BigBind: Learning from Nonstructural Data for Structure-Based Virtual Screening

Michael Brocidiacono,[*,†] Paul Francoeur,[‡] Rishal Aggarwal,[‡] Konstantin I. Popov,[¶] David Ryan Koes,[‡] and Alexander Tropsha[†]

†*Eshelman School of Pharmacy, University of North Carolina at Chapel Hill*

‡*Department of Computational and Systems Biology, University of Pittsburgh*

¶*Department of Biochemistry and Biophysics, University of North Carolina at Chapel Hill*

E-mail: mixarcid@unc.edu

Abstract

Recent attempts at utilizing deep learning for structure-based virtual screening have focused on training models to predict binding affinity from protein-ligand complexes with known crystal structures. The PDBbind dataset is the current standard for training such models, but its small size (less than 20K binding affinity measurements) leads to models failing to generalize to new targets, and model performance is typically on par with those trained with only ligand information. The CrossDocked dataset expands binding pose data for protein-ligand complexes but does not introduce new affinity data. ChEMBL, on the other hand, contains a wealth of binding affinity information but contains no information about the binding poses. We introduce BigBind, a dataset that maps ChEMBL activity data to protein targets from CrossDocked. This dataset comprises 851K ligand binding affinities and 3D pocket structures. After augmenting this dataset with an equal number of putative inactives for each target, we train BANANA (BAsic NeurAl Network for binding Affinity) to classify actives from inactives.

The resulting model achieved an AUC of 0.72 on BigBind's test set, while a ligand-only model achieved an AUC of 0.64. Our model achieves competitive performance on the LIT-PCBA benchmark (median EF1% 2.06) while running 16,000 times faster than molecular docking with GNINA. Notably, we achieve a state-of-the-art EF1% of 4.95 when we use BANANA to filter out 90% of the compounds prior to docking with GNINA. We hope that BANANA and future models trained on this dataset will prove useful for prospective virtual screening tasks.

Introduction

Structure-based virtual screening aims to identify compounds that bind to a pocket in a target protein structure. A large library of chemicals is screened against a target of interest, and a program is used to rank compounds according to their predicted ability to bind to the receptor. The top scoring compounds are prioritized for experimental validation. Traditionally, this process is achieved using physics-based docking methods. Monte-Carlo sampling is used to produce an ensemble of poses which are then scored according to a physics-based heuristic.[1–4] These docking methods, however, have limited accuracy, and are often too slow to run on new libraries containing billions[5] of compounds.

Recently, deep learning techniques have achieved great success in domains such as image and natural language processing.[6–8] Given this success, we desire to apply these techniques to identifying hit compounds during structure-based virtual screening campaigns. To this end, many groups have used neural networks to score 3D protein-ligand complexes, either with 3D convolutional networks[9,10] or message passing neural networks (MPNNs).[11–13] The goal is to use the Monte-Carlo sampling of traditional docking to produce a set of poses and then use a neural network to score each pose. This is the approach taken by GNINA.[14]

However, such structure-based approaches that score 3D protein-ligand complexes are limited by the available data. Most approaches are trained with the PDBbind dataset, which uses 3D complexes from the Protein Data Bank (PDB)[15] mapped to known binding

affinities. When using naïve data splits, models often appear to perform well. However, this performance is misleading because of the inherent similarity of complexes in the train and test sets. It has been observed that even a ligand-only K nearest neighbors (KNN) regressor can perform well on the PDBbind refined set,[16] but models do much worse when using clustered splits.[17,18] Recently Francoeur et al.[18] introduced the CrossDocked dataset, which ameliorates several issues with PDBbind. CrossDocked introduces 3D structures of ligands docked to cognate receptors, listing the root-mean-square deviation (RMSD) to the crystal pose along with the ligand's binding affinity (if known). It also clusters the pockets according to 3D structural similarity and uses these clusters for the data splits. The inclusion of more poses enables neural networks to discriminate between correct and incorrect poses, but binding affinity data is still restricted to ligands with a known binding pose. Perhaps due to this data limitation, it has been observed that GNINA[14] (which was trained on CrossDocked) still often relies on ligand-only information when predicting binding affinity.[19]

Training deep learning models on protein-ligand interactions with known crystal structures makes sense in the context of augmenting docking techniques that generate possible ligand poses. However, a fully end-to-end architecture (that takes as input just the compound and the protein target) does not need 3D pose data. Several methods have been proposed that predict binding affinity in an end-to-end fashion, utilizing just the ligand chemical graph representation and the 3D receptor structure[20] or the ligand graph and the receptor amino acid sequence.[21–23] However, these methods often use the PDBbind dataset, with the rare exception of models that use Davis[24] or KIBA[25] datasets, but these are both limited to kinases.

We hypothesized that the performance of deep learning models on PDBbind is limited by the dataset's small size (less than 20K data points). In contrast, areas in which deep learning has proven effective typically have much larger datasets.[7,26] ChEMBL has almost 20M compound activities, though they lack associated crystal structures.[27] So we created BigBind, a dataset that maps binding affinity information from CheMBL to the 3D structures

of protein pockets in CrossDocked. We created data splits based on pocket similarity, so we can test model generalization to new pockets. The resulting dataset contains 851,359 activities spanning 531,560 unique compounds and 1,067 protein pockets.

We then developed a simple graph neural network, BANANA (BAsic NeurAl Network for binding Affinity), to directly predict binding affinity from the pocket and ligand graphs. Our initial results demonstrated the same problems seen in PDBbind – namely, that the model overfit to targets in the training set, and has similar performance with and without receptor information. It seems that simply adding more data doesn't automatically yield better generalization. We hypothesized that the source of this issue is bias in the dataset. Since ChEMBL activities are curated from publications, many molecules were specifically designed to bind to the target they were tested on. Thus it may be possible to guess the relevant target information simply by analyzing the ligand (as our models appear to be doing). This process, however, does not generalize well to new targets, which is why the models overfit.

To combat this, we use Stochastic Negative Addition (SNA), a technique proposed by Cáceres et al.[28]. Since protein-ligand binding is rare, if we choose a random molecule and random target from the dataset, we can assume that the compound is inactive against the target. By adding these putative inactives to our dataset, we alleviate the issue of being able to guess target properties by simply looking at the ligand.

The results are consistent with our hypothesis. When we trained a classification model without SNA, a ligand-only version outperformed the full version on the test set (AUC 0.75 versus 0.64, respectively). When we added SNA, the performance of the two approaches swapped: the full version achieved an AUC of 0.72 while the ligand-only version only achieved an AUC of 0.64. (Note that, since the test sets for the SNA models were also augmented with SNA, the performance of the SNA and non-SNA models on their respective test sets should not be directly compared.)

Encouraged by these results, we tested the model on LIT-PCBA,[29] a difficult benchmark

consisting of experimentally verified active and inactive molecules for a set of 15 targets. When used alone, BANANA achieves competitive performance with GNINA (median EF1% of 2.06 versus GNINA's EF1% of 1.88 for the default ensemble and 2.58 for the dense ensemble). Additionally, we show that using BANANA to quickly filter out 90% of compounds prior to docking with GNINA achieves a state-of-the-art median EF1% of 4.95.

Overall, we demonstrate that a model trained on this dataset can successfully generalize to new targets and shows promise for prospective virtual screening campaigns. We hope that newer, more advanced, models will use BigBind to achieve even greater performance.

Methods

Dataset Creation

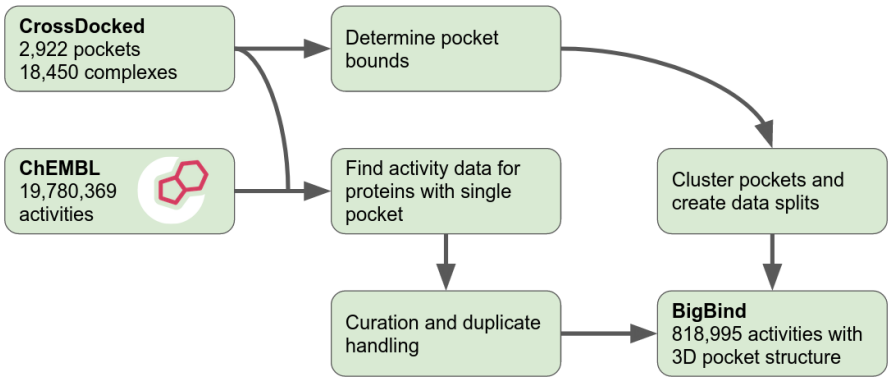

Figure 1: Workflow to create the BigBind dataset.

To generate the BigBind dataset, we first determine the Uniprot[30] accessions of each protein in the CrossDocked dataset using the SIFTs dataset.[31] Since the receptor structures from CrossDocked are clustered into pocket folders from Pocketome,[32] we determined which proteins only contained one pocket. We then queried the ChEMBL 30 database to find all molecules which known binding affinities to those proteins. We assume that everything in ChEMBL which binds to those proteins binds in that known pocket (this assumption is likely incorrect for many compounds, leading to noise in the dataset).

We then filtered the resulting molecules. Following the convention of ZINC,[33] we filtered out any molecule containing atoms other than H, C, N, O, F, S, P, Cl, Br, or I. We also filtered out all mixtures and ensured that each molecule contains at least 5 atoms and has a molecular weight of less than 1,000 amu. Following the example of Fourches et al.[34], we also curated any duplicate activity values by using a KNN model to break ties. We also used RDKit[35] to generate a 3D structure for each molecule and optimize the resulting structure using UFF.[36] If RDKit failed to generate an optimized structure for a compound, it was removed. Additional details about the data curation process are in the supporting information.

Using the aligned crystal structures from CrossDocked, we also determined the extent of each protein binding pocket. For each pocket, we superimposed all ligand crystal structures that bind to that pocket, and, for each receptor crystal structure, we chose all residues within 5 Å of any ligand atom. We save a separate pocket PDB file for each receptor. We also define the pocket 3D bounding box to be the minimum box that contains all crystallized ligands with 4 Å of padding on all sites. We filter out all pocket files with less than 5 residues or with bounding boxes of more than 42 Å on any side. When training the models on this dataset, we choose a random pocket PDB file from the relevant pocket folder.

Finally, we generated the data splits. Following CrossDocked, we used ProBis[37] to generate similarity scores between each pocket (details in supplemental). We then clustered pockets together if their z score was greater than or equal to 3.5. We then generated 80:10:10 train:test:validation splits, ensuring that all pockets in the same cluster were also in the same split. To ensure we can use performance on LIT-PCBA as an evaluation metric, we also ensured that all pockets in the same cluster as any LIT-PCBA target were also in the test set.

Stochastic Negative Addition

To utilize SNA, we first turned the problem into a classification problem. For every data point in the original dataset, we labeled the compound active if its binding affinity was less then 10 μM (pChEMBL value greater than 5). Then, for each target, we add an equal amount of randomly selected compounds that we label as inactive. When selecting these presumed inactives, we ensure that the compounds are not known to bind to any target whose pocket has ProBis z-score similarity of greater than 3 to the target in question. This cutoff (smaller than the cutoff used to create data splits) was chosen because it clusters the kinases in the dataset together. Thus we ensure that any compound that binds to one kinase will not be labeled as inactive against a different kinase.

Model Architecture and Training

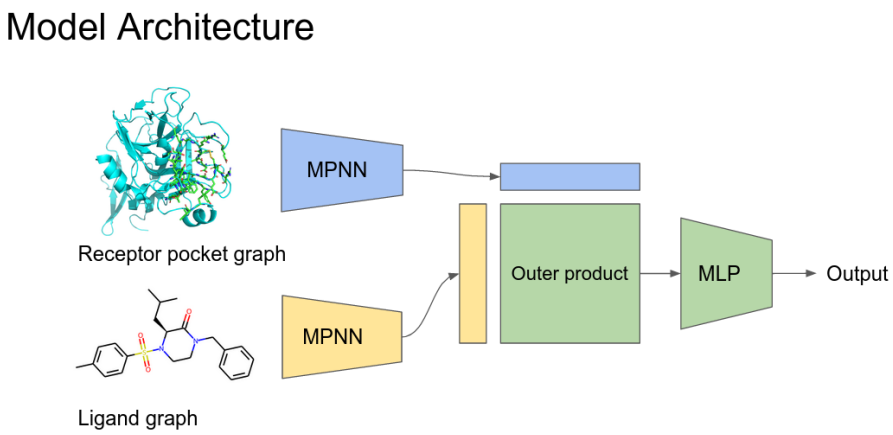

Figure 2: BANANA architecture.

The architecture of BANANA is shown in Figure 2. For the model input, we prepare graphs for both the receptor binding pocket and the ligand. Following several others,[38–40] the nodes of the receptor graph are the residues, labeled with the amino acid. An edge exists between two nodes in the receptor graph if their α-carbons are within 20 Å of each other. The scalar distance between them is used as edge data. For the ligand graph, we simply use the molecule graph. The heavy atoms are nodes and are labeled with the element,

formal charge, hybridization, number of bonded hydrogens, and whether or not the atom is aromatic. The bonds are edges and are labeled with the bond order.

Two separate MPNNs are used to create output vectors v_L and v_R for the ligand and receptor, respectively. Similarly to Krasoulis et al.[20], we then compute the outer product $v_L v_R^\intercal$. After flattening, we use a multi-layer perceptron (MLP) to compute the scalar output. For the regression task of predicting binding affinity we use this output directly, and for the classification task of predicting whether or not the ligand is active against the pocket, we use a sigmoid to give us the output probability.

For all experiments, we trained the model with and without receptor information. When we wanted to remove receptor information, we kept the model architecture the same but only gave it the first receptor pocket in the dataset.

When training, we used a mean squared error (MSE) loss for the regression task and a binary cross-entropy (BCE) loss for the classification task. We used the AdamW optimizer[41] with a learning rate of 10^{-5} and a batch size of 16. We trained the classification models for 5 epochs and the regression models for 50 epochs. The remaining hyper-parameters and training details can be found in the supporting information.

Model Evaluation

To test the classification models, we looked at the area under the curve (AUC) of the receiver operating characteristic (ROC) on the BigBind test set. This gives an overall view of how well the model classifies actives from inactives. However, for practical virtual screening applications, one cares more about whether or not the model can select actives from a large set of mostly inactive molecules. To test this, we evaluated the final ligand-and-receptor classification model's top 1% enrichment factor (EF1%) and normalized enrichment factor (NEF1%) on the targets from LIT-PCBA. These results are then compared with GNINA, as reported by Sunseri and Koes[19].

To benchmark the speed of BANANA, we evaluated the model on the complexes in the

PDBbind 2016 core set on a laptop with an NVIDIA GeForce RTX 2060 Mobile GPU. This speed was then compared to the speed of GNINA (default ensemble) as reported by McNutt et al.[14].

Since BANANA is significantly faster than traditional docking tools, we wondered whether it could be useful for filtering out compounds in a virtual screen prior to conventional docking. To answer this, we used BANANA to filter out 90% of the compounds and used GNINA (with both the default and dense ensembles) to rerank the remaining 10%. We tested this on LIT-PCBA and report the resulting enrichment factors.

Results

The Importance of Stochastic Negatives

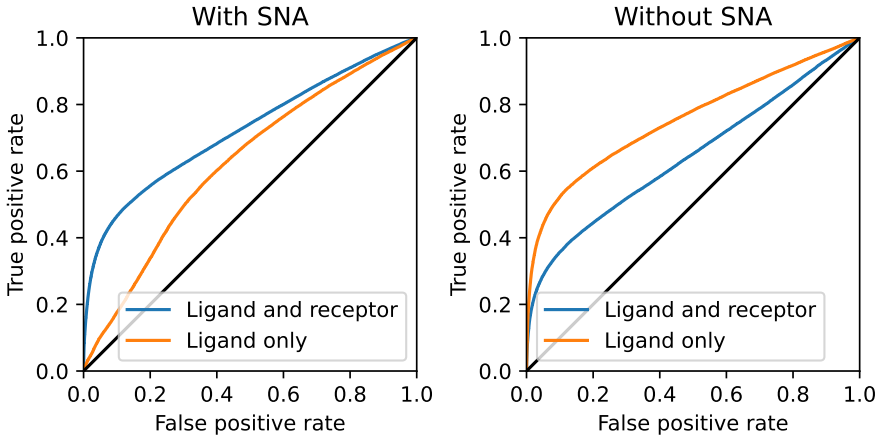

Figure 3: ROC curves for models trained with and without SNA. Left: Test ROC curves for the ligand-and-receptor model (AUC 0.72) and ligand-only model (AUC 0.64) when utilizing SNA. Right: Test ROC curves for the ligand-and-receptor model (AUC 0.64) and ligand-only model (AUC 0.75) without SNA. Note that the AUC values of the SNA models should not be compared to the non-SNA models because the test sets are different.

As can be seen in Figure 3, when BANANA is trained without SNA, the ligand-only model outperforms the ligand-and-receptor model. While both achieve decent performance

on the non-SNA test set, the fact that receptor information reduces performance implies that this performance is entirely due to biases within the dataset. Thus, these models are unsuitable for prospective virtual screening tasks. On the other hand, when the models are trained with SNA, the ligand-and-receptor model significantly outperforms the ligand-only model. This supports our hypothesis that SNA provides a way to force the model to learn information about the ligand-receptor interaction rather than simply exploiting the biased nature of the dataset.

Model Performance on LIT-PCBA

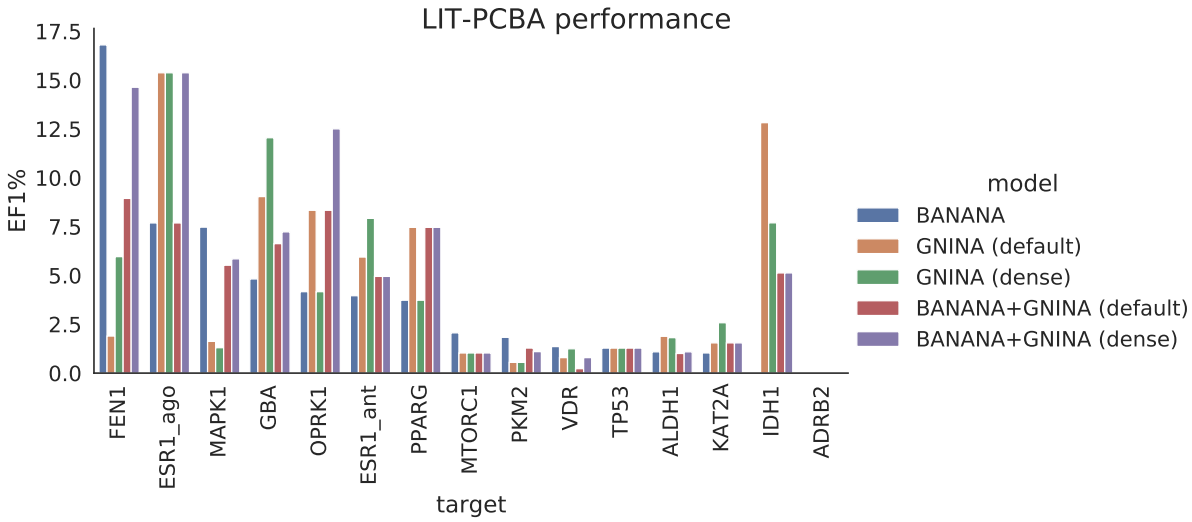

Figure 4: Performance of all the models on LIT-PCBA targets.

As Table 1 demonstrates, our model achieves performance comparable to GNINA with the default ensemble (EF1% 2.06 versus 1.88), though still falls behind GNINA with the dense model (EF1% 2.58). The performance of BANANA seems largely uncorrelated with the performance of GNINA; Figure 4 shows that GNINA (either default or dense) can perform much better or much worse than BANANA depending on the specific target. Perhaps because of this, the combination models show marked improvement; BANANA+GNINA (default) and BANANA+GNINA (dense) both achieve a state-of-the-art median EF1% of 4.95. It appears that BANANA is able to quickly filter out inactive compounds that GNINA would

otherwise rank highly.

Overall, BANANA achieves enrichment factors that are competitive with GNINA. The most promising aspect of the model, however, is the speed. We find that BANANA takes an average of 1.7 ms to evaluate a single protein-ligand complex from the PDBbind 2016 core set. Since GNINA (default) takes an average of 27 s on the same set, our model shows a speedup factor of 16,000. This enormous speedup means that the combination models BANANA+GNINA are able to run 10 times faster than GNINA alone, while still achieving higher performance.

Table 1: Median EF1%, NEF1%, and AUC values for all the models on LIT-PCBA. No AUC values can be computed for the BANANA+GNINA models because they do not explicitly score each compound.

Model	**EF1%**	**NEF1%**	**AUC**
BANANA	2.06	0.04	0.6
GNINA (default)	1.88	0.02	0.61
GNINA (dense)	2.58	0.04	0.62
BANANA+GNINA (default)	4.95	0.05	
BANANA+GNINA (dense)	4.95	0.05	

Discussion

In recent years there have been many advances in deep learning model architectures for analyzing molecules.[42–46] A deep learning model, however, is only as good as the dataset it is trained on. Previous attempts to utilize machine learning for protein-ligand binding affinity prediction often trained on PDBbind, but the small size and bias inherent to the dataset have hindered their utility. Thus, we developed BigBind, a dataset of 851K protein-ligand binding affinities along with the 3D structure of the putative receptor binding pocket. We also added putative inactives to debias the dataset and showed that a model trained on the debiased dataset is forced to learn information about protein-ligand interactions and can generalize to new targets. We then showed that our model, when used alone, performs comparably to docking with GNINA on the LIT-PCBA benchmark while running 16,000

times faster. Additionally, when BANANA is used to filter out 90% of compounds before re-scoring with GNINA, we achieve a state-of-the-art median EF1% of 4.95. Thus BANANA demonstrates immediate utility for virtual screening. Notably, since the model takes only 1.7 ms to evaluate a single ligand, it shows promise for screening massive sets such as Enamine's REAL database.[5]

The model described in this paper is relatively simple, and we plan on exploring more advanced architectures in the future. We are especially interested in exploring models that hypothesize a 3D pose for the ligand in order to explain the activities. Perhaps having more inductive biases about 3D space will improve model performance. Additionally, we hope to expand BigBind in the future. PubChem,[47] for instance, has data from high-throughput screens not seen in ChEMBL. This data is noisy, but it is possible that adding it will improve model performance.

We hope that the scientific community will use this dataset to train new models. The code for creating the dataset can be found at https://github.com/molecularmodelinglab/bigbind, and the full dataset can be downloaded at https://storage.googleapis.com/bigbind/BigBindV1.tar.bz2. The code for training and running BANANA is available at https://github.com/molecularmodelinglab/banana.

Acknowledgement

The authors thank Henry Dieckhaus, James Wellnitz, Josh Hochuli, Kathryn Kirchoff, and Travis Maxfield for support and insightful discussions. We also thank Jack Lynch for his input, support, and GPUs. Studies reported in this paper were supported by the NIH grant R01AI163514.

References

(1) Eberhardt, J.; Santos-Martins, D.; Tillack, A. F.; Forli, S. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling* **2021**, *61*, 3891–3898, Publisher: American Chemical Society.

(2) 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. *Journal of Medicinal Chemistry* **2004**, *47*, 1739–1749, Publisher: American Chemical Society.

(3) Kuntz, I. D.; Blaney, J. M.; Oatley, S. J.; Langridge, R.; Ferrin, T. E. A geometric approach to macromolecule-ligand interactions. *Journal of Molecular Biology* **1982**, *161*, 269–288.

(4) McGann, M. FRED Pose Prediction and Virtual Screening Accuracy. *Journal of Chemical Information and Modeling* **2011**, *51*, 578–596, Publisher: American Chemical Society.

(5) Grygorenko, O. O.; Radchenko, D. S.; Dziuba, I.; Chuprina, A.; Gubina, K. E.; Moroz, Y. S. Generating Multibillion Chemical Space of Readily Accessible Screening Compounds. *iScience* **2020**, *23*, 101681.

(6) Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; Krueger, G.; Sutskever, I. Learning Transferable Visual Models From Natural Language Supervision. 2021; `http://arxiv.org/abs/2103.00020`, arXiv:2103.00020 [cs].

(7) Brown, T. B. et al. Language Models are Few-Shot Learners. 2020; `http://arxiv.org/abs/2005.14165`, arXiv:2005.14165 [cs].

(8) He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. 2015; http://arxiv.org/abs/1512.03385, arXiv:1512.03385 [cs].

(9) Ragoza, M.; Hochuli, J.; Idrobo, E.; Sunseri, J.; Koes, D. R. Protein–Ligand Scoring with Convolutional Neural Networks. *Journal of Chemical Information and Modeling* **2017**, *57*, 942–957.

(10) Wallach, I.; Dzamba, M.; Heifets, A. AtomNet: A Deep Convolutional Neural Network for Bioactivity Prediction in Structure-based Drug Discovery. 2015; http://arxiv.org/abs/1510.02855, arXiv:1510.02855 [cs, q-bio, stat].

(11) Zhang, S.; Jin, Y.; Liu, T.; Wang, Q.; Zhang, Z.; Zhao, S.; Shan, B. SS-GNN: A Simple-Structured Graph Neural Network for Affinity Prediction. 2022; http://arxiv.org/abs/2206.07015, arXiv:2206.07015 [cs, q-bio].

(12) Gomes, J.; Ramsundar, B.; Feinberg, E. N.; Pande, V. S. Atomic Convolutional Networks for Predicting Protein-Ligand Binding Affinity. 2017; http://arxiv.org/abs/1703.10603, arXiv:1703.10603 [physics, stat].

(13) Feinberg, E. N.; Sur, D.; Wu, Z.; Husic, B. E.; Mai, H.; Li, Y.; Sun, S.; Yang, J.; Ramsundar, B.; Pande, V. S. PotentialNet for Molecular Property Prediction. *ACS Central Science* **2018**, *4*, 1520–1530, Publisher: American Chemical Society.

(14) McNutt, A. T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D. R. GNINA 1.0: molecular docking with deep learning. *Journal of Cheminformatics* **2021**, *13*, 43.

(15) Zardecki, C.; Dutta, S.; Goodsell, D. S.; Voigt, M.; Burley, S. K. RCSB Protein Data Bank: A Resource for Chemical, Biochemical, and Structural Explorations of Large and Small Biomolecules. *Journal of Chemical Education* **2016**, *93*, 569–575, Publisher: American Chemical Society.

(16) Volkov, M.; Turk, J.-A.; Drizard, N.; Martin, N.; Hoffmann, B.; Gaston-Mathé, Y.; Rognan, D. On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks. *Journal of Medicinal Chemistry* **2022**, *65*, 7946–7958.

(17) Yang, J.; Shen, C.; Huang, N. Predicting or Pretending: Artificial Intelligence for Protein-Ligand Interactions Lack of Sufficiently Large and Unbiased Datasets. *Frontiers in Pharmacology* **2020**, *11*, 69.

(18) Francoeur, P. G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R. B.; Snyder, I.; Koes, D. R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *Journal of Chemical Information and Modeling* **2020**, *60*, 4200–4215.

(19) Sunseri, J.; Koes, D. R. Virtual Screening with Gnina 1.0. *Molecules (Basel, Switzerland)* **2021**, *26*, 7369.

(20) Krasoulis, A.; Antonopoulos, N.; Pitsikalis, V.; Theodorakis, S. DENVIS: Scalable and High-Throughput Virtual Screening Using Graph Neural Networks with Atomic and Surface Protein Pocket Features. *Journal of Chemical Information and Modeling* **2022**, *62*, 4642–4659, Publisher: American Chemical Society.

(21) Öztürk, H.; Özgür, A.; Ozkirimli, E. DeepDTA: deep drug–target binding affinity prediction. *Bioinformatics* **2018**, *34*, i821–i829.

(22) Nguyen, T.; Le, H.; Quinn, T. P.; Nguyen, T.; Le, T. D.; Venkatesh, S. GraphDTA: predicting drug–target binding affinity with graph neural networks. *Bioinformatics* **2021**, *37*, 1140–1147.

(23) Wang, J.; Dokholyan, N. V. Yuel: Improving the Generalizability of Structure-Free Compound–Protein Interaction Prediction. *Journal of Chemical Information and Modeling* **2022**, *62*, 463–471.

(24) Davis, M. I.; Hunt, J. P.; Herrgard, S.; Ciceri, P.; Wodicka, L. M.; Pallares, G.; Hocker, M.; Treiber, D. K.; Zarrinkar, P. P. Comprehensive analysis of kinase inhibitor selectivity. *Nature Biotechnology* **2011**, *29*, 1046–1051, Number: 11 Publisher: Nature Publishing Group.

(25) Tang, J.; Szwajda, A.; Shakyawar, S.; Xu, T.; Hintsanen, P.; Wennerberg, K.; Aittokallio, T. Making Sense of Large-Scale Kinase Inhibitor Bioactivity Data Sets: A Comparative and Integrative Analysis. *Journal of Chemical Information and Modeling* **2014**, *54*, 735–743, Publisher: American Chemical Society.

(26) Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. 2009 IEEE Conference on Computer Vision and Pattern Recognition. 2009; pp 248–255, ISSN: 1063-6919.

(27) Gaulton, A.; Bellis, L. J.; Bento, A. P.; Chambers, J.; Davies, M.; Hersey, A.; Light, Y.; McGlinchey, S.; Michalovich, D.; Al-Lazikani, B.; Overington, J. P. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Research* **2012**, *40*, D1100–D1107.

(28) Cáceres, E. L.; Mew, N. C.; Keiser, M. J. Adding Stochastic Negative Examples into Machine Learning Improves Molecular Bioactivity Prediction. *Journal of Chemical Information and Modeling* **2020**, Publisher: American Chemical Society.

(29) Tran-Nguyen, V.-K.; Jacquemard, C.; Rognan, D. LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening. *Journal of Chemical Information and Modeling* **2020**, *60*, 4263–4273.

(30) Apweiler, R.; Bairoch, A.; Wu, C. H.; Barker, W. C.; Boeckmann, B.; Ferro, S.; Gasteiger, E.; Huang, H.; Lopez, R.; Magrane, M.; Martin, M. J.; Natale, D. A.; O'Donovan, C.; Redaschi, N.; Yeh, L.-S. L. UniProt: the Universal Protein knowledgebase. *Nucleic Acids Research* **2004**, *32*, D115–D119.

(31) Dana, J. M.; Gutmanas, A.; Tyagi, N.; Qi, G.; O'Donovan, C.; Martin, M.; Velankar, S. SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Research* **2019**, *47*, D482–D489.

(32) Kufareva, I.; Ilatovskiy, A. V.; Abagyan, R. Pocketome: an encyclopedia of small-molecule binding sites in 4D. *Nucleic Acids Research* **2012**, *40*, D535–D540.

(33) Irwin, J. J.; Shoichet, B. K. ZINC – A Free Database of Commercially Available Compounds for Virtual Screening. *Journal of chemical information and modeling* **2005**, *45*, 177–182.

(34) Fourches, D.; Muratov, E.; Tropsha, A. Trust, but Verify II: A Practical Guide to Chemogenomics Data Curation. *Journal of Chemical Information and Modeling* **2016**, *56*, 1243–1252, Publisher: American Chemical Society.

(35) RDKit: Open-source cheminformatics. `http://www.rdkit.org/`.

(36) Rappe, A. K.; Casewit, C. J.; Colwell, K. S.; III, W. A. G.; Skiff, W. M. UFF, a full periodic table force field for molecular mechanics and molecular dynamics simulations. 2002; `https://pubs.acs.org/doi/pdf/10.1021/ja00051a040`, Archive Location: world Publisher: American Chemical Society.

(37) Konc, J.; Janežič, D. ProBiS: a web server for detection of structurally similar protein binding sites. *Nucleic Acids Research* **2010**, *38*, W436–W440.

(38) Jing, B.; Eismann, S.; Suriana, P.; Townshend, R. J. L.; Dror, R. Learning from Protein Structure with Geometric Vector Perceptrons. 2021; `http://arxiv.org/abs/2009.01411`, arXiv:2009.01411 [cs, q-bio, stat].

(39) Gligorijević, V.; Renfrew, P. D.; Kosciolek, T.; Leman, J. K.; Berenberg, D.; Vatanen, T.; Chandler, C.; Taylor, B. C.; Fisk, I. M.; Vlamakis, H.; Xavier, R. J.; Knight, R.;

Cho, K.; Bonneau, R. Structure-based protein function prediction using graph convolutional networks. *Nature Communications* **2021**, *12*, 3168, Number: 1 Publisher: Nature Publishing Group.

(40) Stärk, H.; Ganea, O.-E.; Pattanaik, L.; Barzilay, R.; Jaakkola, T. EquiBind: Geometric Deep Learning for Drug Binding Structure Prediction. 2022; `http://arxiv.org/abs/2202.05146`, arXiv:2202.05146 [cs, q-bio].

(41) Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. 2019; `http://arxiv.org/abs/1711.05101`, arXiv:1711.05101 [cs, math].

(42) Schütt, K. T.; Kindermans, P.-J.; Sauceda, H. E.; Chmiela, S.; Tkatchenko, A.; Müller, K.-R. SchNet: A continuous-filter convolutional neural network for modeling quantum interactions. 2017; `http://arxiv.org/abs/1706.08566`, arXiv:1706.08566 [physics, stat].

(43) Gómez-Bombarelli, R.; Wei, J. N.; Duvenaud, D.; Hernández-Lobato, J. M.; Sánchez-Lengeling, B.; Sheberla, D.; Aguilera-Iparraguirre, J.; Hirzel, T. D.; Adams, R. P.; Aspuru-Guzik, A. Automatic Chemical Design Using a Data-Driven Continuous Representation of Molecules. *ACS Central Science* **2018**, *4*, 268–276, Publisher: American Chemical Society.

(44) Yang, K.; Swanson, K.; Jin, W.; Coley, C.; Eiden, P.; Gao, H.; Guzman-Perez, A.; Hopper, T.; Kelley, B.; Mathea, M.; Palmer, A.; Settels, V.; Jaakkola, T.; Jensen, K.; Barzilay, R. Analyzing Learned Molecular Representations for Property Prediction. *Journal of Chemical Information and Modeling* **2019**, *59*, 3370–3388, Publisher: American Chemical Society.

(45) Doerr, S.; Majewsk, M.; Pérez, A.; Krämer, A.; Clementi, C.; Noe, F.; Giorgino, T.; De Fabritiis, G. TorchMD: A deep learning framework for molecular simulations. *Jour-*

nal of Chemical Theory and Computation **2021**, *17*, 2355–2363, arXiv:2012.12106 [physics].

(46) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. 2022; `http://arxiv.org/abs/2210.01776`, arXiv:2210.01776 [physics, q-bio].

(47) Kim, S.; Chen, J.; Cheng, T.; Gindulyte, A.; He, J.; He, S.; Li, Q.; Shoemaker, B. A.; Thiessen, P. A.; Yu, B.; Zaslavsky, L.; Zhang, J.; Bolton, E. E. PubChem in 2021: new data content and improved web interfaces. *Nucleic Acids Research* **2021**, *49*, D1388–D1395.