Docking-informed machine learning for kinome-wide affinity prediction

Schifferstein, J.; Bernatavicius, A.; Janssen, A.P.A.

JCIM JOURNAL OF CHEMICAL INFORMATION AND MODELING

Open Access

pubs.acs.org/jcim

Article

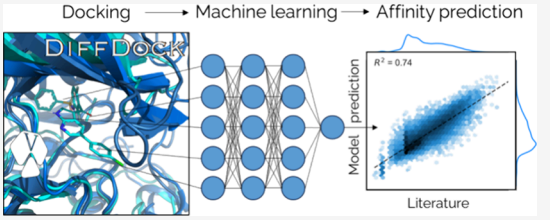
Docking-Informed Machine Learning for Kinome-wide Affinity Prediction

Jordy Schifferstein, Andrius Bernatavicius, and Antonius P.A. Janssen*

Cite This: J. Chem. Inf. Model. 2024, 64, 9196−9204

Read Online

ABSTRACT: Kinase inhibitors are an important class of anticancer drugs, with 80 inhibitors clinically approved and >100 in active clinical testing. Most bind competitively in the ATP-binding site, leading to challenges with selectivity for a specific kinase, resulting in risks for toxicity and general off-target effects. Assessing the binding of an inhibitor for the entire kinome is experimentally possible but expensive. A reliable and interpretable computational prediction of kinase selectivity would greatly benefit the inhibitor discovery and optimization process. Here, we use machine learning on docked poses to address this need. To this end, we aggregated all known inhibitor-kinase affinities and generated the complete accompanying 3D interactome by docking all inhibitors to the respective high-quality X-ray structures. We then used this resource to train a neural network as a kinase-specific scoring function, which achieved an overall performance (R^2) of 0.63−0.74 on unseen inhibitors across the kinome. The entire pipeline from molecule to 3D-based affinity prediction has been fully automated and wrapped in a freely available package. This has a graphical user interface that is tightly integrated with PyMOL to allow immediate adoption in the medicinal chemistry practice.



INTRODUCTION

Protein kinases are one of the main protein families targeted by anticancer drugs, with 80 approved drugs and around 150 in clinical testing.[1] However, current FDA-approved kinase inhibitors are designed to target only a few percent of the entire protein family.[2] The so-far untargeted kinases, thus, offer great opportunities for the development of novel molecular therapies.

The chances of success for any drug greatly depend on two parameters: affinity of the drug for the intended target protein, and selectivity over the rest of the protein family. Off-target activity is often the main cause of (pre)clinical toxicity, and side-effects in general.[3] This issue is particularly pressing for kinase inhibitors, as these in most cases target the ATP-binding site of the protein, which is highly conserved across this large protein family.[4] This leads to many kinase inhibitors potently binding to many family members, sometimes inhibiting as much as 70% of all kinases.[5] Determining the specificity of an inhibitor over all ±500 kinases is experimentally feasible, but is prohibitively expensive in terms of time, material and funds to perform on a routine basis.

In recent years, various computational methods of predicting kinase inhibitor selectivity have thus been developed.[6−8] Approaches vary from "classical" protein structure-based techniques such as molecular docking to machine learning approaches such as Quantitative Structure Activity Relationship (QSAR) studies. The revolution of artificial intelligence (AI) has not gone unnoticed in this field, and e.g. AlphaFold[9] will have a tremendous impact in the coming years, giving direct access to structures for all proteins. Structure-based methods typically rely on either classical physics-based scoring functions to "score" a generated protein−ligand complex. More recently, machine learning-based scoring functions such as RFScore have reached state-of-the-art performance.[10−12] These scoring functions were trained on experimental data sets such as the PDBbind, offering a relatively broad set of protein-inhibitor complexes and their bioactivity data.[13] Some kinase specific tools have also appeared in recent years. KinomeX, a multitask classification DNN trained only on ligand molecular fingerprints can be classified as one of the QSAR based models.[14] It was available online as a service but did not provide the option of installing locally for proprietary data applications. KinomeX has more or less been superseded by KinomeMETA of the same research group.[15] KinomeMETA uses a GNN architecture on ligand structures to predict the Boolean activity on >600 kinases. KinomeMETA is available as an online service, but no local variant is provided. Neither of these models has an obvious way to inspect the prediction origin. Closer to what we envision is ProfKin, a structure-based tool that compares docked poses to 4219 experimentally determined kinase-ligand complexes.[16] The interaction fingerprint and similarity scoring is used to provide expected kinase

Received: July 17, 2024
Revised: October 31, 2024
Accepted: December 2, 2024
Published: December 10, 2024

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

Figure 1. Kinase activity data set | A) Distribution of kinase inhibition values reported per kinase group; B) Distribution of the reported inhibitory values; C) Number of pChEMBL values for unique kinases reported per kinase inhibitor, *i.e.*, against how many kinases was a compound tested; D) Number of reported inhibitors per kinase, *i.e.*, how many compounds were tested for a kinase; E) Overview of the workflow of the work in this paper.

targets for a given ligand. This tool was also only available as a server.

We set out to develop a fully automated docking-based tool akin to ProfKin, but with the aim of predicting binding affinities for kinases. As it is generally accepted that pose finding for most docking algorithms is very good,[17] we envisioned that a large docking-based protein-inhibitor data set for which biochemical data is known should also function as a basis for training a scoring function. We demonstrated this approach by generating protein-inhibitor complexes for all kinase inhibitors in the Papyrus data set,[18] a large aggregation of literature binding data, for kinases of which a high-quality experimental protein structure is available in the KLIFS database, a kinase specific mirror of the PDB.[19,20] We used two docking algorithms: Autodock VinaGPU[21] and DiffDock.[22] This generated database is then used to train a multilayered Neural Network as scoring function, that shows excellent performance on bioactivity predictions for unseen inhibitors.

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

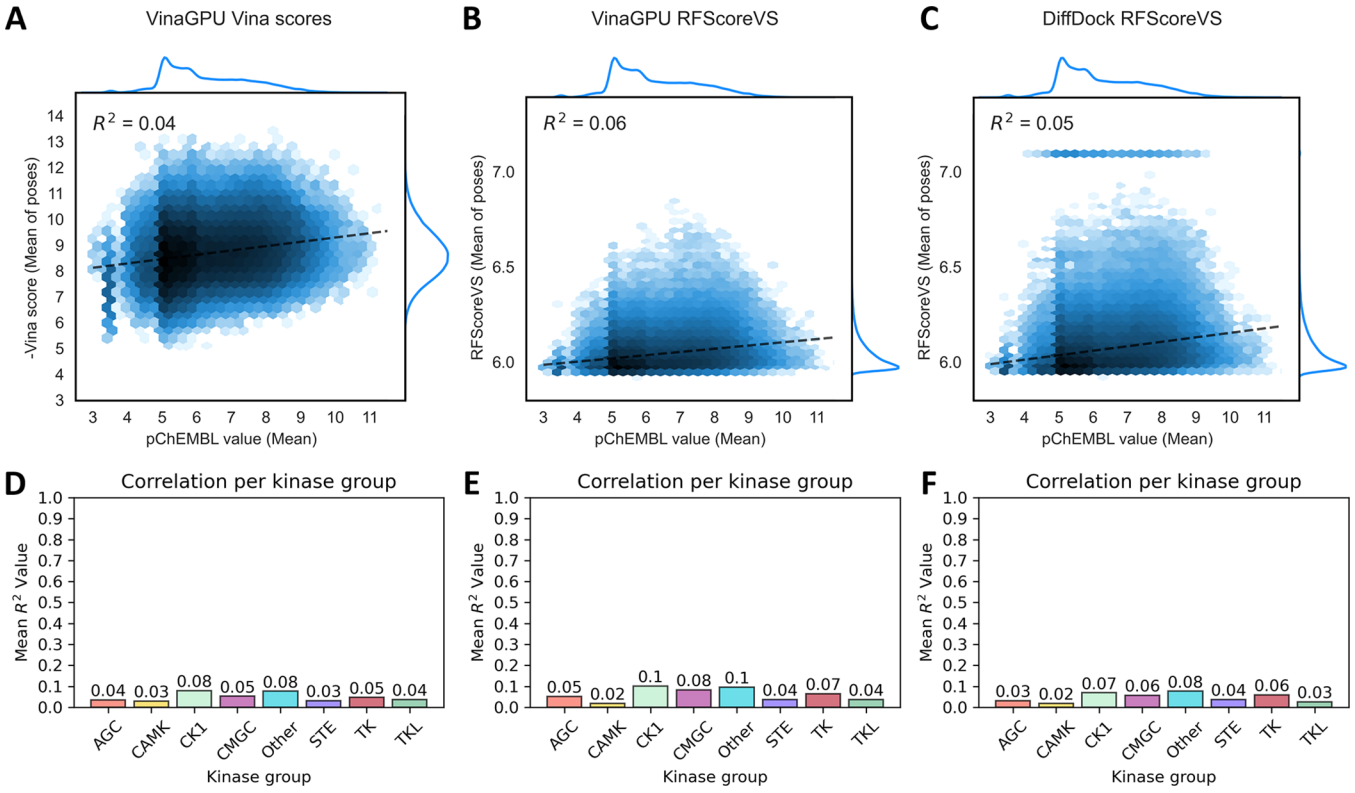


Figure 2. Correlation of Vina and RFScoreVS scoring functions with Papyrus pChEMBL data | Predicted affinity values vs literature values displayed as logarithmic hexbin plots, as based on the -Vina score for all VinaGPU poses (A), RFScoreVS for all VinaGPU poses (B), RFScoreVS for all DiffDock poses (C), R^2 calculated per kinase and aggregated per kinase group for Vina scores (D), RFScoreVS for VinaGPU poses (E) and RFScoreVS for DiffDock poses (F).

The automated workflow has been wrapped in an easily installable Docker container[23] with a convenient PyMOL Graphical User Interface (GUI) plugin, allowing broad access to the methodology.

RESULTS

Extracting Literature Biochemical and Structural Data. To generate our desired docking-based training data set, we first needed to select all kinases of which we have a high-quality experimental structure. As a source of well curated and annotated kinase protein structures available in the Protein Data Bank we used the KLIFS database. These structures were filtered based on their resolution (≤2.5 Å) and KLIFS quality metric (≥8). We selected the best of each of the four possible combinations of DFG-in/out and α-C helix in/out as annotated in the KLIFS database. In total, this led to 345 protein structures for 226 unique kinases.

Next, we extracted all reported inhibitory activities for these kinases in the Leiden Papyrus data set, a curated resource combining data from resources such as ChEMBL, PubChem and others. We chose to indiscriminately use pIC_{50}, pK_i and pK_d values, collectively from hereon: pChEMBL values. We filtered the compounds to entail only the more drug-like small molecules using quite lax criteria (MW ≤ 750, NumHBD ≤ 10, NumHBA ≤ 15, Rotatable Bonds ≤15), which should have reasonable chance to dock well and form a representative training set for real world medicinal chemistry applications. An overview of the resulting physicochemical properties and chemical diversity is plotted in Figure S1.

This procedure led to a completed data set of in total 205,190 affinity values for 87,951 unique compounds against 226 unique kinases. A summative view of the workflow and complete resulting data set is depicted in Figure 1 and Figure S1.

Large Scale Molecular Docking Using Open Source Software. We set up an automated docking pipeline to generate a set of docking poses for all inhibitor-protein pairs in the created data set (Figure 1E). To this end, inhibitors were prepared for docking using an RDKit[24] pipeline, which enumerates potential stereo- and double bond isomers, and generates a 3D conformer. For each protein structure, a binding site was defined using PyVOL to guide the VinaGPU docking algorithm.[25] All prepared isomers were consecutively docked in the known targets of these inhibitors using two docking algorithms: Autodock VinaGPU and the diffusion-based DiffDock algorithm (version of December 2022).

For all compound-protein structure pairs, a maximum of 5 poses were generated. The poses were filtered for excessive atomic overlap based on a tailored clash-score (see Methods and Figure S2) to get rid of unphysical poses generated, a problem especially prevalent in DiffDock generated poses. For the inhibitor-kinase pairs in our data set for which an experimental pose has been determined (only ±0.2% of the 205,000), the root mean squared deviation (RMSD) was calculated for both docking algorithms. Median ± absolute deviation for DiffDock and VinaGPU were 1.296 ± 0.587 Å and 5.659 ± 4.177 Å, respectively.

The results of this large-scale docking project were aggregated and have been made available in an SQLite database that holds all activities, compounds, isomers, protein information, kinase structure information and all poses for both docking tools. A simplified schema of this database with

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

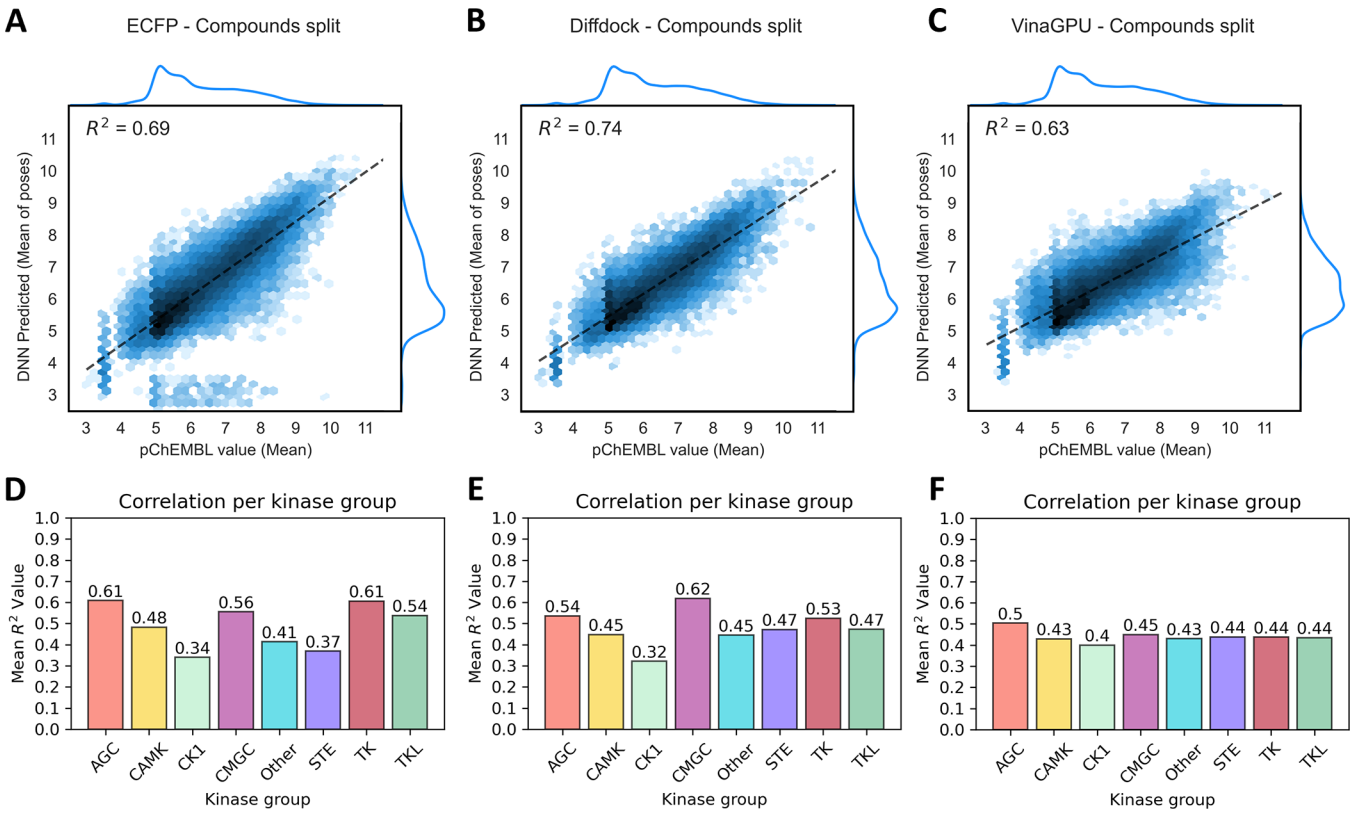


Figure 3. Model performance | Predicted affinity values vs literature values for the compounds-split test set displayed as logarithmic hexbin plots, as based on predictions for ECFP models (A), the DNN trained on DiffDock poses (B) and on the VinaGPU poses (C). Panels D, E and F show the average performance per kinase group for ECFP, DiffDock and VinaGPU models, respectively.

statistics per table is depicted in Figure S3A. The database includes all.mol-formatted poses in a compressed format, as well as the.pdb files for all KLIFS structures used. The database was designed to be readily usable for machine learning applications. Additionally, a KNIME-based user interface has been built to browse and query the generated docking complexes (Figure S3B). The full database and accompanying application are freely available on Zenodo and GitHub (*vide infra*).

Baseline Performance of Readily Available Docking Scores. The performance of two readily available docking scores was assessed to establish a baseline for bioactivity prediction. To this end, we assessed the predicted binding affinity by the Vina score, and used RFScoreVS[26] to rescore all poses generated by VinaGPU and DiffDock. The results are aggregated in Figure 2. Unsurprisingly, neither of the scoring algorithms showed any productive correlation with the Papyrus pChEMBL values, either when looking at the entire data set (Figure 2A-C) or when aggregating the per-kinase correlation coefficient (R^2) over the kinase groups (Figure 2D-F).

Kinome-wide Activity Predictions Learned from Docked Poses. We then set out to train a more performant kinase specific scoring function on this unprecedently large structural data set. First, the database was used to generate protein−ligand extended connectivity (PLEC)[27] fingerprints for the first three poses of every protein structure-inhibitor pair. All PLEC fingerprints were used as input for one single 3-layer Deep Neural Network tasked with predicting the affinity value based on a given fingerprint. This was done separately for the two docking algorithms, to compare their relative performance in this task. The generated models, which function as kinase-specific scoring functions, were trained on either a random 80:20 split of protein-inhibitor activity pairs, an 80:20 compound-based split (completely unseen compounds) or an 80:20 split based on kinases (completely unseen kinases as test set). These latter splits are intended to assess the generalization capabilities of the models toward newly designed inhibitors or unseen kinase targets, respectively. As a nondocking 2D comparison, in parallel we trained the same DNN on only the ECFP4 fingerprints of the inhibitors, to assess the added value of using docked poses as input. In this case we trained one model per kinase for all kinases that had at least 100 unique inhibitors known (172 out of 226 kinases in the data set). The performance results of these models are shown in Figure 3, Figure S4 and Figure S5 and specified per kinase in Supplementary Tables 1−3 (Supporting Information).

Regardless of the underlying docking algorithm, the performance of the DNNs trained on the compound splits (R^2 = 0.63−0.74) vastly outperformed both the original Vina scoring (R^2 = 0.04) as well as the rescoring using RFScoreVS (R^2 = 0.05−0.06). For the DiffDock model, for 86 out of 214 kinases (40%) the R^2 of the compound split was higher than 0.6, yielding predictions of sufficient quality to be genuinely informative in drug discovery projects. This value is comparable to the ECFP models, where 84 out of the 172 were ≥0.6. Of note, the ECFP models were only trained for kinases with ≥100 compounds, which leads to fewer kinases covered overall. The DiffDock model can extrapolate to some extend to the lower coverage kinases that are lacking in de ECFP models, with an $R^2 \geq 0.6$ for 18% of these (8 out of 42). The VinaGPU model shows somewhat lower overall perform-

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

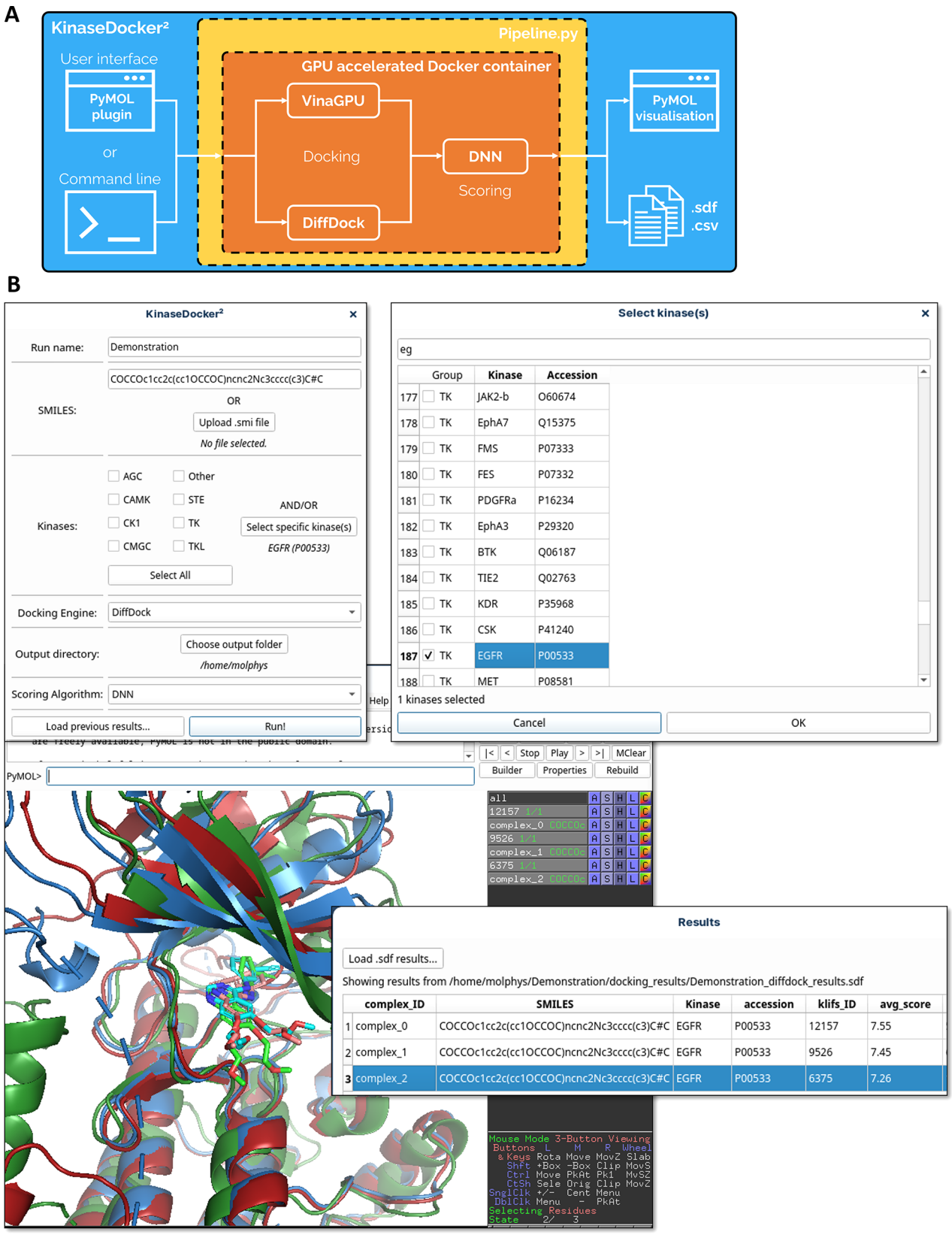


Figure 4. A user-friendly application: KinaseDocker² | (A) Schematic overview of the software setup; (B) screenshots of the Graphical User Interface of KinaseDocker² showing the initial configuration and kinase selection screens, as well as the result after running the program. The prediction results are shown in a table on screen, with the affinity value (avg_score) and details of the docking run. Docked complexes can be loaded in 3D for further assessment.

ance, with 65 out of all 220 models having an $R^2 \geq 0.6$, and 5 of the low-coverage kinases. This corresponds to the overall higher RMSD as observed in the docking procedure, pointing to the lower quality of the underlying training data.

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

Comparing the DiffDock and VinaGPU-based models shows some intriguing results. There is only a low correlation between the performances per kinase (Figure S6−7). This can partially be attributed to the smaller number of successful docking poses DiffDock generated but could also be due to the intrinsic differences between the pose finding tools.

The different splits clearly showed that the random splits performed best overall, although only slightly outcompeting the compound split. This is to be expected as for 78% of the compounds there is only 1 activity in the data set, meaning that the random and compound splits have highly similar difficulties in practice. However, for unseen kinases the performance drops significantly (Figure S5). This seems to indicate that the model strongly relies on the kinase structure underlying the complexes and suggests that when appending new kinases or KLIFS structures to the data set, retraining of the model is warranted.

KinaseDocker[2] Release for Direct Local Application. Encouraged by the strong performance across the kinome we decided to wrap our workflow and models in a user-friendly application that allows predictions to be generated by a medicinal chemist in real-world applications. Because the model inherently generates docking poses on which the affinity prediction is based, interpretation of the reliability of the output can be done on a per-compound and per-kinase basis. With this interpretability end point in mind, we chose the open-source molecular viewer PyMOL as the basis for the program. Schematically shown in Figure 4A, we built a PyMOL plugin that launches a pipeline to handle the predictions. The docking and consecutive bioactivity prediction by the neural network is handled by a Docker image that requires minimal installation by the user.

The user interface is shown in screenshots in Figure 4B, where the plugin allows the input of a (list of) SMILES strings, the selection of a (list of) kinases and the choice of docking engine. After the prediction is run, the output data is written to files as well as presented in a Results table on screen, showing the generated complexes with their predicted pIC_{50} (avg_score in the screenshot). Generated complexes can be loaded and inspected in the PyMOL session. Programmatic access is available if larger scale runs are desired. The whole codebase has been designed to be modular, allowing the future implementation of different model architectures or structure encodings. All code and Docker images are openly available, see section Code Availability.

DISCUSSION

The homogeneity of the sources of biochemical data in the Papyrus data set (and nearly every other publicly available data set) inherently means that there is considerable noise in the data. Realistically, R^2 values of around 0.8 are as high as can be achieved when taking experimental error into account.[28] This means that the DiffDock model for 42 kinases (±20%) already reached this maximum. For these, no significant improvement on this metric can be expected regardless of the methodological improvements or addition of further data. Adding more (diverse) compounds would for these kinases merely expand the chemical space where the model is applicable. For the kinases with lower performing predictions, the addition of more data and/or more structures could still increase performance.

Training (machine learning-based) scoring functions on structural data has been a successful strategy for years, enabled by data sets such as the PDBbind, as demonstrated by, for example, the RFScore series.[12,26,29−31] Utilizing the accuracy of pose finding in docking algorithms to synthesize an orders of magnitude larger training data set has, to the best of our knowledge, not been attempted before. Here we clearly showed that the approach in the basis works and outperforms current state-of-the-art in this kinase-specific use-case. There are many possibilities for future improvement over the current machine learning implementation. The docking performance of our VinaGPU workflow was not very high, with an average RMSD > 5 Å. More manual curation of the data set could reduce the amount of flawed docking poses, arguably positively impacting the quality of the training data.

From a machine learning perspective, the current choice of encoding the poses (3D) using PLEC fingerprints (1D) and utilizing a basic DNN architecture is inherently lossy. Implementing geometric deep learning models directly on the 3D data could positively impact performance if it can make better use of the available information. Additionally, the attention mechanism of the Transformer architecture could be used to highlight important regions in the complex for the generated prediction, yielding better interpretability and guidance for compound optimization.

There are more domain-focused improvements that could improve the performance too. The current implementation uses every KLIFS structure available for a certain kinase when docking a compound, regardless of inhibitor type (type I, II, III).[32,33] Previous work has shown that ML models can differentiate Type I and II inhibitors based on structure to a reasonable extend.[34] By only considering the poses of a molecule in their preferred activity state (DFG-in or -out), when available, the predictions should theoretically be improved. Another limitation is the domain of covalent kinase inhibitors. Though reliable poses can be obtained with known covalent drugs, noncovalent docking poses can never capture the influence of covalent bond formation.

To broaden the scope of kinases for which predictions can be made, structural data on the proteins is currently the main bottleneck. Of the 636 kinases, 226 (±35%) have crystal structures that meet our criteria. Of these, only about 26% (59) have both DFG-in and -out(-like) structures available. A strategy to enrich this data set could be through homology modeling. Considering the high sequence and structural similarity in the kinase domains, for many if not most kinases a reliable homology model in both DFG-states should be feasible to obtain. Adding these to the data set would not only considerably extend the applicability of the model to the entire kinome, it would also grow the size of the available biochemical training data with >100,000 data points for which currently no high quality experimental structure is available.

CONCLUSIONS

Kinase inhibitors are an essential part of anticancer therapy. Developing new kinase inhibitors suitable for clinical use requires these to be as specific as possible, targeting primarily the intended kinase. Due to the high homology in kinase domains, this is not a trivial requirement. Computational tools to aid in the development of these inhibitors by predicting inhibition across the kinome can be of great value. Current state-of-the-art struggles to perform well across the protein family, in part due to the lack of suitable data. Here, we generate a large data set of predicted binding poses, each

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

corresponding to an experimental binding affinity in the Papyrus data set, where a high-quality kinase domain structure of the target is available in the KLIFS database. We showed that this data set forms a strong basis on which to train machine learning models that can predict binding affinities of compounds for a wide variety of targets. We trained a Deep Neural Network on a 1D protein−ligand interaction fingerprint representation (PLEC) and showed that this vastly outperforms readily available (re)scoring functions like Vina score and RFscoreVS. Encouraged by these results, we developed a user-friendly interface to bring the automated docking procedure and scoring function as a freely available tool called KinaseDocker[2] to the bench chemist. Simultaneously, we ensured the modularity of the code, so that exchanging the protein−ligand complex encoding or the predictive model for more advanced approaches is feasible. Finally, we setup an interface for the database of docking poses to expose the data encapsulated in this to the general (bio)chemist.

We expect that the scoring functions trained here are useful as is, but also that, together with the data set generated here, they form a starting point to further tackle the kinase selectivity question, enabling the reliable prediction of affinities across the kinome to aid in bringing new and safe anticancer drugs to patients.

METHODS

Biochemical Data. Data was retrieved from Papyrus v5.5,[18] filtering on the Uniprot Protein Class "Kinase" and data quality "High". The data was matched to the KLIFS[20] data set based on Uniprot[35] accessions. Mutations were disregarded and averages for unique compound − Uniprot pairs were used as activity value (pChEMBL). Included bioactivities were filtered based on the drug-likeness of the measured compounds. Filters used were MW between 250 and 750 Da, rotatable bonds ≤15, number of hydrogen bond donors ≤10 and number of hydrogen bond acceptors ≤15, calculated using RDKit.[24]

Structural Data. Kinase structures and annotations were retrieved from KLIFS in October 2022. The structures were filtered on resolution (≤2.5 Å) and missing residues (≤5) after which the highest quality (KLIFS metric) structure was selected based on DFG-in/out and αC-helix states as annotated in KLIFS, if available. The.mol2 files were downloaded and converted to PDB files using OpenBabel.[36] PDB structures thus generated were used as is for DiffDock or further converted to.pdbqt format using the Open Drug Discovery Toolkit[37] for use with Autodock VinaGPU.

Docking Benchmark Set. Ligands from the KLIFS database were extracted and used as a benchmark data set for the two docking algorithms used. RMSD was determined using the CalcLigRMSD extension for RDkit.[38]

Pocket Definition. Pockets for Autodock Vina docking were automatically generated using PyVOL[25] using default settings with manual curation to encompass the entire ATP-binding pocket. The largest pocket detected in most cases represented the ATP-binding site, to which a 5 Å padding was added for the docking box. DiffDock was executed without restraints on binding site location (i.e., blind docking).

Ligand Preparation. SMILES strings from the Papyrus data set were transformed into 2D structures using default settings and enantiomers and cis/trans isomers were enumerated using RDKit.[24] These RDKit objects were converted to.pdbqt format for VinaGPU docking using the Open Drug Discovery Toolkit.[37] The RDKit objects were written to.csv files in canonical SMILES format with stereo information to use as DiffDock input.

Docking. Two docking procedures were employed: DiffDock[22] and AutoDock VinaGPU,[21,39] both installed through Docker.[23] Generated VinaGPU poses were converted to mol-format using RDKit.

AutoDock VinaGPU. A Docker image of AutoDock VinaGPU[21,40] was used, running on commercial RTX4070 or RTX3070 GPUs. For each protein, the corresponding KLIFS structures with predefined binding site boxes were iterated and all compounds with known activities docked. The AutoDock VinaGPU implementation differs slightly from the well characterized CPU version in its docking settings, where the *exhaustiveness* parameter is now replaced by *search_depth* and *thread*. A small parameter optimization was performed to benchmark the performance of VinaGPU on this data set, resulting in the final settings *search_depth* = 10, *threads* = 8192 which resulted in balanced performance vs run time (data not shown). Output.pdbqt formatted poses were converted to.mol format using OpenBabel and aggregated in a tabular format for inclusion in the database.

DiffDock. The original DiffDock Github release of October 2021 was used. Compounds were provided in canonical SMILES format with explicit stereochemistry. ESM embeddings were generated using the provided scripts and default settings:

```
--repr_layers 33 --include per_tok
```

For inference, the release inference.py script was used with minor changes relating to the output data structure. The author recommended settings for high throughput inference were used:

```
--inference_steps 20 --samples_per_complex 5 --batch_size 10 --actual_steps 18
--no_final_step_noise
```

Output.sdf formatted poses were expanded to.mol format and aggregated in a tabular format for inclusion in the database.

Clash-Score Filtering. The filter criterion (clash <10) was based on the Vina output, where after fitting a normal distribution on the clash scores a 3σ upper limit was calculated to be 10.02, which was visually inspected to be sensible and used for both docking algorithms. The clash-score was calculated per atom using the formula:

$$\max\left[0,\ 1 - \frac{d}{(r_1 + r_2)}\right]$$

where d is the Euclidian distance between the atoms, and r_1 and r_2 are the van der Waals radii of the respective atom types.[41] This per-atom contribution was calculated based on selections made using the PyMOL API. In brief, KLIFS.pdb and docking pose.mol-files were loaded in PyMOL. A selection around 4 Å of the ligand was made, and for all resulting atoms pairs the clashing contribution was determined. All contributions were summed to yield the pose clash-score.

Machine Learning. All machine learning algorithms were implemented in PyTorch 2.0. Splits were curated to ensure that the test set pChEMBL distribution is similar to the train set distribution. All DNNs were 3-layer fully connected NNs with the input layer either 2048 bits (ECFP) or 65536 bits (PLEC) to 4000, the hidden layer 4000 inputs to 1000 outputs

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204

and the output layer using the 1000 inputs to 1 output value. All layers use ReLU activation functions and the input and hidden layers use a dropout rate of 25% during training. The learning rate is fixed at 10^{-5}, batch size 128 with 100 epochs as fixed termination. After every epoch the performance on the test set is evaluated and the best model is stored. Typically, 50−70 epochs are required to reach a plateau.

Prediction Aggregation. For any given kinase-compound combination, the top 3 poses for all available KLIFS structures were scored by the DNN. To get to a final activity prediction, we tested several aggregation strategies. Taking the mean value of all options (aggregating the various KLIFS, all available stereoisomers, and the top 3 poses) yielded consistently the highest R^2 (Figure S8). As expected, using only the top 1 pose (according to Vina or DiffDock ranking) performed slightly better than taking only the second or third ranked pose, showing that on average the built-in scoring mechanism of both algorithms is able to prioritize the most relevant poses. However, averaging either the top 2 or top 3 poses consistently improved the performance.

■ ASSOCIATED CONTENT

Data Availability Statement

The 3D structure database generated as part of this work is available as an .sqlite database on Zenodo (10.5281/zenodo.10894122), together with the KNIME workflow that provides a simple user interface to search it. Code to reproduce the work described in this paper is available on GitHub (https://github.com/APAJanssen/KinaseDocker2-Paper-code). The PyMOL plugin is available on its own GitHub (https://github.com/APAJanssen/KinaseDocker2), which contains instructions on how to set up the Docker environment. The Docker image is available on Docker Hub (https://hub.docker.com/repository/docker/apajanssen/kinasedocker2).

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acs.jcim.4c01260.

Regression statistics for the individual kinases in Tables S1−S3 (XLSX)

Additional materials including Figures S1−S8 (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Antonius P.A. Janssen − *Department of Molecular Physiology, Leiden Institute of Chemistry, Leiden University, Leiden 2333CC, The Netherlands; Oncode Institute, Utrecht 3521AL, The Netherlands;* orcid.org/0000-0003-4203-261X; Email: a.p.a.janssen@lic.leidenuniv.nl

Authors

Jordy Schifferstein − *Department of Molecular Physiology, Leiden Institute of Chemistry, Leiden University, Leiden 2333CC, The Netherlands; Oncode Institute, Utrecht 3521AL, The Netherlands*

Andrius Bernatavicius − *Leiden Institute of Advanced Computer Science, Leiden University, Leiden 2333CC, The Netherlands*

Complete contact information is available at: https://pubs.acs.org/10.1021/acs.jcim.4c01260

Author Contributions

APAJ conceived the project. JS, AB, and APAJ performed the work described herein. JS and APAJ wrote the paper.

Funding

Research reported in this publication was supported by the Oncode Institute and Oncode Accelerator, a Dutch National Growth Fund project under grant NGFOP2201.

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

The authors thank Roelof van der Kleij for helping with the usage of the university's computational resources. Dr. Olivier Béquignon is acknowledged for fruitful discussions regarding the high-throughput docking. A.P.A.J. expresses his sincere gratitude to prof.dr. Gerard J.P. van Westen and prof.dr. Mario van der Stelt for their mentorship and critical feedback on the manuscript.

■ REFERENCES

(1) Roskoski, R. Properties of FDA-Approved Small Molecule Protein Kinase Inhibitors: A 2024 Update. *Pharmacol. Res.* **2024**, *200*, No. 107059.

(2) Fedorov, O.; Müller, S.; Knapp, S. The (Un)Targeted Cancer Kinome. *Nat. Chem. Biol.* **2010**, *6* (3), 166−169.

(3) van Esbroeck, A. C. M.; Janssen, A. P. A.; Cognetta, A. B.; Ogasawara, D.; Shpak, G.; van der Kroeg, M.; Kantae, V.; Baggelaar, M. P.; de Vrij, F. M. S.; Deng, H.; Allarà, M.; Fezza, F.; Lin, Z.; van der Wel, T.; Soethoudt, M.; Mock, E. D.; den Dulk, H.; Baak, I. L.; Florea, B. I.; Hendriks, G.; De Petrocellis, L.; Overkleeft, H. S.; Hankemeier, T.; De Zeeuw, C. I.; Di Marzo, V.; Maccarrone, M.; Cravatt, B. F.; Kushner, S. A.; van der Stelt, M. Activity-Based Protein Profiling Reveals off-Target Proteins of the FAAH Inhibitor BIA 10−2474. *Science* **2017**, *356* (6342), 1084−1087.

(4) Manning, G.; Whyte, D. B.; Martinez, R.; Hunter, T.; Sudarsanam, S. The Protein Kinase Complement of the Human Genome. *Science* **2002**, *298* (5600), 1912−1934.

(5) Zarrinkar, P. P.; Gunawardane, R. N.; Cramer, M. D.; Gardner, M. F.; Brigham, D.; Belli, B.; Karaman, M. W.; Pratz, K. W.; Pallares, G.; Chao, Q.; Sprankle, K. G.; Patel, H. K.; Levis, M.; Armstrong, R. C.; James, J.; Bhagwat, S. S. AC220 Is a Uniquely Potent and Selective Inhibitor of FLT3 for the Treatment of Acute Myeloid Leukemia (AML). *Blood* **2009**, *114* (14), 2984−2992.

(6) Sorgenfrei, F. A.; Fulle, S.; Merget, B. Kinome-Wide Profiling Prediction of Small Molecules. *ChemMedChem.* **2018**, *13* (6), 495−499.

(7) Merget, B.; Turk, S.; Eid, S.; Rippmann, F.; Fulle, S. Profiling Prediction of Kinase Inhibitors: Toward the Virtual Assay. *J. Med. Chem.* **2017**, *60* (1), 474−485.

(8) Cichonska, A.; Ravikumar, B.; Parri, E.; Timonen, S.; Pahikkala, T.; Airola, A.; Wennerberg, K.; Rousu, J.; Aittokallio, T. Computational-Experimental Approach to Drug-Target Interaction Mapping: A Case Study on Kinase Inhibitors. *PLOS Comput. Biol.* **2017**, *13* (8), No. e1005678.

(9) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; Bridgland, A.; Meyer, C.; Kohl, S. A. A.; Ballard, A. J.; Cowie, A.; Romera-Paredes, B.; Nikolov, S.; Jain, R.; Adler, J.; Back, T.; Petersen, S.; Reiman, D.; Clancy, E.; Zielinski, M.; Steinegger, M.; Pacholska, M.; Berghammer, T.; Bodenstein, S.; Silver, D.; Vinyals, O.; Senior, A. W.; Kavukcuoglu, K.; Kohli, P.; Hassabis, D. Highly Accurate Protein Structure Prediction with AlphaFold. *Nature* **2021**, *596* (7873), 583−589.

(10) Ain, Q. U.; Aleksandrova, A.; Roessler, F. D.; Ballester, P. J. Machine-Learning Scoring Functions to Improve Structure-Based

Binding Affinity Prediction and Virtual Screening. *WIREs Comput. Mol. Sci.* **2015**, *5* (6), 405−424.
(11) Li, H.; Sze, K.; Lu, G.; Ballester, P. J. Machine-learning Scoring Functions for Structure-based Virtual Screening. *WIREs Comput. Mol. Sci.* **2021**.
(12) Ballester, P. J.; Mitchell, J. B. O. A Machine Learning Approach to Predicting Protein−Ligand Binding Affinity with Applications to Molecular Docking. *Bioinformatics* **2010**, *26* (9), 1169−1175.
(13) Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-Wide Collection of Binding Data: Current Status of the PDBbind Database. *Bioinformatics* **2015**, *31* (3), 405−412.
(14) Li, Z.; Li, X.; Liu, X.; Fu, Z.; Xiong, Z.; Wu, X.; Tan, X.; Zhao, J.; Zhong, F.; Wan, X.; Luo, X.; Chen, K.; Jiang, H.; Zheng, M. KinomeX: A Web Application for Predicting Kinome-Wide Polypharmacology Effect of Small Molecules. *Bioinformatics* **2019**, *35* (24), 5354−5356.
(15) Li, Z.; Qu, N.; Zhou, J.; Sun, J.; Ren, Q.; Meng, J.; Wang, G.; Wang, R.; Liu, J.; Chen, Y.; Zhang, S.; Zheng, M.; Li, X. KinomeMETA: A Web Platform for Kinome-Wide Polypharmacology Profiling with Meta-Learning. *Nucleic Acids Res.* **2024**, *52* (W1), W489−W497.
(16) Shen, Z.; Yan, Y.-H.; Yang, S.; Zhu, S.; Yuan, Y.; Qiu, Z.; Jia, H.; Wang, R.; Li, G.-B.; Li, H. ProfKin: A Comprehensive Web Server for Structure-Based Kinase Profiling. *Eur. J. Med. Chem.* **2021**, *225*, No. 113772.
(17) Leach, A. R.; Shoichet, B. K.; Peishoff, C. E. Prediction of Protein−Ligand Interactions. Docking and Scoring: Successes and Gaps. *J. Med. Chem.* **2006**, *49* (20), 5851−5855.
(18) Béquignon, O. J. M.; Bongers, B. J.; Jespers, W.; IJzerman, A. P.; van der Water, B.; van Westen, G. J. P. Papyrus: A Large-Scale Curated Dataset Aimed at Bioactivity Predictions. *J. Cheminformatics* **2023**, *15* (1), 3.
(19) Kooistra, A. J.; Kanev, G. K.; van Linden, O. P. J.; Leurs, R.; de Esch, I. J. P.; de Graaf, C. KLIFS: A Structural Kinase-Ligand Interaction Database. *Nucleic Acids Res.* **2015**.
(20) Kanev, G. K.; de Graaf, C.; Westerman, B. A.; de Esch, I. J. P.; Kooistra, A. J. KLIFS: An Overhaul after the First 5 Years of Supporting Kinase Research. *Nucleic Acids Res.* **2021**, *49* (D1), D562−D569.
(21) Ding, J.; Tang, S.; Mei, Z.; Wang, L.; Huang, Q.; Hu, H.; Ling, M.; Wu, J. Vina-GPU 2.0: Further Accelerating AutoDock Vina and Its Derivatives with Graphics Processing Units. *J. Chem. Inf. Model.* **2023**, *63* (7), 1982−1998.
(22) Corso, G.; Stärk, H.; Jing, B.; Barzilay, R.; Jaakkola, T. DiffDock: Diffusion Steps, Twists, and Turns for Molecular Docking. *arXiv* October 4, 2022. http://arxiv.org/abs/2210.01776 (accessed 2022−10−24).
(23) Merkel, D. Docker: Lightweight Linux Containers for Consistent Development and Deployment. *Linux J.* **2014**, *2014* (239), 2.
(24) Landrum, G. *RDKit: Open-Source Cheminformatics*; Http://Www.Rdkit.Org. http://www.rdkit.org.
(25) Smith, R. H. B.; Dar, A. C.; Schlessinger, A. PyVOL: A PyMOL Plugin for Visualization, Comparison, and Volume Calculation of Drug-Binding Sites. *bioRxiv* **2019**, 816702.
(26) Wójcikowski, M.; Ballester, P. J.; Siedlecki, P. Performance of Machine-Learning Scoring Functions in Structure-Based Virtual Screening. *Sci. Rep.* **2017**, *7*, 46710.
(27) Wójcikowski, M.; Kukiełka, M.; Stepniewska-Dziubinska, M. M.; Siedlecki, P. Development of a Protein-Ligand Extended Connectivity (PLEC) Fingerprint and Its Application for Binding Affinity Predictions. *Bioinforma. Oxf. Engl.* **2019**, *35* (8), 1334−1341.
(28) Landrum, G. A.; Riniker, S. Combining IC50 or Ki Values from Different Sources Is a Source of Significant Noise. *J. Chem. Inf. Model.* **2024**, *64*, 1560.
(29) Ballester, P. J.; Schreyer, A.; Blundell, T. L. Does a More Precise Chemical Description of Protein−Ligand Complexes Lead to More Accurate Prediction of Binding Affinity? *J. Chem. Inf. Model.* **2014**, *54* (3), 944−955.
(30) Li, H.; Leung, K.-S.; Wong, M.-H.; Ballester, P. J. Improving AutoDock Vina Using Random Forest: The Growing Accuracy of Binding Affinity Prediction by the Effective Exploitation of Larger Data Sets. *Mol. Inform.* **2015**, *34* (2−3), 115−126.
(31) Li, H.; Leung, K.-S.; Wong, M.-H.; Ballester, P. J. Correcting the Impact of Docking Pose Generation Error on Binding Affinity Prediction. *BMC Bioinformatics* **2016**, *17* (11), 308.
(32) Dar, A. C.; Shokat, K. M. The Evolution of Protein Kinase Inhibitors from Antagonists to Agonists of Cellular Signaling. *Annu. Rev. Biochem.* **2011**, *80* (1), 769−795.
(33) Gavrin, L. K.; Saiah, E. Approaches to Discover Non-ATP Site Kinase Inhibitors. *MedChemComm* **2013**, *4* (1), 41−51.
(34) Abdelbaky, I.; Tayara, H.; Chong, K. T. Prediction of Kinase Inhibitors Binding Modes with Machine Learning and Reduced Descriptor Sets. *Sci. Rep. 2021 111* **2021**, *11* (1), 1−13.
(35) UniProt Consortiums. UniProt: The Universal Protein Knowledgebase. *Nucleic Acids Res.* **2017**, *45* (D1), D158−D169.
(36) O'Boyle, N. M.; Banck, M.; James, C. A.; Morley, C.; Vandermeersch, T.; Hutchison, G. R. Open Babel: An Open Chemical Toolbox. *J. Cheminformatics* **2011**, *3* (1), 33.
(37) Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): A New Open-Source Player in the Drug Discovery Field. *J. Cheminformatics* **2015**, *7* (1), 26.
(38) *CalcLigRMSD - GitHub*. GitHub. https://github.com/rdkit/rdkit/tree/master/Contrib/CalcLigRMSD (accessed 2023−04−27).
(39) Trott, O.; Olson, A. J. AutoDock Vina: Improving the Speed and Accuracy of Docking with a New Scoring Function, Efficient Optimization, and Multithreading. *J. Comput. Chem.* **2010**, *31* (2), 455−461.
(40) Andrius, Bernataviicus*Highly Parallel Molecular Docking Pipeline Using Vina-GPU (Dockerized) + AutoDock Vina CPU*, 2024. https://github.com/andriusbern/vinaGPU (accessed 2024−02−27).
(41) *PeriodicTable.com*. https://periodictable.com/ (accessed 2023−02−26).

https://doi.org/10.1021/acs.jcim.4c01260
J. Chem. Inf. Model. 2024, 64, 9196−9204