Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Accelerating molecular docking using machine learning methods

Abdulsalam Yazıd Bande
Istanbul Technical University
Sefer Baday
badays@itu.edu.tr

Istanbul Technical University

Research Article

Keywords: Virtual Screening, Molecular Docking, Protein-ligand interaction, Machine Learning

Posted Date: November 8th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3099459/v2

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Version of Record: A version of this preprint was published at Molecular Informatics on June 8th, 2024.
See the published version at https://doi.org/10.1002/minf.202300167.

Abstract

Virtual screening (VS) is one of the well-established approaches in drug discovery which speeds up the search for a bioactive molecule and, reduces costs and efforts associated with experiments. VS helps to narrow down the search space of chemical space and allows selecting fewer and more probable candidate compounds for experimental testing. Docking calculations are one of the commonly used and highly appreciated structure-based drug discovery methods. Databases for chemical structures of small molecules have been growing rapidly. However, at the moment virtual screening of large libraries via docking is not very common. In this work, we aim to accelerate docking studies by predicting docking scores without explicitly performing docking calculations. We experimented with an attention based long short-term memory (LSTM) neural network for an efficient prediction of docking scores as well as other machine learning models such as XGBoost. By using docking scores of a small number of ligands we trained our models and predicted docking scores of a few million molecules. Specifically, we tested our approaches seven datasets that were produced in-house drug discovery studies. In one of the targets, by training only 7000 molecules we predicted docking scores for 3 million molecules with R^2 (coefficient of determination) of 0.84. We designed the system with ease of use in mind. All the user needs to provide is a csv file containing smiles and their respective docking scores, the system then outputs a model that the user can use for the prediction of docking score for a new molecule.

Introduction

Finding a candidate molecule that can exhibit desired bioactivity to a target protein of interest is in the center of early-drug development process. Usually, the search for selecting suitable molecules among vast chemical space is initiated with virtual screening (VS). When three-dimensional structure of the target protein is known experimentally or modelled by molecular modelling methods, structure-based virtual screening (SBVS) methods are applied. Docking calculations are one of the commonly used and highly appreciated structure-based drug discovery methods. Typically, a docking program produces a number of binding pose and ranks these binding poses based on a scoring method used. Generally, in VS applications, a library of molecules is screened using docking programs and top-ranked molecules are selected for experiments.

Over the last years, the amount of molecular data in databases has skyrocketed and methods such as molecular docking for VS just cannot keep up with huge amount of data. There are a very good examples of the screening of ultra-large chemical libraries. However, usual applications of large library compound docking studies are limited to a few million molecules at best. Even docking of a few million molecules can be challenging. Despite the fact that docking of one molecule is quite fast and can be done on a regular laptop, docking of a few million molecules requires a good computing infrastructure such as a workstation or the access to a HPC center. Even in the presence of an appropriate computing facility, preparation of input files, execution of the computation, storage and analysis of the results need some

effort and expertise. Moreover, in some cases, docking of large libraries can be hampered by the high license cost of paid docking programs. For example, one needs to pay more to run a docking calculation using several CPUs compared to a single CPU calculation using Schrödinger software.

Due the aforementioned challenges, screening of chemical libraries containing millions of molecules is not an ordinary task in virtual screening studies. Thus, strategies for enabling scientist to carry out docking calculations more efficiently are needed. The need of performing docking screens more efficiently led to studies utilizing machine learning (ML) methods. In a recent work, named as Lean-Docking, linear support regressors were trained on docking scores of %25 of a chemical library and prediction of the docking scores for the rest %75 of the chemical library was performed for several target proteins [1]. Predictions were done on molecule sets containing less than 100.000 molecules. A work by Jastrzeb□ski et al., named as “Emulating Docking”, converts the docking pose prediction problem into the prediction of interaction fingerprints as a classification problem. They experimented with multiple G protein-coupled receptors as targets and CYP enzymes on ligand set containing experimentally known active and inactive molecules for these receptors, and randomly chosen 500.000 molecules from the ZINC database. Despite reaching 0.8 AUC score to retain known actives, reported correlation of the predicted docking score and calculated docking is around 0.6 [9]. In another work, Svensson et al. [19] uses an iterative approach for high throughput screening. The way the system works is first by selecting a group of initial molecules for screening, then the result is used in categorizing the remaining of the molecules. The authors applied their methodology on 41 different target proteins from DUD-E [20] and reported that by just using 9.4% of the whole molecular data, 57% of the remaining active compounds could be identified. The ligand set for each target protein contains a number of active and inactive molecules, in total varying from three to fifty thousand. Another iterative approach is “Deep-Docking” by Gentile et al. [7] that uses a QSAR model as the input and predicts docking scores. They used the FRED docking software and data from the ZINC15 library to calculate docking scores of 1.36 billion molecules on six different target proteins. They could catch %60 of the active molecules using docking score of a few hundred thousand molecules. The work by Yanagisawa et al. Follows a different approach where they identify common fragments in a ligand library and perform docking of the identified fragments instead of the docking of all the molecules in the ligand library. To predict a docking score of a molecule they combine the docking score of the fragments in that molecule. Based on the tests on three targets from DUD-E they could get ~0.6 correlation score between the predicted docking score and calculated docking score.

In this work, multiple ML models were experimented to predict the docking scores of a ligand for a target protein without explicitly performing docking calculations. The goal is to find a regressor that can give accurate docking scores of the ligands in a chemical library to a target protein using the information obtained from explicit docking of just a handful of molecules. One of the machine learning models used

is neural network based which is an LSTM [17]. LSTMs are usually used for sequence data in Natural Language Processing (NLP) applications like speech recognition. The LSTM model used in this work is coupled with an attention mechanism [5] for the neural network model to further extract useful features from an input ligand data. The LSTM is implemented using the popular python machine learning framework Pytorch [12]. Other models experimented with are XGBoost which is implemented using the XGBoost [2] python library, decision tree regressor and stochastic gradient descent which are from the scikit-learn [13] python library.

All the experiments in this work rely on datasets that were all produced in-house. Furthermore, the experiments were done with different training set sizes to see how increase in training data size influences prediction of docking scores. For the featurization of molecules, SMILES of molecules were converted into Extended Connectivity Circular Fingerprints (ECFPs) [16], Molecular ACCess System (MACCSS) [6], and One Hot Encoding. In addition, we tested the performance of the models using these fingerprints alone and combined.

Methods

Materials and Methods

Receptors and Ligands used in docking calculations

The docking scores reported in this work were produced in-house drug discovery research projects. Two different ligand sets were screened on several target proteins using AutoDock-Vina program [18]. 2D information of ligands were downloaded from the ZINC15 database as SDF format. The ligand set 1 was obtained from an extended version of lead-like ready to purchase clean tranche of ZINC15. In this set, molecular weights of the molecules are between 200-425 daltons and LogP values range from -1 to 3.5. The ligand set 1 roughly contains 3.4 million molecules. The ligand set 1 was screened on GTP binding site of Drp1 GTPase, dantrolene binding site of Ryanodine Receptor and MiD49 interaction interface of Drp1 GTPase using AutoDock-Vina program. The Ligand set 2 was compiled by randomly selecting 400.000 molecules from ~6.2 million molecules that is under drug-like clean ready to purchase tranche of ZINC15. These 400.000 molecules were docked to the angiotensin-converting enzyme (ace)/ spike-ace interaction interface, spike/ spike-ace interaction interface, SARS-CoV-2 non-structural protein 16 (Nsp16) /SARS-CoV-2 non-structural protein 10 (Nsp10) interaction interface, and SARS-CoV-2 non-structural protein 16 (Nsp16) S-adenosylmethionine binding site. The exact number of docking scores might change for each target due to the fact that sometimes in docking calculations some of the molecules might not be docked because of technical problems. For example, in RyR2 we have 1.8M

molecules that were docked into the right binding site and the rest of the molecules were docked in another binding site. Table 1 summarizes the receptors and molecules docked to these receptors.

Table 1 Targets and their aliases

Target protein	Alias	Number of available docking scores	Ligand set
Drp1 GTPase/ GTP binding site	Drp1_GTPase	3 500 000	Ligand set1
Ryanodine Receptor/ dantrolene binding site	RyR2	1 756 665	Ligand set1
Drp1 GTPase/ MiD49 interaction interface	Drp1_MiD49	3 334 534	Ligand set1
angiotensin-converting enzyme (ace)/ spike-ace interaction interface	ace	400 000	Ligand set2
spike/ spike-ace interaction interface	spike	400 000	Ligand set2
SARS-CoV-2 non-structural protein 16(/SARS-CoV-2 non-structural protein 10 interaction interface	Nsp16-Nsp10	400 000	Ligand set2
SARS-CoV-2 non-structural protein 16 S-adenosylmethionine binding site	Nsp16-sam	400 000	Ligand set2

Descriptors

Simplified molecular-input line-entry system (SMILES) is used for representing ligands. Using SMILES descriptors for ligands were produced. Machine and deep learning algorithms were experimented with different descriptors that were created using the DeepChem library [15]. The descriptors used in this work are Extended Connectivity Circular Fingerprints (ECFPs) [16], MACCS (Molecular ACCess System) [6] and One Hot Encoding. ECFPs uses an iterative approach to describe a molecule, it uses a combination of atomic properties such as atomic number and hashing in the description process. The MACCS descriptor starts a 167-bit vector of zeros, the vector is then populated with *ones* for any position that satisfy a molecular property such as Nitrogen availability. The last descriptor used was One Hot Encoding, this also works like the previous descriptor. Here, each character of the SMILES is transformed into a 35-bit vector: *one* is placed at the index corresponding to the character's position in the predefined set of characters ['#', ')', '(', '+', '-', '/', '1', '3', '2', '5', '4', '7', '6', '8', '=', '@', 'C', 'B', 'F', 'I', 'H', 'O', 'N', 'S', '[', ']', '\', 'c', 'l', 'o', 'n', 'p', 's', 'r'], with *zeroes* elsewhere, and the last bit is set to '1' if the input character is not in the list. The default maximum length for the SMILES string is 100; if the input is shorter than this,

the output is padded with zeros to ensure a consistent output size. Table 2 shows the information on the descriptors used.

Table 2 Descriptors used and respective number of features

Descriptor Identifier	Feturizer	Total Features
d1	One Hot Encoding	3500
d2	MACCS	167
d3	CircularFingerprint+ One Hot Encoding + MACCS	4691

Selecting a Percentage of Molecules for Training

Since we aim to use only a small size of docking results to predict the docking scores of a large library, we experimented with various size of training set. It is important to understand how much data is sufficient to train a machine learning model that could be used to screen a large library of molecules. Thus, the training set size is chosen from the set {7000, 10000, 20000, 50000, 100000, 350000}. Then 5-fold CV is used in training where each training size is multiplied by five so that training is performed on one fifth of the data and the CV evaluation on the remaining four fifths. For example, for training on 7000 molecules, 35 000 (5 x 7000) molecules are randomly selected, and, in each fold, 7000 molecules are used in training while the remaining 28 000 molecules are used in evaluation. This process works like the opposite of traditional CV. An advantage that CV provides is it ensures a good estimate of a model's prediction performance. For final testing after the CV process, a separate testing set having a size depending on the training set size is used. To maintain consistency, the final testing size is delimited to 3 000 000 for target proteins when applicable. In the case of a training size of 350 000, 1 750 000 molecules become in the CV set. Thus, when training set is 350 000 , then we have around 1 800 000 molecules in the final test set (3 500 000 - 1 750 000). The ligand set one training and testing set sizes is summarized in Table S1.

For the ligand set two, due to the dataset size limitations, the training set size is chosen from the set {7000, 10000, 50000} and except for the size of 7000 which uses 5-fold CV, 3-fold CV is used in the remaining sizes. The ligand set two training and testing set sizes are summarized in Table S1.

Machine learning models

In order to get a better result, several machine learning models were trained. One of the models experimented on was a bidirectional LSTM with an attention mechanism [5]. LSTM is a type of recurrent neural network (RNN) designed to learn and retain long-term dependencies in sequential data effectively. Bidirectional LSTMs process the input sequence in both forward and backward directions, allowing the model to capture context from both past and future time steps. The attention mechanism enhances the bidirectional LSTM model by dynamically focusing on relevant input parts during decoding, improving its handling of long sequences and complex relationships. In addition, a vanilla neural network is incorporated, which takes the output from the attention-enhanced bidirectional LSTM and makes the final prediction.

After performing a hyperparameter tuning, the loss function settled on was mean squared error (MSE). The optimization function used was Adam optimizer with a learning rate of 0.001. During training, a batch size of 128 was used. A higher batch size could have been used but 128 is just enough to be used on a home machine as ease of use was considered. The experiments for this project were conducted on a Linux-based machine equipped with 126 GB of memory and a multi-core processor consisting of 28 cores (Intel Xeon E5-2680 v4). Pytorch [12] is used in training the LSTM. Other models used are from the scikit-learn [13] library. These models are decision tree regressor and stochastic gradient descent regression. XGBoost Regressor from the XGBoost library was also experimented with.

Results and Discussions

Performance of models

The results in Table 3 and Table 4 show performances of the models tested for each target protein in terms of R^2 and MAE scores respectively. Note that in these two tables, values show scores for the tested molecules where the size of the test set varies by the training set size and target protein (exact number of molecules in the test sets can be found Table S1). When all results are analyzed together it seems that LSTM achieves the best score among other models tested. LSTM is followed by SGDR, XGBoost and decision tree. When only 7K docking scores used in the training phase, we could reach average 0.82 R^2 for three targets in the ligand set one and 0.69 R^2 for four targets in ligand set two. The discrepancy between the results for the ligand set one and set two arises from the nature of the targets and ligand molecules. The target proteins in the ligand set one have well-formed binding pockets whereas the majority of the targets (except for nsp-sam) in the ligand set two does not have a binding pocket. The binding sites of the targets in the ligand set two are protein-protein interface sites which are very challenging to be targeted by small molecules. We can deduce the fact that we can reach a structure-activity (let's presume docking score shows the activity) relationship model on the targets in the ligand set one where the binding pockets allow molecules having a distinct chemical feature. In

contrast, a higher chemical diversity due to more shallow binding sites for the targets in the ligand set two does not give a SAR model as strong as in the case for the targets in the ligand set one. We tested models' performances with different size of the training sets. In fact, using more ligands in the training set increase the performances of the models slightly. This implies that ligand set around 7K molecule is good enough to establish a SAR model. In order to get a feeling of performance, scatter plots for predicted and calculated docking scores for each model trained with 7000 ligands are shown on Fig. 2.

Table 3 R^2 scores for predictions made by models with 7k, 50k and 350k training sizes

Results for the Ligand set one												
	Training on 7k ligands				Training on 50k ligands				Training on 350k ligands			
Target	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost
Drp1_GTPase	0.83	0.53	0.81	0.77	0.85	0.60	0.83	0.81	0.86	0.67	0.84	0.83
RyR2	0.84	0.53	0.82	0.78	0.86	0.61	0.84	0.83	0.90	0.69	0.86	0.85
Drp1_MiD49	0.79	0.43	0.75	0.70	0.80	0.50	0.77	0.76	0.83	0.58	0.79	0.78
Avg.	0.82	0.50	0.79	0.75	0.84	0.57	0.81	0.80	0.86	0.65	0.83	0.82
Results for the Ligand set two												
	Training on 7k ligands				Training on 50k ligands				Training on 350k ligands			
Target	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost
ace	0.59	0.00	0.57	0.51	0.58	0.02	0.59	0.53	0.62	0.09	0.64	0.61
spike	0.70	0.24	0.70	0.67	0.67	0.26	0.72	0.69	0.70	0.33	0.75	0.73
nsp	0.71	0.17	0.68	0.63	0.71	0.16	0.70	0.63	0.73	0.27	0.75	0.71
nsp-sam	0.77	0.37	0.78	0.74	0.78	0.38	0.79	0.75	0.80	0.46	0.81	0.79
Avg.	0.69	0.20	0.68	0.64	0.69	0.21	0.70	0.65	0.71	0.29	0.74	0.71

Table 4 MAE scores for predictions made by models with 7k, 50k and 350k training sizes

Results for the Ligand set one												
	Training on 7k ligands				Training on 50k ligands				Training on 350k ligands			
Target	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost
Drp1_GTPase	0.30	0.50	0.29	0.32	0.28	0.46	0.27	0.29	0.27	0.41	0.27	0.28
RyR2	0.28	0.48	0.28	0.31	0.26	0.44	0.26	0.27	0.23	0.38	0.25	0.25
Drp1_MiD49	0.23	0.38	0.22	0.24	0.23	0.35	0.21	0.22	0.20	0.32	0.21	0.21
Avg.	0.27	0.45	0.26	0.29	0.26	0.42	0.25	0.26	0.23	0.37	0.24	0.25
Results for the Ligand set two												
	Training on 7k ligands				Training on 50k ligands				Training on 350k ligands			
Target	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost	LSTM	decisio-n tree	SGDR	XGBo-ost
ace	0.33	0.52	0.34	0.36	0.33	0.51	0.33	0.35	0.31	0.49	0.31	0.32
spike	0.28	0.44	0.27	0.29	0.29	0.44	0.27	0.28	0.27	0.41	0.25	0.26
nsp	0.30	0.51	0.31	0.34	0.30	0.51	0.30	0.34	0.29	0.47	0.27	0.30
nsp-sam	0.32	0.53	0.31	0.34	0.31	0.53	0.31	0.33	0.30	0.49	0.29	0.30
Avg.	0.31	0.50	0.31	0.33	0.31	0.50	0.30	0.33	0.29	0.46	0.28	0.30

Comparison of the chemical diversity of the molecules

When R^2 scores are considered, most of the results for the targets in the ligand set two are comparingly lower than that of the targets in dataset one. The difference in the results can be attributed to differences in the binding site of the targets and molecular diversity of the ligand sets. Thus, a more detailed comparison of the diversity of the molecules is performed. To compare the ligand sets, an analysis for the distribution of fingerprint similarity of the molecules in each ligand set was performed (Fig 3). The fingerprint similarity was calculated in terms of average, maximum and minimum distance frequencies. For example, in the case of distribution of average distance frequencies, for each smile of a molecule in the ligand set one, the fingerprint similarity is computed against all the other smiles in the ligand set one, then the results are then averaged. Furthermore, computing similarities between a SMILE and all others for target one is inefficient as the complexity is $O(n^2)$, where n is the number of molecules. Thus, from the ligand set one only 400 000 molecules were sampled, this is because we have 400 000 molecules in the ligand set two. Distribution plots for the minimum fingerprint similarity and average fingerprint similarity suggests that chemical diversity is higher in the ligand set two compared to the ligand set one.

Computation time analysis

This work allows you to significantly reduce experiment times for VS of a large molecule library. The ease of use of the models in this work provides a great advantage for who do not have access to a HPC center or a workstation. Extensive data on the computation time for training and testing the models for one of the target proteins (Drp1_GTPase) are given in Table S2. The training of the model with 7k model took 1.9 minutes and prediction the docking scores of 3 million molecules 64.7 minutes. Though other models run faster than LSTM models, we get better performance in LSTM models. Training 50K and 350K docking scores took 32.6 and 245.8 minutes. So, it is quite feasible to build a model with 350K molecules and make predictions. However, the actual bottleneck is to get docking scores of 350K molecules. These CPU time results obtained using a computing node having 28 CPU cores (Intel Xeon CPU E5-2680 v4 2.3 GHz). Additional computational time tests were also performed on a laptop computer equipped with 16GB of RAM and a 2.3 GHz 8-Core Intel Core i9 processor. This hardware configuration was applied to run LSTM models under a single-fold cross-validation setting. Training with 7k molecules can be done in approximately 10 minutes and prediction of 100 000 molecules can be done in 4 minutes. As a result, it is very feasible to apply our methodology on a regular laptop or desktop computer.

Comparison of Descriptors used

The selection of the descriptors was not trivial. Not only we needed to find good descriptors, but we also wanted combine descriptors that lead to a good result. That is why we trained our models with different descriptors. When results given for the LSTM model trained with 7K molecules in Table 5 are analyzed, it is seen that the descriptor 3, concatenation of Circular Fingerprint, One Hot Encoding and MACCS, gave the best performance for all the targets.

Table 5 Comparison of the performances (R^2 scores) of the LSTM models trained on 7K molecules with different descriptors

Target	d1	d2	d3
Drp1_GTPase	0.79	0.73	0.83
RyR2	0.78	0.72	0.84
Drp1_MiD49	0.72	0.65	0.79
ace	0.48	0.50	0.59
spike	0.63	0.56	0.70
nsp	0.63	0.55	0.71
nsp-sam	0.69	0.58	0.77

Conclusion

In this work we present a methodology to accelerate molecular docking of large chemical libraries without actually performing docking calculations on the whole molecule set. Here, we performed docking calculations only on a very small subset of molecules, for example 7000 molecules. Then built machine learning models to predict docking scores of 3 million molecules. This approach saves time and money to screen 3 million molecules with explicit docking calculations. We built various machine learning models on docking scores obtained on seven different targets that were studied in-house drug discovery projects. We could reach 0.84 R^2 for one the targets. The performance of the models varies by the diversity of the chemical libraries and the binding site properties. Yet, our approach can really help the researchers who want to explore large chemical libraries through docking calculations with less time, effort and resources.

Declarations

Supplementary Information The supplementary material can be found at Supplementary Material.docx file.

Funding None

Acknowledgements

Computing resources used in this work were provided by the National Center for High Performance Computing of Turkey (UHeM).

Code availability The source code can be found at https://github.com/abdulsalam-bande/swifty .The authors can also be contacted in case of any issue faced from the source code.

Data availability Csv Files containing docking data for some of targets used in the experiments can be found at https://drive.google.com/drive/folders/1OpcIsGK3fdTOW9NLGi9qFhPUCoUEy8-Z?usp=share_link

Competing interests The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Berenger, F., Kumar, A., Zhang, K. Y., & Yamanishi, Y. (2021). Lean-docking: exploiting ligands' predicted docking scores to accelerate molecular docking. Journal of Chemical Information and Modeling, 61(5), 2341-2352
2. Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining (pp. 785-794)
3. Cherkasov, A., Ban, F., Li, Y., Fallahi, M., & Hammond, G. L. (2006). Progressive docking: a hybrid QSAR/docking approach for accelerating in silico high throughput screening. Journal of medicinal chemistry, 49(25), 7466-7478
4. Chupakhin, V., Marcou, G., Baskin, I., Varnek, A., & Rognan, D. (2013). Predicting ligand binding modes from neural networks trained on protein–ligand interaction fingerprints. Journal of chemical information and modeling, 53(4), 763-772
5. Bahdanau, D., Cho, K., & Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473
6. Durant, J. L., Leland, B. A., Henry, D. R., & Nourse, J. G. (2002). Reoptimization of MDL keys for use in drug discovery. Journal of chemical information and computer sciences, 42(6), 1273-1280
7. Gentile, F., Agrawal, V., Hsing, M., Ton, A. T., Ban, F., Norinder, U., ... & Cherkasov, A. (2020). Deep docking: a deep learning platform for augmentation of structure based drug discovery. ACS central science, 6(6), 939-949
8. Gorgulla, C., Boeszoermenyi, A., Wang, Z. F., Fischer, P. D., Coote, P. W., Padmanabha Das, K. M., ... & Arthanari, H. (2020). An open-source drug discovery platform enables ultra-large virtual screens. Nature, 580(7805), 663-668
9. Jastrzebski, S., Szymczak, M., Pocha, A., Mordalski, S., Tabor, J., Bojarski, A. J., & Podlewska, S. (2020). Emulating docking results using a deep neural network: a new perspective for virtual screening. Journal of Chemical Information and Modeling, 60(9), 4246-4262
10. Morris, P., St. Clair, R., Hahn, W. E., & Barenholtz, E. (2020). Predicting binding from screening assays with transformer network embeddings. Journal of Chemical Information and Modeling, 60(9), 4191-4199
11. Yanagisawa, K., Komine, S., Suzuki, S. D., Ohue, M., Ishida, T., & Akiyama, Y. (2017). Spresso: an ultrafast compound pre-screening method based on compound decomposition. Bioinformatics, 33(23), 3836-3843

12. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32
13. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. the Journal of machine Learning research 12
14. Quiroga, R., & Villarreal, M. A. (2016). Vinardo: A scoring function based on autodock vina improves scoring, docking, and virtual screening. PloS one, 11(5), e0155183
15. Ramsundar, B., Eastman, P., Walters, P., & Pande, V. (2019). Deep learning for the life sciences: applying deep learning to genomics, microscopy, drug discovery, and more. O'Reilly Media
16. Rogers, D., & Hahn, M. (2010). Extended-connectivity fingerprints. Journal of chemical information and modeling, 50(5), 742-754
17. Sak, H., Senior, A. W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling
18. Trott, O., & Olson, A. J. (2010). AutoDock Vina: improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. Journal of computational chemistry, 31(2), 455-461
19. Svensson, F., Norinder, U., & Bender, A. (2017). Improving screening efficiency through iterative screening using docking and conformal prediction. Journal of chemical information and modeling, 57(3), 439-444
20. Mysinger, M. M., Carchia, M., Irwin, J. J., & Shoichet, B. K. (2012). Directory of useful decoys, enhanced (DUD-E): better ligands and decoys for better benchmarking. Journal of medicinal chemistry, 55(14), 6582-6594.

Figures

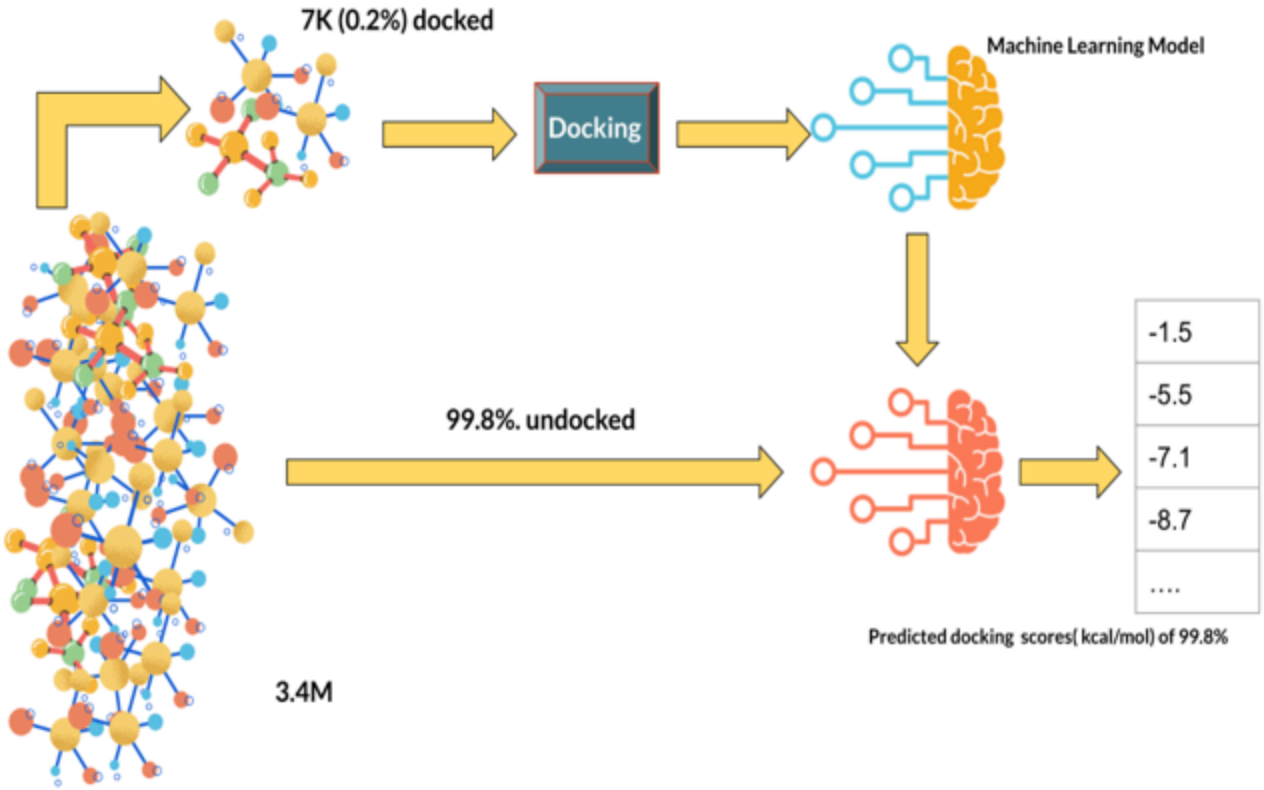


Figure 1

Flow diagram of the work using a training size of 7k

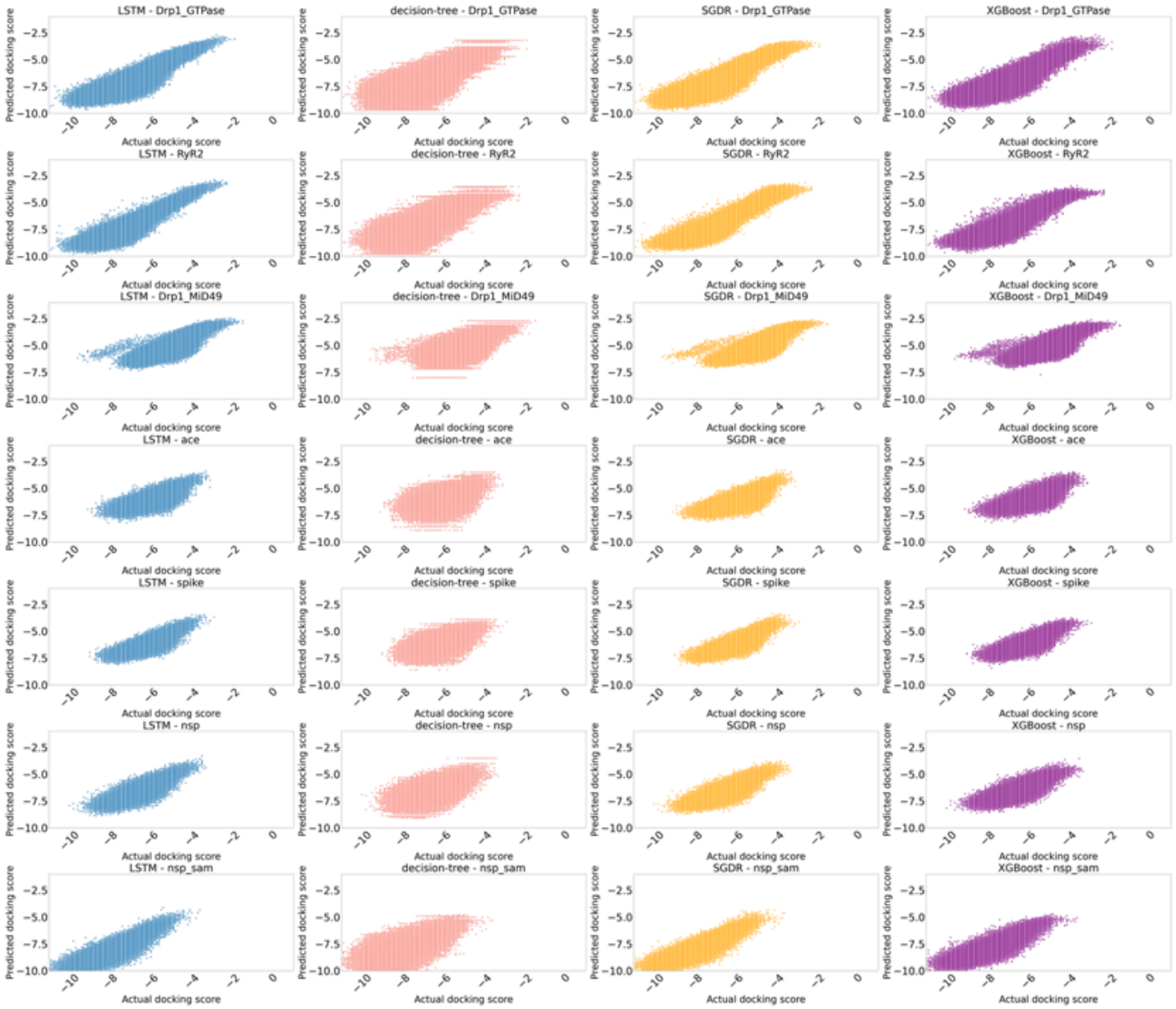


Figure 2

Scatter Plot of predicted docking scores by ML models trained on 7K molecules vs actual docking scores calculated using Vina program for each target protein.

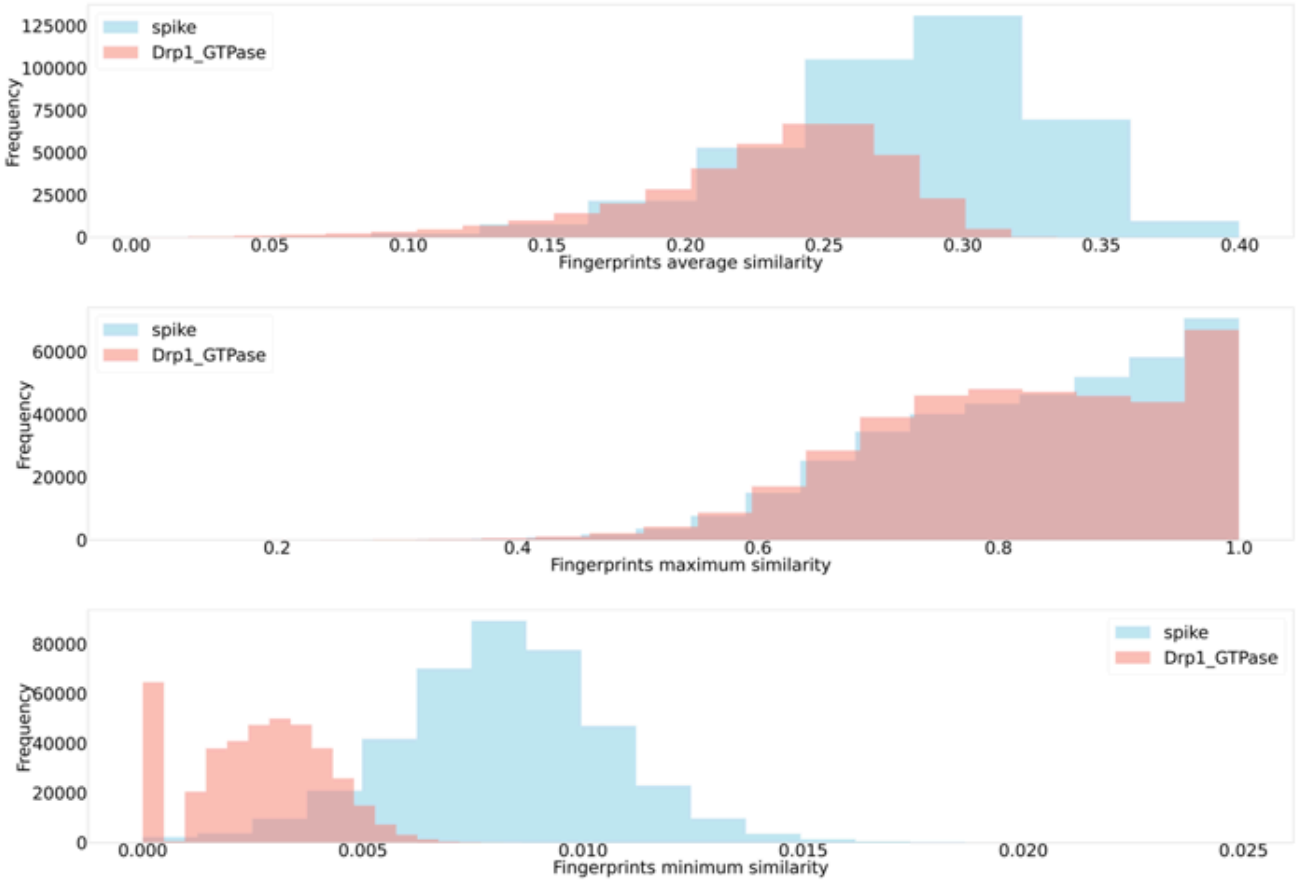


Figure 3

Analysis of the diversity of molecules docked to spike and Drp1_GTPase. The x label represents the computed type of fingerprint similarity, and the y axis represents the frequency

Supplementary Files

This is a list of supplementary files associated with this preprint. Click to download.

- SupplementaryMaterial.docx