Article

Machine Learning Scoring Functions for Drug Discovery from Experimental and Computer-Generated Protein–Ligand Structures: Towards Per-Target Scoring Functions

Francesco Pellicani [1], Diego Dal Ben [2], Andrea Perali [3] and Sebastiano Pilati [1,4,*]

1   Physics Division, School of Science and Technology, University of Camerino, I-62032 Camerino, MC, Italy
2   Medicinal Chemistry Unit, School of Pharmacy, University of Camerino, I-62032 Camerino, MC, Italy
3   Physics Unit, School of Pharmacy, University of Camerino, I-62032 Camerino, MC, Italy
4   INFN-Sezione di Perugia, I-06123 Perugia, PG, Italy
*   Correspondence: sebastiano.pilati@unicam.it

Abstract: In recent years, machine learning has been proposed as a promising strategy to build accurate scoring functions for computational docking finalized to numerically empowered drug discovery. However, the latest studies have suggested that over-optimistic results had been reported due to the correlations present in the experimental databases used for training and testing. Here, we investigate the performance of an artificial neural network in binding affinity predictions, comparing results obtained using both experimental protein–ligand structures as well as larger sets of computer-generated structures created using commercial software. Interestingly, similar performances are obtained on both databases. We find a noticeable performance suppression when moving from random horizontal tests to vertical tests performed on target proteins not included in the training data. The possibility to train the network on relatively easily created computer-generated databases leads us to explore per-target scoring functions, trained and tested ad-hoc on complexes including only one target protein. Encouraging results are obtained, depending on the type of protein being addressed.

Keywords: molecular docking; scoring functions; machine learning

1. Introduction

Scoring functions (SFs) are popular models often adopted by medicinal chemists to perform computational docking and to score poses of candidate ligands in the pockets of target proteins [1,2]. They are expected to attribute the best score to the correct pose, to accurately predict the binding affinity (or a score proportional to the affinity), and to prioritize active ligands over inactive ones. The latter task, in particular, plays a pivotal role in virtual screening [3]. It allows for identifying the most promising drug molecules, thus reducing the time and the cost otherwise spent for in vitro screening [4]. Due to their importance in drug discovery [5], the development of SFs has been the focus of intense research endeavors for decades. Modern SFs are often grouped [6] into empirical [7], knowledge based [8], and force fields [9] SFs. More recently, SFs have been implemented also using data-based approaches via machine-learning (ML) techniques [10–12]. This strategy is favored by the increasing amount of crystallographic protein–ligand structures [13], and it aims at exploiting the effectiveness of ML techniques in extracting useful information from big databases. An influential article was published in 2010 [14]. Therein, the authors adopted a relatively simple ML model, namely, random forest regression [15], and they trained it to predict binding affinities using about 1200 protein–ligand complexes extracted from the PDBBind databases [16,17]. They reported a promising Pearson correlation coefficient $R_p = 0.776$, which indicates a strong correlation between predicted and experimental binding affinities [18]. Notably, the ML-based SF outperformed all considered classic ones. Later, a similar ML-based SF was specialized to virtual screening tasks, thus also addressing

the comments of Ref. [19] on the poor performance of ML-based SFs in this task. Various subsequent studies have adopted more advanced ML models, such as dense neural networks (NNs) [20], convolutional NNs [21–24], and graph NNs [25], often further increasing the performance in binding affinity prediction. For a recent review on neural networks and deep learning techniques, see, e.g., Ref. [26]. Different complex representations have been explored, ranging from distance-counts of protein–ligand atomic pairs (combined with kernel ridge regression) [14,27] to three-dimensional (3D) grids of atomic features (combined with convolutional NNs) [21–24]. However, very recent studies have put these optimistic results into question [28]. They reported a drastic performance reduction when ML SFs are subjected to more stringent benchmarks. Noticeable examples are so-called vertical tests, whereby predictions are made for proteins not included in the training set, as opposed to the more common and less stringent horizontal tests, whereby a protein might be present both in the training and in the test sets, albeit bound to a different ligand. Along this line, Refs. [22,27] considered scaffolding tests designed to quantify how the prediction accuracy varies with the degree of structural similarity between the proteins in the training and in the test sets. Notably, Ref. [28] reported that the performance of an exemplary ML-based SF did not change when it was fed with only the protein structure, or only the ligand structure (as opposed to the whole complex). The authors thus suggested that the ML-based SF had not learned any information about the actual protein–ligand binding mechanism.

A critical problem for the development of ML techniques for computational docking is the limited number of experimental structures available for model training [18,29]. From the most relevant database, namely, the PDBBind repository, one may extract a few thousand complex structures, depending on the required resolution [29]. While this number is steadily increasing, it is still orders of magnitude smaller than the amount of data typically used in other fields where ML has proven astonishingly successful, such as, e.g., computer vision. For example, state-of-the-art NNs for object detection are usually trained on databases including millions of images, e.g., the ImageNet database [30]. This issue is particularly relevant for deep NNs. These are often preferred to simpler ML regression models, such as kernel ridge regression or support vector machines, due to their superior generalization properties. However, in general, they require larger training databases to avoid overfitting problems. Beyond the small size, training databases might be affected by anthropogenic factors, leading to a biased selection towards particularly favourable instances, as recently found in studies on chemical reactions [31]. The contrasting findings discussed above call for further quantitative performance analyses on ML-based SFs, considering in particular the role of the training set and of the test type. One of the main questions we aim to address in this article is whether ML-based SFs can be trained using computer-generated complex structures created using docking software, providing access to larger and more tunable databases.

In this Article, we implement an SF using an NN with fully connected layers (FCNN). Following Refs. [14,27], the descriptors used to represent the complex structures are counts of atomic pairs, one belonging to the protein and the other to the ligand, within various distance intervals. Two databases are used for training and testing. The first includes 2408 crystallographic structures extracted from the PDBBind database. Notably, we carefully check and prepare these structures with the inclusion of hydrogen atoms, in contrast to previous related studies that only considered heavy atoms. The second database includes 28,200 computer-generated structures created using the CCDC Gold docking engine within the MOE (Molecular Operating Environment) software interface [32–34]. In both databases, the complex structures are associated with the corresponding experimental binding affinities. The FCNN is trained to predict binding affinities of previously unseen complexes via a supervised learning algorithm. One of our goals is to quantify the performance of a FCNN combined with a distance-count description of protein–ligand complexes. Chiefly, we compare the performances reached using experimental and computer-generated databases, considering in both cases both horizontal as well as more challenging vertical tests. Fi-

nally, taking advantage of the creation of computer-generated databases, we explore the development of per-target SFs designed to predict binding affinities for a specific target protein. We consider 17 exemplary targets, training a specific SF for each of them using computer-generated databases including many complexes with different ligands docked into the same target protein. The performances obtained on the exemplary targets are analysed, also varying the size of the training set. In general, the obtained performances are encouraging, with noticeable variations depending on the target. To shed some light on these different performances, we make a comparison against basic linear regression models based on the molecular weight, optimized for each target.

2. Materials and Methods

2.1. Experimental Database

The first database includes 2408 complex structures obtained through X-ray crystallography [35] and deposited into the Protein Data Bank [36,37] . With this experimental technique, the positions of the heavy atoms, i.e., excluding hydrogen atoms, are estimated with a finite resolution. We select structures with a resolution degree lower than 3 Å and whose ligand–target interaction information is available from the PDBbind database [16,17]. The 3D structures are manually checked for possible inconsistencies and curated with the addition of the hydrogen atoms, using the MOE software. This process is time consuming, thus limiting the size of the experimental database. However, it allows us to adopt a more complete representation, as opposed to various previous related studies that considered only heavy atoms. Furthermore, the inclusion of hydrogen atoms allows for performing a more direct comparison against the computer-generated complexes described below. The target–ligand structures are associated with the corresponding dissociation constant K_d. In fact, we consider the values of $pK_d \equiv -\log_{10}(K_d)$. These values are retrieved from the PDBbind database. For completeness, we report here the formal definition of dissociation constant:

$$pK_d = -\log_{10}(K_d) = -\log_{10}\left(\frac{[P][L]}{[C]}\right),\qquad(1)$$

where $[P]$, $[L]$, and $[C]$ represent the concentrations of the protein, of the ligand, and of the complex, respectively. Some relevant details on our databases are summarized in Table 1.

Table 1. Brief description of the experimental and the computer-generated databases. (*) The sign of the docking score is changed from negative to positive for consistency with the affinity values. These databases are freely accessible using the link in Ref. [38].

Database	Number of Complexes	Mean Affinity	Mean Docking Score (*)
Experimental	2408	5.98 (pK_d)	
Computer generated	28,200	7.48 (pK_i)	11.43

2.2. Computer-Generated Database

The second database includes structures generated via computer simulations. We refer to it as the computer-generated database. To build it, the 3D structures of 17 selected target proteins are retrieved from the PDB repository. The selection focuses on diverse targets with many ligands deposited in the BindingDB database [39–41]. Only complexes with a single ligand at the binding site and no cofactors are considered. For each target, a numerous set of ligands is retrieved. The choice of ligands is restricted among those whose experimental binding affinity for each selected target is available as expressed by the pK_i score. This score measures the target–ligand affinity using a reference radio-ligand. Its value is generally close to pK_d. These ligands are docked into the respective target using the GOLD docking engine through the MOE software. Our simulations produce ten poses for each target–ligand pair using the GOLD docking engine. These poses are

then rescored within MOE by the GBVI/WSA dG scoring function, following the protocols of previous studies; see., e.g., Refs. [42,43]. The target–ligand pose assigned of the best docking score is selected. The final computer-generated database includes only the poses corresponding to the best score for each ligand at the respective target, totalling 28,200 protein–ligand complexes. It is worth pointing out that, due to the possible inaccuracy of the classical SF, the best pose does not necessarily match the correct experimental one. Still, avoiding mediocre scores likely increases the chances of excluding odd poses featuring clear inconsistencies, which might prevent the NN from learning to predict the correct affinity. The number of pairs per target ranges from 384 for the PIM2 protein, to 6568 for the D2 protein. The selected target proteins and the corresponding number of complexes are summarized in Table 2. Evidently, the computer-generated database is significantly larger than the experimental one. This allows us to better analyse the learning speed of NNs. However, it is worth emphasizing that the docking pose created by the docking software is affected by the possible inaccuracy of the chosen docking engine. Instead, the 3D crystallographic structures are expected to correspond to the actual spatial configuration. It is worth pointing out that, in fact, spurious distortions might be present also in the crystallographic structures, but this is believed to rarely happen. Clearly, while the binding affinity information present in the computer-generated database is, in fact, experimental, the 3D complex structure is affected by the choice of the SF used by the docking engine and, henceforth, by its possible inaccuracies. It should be noted, however, that this does not necessarily represent a drawback. Indeed, in virtual screening campaigns, SFs are often used to select promising ligands from software-generated poses. Therefore, training the SF on the type of structures it will be asked to rank might actually be instrumental.

Table 2. Breakdown of the protein–ligand complexes included in the computer-generated database. The following legend defines the protein acronyms: 5HT2A: human 5-HT$_{2A}$ receptor (pdbcode: 6A94); A2A: human A$_{2A}$ Adenosine receptor (pdbcode: 5NM4); ACE: Human acetylcholinesterase (pdbcode: 4EY5); BACE1: human BACE-1 enzyme (pdbcode: 6UVP); D2: human D2 dopamine receptor (pdbcode: 6CM4); DOP: human delta opioid receptor (pdbcode: 4N6H); FAAH: humanized variant of fatty acid amide hydrolase (pdbcode: 3PPM); GR: human glucocorticoid receptor (pdbcode: 4UDD); H1: human histamine H$_1$ receptor (pdbcode: 3RZE); JAK1: human Janus kinase 1 (pdbcode: 6N7A); JAK2: human Janus kinase 2 (pdbcode: 6VN8); KOP: human kappa opioid receptor (pdbcode: 4DJH); M1: human M1 muscarinic acetylcholine receptor (pdbcode: 5CXV); MCL1: human Mcl-1 (pdbcode: 6UDV); OX2: human orexin 2 receptor (pdbcode: 5QWC); PI3K: human Phosphatidylinositol 4,5-bisphosphate 3-kinase catalytic subunit alpha isoform (pdbcode: 6PYS); PIM2: human PIM2 kinase (pdbcode: 4X7Q).

Protein	5HT2A	A2A	BACE1	DOP	FAAH	GR	H1	JAK1	PI3K
N. of complexes	2763	2914	1413	1243	508	843	1070	1213	1064

Protein	PIM2	ACE	KOP	M1	MCL1	JAK2	OX2	D2	
N. of complexes	384	488	2431	1056	688	1394	2160	6568	

2.3. Complex Representation

The input provided to the FCNN must be designed to represent, with good approximation, the 3D structure of the protein–ligand complexes. The description we adopt follows the archetypes proposed in previous studies [14,27]. Specifically, each descriptor corresponds to the count of atom pairs within a specified distance interval, whereby the first atom belongs to the target, the second to the ligand. We consider the following 10 atomic species: H, C, N, O, F, P, S, Cl, Br, and I. This choice follows the study of Ref. [14] (apart for hydrogen). See also the following studies of Refs. [23,24,27]. For the ligands, only the species H, C, N, O, P, and S are considered here. This restriction is due to our choice of addressing only ligands not including halogens and metallic atoms. While this is a possible limitation of our analysis, the selected atomic species are sufficient to describe many drug

molecules and, therefore, we expect that our findings concerning training and testing protocols and per-target SFs are sufficiently general. The above choices lead to 60 descriptors per distance interval. Notice that halogens could be avoided also for proteins. However, NNs quickly learn to ignore constant descriptors. Notably, our representation takes into account H atoms. Furthermore, originally, Ref. [14] considered only one distance interval, namely, 0–12 Å. Subsequently, Ref. [27] adopted a more detailed representation, dividing the 12 Å range into six intervals. In this article, we explore different representations, varying both the number of intervals and their width. The goal is to identify the optimal compromise between representation accuracy and conciseness. We point out that the original training matrix includes descriptors which might differ by orders of magnitudes. For this reason, a normalization operation is helpful. Our analysis on the experimental database shows that the most effective operation is dividing by the maximum value of all descriptors. To formally define this normalization procedure and, more in general, the descriptor vector, it is useful to introduce the following notation: the (unnormalized) number of pairs of atomic species $A \in (H, C, N, O, F, P, S, Cl, Br, I)$ (for the target) and $A' \in (H, C, N, O, P, S)$ (for the ligand) that lie in the distance interval labeled by the index $k = 1, 2, \ldots, k_{\max}$, where $k_{\max}$ is the chosen number of intervals, is denoted as $N_k^{AA'}$. Therefore, we have 60 pairs of atomic species, meaning that each complex is characterized by $60k_{\max}$ descriptors. The interval widths we consider are $\ell = 1.5$ Å, $\ell = 2$ Å, or $\ell = 3$ Å. The intervals are defined by the following minimum and maximum distances: $r_{\min} = (k - 1)\ell$ and $r_{\max} = k\ell$. Different numbers of intervals $k_{\max}$ are addressed. The smallest value is $k_{\max} = 1$ for all three interval widths. Instead, the largest number differs depending on the interval width, namely, $k_{\max} = 6$ for $\ell = 1.5$ Å, $k_{\max} = 7$ for $\ell = 2$ Å, and $k_{\max} = 5$ for $\ell = 3$ Å. The normalized descriptors are $N_k^{AA'}/\max\{N_k^{AA'}\}$, where the maximum value is taken over all descriptors of all instances in the corresponding database. This normalization is adopted for all results reported below.

2.4. Target Values

SFs are designed to predict a score proportional to the binding affinity. As already mentioned, we train and test our FCNN SFusing, as regression target, the pK_d and the pK_i values, for the experimental and the computer-generated databases, respectively. For the latter database, in only one analysis aiming at verifying if and to what extent a ML-based SF can mimic a classic SF, we also consider the docking score. For convenience, we adopt as a target value the negative of the docking score, so that higher values correspond to putatively higher affinities. The mean target values of our databases are summarized in Table 1.

Notice that, during training, we adopt the standardized targets $d' = \frac{d-\mu}{\sigma}$, where d is the original value of binding affinity (pK_d or pK_i, depending on the considered database) or negative docking score (only when the FCNN SFis trained to reproduce the prediction of the classic SF), μ the database mean, and σ the corresponding standard deviation. This (linear) standardization does not affect the correlation coefficient R_p, and it is introduced only to allow us to better compare the phenomenology of the training processes on different databases. Furthermore, to favor comparison with other studies, the mean squared error (MSE) values reported below are computed considering un-normalized predictions and targets d, obtained by inverting the standardization formula.

2.5. Regression Model and Training Protocol

The goal of supervised learning is to train a regression model to map the complex descriptors to the target value, namely, the binding affinity (or the negative docking score). The regression model adopted in this article is an FCNN, namely, a dense NN with all-to-all interlayer connectivity. The numbers of (hidden) layers N_l and of hidden neurons per layer N_h are chosen through the analysis described in Section 3. Note that N_l does not count the descriptor layer, nor the output layer featuring a single neuron. The activation function in the hidden layers is the hyperbolic tangent. The network weights and biases

are optimized by minimizing the loss function, namely, the MSE between the network's predictions and the ground-truth target values. To contrast possible overfitting phenomena, the loss function is augmented with a standard L_2 regularization term [44]. However, this does not lead to significant benefits, and the results we report hereafter correspond to a negligibly small regularization parameter. Our neural networks are implemented and trained using a very popular framework for deep learning, namely, the Keras Python library [45]. Furthermore, we provide the code with the trained model corresponding to one of our most relevant SFs (see Section 3.3) at the repository of Ref. [38]. The optimization is performed using a variant of stochastic gradient descent, named ADAM [**?**]. An adequate mini-batch size turns out to be around 50 and 200, for the experimental and the computer-generated databases, respectively.

The training epochs are iterated until the prediction accuracy on the test set stops improving, i.e., before entering the regime where overfitting phenomena dominate. Due to the small size of the experimental database, introducing a validation set is not practical. For consistency, the same protocol is adopted also for the computer-generated database. This is intended to estimate the optimal potential performance. While, in principle, it might lead to a slight overestimation, this effect is found not to be important. In particular, in the most critical per-target FCNN SFtests, the prediction accuracy is found to be quite stable as a function of the training epochs. To monitor the prediction accuracy, two metrics are considered, namely, the Pearson correlation coefficient R_p and the MSE. These are computed on a test set which includes around 20% of the whole database, while the remaining instances are used for training. The training and testing process is repeated ten times, considering just as many different random splittings between training and test instances, or simply different (random) selections of training data and mini-batches for gradient descent. The accuracy scores reported hereafter correspond to the average of the test scores, while error bars correspond to the estimated standard deviation of the average. This averaging procedure avoids spurious fluctuations due to accidentally favourable or adverse selections of the test instances, providing a more reliable estimate of the prediction accuracy in a realistic scenario.

3. Results

3.1. Selection of Descriptors and Network Structure

To identify the optimal choice for the complex–structure representation, we analyse the prediction accuracy on test sets of 300 randomly chosen experimental complexes, as a function of the number of distance intervals of atom-pair counts. The choice of considering 300 complexes (randomly selected from the whole experimental database) for testing is a trade off between the need of the largest possible training set and the minimal number required for a reliable test. As discussed in Section 2.5, 10 random non-overlapping splittings are considered to reduce the role of statistical fluctuations due to particularly favourable or adverse testing complexes. Notice that, while the test protein–ligand complexes are distinct from those used for the optimization of weights and biases, some target proteins might occur in both training and test sets, albeit bound to a different ligand. This is what we refer to as a horizontal test. The results are shown in Figure 1, considering the three interval widths, namely, $\ell = 1.5$ Å, $\ell = 2$ Å, and $\ell = 3$ Å. For a complete definition of the descriptor vector, we refer the reader to Section 2.3. One notices that the maximum R_p score, which corresponds to the optimal representation, is obtained using $k_{\max} = 4$ intervals of width $\ell = 2$ Å, meaning that atom pairs are counted only if the two atoms are less than 8 Å apart. Considering that 60 pairs of atom species are considered, the total number of descriptors is 240. This representation is adopted for all results reported hereafter.
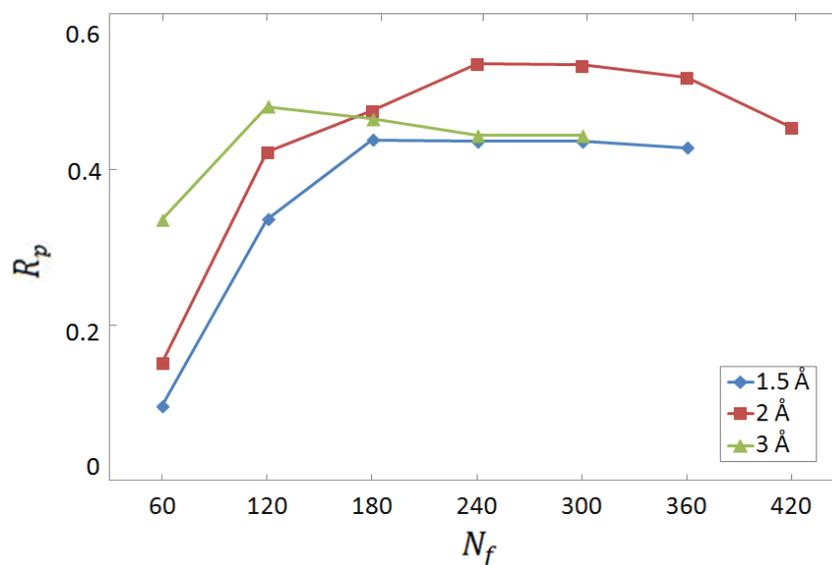
Figure 1. Pearson correlation coefficient R_p for the predictions of binding affinity provided by the FCNN SF, as a function of the number of descriptors N_f. This FCNN SFis trained and tested on the experimental database. The protein–ligand complex structure is represented by atomic-pair counts within a variable number of distance intervals. The three datasets correspond to different interval widths. Sixty pairs of atomic species are considered, including pairs with hydrogen.

We identify the optimal depth and width of the neural network by analysing the R_p score on the experimental test set. The results are shown in Figure 2 as a function of the number of training instances N_t. Different numbers of layers N_l and of neurons per (hidden) layer N_h are considered. The highest performance is obtained for $N_l \times N_h = 2 \times 20$. An analogous analysis performed on the computer-generated data (not shown) indicates that, in that case, the optimal network structure is $N_l \times N_h = 4 \times 40$. The need of a deeper network can be attributed to the larger size of the computer-generated database. Indeed, it is known that deeper NNs are more effective in extracting useful information from larger databases, while they are more susceptible to overfitting phenomena when the database is sparse. These two NN structures are adopted for all results reported below, unless otherwise specified.
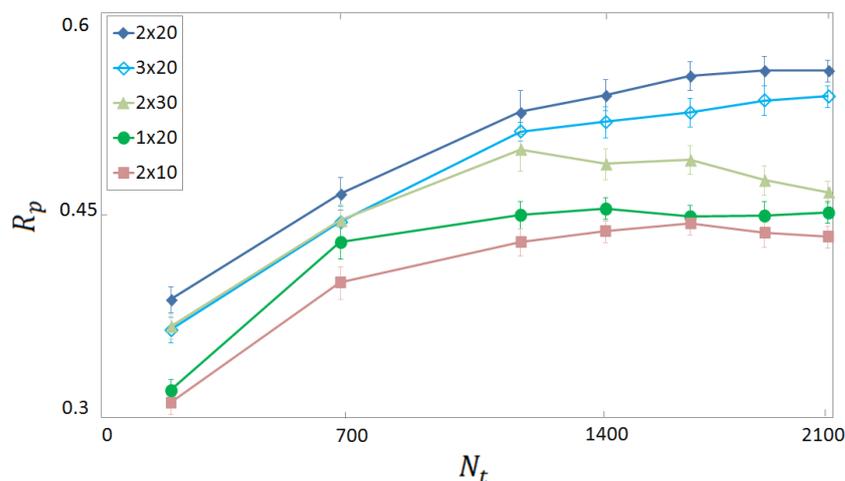

Figure 2. R_p score for binding affinity prediction as a function of the size N_t of the (experimental) training set. Each dataset corresponds to a choice $N_l \times N_h$ for the number of hidden layers N_l and of neurons per layer N_h in the fully connected neural network (FCNN). The complex representation includes $N_d = 60$ descriptors, with four distance intervals of width 2A.

3.2. Horizontal Tests on Experimental and Computer-Generated Structures

One of our main goals is to compare the performances of ML-based SFs trained and tested on experimental and on computer-generated databases. The two learning curves for our FCNN SFare compared in Figure 3, where the R_p score for the binding affinity prediction is plotted as a function of the training set size N_t. As a term of comparison, the R_p value obtained when the (negative) docking score is used as target value, both during training and testing phases, is also shown. Noticeably, remarkably high performances are obtained in this latter test, namely, $R_p \simeq 0.82$. This indicates that the chosen combination of representation and regression models is capable of learning, within good approximation, the function corresponding to the docking score. Clearly, while this result suggests that a ML-based SF can at least imitate a classic SF, it does not imply a good performance for binding affinity prediction since, as already discussed, classic SFs do not always perform well in this task. Indeed, the maximum R_p score corresponding to the binding affinity prediction is somewhat smaller, namely, $R_p \simeq 0.55$ and $R_p \simeq 0.60$, for the experimental and the computer-generated databases, respectively. These scores still correspond to moderately strong correlations between predictions and ground-truth (i.e., experimental) binding affinities. The lower score at the maximum N_t on the experimental database can be attributed to the larger size of the computer-generated one. Actually, it appears that the learning is faster on the experimental database, since higher accuracies are in fact obtained when the training set size N_t is comparable. To further inspect this effect, the same R_p scores are plotted in Figure 4 as a function of the percentage of training instances compared to the size N of the whole database. One observes an approximately linear increase. The slopes corresponding to the experimental and to the computer-generated databases are comparable. This suggests that the performance improves due to the increasing probability of finding the same or similar proteins in both the training and the test sets. In concord with this supposition, the R_p score is consistently higher on the computer-generated database, whereby the number of proteins is lower. A similar supposition has been put forward in Ref. [28], and it is supported also by the vertical tests discussed below.
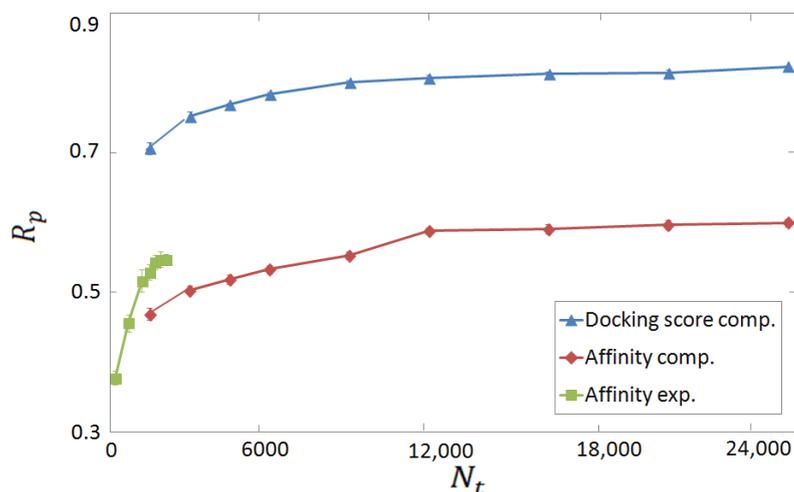
Figure 3. R_p prediction–accuracy score as a function of the training set size N_t. The green squares and the red rhombi correspond to the binding affinity prediction by the FCNN SFtrained on the experimental complexes and on the computer-generated databases, respectively. The blue triangles correspond to the predictions of the docking score by the FCNN SFtrained on the computer-generated database. The network structure is described in Figures 1 and 2.
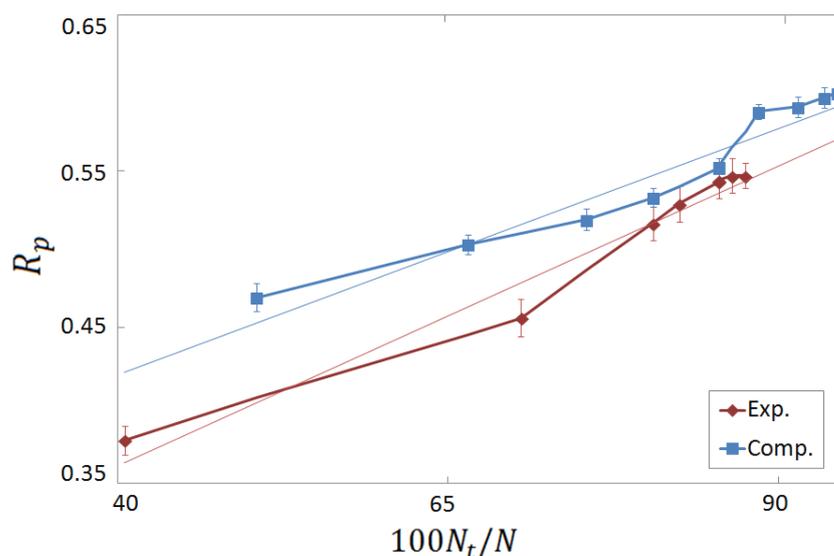
Figure 4. R_p prediction–accuracy score as a function of the percentage of the training set size N_t compared to the whole database size N, namely, $100N_t/N$. The red rhombi and the blue squares correspond to the FCNN SFtrained on the experimental complexes and on the computer-generated complexes, respectively.
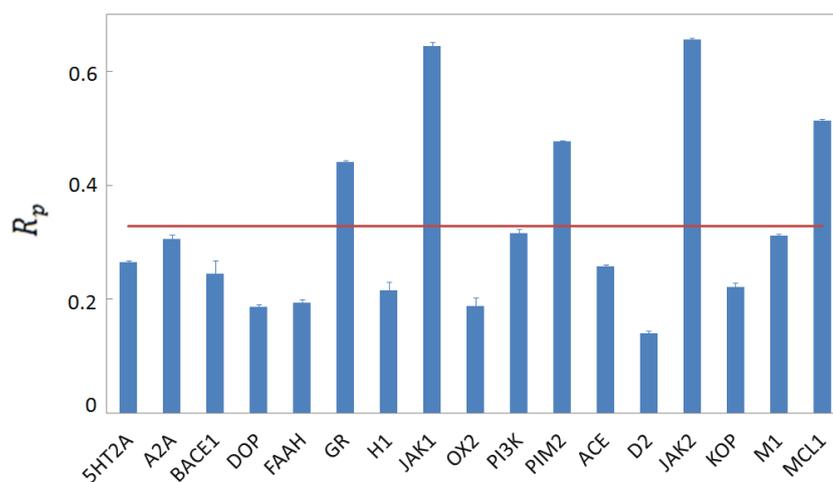
3.3. Vertical Tests

The horizontal tests described above provide encouraging results. As anticipated, however, these might be over-optimistic, being biased by the similarities among the complexes present in the training and in the test sets. In a real-case scenario, universal SFs are expected to describe the binding strength of candidate ligands into novel proteins under investigation. Quite likely, these proteins are dissimilar from those included in the databases available at the time of model definition. A fairer performance assessment is therefore provided by so-called vertical tests, whereby complexes made from proteins present in the test set are excluded from the training set. The first vertical test we consider is performed on computer-generated complexes made from the four proteins FAAH, PIM2, ACE, and MCL1, totaling 2068 complexes. The FCNN SFis trained on N_t complexes made from the remaining 13 proteins of our computer-generated database (see Table 2). The R_p score is shown in Figure 5 as a function of N_t. One notices a significantly reduced performance compared to the horizontal test. The accuracy score, $R_p \approx 0.4$, corresponds to an only moderate correlation between predicted and experimental binding affinities. It is worth mentioning that a similar vertical test on computer-generated complex structures was performed also in Ref. [27] using a random-forest regression model. That study reports an even lower score, namely, $R_p \simeq 0.2$. We attribute the better performance of our FCNN SFto the use of an FCNN, which is more suited for extracting useful information from large databases. It is also worth comparing the performance of our FCNN SFwith the one displayed by a popular classic SF. We consider the GBVI/WSA dG scoring function of the MOE software. Since the corresponding docking score is negative, with lower values corresponding to more favourable poses, we consider the negative of the docking score. Its correlation with the binding affinity turns out to be only marginal, corresponding to $R_p \simeq 0.2$. To favor further comparative studies, the code with our FCNN SF, trained on the whole subset including the 13 groups of complexes discussed above, is provided via the repository of Ref. [38]. While our FCNN SFseems to perform relatively better than the two considered benchmarks, its performance is not fully satisfactory. Chiefly, one notices that the performance does not improve with the training set size N_t. To further elucidate this finding, we perform a series of per-target vertical tests. Seventeen FCNN SF's are trained on just as many computer-generated databases obtained by excluding all complexes made from each of the 17 proteins. The excluded complexes are used as

test sets for the corresponding SF. The 17 corresponding R_p scores are shown in Figure 6. Again, the performance is, on average, appreciably lower compared to the horizontal test discussed above. This corroborates the claim that horizontal tests provide over-optimistic performance measures, probably due to the correlations and similarities among training and test complexes. In addition, the large performance fluctuations among the 17 FCNN SF's corresponding to the different targets are noteworthy. For example, remarkably high scores are obtained for the JAX1 and JAX2 proteins. These might be attributed to the similarity between these two proteins, meaning that including in the training set complexes derived from one of them allows the FCNN SFlearning how to predict binding affinities for the other. However, the close relationship between binding affinity and the ligand molecular weight might also play a role. This is further discussed below.
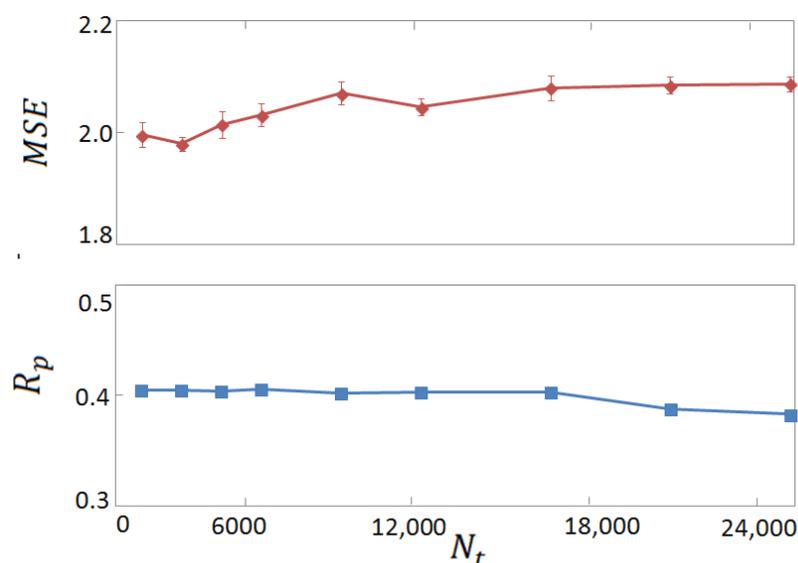

Figure 5. Mean squared error (MSE, upper panel) and R_p score (lower panel) for binding affinity prediction in the vertical test, as a function of the training set size N_t. The FCNN SFis trained on a computer-generated database including complexes made from 13 proteins, and tested on complexes made from four proteins not included in the training set.


Figure 6. Accuracy scores R_p for binding affinity affinity predictions for 17 FCNN SF's in per-target vertical tests. Each SF is trained on the computer-generated database excluding the complexes made from the protein indicated on the horizontal axis, and tested on the excluded complexes. The horizontal red line indicates the average score.

3.4. Per-Target Scoring Functions

The unremarkable performances displayed by the universal ML-based SF in the vertical tests lead us to explore different strategies. As discussed in Section 3.2, the experimental and the computer-generated databases appear to be comparably effective for training SFs for binding affinity predictions. Clearly, the computer-generated complexes can be relatively easily generated. This suggests the idea of developing per-target SFs on-demand, whenever a novel protein is targeted. Such a SF would be trained on a purposely created database including only complexes made from the target protein. To explore this direction, we consider the six proteins with more complexes in our computer-generated database, namely, D2, A2A, 5HT2A, KOP, OX2, and JAX2. Six per-target FCNN SF's are trained and tested only on complexes made from one protein. Motivated by the sizes of the six databases, we adopt FCNNs with $N_l \times N_h = 2 \times 20$, apart for the D2 target, for which the parameters $N_l \times N_h = 3 \times 20$ are expected to be more adequate. The corresponding R_p scores for binding affinity prediction are shown in Figure 7, as a function of the training set size N_t. The tests are performed on 300 complexes. The performances are relatively high. The average R_p obtained using, for each target, $N_t = 900$ training instances, is $R_p = 0.44$. When the largest available N_t for each target is employed, the average score is $R_p = 0.52$. These scores are to be compared with the average vertical test on these six targets, corresponding to $R_p = 0.30$. Chiefly, in all six tests, the R_p score systematically increases with N_t, suggesting that sufficiently performant per-target SFs can be obtained when adequately large computer-generated training databases are available. There are also noticeable performance differences among the six targets, ranging from $R_p \approx 0.4$ for D2, to $R_p \approx 0.67$ for JAK2. To shed some light on these differences, we compare the per-target FCNN SF's to simple linear regression models. Specifically, we assume the linear law $pK_i = A + mB$, where m is the ligand molecular weight (MW), and A and B are the fitting coefficients. These are fixed via MSE minimization on the whole per-target database. The comparison between the per-target FCNN SF and the corresponding MW SF is shown in Figure 8, for two exemplary targets, namely the JAK2 and OX2 proteins. Notice that the corresponding scores in the per-target vertical tests are also shown. One notices that, for the JAK2 protein, even the simple MW SF provides a remarkable performance, namely, $R_p \simeq 0.62$. Tentatively, this effect might be attributed to the large pocket size at the binding site for this particular protein, meaning that the binding strength is simply proportional to the ligand size. Clearly, additional analyses would be required to further corroborate this tentative explanation. Assuming this explanation is indeed sound, it not so surprising that both the per-target FCNN SF and the universal FCNN SF in the per-target vertical test provide comparable (but still superior) scores. Instead, for the OX2 protein, the predictions of the MW SF have essentially zero correlation with the binding affinity, corresponding to $R_p \approx 0$. The per-target FCNN SF reaches an encouraging $R_p \simeq 0.53$ at the largest N_t. Notably, this per-target FCNN SF significantly overcomes the score of the universal FCNN SF in the corresponding per-target vertical test ($R_p \simeq 0.18$). These findings suggest that per-target FCNN SF's are able to learn non-trivial mappings from computationally feasible computer-generated databases.
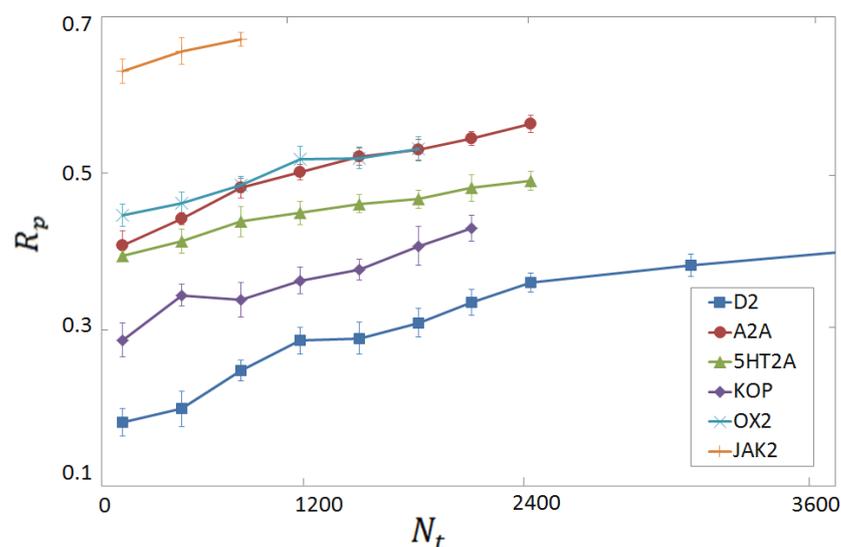
Figure 7. R_p score for binding affinity predictions from six per-target FCNN SF's, as a function of the training set size N_t. Each SF is trained and tested on computer-generated complexes made from the protein indicated in the legend.
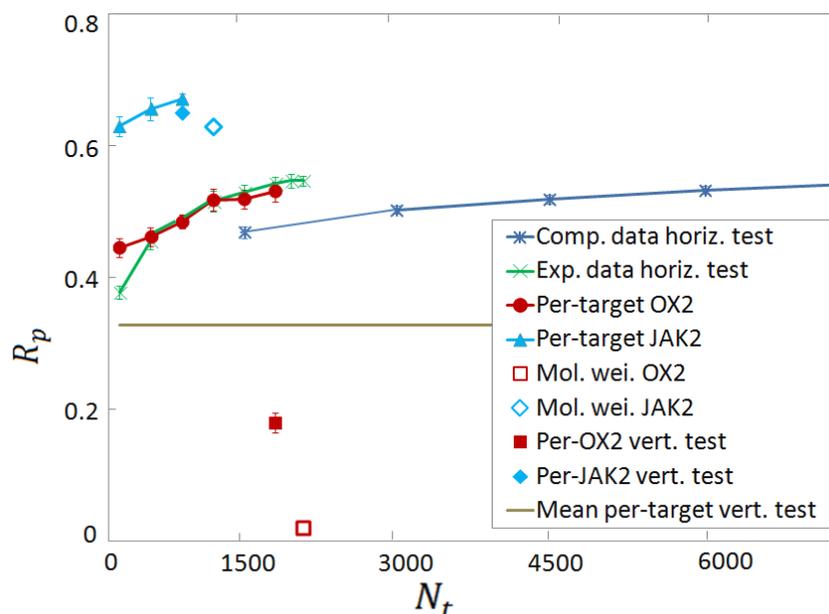
Figure 8. R_p accuracy scores as a function of the training set size N_t, for two exemplary target proteins, namely, JAK2 (cyan) and OX2 (red). The performances of the per-target FCNN SF's (cyan triangles for JAK2 and red circles for OX2) are compared to the corresponding scores of the FCNN SF in the vertical test (cyan full diamond and red full square) and of the molecular weight linear regression (cyan empty diamond and red empty square). The large green x's correspond to the universal FCNN SF in the horizontal test on the experimental database, while the small blue x's correspond to the analogous test of the computer-generated database. The horizontal gray line represents the average score on the 17 per-target vertical tests.

4. Discussion

We have analysed the performance of scoring functions (SFs) based on fully-connected neural networks (dubbed FCNN SF's) in predicting the binding affinities of protein–ligand complexes from a representation of the 3D complex structure. Our choice for the structure representation is based on atomic-pair counts within suitably chosen distance intervals, and we investigated the optimal number and width of such intervals. The effectiveness of crys-

tallographic 3D structures for SF training has been compared to that of computer-generated structures created with popular commercial docking-simulation software. Notably, the two types of data turn out to be comparably effective, suggesting that more accessible computer-generated database can be used for SF development, rather than the less copious experimental structures. Importantly, our FCNN SF's have been assessed in horizontal tests as well as in vertical tests, whereby no protein is included both in the training and in the test databases. A significant performance degradation is found in the latter, as in fact reported also in the study of Ref. [27] using different regression models compared to the neural networks adopted here. This corroborates the contention of the authors of Ref. [28], who argued that SFs based on machine learning might not learn the ligand–target binding mechanism, but rather the correlations among complexes present in the training and in the test sets. These findings indicate that vertical tests represent fairer assessments for the performances of SFs in real-case scenarios. The relative easy of creating computer-generated databases led us to explore the development of per-target FCNN SF's. Six exemplary targets have been considered, obtaining encouraging results. Chiefly, a systematic performance improvement with the training set size was observed. Furthermore, we found instructive results by comparing the FCNN SF's to linear-regression models based on the molecular weights. Interestingly, in some cases, even such simple theories reach high correlation with experimental binding affinities.

SFs are an important tool to accelerate drug discovery. Massive research endeavours have been devoted to implementing different families of SFs. However, the development of SFs based on machine learning is still in an early stage. The variable and sometimes contrasting results reported in the literature indicate that more research is due to establish reliable benchmarking protocols. We argue that the ease of generating training data via computer simulations, together with the first encouraging findings reported in this article, will favour further research endeavours aiming at developing per-target SFs based on machine learning, tailored at specific proteins or protein families. Additional techniques borrowed from the field of artificial intelligence might help with facing the problems associated with sparse, possibly biased training databases [47]. Along this line, generative neural networks have been employed to implement de-novo molecular design, avoiding virtual screening of excessively large databases [?]. Further research should also consider the use of additional complex descriptors, taking into account, e.g., entropic contributions, hydrogen-bond information, and/or pharmacophore models. Extending the considered atomic species, addressing, e.g., halogens, is also important; indeed, halogens species are included in many drug-like molecules. To facilitate future comparative studies, we provide our databases (experimental and computer-generated) via the link available in Ref. [38], including the 3D structures, deposited in PDB files, as well as the binding affinities and the docking scores. This repository also stores the code with one of our FCNN SF's. The provided SF is trained on a subset of our computer-generated database, as discussed in Section 3.3.

References

1. Kulharia, M.; Goody, R.S.; Jackson, R.M. Information Theory-Based Scoring Function for the Structure-Based Prediction of Protein- Ligand Binding Affinity. *J. Chem. Inf. Model.* **2008**, *48*, 1990–1998. [CrossRef] [PubMed]
2. Jain, A.N. Scoring functions for protein–ligand docking. *Curr. Protein Pept. Sci.* **2006**, *7*, 407–420. . [CrossRef] [PubMed]
3. Walters, W.P.; Stahl, M.T.; Murcko, M.A. Virtual screening—An overview. *Drug Discov. Today* **1998**, *3*, 160–178. . [CrossRef]
4. Wienkers, L.C.; Heath, T.G. Predicting in vivo drug interactions from in vitro drug discovery data. *Nat. Rev. Drug Discov.* **2005**, *4*, 825–833. [CrossRef] [PubMed]
5. Drews, J. Drug discovery: A historical perspective. *Science* **2000**, *287*, 1960–1964. . [CrossRef] [PubMed]
6. Liu, J.; Wang, R. Classification of current scoring functions. *J. Chem. Inf. Model.* **2015**, *55*, 475–482. . [CrossRef]
7. Gohlke, H.; Klebe, G. Statistical potentials and scoring functions applied to protein–ligand binding. *Curr. Opin. Struct. Biol.* **2001**, *11*, 231–235. . [CrossRef]
8. Gohlke, H.; Hendlich, M.; Klebe, G. Knowledge-based scoring function to predict protein–ligand interactions. *J. Mol. Biol.* **2000**, *295*, 337–356. [CrossRef]
9. Yin, S.; Biedermannova, L.; Vondrasek, J.; Dokholyan, N.V. MedusaScore: An accurate force field-based scoring function for virtual drug screening. *J. Chem. Inf. Model.* **2008**, *48*, 1656–1662. [CrossRef]
10. 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. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2015**, *5*, 405–424. [CrossRef]
11. Li, H.; Sze, K.H.; Lu, G.; Ballester, P.J. Machine-learning scoring functions for structure-based drug lead optimization. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2020**, *10*, e1465. [CrossRef]
12. Li, H.; Sze, K.H.; Lu, G.; Ballester, P.J. Machine-learning scoring functions for structure-based virtual screening. *Wiley Interdiscip. Rev. Comput. Mol. Sci.* **2021**, *11*, e1478. [CrossRef]
13. Palmer, R.A.; Niwa, H. X-ray crystallographic studies of protein–ligand interactions. *Biochem. Soc. Trans.* **2003**, *31*, 973–979. [CrossRef]
14. Ballester, P.J.; Mitchell, J.B.O. A machine learning approach to predicting protein–ligand binding affinity with applications to molecular docking. *Bioinformatics* **2010**, *26*, 1169–1175. [CrossRef] [PubMed]
15. Svetnik, V.; Liaw, A.; Tong, C.; Culberson, J.C.; Sheridan, R.P.; Feuston, B.P. Random forest: A classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 1947–1958. [CrossRef] [PubMed]
16. 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. [CrossRef] [PubMed]
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. [CrossRef]
18. Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the basis for developing protein–ligand interaction scoring functions. *Accounts Chem. Res.* **2017**, *50*, 302–309. [CrossRef]
19. 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. [CrossRef]
20. 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. [CrossRef]
21. 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. [CrossRef]
22. Gomes, J.; Ramsundar, B.; Feinberg, E.N.; Pande, V.S. Atomic convolutional networks for predicting protein–ligand binding affinity. *arXiv* **2017**, arXiv:1703.10603.
23. Seo, S.; Choi, J.; Park, S.; Ahn, J. Binding affinity prediction for protein–ligand complex using deep attention mechanism based on intermolecular interactions. *BMC Bioinform.* **2021**, *22*, 542. [CrossRef] [PubMed]
24. 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. [CrossRef] [PubMed]

25. Li, S.; Zhou, J.; Xu, T.; Huang, L.; Wang, F.; Xiong, H.; Huang, W.; Dou, D.; Xiong, H. Structure-aware interactive graph neural networks for the prediction of protein–ligand binding affinity. In Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Virtual Event, 14–18 August 2021; pp. 975–985. [CrossRef]

26. Emmert-Streib, F.; Yang, Z.; Feng, H.; Tripathi, S.; Dehmer, M. An Introductory Review of Deep Learning for Prediction Models With Big Data. *Front. Artif. Intell.* **2020**, *3*, 4. [CrossRef]

27. Wójcikowski, M.; Ballester, P.J.; Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci. Rep.* **2017**, *7*, 46710. [CrossRef] [PubMed]

28. 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. [CrossRef]

29. Warren, G.L.; Do, T.D.; Kelley, B.P.; Nicholls, A.; Warren, S.D. Essential considerations for using protein–ligand structures in drug discovery. *Drug Discov. Today* **2012**, *17*, 1270–1281. [CrossRef]

30. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]

31. Jia, X.; Lynch, A.; Huang, Y.; Danielson, M.; Lang'at, I.; Milder, A.; Ruby, A.E.; Wang, H.; Friedler, S.A.; Norquist, A.J.; et al. Anthropogenic biases in chemical reaction data hinder exploratory inorganic synthesis. *Nature* **2019**, *573*, 251–255. [CrossRef]

32. Molecular Operating Environment (MOE), 2022.02 Chemical Computing Group ULC, 1010 Sherbooke St. West, Suite #910, Montreal, QC, Canada, H3A 2R7, 2023. Available online: https://www.chemcomp.com/index.htm (accessed on 1 February 2020).

33. Jones, G.; Willett, P.; Glen, R.C.; Leach, A.R.; Taylor, R. Development and validation of a genetic algorithm for flexible docking. *J. Mol. Biol.* **1997**, *267*, 727–748. [CrossRef]

34. Greenidge, P.A.; Lewis, R.A.; Ertl, P. Boosting Pose Ranking Performance via Rescoring with MM-GBSA. *Chem. Biol. Drug Des.* **2016**, *88*, 317–328.

35. Drenth, J. *Principles of Protein X-ray Crystallography*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2007.

36. 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. [CrossRef]

37. The Protein Data Bank. Available online: https://www.rcsb.org/ (accessed on 1 February 2020).

38. Pellicani, F.; Dal Ben, D.; Perali, A.; Pilati, S. Data for "Machine Learning Scoring Functions for Drug Discovery from Experimental and Computer-Generated Protein–Ligand Structures: Towards Per-Target Scoring Functions". Available online: https://zenodo.org/record/7514055#.Y-SoVEFBxD8 (accessed on 1 December 2022).

39. Chen, X.; Liu, M.; Gilson, M.K. BindingDB: A web-accessible molecular recognition database. *Comb. Chem. High Throughput Screen.* **2001**, *4*, 719–725. [CrossRef] [PubMed]

40. Chen, X.; Lin, Y.; Liu, M.; Gilson, M.K. The Binding Database: Data management and interface design. *Bioinformatics* **2002**, *18*, 130–139. [CrossRef] [PubMed]

41. Liu, T.; Lin, Y.; Wen, X.; Jorissen, R.N.; Gilson, M.K. BindingDB: A web-accessible database of experimentally determined protein–ligand binding affinities. *Nucleic Acids Res.* **2007**, *35*, D198–D201. [CrossRef] [PubMed]

42. Falsini, M.; Catarzi, D.; Varano, F.; Dal Ben, D.; Marucci, G.; Buccioni, M.; Volpini, R.; Di Cesare Mannelli, L.; Ghelardini, C.; Colotta, V. Novel 8-amino-1,2,4-triazolo[4,3-a]pyrazin-3-one derivatives as potent human adenosine A1 and A2A receptor antagonists. Evaluation of their protective effect against *β*-amyloid-induced neurotoxicity in SH-SY5Y cells. *Bioorganic Chem.* **2019**, *87*, 380–394. [CrossRef]

43. Ceni, C.; Catarzi, D.; Varano, F.; Ben, D.D.; Marucci, G.; Buccioni, M.; Volpini, R.; Angeli, A.; Nocentini, A.; Gratteri, P.; et al. Discovery of first-in-class multi-target adenosine A2A receptor antagonists-carbonic anhydrase IX and XII inhibitors. 8-Amino-6-aryl-2-phenyl-1,2,4-triazolo [4,3-a]pyrazin-3-one derivatives as new potential antitumor agents. *Eur. J. Med. Chem.* **2020**, *201*, 112478. [CrossRef]

44. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.

45. Chollet, F. Keras. 2015. Available online: https://keras.io (accessed on 1 June 2020).

48. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**,

47. Brown, N.; Cambruzzi, J.; Cox, P.J.; Davies, M.; Dunbar, J.; Plumbley, D.; Sellwood, M.A.; Sim, A.; Williams-Jones, B.I.; Zwierzyna, M.; et al. Big Data in Drug Discovery. *Prog. Med. Chem.* **2018**, *57*, 277–356. [CrossRef]

48. Brown, N.; Fiscato, M.; Segler, M.H.; Vaucher, A.C. GuacaMol: Benchmarking Models for de Novo Molecular Design. *J. Chem. Inf. Model.* **2019**, *59*, 1096–1108. . [CrossRef]