MILCDock: Machine Learning-Enhanced Consensus Docking for Virtual Screening in Drug Discovery

Connor J. Morris

A senior thesis submitted to the faculty of
Brigham Young University
in partial fulfillment of the requirements for the degree of

Bachelor of Science

Dennis Della Corte, Advisor

Department of Physics and Astronomy

Brigham Young University

April 2022

ABSTRACT

MILCDock: Machine Learning-Enhanced Consensus Docking for Virtual Screening in Drug Discovery

Connor J. Morris
Department of Physics and Astronomy, BYU
Bachelor of Science

Molecular docking tools are regularly used to computationally identify new drug molecules in virtual screening for drug discovery. However, docking tools suffer from inaccurate scoring functions with widely varying performance on different proteins. To enable more accurate ranking of active over inactive ligands in virtual screening, we created a machine learning consensus docking tool, MILCDock, that uses predictions from five traditional molecular docking tools to predict the probability a ligand binds to a protein. MILCDock was trained and tested on data from both the DUD-E and LIT-PCBA docking datasets and shows improved performance over traditional molecular docking tools and other consensus docking methods on the DUD-E dataset. LIT-PCBA targets proved to be difficult for all methods tested. We also find that DUD-E data, though biased, can be effective in training machine learning tools if care is taken to avoid DUD-E's biases during training. Improved docking datasets with more diverse data and improved docking methods are needed to further improve consensus docking performance.

Keywords: molecular docking, consensus docking, machine learning, multi-layer perceptron, MLP, DUD-E, LIT-PCBA

ACKNOWLEDGMENTS

I would like to acknowledge the mentorship of Dr. Dennis Della Corte, who has given his time to train me as a scientist since early in my freshman year. His efforts have solidified my desire to pursue research as a career, have helped me succeed as an undergraduate student, and have provided me with the experiences necessary to be accepted to my dream graduate school program. I would also like to acknowledge the help of Brenden Stark, who helped me write docking scripts and run millions of tedious molecular docking simulations; Max Christopherson, who helped train machine learning models; and Jacob Stern, who helped train machine learning models and analyze the results.

Above all, I'd like to thank my wife Madeline and son Samuel, who have supported me on the ambitious path of becoming a scientist and make me excited to come home each day.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The exponentially increasing cost of discovering new drugs [1] suggests that drug discovery methods must continue to be improved. Molecular docking tools are one commonly used computational drug discovery tool [2]. These tools are regularly used in drug discovery to screen libraries of millions [3] to billions [4] of small molecules (also known as ligands) against target proteins in a process called virtual screening [5-8]. They predict the binding pose and affinity of the protein-ligand complex and are designed to be computationally efficient for screening of large compound libraries [2, 9-11]. The affinity score of each ligand can be used to rank-order a library of compounds to predict *in silico* which ones are most likely to bind to the target protein and become an effective drug. Virtual screening methods can increase the efficiency of drug discovery pipelines by reducing the number of experimental tests that must be done to find promising drug candidates.

Despite the great potential molecular docking tools have to reduce drug discovery costs, accurate differentiation between active and inactive ligands is a difficult task [12-14]. Docking tools often rank ligands inaccurately in virtual screening and their performance can vary widely across different protein targets, making it difficult to predict *a priori* whether a docking tool will perform well on a new virtual screening target [15].

Consensus docking is one method that can improve the accuracy of docking scoring functions [2, 9, 10]. Consensus docking incorporates docking scores from multiple scoring functions that use different methods to generate docking scores. This allows scoring functions that perform well on certain systems to compensate for weaker scoring functions, creating a consensus score that is more accurate and robust to varying systems [16]. Consensus methods were first proposed in 1999 [17] and have since been the focus of many studies. Clark *et al.* found that combining four scoring functions into a single score outperformed individual tools [18], Palacio-Rodriguez *et al.* combined results from multiple scoring functions using exponential consensus ranking to outperform traditional consensus methods [19], and Pedretti *et al.* used Enrichment Factor Optimization to create target-specific linear combinations of up to three scoring functions [20]. Other groups have used pose consensus, or selecting only docking poses that were predicted by multiple docking programs, to reduce the false positive rate of virtual screens. Houston *et al.* found that accepting only poses with less than 2 Å root-mean-square-deviation (RMSD) from poses generated by other docking programs increased pose accuracy and reduced the false positive rate [21]. Gimeno *et al*. used pose consensus to predict potential inhibitors of the SARS CoV-2 main protease [22].

Recently, machine learning (ML) emerged as an alternative method for combining results from multiple docking tools in consensus docking. ML methods rely on datasets like DUD[23], DUD-E [24], MUV[25], PDBbind[26], and LIT-PCBA[27] for training and testing. These datasets consist of protein targets associated with libraries of small molecules that are labelled as active, decoy, or inactive against that protein. Active molecules are experimentally verified molecules that bind to the protein, inactive molecules are experimentally verified molecules that do not bind to the protein, and decoys are computationally generated molecules that probably do not bind to the protein but are not experimentally verified.

Several groups have used these datasets to train ML consensus methods. Erickson *et al.* used gradient boosting to train an ensemble of decision tree classifiers on 21 targets from the DUD-E database and found that it outperformed non-ML methods on each considered target [16]. Perez-Castillo *et al.* used a genetic algorithm to find the combination of scoring function components that maximizes performance [28]. Wang *et al.* used random forests to train an improved scoring function based on 20 descriptors in addition to an Autodock Vina score [29]. Ye *et al.* trained target-specific scoring functions on eight diverse DUD-E targets using XGBoost, which suggested improved results over classical scoring functions and some machine learning-based scoring functions [30].

Many ML methods suffer from exclusive training and testing on the DUD-E dataset[16, 28, 30] which has been shown to be biased [31] and not realistic for virtual screening applications [27]. The bias in DUD-E originates in the decoy molecule generation. ML methods like convolutional neural networks trained on ligand structures and molecular properties have learned to differentiate between active and decoy ligands simply by memorizing ligand structures rather than by predicting protein-ligand interactions [31-33]. The binding affinity of DUD-E active ligands and the ratio of actives to decoys also differs from real virtual screening scenarios. Therefore, a high performance on the DUD-E dataset alone does not qualify a method as useful for virtual screening [27].

LIT-PCBA is a dataset that can overcome the DUD-E bias [27]. It contains only 15 targets, compared to 102 in DUD-E, but its ligand libraries were experimentally verified to follow an affinity distribution and active-to-decoy ratio that is similar to real virtual screening. This suggests that virtual screening tests on LIT-PCBA are more reliable than tests on the DUD-E dataset alone.

Here, we present a machine learning consensus docking method that is trained and tested on data from DUD-E and LIT-PCBA. This method uses docking scores and RMSD measurements between poses from five molecular docking tools to predict the probability of ligand binding. Using only outputs from diverse docking scoring functions and pose sampling methods allows the model to learn without seeing the specific ligand descriptors that form the root cause of the DUD-E bias. We find that our method significantly outperforms traditional docking tools and classical machine learning methods on the DUD-E dataset, and sets a benchmark for other docking methods on the LIT-PCBA dataset.

Chapter 2

Methods

The machine learning consensus (MILC) docking method uses a multi-layer perceptron (MLP) to combine binding affinity and pose predictions from five traditional molecular docking tools (see Fig. 2.1). This section describes the datasets and docking methods used to generate training data, the training method for the machine learning model, and the metrics used to evaluate performance.

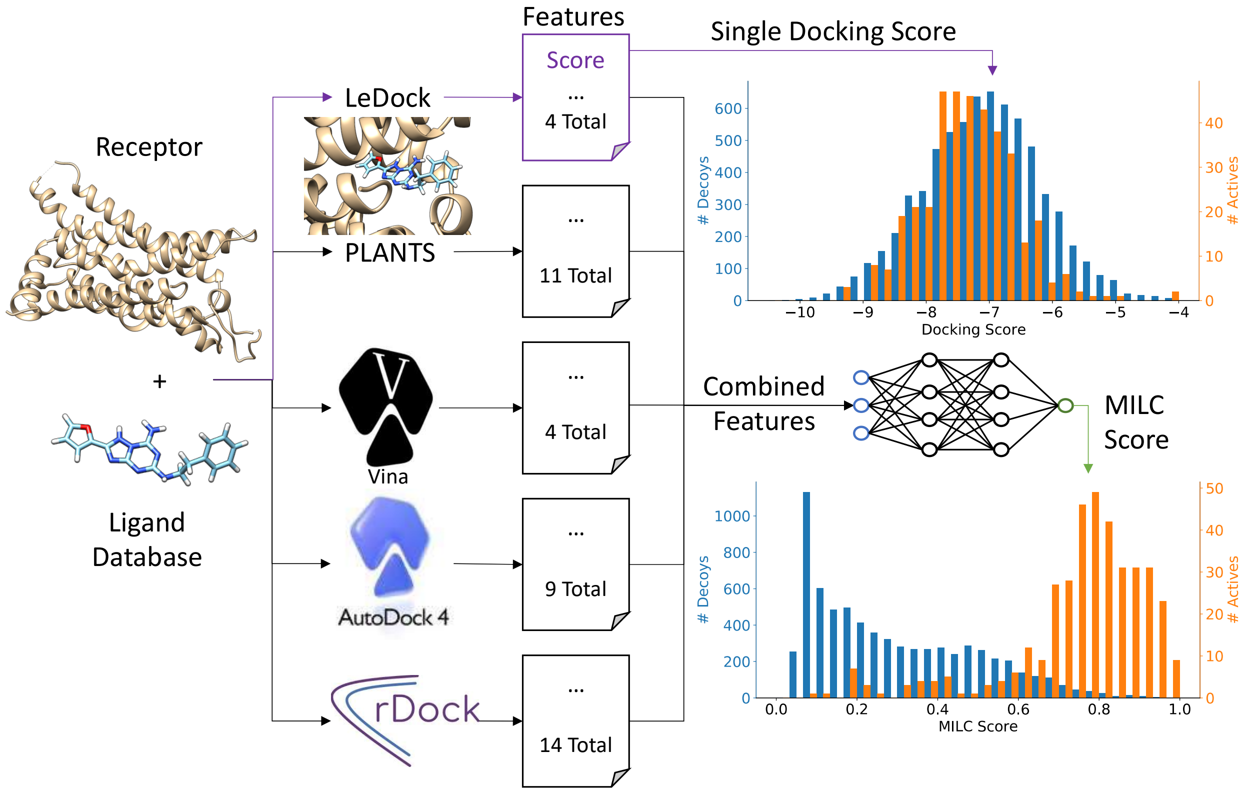

Figure 2.1 Visualization of data flow in Machine Learning Consensus Docking. A receptor (protease Renin) and a database of ligands (from DUD-E) are passed through five docking tools. Top right histogram shows the separation in distributions for docking scores between actives (orange) and decoys (blue) as predicted by the

single docking tool LeDock. A total of 52 feature (37 directly from tools, 15 pairwise RMSD values of predicted poses) is combined and passed through an ensemble machine learning model to predict the MILC score. Lower right histogram shows the separation of distributions as predicted by MILC for the same test case.

2.1 Receptor and Ligand Databases

The DUD-E database contains 102 protein targets, each paired with an average of 224 experimentally verified active ligands and 50 decoy ligands per active. The decoy ligands are not experimentally verified and were generated to be physically similar but topologically different from the active molecules[31].

LIT-PCBA contains 15 protein targets, each paired with an average of 669 active and 187,000 inactive ligands. Inactive LIT-PCBA ligands are experimentally verified. LIT-PCBA was created to be unbiased, largely in response to bias that exists in other docking datasets like DUD-E[31].

Of the 102 DUD-E protein targets available, 86 were selected for use in this project. The remaining targets were excluded since they seemed non-ideal for the docking tools we used (i.e. metal ions or other ligands near the binding site). Two DUD-E targets failed at some point while docking with each of the five docking tools and were excluded from use in machine learning training and testing, leaving 84 total targets from the DUD-E database. All 15 LIT-PCBA targets were used in this project. In total, we docked over 3.8 million unique protein-ligand pairs across both the DUD-E and LIT-PCBA datasets using five docking tools, resulting in nearly 20 million docking simulations done in preparation for machine learning training and testing.

2.2 Docking

Five docking tools were used to dock active and decoy ligands from DUD-E[24] and LIT-PCBA[27] to their respective protein targets: Autodock Vina [34], Autodock4 [35], PLANTS [36], rDock [37], and LeDock [38]. Custom scripts were written for all five docking tools to run with the same command and input files. Binding sites for all programs were automatically generated by creating a box around the crystal ligand structures provided with each target that was at least 25 Å in each dimension. For docking tools that use spherical binding sites as input (rDock and Plants), a similarly sized sphere was generated instead. Further details about individual docking tools follow.

Autodock Vina version 1.1.2

Receptor files were prepared by manually converting histidine residue names to Amber format, then using prepare_receptor4.py from AutoDockTools to convert to pdbqt format. Ligand files were prepared using the prepare_ligand4.py script from AutoDockTools. The exhaustiveness parameter was left at its default, 8, and num_modes was set to 20. Vina uses an empirical/knowledge-based scoring function.

Autodock4 version 4.2.6

Docking grids were prepared using prepare_gpf4.py from AutoDockTools and the autogrid4 executable file. Receptor and ligand files were prepared using prepare_receptor4.py and prepare_ligand4.py, respectively, from AutoDockTools. Since Autodock4 outputs many intermediate scores from its docking calculations in addition to a final docking score, many of these intermediate scores were included in machine learning input vectors. Autodock4 uses an empirical/knowledge-based scoring function.

PLANTS version 1.2

Receptor files were prepared with SPORES version 1.3, using the 'complete' running mode. In the configuration file, 'chemplp' was used for the scoring function setting, with 'speed1' for the search speed. For the clustering algorithm, ‘cluster_structures’ was set to 20, and ‘cluster_rmsd’ set to 2.0. Since PLANTS outputs many intermediate scores from its docking calculations in addition to a final docking score, many of these intermediate scores were included in machine learning input vectors. PLANTS uses an empirical scoring function.

rDock version 2013.1

Receptor files were converted to MOL2 format and ligand files were converted to SD format using OpenBabel[39]. Docking cavities were mapped using the rbcavity executable file, using the two sphere method, with small sphere radius of 1.5 Å and large sphere radius of 4.0 Å. The scoring function used was RbtCavityGridSF with WEIGHT 1.0, RMAX 0.1, and QUADRATIC FALSE. Docking was performed using the rbdock executable and default options for free docking. Since rDock outputs many intermediate scores from its scoring function, intermolecular scores and normalized scores were used as inputs into the machine learning model in addition to its final score. rDock uses an empirical/force field-based scoring function.

LeDock version 1.0

Receptor files were prepared by converting histidine and zinc residue names to CHARMM format, then running the lepro executable on them. LeDock uses an empirical/force field-based scoring function.

2.3 Creation of Train/Validation/Test Split

Basic MLP: To construct a train/validation/test split, a BLAST search [40] was used to find the pairwise alignment similarities of all protein targets in the dataset, resulting in a 99 x 99 matrix. Spectral clustering was then used on the pairwise similarity matrix to create 10 clusters, each containing a set of protein targets. Protein targets that were clustered together are more similar to each other than to proteins in other clusters. The clusters were then merged to create splits that placed ~50% of the targets in the train set, ~25% in the validation set, and ~25% in the test set, with similar percentages of DUD-E and LIT-PCBA targets in each. These proportionally large validation and test splits were chosen due to the low number of unique targets to allow more accurate testing.

Ensemble MLP: After the hyperparameter search, to train on a larger dataset, train and validation targets were combined into one set. Spectral clustering was again used to create 30 clusters from this combined set, followed by cluster merging, to obtain 8 sets with ~10 targets in each. These folds were used for the multi-fold ensemble training described in section 2.6.

2.4 Inputs

Input vectors for machine learning were created by extracting docking pose and binding affinity information from docking output files using custom scripts. A total of 52 features were used in each input vector to describe the protein-ligand interaction. 37 of those features were derived from docking tool energy score predictions and 15 were derived from RMSD calculations between poses generated by the docking tools, inspired by Ref. [21].

For each docking tool, the best energy score, the average of the top three energy scores, and the range of the top three energy scores were included as features. Intermediate score values were also extracted from Autodock4, PLANTS, and rDock outputs and included in the input vector. For the RMSD-based features, the RMSD between the top scoring pose from each docking tool and the average RMSD between the top 3 scoring poses of each tool were included. A small fraction of the total docked ligands produced errors during creation of machine learning input vectors. These ligands were excluded from any downstream application in machine learning training, testing, results, etc.

2.5 Class Balancing

Significant class imbalance exists in both the DUD-E and LIT-PCBA datasets. There are nearly 3.8 million decoys in the combined DUD-E+LIT-PCBA dataset, compared to only 41,420 active ligands. Furthermore, there are nearly 350,000 total ligands associated with FEN1, the target with the most ligands, compared to 688 ligands associated with fgfr1, the target with the fewest ligands. To train the models with equal weighting of actives and decoys as well as equal weight per target, ligands underwent two steps of oversampling. First, actives were oversampled to be approximately equal in number to the decoys for each target. Second, all actives and decoys for each target were oversampled to be approximately equal in number to the total number of ligands for the target with the most ligands.

2.6 Model Selection

We compared five consensus methods on the validation dataset: a naïve consensus method, three classical machine learning methods (XGBoost, Random Forest, and Naïve Bayes classifier), and an MLP.

The naïve consensus method was used as baseline, non-ML consensus method to compare against. This method created z-scores from each traditional docking tool by subtracting the mean and dividing by the standard deviation of the top docking score for each ligand in the training set. The final naïve consensus score was obtained by summing the five z-scores.

XGBoost, Random Forest, and Naïve Bayes classifiers were all trained using default parameters on the associated tools in the scikit-learn python library. The training dataset was reduced to have a maximum of 80,000 decoy ligands associated with each target while training these methods to enable the entire dataset to fit in computer memory at the same time. XGBoost performed the best on the validation dataset of these methods, so only XGBoost results are reported on the test set to represent the best of the classical ML methods.

The MLP was created using a binary cross-entropy loss function. To find the optimal hyperparameters, we trained many different versions of the MLP with different layer dimensions, dropout, weight decay, activation functions, and weight gain. The model that performed best according to the average BEDROC score on the validation set was selected as the best MLP and then evaluated on the test set (see Chapter 3).

To increase the amount of data used to train the MLP, an ensemble model was also created using k-fold cross validation on the combined train/validation set described in section 2.3. The final scores generated by the ensemble model are combinations of 8 different MLPs, each trained on different subsets of the combined train/validation set.

2.7 Evaluation

Prediction performance was evaluated using the enrichment factor (EF), BEDROC[41], and area under the receiver operating characteristic curve (AUC) scores.

EF considers only the top x% of ranked ligands and gives a score based on how "enriched" with active ligands that subset is. The equation for calculating EF for the top 1% of the database is given in Eq. (2.1). In Eq. (2.1), A represents the number of active ligands, and N represents the total number of ligands (including both actives and decoys). The subscripts *top1* and *dataset* indicate whether to count those ligands from only the top 1% of ranked ligands, or to count ligands from the entire dataset. All EF scores in this paper represent EF of the top 1% of the dataset. An EF of 2 means that the top 1% of ranked ligands contain 2 times more actives than would appear in a random ordering.

$$\left(\frac{A_{top1}}{N_{top1}}\right)/\left(\frac{A_{dataset}}{N_{dataset}}\right) \tag{2.1}$$

AUC measures a method's ability to accurately rank active ligands over decoys across the entire dataset, as opposed to EF that looks only at the top 1% of the ranked poses. It is bounded between 0 and 1 with a score of 0.5 corresponding to random sorting and scores greater than 0.5 being better than random. AUC is useful as an orthogonal metric to EF to see how a method performs across the entire dataset, since EF scores can be volatile if the active-decoy ratio is very low for a specific target.

BEDROC is like AUC in that it is bounded between 0 and 1, but it is modified for the early enrichment problem, where actives sorted early on in the database are given higher weight. Since BEDROC emphasizes ligands found earlier in the dataset, its scores tend to follow the same trends as EF scores. BEDROC has a parameter, α, that modifies how much weight is given to early active ligands. This value is set to α=160.9 to correspond to an EF at 1% for all BEDROC scores in this paper.

Chapter 3

Results and Discussion

Figure 3.1 shows the average performance of our multi-layer perceptron (MLP) methods compared to traditional docking and other consensus methods across test targets from both the DUD-E and LIT-PCBA datasets. On average, the MLP methods significantly outperform the traditional docking tools, XGBoost, and Naïve Consensus methods at enriching the number of actives in the top 1% of the dataset.

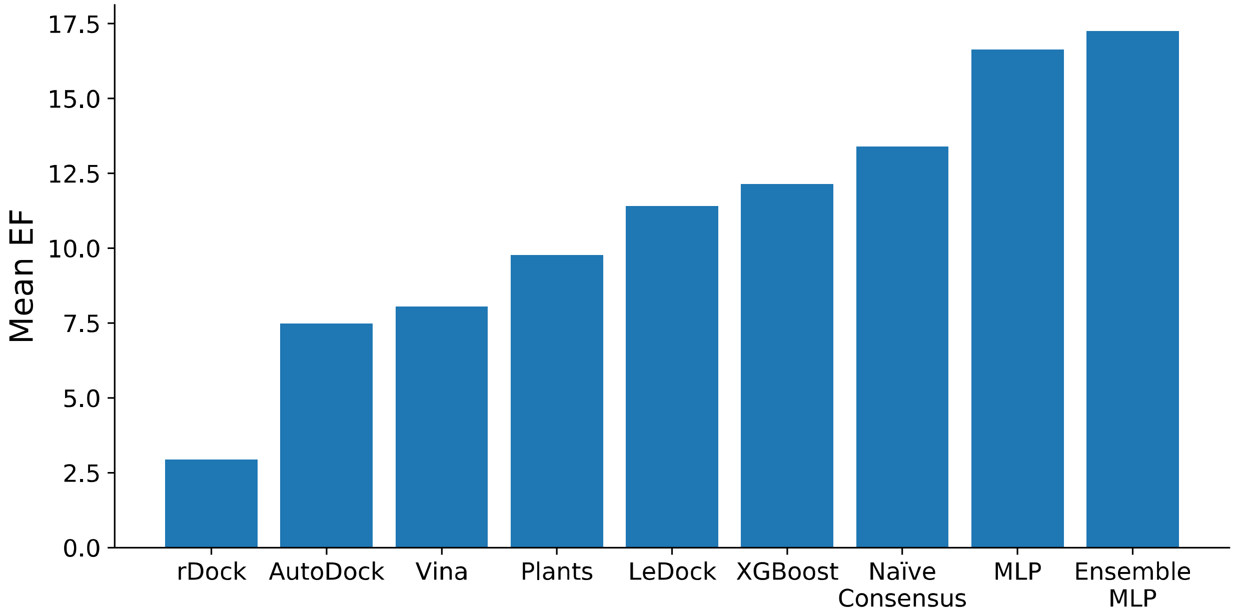

Figure 3.1 Bar chart of average enrichment factor (EF) of the top 1% of ranked poses for docking tools and consensus methods on a test set of four LIT-PCBA and twenty DUD-E receptors. Multi-layer Perceptron (MLP) methods achieve the best results.

More detailed results are shown in Table 3.1. On the DUD-E data, the Ensemble MLP and MLP perform best with average EFs of 19.81 and 18.96, respectively. This means the MLP methods on average sort approximately 19 times more actives into the top 1% of the dataset than would be there if the dataset were randomly sorted. The MLP methods clearly outperform both the traditional docking methods and the other consensus methods on the DUD-E dataset, with the next-best average EF being the Naïve Consensus score of 15.34. The AUC scores (see Table A.1) show similar trends for DUD-E targets.

Table 3.1 Detailed results for all docking tools and consensus methods on test targets from LIT-PCBA and DUD-E databases, as measured by enrichment factor (EF). Bolded values indicate the best score in each row. For tables summarizing AUC scores, see Appendix A.

LIT-PCBA EF Scores

Targets	Vina	rDock	Plants	LeDock	AutoDock4	Naïve Consensus	XGBoost	MLP	Ensemble MLP
ADRB2	0.00	0.00	0.00	0.00	0.00	0.00	5.88	**11.77**	5.88
ALDH1	1.73	1.70	1.66	1.62	**2.11**	2.05	1.42	0.80	0.91
GBA	4.91	3.07	4.91	1.84	5.52	5.52	3.07	4.91	**6.14**
IDH1	2.56	2.56	2.56	**5.13**	0.00	**5.13**	**5.13**	0.00	2.56
LIT-PCBA-MEAN	2.30	1.83	2.28	2.15	1.91	3.18	3.87	**4.37**	3.87
LIT-PCBA STD	2.04	**1.35**	2.05	2.15	2.61	2.63	2.03	5.38	2.56

DUD-E EF Scores

Targets	Vina	rDock	Plants	LeDock	AutoDock4	Naïve Consensus	XGBoost	MLP	Ensemble MLP
adrb2	2.75	0.69	16.76	4.82	9.41	6.89	15.84	22.95	24.79
csf1r	1.76	5.62	5.97	14.40	2.46	**15.46**	8.78	11.59	11.94
egfr	4.73	0.36	12.72	**20.60**	5.33	16.24	12.48	19.99	18.90
fak1	18.45	0.00	15.82	10.54	1.76	13.18	12.30	12.30	**22.84**
fgfr1	1.02	1.53	1.02	1.53	1.53	**2.55**	1.53	0.51	0.51
glcm	0.00	0.00	**13.09**	1.64	0.00	0.65	7.20	5.24	9.16
hmdh	3.75	1.71	13.98	6.48	4.09	8.87	16.71	22.85	23.19
igf1r	13.30	0.44	15.96	20.84	11.97	24.39	15.96	**24.83**	21.73

jak2	15.78	0.00	11.18	15.12	5.92	15.12	13.81	21.04	**23.01**
kpcb	17.49	4.07	16.67	14.23	17.89	26.84	23.99	**30.50**	**30.50**
lck	7.92	1.61	1.47	12.90	13.05	18.91	13.19	24.48	25.36
mapk2	8.31	0.00	7.33	4.89	3.91	12.22	16.62	19.55	**24.93**
met	12.02	0.41	11.19	37.30	16.58	33.57	25.28	36.89	39.79
mk14	5.03	0.87	0.22	2.73	2.62	4.70	3.83	**7.43**	6.67
plk1	1.30	22.67	14.25	20.08	3.24	**26.56**	9.72	22.02	19.43
pur2	2.00	8.00	**14.50**	14.00	14.00	**14.50**	**14.50**	**14.50**	**14.50**
reni	1.84	0.00	10.25	0.26	8.15	5.78	14.20	**18.67**	16.56
src	4.48	0.48	0.36	4.48	3.87	6.53	5.93	**15.24**	14.03
tgfr1	9.79	0.36	2.18	**12.69**	5.44	10.15	7.25	11.97	11.60
vgfr2	**15.19**	0.67	3.67	12.19	2.50	12.52	2.50	9.18	10.02
wee1	44.96	16.95	**46.43**	44.96	45.69	**46.43**	**46.43**	**46.43**	**46.43**
DUD-E MEAN	9.14	3.16	11.19	13.18	8.54	15.34	13.72	18.96	**19.81**
DUDE STD	10.11	**5.97**	9.99	11.34	9.97	11.13	9.74	10.63	10.71

Overall EF Scores

Targets	Vina	rDock	Plants	LeDock	AutoDock4	Naïve Consensus	XGBoost	MLP	Ensemble MLP
OVERALL MEAN	8.04	2.95	9.77	11.41	7.48	13.39	12.14	16.63	**17.26**
OVERALL STD	9.61	**5.50**	9.74	11.17	9.48	11.17	9.65	11.29	11.49

The LIT-PCBA test results, however, are more nuanced. According to EF (Table 3.1), the MLP does best on average, with the Ensemble MLP and XGBoost close behind. But the AUC scores (Table A.1) tell a different story. Vina, Autodock4, and the MLP all rank at the top with approximately the same AUC score on average, with the Ensemble MLP being one of the worst performing methods. This result is surprising, considering that the MLP methods perform better according to the EF metric.

A closer investigation of the EF metric on LIT-PCBA targets reveals that the very small active-to-inactive ratio on several LIT-PCBA targets makes the EF metric highly volatile. For example, on ADRB2, there are only 17 active ligands compared to 301808 inactive ligands. If virtual screening sorts even a single active ligand into the top 1% of the dataset, the EF factor will jump from 0.0 to 5.88. This can create high variance in the results based on potential random sorting of a single active in the top 1% of the dataset. Thus, on targets with a very low active-to-inactive ratio, it is important to qualify EF results with AUC results, which measure a method's ability to sort a dataset better than random across the entire dataset. GBA, IDH1, and ADRB2 targets from the LIT-PCBA test set all have less than a 0.1% active-inactive ratio, and thus are subject to volatility in their EF scores. The average performance of the MLP according to both the EF and AUC metrics suggests that the MLP, which outperforms traditional docking tools on the DUD-E dataset, only performs approximately as well as the best individual docking tools and the other consensus methods on the LIT-PCBA dataset on average. When looking at robustness to different targets, XGBoost seems to be the most consistent since it has no EF scores of less than 1 and it has the lowest standard deviation on its AUC scores. The small target sample size and the volatility of the EF metric, however, make it difficult to choose a clear best method on the LIT-PCBA data. The overall poor performance of all methods here may indicate that general docking tools must be further improved before any consensus method is able to perform highly on LIT-PCBA targets.

Table 3.2 Comparison of Ensemble MLP models trained and tested on different combinations of the DUD-E and LIT-PCBA datasets. Numbers shown are average EF scores on the associated test set. Rows correspond to test sets, columns to training sets.

	DUD-E – Train	LIT-PCBA – Train	DUD-E + LIT-PCBA – Train
DUD-E – Test	19.48	2.01	**19.81**
LIT-PCBA – Test	**5.13**	2.26	3.87
DUD-E + LIT-PCBA – Test	17.19	2.05	**17.26**

Table 3.2 compares models trained and tested on DUD-E alone, LIT-PCBA alone, and the DUD-E+LIT-PCBA combined dataset. In terms of EF score, the model trained on the combined DUD-E and LIT-PCBA training sets achieves best performance on DUD-E and the combined test set while the model trained on only DUD-E achieves best performance on the LIT-PCBA test set. By the AUC metric, the model trained only on DUD-E achieves best performance on all test sets (Table A.2). This suggests that including LIT-PCBA in the training dataset offers little to no benefit in the final trained model. This is surprising, since LIT-PCBA contains a different ligand potency distribution than DUD-E, so it is natural to expect that training on LIT-PCBA targets would improve performance on other LIT-PCBA targets. Two factors that may contribute to the irrelevance of training on LIT-PCBA data are: 1) the low number of LIT-PCBA targets compared to the number of DUD-E targets, which may have made their contribution to the overall network training negligible; and 2) the general low performance of docking tools on LIT-PCBA targets, which may have made it difficult for the network to learn anything meaningful from the LIT-PCBA data.

A direct comparison of our MLP methods with other published ML consensus methods is difficult since training and test sets are often not made available. We therefore trained models comparable to the naïve consensus method by Clark *et al*[18], the boosting method by Ericksen *et al*[16], and the random forests method by Wang *et al*[29] on the same dataset as the MLP. Our method is closest to a combination of the work by Ericksen *et al* and Wang *et al.* Like Ericksen *et al*, our method is a machine-learning based consensus docking method and, like Wang *et al*, it learns on multiple features extracted from the docking programs.

To our knowledge, our work is the first to employ machine learning consensus docking over both docking score and pose RMSD features extracted from multiple docking programs, the first to use an MLP, and the first consensus docking method to train or test on the more challenging LIT-PCBA dataset. It avoids DUD-E's biases in its training, clearly outperforms other methods on a DUD-E test set, and provides a baseline for improving virtual screening methods on the LIT-PCBA test set.

3.1 Conclusion

Docking plays an important role in large-scale library screens for drug discovery. Individual docking tools have strengths and weaknesses that result from differences in their scoring functions. Consensus-based methods allow more robust prediction of binding affinities and better separation of active from inactive molecules. From all tested consensus methods, our MLP methods performed best on the DUD-E test set, while yielding results comparable to other traditional and consensus docking methods on the LIT-PCBA test set. The ability of the consensus methods to perform well on LIT-PCBA seems to be limited by the lack of diversity in targets for training, and the poor performance of traditional docking tools this database. As individual docking tools improve and as datasets evolve to better represent virtual screening scenarios, consensus docking methods will likewise become more robust and accurate.

Appendix A

Additional Results

Table A.1 AUC scores of different docking methods on the DUD-E and LIT-PCBA datasets.

LIT-PCBA AUC Scores

Targets	Vina	rDock	Plants	LeDock	AutoDock4	Naïve Consensus	XGBoost	MLP	Ensemble MLP
ADRB2	0.347	0.400	0.381	0.403	0.389	0.296	**0.492**	0.443	0.418
ALDH1	0.589	0.608	0.566	0.584	**0.624**	0.615	0.543	0.498	0.522
GBA	0.645	0.554	0.635	0.635	0.651	**0.686**	0.652	0.675	0.618
IDH1	**0.681**	0.430	0.392	0.554	0.593	0.562	0.547	0.638	0.563
LIT-PCBA-MEAN	**0.565**	0.498	0.494	0.544	0.564	0.540	0.559	0.564	0.531
LIT-PCBA STD	0.151	0.099	0.127	0.100	0.119	0.170	**0.067**	0.111	0.084

DUD-E AUC Scores

Targets	Vina	rDock	Plants	LeDock	AutoDock4	Naïve Consensus	XGBoost	MLP	Ensemble MLP
adrb2	0.676	0.530	0.835	0.595	0.679	0.709	0.809	**0.879**	0.875
csf1r	0.681	0.711	0.448	**0.803**	0.643	0.777	0.609	0.646	0.646
egfr	0.628	0.672	0.647	**0.812**	0.575	0.755	0.650	0.752	0.741
fak1	0.796	0.759	0.725	0.799	0.487	**0.835**	0.745	0.828	0.770
fgfr1	**0.561**	0.488	0.495	0.549	0.535	0.535	0.521	0.543	0.514

glcm	0.472	0.454	0.826	0.707	0.509	0.658	0.857	**0.901**	0.888
hmdh	0.750	0.468	0.794	0.529	0.767	0.718	**0.823**	0.796	0.767
igf1r	0.839	0.686	0.667	0.836	0.762	**0.880**	0.689	0.838	0.817
jak2	0.761	0.711	0.785	0.816	0.529	0.783	0.811	**0.902**	0.874
kpcb	0.722	0.675	0.683	0.733	0.722	0.779	0.875	0.923	**0.930**
lck	0.769	0.718	0.579	0.818	0.733	0.838	0.723	**0.857**	0.828
mapk2	0.791	0.746	0.762	0.704	0.722	0.780	0.803	**0.905**	0.886
met	0.804	0.764	0.803	0.866	0.747	0.883	0.835	0.899	**0.901**
mk14	0.710	0.439	0.441	**0.756**	0.596	0.682	0.577	0.634	0.534
plk1	0.591	0.695	0.789	**0.842**	0.531	0.768	0.716	0.826	0.805
pur2	0.903	0.789	0.995	0.977	0.997	0.997	0.963	**0.999**	0.997
reni	0.608	0.314	0.852	0.604	0.361	0.494	0.902	0.947	**0.951**
src	0.662	0.620	0.525	**0.771**	0.599	0.736	0.623	0.763	0.717
tgfr1	**0.885**	0.570	0.731	0.870	0.723	0.863	0.666	0.871	0.763
vgfr2	0.749	0.628	0.600	**0.832**	0.668	0.774	0.547	0.777	0.623
wee1	0.957	0.969	0.978	0.959	0.956	0.989	0.981	**0.991**	0.989
DUD-E MEAN	0.729	0.638	0.712	0.770	0.659	0.773	0.749	**0.832**	0.801
DUDE STD	0.119	0.149	0.157	0.121	0.151	0.123	0.133	**0.117**	0.137

Overall AUC Scores

Targets	**Vina**	**rDock**	**Plants**	**LeDock**	**AutoDock4**	**Naïve Consensus**	**XGBoost**	**MLP**	**Ensemble MLP**
OVERALL MEAN	0.703	0.616	0.677	0.734	0.644	0.736	0.718	**0.789**	0.758
OVERALL STD	**0.136**	0.150	0.171	0.144	0.148	0.154	0.143	0.151	0.163

Table A.2 Comparison of Ensemble MLP models trained and tested on different combinations of the DUD-E and LIT-PCBA datasets. Numbers shown are average AUC scores on the associated test set. Rows correspond to test sets, columns to training sets.

	DUD-E – Train	LIT-PCBA – Train	DUD-E + LIT-PCBA – Train
DUD-E – Test	**.848**	.569	.832
LIT-PCBA – Test	**.583**	.533	.564
DUD-E + LIT-PCBA – Test	**.806**	.563	.789

Bibliography

1. Scannell, J.W., et al., *Diagnosing the decline in pharmaceutical R&D efficiency.* Nature reviews Drug discovery, 2012. **11**(3): p. 191-200.
2. Morris, C.J. and D.D. Corte, *Using molecular docking and molecular dynamics to investigate protein-ligand interactions.* Modern Physics Letters B, 2021. **35**(08): p. 2130002.
3. Lyu, J., et al., *Ultra-large library docking for discovering new chemotypes.* Nature, 2019. **566**(7743): p. 224-229.
4. Gorgulla, C., et al., *An open-source drug discovery platform enables ultra-large virtual screens.* Nature, 2020. **580**(7805): p. 663-668.
5. Wang, X., et al., *Study on structure activity relationship of natural flavonoids against thrombin by molecular docking virtual screening combined with activity evaluation in vitro.* Molecules, 2020. **25**(2): p. 422.
6. Marinho, E.M., et al., *Virtual screening based on molecular docking of possible inhibitors of Covid-19 main protease.* Microbial Pathogenesis, 2020. **148**: p. 104365.
7. Kist, R. and R.A. Caceres, *New potential inhibitors of mTOR: a computational investigation integrating molecular docking, virtual screening and molecular dynamics simulation.* Journal of Biomolecular Structure and Dynamics, 2017. **35**(16): p. 3555-3568.
8. Hosseini, F.S. and M. Amanlou, *Anti-HCV and anti-malaria agent, potential candidates to repurpose for coronavirus infection: virtual screening, molecular docking, and molecular dynamics simulation study.* Life sciences, 2020. **258**: p. 118205.
9. Torres, P.H., et al., *Key topics in molecular docking for drug design.* International journal of molecular sciences, 2019. **20**(18): p. 4574.
10. Wang, Z., et al., *Combined strategies in structure-based virtual screening.* Physical Chemistry Chemical Physics, 2020. **22**(6): p. 3149-3159.
11. Berenger, F., et al., *Lean-Docking: Exploiting Ligands' Predicted Docking Scores to Accelerate Molecular Docking.* Journal of Chemical Information and Modeling, 2021. **61**(5): p. 2341-2352.
12. Wang, Z., et al., *Comprehensive evaluation of ten docking programs on a diverse set of protein–ligand complexes: the prediction accuracy of sampling power and scoring power.* Physical Chemistry Chemical Physics, 2016. **18**(18): p. 12964-12975.
13. Cheng, T., et al., *Comparative assessment of scoring functions on a diverse test set.* Journal of chemical information and modeling, 2009. **49**(4): p. 1079-1093.
14. Su, M., et al., *Comparative assessment of scoring functions: the CASF-2016 update.* Journal of chemical information and modeling, 2018. **59**(2): p. 895-913.
15. Warren, G.L., et al., *A critical assessment of docking programs and scoring functions.* Journal of medicinal chemistry, 2006. **49**(20): p. 5912-5931.
16. Ericksen, S.S., et al., *Machine learning consensus scoring improves performance across targets in structure-based virtual screening.* Journal of chemical information and modeling, 2017. **57**(7): p. 1579-1590.

17. Charifson, P.S., et al., *Consensus scoring: A method for obtaining improved hit rates from docking databases of three-dimensional structures into proteins.* Journal of medicinal chemistry, 1999. **42**(25): p. 5100-5109.
18. Clark, R.D., et al., *Consensus scoring for ligand/protein interactions.* Journal of Molecular Graphics and Modelling, 2002. **20**(4): p. 281-295.
19. Palacio-Rodríguez, K., et al., *Exponential consensus ranking improves the outcome in docking and receptor ensemble docking.* Scientific reports, 2019. **9**(1): p. 1-14.
20. Pedretti, A., et al., *Rescoring and linearly combining: A highly effective consensus strategy for virtual screening campaigns.* International journal of molecular sciences, 2019. **20**(9): p. 2060.
21. Houston, D.R. and M.D. Walkinshaw, *Consensus docking: improving the reliability of docking in a virtual screening context.* Journal of chemical information and modeling, 2013. **53**(2): p. 384-390.
22. Gimeno, A., et al., *Prediction of novel inhibitors of the main protease (M-pro) of SARS-CoV-2 through consensus docking and drug reposition.* International journal of molecular sciences, 2020. **21**(11): p. 3793.
23. Huang, N., B.K. Shoichet, and J.J. Irwin, *Benchmarking sets for molecular docking.* Journal of medicinal chemistry, 2006. **49**(23): p. 6789-6801.
24. Mysinger, M.M., et al., *Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking.* Journal of medicinal chemistry, 2012. **55**(14): p. 6582-6594.
25. Rohrer, S.G. and K. Baumann, *Maximum unbiased validation (MUV) data sets for virtual screening based on PubChem bioactivity data.* Journal of chemical information and modeling, 2009. **49**(2): p. 169-184.
26. Liu, Z., et al., *PDB-wide collection of binding data: current status of the PDBbind database.* Bioinformatics, 2015. **31**(3): p. 405-412.
27. Tran-Nguyen, V.-K., C. Jacquemard, and D. Rognan, *LIT-PCBA: An unbiased data set for machine learning and virtual screening.* Journal of chemical information and modeling, 2020. **60**(9): p. 4263-4273.
28. Perez-Castillo, Y., et al., *CompScore: boosting structure-based virtual screening performance by incorporating docking scoring function components into consensus scoring.* Journal of chemical information and modeling, 2019. **59**(9): p. 3655-3666.
29. Wang, C. and Y. Zhang, *Improving scoring-docking-screening powers of protein–ligand scoring functions using random forest.* Journal of computational chemistry, 2017. **38**(3): p. 169-177.
30. Ye, W.-L., et al., *Improving docking-based virtual screening ability by integrating multiple energy auxiliary terms from molecular docking scoring.* Journal of chemical information and modeling, 2020. **60**(9): p. 4216-4230.
31. Chen, L., et al., *Hidden bias in the DUD-E dataset leads to misleading performance of deep learning in structure-based virtual screening.* PloS one, 2019. **14**(8): p. e0220113.
32. Pereira, J.C., E.R. Caffarena, and C.N. Dos Santos, *Boosting docking-based virtual screening with deep learning.* Journal of chemical information and modeling, 2016. **56**(12): p. 2495-2506.
33. Wallach, I., M. Dzamba, and A. Heifets, *AtomNet: a deep convolutional neural network for bioactivity prediction in structure-based drug discovery.* arXiv preprint arXiv:1510.02855, 2015.
34. Trott, O. and A.J. Olson, *AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading.* Journal of computational chemistry, 2010. **31**(2): p. 455-461.
35. Morris, G.M., et al., *AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility.* Journal of computational chemistry, 2009. **30**(16): p. 2785-2791.

36. Korb, O., T. Stützle, and T.E. Exner. *PLANTS: Application of ant colony optimization to structure-based drug design*. in *International workshop on ant colony optimization and swarm intelligence*. 2006. Springer.

37. Ruiz-Carmona, S., et al., *rDock: a fast, versatile and open source program for docking ligands to proteins and nucleic acids.* PLoS computational biology, 2014. **10**(4): p. e1003571.

38. Zhang, N. and H. Zhao, *Enriching screening libraries with bioactive fragment space.* Bioorganic & Medicinal Chemistry Letters, 2016. **26**(15): p. 3594-3597.

39. O'Boyle, N.M., et al., *Open Babel: An open chemical toolbox.* Journal of cheminformatics, 2011. **3**(1): p. 1-14.

40. Altschul, S.F., et al., *Basic local alignment search tool.* Journal of molecular biology, 1990. **215**(3): p. 403-410.

41. Truchon, J.-F. and C.I. Bayly, *Evaluating virtual screening methods: good and bad metrics for the "early recognition" problem.* Journal of chemical information and modeling, 2007. **47**(2): p. 488-508.