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, Chapel Hill, NC 27599*

‡*Department of Computational and Systems Biology, University of Pittsburgh, Pittsburgh, PA 15260*

E-mail: mixarcid@unc.edu

Abstract

Deep learning methods that predict protein-ligand binding have recently been used for structure-based virtual screening. Many such models have been trained using protein-ligand complexes with known crystal structures and activities from the PDBBind dataset. However, because PDBbind only includes 20K complexes, models typically fail to generalize to new targets, and model performance is on par with models trained with only ligand information. Conversely, the ChEMBL database contains a wealth of chemical activity information but includes no information about binding poses. We introduce BigBind, a dataset that maps ChEMBL activity data to proteins from the CrossDocked dataset. BigBind comprises 583K ligand activities and includes 3D structures of the protein binding pockets. Additionally, we augmented the data by adding an equal number of putative inactives for each target. Using this data, we developed Banana (BAsic NeurAl Network for binding Affinity), a neural network-based model to classify active from inactive compounds, defined by a 10 μM cutoff.

Our model achieved an AUC of 0.72 on BigBind's test set, while a ligand-only model achieved an AUC of 0.59. Furthermore, BANANA achieved competitive performance on the LIT-PCBA benchmark (median EF1% 1.81) while running 16,000 times faster than molecular docking with GNINA. We suggest that BANANA, as well as other models trained on this dataset, will significantly improve the outcomes of prospective virtual screening tasks.

Introduction

Structure-based virtual screening aims to identify compounds that bind to a pocket in a target protein structure by ranking compounds from a large chemical library according to predicted binding scores. The top-scoring compounds are prioritized for experimental validation. Traditional algorithms generate and score possible binding poses for each compound using molecular docking and physics-inspired heuristics.[1–4] These docking methods, however, have limited accuracy. Moreover, their slow speed presents challenges for screening new libraries containing billions[5] of compounds.[6]

Recently, deep learning techniques have been introduced to the field of structure-based virtual screening to accelerate both pose generation and scoring of protein-ligand complexes. To this end, many groups have used neural networks to score 3D protein-ligand complexes. For instance, several groups have used 3D convolutional neural networks (CNNs) on voxel grids defined by protein-ligand complexes.[7,8] Notably, this is the approach taken by GNINA.[9] Alternatively, graph neural network architectures such as message-passing neural networks (MPNNs) on the 3D interaction graph of the complex have been proposed.[10–12]

Regardless of the architecture, the performance of all deep learning models that score 3D protein-ligand complexes is limited by the available data. For instance, many models have been trained with the PDBbind dataset, which uses 3D protein-ligand complexes from the Protein Data Bank (PDB)[13] mapped to known activities. This dataset incorporates ca. 20K protein-ligand complexes, a relatively small amount of data compared to well-known

and successful cases of using deep learning in domains such as image recognition and natural language processing.[14,15]

Because of its relatively small size, the accuracy of models trained on the PDBbind dataset has been limited. While many papers report promising results,[10–12] these results are often inflated due to data leakage. This problem arises when using data splits that have the same protein in the train and test sets. The high apparent performance of such models 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] whereas models perform much worse when using clustered splits.[17,18]

Recently Francoeur et al.[18] introduced the CrossDocked dataset, which ameliorates several issues with PDBbind. CrossDocked clusters experimental protein-ligand complexes into conserved pockets, following Pocketome.[19] It includes complexes with no known activity data and augments the dataset with poses of ligands cross-docked to other receptors with the conserved binding site. It also clusters the pockets according to 3D structural similarity and uses these clusters for the data splits. The substantial expansion of the dataset with these docked poses improved the performance of neural networks that select realistic docked poses. Activity data, however, is still restricted to ligands with a known binding pose.

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, such explicit knowledge about ligand pose in the binding pocket may be unnecessary for predicting ligand binding scores. Indeed, several methods have been proposed that predict activity given just the ligand chemical graph representation and the 3D receptor structure[20] or the ligand graph and the receptor amino acid sequence.[21–23] Additionally, it is possible to train models that rely solely on docked poses, as is done by Liu et al.[24].

As mentioned above, deep learning has proven effective when using much more data than

the 20K activity data points available in PDBbind or CrossDocked. Thus, we hypothesized that an expanded dataset with orders of magnitude more binding data would result in more accurate models for predicting binders to novel proteins. Herein, we describe the development of BigBind, a dataset that maps molecular activities from ChEMBL[25] to proteins in CrossDocked. The current version of BigBind contains 582,957 activities comprising 399,090 unique compounds and 1,107 protein pockets. Additionally, we created rigorous data splits based on pocket similarity so we can test model generalization to new pockets. We emphasize that, since this dataset does not contain explicit knowledge of ligand binding poses, models must use separate ligand and receptor graphs rather than an interaction graph. They may, however, utilize interaction graphs resulting from docked poses.

Other datasets exist that map protein 3D structure (or sequence) to ligand activity without crystal poses, but none have the scope of BigBind. DAVIS,[26] KIBA,[27] and KinCo,[24] for instance, are limited to kinases. Benchmarking datasets such as DUD-E,[28] DEKOIS,[29] and LIT-PCBA[30] also contain activity data without crystal poses, but these are designed to benchmark rather than train SBVS models. As such, they do not contain the same breadth of data as BigBind, and the data is highly imbalanced toward inactive molecules. Additionally, it is inappropriate to use DUD-E as a training set due to subtle biases that models can exploit.[31,32]

We then developed a simple graph neural network, Banana (BAsic NeurAl Network for binding Affinity), to directly predict activity from the pocket and ligand graphs. In order to address overfitting in our initial regression model, we used Stochastic Negative Addition (SNA)[33] to augment the dataset with putative inactives. We then trained a classification model on the augmented dataset, which achieved an AUC of 0.72 on the BigBind test set. In contrast, a ligand-only version of the model only achieved an AUC of 0.59, and a KNN baseline achieved an AUC of 0.55.

Encouraged by these results, we tested the model on LIT-PCBA,[30] 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 1.81 versus GNINA's EF1% of 1.88 for the default ensemble and 2.58 for the dense ensemble).

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

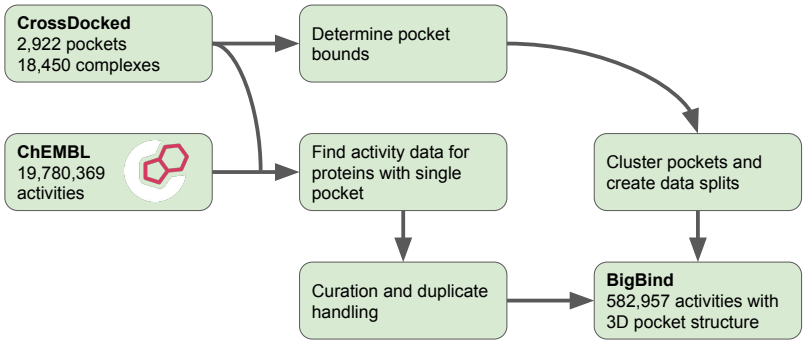

Figure 1: Workflow to create the BigBind dataset.

To generate the BigBind dataset, we first determined the Uniprot[34] accession numbers for each protein in the CrossDocked dataset using the SIFTS dataset.[35] The receptor structures from CrossDocked are clustered into conserved binding sites from Pocketome,;[19] we filtered out all proteins with multiple such binding sites. We then queried the ChEMBL 33 database to find all molecules with known activities for those proteins. We only used assays with `target_type` of `"PROTEIN"` to ensure each measurement was target-specific rather than cell-based. We assumed that all compounds in ChEMBL that are annotated as active against those proteins bind in that known pocket (this assumption may be incorrect for

some compounds, leading to noise in the dataset). We also note that we used the pChEMBL value as an activity score for each protein-ligand pair; this adds additional noise because it aggregates different types of experimental information (such as K_i, K_d, and IC50) into a single value. To reduce this noise in the dataset, we additionally removed data from primary high-throughput screening (HTS) assays (`standard_type` equal to `"Potency"`).

We then curated the resulting molecules. Following the convention of ZINC,[36] we filtered out any molecule containing elements other than H, C, N, O, F, S, P, Cl, Br, or I. We also filtered out all compounds with disconnected chemical graphs (e.g. salts) and ensured that each molecule contained at least 5 atoms and had a molecular weight of less than 1,000 amu. If a protein-ligand pair had multiple activities in ChEMBL, we recorded the median value. We also used RDKit[37] to generate a 3D structure for each molecule and optimized the resulting structure using UFF.[38] If RDKit failed to generate an optimized structure for a compound, it was removed. We note that the UFF structures were created so we can test docking methods on the dataset; the method described in this paper only uses 2D ligand 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 saved a separate pocket PDB file for each receptor. We also defined the pocket 3D bounding box to be the minimum box that contains all crystallized ligands with 4 Å of padding on all sites. We filtered out all pockets with less than 5 residues or with bounding boxes of more than 42 Å on any side. We additionally used PDBFixer[39] to add missing atoms to each PDB file. When training the models on this dataset, we chose a random pocket from the relevant pocket folder.

Computing the Data Splits

We split the dataset into train, validation, and test sets according to both protein pocket and ligand similarity scores. To compute pocket similarity, we used a pocket-level version of the TM-align algorithm[40] (details in the Supporting Information). Since a pocket may have multiple 3D structures, we defined the similarity between two pockets as the maximum pocket-level TM score between any two of their structures. The ligand similarity was defined as the Tanimoto similarity of two ligands' 2048-bit Morgan fingerprints with radius 3. Two pockets were placed in the same cluster if their TM scores were greater than 0.89. The TM cutoff was lowered to 0.82 if the pockets were known to bind to any ligand with Tanimoto similarity greater than 0.7. The rationale for choosing these two cutoff values is laid out in more detail in the Supporting Information. We split the dataset according to these pocket clusters. 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 activity was less than 10 μM (pChEMBL value greater than 5). Then, for each target, we added an equal amount of randomly selected compounds that we labeled as inactive. When selecting these presumed inactive, we ensured that the compounds were not known to bind to any target in the same pocket similarity cluster as the current target in question. In addition to encouraging models to pay attention to both ligand and protein features, SNA also serves to balance the dataset. Before applying SNA, 84% of activities in the dataset are considered active; after applying SNA, 41% of the dataset is active.

KNN Baseline

To score a protein-ligand interaction using a KNN classifier, we must first define a single similarity measure over protein and ligand pairs. If L is the ligand Tanimoto similarity and R is the protein pocket TM score between two datapoints, we want to know the optimal linear combination of these scores to compute the total similarity. To do so, we estimate the joint probability distribution $P(L, R)$ for all pairs of datapoints in BigBind. We then fit a linear model to predict $\log \frac{P(L,R)}{P(L)P(R)}$. This ratio measures how likely two ligand and protein similarities are to co-occur relative to what would be expected from random co-occurrence. We use the coefficients on L and R to define the global similarity measure $S = 0.18L + 3.57R$. To run the KNN model (with K=1) on a new protein-ligand pair, we simply find the molecule with the closest S in the BigBind training set (without SNA) and return S as the activity score. We note that, unlike the KNN used by Volkov et al.[16], this KNN incorporates both ligand and receptor information and is thus a more robust baseline.

Model Architecture and Training

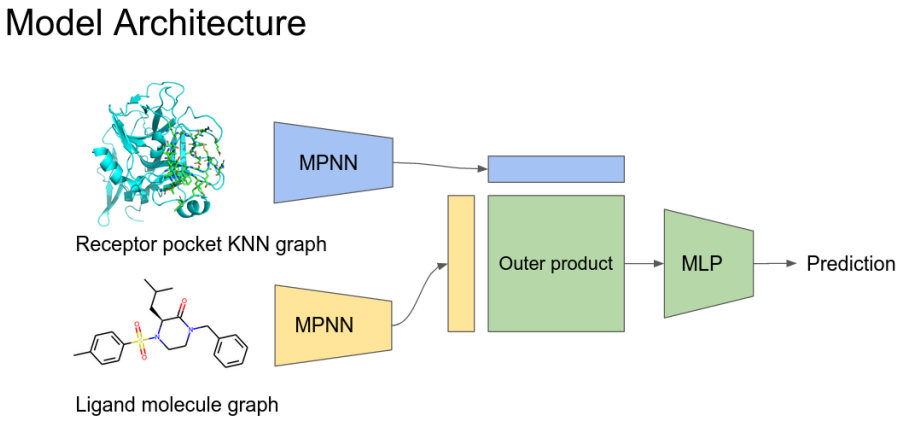

Figure 2: BANANA architecture.

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 previous studies,[41–43] the nodes of the receptor graph are the residues, labeled with the amino acid name. 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 nodes are heavy atoms and are labeled with the element, formal charge, hybridization, number of bonded hydrogens, and whether or not the atom is aromatic. The edges are bonds 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 activity 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. To remove receptor information, we kept the model architecture the same but only gave it the first receptor pocket in the dataset. We note that this results in a ligand-only architecture that is unnecessarily complex, but we nonetheless wanted to ensure the architecture was the same as the ligand-and receptor model to provide a robust control.

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[44] 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 on LIT-PCBA, a challenging benchmark composed of data from high-throughput screens. LIT-PCBA includes lists of experimentally verified active and inactive molecules for all 13 targets, each of which has several 3D structures cocrystallized to different ligands. We use the ligands in these cocrystal structures to define the binding pocket in the same way we created all the pockets in the BigBind dataset. We emphasize that, since all the targets from LIT-PCBA were placed in the BigBind test set, each target is entirely new to Banana. We measured the model's top 1% enrichment factor (EF1%) and normalized enrichment factor (NEF1%) on each target in LIT-PCBA. The EF1% is the ratio of actives in the top 1% of ranked compounds divided by the ratio of actives in the whole set. The NEF% is the EF1% divided by the maximum achievable EF1% for the target, thus normalizing the value between zero and one. These results were compared with AutoDock Vina[1] and Gnina, as reported by Sunseri and Koes[45]. We wish to emphasize that, though LIT-PCBA is a promising benchmark due to its use of experimental results rather than putative decoys, many of the assays used are cell-based rather than target-specific. Thus the activity annotations are noisy and not entirely dependent on the molecule's engagement with the protein target in question.

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.[9].

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

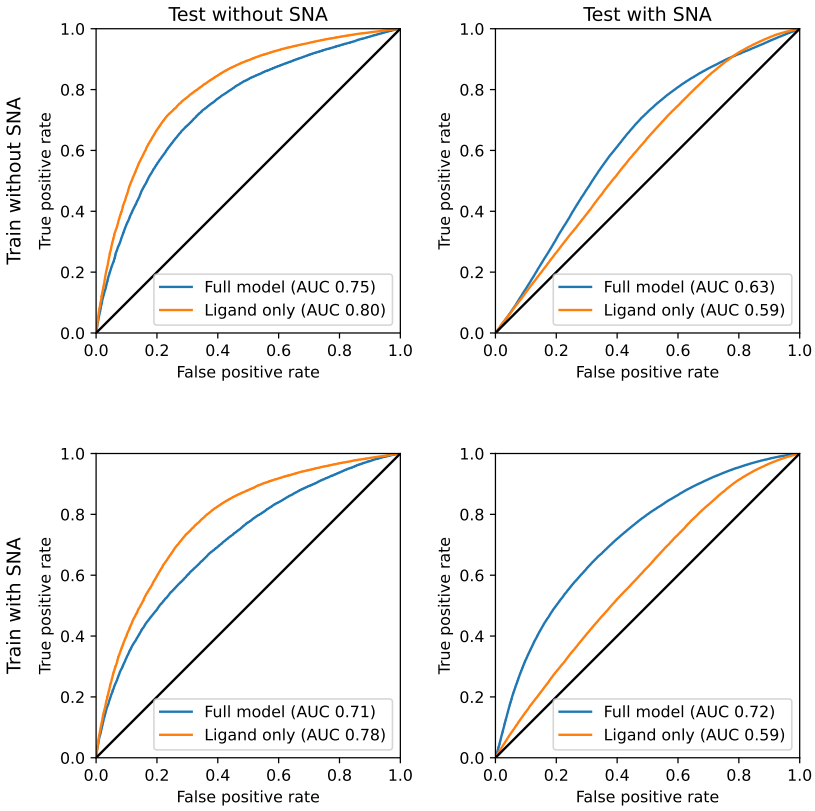

Figure 3: ROC curves for models trained and evaluated with and without SNA. When we train and evaluate the model without SNA (top left), the ligand-only model appears to outperform the full model. However, those same models both perform poorly after SNA is applied to the test set (top right). When we train with SNA, the performance of both models decreases on the non-SNA test set (bottom left). When evaluated on the SNA test set, only the full model maintains high performance (bottom right).

As can be seen in Figure 3, when BANANA is trained without SNA, the ligand-only model outperforms the ligand-and-receptor model (AUC 0.80 versus 0.75). The fact that receptor information reduces performance implies that this performance is entirely due to biased protein-ligand co-occurrence within the dataset. Indeed, when we apply SNA to the test set and try the non-SNA models, we see that both versions perform poorly (AUC

0.63 versus 0.59). 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 on the SNA test set (AUC 0.72 versus 0.59). Curiously, training with SNA slightly decreases the performance of both models on the non-SNA test set (AUC 0.71 and 0.78). These observations support our hypothesis that SNA provides a way to force the model to learn information about the ligand-receptor interaction rather than simply exploiting the fact that certain ligand scaffolds are only tested against certain protein classes in the dataset. All models tested outperform the KNN baseline (AUC 0.55).

Model Performance on LIT-PCBA

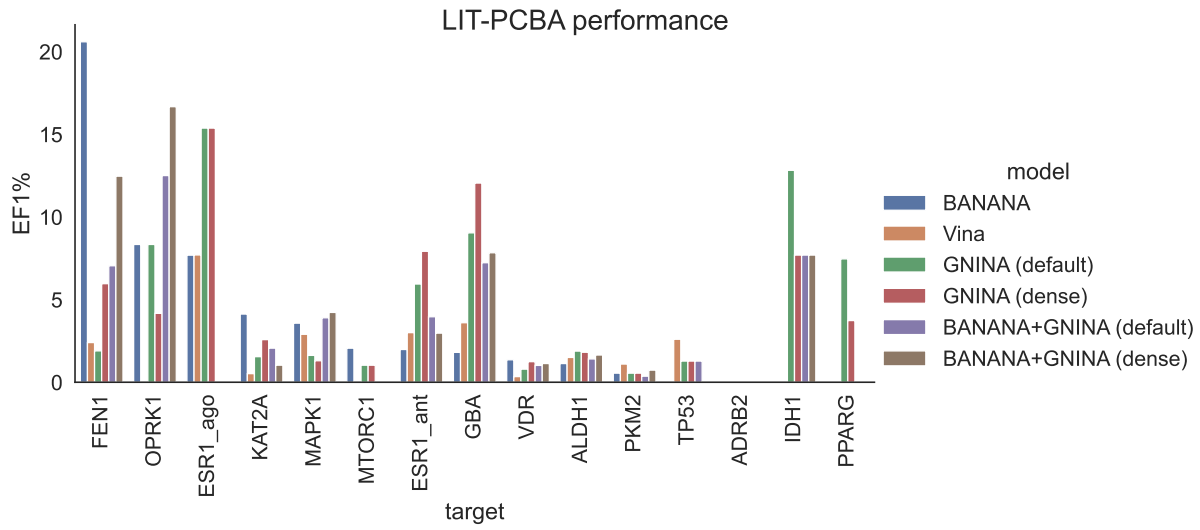

Figure 4: Performance of all the models on LIT-PCBA targets.

As Table 1 demonstrates, our model achieves performance comparable to Gnina with the default ensemble (EF1% 1.81 versus 1.88) on LIT-PCBA, though still falls behind Gnina with the dense model (EF1% 2.58). Intriguingly, the combined models are slightly worse than either individually; Banana+Gnina (default) and Banana+Gnina (dense) achieve AUCs of 1.41 and 1.13, respectively. Nonetheless, Banana, Gnina, and all combinations thereof do demonstrate superior performance to AutoDock Vina.

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.

Table 1: Median EF1%, NEF1%, and AUC values for all the models on the LIT-PCBA benchmark. Note that the combined BANANA+GNINA models are algorithms for selecting top compounds in a virtual screen that do not compute a single score for each compound. Thus they have no AUC values.

Model	**EF1%**	**NEF1%**	**AUC**
BANANA	1.81	0.02	0.58
Vina	1.1	0.01	0.58
GNINA (default)	1.88	0.02	0.61
GNINA (dense)	2.58	0.04	0.62
BANANA+GNINA (default)	1.41	0.02	
BANANA+GNINA (dense)	1.13	0.01	

Discussion

In recent years there have been many advances in deep learning model architectures for analyzing molecules.[46–50] Following traditional methods, deep learning models have often attempted to predict ligand activity from 3D protein-ligand complexes. However, this requires a dataset that contains activity data for known co-crystallized complexes, and such datasets are small and biased. Most methods have been trained on PDBbind, which only has about 20K activity values. CrossDocked augments this dataset with additional pose data, but does not expand on activity data. In principle, however, ligand selection can be accomplished without the knowledge of the binding site. Thus, we developed BigBind, a dataset of 583K protein-ligand activities along with the 3D structure of the respective receptor binding pocket.

We first tried training regression models to directly predict protein-ligand activity, but

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 was biased protein-ligand co-occurrence 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 appeared to be doing). This process, however, does not generalize well to new targets, which is why the models overfit.

To combat this, we used Stochastic Negative Addition. 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. This increased the size of the dataset to 1.2M datapoints. Our results are consistent with this hypothesis. We showed that a model trained on the resulting dataset is forced to learn information about protein-ligand interactions and can indeed 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. Banana can achieve such a large speed improvement because it does not require generating possible 3D structures for the protein-ligand complex, a bottleneck of traditional docking methods. 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 ultra-large libraries such as Enamine's REAL Space.[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 to explain the activities. Perhaps having more domain

knowledge about 3D space will improve model performance. Additionally, we hope to expand BigBind in the future. PubChem,[51] 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.

Data and software availability

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_data/BigBindV1.5.tar.gz. 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 their 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 R01GM140154.

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) Bender, B. J.; Gahbauer, S.; Luttens, A.; Lyu, J.; Webb, C. M.; Stein, R. M.; Fink, E. A.; Balius, T. E.; Carlsson, J.; Irwin, J. J.; Shoichet, B. K. A practical guide to large-scale docking. *Nature Protocols* **2021**, *16*, 4799–4832, Number: 10 Publisher: Nature Publishing Group.

(7) Wallach, I.; Heifets, A. Most Ligand-Based Classification Benchmarks Reward Memorization Rather than Generalization. *Journal of Chemical Information and Modeling* **2018**, *58*, 916–932, Publisher: American Chemical Society.

(8) 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.

(9) 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.

(10) 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].

(11) 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].

(12) 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.

(13) 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.

(14) 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.

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

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

(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) Liu, C.; Kutchukian, P.; Nguyen, N. D.; AlQuraishi, M.; Sorger, P. K. A Hybrid Structure-Based Machine Learning Approach for Predicting Kinase Inhibition by Small Molecules. *Journal of Chemical Information and Modeling* **2023**, *63*, 5457–5472, Publisher: American Chemical Society.

(25) 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.

(26) 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.

(27) 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.

(28) Mysinger, M. M.; Carchia, M.; Irwin, J. J.; Shoichet, B. K. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry* **2012**, *55*, 6582–6594, Publisher: American Chemical Society.

(29) Bauer, M. R.; Ibrahim, T. M.; Vogel, S. M.; Boeckler, F. M. Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets. *Journal of Chemical Information and Modeling* **2013**, *53*, 1447–1462, Publisher: American Chemical Society.

(30) 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.

(31) Chen, L.; Cruz, A.; Ramsey, S.; Dickson, C. J.; Duca, J. S.; Hornak, V.; Koes, D. R.; Kurtzman, T. Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening. *PLoS ONE* **2019**, *14*, e0220113.

(32) Sieg, J.; Flachsenberg, F.; Rarey, M. In Need of Bias Control: Evaluating Chemical

Data for Machine Learning in Structure-Based Virtual Screening. *Journal of Chemical Information and Modeling* **2019**, *59*, 947–961, Publisher: American Chemical Society.

(33) 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.

(34) 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.

(35) 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.

(36) 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.

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

(38) 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.

(39) PDBFixer. 2023; `https://github.com/openmm/pdbfixer`, original-date: 2013-08-29T22:29:24Z.

(40) Zhang, Y.; Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Research* **2005**, *33*, 2302–2309.

(41) 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].

(42) 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.

(43) 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].

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

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

(46) 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].

(47) 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 Rep-

resentation of Molecules. *ACS Central Science* **2018**, *4*, 268–276, Publisher: American Chemical Society.

(48) 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.

(49) 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. *Journal of Chemical Theory and Computation* **2021**, *17*, 2355–2363, arXiv:2012.12106 [physics].

(50) 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].

(51) 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.

TOC Graphic