAM, S-adenosyl methionine; t-UMAP, Uniform Manifold Approximation and Projection.

**REFERENCES**


1. Mobley, D. L.; Gilson, M. K., Predicting Binding Free Energies: Frontiers and Benchmarks. *Annu. Rev. Biophys.,* **2017,** *46*, 531-558.
2. Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O'Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J., Ultra-Large Library Docking for Discovering New Chemotypes. *Nature,* **2019,** *566*, 224-229.
3. Gorgulla, C.; Boeszoermenyi, A.; Wang, Z. F.; Fischer, P. D.; Coote, P. W.; Padmanabha Das, K. M.; Malets, Y. S.; Radchenko, D. S.; Moroz, Y. S.; Scott, D. A.; Fackeldey, K.; Hoffmann, M.; Iavniuk, I.; Wagner, G.; Arthanari, H., An Open-Source Drug Discovery Platform Enables Ultra-Large Virtual Screens. *Nature,* **2020,** *580*, 663-668.
4. Hoffmann, T.; Gastreich, M., The Next Level in Chemical Space Navigation: Going Far Beyond Enumerable Compound Libraries. *Drug Discov. Today,* **2019,** *24*, 1148-1156.
5. Burley, S. K.; Bhikadiya, C.; Bi, C.; Bittrich, S.; Chen, L.; Crichlow, G. V.; Duarte, J. M.; Dutta, S.; Fayazi, M.; Feng, Z.; Flatt, J. W.; Ganesan, S. J.; Goodsell, D. S.; Ghosh, S.; Kramer Green, R.; Guranovic, V.; Henry, J.; Hudson, B. P.; Lawson, C. L.; Liang, Y.; Lowe, R.; Peisach, E.; Persikova, I.;

Piehl, D. W.; Rose, Y.; Sali, A.; Segura, J.; Sekharan, M.; Shao, C.; Vallat, B.; Voigt, M.; Westbrook, J. D.; Whetstone, S.; Young, J. Y.; Zardecki, C., RCSB Protein Data Bank: Celebrating 50 Years of the PDB with New Tools for Understanding and Visualizing Biological Macromolecules in 3D. *Protein Sci.,* **2022,** 31, 187-208.

6. Kollman, P. A., Free Energy Calculations: Applications to Chemical and Biochemical Phenomena. *Chem. Rev.,* **1993,** *93*, 2395-2417.
7. Guedes, I. A.; Pereira, F. S. S.; Dardenne, L. E., Empirical Scoring Functions for Structure-Based Virtual Screening: Applications, Critical Aspects, and Challenges. *Front. Pharmacol.,* **2018,** *9*, 1089.
8. Ain, Q. U.; Aleksandrova, A.; Roessler, F. D.; Ballester, P. J., Machine-Learning Scoring Functions to Improve Structure-Based Binding Affinity Prediction and Virtual Screening. *WIREs Comput. Mol. Sci.,* **2015,** *5*, 405-424.
9. Kim, J.; Park, S.; Min, D.; Kim, W. Y., Comprehensive Survey of Recent Drug Discovery Using Deep Learning. *Int. J. Mol. Sci.,* **2021,** *22*, 9983.
10. Kimber, T. B.; Chen, Y.; Volkamer, A., Deep Learning in Virtual Screening: Recent Applications and Developments. *Int. J. Mol. Sci.,* **2021,** *22*.
11. Cang, Z.; Wei, G. W., TopologyNet: Topology Based Deep Convolutional and Multi-Task Neural Networks for Biomolecular Property Predictions. *PLoS Comput. Biol.,* **2017,** *13*, e1005690.
12. Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S., Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. **2017**, arXiv:1703.10603 (accessed 2022-03-12).
13. Lau, T.; Dror, R., Brendan-A Deep Convolutional Network for Representing Latent Features of Protein-Ligand Binding Poses. **2017**, http://cs231n.stanford.edu/reports/2017/pdfs/2531.pdf (accessed 2022-03-12).
14. Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S., PotentialNet for Molecular Property Prediction. *ACS Cent. Sci.,* **2018,** *4*, 1520-1530.
15. Jimenez, J.; Skalic, M.; Martinez-Rosell, G.; De Fabritiis, G., KDEEP: Protein-Ligand Absolute Binding Affinity Prediction Via 3D-Convolutional Neural Networks. *J. Chem. Inf. Model.,* **2018,** *58*, 287-296.

16. Stepniewska-Dziubinska, M. M.; Zielenkiewicz, P.; Siedlecki, P., Development and Evaluation of a Deep Learning Model for Protein-Ligand Binding Affinity Prediction. *Bioinformatics,* **2018,** *34*, 3666-3674.

17. Wu, Z.; Ramsundar, B.; Feinberg, E. N.; Gomes, J.; Geniesse, C.; Pappu, A. S.; Leswing, K.; Pande, V., MoleculeNet: A Benchmark for Molecular Machine Learning. *Chem. Sci.,* **2018,** *9*, 513-530.

18. Li, Y.; Rezaei, M.; Li, C.; Li, X.; Wu, D., DeepAtom: A Framework for Protein-Ligand Bidning Affinity Prediction. **2019,** arXiv:1912.00318v1.

19. Lim, J.; Ryu, S.; Park, K.; Choe, Y. J.; Ham, J.; Kim, W. Y., Predicting Drug-Target Interaction Using a Novel Graph Neural Network with 3D Structure-Embedded Graph Representation. *J. Chem. Inf. Model.,* **2019,** *59*, 3981-3988.

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

21. Zhang, H.; Liao, L.; Saravanan, K. M.; Yin, P.; Wei, Y., DeepBindRG: A Deep Learning Based Method for Estimating Effective Protein-Ligand Affinity. *PeerJ,* **2019,** *7*, e7362.

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. Hassan-Harrirou, H.; Zhang, C.; Lemmin, T., RoseNet: Improving Binding Affinity Prediction by Leveraging Molecular Mechanics Energies with an Ensemble of 3D Convolutional Neural Networks. *J. Chem. Inf. Model.,* **2020,** *60*, 2791-2802.

24. Karlov, D. S.; Sosnin, S.; Fedorov, M. V.; Popov, P., GraphDelta: MPNN Scoring Function for the Affinity Prediction of Protein-Ligand Complexes. *ACS Omega,* **2020,** *5*, 5150-5159.

25. Kwon, Y.; Shin, W. H.; Ko, J.; Lee, J., AK-Score: Accurate Protein-Ligand Binding Affinity Prediction Using an Ensemble of 3D-Convolutional Neural Networks. *Int. J. Mol. Sci.,* **2020,** *21*.

26. Li, S.; Wan, F.; Shu, H.; Jiang, T.; Zhao, S.; Zeng, J., MONN: A Multi-Objetcive Neural Network for Predicting Compound-Protein Interactions and Affinities. *Cell Systems,* **2020,** *10*, 308-322.

27. Wang, S.; Liu, D.; Ding, M.; Du, Z.; Zhong, Y.; Song, T.; Zhu, J.; Zhao, R., SE-OnionNet: A Convolution Neural Network for Protein-Ligand Binding Affinity Prediction. *Front. Genet.,* **2020,** *11*, 607824.

28. Xie, L.; Xu, L.; Chang, S.; Xu, X.; Meng, L., Multitask Deep Networks with Grid Featurization Achieve Improved Scoring Performance for Protein-Ligand Binding. *Chem. Biol. Drug Des.,* **2020,** *96*, 973-983.

29. 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*, 69.

30. Zhu, F.; Zhang, X.; Allen, J. E.; Jones, D.; Lightstone, F. C., Binding Affinity Prediction by Pairwise Function Based on Neural Network. *J. Chem. Inf .Model.,* **2020,** *60*, 2766-2772.

31. Ahmed, A.; Mam, B.; Sowdhamini, R., DEELIG: A Deep Learning Approach to Predict Protein-Ligand Binding Affinity. *Bioinform. Biol. Insights,* **2021,** *15*, 11779322211030364.

32. Jiang, D.; Hsieh, C. Y.; Wu, Z.; Kang, Y.; Wang, J.; Wang, E.; Liao, B.; Shen, C.; Xu, L.; Wu, J.; Cao, D.; Hou, T., InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein-Ligand Interaction Predictions. *J. Med. Chem.,* **2021,** *64*, 18209-18232.

33. Jones, D.; Kim, H.; Zhang, X.; Zemla, A.; Stevenson, G.; Bennett, W. F. 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.

34. Kumar, S.; Kim, M. H., SMPLIP-Score: Predicting Ligand Binding Affinity from Simple and Interpretable on-the-Fly Interaction Fingerprint Pattern Descriptors. *J Cheminform.,* **2021,** *13*.

35. Liu, Q.; Wang, P. S.; Zhu, C.; Gaines, B. B.; Zhu, T.; Bi, J.; Song, M., OctSurf: Efficient Hierarchical Voxel-Based Molecular Surface Representation for Protein-Ligand Affinity Prediction. *J. Mol. Graph. Model.,* **2021,** *105*, 107865.

36. Seo, S.; Choi, J.; Park, S.; Ahn, J., Binding Affinity Prediction for Protein-Ligand Complex Using Deep Attention Mechanism Based on Intermolecular Interactions. *BMC Bioinformatics,* **2021,** *22*, 542.

37. Shen, H.; Zhang, Y.; Zheng, C.; Wang, B.; Chen, P., A Cascade Graph Convolutional Network for Predicting Protein-Ligand Binding Affinity. *Int. J. Mol. Sci.,* **2021,** *22*.

38. Son, J.; Kim, D., Development of a Graph Convolutional Neural Network Model for Efficient Prediction of Protein-Ligand Binding Affinities. *PLoS One,* **2021,** *16*, e0249404.

39. Xiong, J.; Xiong, Z.; Chen, K.; Jiang, H.; Zheng, M., Graph Neural Networks for Automated De Novo Drug Design. *Drug Discov. Today,* **2021,** *26*, 1382-1393.

40. Wang, R.; Fang, X.; Lu, Y.; Wang, S., The PDBbind Database: Collection of Binding Affinities for Protein-Ligand Complexes with Known Three-Dimensional Structures. *J. Med. Chem.,* **2004,** *47*, 2977-2980.

41. PDBbind, http://www.pdbbind.org.cn/ (accessed 2022-03-12).

42. Wang, J.; Dokholyan, N. V., Yuel: Improving the Generalizability of Structure-Free Compound-Protein Interaction Prediction. *J. Chem. Inf. Model.,* **2022,** *62*, 463-471.

43. Gabel, J.; Desaphy, J.; Rognan, D., Beware of Machine Learning-Based Scoring Functions-on the Danger of Developing Black Boxes. *J .Chem. Inf. Model.,* **2014,** *54*, 2807-2815.

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

45. Shen, C.; Hu, Y.; Wang, Z.; Zhang, X.; Pang, J.; Wang, G.; Zhong, H.; Xu, L.; Cao, D.; Hou, T., Beware of the Generic Machine Learning-Based Scoring Functions in Structure-Based Virtual Screening. *Brief. Bioinform.,* **2021,** *22*.

46. 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*, e0220113.

47. Ozturk, H.; Ozgur, A.; Ozkirimli, E., DeepDTA: Deep Drug-Target Binding Affinity Prediction. *Bioinformatics,* **2018,** *34*, i821-i829.

48. 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,* **2021,** *37*, 1140-1147.

49. Schmitt, S.; Kuhn, D.; Klebe, G., A New Method to Detect Related Function among Proteins Independent of Sequence and Fold Homology. *J. Mol. Biol.,* **2002,** *323*, 387-406.

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

51. Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; Dahl, G. E., Neural Message Passing for Quantum Chemistry. **2017**, arXiv:1704.01212

52. Martin, E. J.; Polyakov, V. R.; Zhu, X. W.; Tian, L.; Mukherjee, P.; Liu, X., All-Assay-Max2 PQSAR: Activity Predictions as Accurate as Four-Concentration IC50s for 8558 Novartis Assays. *J. Chem. Inf. Model.,* **2019,** *59*, 4450-4459.

53. Wallach, I.; Heifets, A., Most Ligand-Based Classification Benchmarks Reward Memorization Rather Than Generalization. *J. Chem. Inf. Model.,* **2018,** *58*, 916-932.

54. Tran-Nguyen, V. K.; Bret, G.; Rognan, D., True Accuracy of Fast Scoring Functions to Predict High-Throughput Screening Data from Docking Poses: The Simpler the Better. *J. Chem. Inf. Model.,* **2021,** *61*, 2788-2797.

55. Marcou, G.; Rognan, D., Optimizing Fragment and Scaffold Docking by Use of Molecular Interaction Fingerprints. *J. Chem. Inf. Model.,* **2007,** *47*, 195-207.

56. 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.

57. Metz, J. T.; Johnson, E. F.; Soni, N. B.; Merta, P. J.; Kifle, L.; Hajduk, P. J., Navigating the Kinome. *Nat. Chem. Biol.,* **2011,** *7*, 200-202.

58. Protein Data Bank, https://www.rcsb.org (accessed 2022-12-03).

59. Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M., Protoss: A Holistic Approach to Predict Tautomers and Protonation States in Protein-Ligand Complexes. *J. Cheminform.,* **2014,** *6*, 12.

60. McInnes, L.; Healy, J.; Melville, J., Umap: Uniform Manifold Approximation and Projection for Dimension Reduction., **2021**, arXiv:1802.03426v3.

61. Networkx- Network Analysis in Python, https://networkx.org (accessed 2022-03-12).

62. Pytorch, https://pytorch.org/ (accessed 2022-03-12).

63. Lightning, https://www.pytorchlightning.ai/ (accessed 2022-03-12).

64. Deep Graph Library, https://www.dgl.ai/ (accessed 2022-03-12).

65. Pafnucy Website, https://gitlab.com/cheminfibb/pafnucy (accessed 2022-12-03).

66. Rogers, D.; Hahn, M., Extended-Connectivity Fingerprints. *J. Chem. Inf. Model.,* **2010,** *50*, 742-754.

**Table of contents graphic**