JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Article

Cite This: J. Chem. Inf. Model. 2018, 58, 287−296

pubs.acs.org/jcim

K_{DEEP}: Protein−Ligand Absolute Binding Affinity Prediction via 3D-Convolutional Neural Networks

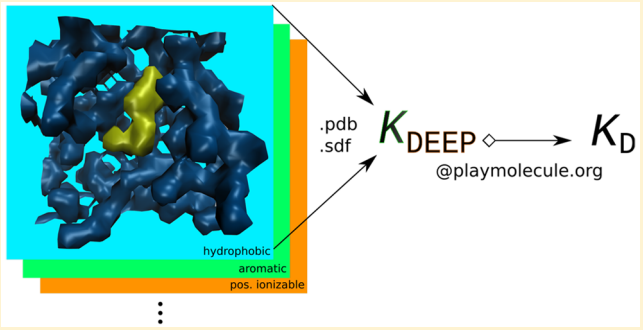
José Jiménez,[†] Miha Škalič,[†] Gerard Martínez-Rosell,[†] and Gianni De Fabritiis*[,†,‡]

[†]Computational Biophysics Laboratory, Universitat Pompeu Fabra, Parc de Recerca Biomèdica de Barcelona, Carrer del Dr. Aiguader 88, Barcelona 08003, Spain

[‡]Institució Catalana de Recerca i Estudis Avançats (ICREA), Passeig Lluis Companys 23, 08010 Barcelona, Spain

Supporting Information

ABSTRACT: Accurately predicting protein−ligand binding affinities is an important problem in computational chemistry since it can substantially accelerate drug discovery for virtual screening and lead optimization. We propose here a fast machine-learning approach for predicting binding affinities using state-of-the-art 3D-convolutional neural networks and compare this approach to other machine-learning and scoring methods using several diverse data sets. The results for the standard PDBbind (v.2016) core test-set are state-of-the-art with a Pearson's correlation coefficient of 0.82 and a RMSE of 1.27 in pK units between experimental and predicted affinity, but accuracy is still very sensitive to the specific protein used. K_{DEEP} is made available via PlayMolecule.org for users to test easily their own protein−ligand complexes, with each prediction taking a fraction of a second. We believe that the speed, performance, and ease of use of K_{DEEP} makes it already an attractive scoring function for modern computational chemistry pipelines.



INTRODUCTION

Docking and virtual screening pipelines use *scoring functions* to rank putative poses of a particular ligand in a binding site. Among these, popular ones are the ones used by GOLD,[1] SurFlex Dock,[2] Glide,[3] or AutoDock Vina.[4] These functions typically fulfill several desirable properties, depending on the task: (1) They rank putative poses of a ligand higher than others (docking power), (2) distinguish binding from nonbinding ligands (binding power), and (3) correlate with experimentally determined binding affinity (scoring power). Here, we focus on the latter: accurately predicting protein−ligand binding affinity using structural information from both. Simulation-based approaches such as free energy perturbation methods, despite being sensitive to ligand parametrization and force field selection,[5] have proven to be very successful in this task.[6−9] However, their computational cost limits their use in large libraries of compounds.

Machine-learning approaches are typically several orders of magnitude faster, their performance heavily depending on both featurization and choice of model.[10] These algorithms, commonly posed as a regression problems for predicting protein−ligand binding affinities are not novel. Some early approaches[11,12] use classical statistical approaches such as linear models, their learned coefficients taking into account features such as hydrogen bonds, hydrophobicity, or van der Waals surface. It has been discussed[10] that these approaches cannot efficiently approximate free energies due to the nonflexible and simple nature of their linear modeling relationship. Arguably, the first machine-learning method to achieve high performance on a well-known benchmark was RF-Score,[13] using a combination of protein−ligand atom-type pair counts on the binding site neighborhood and a random forest (RF)[14] model. Other similar approaches followed with similar performance. ID-Score,[15] for instance, uses 50 different protein−ligand binding descriptors such as van der Waals interactions, electrostatics, dissolution effects, or shape-related and a Support Vector Machine (SVM) model.[16] Other authors[17] have used descriptors provided by Autodock Vina and a RF model, finding increased performance. SFCScore(RF)[18] uses this approach with the original SFCScore[12] descriptors, finding similar results. An in-depth review of machine-learning-based scoring functions was recently published.[10]

Recently, deep learning-based[19,20] approaches have rapidly emerged to provide state-of-the-art performance in fields such as computer vision,[21] natural language processing,[22] and generative modeling.[23] However, the promise of deep learning in structural biology and computational chemistry has yet to be fully developed,[24] but it is becoming increasingly common.[25−33] Driven by these developments and to find out how this class of models perform in molecular scoring tasks, we hereby propose an end-to-end framework, named K_{DEEP}, based on 3D-convolutional neural networks for predicting protein−ligand absolute affinities. We extensively compare its performance to other existing approaches in several data sets such as the PDBbind v.2016 benchmark,[34] several CSAR data sets, and other recently published

Received: November 9, 2017
Published: January 8, 2018

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

congeneric series sets. This work also serves as an updated benchmark for the rest of the methodologies on unseen data. Similar to other large-scale scoring tools,[35,36] we provide a web application in PlayMolecule.org so that users can test with their own protein−ligand complexes.

MATERIALS

In this section, we describe in detail the data used for both training and benchmarking, as well as the architecture of the proposed model.

Data Sets. For the work described here, we mainly use the PDBbind (v.2016) database, containing 13,308 protein−ligand complexes and their corresponding experimentally determined binding affinities collected from literature and the Protein Data Bank (PDB), in terms of a dissociation (K_d), inhibition (K_i) or half-concentration (IC_{50}) constant. A smaller *refined* subset (n_r = 4057)[37] is extracted from it following quality protocols addressing structural resolution and experimental precision of the binding measurement. These controls exclude complexes with a resolution higher than 2.5 Å, or an R-factor higher than 0.25, ligands bound through covalent bonds, ternary complexes or steric clashes, and affinity not reported either in K_d or K_i or falling out of a desired range (K_d < 1pM) among other criteria. A *core* set (n_c = 290) is selected from the larger refined set as a representative nonredundant subset for benchmarking purposes. The core set is then clustered using a 90% sequence similarity

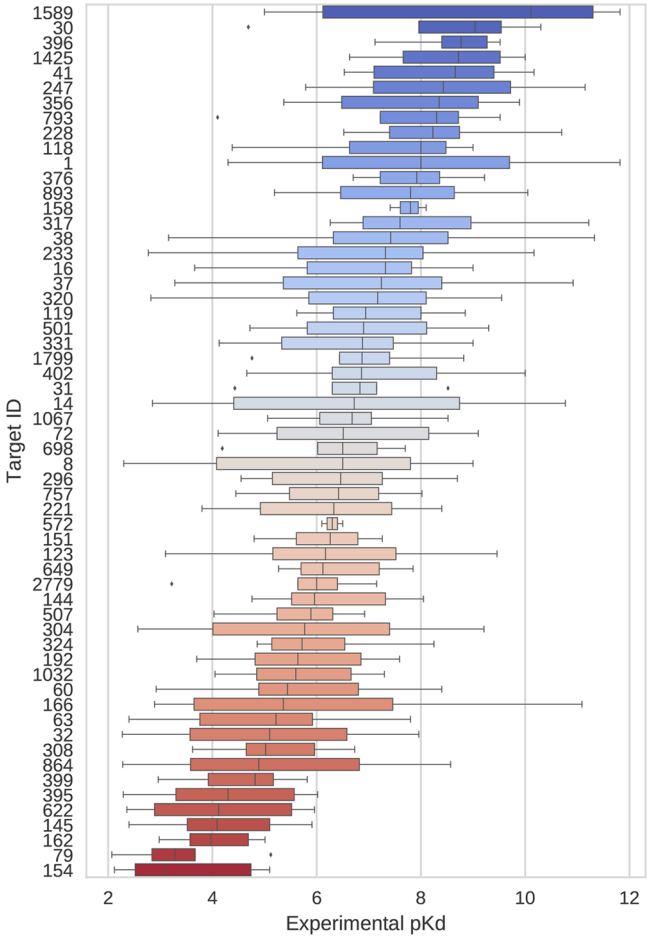


Figure 1. Boxplot of experimental affinities (in *pK* values) for each of the 58 targets in the PDBbind v.2016 *core* set, sorted by their median.

Table 1. Descriptive Information Regarding the Complexes Taken from Wang et al.[6] and Wan et al.[7]

Target	PDB Id	# ligands	Affinity range (kcal/mol)
BACE	4DJW	36	3.5
CDK2	1H1R	16	4.2
JNK1	2GMX	21	3.4
MCL1	4HW3	42	4.2
p38	3FLY	34	3.8
PTP1B	2QBS	23	5.1
Thrombin	2ZFF	11	1.7
Tyk2	4GIH	16	4.3
TRK-Kinase	5JFV	16	4.6

Table 2. Rules Defined for Three-Dimensional Descriptors Described in This Work

Property	Rule
Hydrophobic	Aliphatic or aromatic C
Aromatic	Aromatic C
Hydrogen bond acceptor	Acceptor 1 H-bond or S Spherical N; Acceptor 2 H-bonds or S Spherical O; Acceptor 2 H-bonds S
Hydrogen bond donor	Donor 1 H-bond or Donor S Spherical H with either O or N partner
Positive ionizable	Gasteiger positive charge
Negative ionizable	Gasteiger negative charge
Metallic	Mg, Zn, Mn, Ca, or Fe
Excluded volume	All atom types

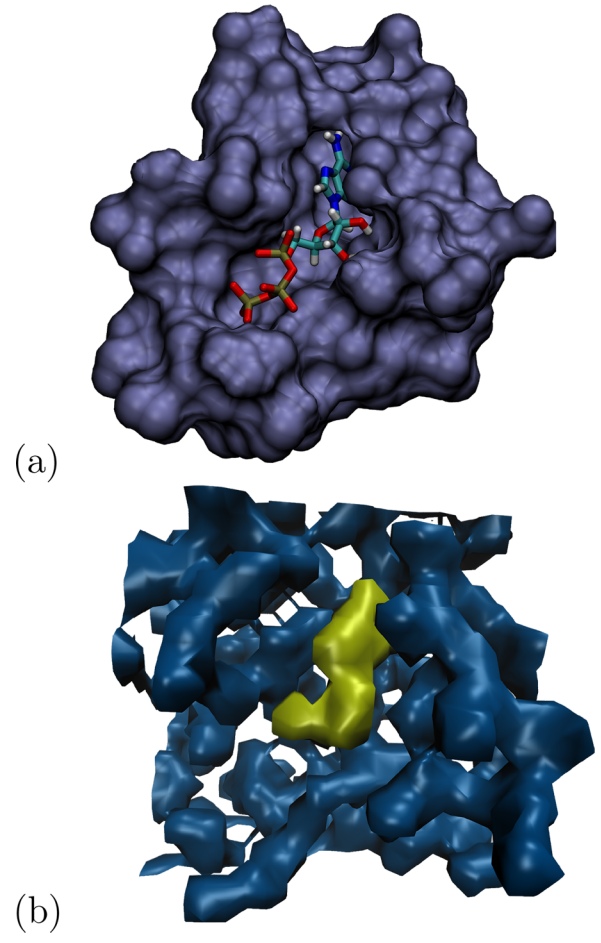


Figure 2. (a) PDB Id 2HMU pocket and bound ligand ATP. (b) Voxel representation of the hydrophobic channel for both protein (blue) and ligand (yellow).

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

cutoff, providing in practice 58 different targets. Disaggregated experimental information for each of these targets is provided in Figure 1, where most targets exhibit a wide range of affinities, with an average standard deviation of 1.77 pK units. Previous iterations of the PDBbind database have been extensively used for the same goal of benchmarking binding affinity prediction methods.[38]

Some authors[39−41] have highlighted that the PDBbind database standard train and test splitting procedure tends to yield overly optimiztic results when used for developing machine-learning models. In order to provide fairer evaluation of our proposed model against other methodologies, we used several additional data sets found in the literature. First, we consider several iterations of the well-known data sets provided by CSAR

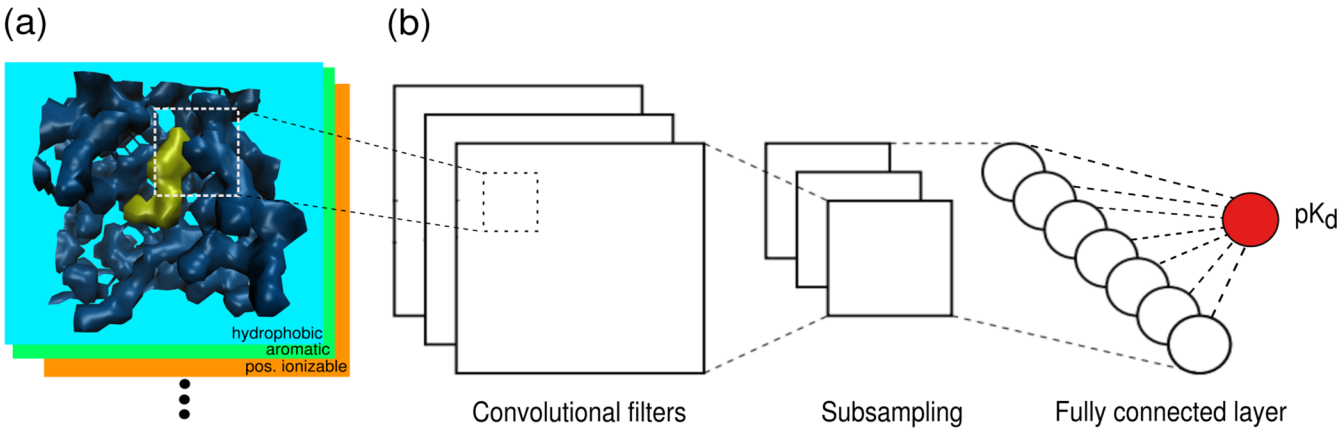


Figure 3. Workflow schema of the proposed model. (a) Both protein and ligand are featurized via a voxelized 24 Å representation of the binding site considering eight pharmacophoric-like properties. (b) These descriptors are used by a three-dimensional convolutional neural network model, which in turn learns the binding affinity of the complex given enough training examples. Once trained, the network is used to predict unseen examples.

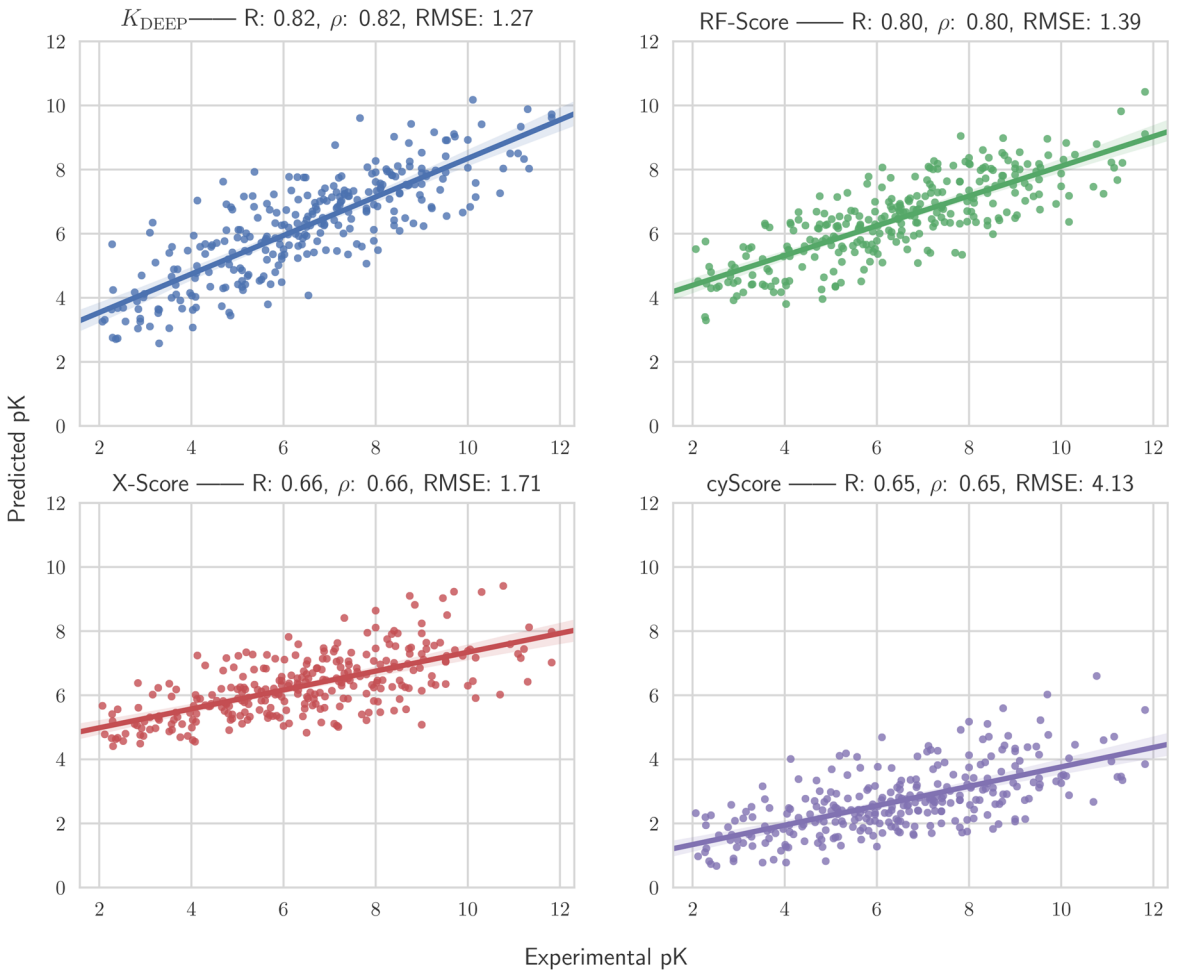


Figure 4. Comparison of prediction methods in the PDBbind v.2016 core set (n_c = 290). Metrics reported are Pearson's correlation coefficient (R), Spearman's correlation (ρ), and Root Mean Squared Error (RMSE). Baselines log P and molecular weight correlations R_p = 0.32 and R_w = 0.49, respectively.

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

(csardock.org). Four different data sets are taken into account, the CSAR NRC-HiQ data set,[42] composed of two subsets composed of 176 and 167 protein−ligand complexes, respectively, extracted from both the BindingMOAD[43] database and previous iterations of the PDBbind data set. Due to the overlap with our training set, complexes present in both were removed for testing purposes based on PDB Id, resulting in two sets of 55 and 49 complexes, respectively. Another data set, named here CSAR2012, was constructed with 57 protein−ligand complexes detailed in ref 44 and downloaded from the D3R challenge (https://drugdesigndata.org/) web site, composed of complexes CDK2-Cyclin A, CDK2-Kinase, CHK1-Kinase, ERK2, LpXc, and Urokinase. Similarly, a final data set named here CSAR2014, with 47 complexes provided by the D3R consortium was built compassing targets SYK, tRMD, and HSP90. Finally, we consider several congeneric series data sets[6,7] used in free energy perturbation studies, compassing targets such as CDK2, BACE, TRK-Kinase, or p38 among others (Table 1).

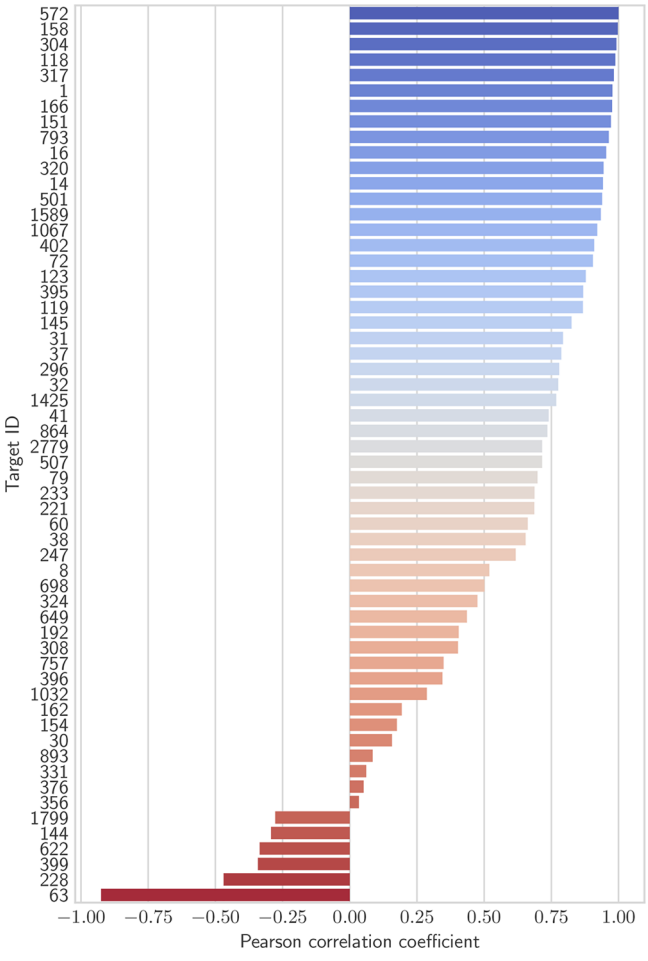


Figure 5. Dissagregated Pearson's correlation coefficient for the 58 targets in the PDBbind v.2016 *core* set, in descending order (higher is better).

Training and Test Splits. For training of our models, we consider different splits. First, as is common in previous studies,[13,18] we use the difference between the *refined* and *core* subsets in the PDBbind database for model fitting. This ensures that the same protein−ligand complex does not fall in both train and test split simultaneously. PDB Ids for all complexes in each of the described splits are provided in the Supporting Information. For all purposes, we do no make the distinction between K_d and K_i dissociation and inhibition constants and consider the $pK = -\log_{10} K$ version of these values. The resulting training and test set sizes are composed of $n_t = 3767$ and $n_c = 290$ protein−ligand complexes, respectively. The rest of the complexes considered in the study are only used for benchmarking purposes. Regarding the set of ligands in Wang et al,.[6] we used the input structures provided by the authors in the Supporting Information, while for the ones in Wan et al.[7] we took the crystal structure deposited in the PDB Id (5JFV), generated and protonated conformers of each of the provided SMILEs with rdkit,[45] and used the tethered docking protocol in rDock[46] along with the maximum substructure procedure (MCS) available in rdkit.

METHODS

Descriptors. We adapted the set of descriptors detailed in Jiménez et al.[25] for both protein and ligand. Other authors[27] have used a similar set of descriptors to represent protein−ligand complexes in order to discriminate active compounds. In particular, we use a 3D voxel representation of both proteins and ligand using a van der Waals radius r_{vdw} for each atom type, which in turns gets assigned to a particular property channel (hydrophobic, hydrogen-bond donor or acceptor, aromatic, positive or negative ionizable, metallic and total excluded volume), according to the rules described in Table 2. The contribution of each atom to each grid point depends on their Euclidean distance r according to eq 1.

$$n(r) = 1 - \exp\left(-\left(\frac{r_{vdw}}{r}\right)^{12}\right) \quad (1)$$

For the work described here, we duplicate the number of properties to account for both protein and ligand, by using the same ones in each, up to a total of 16 different channels. These descriptors are computed on a fixed 24 Å^3 subgrid centered on the geometric center of the ligand, in practice capturing a neighborhood of the binding site. A plot with the hydrophobic voxelized representation for both protein and ligand are shown in Figure 2. The voxelization routines are available within the HTMD Python framework for molecular discovery.[47] Initially we also considered including channels with crystal water and B-value information. For the sake of simplicity, we decided to keep the descriptors as simple as possible.

Table 3. Pearson's Correlation (R) between Experimental and Predicted Binding Affinity in Four CSAR Data Sets

	K_{DEEP}	RF-Score	cyScore	X-Score	log P	mol. weight
CSAR NRC-HiQ set 1	0.72[a]	0.77[a]	0.65[a]	0.6[a]	0.33	0.28
CSAR NRC-HiQ set 2	0.65[a]	0.75[a]	0.64[a]	0.65[a]	0.44[a]	0.44[a]
CSAR12	0.37[a]	0.46[a]	0.26	0.48[a]	0.17	0.4[a]
CSAR14	0.61[a]	0.8[a]	0.67[a]	0.82[a]	0.22	0.82[a]
Weighted average	0.58	**0.69**	0.55	0.63	0.29	0.47
Simple average	0.59	**0.7**	0.56	0.64	0.29	0.54

[a]Correlation significant at $\alpha = 0.01$.

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

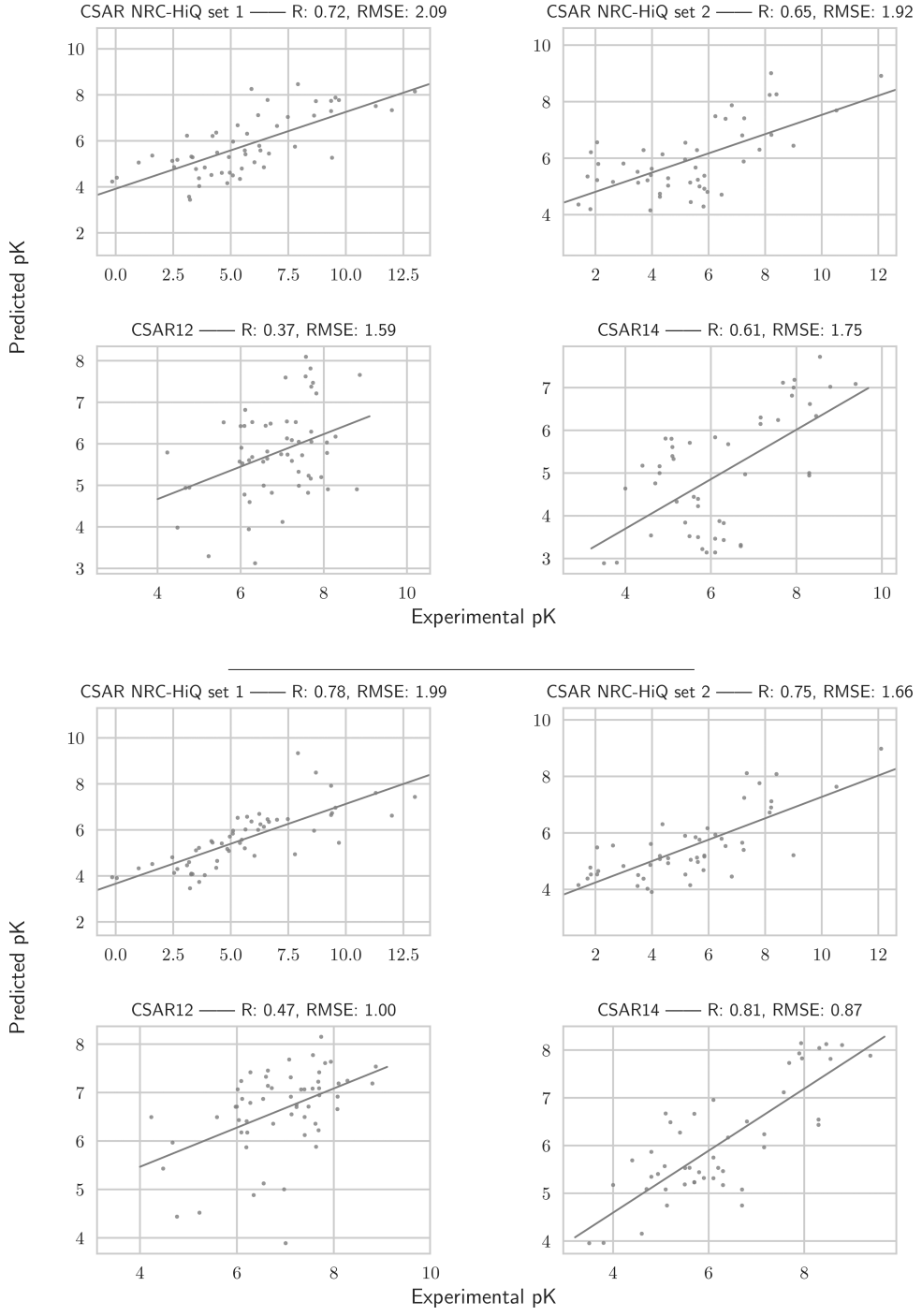


Figure 6. CSAR data set results, for K_{DEEP} (above) and RF-Score (below). In this case, RF-Score outperforms the CNN model in terms of correlation between experimental and predicted binding affinity.

Convolutional Neural Network Architecture. CNNs (Convolutional Neural Networks) models are similar to regular fully connected neural networks. In the latter case, the output of a given neuron ϕ is a nonlinear transformation f of a dot product of the neurons $\boldsymbol{x}$ of the previous layer and learnable weights $\boldsymbol{w}$, plus a bias b:

$$\phi = f\left(\sum_i w_i x_i + b\right) \tag{2}$$

These neurons are arranged sequentially in several layers, therefore providing means of fitting highly complex nonlinear functions. However, these networks tend not to scale well when the input is high dimensional (such as the case of images or voxels) (Figure 3). CNNs on the other hand, are a type of neural network specifically designed to deal with data where local spatial information needs to be taken into account. While a regular network would practically ignore such interactions, a convolutional one arranges its neurons spatially, and only connects locally to the output of the previous layer. The design of CNNs entails many architectural choices to account for number of hidden layers, number of filters, or their size are some, to name a few. It is common practice to adapt one of the architectures proven successful in computer vision applications, such as Resnets,[48] VGG,[49]

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

or SqueezeNet[50] networks, to a particular application. In the work described here, we adapt the latter for 3D convolution purposes, simplifying its depth due to training sample size and image resolution constraints but keeping the rest of the architecture fixed, including the choice of rectified linear units (ReLUs) as activation function. 3D-convolutions are more expensive than 2D-convolutions, as they require both more memory and computation. Despite so, we did not encounter any computational related problems when training on voxels of the described size. A small graph representation of the network is shown in Figure S1. Originally, this architecture was inspired by AlexNet,[21] achieving a similar level of performance on the ImageNet[51] challenge by considering smaller filter sizes of 3 and 1, respectively, causing overall a smaller number of trainable parameters. In our case, the total number of learnable parameters adds up to 1,340,769. Regarding the choice of network optimizer, it is common practice to use stochastic gradient descent or an adaptive variant. We chose Adam,[52] with default parameters for momentum scheduling ($\beta_1 = 0.99$, $\beta_2 = 0.999$), using a batch size of 128 samples for 50 epochs and an initial learning rate of 10^{-4}. We train our network from scratch using Glorot uniform weight initialization.[53] Once the model is trained, it can be used for predicting unseen protein−ligand pairs. We augment our data by rotating each subgrid 90 deg, providing 24 times our initial training size. This augmenting methodology is also used at prediction time and then averaged out, reducing variance.

Protocol for Other Methods. In the Results section, we compare with three other machine-learning scoring functions: the well-known RF-Score,[13] X-Score,[54] and cyScore.[55] We reimplemented the first according to the instructions provided in Ballester et al.[13] as it was originally trained using a different version of the PDBbind database (v.2007), using the scikit-learn[56] random forest implementation and its third iteration of descriptors (RF-Score v.3). We followed the random forest modeling guidelines of using $p = 500$ trees and optimized for the maximum selected number of features in each subtree using out of bag (OOB) error in the *refined* minus *core* set. We fix this optimal value to be 6, but error estimates for different values of this hyperparameter can be found in Figure S2. For X-Score and cyScore, we used the binaries publicly made available by the authors. Comparisons with log P and molecular weight are also presented for sanity checking in all cases, also providing baselines.

RESULTS

As in previous studies,[13,57] we measure Root Mean Square Error (RMSE) and Pearson's correlation coefficient (R)

$$\mathrm{RMSE}(\boldsymbol{y}, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3}$$

$$R(\boldsymbol{y}, \hat{y}) = \frac{\sum_{i=1}^{n}(y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}} \tag{4}$$

where y_i and $\hat{y}_i$ represent experimental and predicted affinity i. The first measures the ability of the method to correctly identify a small enough prediction range where experimental affinities lie, while the second measures the performance of the method when ranking complexes linearly.

We first present results concerning the $n_c = 290$ structures corresponding to the PDBbind *core* set in Figure 4. K_{DEEP} is able to outperform the rest of the methods, with a similar correlation coefficient as RF-Score (one-tailed correlation difference test $z = -0.7$, p−value = 0.242), while achieving significantly lower error in terms of RMSE (one-tailed t test $t = 2.21$, p−value = 0.014). All models presented here outperform the simple log P and molecular weight benchmarks ($R_p = 0.32$, $\rho_p = 0.37$ and $R_w = 0.59$, $\rho_w = 0.51$ respectively). Dissagregated Pearson correlation performance per target (Figure 5) suggest that our model is able to predict accurately absolute binding affinity for most examples in this data set, given enough training samples. Additional metrics such as RMSE and Spearman's ρ are available in Figures S3 and S4.

Results for the CSAR sets are shown in Table 3 and Figure 6. RF-Score offers the best average performance in these sets, although correlations are considerably more moderate for both this method and K_{DEEP}, while more consistent in the case of the other two simpler approaches, supporting the hypothesis that more complex machine-learning methods tend to underperform outside the training manifold.[40] Using a significance level of alpha = 0.01, RF-Scores does not significantly outperform any other method apart from log P and molecular weight in any of these sets. Since measuring the overall ability of our method to score unseen compounds is the ultimate goal of this study, we mainly report Pearson's correlation coefficient (R), but other metrics such as Spearman's rank correlation (ρ) and RMSE are available in Tables S1 and S2 and Figures S5 and S6. Our main conclusion on these sets is that all methods perform similarly, with RF-Score providing a slight advantage in terms of average correlation.

The rest of benchmarking set results are presented in Table 4 and Figure 7. Here, our model performs better than the its alternatives for most of the presented sets in terms of ranking,

Table 4. Pearson's Correlation (R) among Predicted and Experimental Affinities on Blind Targets Described in Table 1

Target	K_{DEEP}	RF-Score	cyScore	X-Score	log P	mol. weight
BACE	−0.06	−0.14	0.24	−0.12	−0.07	−0.15
CDK2	0.69[a]	−0.23	−0.55	−0.17	−0.84	0.49
JNK1	0.69[a]	0.61[a]	0.72[a]	−0.56	−0.54	0.06
MCL1	0.34	0.52[a]	0.48[a]	0.39[a]	0.48[a]	0.53[a]
p38	0.36	0.48[a]	−0.14	0.36	0.27	0.57
pTP1B	0.58[a]	0.26	0.48	0.73[a]	0.39	0.74[a]
Thrombin	0.58	0.08	0.83[a]	0.78[a]	0.91[a]	0.47
Tyk2	−0.22	0.41	0.82[a]	−0.19	−0.24	0.05
TRK-Kinase	0.46	0.53	0.38	0.33	−0.33	0.23
Weighted average	**0.34**	0.3	0.32	0.17	0.05	0.33
Simple average	**0.38**	0.28	0.36	0.17	0.003	0.33

[a]Correlation significant at $\alpha = 0.01$.

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

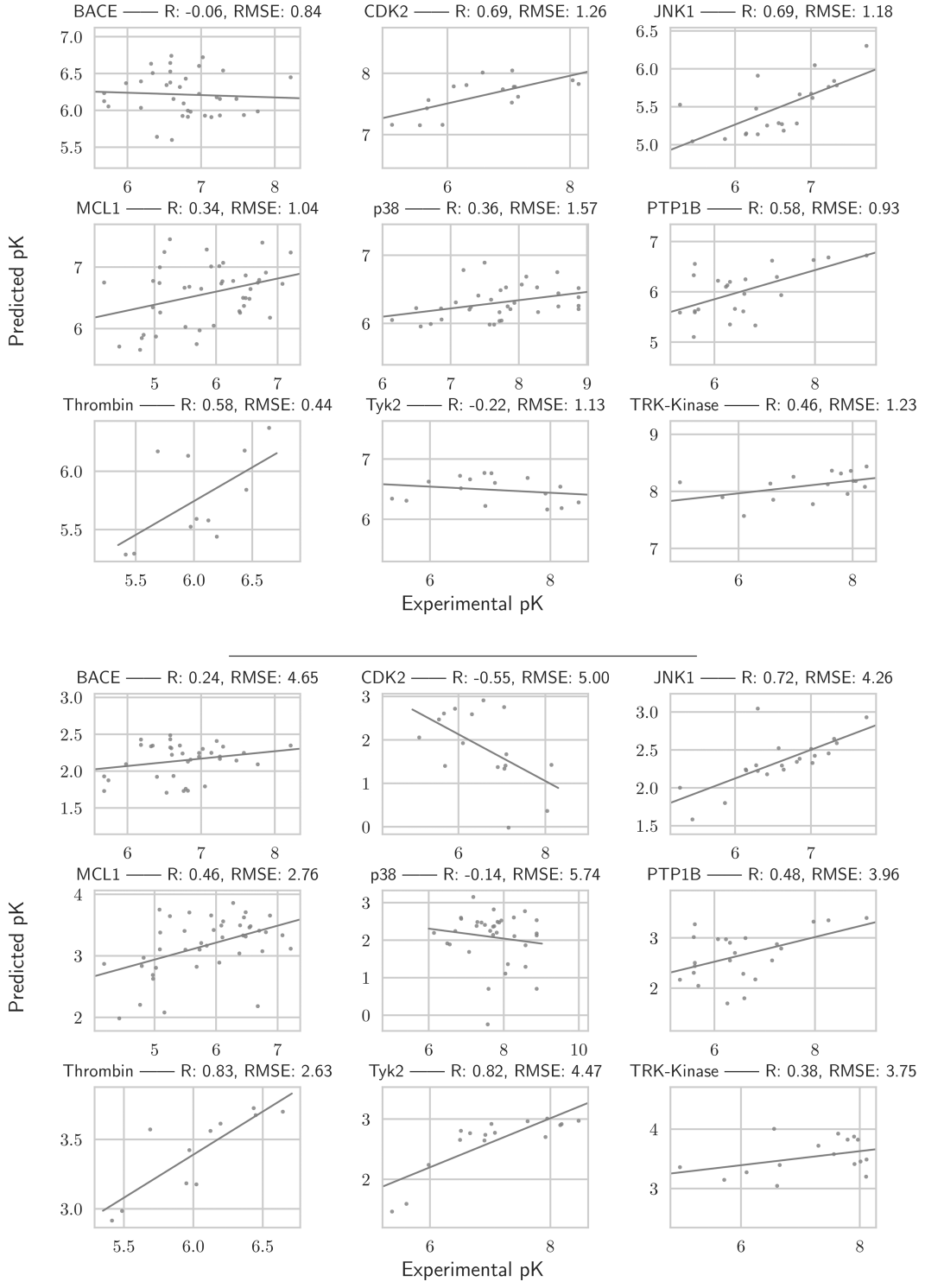


Figure 7. Results from the data sets taken from Wang et al.[6] and Wan et al.[7] for both K_{DEEP} (above) and cyScore (below). The CNN model outperforms RF-Score in these sets and is competitive with the second best method, cyScore, in terms of correlation between predicted and experimental binding affinity.

providing helpful predictions for several targets. Interestingly, the molecular weight baseline proves to be competitive with the rest of the methods, surpassing all approaches except K_{DEEP} in average correlation. Using a significance level of α = 0.01, K_{DEEP} outperforms the second best performing model (in these sets cyScore) in the CDK2 target, while the latter outperforms the former in target Tyk2. The same conclusion is drawn for a comparison with RF-Score. Against X-Score, K_{DEEP} performs significantly better in targets CDK2 and JNK1. No other significant difference against K_{DEEP} is seen for the remaining targets. Additional evaluation metrics can be checked in Tables S3 and S4 and Figures S7 and S8. To draw a comparison to other methodologies, in Wang et al.,[6] an average R_{FEP} = 0.67 was reported for the more expensive FEP methods and an average of R_{GL} = 0.29 for Glide-SP. Results in general suggest that K_{DEEP} is better than the rest of the other scoring functions in

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

terms of average correlation, but said conclusion is overshadowed by the fact that molecular weight provides such a strong baseline in these sets. Furthermore, it is important to stress that current machine-learning scoring function performance is still significantly behind modern free energy perturbation methods, if cost is not taken into account.

Finally, a previous study[58] suggested that including low-quality affinity data tends to improve the accuracy of machine-learning scoring functions. We tested this by using the PDBbind *full* minus *core* set as training and saw no significant performance difference in any test set.

IMPLEMENTATION AND ACCESSIBILITY

Users can access K_{DEEP} though an intuitive web application available in PlayMolecule.org, where users can submit hundreds of protein−ligand complexes per day to our GPU-equipped cluster of servers in sdf format and can both visualize predictions and download them to their local machine for further analyses. Preparation steps, such as hydrogen additions are taken as default behavior using ProteinPrepare[59] and rdkit. Our production model was trained using the entire *refined* PDBbind set in order to provide the best performance possible, so users can expect small result improvements when testing on the same compounds reported in this work.

Our model was implemented using Keras and Theano (version 0.9)[60] for all neural-network and tensor-related computations in the Python programming language. All molecular manipulations were handled using rdkit and HTMD. Modeling experiments and benchmarking were carried out using a machine with an Intel(R) Core i7-3930K @ 3.20 GHz CPU, 64 GiB of RAM and a NVIDIA GeForce GTX 1080 graphics card. A GPU is not compulsory to perform training or testing of the network, although it can heavily speed up tensor operations. Up to 300 complexes can be predicted approximately using a modern GPU in less than a minute wall-clock time, and approximately 100 on the same amount of time with a multicore CPU.

DISCUSSION

In this work, we aimed to develop a protein−ligand affinity predictor based on 3D-convolutional neural networks and have demonstrated that they show promising performance in a variety of data sets. However, we believe it is important to acknowledge that no single method examined here outperforms the rest in all data sets tested. Even if trained using the same data set of complexes (as is the case of the CNN model and RF-Score), the fact that their predictions exhibit some degree of orthogonality hints that they exploit different patterns in the training data. This useful characteristic can later be used in meta-modeling in order to provide even more accurate predictions. It is also worth noting that surprisingly molecular weight is sometimes able to match more complex modeling procedures, while FEP-based methods performance is still significantly better than the current generation of scoring functions. These two facts suggest that there is still significant room of improvement for future machine-learning-based methodologies. We believe a more thorough exploration of descriptor representation and network architectural choices could yield significant boosts in performance.

In general, a clear limitation of most structure-based machine-learning predictors is that they require either the co-crystal structure or, in a more realistic scenario, a docked pose, which introduces a certain amount of noise in predictions depending on the chosen docking software, although it has been suggested[61] that this has little impact on performance. Additionally, for these methods to be used in large-scale analyses, hydrogens need to be added to the ligand, and we have seen this fact can significantly affect results. Analyses of the sensibility and further improvements of the discussed methods regarding these issues are subject to further studies.

ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge on the ACS Publications website at DOI: 10.1021/acs.jcim.7b00650.

Additional Figures and Tables with analyses, training and test compounds and model architecture detail.(PDF)

AUTHOR INFORMATION

Corresponding Author

*E-mail: gianni.defabritiis@upf.edu.

ORCID

José Jiménez: 0000-0002-5335-7834

Miha Škalič: 0000-0003-4143-4609

Gerard Martínez-Rosell: 0000-0001-6277-6769

Notes

The authors declare no competing financial interest.

ACKNOWLEDGMENTS

The authors thank Acellera Ltd. for funding. G.D.F. acknowledges support from MINECO (BIO2017-82628-P) and FEDER. This project has received funding from the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 675451 (CompBioMed project). We also thank X. Barril for helping in setting up rDock's tethered docking protocol.

REFERENCES

(1) 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.

(2) Jain, A. N. Surflex: Fully automatic flexible molecular docking using a molecular similarity-based search engine. *J. Med. Chem.* **2003**, *46*, 499−511.

(3) Friesner, R. A.; Banks, J. L.; Murphy, R. B.; Halgren, T. A.; Klicic, J. J.; Mainz, D. T.; Repasky, M. P.; Knoll, E. H.; Shelley, M.; Perry, J. K.; Shaw, D. E.; Francis, P.; Shenkin, P. S. Glide: A New Approach for Rapid, Accurate Docking and Scoring. 1. Method and Assessment of Docking Accuracy. *J. Med. Chem.* **2004**, *47*, 1739−1749.

(4) Trott, O.; Olson, A. J. AutoDock Vina. *J. Comput. Chem.* **2010**, *31*, 445−461.

(5) Shivakumar, D.; Williams, J.; Wu, Y.; Damm, W.; Shelley, J.; Sherman, W. Prediction of Absolute Solvation Free Energies using Molecular Dynamics Free Energy Perturbation and the OPLS Force Field. *J. Chem. Theory Comput.* **2010**, *6*, 1509−1519. PMID: 26615687.

(6) Wang, L.; Wu, Y.; Deng, Y.; Kim, B.; Pierce, L.; Krilov, G.; Lupyan, D.; Robinson, S.; Dahlgren, M. K.; Greenwood, J.; Romero, D. L.; Masse, C.; Knight, J. L.; Steinbrecher, T.; Beuming, T.; Damm, W.; Harder, E.; Sherman, W.; Brewer, M.; Wester, R.; Murcko, M.; Frye, L.; Farid, R.; Lin, T.; Mobley, D. L.; Jorgensen, W. L.; Berne, B. J.; Friesner, R. A.; Abel, R. Accurate and reliable prediction of relative ligand binding potency in prospective drug discovery by way of a modern free-energy calculation protocol and force field. *J. Am. Chem. Soc.* **2015**, *137*, 2695−2703.

(7) Wan, S.; Bhati, A. P.; Skerratt, S.; Omoto, K.; Shanmugasundaram, V.; Bagal, S. K.; Coveney, P. V. Evaluation and Characterization of Trk Kinase Inhibitors for the Treatment of Pain:

Reliable Binding Affinity Predictions from Theory and Computation. *J. Chem. Inf. Model.* **2017**, *57*, 897−909.
(8) Ciordia, M.; Pérez-Benito, L.; Delgado, F.; Trabanco, A. A.; Tresadern, G. Application of Free Energy Perturbation for the Design of BACE1 Inhibitors. *J. Chem. Inf. Model.* **2016**, *56*, 1856−1871.
(9) Keränen, H.; Pérez-Benito, L.; Ciordia, M.; Delgado, F.; Steinbrecher, T. B.; Oehlrich, D.; van Vlijmen, H. W. T.; Trabanco, A. A.; Tresadern, G. Acylguanidine Beta Secretase 1 Inhibitors: A Combined Experimental and Free Energy Perturbation Study. *J. Chem. Theory Comput.* **2017**, *13*, 1439−1453.
(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.
(11) Wang, R.; Lu, Y.; Wang, S. Comparative evaluation of 11 scoring functions for molecular docking. *J. Med. Chem.* **2003**, *46*, 2287−2303.
(12) Sotriffer, C. A.; Sanschagrin, P.; Matter, H.; Klebe, G. SFCscore: Scoring functions for affinity prediction of protein-ligand complexes. *Proteins: Struct., Funct., Genet.* **2008**, *73*, 395−419.
(13) 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.
(14) Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5−32.
(15) Li, G. B.; Yang, L. L.; Wang, W. J.; Li, L. L.; Yang, S. Y. ID-score: A new empirical scoring function based on a comprehensive set of descriptors related to protein-ligand interactions. *J. Chem. Inf. Model.* **2013**, *53*, 592−600.
(16) Cortes, C.; Vapnik, V. Support vector machine. *Mach. Learn.* **1995**, *20*, 273−297.
(17) Li, H.; Leung, K. S.; Wong, M. H.; Ballester, P. J. Improving autodock vina using random forest: The growing accuracy of binding affinity prediction by the effective exploitation of larger data sets. *Mol. Inf.* **2015**, *34*, 115−126.
(18) Zilian, D.; Sotri, C. SFCscoreRF: A Random Forest-Based Scoring Function for Improved Affinity Prediction of Protein-Ligand Complexes. *J. Chem. Inf. Model.* **2013**, *53*, 1923−1933.
(19) LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436−444.
(20) Schmidhuber, J. Deep Learning in Neural Networks: An Overview. *Neural Netw.* **2015**, *61*, 85−117.
(21) Krizhevsky, A.; Sutskever, I.; Hinton, G. E. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process Syst.* **2012**, 1−9.
(22) Goldberg, Y. A primer on neural network models for natural language processing. *J. Artif. Intell. Res.* **2016**, *57*, 345−420.
(23) Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. In *Adv. Neural Inf. Process Syst.*; Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N. D., Weinberger, K. Q., Eds.; Curran Associates, Inc., 2014; pp 2672−2680.
(24) Angermueller, C.; Pärnamaa, T.; Parts, L.; Stegle, O. Deep Learning for Computational Biology. *Mol. Syst. Biol.* **2016**, *12*, 1−16.
(25) Jiménez, J.; Doerr, S.; Martínez-Rosell, G.; Rose, A. S.; De Fabritiis, G. DeepSite: protein-binding site predictor using 3D-convolutional neural networks. *Bioinformatics* **2017**, *33*, 3036−3042.
(26) Mayr, A.; Klambauer, G.; Unterthiner, T.; Hochreiter, S. DeepTox: Toxicity Prediction using Deep Learning. *Front. Environ. Sci.* **2016**, *3*, 1−15.
(27) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein-Ligand Scoring with Convolutional Neural Networks. *J. Chem. Inf. Model.* **2017**, *57*, 942−957.
(28) Cang, Z.; Wei, G. TopologyNet: Topology based deep convolutional and multi-task neural networks for biomolecular property predictions. *PLoS Comput. Biol.* **2017**, *13*, e1005690.
(29) Pereira, J. C.; Caffarena, E. R.; Dos Santos, C. N. Boosting Docking-Based Virtual Screening with Deep Learning. *J. Chem. Inf. Model.* **2016**, *56*, 2495−2506.
(30) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of machine-learning scoring functions in structure-based virtual screening. *Sci. Rep.* **2017**, *7*, 46710.
(31) Lusci, A.; Pollastri, G.; Baldi, P. Deep architectures and deep learning in chemoinformatics: The prediction of aqueous solubility for drug-like molecules. *J. Chem. Inf. Model.* **2013**, *53*, 1563−1575.
(32) Duvenaud, D. K.; Maclaurin, D.; Aguilera-Iparraguirre, J.; Gómez-Bombarelli, R.; Hirzel, T.; Aspuru-Guzik, A.; Adams, R. P. Convolutional Networks on Graphs for Learning Molecular Fingerprints, 2015, arXiv:1509.09292.
(33) Baldi, P. The inner and outer approaches to the design of recursive neural architectures. *Data Min. Knowl. Discov* **2018**, *32*, 218.
(34) Liu, Z.; Su, M.; Han, L.; Liu, J.; Yang, Q.; Li, Y.; Wang, R. Forging the Basis for Developing Protein-Ligand Interaction Scoring Functions. *Acc. Chem. Res.* **2017**, *50*, 302−309. PMID: 28182403.
(35) Pires, D. E.; Ascher, D. B. CSM-lig: a web server for assessing and comparing protein-small molecule affinities. *Nucleic Acids Res.* **2016**, *44*, W557−W561.
(36) Li, H.; Leung, K. S.; Ballester, P. J.; Wong, M. H. Istar: A web platform for large-scale protein-ligand docking. *PLoS One* **2014**, 9.e8567810.1371/journal.pone.0085678
(37) Li, Y.; Liu, Z.; Li, J.; Han, L.; Liu, J.; Zhao, Z.; Wang, R. Comparative Assessment of Scoring Functions on an Updated Benchmark: 1. Compilation of the Test Set. *J. Chem. Inf. Model.* **2014**, *54*, 1700−1716. PMID: 24716849.
(38) Cheng, T.; Li, X.; Li, Y.; Liu, Z.; Wang, R. Comparative Assessment of Scoring Functions on a Diverse Test Set. *J. Chem. Inf. Model.* **2009**, *49*, 1079−1093. PMID: 19358517.
(39) 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.
(40) Kramer, C.; Gedeck, P. Leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets. *J. Chem. Inf. Model.* **2010**, *50*, 1961−1969.
(41) Ballester, P. J.; Mitchell, J. B. O. Comments on "leave-cluster-out cross-validation is appropriate for scoring functions derived from diverse protein data sets": Significance for the validation of scoring functions. *J. Chem. Inf. Model.* **2011**, *51*, 1739−1741.
(42) Dunbar, J. B.; Smith, R. D.; Yang, C. Y.; Ung, P. M. U.; Lexa, K. W.; Khazanov, N. A.; Stuckey, J. A.; Wang, S.; Carlson, H. A. CSAR benchmark exercise of 2010: Selection of the protein-ligand complexes. *J. Chem. Inf. Model.* **2011**, *51*, 2036−2046.
(43) Benson, M. L.; Smith, R. D.; Khazanov, N. A.; Dimcheff, B.; Beaver, J.; Dresslar, P.; Nerothin, J.; Carlson, H. A. Binding MOAD, a high-quality protein-ligand database. *Nucleic Acids Res.* **2008**, *36*, D674.
(44) Dunbar, J. B.; Smith, R. D.; Damm-Ganamet, K. L.; Ahmed, A.; Esposito, E. X.; Delproposto, J.; Chinnaswamy, K.; Kang, Y. N.; Kubish, G.; Gestwicki, J. E.; Stuckey, J. A.; Carlson, H. A. CSAR data set release 2012: Ligands, affinities, complexes, and docking decoys. *J. Chem. Inf. Model.* **2013**, *53*, 1842−1852.
(45) Landrum, G. RDKit: Open-source cheminformatics, 2012. http://www.rdkit.org (accessed January 2018).
(46) Ruiz-Carmona, S.; Alvarez-Garcia, D.; Foloppe, N.; Garmendia-Doval, A. B.; Juhos, S.; Schmidtke, P.; Barril, X.; Hubbard, R. E.; Morley, S. D. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLoS Comput. Biol.* **2014**, *10*, e1003571.
(47) Doerr, S.; Harvey, M. J.; Noé, F.; De Fabritiis, G. HTMD: High-Throughput Molecular Dynamics for Molecular Discovery. *J. Chem. Theory Comput.* **2016**, *12*, 1845−1852.
(48) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition; IEEE CVPR; 2016; pp 770−778.
(49) Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *Int. Conf. Learn. Represent.* **2015**, 1−14.
(50) Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and less than 0.5MB model size, 2015, arXiv:1602.07360.
(51) Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Li, F.-F. *ImageNet: A Large-Scale Hierarchical Image Database*; IEEE CVPR, 2009; pp 248−255.

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296

(52) Kingma, D. P.; Ba, J. L. Adam: A Method for Stochastic Optimization. *Int. Conf. Learn. Represent.* **2015**, 1−15.
(53) Glorot, X.; Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. *AISTATS 13* **2010**, *9*, 249−256.
(54) Wang, R.; Lai, L.; Wang, S. Further development and validation of empirical scoring functions for structure-based binding affinity prediction. *J. Comput.-Aided Mol. Des.* **2002**, *16*, 11−26.
(55) Cao, Y.; Li, L. Improved protein-ligand binding affinity prediction by using a curvature-dependent surface-area model. *Bioinformatics* **2014**, *30*, 1674−1680.
(56) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825−2830.
(57) Das, S.; Krein, M. P.; Breneman, C. M. Binding affinity prediction with property-encoded shape distribution signatures. *J. Chem. Inf. Model.* **2010**, *50*, 298−308.
(58) Li, H.; Leung, K. S.; Wong, M. H.; Ballester, P. J. Low-quality structural and interaction data improves binding affinity prediction via random forest. *Molecules* **2015**, *20*, 10947−10962.
(59) Martínez-Rosell, G.; Giorgino, T.; De Fabritiis, G. PlayMolecule ProteinPrepare: A Web Application for Protein Preparation for Molecular Dynamics Simulations. *J. Chem. Inf. Model.* **2017**, *57*, 1511−1516.
(60) Theano Development Team. Theano: A Python framework for fast computation of mathematical expressions, 2016, arXiv:1605.02688.
(61) Li, H.; Leung, K.-S.; Wong, M.-H.; Ballester, P. J. Correcting the impact of docking pose generation error on binding affinity prediction. *BMC Bioinf.* **2016**, *17*, 308.

DOI: 10.1021/acs.jcim.7b00650
J. Chem. Inf. Model. 2018, 58, 287−296