DockM8: An All-in-One Open-Source Platform for Consensus Virtual Screening in Drug Design

A PREPRINT

Antoine Lacour
Helmholtz Institute for Pharmaceutical Research Saarland (HIPS)
Helmholtz Centre for Infection Research (HZI)
Campus E8.1
66123 Saarbrücken
Germany
anla00008@uni-saarland.de

Hamza Ibrahim
Data Driven Drug Design
Faculty of Mathematics and Computer Sciences
Saarland University
Campus E2.1 66123, Saarbrücken
Germany
haib00001@uni-saarland.de

Andrea Volkamer
Data Driven Drug Design
Faculty of Mathematics and Computer Sciences
Saarland University
Campus E2.1 66123, Saarbrücken
Germany
volkamer@cs.uni-saarland.de

Anna K. H. Hirsch
Helmholtz Institute for Pharmaceutical Research Saarland (HIPS)
Helmholtz Centre for Infection Research (HZI)
Campus E8.1
66123 Saarbrücken
Germany
anna.hirsch@helmholtz-hips.de

July 22, 2024

Abstract

In this study, we introduce DockM8, an innovative open-source platform designed for consensus virtual screening in drug design. Leveraging various docking algorithms and scoring functions, DockM8 provides a highly customizable workflow for structure-based virtual screening. Through extensive testing on DEKOIS 2.0, DUD-E, and Lit-PCBA datasets, we show that DockM8 demonstrates state-of-the-art performance compared to existing methods, highlighting its adaptability and generalizability across diverse targets. The study emphasizes the importance of tailoring the virtual screening strategy to specific targets, suggesting that no single pose selection or consensus method universally outperforms others. DockM8's user-friendly interface and minimal programming requirements make advanced virtual screening accessible to a broader scientific community. DockM8 is freely available at https://github.com/DrugBud-Suite/DockM8. We invite the computational chemistry community to participate in the further development of DockM8, envisioning its evolution as a powerful tool in drug discovery and medicinal chemistry.

Keywords Virtual Screening · Docking · Consensus · Open-Source

1 Introduction

In the pursuit of accelerating drug discovery, computational methods have become indispensable, with structure-based virtual screening (SBVS) emerging as a cornerstone strategy. Docking-based VS uses 3D structural information and computational algorithms to sift through large molecular libraries, identifying compounds with potential biological affinity for a specified target. Consequently, this screening reduces the number of compounds subjected to "wet-lab" assays, substantially curbing associated costs while hoping to discover novel active compounds [1, 2].

Molecular docking simulates the interaction between a ligand and its target, attempting to predict the binding pose of the ligand of interest and computing a docking score used to rank potential ligands based on predicted binding affinities [3]. Determining the accurate binding pose of a small molecule is essential not only for assessing its binding affinity but also for leveraging the pose in lead optimization. Furthermore, accurate

scoring of compounds is crucial for comparing ligands in a screening scenario and determining whether a particular ligand should be considered for experimental validation.

Conventional molecular docking software, including AutoDock [4], AutoDock Vina [5, 6], and Glide [7], predominantly employ heuristic search algorithms to systematically examine a range of potential ligand conformations. Although the sampling of binding poses performed by these algorithms has seen significant advances in recent years, particularly with recent developments, challenges in scoring and ranking the resulting poses still remain due to known shortcomings such as not considering protein flexibility, water-mediated interactions, or changes in entropy upon binding.

Over the past decades, a multitude of scoring functions (SFs) have been developed, ranging from empirical, physics-based, and knowledge-based SFs to the more recently developed machine-learning (ML) based SFs [8, 9]. Despite significant recent advances in this field, these scoring functions still lack universal applicability as they necessarily rely on a simplification of the complexity of protein–ligand binding. Indeed, the performance of any given scoring function on a given target does not necessarily correlate to performance on other targets [10]. Moreover, single scoring functions often struggle to handle both accurate binding affinity prediction and ligand ranking [11]. Several solutions have been proposed to deal with the issues surrounding the performance of molecular docking, including consensus docking (using multiple methods) and ensemble docking (incorporating several protein conformations).

Ensemble docking uses multiple receptor conformations to tackle the inherent challenge of protein flexibility. Indeed, early SBVS campaigns are mostly performed using flexible ligands and rigid receptor docking. Although this accelerates the calculation, it fails to represent the dynamics of the protein–ligand system under study. Ensemble docking suggests that using multiple protein states instead of a single static structure offers a more holistic view of ligand binding. This approach improves the predictive accuracy of molecular docking by considering scores and rankings across these various protein states [12].

Consensus docking has also emerged as a modality seeking to enhance the precision of molecular docking. By combining diverse docking algorithms and scoring functions, consensus approaches circumvent individual limitations through aggregation. This strategy was shown to yield more reliable binding pose predictions, better absolute and relative scoring of ligands, and superior generalizability [13, 14, 15, 16]. In recent years, a variety of consensus docking tools have been developed (Table 1). DockBox uses a variety of docking algorithms (AutoDock4 [4], Vina [5] and DOCK [17]) along with score-based consensus docking to enhance pose prediction and ligand ranking [18]. dockECR uses LeDock [19], rDock [20], Smina [21] and Vina along with exponential consensus ranking (ECR) to improve binding pose prediction [22]. Approaches such as MetaDOCK [23], CompScore [24] or ESSENCE-Dock [25] attempt to find novel methods of carrying out the consensus by using docking-pose based clustering, incorporating additional SF components in the consensus, or using pose RMSD and the number of rotatable bonds to further refine the consensus scoring. Also recently, DockingPie [26] offers a graphical user interface (GUI) in PyMOL [27] to aid in consensus screening.

Table 1: Comparison of Docking Tools. 'x' means respective column applies to this tool, '-' not applicable. The more x's the better.

Tool	Open-Source	Docking	GUI or Server	Number of Methods		No External Licenses Required	Year
				Consensus	Scoring		
VoteDock [28]	-	x	-	1	>5	-	2011
CompScore [24]	x	-	x	1	0	x	2019
DockBox [18]	x	x	-	2	5	-	2019
dockECR [22]	x	x	-	1	4	-	2021
DockStream [29]	x	x	-	0	>5	-	2021
pyscreener [30]	x	x	-	0	>5	x	2021
DockingPie [26]	x	x	x	1	4	-	2022
MetaDOCK [23]	-	x	x	1	3	x	2023
ESSENCE-DOCK [25]	x	-	-	1	3	-	2023
ChemFlow [31]	x	x	-	0	3	-	2023
easydock [32]	x	x	-	0	>2	-	2023
Dockey [33]	x	x	x	0	3	-	2023
DockM8 (ours)	x	x	x	10	16	x	2024

Despite their potential, the current consensus docking tools pose challenges, especially in the context of academic research (Table 1). Some of these tools lack user-friendliness and/or accessibility as open-source

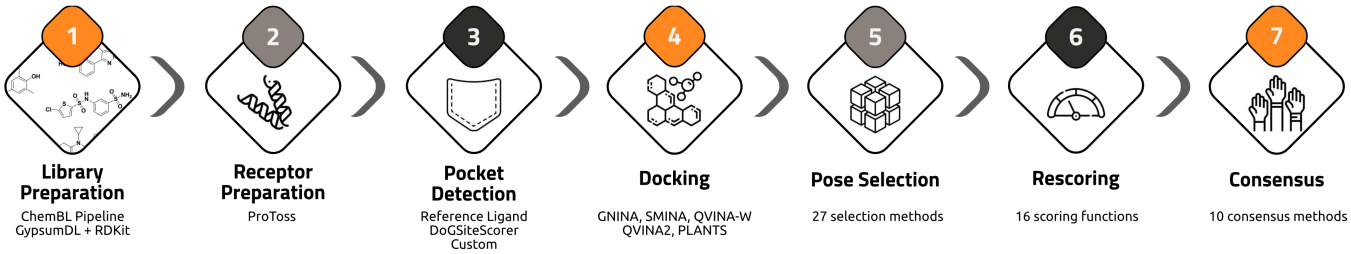

Figure 1: General workflow for consensus docking and scoring within DockM8.

software, hindering their widespread acceptance and incorporation into academic drug discovery workflows. Moreover, the majority depend on the scoring functions included in the docking tools, rather than leveraging the abundance of modern scoring functions available [9]. Furthermore, many necessitate manual compilation of code libraries or require software to be sourced from a variety of commercial or academic sources.

To mitigate these shortfalls, we developed DockM8, a fully-fledged open-source workflow for consensus virtual screening. The tool coordinates and manages a collection of various programs and APIs for protein and ligand protonation and preparation, binding site determination, docking, pose selection, rescoring, and consensus scoring (Figure 1). Additionally, DockM8 can run both in single and ensemble docking mode, both using up to five different docking tools(number as chosen by the user). We provide the ability to select poses based on eleven pose selection methods and rescore ligands using any of the sixteen scoring functions currently implemented. Ten consensus methodologies are implemented which, along with scoring function selection, allow for customization of the workflow to the protein target of interest. Most importantly, we designed DockM8 to be easy to use for computational and medicinal chemists alike and intended it to be a central platform for the virtual screening community. With this in mind we provide the ability to run DockM8 through a simple graphical user interface (GUI), via the command line or via Jupyter Notebooks. We actively invite collaboration to further enhance DockM8's capabilities. In this work, we introduce and extensively test DockM8's performance on subsets of the DUD-E [34], DEKOIS [35] and Lit-PCBA [36] datasets and show that it not only outperforms any single scoring function but allows for the customization of the protocol to individual targets.

2 DockM8 Pipeline

The DockM8 pipeline consists of seven steps, automating the entire process from data pre-processing, followed by docking and scoring, and finally consensus ranking.

2.1 1. Ligand Library Preparation

Ligands are supplied to DockM8 as an `.sdf` file. The ligand library is first standardized using the ChEMBL-structure-pipeline [37] library, which uses RDKit [38] to check, standardize and desalt ligands. Accurately predicting ligand protonation states remains a challenge in computational chemistry and few open-source libraries are available to perform this task [39]. As a result, we used Gypsum-DL for both compound protonation and 3D conformation generation [39].

2.2 2. Receptor Preparation

DockM8 gives users the option to prepare protein structures through querying the Protoss program via the proteins.plus webserver [40]. Protoss optimizes the hydrogen bonding network in protein–ligand complexes, addressing the complexities of hydrogen bonding, tautomerism, and ionization to enable precise analysis of binding modes and calculation of associated binding energies [41]. This provides streamlined access to ready-to-dock protein structures. Currently, DockM8 does not allow for modification of these automatically assigned protonation states, which may be critical in certain targets. In these cases, we encourage users to carefully modify the protein input file beforehand using the tools of their choice (e.g. PyMOL [27]).

2.3 3. Pocket Detection

Users are presented with multiple strategies for binding site selection. The `Reference` and `RoG` (radius of gyration) options make use of a co-crystallized ligand, which needs to be supplied as a separate file. Alternatively, an already docked pose could also be used. In the `Reference` method, DockM8 automatically

determines the ligand's center of mass (CoM) and defines a box (the dimensions of which can be specified by the user) around this CoM. The `RoG` method determines the box's center coordinates in the same way as before but uses the previously defined reference ligand's radius of gyration (`RoG`) to define the box size. This was shown to be a suitable method to determine box size for AutoDock Vina [42].

Alternatively, if no crystal ligand is available, DockM8 can make use of the DoGSiteScorer algorithm which combines pocket finding along with pocket characterization to select druggable binding sites [43]. DockM8's query of DoGSiteScorer via the proteins.plus webserver outputs the largest binding site by default. An option to select binding sites based on other metrics provided by DogSiteScorer is also included (available metrics are: Volume, Surface, Depth and Druggability Score).

Finally, the user can supply custom box center and size values, both in the GUI and by command line. Details can be found in the DockM8 Usage Guide. The binding site definition is then used for all further docking and rescoring functions.

2.4 4. Docking

DockM8 currently supports five distinct docking algorithms: (1) SMINA, a fork of AutoDock Vina with an improved scoring function and energy minimization processes [21]. (2) GNINA, an offshoot of SMINA, which employs convolutional neural networks (CNN) as a scoring function [3]. (3) QVINA-W [44] and (4) QVINA2 [45], both derived from AutoDock Vina [6], which introduce advancements in computational speed, thereby expediting the docking process. And (5) PLANTS, which is based on an ant colony optimization algorithm [46]. The user can choose from any combination of the above mentioned docking algorithms integrated in DockM8.

In all cases, the docking calculations are parallelized by running one docking run on one CPU core and using the *multiprocessing* or *joblib* libraries to manage the running jobs. The ligand library is split beforehand according to the number of CPU cores defined by the user to allow parallel handling of the docking tasks. For QVINA-W and QVINA2, the necessary conversion of the ligand and receptor files to *pdbqt*-files is handled by Meeko [47] and the Python implementation of OpenBabel with the addition of Gasteiger charges [48], respectively.

After the docking has been completed, the poses are filtered using the recently developed PoseBusters library[49], which implements pose quality checks using RDKit. We developed a custom configuration file (*posebusters-config.yml*) to reduce the calculation times while retaining quality checks considered critical, which can be adjusted by the user.

2.5 5. Pose Selection

One of the challenges associated with consensus docking is the fact that the number of generated poses scales with the number of docking algorithms used. To address this, DockM8 supports a variety of pose selection options. Traditionally, the best pose (as determined by the docking score) for each ligand is selected for further analysis. DockM8 supports this through the `bestpose` series of pose selection options. The best pose for a single docking program can be used (e.g. `bestpose_SMINA` or `bestpose_PLANTS`), which will return a single pose for each ligand in the library. Alternatively, the `bestpose` option (without tool specification) selects the best pose of each ligand for each selected docking algorithm, outputting the same amount of poses as docking algorithms selected (for each ligand). Additionally, due to the large number of poses being generated, we implemented a module to cluster the docking poses based on various metrics. Indeed, if several docking poses of the same ligand are very similar, further calculations can be simplified by using one of those poses as the representative pose. To achieve this, all the docked poses are put into an identity matrix and a clustering metric is calculated. Currently, the following metrics are available in DockM8:

- RMSD: heavy-atoms root mean square displacement (RMSD), used to cluster poses by their geometric similarity
- spyRMSD [50]: symmetry-corrected heavy-atoms RMSD, used to cluster poses by their geometric similarity
- espsim [51]: electrostatic shape similarity, used to cluster poses by similar interaction potential
- USRCAT [52]: shape similarity, used to cluster poses by similar 3D shape
- 3DScore [28]: the mean spyRMSD of each pose relative to all other poses is calculated and the pose with the lowest 3DScore is retained

For the first four metrics, the identity matrix is then subjected to clustering. Two options are available: K-medoids [53] and Affinity Propagation [54]. These methods were selected because they output a cluster member as the cluster center, which represents a real ligand pose. For the K-medoids algorithm, the ideal

Table 2: List of scoring functions available in DockM8

Scoring Function	Type	Note
AutoDock4 [4]	Semi-empirical	Semi-empirical scoring function from AutoDock4
GNINA-Affinity [3]	Empirical	Empirical scoring function bundled in GNINA
AAScore [57]	Empirical	Amino-acid specific empirical scoring
LinF9 [58]	Empirical	Improved scoring based on 9 empirical terms
CHEMPLP [46]	Empirical	Expanded PLP with GOLD terms for metals and torsions
PLP [46]	Empirical	Piecewise linear potential (4 terms)
Vinardo [59]	Empirical	Improved empirical scoring based on Vina
KORP-PL [60]	Knowledge-based	Coarse-grained knowledge-based scoring
Convex-PL^{R} [61, 62]	Knowledge-based	Scoring function incorporating entropic terms
CNN-Score [3]	Machine-learning	CNN pose scoring model bundled in GNINA
CNN-Affinity [3]	Machine-learning	CNN binding affinity prediction bundled in GNINA
RF-Score-VS [63]	Machine-learning	Random Forest-based scoring function
SCORCH [64]	Machine-learning	Consensus ML scoring function
RTMScore [11]	Machine-learning	Graph transformer-based scoring function
PLEC [65]	Machine-learning	Protein–ligand fingerprint binding affinity prediction
NNScore [66]	Machine-learning	Neural network-based affinity prediction

number of clusters is determined by using the silhouette score [55] and an elbow-finding library [56]. Only the cluster center poses are selected for further processing. Finally, any one of the 16 scoring functions supported by DockM8 can be used to select docking poses. In this case, the best scoring pose for each ligand is used for further calculations.

2.6 6. Rescoring

DockM8 is bundled with 16 different scoring functions, which include empirical, semi-empirical, knowledge-based, and machine learning (ML) categories. Any combination of these scoring functions can be used during rescoring and during the consensus score calculation. As mentioned previously, any of these scoring functions can also be used to select docking poses. A list and description of the scoring functions available in DockM8 is shown in Table 2.

2.7 7. Consensus Methodologies

DockM8 currently supports ten different consensus methods. To give the user flexibility and the ability to adapt the protocol to the target under study, we included both traditional and more modern consensus strategies. All the methods described below operate on the poses selected by the methods described above and use the scores from the scoring functions as variables. All such scores are standardized using min-max standardization before calculating the consensus scores. Figure 2 below provides a visual representation of the various consensus strategies.

For each consensus method described below, there are two variants: `*_best` and `*_avg`. The `*_best` variant considers only the pose with the best consensus score for the final scoring (for each molecule ID). The `*_avg` variant takes the average consensus score of all the selected poses for a given molecule for the final consensus. Additionally, for ECR, there are two more variants: `avg_ECR` and `avg_R_ECR`, which are explained in their respective section. See Figure 2 for a visual explanation of these methods.

2.7.1 Rank by Rank (RbR)

The candidate molecules are selected based on the rank for all the selected scoring functions (Equation 1) [16]. Let r_y^x be the rank of the molecule x for scoring function y and n is the total number of scoring functions; then the output of the RbR consensus is given by:

$$RbR_x = \frac{1}{n} \sum_y r_y^x, \tag{1}$$

2.7.2 Rank by Vote (RbV)

Molecules receive votes if they rank above a certain threshold for a specific scoring function. The total score for each molecule is determined by the sum of votes across all scoring functions, ranging from zero to the

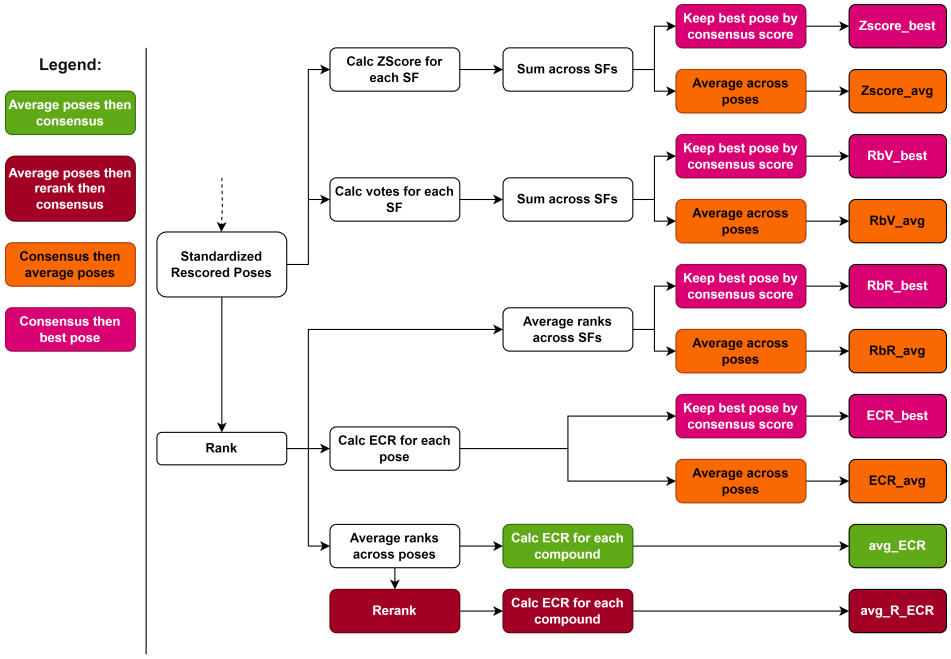

Figure 2: Visual overview of the various consensus strategies implemented in DockM8. The flowchart shows the steps taken (i.e. averaging pose data first or consensus scoring first) to arrive at the final consensus score.

total number of considered scoring functions. Candidates are then ranked according to their final number of votes [16].

2.7.3 Z-score

The molecule score (s) undergoes scaling using the average (μ) and standard deviation (σ) of scores for all molecules within each scoring function. The final score is the average of the scaled scores across all scoring functions (Equation 2) [67]. Let n be the total number of scoring functions, s_y^x the score of molecule x for scoring function y, μ^y the average score for scoring function y and σ^y the standard deviation of scoring function y; then the output of the Z-score consensus is given by:

$$Z - score_x = \frac{1}{n} \sum_y \frac{s_y^x - \mu^y}{\sigma^y}, \tag{2}$$

2.7.4 Exponential Consensus Ranking (ECR)

The ECR calculation method was taken from Palacio-Rodríguez et al. (Equation 3). [68]. Let σ be the expected value of the exponential distribution, which represents the threshold of the data to be taken into account in the consensus. Let r_y^x be the rank of the molecule x for scoring function y; then the ECR score is given by:

$$ECR_x = \frac{1}{\sigma} \sum_y \exp(-\frac{r_y^x}{\sigma}), \tag{3}$$

In addition to the standard `*_best` and `*_avg` variants, ECR has two more sub-methods:

- `avg_ECR`: First averages the ranks of all the selected poses for a given molecule, then calculates the ECR score based on the average rank values.
- `avg_R_ECR`: Similar to `avg_ECR` but first re-ranks all the molecules after the rank averaging.

2.8 Available Docking Modes

In order to partially account for protein flexibility, DockM8 supports two major operational modes:

Single Docking Mode: This mode performs "traditional" molecular docking against a single protein structure. In this case, the output of the workflow is a list of compounds ranked by their consensus score.

Ensemble Docking Mode: In this mode, users can choose multiple protein structures for the docking process. Each structure is processed sequentially, similar to the single docking mode. When utilizing this mode, users need to define a threshold as a percentage. The output displays the highest-scoring compounds, determined by the selected consensus method, across various protein targets. Only compounds that rank as top scorers (according to the consensus score) within the specified user threshold in all protein conformations are selected.

2.9 Performance Evaluation

A variety of metrics [69] were considered when analyzing the performance of the DockM8 workflow on the three data sets, containing labeled actives and decoys (see Section 3.1). A custom implementation of the enrichment factor (EF)[69] was implemented while for the Boltzmann-Enhanced Discrimination of ROC (BEDROC)[70], the `rdkit.Scoring` module was used. For the Area under the Receiver Operating Characteristic Curve (AUC-ROC) metric, the `sklearn` library was used.

The equation used for the enrichment factor is shown in Equation 4, where x is a percentage threshold, H^x is the number of hits above that threshold, N^x is the number of compounds at that threshold, H^{100} is the total number of hits and N^{100} is the total number of compounds:

$$EF_x = \frac{H^x}{N^x} \times \frac{N^{100}}{H^{100}}, \tag{4}$$

We developed a function (`calculate_performance`) that takes as input a rescored library and the labeled docking library (with actives and decoys) and determines the EF, BEDROC, and AUC-ROC metrics. This was used to determine DockM8's performance on the various benchmark datasets. It is also used to select the docking, pose selection, rescoring, and consensus conditions when using the `--gen_decoys` option.

2.10 Decoy Generation

Due to the considerable number of possible conditions under which DockM8 can be run, the user needs to be able to determine which choices are most appropriate for the target under study. One of the ways to achieve this is to generate decoy molecules that closely resemble the physicochemical properties of known active compounds for that particular target. If the screening software can discern the actives from the decoys effectively, the chances of being able to discern potentially active compounds from a real docking library are increased. When decoy generation is activated and a list of known active compounds is supplied, DockM8 will make use of the DeepCoy [71] library to generate a user-specified number of decoys.

Subsequently, the user can choose which conditions (docking, pose selection, and rescoring functions) are to be explored. This allows the user to limit the search space of conditions if desired. DockM8 will then run the workflow using every possible combination of the supplied conditions to determine which one is best able to distinguish active compounds from decoy molecules. The optimum conditions will then be automatically used for the screening of the library of interest.

2.11 Graphical User Interface

A simple graphical user interface (GUI) is provided for DockM8 via the Streamlit framework [72]. If the `streamlit` library is installed, the GUI can be launched by the command: `streamlit run gui.py`. The interface is shown in Figure 3.

2.12 Technical Details

The DockM8 workflow (Figure 1) was developed for Python 3.10 and the Ubuntu (version 20 and above) operating system, although it can be run in Windows Subsystem for Linux version 2 if required. Multiprocessing capabilities are integrated into DockM8, particularly during the docking and rescoring phases as these are the most resource-intensive. Allocation of individual processing runs across available CPU cores is managed by utilizing the `multiprocessing` or `joblib` libraries. DockM8 maintains a structured approach to file management, wherein a dedicated working directory houses the data generated after each step of ligand preparation, docking, clustering, and scoring procedures, thereby allowing subsequent analysis.

3 Data and Methods

DockM8 v1.0.3 was used for all benchmarking efforts.

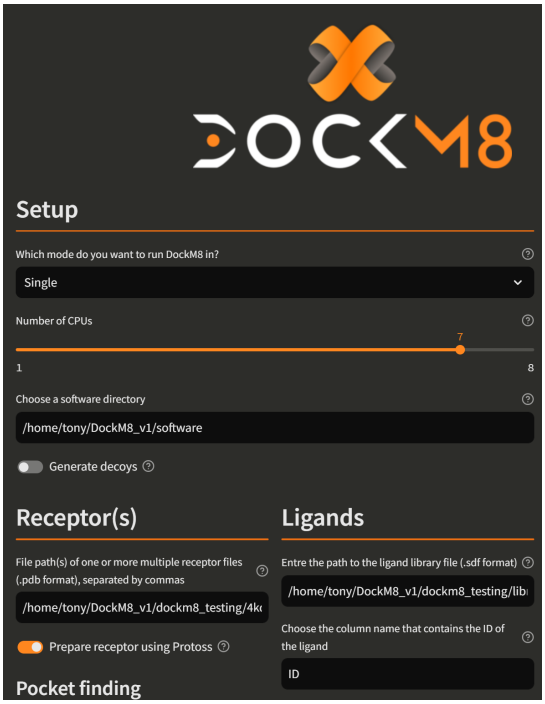

Figure 3: Screenshot of the DockM8 GUI running in Streamlit, showcasing the main interface for configuring inputs for the molecular docking simulations.

3.1 Benchmark Datasets

3.1.1 DEKOIS 2.0

A subset (see Supporting Information Section 1) of the DEKOIS 2.0 library (79 targets) was downloaded via the link provided in the original publication [35]. The reference ligand and protein files were used as is in the DockM8 workflow. The docking library was generated by combining the actives and decoys into one SDF file and labeling each with 0 (for decoys) or 1 (for actives) to allow for the calculation of the performance metrics. The DockM8 conditions were set as follows :

- Protein preparation: the protein file was protonated using the `--prepare_proteins = True` option.
- Pocket definition: the pocket was defined using the `--pocket = reference` option with a cutoff of 8 Å (box size of 16 Å).
- Ligand preparation: the ligands were protonated and 3D conformers were generated using Gypsum-DL.
- Docking: PLANTS, SMINA, GNINA, QVINA2, QVINAW: in all cases, 10 poses per ligand were generated.
- Pose selection: due to the high computational cost of exploring each scoring function as a pose selection tool, we elected to use only the following methods: `RMSD`, `spyRMSD`, `espsim`, `3DScore`, `bestpose`, `bestpose_GNINA`, `bestpose_SMINA`, `bestpose_PLANTS`, `bestpose_QVINA2`, `bestpose_QVINAW`, `KORP-PL`, `ConvexPLR` and `RTMScore`.
- Pose busting: Due to the high computational cost of using Pose Busters on this many docking poses, this feature was disabled.
- Rescoring: `GNINA-Affinity`, `CNN-Score`, `CNN-Affinity`, `Vinardo`, `AD4`, `KORP-PL`, `ConvexPLR`, `LinF9`, `RTMScore`, `RFScoreVS`, `CHEMPLP`, `NNScore`, `PLECScore` (`SCORCH` and `AAScore` were omitted in the benchmarking used due to high computational cost. `PLP` was omitted due to high correlation with `CHEMPLP`).
- Consensus: for benchmarking purposes, every consensus method available in DockM8 was used. Moreover, the consensus was considered for every pose selection method and every combination of the selected scoring functions. This represents 621621 possible final compound rankings for each target in the dataset.

3.1.2 DUD-E

A subset (see Supporting Information Section 1) of the DUD-E library (28 targets) was downloaded via the link provided in the original publication [34]. Dataset preparation was carried out in the same manner as for the DEKOIS dataset. The DockM8 conditions were set as follows :

- Protein preparation: the protein file was protonated using the `--prepare_proteins = True` option.
- Pocket definition: the pocket was defined using the `--pocket = reference` option with a cutoff of 8 Å (box size of 16 Å).
- Ligand preparation: the ligands were protonated and 3D conformers were generated using Gypsum-DL.
- Docking: PLANTS, SMINA, GNINA: in all cases, 10 poses per ligand were generated.
- Pose selection: due to the high computational cost of exploring each scoring function as a pose selection tool, we elected to use only the following methods: `RMSD`, `spyRMSD`, `espsim`, `3DScore`, `bestpose`, `bestpose_GNINA`, `bestpose_SMINA`, `bestpose_PLANTS`, `KORP-PL`, `ConvexPLR` and `RTMScore`.
- Pose busting: Due to the high computational cost of using Pose Busters on this many docking poses, we elected to not enable this feature.
- Rescoring: `GNINA-Affinity`, `CNN-Score`, `CNN-Affinity`, `Vinardo`, `AD4`, `KORP-PL`, `ConvexPLR`, `LinF9`, `RTMScore`, `RFScoreVS`, `CHEMPLP`, `NNScore`, `PLECScore` (`SCORCH` and `AAScore` were omitted in the benchmarking used due to high computational cost. `PLP` was omitted due to high correlation with `CHEMPLP`).
- Consensus: for benchmarking purposes, every consensus method available in DockM8 was used. Moreover, the consensus was considered for every pose selection method and every combination of the selected scoring functions. This represents 556179 possible final compound rankings for each target in the dataset.

3.1.3 Lit-PCBA

A subset (see Supporting Information Section 1) of the Lit-PCBA library (seven targets) was downloaded via the link provided in the original publication [36]. For each of the targets, the PDB entry with the best resolution was used. The reference ligand and protein files were used as is in the DockM8 workflow. Dataset preparation was carried out in the same manner as for the DEKOIS dataset. The DockM8 conditions were set as follows:

- Protein preparation: the protein file was protonated using the `--prepare_proteins = True` option.
- Pocket definition: the pocket was defined using the `--pocket = reference` option with a cutoff of 8 Å (box size of 16 Å).
- Ligand preparation: the ligands were protonated and 3D conformers were generated using Gypsum-DL.
- Docking: PLANTS, SMINA, GNINA: in all cases, 10 poses per ligand were generated.
- Pose selection: due to the high computational cost of exploring all the pose selection methods, we elected to use only the following methods: `bestpose`, `bestpose_GNINA`, `bestpose_SMINA`, `bestpose_PLANTS`, `KORP-PL`, `ConvexPLR`.
- Pose busting: Due to the high computational cost of using Pose Busters on this many docking poses, we elected to not enable this feature.
- Rescoring: `GNINA-Affinity`, `CNN-Score`, `CNN-Affinity`, `Vinardo`, `AD4`, `KORP-PL`, `ConvexPLR`, `LinF9`, `RTMScore`, `RFScoreVS`, `CHEMPLP` (`SCORCH`, `AAScore`, `NNScore`, `PLECScore` were omitted in the benchmarking used due to high computational cost. `PLP` was omitted due to high correlation with `CHEMPLP`).
- Consensus: for benchmarking purposes, every consensus method available in DockM8 was used. Moreover, the consensus was considered for every pose selection method and every combination of the selected scoring functions. This represents 69267 possible final compound rankings for each target in the dataset.

4 Results

We utilize two distinct methodologies to evaluate the performance of DockM8. The first metric, `DockM8-max`, involves assessing each target individually by exploring all potential combinations of consensus algorithms, scoring functions, and pose selection strategies to identify and report the best-performing combination for each specific target. Conversely, the `DockM8-best` metric considers the same range of combinations but focuses on the one that delivers the highest average performance across the entire dataset of targets. Thus,

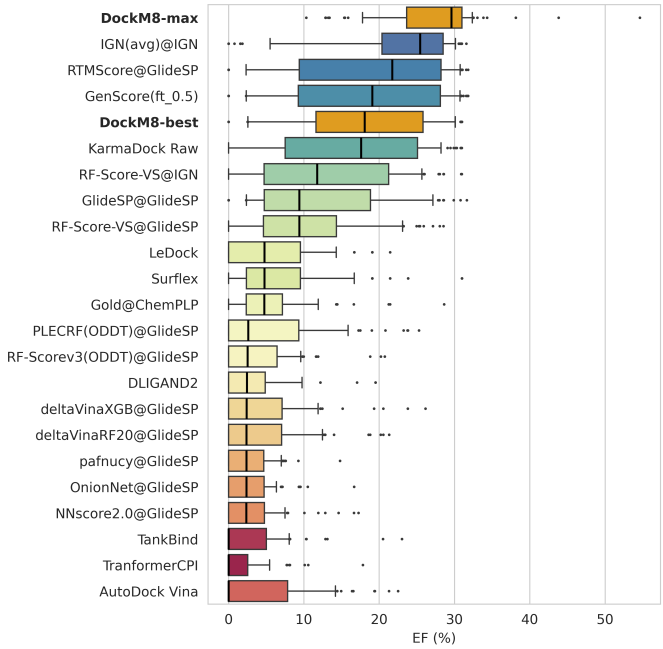

Figure 4: Evaluation of EF at 1% of DockM8 compared to 21 published protocols on the DEKOIS dataset. Methods are sorted by their median value. Whiskers denote the 10th and 90th percentiles. Methods containing the @ symbol refer to either the provenance of the data (@IGN and @Karmadock) or the source of the docking poses, which were used for scoring (@GlideSP).

`DockM8-max` is target-specific, while `DockM8-best` emphasizes overall consistency and generalizability across multiple targets. Performance will be assessed on the three datasets DEKOIS 2.0, DUD-E and Lit-PCBA (see section 3.1). Then, a general analysis of the impact of pose, consensus method and scoring function selection will be performed.

4.0.1 Assessment of screening power on DEKOIS 2.0

In this section, the performance of DockM8 was evaluated on a subset of the DEKOIS 2.0 dataset (79 targets, see Supporting Information). A variety of other methods were included in the comparison, with the enrichment factor (EF) data being taken from the relevant publications. We included a variety of methods in the comparison, including machine learning (IGN [73], KarmaDock [74], RF-Score-VS [63], TankBind [75]) and more traditional methods (GlideSP [7], Surflex [76], LeDock [77], AutoDock Vina[5]) (for all associated data, see the Supporting Information). Figure 4 shows the EF at 1% ($EF_{1\%}$) for the various methods.

Considering median $EF_{1\%}$ values, `DockM8-max` outperforms all state-of-the-art methods with an median $EF_{1\%}$ of 29.6%, while `DockM8-best` lands at rank 5 (18.05%). The other three top-scoring methods are IGN, RTMScore and GenScore, with median $EF_{1\%}$s between 19 and 26%. Another promising result is that the `DockM8-max` combination always results in an $EF_{1\%}$ higher than 10% for any of the targets in the dataset (the lowest being 10.32% for `cpy2a6`), while all other methods, including `DockM8-best`, show poor performance for individual targets (down to $EF_{1\%}$ of 0%). These observations are also supported by the AUC-ROC and BEDROC metrics for which `DockM8-max` achieves median values of 0.813 and 0.633 respectively (see Supporting Information). The generalizability of scoring functions and virtual screening protocols has been a long-standing challenge in computer-aided drug design. From these data, it is clear that DockM8 provides an avenue to partially remediate this problem by providing the user with a large number of possible combinations, allowing the subsequent tailoring of the conditions to the target being studied. The five best (on average according to EF1%) DockM8 methods are shown in Table 3.

4.0.2 Assessment of screening power on DUD-E

We also evaluated the performance of DockM8 on a subset of the DUD-E benchmarking dataset (28 targets, Figure 5). As above, a variety of other methods collected from the literature were included in the comparison, including knowledge-based, empirical, and machine-learning scoring functions (MLSFs), as well as a variety of consensus methodologies (for all associated data, see the Supporting Information). Since the `RFScoreVS` was partially trained on the DUD-E dataset[63], we also report the performance of `DockM8-max-noRF` and

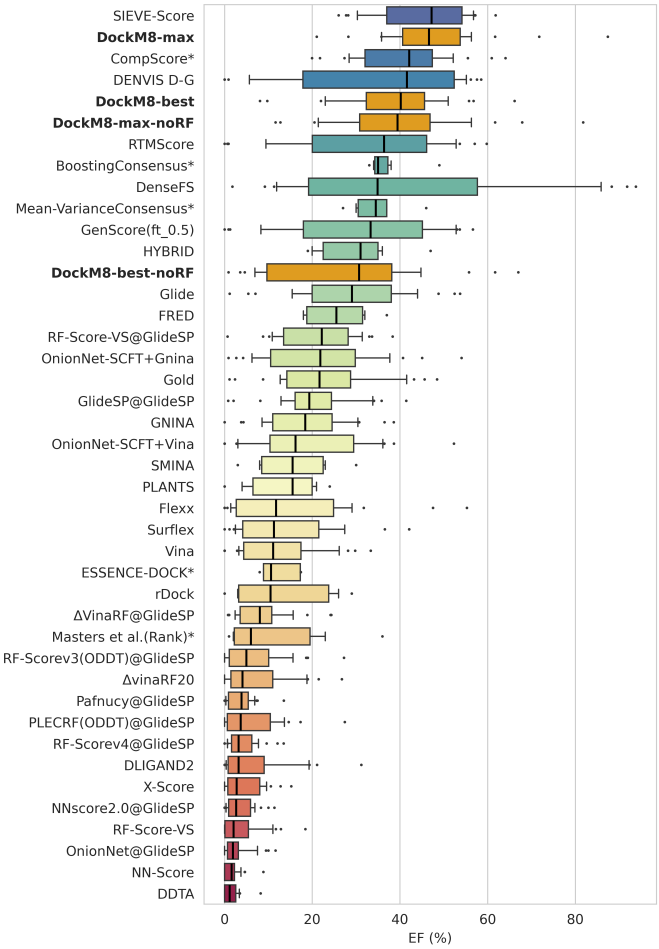

Figure 5: Evaluation of EF at 1% of DockM8 compared to 38 published protocols on 28 targets of the DUD-E dataset (full data for the literature methods is available in the Supporting Information). Methods are sorted by their median value. Whiskers denote the 10th and 90th percentiles. Methods based on a consensus strategy are labeled with an asterisk(*).

`DockM8-best-noRF` which refer to the above methodologies while excluding the `RFScoreVS` scoring function to avoid overestimating the performance of DockM8, due to data leakage when applying RFScoreVS to this data set. `DockM8-max` shows comparable performance to state-of-the-art methods (46.63%) with only `SIEVE-Score` showing better median performance (47.2%). `DockM8-max-noRF` exhibited a slight drop in performance (39.45%) due to the omission of the `RFScoreVS` scoring function while still outperforming state-of-the-art methods such as RTMScore (36.38%). `DockM8-best-noRF` and `DockM8-best-noRF` achieved median $EF_{1\%}$ of 40.2% and 30.66% respectively. As with the DEKOIS dataset, `DockM8-max` outperforms nearly every other method in terms of generalizability, with the lowest $EF_{1\%}$ of 21% being obtained for `hdac2`. This again highlights the ability of the DockM8 workflow to be customized to the target under consideration. These observations are also supported by the AUC-ROC and BEDROC metrics for which `DockM8-max` achieves median values of 0.891 and 0.653 respectively (see Supporting Information). It is worth noting that the removal of the `RFScoreVS` scoring did lead to an appreciable loss in performance for both `DockM8-max-noRF` and `DockM8-best-noRF`. This is not in itself surprising as a scoring function trained on most of the dataset would be expected to be a strong contributor to the best average combination of DockM8 conditions. However, we do note the significant performance increase afforded by our consensus methodologies relative to using the `RFScoreVS` scoring function on its own (which achieved a median performance of 22.17%). The five best (on average according to EF1%) DockM8 methods are shown in Table 3.

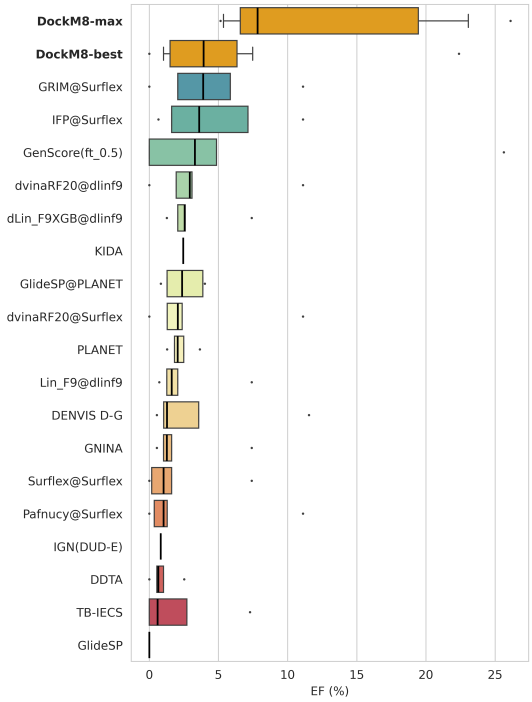

Figure 6: Evaluation of EF at 1% of DockM8 compared to 18 published docking protocols on 7 targets of the Lit-PCBA dataset (details in Supporting Information). Methods are sorted by their median value. Whiskers denote the 10th and 90th percentiles.

4.0.3 Assessment of screening power on Lit-PCBA

We evaluated the performance of DockM8 on a subset of the challenging Lit-PCBA dataset (seven targets, Figure 6). Lit-PCBA was designed to be an unbiased dataset for the evaluation of docking protocols and scoring functions, is based on real bioassay data, and aims to mimic experimental screening conditions. As above, a variety of other methods collected from the literature were included in the comparison (for all associated data, see the Supporting Information). In this experiment, DockM8 outperforms all other state-of-the-art methods in the literature. The median $EF_{1\%}$ for both `DockM8-max` with 7.83% and `DockM8-best` with 3.93% surpassed those of all other models for which enrichment factor data was accessible. Consistent with our findings on the previous two benchmarking datasets, The superior behavior of `DockM8-max` in generalizability is even more pronounced in this study, exemplified by its lowest enrichment rate of 5.15% for the `mtorc1` target, which lies even above the median $EF_{1\%}$ value for all other methods. This further underscores the ability of the DockM8 workflow to be adapted to the target under consideration. These observations are also supported by the AUC-ROC and BEDROC metrics for which `DockM8-max` achieves median values of 0.623 and 0.107 respectively (see Supporting Information). The five best (on average according to EF1%) DockM8 methods are shown in Table 3.

4.0.4 Evaluation of the performance of pose selection, consensus method, and scoring function selection

We then focused on evaluating the impact of the selection of consensus methods, pose selection methods, scoring function selection, and the number of scoring functions used during the consensus procedure on DockM8's performance. For the rest of this discussion, we define a "DockM8 combination" as a unique combination of a pose selection method, a consensus method, and one or more scoring functions. For each DockM8 combination, we computed the average performance across all targets within the given dataset, using the $EF_{1\%}$ metric as a standard measure. Subsequently, we recorded the frequency with which specific pose selection methods, consensus methods, or scoring functions ranked among the top DockM8 methods when sorted by the $EF_{1\%}$ metric. These rankings were established at several performance thresholds: 1%, 0.1%, and 0.01%. For instance, if we had 1000 DockM8 combinations, the top 1% threshold would include the 10 combinations with the highest $EF_{1\%}$ values. This method was chosen as opposed to calculating the average performance of each method across the whole dataset, which would be less indicative of top-level performance and may favor poorer-performing methods.

Table 3: Performance metrics of the top-5 best DockM8 methods (according to the $EF_{1\%}$ metric) for the three benchmark datasets. Performance metrics reported in the table are averages across all targets in the respective dataset.

Dataset	Pose Selection	Consensus	SFs	EF1%	EF0.5%	EF5%	AUCROC	BEDROC
DEKOIS	bestpose	Zscore_best	CHEMPLP_CNN-Score_KORP-PL_RTMScore_GNINA-Affinity	18.02	20.28	7.51	0.75	0.47
	bestpose	Zscore_best	CNN-Score_KORP-PL_RTMScore_GNINA-Affinity	18.02	19.5	7.79	0.76	0.48
	KORP-PL	Zscore_best	CNN-Score_KORP-PL_RTMScore_Vinardo	17.97	19.97	7.83	0.77	0.48
	KORP-PL	Zscore_best	CNN-Score_KORP-PL_RTMScore_GNINA-Affinity	17.87	19.91	7.86	0.77	0.48
	KORP-PL	Zscore_best	CHEMPLP_CNN-Score_KORP-PL_RTMScore_Vinardo	17.84	19.78	7.56	0.75	0.47
DUD-E	bestpose	Zscore_best	KORP-PL_CNN-Score_RFScoreVS_Vinardo_RTMScore	38.32	49.61	11.87	0.86	0.59
	bestpose	Zscore_best	KORP-PL_CHEMPLP_CNN-Score_RFScoreVS_GNINA-Affinity_RTMScore	38.26	49.18	11.63	0.85	0.59
	bestpose	Zscore_best	KORP-PL_CHEMPLP_CNN-Score_RFScoreVS_Vinardo_GNINA-Affinity_RTMScore	38.18	49.39	11.54	0.85	0.58
	bestpose	Zscore_best	KORP-PL_CNN-Score_RFScoreVS_GNINA-Affinity_RTMScore	38.16	49.25	11.86	0.86	0.59
	bestpose	Zscore_best	KORP-PL_CHEMPLP_CNN-Score_RFScoreVS_Vinardo_RTMScore	38.16	49.83	11.62	0.85	0.59
Lit-PCBA	bestpose_PLANTS	RbR_best	RFScoreVS_ConvexPLR_CNN-Score_RTMScore_Vinardo_GNINA-Affinity	6	3.77	2.4	0.61	0.07
	KORP-PL	RbR_best	RFScoreVS_CNN-Score_RTMScore	5.93	6.48	2.39	0.59	0.07
	KORP-PL	RbR_best	RFScoreVS_CNN-Affinity_CNN-Score_RTMScore	5.84	7.59	2.85	0.62	0.07
	bestpose	RbR_avg	RFScoreVS_CNN-Affinity_CNN-Score_KORP-PL_LinF9	5.77	2.72	2.48	0.65	0.06
	KORP-PL	RbR_best	CNN-Affinity_CNN-Score_RTMScore	5.75	4.42	3.14	0.62	0.07

Figure 7 illustrates the frequency of occurrence of the DockM8 pose selection methods on the DEKOIS, DUD-E, and LIT-PCBA datasets. From these data, we can see that in general, the scoring-based pose selection methods (`KORP-PL`, `RTMScore`, `ConvexPLR` and the `bestpose_*` methods) outperform the descriptor-based ones (`spyRMSD`, `RMSD` and `espsim`) on both the DEKOIS and DUD-E datasets. Indeed, the descriptor-based methods do not appear in the 0.01% bracket and only appear rarely in the 0.1% bracket. Due to computational limitations, the descriptor-based methods were not tested on the Lit-PCBA dataset. The `KORP-PL` and `bestpose` methods appear the most often in the 0.01% for the DEKOIS (39 and 23 counts respectively) dataset, while the `bestpose` method was identified as being the top performer in the DUD-E dataset (55 counts in the 0.01% threshold). Additionally, the `bestpose_PLANTS` method is significantly more represented in the top brackets for the Lit-PCBA dataset. Although we observe a clear advantage in using certain pose selection methods on each of the three datasets, the presence, and nature of a relationship between the type of dataset (decoy or experimental) and the preferred pose selection method requires further investigation.

We then investigated the performance of the various consensus methods available in DockM8. From Figure 8, we can see that the `Zscore_best` method outperforms the other methods on both the DUD-E and DEKOIS datasets (counts of 59 and 55 in the top 0.01% bracket respectively). For the Lit-PCBA dataset, the `RbR_best` method significantly outperformed the rest (only method represented in the top 0.01% threshold with 6 counts). Interestingly, the Lit-PCBA dataset seems to show different trends relative to the decoy-based datasets (DEKOIS and DUD-E), which we also observed in our above analysis of the pose selection methods. This may reflect inherent differences in the nature of the data and chemical space included in these datasets. Overall, we can observe that the `avg` methods tend to perform more poorly than their respective `best` counterpart. This points to the fact that using the best pose (regardless of how the best pose is determined) is a superior strategy to aggregating information from multiple poses.

To gauge the relative performance and contributions of the various scoring functions, we performed a similar analysis, which is shown in Figure 9. On the DEKOIS dataset, the top performers were `KORP-PL`, `RTMScore` and `CNN-Score`, showing significant presence in the top 0.01% the dataset (62, 62, and 61 counts respectively). In contrast, we found `NNScore`, `PLECscore`, `LinF9`, and `CNN-Affinity` to be relatively poor performers. On DUD-E, `KORP-PL` and `RFScoreVS` were the highest-performing scoring functions, closely followed by `CNN-Score`. This is not surprising concerning `RFScoreVS` as DUD-E was used as the training set for this scoring function and higher relative performance is therefore expected. With that in mind, `KORP-PL`, and `CNN-Score` still showed good performance. As we found for the DEKOIS dataset, `NNScore`, `PLECscore`, `LinF9`, and `CNN-Affinity` all performed poorly on DUD-E. On Lit-PCBA, `RFScoreVS`, `CNN-Score`, `KORP-PL` and `ConvexPLR` showed the best performance although most of the scoring functions were fairly closely ranked. From this analysis, we can conclude that `KORP-PL`, `CNN-Score`, `RTMScore` and `RFScoreVS` showed the highest overall performance across all three datasets while `CNN-Affinity`, `NNScore` and `PLECScore` showed markedly lower performance. Additionally, it is clear that no single scoring function consistently outperforms the rest, which is expected as scoring functions can be highly target-specific.

Additionally, we investigated how the number of scoring functions taken into account during the consensus affects the enrichment performance (Figure 10). For all studied datasets, a Gaussian distribution was observed showing that a consensus of five to seven scoring functions was significantly more represented in the top 0.01% and 0.1% for each of the datasets we studied. These data indicate that including too many scoring functions in the consensus can decrease enrichment performance due to increased noise and disagreement. Given the substantial computational cost, we recommend using more than five scoring functions in DockM8 only if proven beneficial for the specific target.

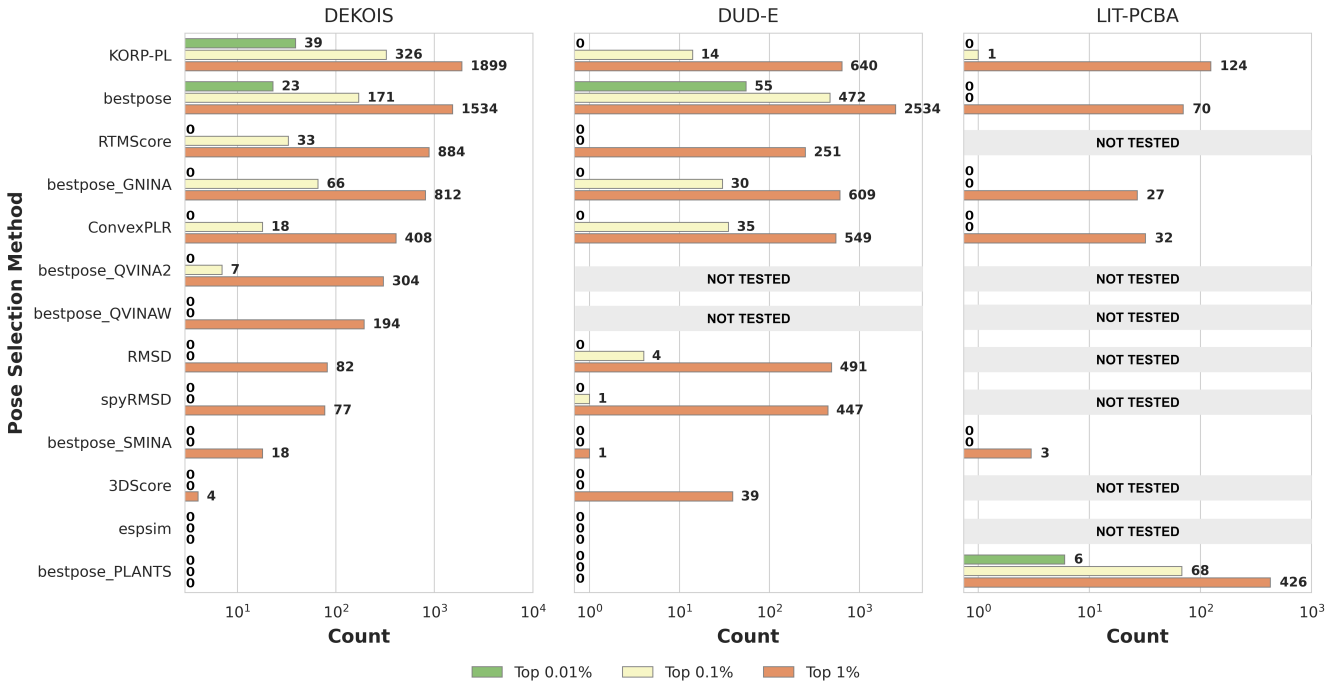

Figure 7: Frequency of DockM8 pose selection methods at top performance thresholds (1%, 0.1%, 0.01%) on the DEKOIS, DUD-E, and LIT-PCBA datasets.

5 Discussion

User-friendly open-source tools: In medicinal chemistry, using virtual-screening (VS) software is crucial but challenging. Open-source tools often require programming knowledge and are rarely updated, limiting their adoption by chemists with limited computational skills. Commercial tools, while user-friendly, are expensive and inaccessible to smaller academic groups. Our analysis shows that commercial tools do not necessarily perform better on our datasets. To address these issues, we introduce DockM8, a user-friendly framework that requires minimal programming expertise and uses open-source components. This approach makes virtual screening more accessible and challenges the idea that higher cost equates to higher quality. DockM8 aims to provide an easy-to-use VS workflow with extensive documentation for operation and installation on various platforms. It includes a basic graphical user interface to minimize coding and command-line usage.

Performance of DockM8: Our assessment of DockM8 across diverse datasets—DEKOIS 2.0, DUD-E, and Lit-PCBA—shows its superior performance in virtual screening (VS). DockM8-max consistently outperforms most other methods in enrichment factor and ranks among the best for AUC-ROC and BEDROC metrics (see Supporting Information). Unlike other methods that perform poorly on several targets, `DockM8-max` demonstrates excellent generalizability, never falling below 10.3% enrichment at 1% in DEKOIS and 21% in DUD-E. Even advanced methods like `IGN` [73], `KarmaDock` [74], `SIEVE-Score` [78], `RTMScore` [11], and `CompScore` [24] lack this level of consistency. DockM8's flexibility allows tuning for each target, provided data on the target and its known binders is available. To address data limitations, DockM8 can generate decoys based on known active compounds, offering an optimization pathway even with limited data.

Impact of structure resolution: Both `DockM8-max` and `DockM8-best` demonstrated exceptional performance on the Lit-PCBA dataset, with `DockM8-max`'s EF1% consistently above 5.15%. While our decision to use the highest resolution available in this dataset may impact these results, this strategy aligns with SBVS best practices. As Tran-Nguyen et al. observed, scoring performance can indeed vary based on the crystal structure used [79]. We are confident in our workflow's inherent strengths beyond structure resolution and aim to further validate its robustness across various structural resolutions in future studies.

Pose selection strategy: Our analysis of the performance of the various pose selection methods revealed no clear superiority of one method over others (Figure 7). This is further highlighted by the fact that the target-specific `DockM8-max` methods (see Supplementary Information) do not show a preference for any of

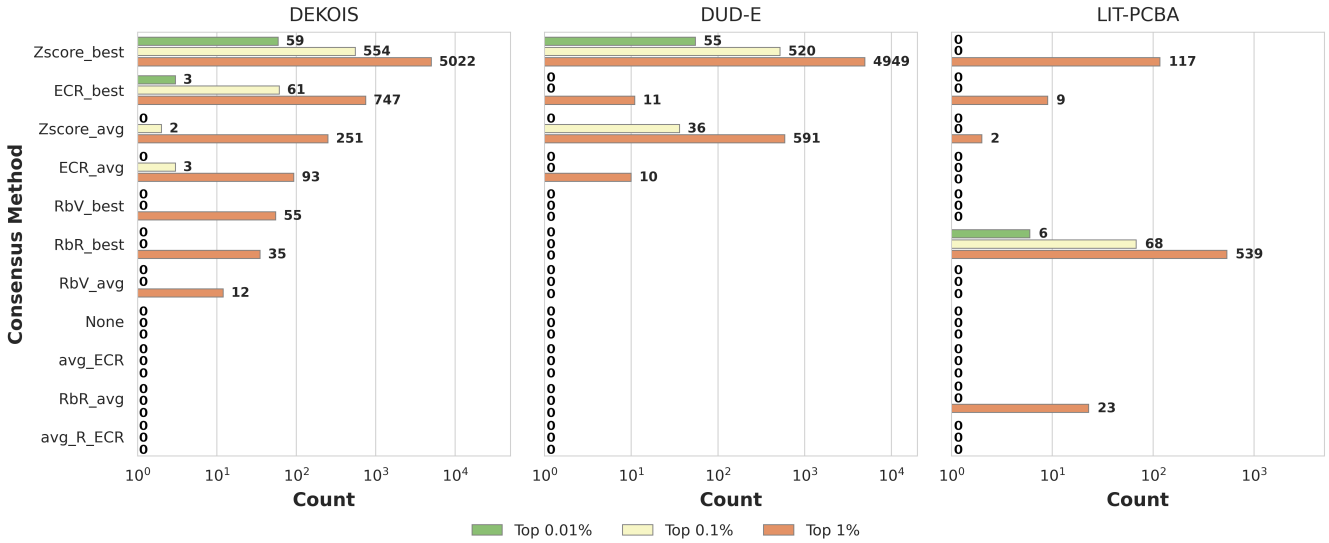

Figure 8: Frequency of DockM8 consensus methods at top performance thresholds (1%, 0.1%, 0.01%) on the DEKOIS, DUD-E, and LIT-PCBA datasets.

the pose selection methods, and this remains true for all three datasets we tested. This indicates that most pose selection methods can be used successfully for VS, although careful benchmarking with known actives and inactives, or using a decoy set is necessary to determine the optimal conditions for a new target.

Choice of consensus method: We observed more variation in performance regarding consensus methods. The `avg` methods consistently performed worse across all three datasets compared to their `best` equivalents, indicating that including information from multiple poses may not benefit virtual screening performance (Figure 8). Specifically, `Zscore_best` performed best on the DUD-E and DEKOIS datasets, while `RbR_best` excelled on the challenging Lit-PCBA dataset. This variation might be due to differences in dataset size and chemical space or our use of subsets rather than complete datasets. This highlights the importance of customizing the VS approach to each target's characteristics. Our findings suggest that the choice of consensus method should be tailored to the specific demands of each target, recognizing that each dataset and target may influence the efficacy of different methods.

Number of scoring functions used: Increasing the number of scoring functions in the consensus does not yield a linear performance increase, with a decrease in performance observed at higher counts. Our data suggests that using four to five scoring functions offers the most reliable performance (Figure 10), likely due to increasing disagreement among functions as their number rises. This finding is beneficial from a computational cost and data analysis perspective. However, for some targets, more than five scoring functions provided the best performance. As with our previous observations and recent literature [80], we recommend adapting the approach to the specific target whenever possible.

Usage recommendation: We acknowledge that although DockM8 was developed as an accessible and easy-to-use tool, the impressive number of combinations of pose selection, consensus, and scoring methods can seem daunting. In light of our data, and if biologically validated active compounds are not available, we can recommend the use of the following settings :

- Consensus: The `Zscore_best` and `RbR_best` methods showed superior performance in all of our test cases, making them a good starting point for a prospective VS campaign.
- Pose selection: `KORP-PL` performed consistently well on all datasets, likely providing a reliable starting point for VS on new targets.
- Docking selection: While we did not specifically compare docking programs, we can derive some data from the `bestpose_*` methods as these take the pose with the best score from the selected docking program. `GNINA` offers the best performance at the expense of higher computation time. `PLANTS` is recommended if computation time is crucial, as its performance can be improved with suitable consensus and scoring methods.
- Scoring functions: For a VS campaign, we recommend using three to six scoring functions for optimal balance between performance and computational cost. The scoring functions `KORP-PL`, `ConvexPLR` and

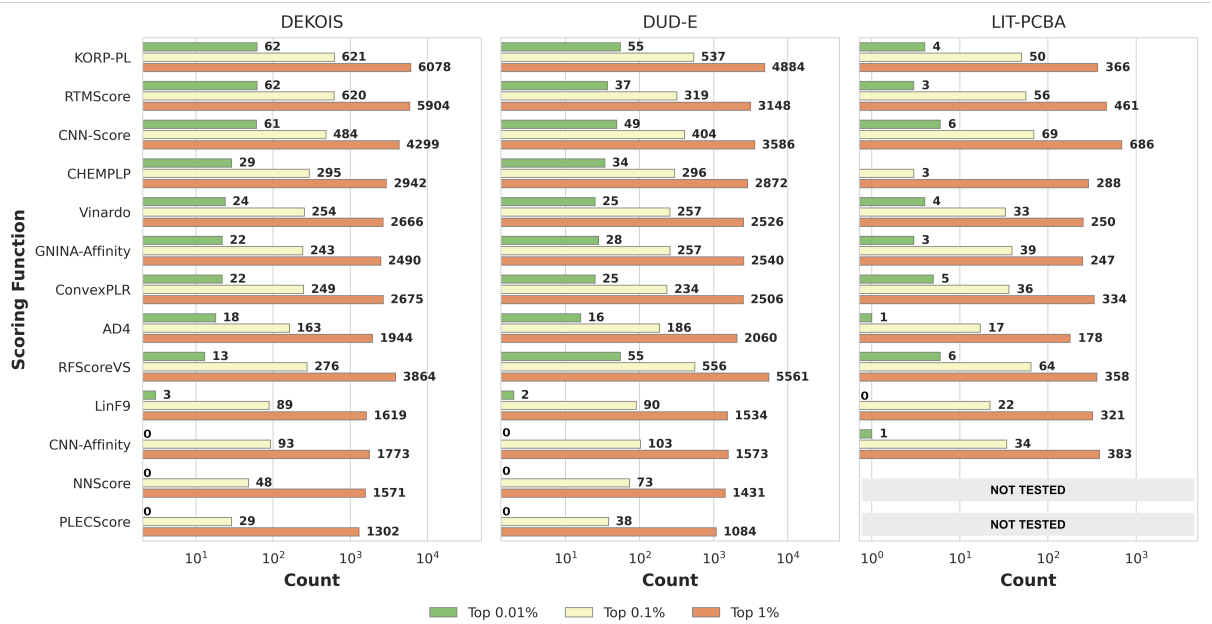

Figure 9: Frequency of DockM8 scoring functions at top performance thresholds (1%, 0.1%, 0.01%) on the DEKOIS, DUD-E, and LIT-PCBA datasets.

`CNN-Score` provide good performance at reasonable computational costs. `RTMScore` or `RFScoreVS` can enhance performance but significantly increase computation time. Whenever available, known well-performing scoring functions on the target at hand should be included in the consensus.

DockM8 continued: While DockM8 demonstrates excellent performance, several areas for improvement remain. Enhancing speed could be achieved by transitioning from multiple `.sdf` files to a database system[81], and incorporating novel machine-learning docking methods like Karmadock [74] or UMol [82] to expedite the time-intensive docking step. Integrating sophisticated scoring functions and pre-filtering techniques, such as pharmacophore models or fingerprint similarity, could further refine performance. Additionally, adopting recent docking score prediction and active-learning approaches like HASTEN [83] and DeepDocking [84] could facilitate rapid virtual screening of extensive chemical libraries. Improvements to the graphical user interface (GUI) are also planned to make data visualization and interaction more intuitive. We encourage the computational chemistry community to contribute to DockM8's development via our GitHub repository.

6 Conclusions

In this work, we introduced a novel open-source consensus docking workflow. We presented our comprehensive evaluation of DockM8, using datasets such as DEKOIS 2.0, DUD-E, and Lit-PCBA. We demonstrated its competitive performance in VS over existing methods, including state-of-the-art machine learning and advanced consensus approaches. The versatility and adaptability of DockM8, characterized by its ability to handle a variety of docking algorithms and scoring functions, enable it to achieve high levels of generalizability and performance across diverse targets. This is a notable advancement over existing tools, which often show limited applicability across different targets.

Furthermore, DockM8 addresses critical challenges in the field, particularly the accessibility and usability of VS tools in academic research. Its user-friendly interface, minimal programming requirements, and open-source nature democratize the use of advanced VS methods. This approach not only makes cutting-edge computational drug discovery tools more accessible to a wider range of scientists but also challenges the prevailing notion that cost is indicative of quality in software solutions.

Despite its current strengths, there remain opportunities for further enhancement of the DockM8 workflow. Areas such as the speed of the procedure, integration of advanced machine-learning docking and active-learning algorithms, and improvements to the GUI are identified as key directions for future development. The potential inclusion of pre-filtering steps and the integration of novel docking score prediction methodologies could further augment the efficacy and efficiency of DockM8 in handling large compound libraries.

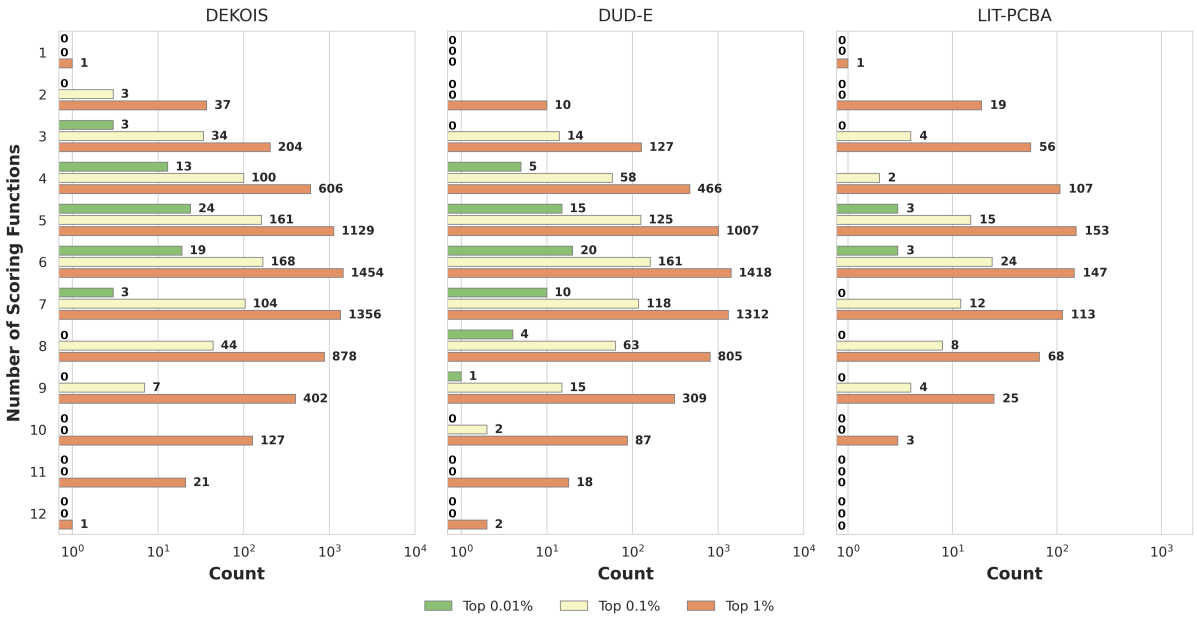

Figure 10: Frequency of the number of scoring functions used in the DockM8 consensus at top performance thresholds (1%, 0.1%, 0.01%) on the DEKOIS, DUD-E, and LIT-PCBA datasets.

Given these findings, we invite the computational and medicinal chemistry communities to participate in the ongoing development of DockM8. Its open-source nature and our commitment to collaboration ensure continuous innovation in virtual drug screening. By leveraging collective expertise, we aim to make DockM8 an even more powerful tool, significantly advancing the field.

7 Data Availability

All the prepared molecules, docking poses, pose selection results, restoring and consensus results as well as performance metrics have been made available on Zenodo under `https://doi.org/10.5281/zenodo.11191685`. This repository also contains further benchmarking plots and the reference literature data used to generate them. We hope this is a useful resource for the development of new scoring functions and for the training of pose prediction models.

The raw benchmark datasets are available from the original sources: DEKOIS at `https://www.pharmchem.uni-tuebingen.de/dekois/`, DUD-E at `https://dude.docking.org/` and Lit-PCBA at `https://drugdesign.unistra.fr/LIT-PCBA/index.html/`.

8 Code Availability

The source code is available on GitHub: `https://github.com/DrugBud-Suite/DockM8`.

9 Competing interests

No competing interest is declared.

10 Author contributions statement

A.L., A.V. and A.H. conceptualized the study. A.L. conceived the experiment(s) and wrote the code, H.I. participated in writing some sections of the code, A.L. analyzed the results. A.L. wrote the manuscript. A.V. and A.H supervised the study and reviewed the manuscript.

11 Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 860816 (MepAnti). Additional funding was pro-

vided by the Google Cloud Educational program. We thank Mario Szeles and Prof. Dr. Olga Kalinina for valuable discussions on the project. A.V. acknowledges funding through the NextAID project.

References

[1] Evanthia Lionta, George Spyrou, Demetrios Vassilatis, and Zoe Cournia. Structure-Based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. *Current Topics in Medicinal Chemistry*, 14(16):1923–1938, October 2014.

[2] Mohd Danishuddin and Asad U. Khan. Structure based virtual screening to discover putative drug candidates: Necessary considerations and successful case studies. *Methods*, 71:135–145, January 2015.

[3] Andrew T. McNutt, Paul Francoeur, Rishal Aggarwal, Tomohide Masuda, Rocco Meli, Matthew Ragoza, Jocelyn Sunseri, and David Ryan Koes. GNINA 1.0: molecular docking with deep learning. *Journal of Cheminformatics*, 13(1):1–20, December 2021. Publisher: BioMed Central Ltd.

[4] Garrett M. Morris, Ruth Huey, William Lindstrom, Michel F. Sanner, Richard K. Belew, David S. Goodsell, and Arthur J. Olson. AutoDock4 and AutoDockTools4: Automated Docking with Selective Receptor Flexibility. *Journal of computational chemistry*, 30(16):2785–2791, December 2009.

[5] Oleg Trott and Arthur J. Olson. 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, 2010. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21334.

[6] Jerome Eberhardt, Diogo Santos-Martins, Andreas F. Tillack, and Stefano Forli. AutoDock Vina 1.2.0: New Docking Methods, Expanded Force Field, and Python Bindings. *Journal of Chemical Information and Modeling*, 61(8):3891–3898, August 2021. Publisher: American Chemical Society.

[7] Richard A. Friesner, Jay L. Banks, Robert B. Murphy, Thomas A. Halgren, Jasna J. Klicic, Daniel T. Mainz, Matthew P. Repasky, Eric H. Knoll, Mee Shelley, Jason K. Perry, David E. Shaw, Perry Francis, and Peter S. Shenkin. Glide: A New Approach for Rapid, Accurate Docking and Scoring. *Journal of Medicinal Chemistry*, 47(7):1739–1749, March 2004. Publisher: American Chemical Society.

[8] Jie Liu and Renxiao Wang. Classification of Current Scoring Functions. *Journal of Chemical Information and Modeling*, 55(3):475–482, March 2015. Publisher: American Chemical Society.

[9] Jin Li, Ailing Fu, and Le Zhang. An Overview of Scoring Functions Used for Protein–Ligand Interactions in Molecular Docking. *Interdisciplinary Sciences – Computational Life Sciences*, 11(2):320–328, June 2019. Publisher: Springer Berlin Heidelberg.

[10] Gregory L. Warren, C. Webster Andrews, Anna-Maria Capelli, Brian Clarke, Judith LaLonde, Millard H. Lambert, Mika Lindvall, Neysa Nevins, Simon F. Semus, Stefan Senger, Giovanna Tedesco, Ian D. Wall, James M. Woolven, Catherine E. Peishoff, and Martha S. Head. A Critical Assessment of Docking Programs and Scoring Functions. *Journal of Medicinal Chemistry*, 49(20):5912–5931, October 2006. Publisher: American Chemical Society.

[11] Chao Shen, Xujun Zhang, Yafeng Deng, Junbo Gao, Dong Wang, Lei Xu, Peichen Pan, Tingjun Hou, and Yu Kang. Boosting Protein–Ligand Binding Pose Prediction and Virtual Screening Based on Residue–Atom Distance Likelihood Potential and Graph Transformer. *Journal of Medicinal Chemistry*, 65(15):10691–10706, August 2022. Publisher: American Chemical Society.

[12] Rommie E. Amaro, Jerome Baudry, John Chodera, Özlem Demir, J. Andrew McCammon, Yinglong Miao, and Jeremy C. Smith. Ensemble Docking in Drug Discovery. *Biophysical Journal*, 114(10):2271–2278, May 2018.

[13] J. Christian Baber, William A. Shirley, Yinghong Gao, and Miklos Feher. The use of consensus scoring in ligand-based virtual screening. In *Journal of Chemical Information and Modeling*, volume 46, pages 277–288. American Chemical Society, January 2006. Issue: 1 ISSN: 1549960X.

[14] Paul S. Charifson, Joseph J. Corkery, Mark A. Murcko, and W. Patrick Walters. Consensus Scoring: A Method for Obtaining Improved Hit Rates from Docking Databases of Three-Dimensional Structures into Proteins. *Journal of Medicinal Chemistry*, 42(25):5100–5109, December 1999. Publisher: American Chemical Society.

[15] Douglas R. Houston and Malcolm D. Walkinshaw. Consensus docking: Improving the reliability of docking in a virtual screening context. *Journal of Chemical Information and Modeling*, 53(2):384–390, February 2013. Publisher: American Chemical Society.

[16] Renxiao Wang and Shaomeng Wang. How Does Consensus Scoring Work for Virtual Library Screening? An Idealized Computer Experiment. *Journal of Chemical Information and Computer Sciences*, 41(5):1422–1426, September 2001. Publisher: American Chemical Society.

[17] Ryan G. Coleman, Michael Carchia, Teague Sterling, John J. Irwin, and Brian K. Shoichet. Ligand Pose and Orientational Sampling in Molecular Docking. *PLOS ONE*, 8(10):e75992, October 2013. Publisher: Public Library of Science.

[18] Jordane Preto and Francesco Gentile. Assessing and improving the performance of consensus docking strategies using the DockBox package. *Journal of Computer-Aided Molecular Design*, 33(9):817–829, September 2019.

[19] Ni Liu and Zhibin Xu. Using LeDock as a docking tool for computational drug design. *IOP Conference Series: Earth and Environmental Science*, 218(1):012143, January 2019. Publisher: IOP Publishing.

[20] Sergio Ruiz-Carmona, Daniel Alvarez-Garcia, Nicolas Foloppe, A. Beatriz Garmendia-Doval, Szilveszter Juhos, Peter Schmidtke, Xavier Barril, Roderick E. Hubbard, and S. David Morley. rDock: A Fast, Versatile and Open Source Program for Docking Ligands to Proteins and Nucleic Acids. *PLOS Computational Biology*, 10(4):e1003571, April 2014. Publisher: Public Library of Science.

[21] David Ryan Koes, Matthew P. Baumgartner, and Carlos J. Camacho. Lessons Learned in Empirical Scoring with smina from the CSAR 2011 Benchmarking Exercise. *Journal of Chemical Information and Modeling*, 53(8):1893–1904, August 2013. Publisher: American Chemical Society.

[22] Rodrigo Ochoa, Karen Palacio-Rodriguez, Camila M. Clemente, and Natalia S. Adler. dockECR: Open consensus docking and ranking protocol for virtual screening of small molecules. *Journal of Molecular Graphics and Modelling*, 109:108023, December 2021. Publisher: Elsevier.

[23] Izaz Monir Kamal and Saikat Chakrabarti. MetaDOCK: A Combinatorial Molecular Docking Approach. *ACS Omega*, 8(6):5850–5860, February 2023.

[24] Yunierkis Perez-Castillo, Stellamaris Sotomayor-Burneo, Karina Jimenes-Vargas, Mario Gonzalez-Rodriguez, Maykel Cruz-Monteagudo, Vinicio Armijos-Jaramillo, M. Natália D. S. Cordeiro, Fernanda Borges, Aminael Sánchez-Rodríguez, and Eduardo Tejera. CompScore: Boosting Structure-Based Virtual Screening Performance by Incorporating Docking Scoring Function Components into Consensus Scoring. *Journal of Chemical Information and Modeling*, 59(9):3655–3666, September 2019. Publisher: American Chemical Society.

[25] Jochem Nelen, Miguel Carmena-Bargueño, Carlos Martínez-Cortés, Alejandro Rodríguez-Martínez, José Manuel Villalgordo-Soto, and Horacio Pérez-Sánchez. ESSENCE-Dock: A Consensus-Based Approach to Enhance Virtual Screening Enrichment in Drug Discovery, November 2023.

[26] Serena Rosignoli and Alessandro Paiardini. DockingPie: a consensus docking plugin for PyMOL. *Bioinformatics*, 38(17):4233–4234, September 2022.

[27] Schrödinger, LLC. The PyMOL Molecular Graphics System, November 2015.

[28] Dariusz Plewczynski, Michał Łażniewski, Marcin Von Grotthuss, Leszek Rychlewski, and Krzysztof Ginalski. VoteDock: Consensus docking method for prediction of protein–ligand interactions. *Journal of Computational Chemistry*, 32(4):568–581, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/jcc.21642.

[29] Jeff Guo, Jon Paul Janet, Matthias R. Bauer, Eva Nittinger, Kathryn A. Giblin, Kostas Papadopoulos, Alexey Voronov, Atanas Patronov, Ola Engkvist, and Christian Margreitter. DockStream: a docking wrapper to enhance de novo molecular design. *Journal of Cheminformatics*, 13(1):89, November 2021.

[30] David E. Graff and Connor W. Coley. pyscreener: A Python Wrapper for Computational Docking Software. *Journal of Open Source Software*, 7(71):3950, March 2022. arXiv:2112.10575 [q-bio].

[31] Diego E. Barreto Gomes, Katia Galentino, Marion Sisquellas, Luca Monari, Cédric Bouysset, and Marco Cecchini. ChemFlowFrom 2D Chemical Libraries to Protein–Ligand Binding Free Energies. *Journal of Chemical Information and Modeling*, 63(2):407–411, January 2023. Publisher: American Chemical Society.

[32] Guzel Minibaeva, Aleksandra Ivanova, and Pavel Polishchuk. EasyDock: customizable and scalable docking tool. *Journal of Cheminformatics*, 15(1):102, November 2023.

[33] Lianming Du, Chaoyue Geng, Qianglin Zeng, Ting Huang, Jie Tang, Yiwen Chu, and Kelei Zhao. Dockey: a modern integrated tool for large-scale molecular docking and virtual screening. *Briefings in Bioinformatics*, 24(2):bbad047, March 2023.

[34] Michael M. Mysinger, Michael Carchia, John. J. Irwin, and Brian K. Shoichet. Directory of Useful Decoys, Enhanced (DUD-E): Better Ligands and Decoys for Better Benchmarking. *Journal of Medicinal Chemistry*, 55(14):6582–6594, July 2012. Publisher: American Chemical Society.

[35] Matthias R. Bauer, Tamer M. Ibrahim, Simon M. Vogel, and Frank M. Boeckler. Evaluation and Optimization of Virtual Screening Workflows with DEKOIS 2.0 – A Public Library of Challenging Docking Benchmark Sets. *Journal of Chemical Information and Modeling*, 53(6):1447–1462, June 2013. Publisher: American Chemical Society.

[36] Viet Khoa Tran-Nguyen, Célien Jacquemard, and Didier Rognan. LIT-PCBA: An Unbiased Data Set for Machine Learning and Virtual Screening. *Journal of chemical information and modeling*, 60(9):4263–4273, September 2020. Publisher: J Chem Inf Model.

[37] A. Patrícia Bento, Anne Hersey, Eloy Félix, Greg Landrum, Anna Gaulton, Francis Atkinson, Louisa J. Bellis, Marleen De Veij, and Andrew R. Leach. An open source chemical structure curation pipeline using RDKit. *Journal of Cheminformatics*, 12(1):51, September 2020.

[38] RDKit: Open-source cheminformatics (Q3 2023 Release).

[39] Patrick J. Ropp, Jacob O. Spiegel, Jennifer L. Walker, Harrison Green, Guillermo A. Morales, Katherine A. Milliken, John J. Ringe, and Jacob D. Durrant. Gypsum-DL: an open-source program for preparing small-molecule libraries for structure-based virtual screening. *Journal of Cheminformatics*, 11(1):34, May 2019.

[40] Rainer Fährrolfes, Stefan Bietz, Florian Flachsenberg, Agnes Meyder, Eva Nittinger, Thomas Otto, Andrea Volkamer, and Matthias Rarey. ProteinsPlus: a web portal for structure analysis of macromolecules. *Nucleic Acids Research*, 45(W1):W337–W343, July 2017.

[41] Stefan Bietz, Sascha Urbaczek, Benjamin Schulz, and Matthias Rarey. Protoss: a holistic approach to predict tautomers and protonation states in protein-ligand complexes. *Journal of Cheminformatics*, 6(1):12, April 2014.

[42] Wei P. Feinstein and Michal Brylinski. Calculating an optimal box size for ligand docking and virtual screening against experimental and predicted binding pockets. *Journal of Cheminformatics*, 7(1):18, December 2015.

[43] Andrea Volkamer, Daniel Kuhn, Friedrich Rippmann, and Matthias Rarey. DoGSiteScorer: a web server for automatic binding site prediction, analysis and druggability assessment. *Bioinformatics*, 28(15):2074–2075, August 2012.

[44] Nafisa M. Hassan, Amr A. Alhossary, Yuguang Mu, and Chee-Keong Kwoh. Protein-Ligand Blind Docking Using QuickVina-W With Inter-Process Spatio-Temporal Integration. *Scientific Reports*, 7(1):15451, November 2017. Number: 1 Publisher: Nature Publishing Group.

[45] Amr Alhossary, Stephanus Daniel Handoko, Yuguang Mu, and Chee-Keong Kwoh. Fast, accurate, and reliable molecular docking with QuickVina 2. *Bioinformatics*, 31(13):2214–2216, July 2015.

[46] Oliver Korb, Thomas Stützle, and Thomas E. Exner. Empirical scoring functions for advanced protein-ligand docking with PLANTS. *Journal of Chemical Information and Modeling*, 49(1):84–96, January 2009.

[47] forlilab/Meeko, May 2024. original-date: 2020-11-07T12:05:36Z.

[48] Noel M. O'Boyle, Michael Banck, Craig A. James, Chris Morley, Tim Vandermeersch, and Geoffrey R. Hutchison. Open Babel: An open chemical toolbox. *Journal of Cheminformatics*, 3:33, October 2011.

[49] Martin Buttenschoen, Garrett M. Morris, and Charlotte M. Deane. PoseBusters: AI-based docking methods fail to generate physically valid poses or generalise to novel sequences, November 2023. arXiv:2308.05777 [physics, q-bio].

[50] Rocco Meli and Philip C. Biggin. spyrmsd: symmetry-corrected RMSD calculations in Python. *Journal of Cheminformatics*, 12(1):49, August 2020.

[51] Giovanni Bolcato, Esther Heid, and Jonas Boström. On the Value of Using 3D Shape and Electrostatic Similarities in Deep Generative Methods. *Journal of Chemical Information and Modeling*, 62(6):1388–1398, March 2022. Publisher: American Chemical Society.

[52] Adrian M. Schreyer and Tom Blundell. USRCAT: real-time ultrafast shape recognition with pharmacophoric constraints. *Journal of Cheminformatics*, 4(1):27, November 2012.

[53] Xin Jin and Jiawei Han. K-Medoids Clustering. In Claude Sammut and Geoffrey I. Webb, editors, *Encyclopedia of Machine Learning*, pages 564–565. Springer US, Boston, MA, 2010.

[54] Brendan J. Frey and Delbert Dueck. Clustering by Passing Messages Between Data Points. *Science*, 315(5814):972–976, February 2007. Publisher: American Association for the Advancement of Science.

[55] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, November 1987.

[56] Kevin Arvai. kneed, July 2023.

[57] Xiaolin Pan, Hao Wang, Yueqing Zhang, Xingyu Wang, Cuiyu Li, Changge Ji, and John Z. H. Zhang. AA-Score: a New Scoring Function Based on Amino Acid-Specific Interaction for Molecular Docking. *Journal of Chemical Information and Modeling*, 62(10):2499–2509, May 2022.

[58] Chao Yang and Yingkai Zhang. Lin_f9: A Linear Empirical Scoring Function for Protein–Ligand Docking. *Journal of Chemical Information and Modeling*, 61(9):4630–4644, September 2021.

[59] Rodrigo Quiroga and Marcos A. Villarreal. Vinardo: A Scoring Function Based on Autodock Vina Improves Scoring, Docking, and Virtual Screening. *PLOS ONE*, 11(5):e0155183, May 2016. Publisher: Public Library of Science.

[60] Maria Kadukova, Karina dos Santos Machado, Pablo Chacón, and Sergei Grudinin. KORP-PL: a coarse-grained knowledge-based scoring function for protein–ligand interactions. *Bioinformatics*, 37(7):943–950, May 2021.

[61] Maria Kadukova and Sergei Grudinin. Convex-PL: a novel knowledge-based potential for protein-ligand interactions deduced from structural databases using convex optimization. *Journal of Computer-Aided Molecular Design*, 31(10):943–958, October 2017.

[62] Maria Kadukova, Vladimir Chupin, and Sergei Grudinin. Convex-PLR – Revisiting affinity predictions and virtual screening using physics-informed machine learning, September 2021. Pages: 2021.09.13.460049 Section: New Results.

[63] Maciej Wójcikowski, Pedro J. Ballester, and Pawel Siedlecki. Performance of machine-learning scoring functions in structure-based virtual screening. *Scientific Reports*, 7(1):46710, April 2017. Number: 1 Publisher: Nature Publishing Group.

[64] Miles McGibbon, Sam Money-Kyrle, Vincent Blay, and Douglas R. Houston. SCORCH: Improving structure-based virtual screening with machine learning classifiers, data augmentation, and uncertainty estimation. *Journal of Advanced Research*, 46:135–147, April 2023. Publisher: Elsevier.

[65] Maciej Wójcikowski, Michał Kukiełka, Marta M Stepniewska-Dziubinska, and Pawel Siedlecki. Development of a protein–ligand extended connectivity (PLEC) fingerprint and its application for binding affinity predictions. *Bioinformatics*, 35(8):1334–1341, April 2019.

[66] Jacob D. Durrant and J. Andrew McCammon. NNScore: a neural-network-based scoring function for the characterization of protein-ligand complexes. *Journal of Chemical Information and Modeling*, 50(10):1865–1871, October 2010.

[67] Shu Liu, Rao Fu, Li-Hua Zhou, and Sheng-Ping Chen. Application of Consensus Scoring and Principal Component Analysis for Virtual Screening against -Secretase (BACE-1). *PLOS ONE*, 7(6):e38086, June 2012. Publisher: Public Library of Science.

[68] Karen Palacio-Rodríguez, Isaias Lans, Claudio N. Cavasotto, and Pilar Cossio. Exponential consensus ranking improves the outcome in docking and receptor ensemble docking. *Scientific Reports*, 9(1):5142, March 2019. Number: 1 Publisher: Nature Publishing Group.

[69] Jean-François Truchon and Christopher I. Bayly. Evaluating Virtual Screening Methods: Good and Bad Metrics for the "Early Recognition" Problem. *Journal of Chemical Information and Modeling*, 47(2):488–508, March 2007. Publisher: American Chemical Society.

[70] Wei Zhao, Kirk E. Hevener, Stephen W. White, Richard E. Lee, and James M. Boyett. A statistical framework to evaluate virtual screening. *BMC Bioinformatics*, 10(1):225, July 2009.

[71] Fergus Imrie, Anthony R Bradley, and Charlotte M Deane. Generating property-matched decoy molecules using deep learning. *Bioinformatics*, 37(15):2134–2141, August 2021.

[72] Streamlit • A faster way to build and share data apps, January 2021.

[73] Dejun Jiang, Chang-Yu Hsieh, Zhenxing Wu, Yu Kang, Jike Wang, Ercheng Wang, Ben Liao, Chao Shen, Lei Xu, Jian Wu, Dongsheng Cao, and Tingjun Hou. InteractionGraphNet: A Novel and Efficient Deep Graph Representation Learning Framework for Accurate Protein–Ligand Interaction Predictions. *Journal of Medicinal Chemistry*, 64(24):18209–18232, December 2021. Publisher: American Chemical Society.

[74] Xujun Zhang, Odin Zhang, Chao Shen, Wanglin Qu, Shicheng Chen, Hanqun Cao, Yu Kang, Zhe Wang, Ercheng Wang, Jintu Zhang, Yafeng Deng, Furui Liu, Tianyue Wang, Hongyan Du, Langcheng Wang, Peichen Pan, Guangyong Chen, Chang-Yu Hsieh, and Tingjun Hou. Efficient and accurate large library ligand docking with KarmaDock. *Nature Computational Science*, 3(9):789–804, September 2023. Number: 9 Publisher: Nature Publishing Group.

[75] Wei Lu, Qifeng Wu, Jixian Zhang, Jiahua Rao, Chengtao Li, and Shuangjia Zheng. TANKBind: Trigonometry-Aware Neural NetworKs for Drug-Protein Binding Structure Prediction. *Advances in Neural Information Processing Systems*, 35:7236–7249, December 2022.

[76] Russell Spitzer and Ajay N. Jain. Surflex-Dock: Docking benchmarks and real-world application. *Journal of Computer-Aided Molecular Design*, 26(6):687–699, June 2012.

[77] Na Zhang and Hongtao Zhao. Enriching screening libraries with bioactive fragment space. *Bioorganic & Medicinal Chemistry Letters*, 26(15):3594–3597, August 2016.

[78] Nobuaki Yasuo and Masakazu Sekijima. Improved Method of Structure-Based Virtual Screening via Interaction-Energy-Based Learning. *Journal of Chemical Information and Modeling*, 59(3):1050–1061, March 2019. Publisher: American Chemical Society.

[79] Viet-Khoa Tran-Nguyen, Guillaume Bret, and Didier Rognan. True Accuracy of Fast Scoring Functions to Predict High-Throughput Screening Data from Docking Poses: The Simpler the Better. *Journal of Chemical Information and Modeling*, 61(6):2788–2797, June 2021. Publisher: American Chemical Society.

[80] F. Gentile, T. I. Oprea, A. Tropsha, and A. Cherkasov. Surely you are joking, Mr Docking! *Chemical Society Reviews*, 52(3):872–878, 2023.

[81] Gregory A. Landrum, Jessica Braun, Paul Katzberger, Marc T. Lehner, and Sereina Riniker. lwreg: A Lightweight System for Chemical Registration and Data Storage, July 2024.

[82] Patrick Bryant, Atharva Kelkar, Andrea Guljas, Cecilia Clementi, and Frank Noé. Structure prediction of protein-ligand complexes from sequence information with Umol. *Nature Communications*, 15(1):4536, May 2024. Publisher: Nature Publishing Group.

[83] Toni Sivula, Laxman Yetukuri, Tuomo Kalliokoski, Heikki Käsnänen, Antti Poso, and Ina Pöhner. Machine Learning-Boosted Docking Enables the Efficient Structure-Based Virtual Screening of Giga-Scale Enumerated Chemical Libraries. *Journal of Chemical Information and Modeling*, 63(18):5773–5783, September 2023. Publisher: American Chemical Society.

[84] Francesco Gentile, Vibudh Agrawal, Michael Hsing, Anh-Tien Ton, Fuqiang Ban, Ulf Norinder, Martin E. Gleave, and Artem Cherkasov. Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Central Science*, 6(6):939–949, June 2020. Publisher: American Chemical Society.