International Journal of
Molecular Sciences

MDPI

Article

vScreenML v2.0: Improved Machine Learning Classification for Reducing False Positives in Structure-Based Virtual Screening

Grigorii V. Andrianov [1,2], Emeline Haroldsen [1] and John Karanicolas [1,3,*]

[1] Cancer Signaling & Microenvironment Program, Fox Chase Cancer Center, Philadelphia, PA 19111, USA; grigorii.andrianov@gmail.com (G.V.A.); eharold1@jh.edu (E.H.)
[2] Institute of Fundamental Medicine and Biology, Kazan Federal University, Kazan 420008, Russia
[3] Moulder Center for Drug Discovery Research, Temple University School of Pharmacy, Philadelphia, PA 19140, USA
* Correspondence: john.karanicolas@abbvie.edu; Tel.: +1-215-728-7067

Abstract: The enthusiastic adoption of make-on-demand chemical libraries for virtual screening has highlighted the need for methods that deliver improved hit-finding discovery rates. Traditional virtual screening methods are often inaccurate, with most compounds nominated in a virtual screen not engaging the intended target protein to any detectable extent. Emerging machine learning approaches have made significant progress in this regard, including our previously described tool vScreenML. The broad adoption of vScreenML was hindered by its challenging usability and dependencies on certain obsolete or proprietary software packages. Here, we introduce vScreenML 2.0 to address each of these limitations with a streamlined Python implementation. Through careful benchmarks, we show that vScreenML 2.0 outperforms other widely used tools for virtual screening hit discovery.

Keywords: drug discovery; virtual screening; machine learning

Citation: Andrianov, G.V.; Haroldsen, E.; Karanicolas, J. vScreenML v2.0: Improved Machine Learning Classification for Reducing False Positives in Structure-Based Virtual Screening. *Int. J. Mol. Sci.* **2024**, *25*, 12350. https://doi.org/10.3390/ijms252212350

Academic Editor: Dong-Jun Yu

Received: 1 October 2024
Revised: 8 November 2024
Accepted: 11 November 2024
Published: 18 November 2024

1. Introduction

Starting points for drug discovery often arise from screening small-molecule libraries. Hits from such collections are typically recognized either by explicitly evaluating each compound in a relevant functional assay (e.g., biochemical screening), by pulling out binders from a pool of barcoded compounds (e.g., DNA-encoded libraries), or by computationally selecting compounds that fit with the desired site on the target protein surface. Although traditional biochemical screening affords the most directly relevant readout, its application is typically limited to collections of about 2 million compounds. By contrast, pooled DNA-encoded libraries can comprise 1 billion compounds; however, many apparent hits do not show activity when re-synthesized without the fused DNA (for reasons that include the misidentification of the screening hit, binding that relies on the linked DNA, and "silent" binders that do not confer any effect on protein function).

In recent years, chemical vendors significantly increased the size of their catalogs of compounds by enumerating specific building blocks that can be combined using robust chemical transformations. This approach can lead to enormous libraries of new compounds that are readily synthetically tractable; for example, Enamine compiled offerings of ~29 billion "make-on-demand" compounds for purchase [1]. Even the most ambitious HTS campaigns cannot explicitly access these enormous "virtual" compound collections; however, this has amplified the importance of using computational screening methods.

Several studies have employed molecular docking to screen multi-million compound libraries, in some cases yielding advanceable hits. The fraction of computational hits that show activity is often dependent on the target class, with GPCR ligands often yielding high hit rates (and initial hits that can also be quite potent). Among GPCR targets, for instance, the screening of a virtual library of 75 million tetrahydropyridines against the serotonin 5-HT2A receptor prioritized 17 compounds for experimental validation: 4 of these (24%)

proved to be active, with binding affinities in the low-micromolar range (0.67–3.9 μM) [2]. A study targeting the MT_1 and MT_2 receptors screened 150 million compounds and found that 15 of 38 compounds tested (39%) were active in the nano- to micromolar range (from 1 nM to 15 μM) [3]. In another study, screening 138 million compounds against the D4 dopamine receptor found that 122 hits from among 549 tested (22%) with more than a 50% inhibition at 10 μM [4]. A screen of 490 million compounds against the σ2 receptor found that 124 of 484 compounds tested (26%) were active in the nano- to micromolar range [5]. Screening over 300 million diverse molecules targeting the α2A-adrenergic receptor led to the selection of 48 compounds, of which 30 compounds (63%) showed activities ranging from 1.7 nM to 9.4 μM [6]. Screening 115 million compounds against the CysLT receptor subtypes yielded 10 of 71 (14%) for CysLT1R and 25 of 68 (37%) for CysLT2R [7]. A focused screen of 140 million triazoles and isoxazoles against CB_2 provided 11 hits, of which 6 compounds (55%) showed activity below 10 μM [8].

For non-GPCR targets, though, hit rates are typically lower. For example, the screening of 99 million compounds against AmpC β-lactamase yielded five active compounds (1.3–400 μM) from 44 tested (11%) [4]. A large-scale screen of 235 million compounds against the SARS-CoV-2 main protease (Mpro) provided 3 hits from 100 compounds tested in an enzyme activity assay (3%), with affinities between 23 and 61 μM [9]. A screen of 400 million lead-like molecules against Mac1 produced 13 hits from 124 selected compounds (10%), with IC_{50} values ranging from 42 to 504 μM [10]. Finally, screening 1.3 billion compounds against KEAP1 identified 69 hits with submicromolar binding affinity from 590 tested compounds (12%) [11].

Hit-finding discovery rates were also notably low in the Critical Assessment of Computational Hit-finding Experiments (CACHE) challenges [12]. In the first of these benchmarking initiatives, for example, 23 participants were offered the opportunity to each choose up to 100 compounds for testing against a target selected by the organizers (the WDR domain of LRRK2). When tested in a primary SPR assay, only 73 compounds (3.2%) showed any hint of activity, and many of these hits did not replicate in secondary assays. Subsequent (ongoing) benchmarks have also shown similar outcomes. It is worth noting that the organizers intentionally select challenging/unprecedented targets for these benchmarks, which likely contribute to the low hit rates.

As highlighted above then, most compounds selected in ultra-large virtual screening campaigns turn out to be false positives. While vendors such as Enamine have dramatically reduced the expense associated with procuring compounds prioritized by the computational screen, the cost of testing these candidates is still significant and scales with the number of compounds to be tested. Whereas false negatives in screening simply represent missed opportunities for alternate hits, false positives represent a very real expense because they consume wet-lab time and reagents. Even in the highly successful screens cited above, the vast majority of virtual screening hits are not active when characterized in biochemical assays (i.e., hit rates that are typically far below 50%).

To address this, we recently developed a machine learning classifier dubbed "vScreenML" [13]. This model was trained to distinguish structures of active complexes from carefully curated decoys that would otherwise represent likely false positives. After validating the model with a series of retrospective benchmarks, we applied vScreenML to select hits from an Enamine library docked against human acetylcholinesterase (AChE). The top 23 compounds were purchased and characterized in a biochemical assay: this experiment revealed that most of the compounds prioritized by vScreenML were indeed AChE inhibitors, with more than half showing IC_{50} lower than 50 μM and the best hit yielding a K_I value of 175 nM. Importantly, none of these hits bore any resemblance to known AChE inhibitors (and AChE had not been used in training), confirming that the model had not simply memorized interactions from training.

Despite the dramatic performance of vScreenML relative to standard approaches, certain hurdles slowed its widespread adoption. Specifically, it required the complicated manual compilation of certain programs needed calculating features that describe each

docked model, including multiple outdated or expensive dependencies that proved to be prohibitive for many users. Here, we therefore report an updated version called vScreenML 2.0 (https://github.com/gandrianov/vScreenML2, accessed on 1 September 2024). This update is far easier to install and use, and also avoids the dependencies that previously proved cumbersome. At the same time, we also updated the model by including newly released structures from PDB and incorporating several additional features for enhanced discriminative power.

2. Results

The overall approach for training vScreenML 2.0 was conceptually similar to that of its predecessor (Figure 1), with key differences described in the Materials and Methods section. As noted above, vScreenML 2.0 reduces inconvenient software dependencies and also includes new features not present in the original implementation. These include ligand potential energy, buried unsatisfied atoms for select polar groups in ligand, additional 2D structural features of ligands, the complete characterization of interface interactions in protein–ligand complexes and pocket-shape features (Table S2). To ensure model generalization (and avoid potential overtraining), we identified the 49 most important features to include in the model, rather than allow all 165 features (Figure 1C).

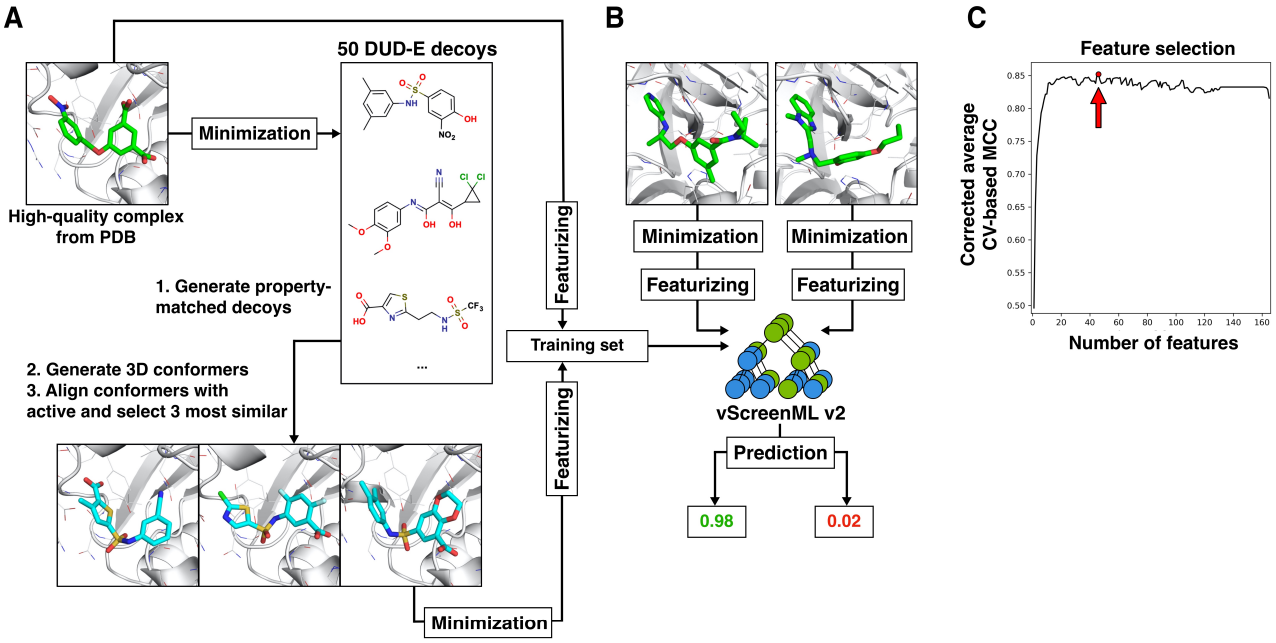

Figure 1. Dataset and training for vScreenML 2.0. (**A**) The dataset for training and testing comprises a set of active complexes (green sticks), along with compelling decoy complexes built to match the active complexes. Actives were drawn from the Protein Data Bank (PDB), refined using Protoss, and then minimized using PyRosetta. Three decoy complexes (blue sticks) were built from each active complex, using ROCS to identify conformations of compounds from DUD-E that can adopt similar 3D structures as the active compound. Decoy complexes were refined the same way as the active complexes. (**B**) The structure of each active and decoy complex (green sticks) was used to calculate 166 numerical features: these were used to train the vScreenML 2.0 classifier. Actives were assigned label 1 and decoys were assigned label 0, so vScreenML 2.0's output scores range between 0 and 1. (**C**) To enhance robustness and avoid potential overtraining, feature reduction was applied. The final vScreenML 2.0 model includes only 49 (red dot and arrow) of the original 166 features, since this model maximizes the Matthews correlation coefficient (MCC).

As a first evaluation of vScreenML 2.0, we compared its performance to that of the original version of vScreenML [13]. A direct comparison is complicated by the fact that the models were trained on different datasets. Each model should only be characterized

using a held-out test set, and there is not a defined set common to either model that was not used in training. Nonetheless, when both models are applied to data from their respective datasets that were not used in training, the performance of vScreenML 2.0 far exceeds that of the original version (Figure 2).

In each case, each of the models was applied to score protein–ligand complexes not used in training, involving exclusively protein targets dissimilar to those in the training set. Both versions of vScreenML recognized the active complexes with high scores (close to 1), and the decoy complexes with low scores (close to 0) (Figure 2A). The primary distinction that is evident between the two distributions is that vScreenML 2.0 miscategorized fewer of the active complexes with low scores, compared to the original: this is reflected numerically in the higher recall value for vScreenML 2.0 (from 0.67 in the original to 0.89 in the new model). The improved recall, coupled with improved precision, leads to a dramatic improvement in the Matthews correlation coefficient (MCC) for the new model (from 0.69 in the original to 0.89 in vScreenML 2.0). The improved performance is also dramatically evident when the same results are plotted as a receiver operating characteristic (ROC) curve (Figure 2B), demonstrating the enhanced performance of vScreenML 2.0 when applied to the classification of held-out protein–ligand complexes.

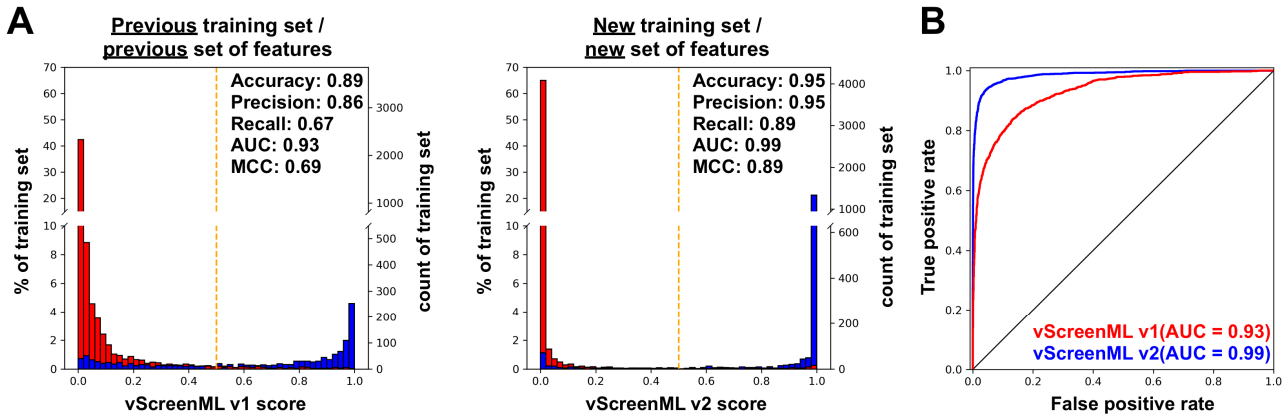

Figure 2. Performance of vScreenML on a test dataset. (**A**) Discriminating the performance of the original vScreenML model (**left**) versus the new vScreenML 2.0 (**right**). The vScreenML score for each data point was calculated using cross-validation, where each fold was prepared by clustering protein sequences using MMSeq2. Active complexes (true class label is 1) are represented by blue data points and decoy complexes (true class label is 0) are represented by red data points. The orange dashed line indicates an equal likelihood of being classified as active or decoy. (**B**) Receiver operator characteristic plots showing performance of vScreenML 2.0 relative to the original model.

Next, we compared the performance of vScreenML 2.0 in comparison with other widely used tools for virtual screening hit discovery. As a representative empirical scoring function, AA-Score was selected, a model that combines a broad variety of energetic components into a single score and showed impressive performance relative to a broad slate of other common scoring functions [14]. As a classic scoring function, we used the GNINA (v1.1) [15] package's implementation of AutoDock Vina [16], historically one of the most widely used tools for virtual screening. Finally, as a representative of the deep learning models that have recently been described in the literature, we selected the GNINA's CNNaffinity score [17]. This scoring method was shown to outperform AutoDock Vina scoring in a standard benchmark [18], though the authors acknowledge that some lingering bias (favoring GNINA's CNNaffinity score) may come from recognizing property distributions rather than the details of molecular interactions.

We evaluated each of these methods using the DEKOIS2 dataset [19,20]. This benchmark comprises 81 protein targets, with the 2D chemical structures of 30–40 actives and 800–1200 property-matched decoys for each target. Each of these ligands was docked to the corresponding protein target using Schrödinger Glide in the context of a separate

study [21,22], and so we re-used these starting poses. Each starting pose was minimized using PyRosetta to make them energetically consistent with the vScreenML 2.0 features.

While additional deep learning-based docking tools have also been developed for virtual screening hit discovery, such as KarmaDock [22] and RTMScore [23], they have seen most of the DEKOIS2 targets in the course of training: this makes it impossible to determine the extent to which their performance on this test set reflects the true performance that should be expected in a prospective application. This point is reinforced through the behavior of KarmaDock on PYGL ligands (Table S3). Structures of this target in the "in" conformation but not the "out" conformation were included in the KarmaDock training set, and we found that KarmaDock yields far superior performance for DEKOIS2 ligands in the "in" conformation. While GNINA's CNNaffinity score was indeed trained on some of the DEKOIS2 targets, it saw fewer of these in training than KarmaDock or RTMScore (Table S3).

To avoid *any* potential for vScreenML 2.0 recognizing features of the DEKOIS2 targets, we used a model for each target that excluded the corresponding protein cluster from training (i.e., all data from any sequence-related proteins were left out of training).

To evaluate te performance of the four selected scoring methods (vScreenML 2.0, AA-Score, AutoDock Vina, and GNINA CNNaffinity), we used each method to rank each of the pre-built DEKOIS2 models for each target and calculated the enrichment factor of actives in the top 1%. For a typical target with 30 actives and 970 decoys (1000 models), one would expect to find, by random chance, 0.3 actives in the top-scoring 10 models; a scoring tool that finds 3 actives within the top-scoring 10 models would therefore have an enrichment factor of 10.

The distribution of EF1% values for the 81 protein targets was first compared for vScreenML 2.0 and for AA-Score (Figure 3A, *top*). It is evident from the superposed histograms that AA-Score had more targets with very low EF1% values (0 to 7), whereas vScreenML 2.0 had more targets with higher EF1% values (10 and above). Extending the analysis to compare individual targets (Figure 3A, *bottom*), there were a few individual targets for which both methods perform well. For the most part, however, the clearest observation is that there are many more points above the diagonal than below, representing examples in which vScreenML 2.0 provides superior performance to AA-Score. Analysis using the binomial test confirms the statistical significance of the observation that vScreenML 2.0 is better for more targets than AA-Score ($p < 9 \times 10^{-6}$). The performance of AutoDock Vina was similarly surpassed by vScreenML 2.0 ($p < 2 \times 10^{-9}$) (Figure 3B).

Comparing vScreenML 2.0 to GNINA CNNaffinity showed another similar performance in this experiment. The superposition of the EF1% distributions shows slightly more low values for GNINA CNNaffinity, and slightly more high values for vScreenML 2.0 (Figure 3C, *top*). Comparing the 81 individual targets, both methods yielded a similar performance for 35 targets (absolute difference in EF1%, less than 3). Of the other 46 targets, vScreenML 2.0 was superior for 28 targets, and GNINA CNNaffinity was superior for the other 18; per the binomial test, this difference is indeed statistically significant ($p = 0.021$) (Figure 3C, *bottom*). We also cannot fully rule out the possibility that GNINA CNNaffinity's performance was bolstered by "recognizing" some interactions from its own training, which includes proteins from many of the target classes included in the DEKOIS2 benchmark. Nonetheless, there appears to be little agreement between which targets are "easiest" for these two methods: this may imply that the methods are recognizing different features in the active complexes, and that a consensus scoring approach may prove superior to either method alone.

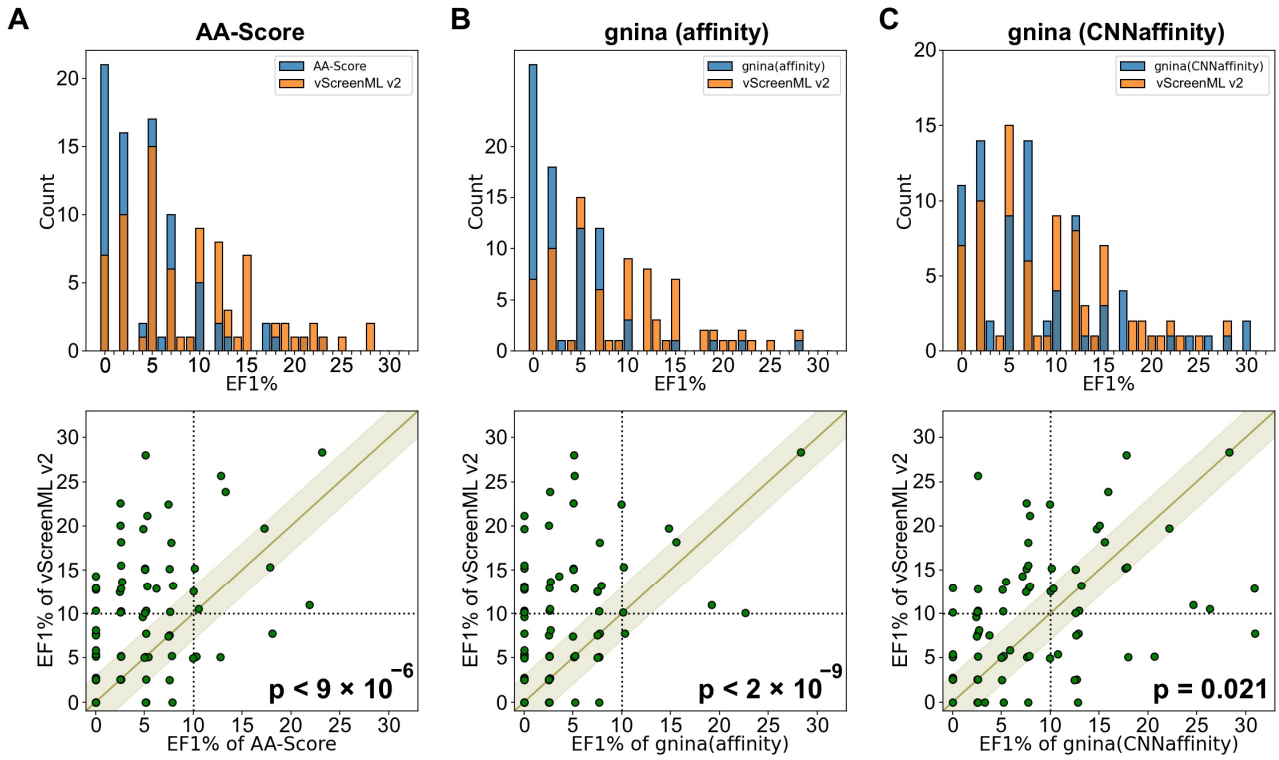

Figure 3. Comparison of vScreenML 2.0 to other modern scoring methods, using the pre-docked models from the DEKOIS2 benchmark. The test set consists of 81 target proteins, each of which is associated with 30–40 active compounds and 800–1200 decoy compounds. The performance of vScreenML 2.0 is compared with that of (**A**) AA-Score, (**B**) AutoDock Vina, and (**C**) GNINA CNNaffinity. In each case, performance is characterized via the enrichment factor of actives in the top 1% (EF1%). ***Top***: distribution of EF1% values across the set of 81 targets (orange is for vScreenML 2.0 and blue is for AA-Score, AutoDock Vina and GNINA). ***Bottom***: comparison of performance (green points) for specific targets. In all three cases, there are more points above the diagonal band than below the diagonal band, indicating that vScreenML 2.0 has superior performance on more targets. *p*-values were calculated using the binomial test (excluding points within the light-green band, for which the absolute difference in EF1% was less than 3). Black dash lines represents 10% of EF1%.

3. Discussion

Structure-based drug design (SBDD) transformed drug discovery utilizing the structural data of target proteins to guide the development of therapeutic agents. By focusing on the atomic-level details of protein binding sites, SBDD enables the design of compounds with considerations of optimal fit in the binding site; this can enable the optimization of binding affinity, efficacy, and selectivity. Unlike traditional HTS methods, which test large numbers of compounds without structural insight, virtual screening takes a more rational approach, increasing the likelihood of identifying promising starting points early in the development process. Notable examples of SBDD's successes include carbonic anhydrase inhibitor dorzolamide [24], kinase inhibitors vemurafenib [25], and ponatinib [26], and protease inhibitors dabigatran [27] and boceprevir [28].

vScreenML 2.0 addresses a key challenge in virtual screening, the high rate of false positives generated by traditional computational methods of SBDD. The initial version of vScreenML proved extremely successful in prospective applications but had strong drawbacks with respect to usability. These included a complex installation process, a complicated pipeline requiring manual intervention, and the inclusion of multiple outdated, complex, or paid dependencies.

Through this update, we addressed each of these issues by rewriting the entire framework in a more accessible programming language (Python); this both promotes ease of

use and installation, while also enabling a straightforward integration of vScreenML 2.0 into existing virtual screening pipelines. These changes thus open the door for a broader user base, including non-experts who may have been previously deterred by the technical complexities of the earlier version. By improving the calculation pipeline and removing dependencies on obsolete or proprietary software, we also made the tool more user-friendly and compatible with modern computational environments.

Finally, we updated the training set for the machine learning model by incorporating the latest crystal structures from the PDB, applying stringent new selection criteria to improve data quality, and introducing several new features that capture more nuanced aspects of protein-ligand interactions. Though we could not directly compare vScreenML 2.0 against the original version using the same test set, the performance of the newer version appears superior. Nonetheless, the new model still remains to be validated in future prospective experiments to evaluate how these enhancements translate to meaningful gains in real-world applications.

Relative to standard empirical scoring functions such as AA-Score and AutoDock Vina, vScreenML 2.0 shows clear superiority for virtual screening hit discovery. While deep learning methods such as KarmaDock, RTMScore, and GNINA use CNN-based scoring functions that seem to provide similar or superior accuracy to vScreenML 2.0 in some respects, their ability to generalize to new targets remains unclear. Rigorous benchmarking can be difficult because these methods often incorporate all available public data in training, such that there is no single held-out test set that can be used to compare different methods. A key benefit of vScreenML is that the simplicity of the model and rigorous splitting between the training and test sets ensures generalizability to new protein target classes. As noted in the context of the DEKOIS2 benchmark, the vScreenML 2.0 performance presented above included no related proteins for each test set target, confirming the generalization of the underlying models.

Looking ahead, we note that the size of ultra-large chemical libraries has grown much faster than the ability to explicitly dock all compounds in a modern library. To address this issue, new strategies have been employed to reduce the computational burden. One such approach entails using active learning together with docking results [29,30]. After explicitly docking a subset of the library against the target, ML models can be trained to rapidly identify compounds likely to yield good docking scores, without explicitly docking them. In using this model, a small subset of compounds can be prioritized for explicit docking from the vast chemical space available in the library. These top-ranked predictions are validated through explicit docking, and the model can be retrained using these additional data and the process repeated. Within these active learning frameworks, vScreenML 2.0 may provide added value by steering the search for compounds worth explicitly docking, in addition to guiding the final selection of compounds for experimental characterization.

4. Materials and Methods

As with the original vScreenML approach [13], our new model is intended to distinguish active compounds from inactive compounds. In a virtual screening workflow, vScreenML 2.0 is thus intended to be used after the contents of a library have been docked to the target protein, for selecting candidate compounds that should be advanced for explicit testing in biochemical or cellular assays—without bringing forward an abundance of false positives (Figure 4).

As described below, this updated version of vScreenML is more user-friendly for broad audiences, avoids using software with potentially costly licenses, and also adds flexibility that facilitates incorporation into complex virtual screening pipelines. To achieve these objectives, we replaced the C++-based Rosetta bundle with the Python-based PyRosetta package (https://www.pyrosetta.org, accessed on 1 September 2024) [31], we replaced MGLTools (v.1.5.7, La Jolla, CA, USA) [32] with ODDT (https://github.com/oddt/oddt, accessed on 1 September 2024) [33], we replaced ChemAxon cxcalc (v.20.17.0, Budapest, Hungary) with the open source RDKit (https://www.rdkit.org) equivalent [34], and we

removed the dependency on OpenEye's SZYBKI (v1.9.0.3, Sante Fe, NM, USA) [35]. When training the new model, we also took this opportunity to incorporate newly released structures in the PDB and apply updated rules for defining which structures to include. Finally, we also incorporated new features to better capture several structural measures that were not well described in the original model: buried unsatisfied polar groups, 2D structure properties of the ligand, and shape-based information from the protein pocket.

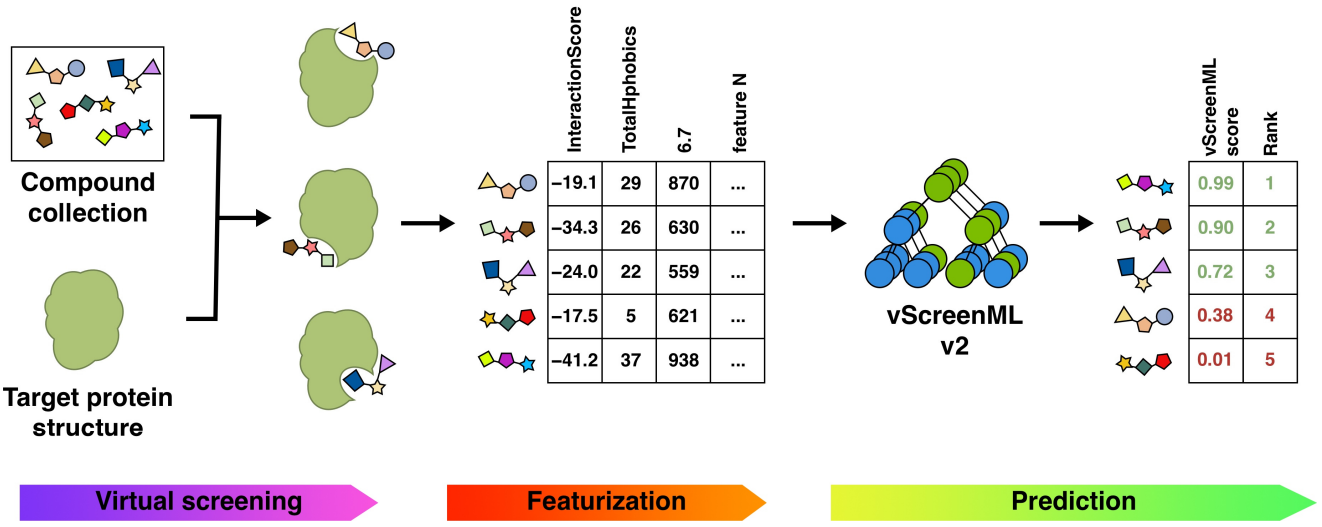

Figure 4. Incorporating vScreenML 2.0 into a virtual screening pipeline. Candidate protein–ligand complexes can be generated by receptor-based methods (i.e., docking or pharmacophoric alignment to one or more known ligands). Scoring each of the resulting complexes with vScreenML 2.0 involves extracting "features" from each model using the PyRosetta (v. 2023.14+release.7132bdc), Binana (v.2.1), RF-Score (v1), RDKit (v.2023.9.5), LUNA (v.0.13.0), and PocketDruggability (v.0.98.1) packages. These features are used as input to vScreenML 2.0, which uses an XGBoost-based model to produce a single score reflecting the prioritization of the compounds in the screening collection.

4.1. Preparation of Dataset (Actives and Decoys)

The original vScreenML approach [13] started from a set of (active) protein-ligand complexes from the RCSB Protein Data Bank (PDB) [36,37]. Each active compound was used to generate three property-matched "decoys" by selecting compounds with high 3D similarity to the corresponding active, from a set of physicochemically matched candidates provided by the "Directory of Useful Decoys, Enhanced" (DUD-E) server [38]. The activity of these decoys has not explicitly been tested for the corresponding protein target, but the decoys are presumed to be inactive. Actives and decoys were refined using the same protocol, to avoid any clues from how the structures were prepared that could lead to artifactual performance by the classification model. The overall approach for training vScreenML 2.0 was conceptually similar, as summarized below (Figure 1).

For vScreenML 2.0, we began with the updated contents of the PDB. We compiled a list of structures from the RCSB Protein Data Bank (PDB) that met the criteria: (A) experimental method "X-Ray diffraction", (B) refinement resolution "<2.5 Å", (C) and polymer entity type "Protein". There were 132,146 PDB entries meeting these criteria. These structures were then filtered by physicochemical and structural features of ligand and protein–ligand complex presented in Table S1; specifically, these filters were primarily intended to restrict the training set actives to "drug-like" ligands (the intended domain of application for vScreenML 2.0) by removing the numerous non-drug-like biological ligands in the PDB (e.g., ATP, sugars, lipids, etc.). Structures were also removed if the model quality was low, if it was incompletely resolved, or if it did not fit with the electron density.

The filtered dataset comprised 1407 PDB entries. For this collection, protonation states were assigned (along with placement of hydrogen atoms) using Protoss; however, 86 structures could not be refined, and were therefore excluded, leaving a total

of 1321 unique PDB entries. Among these, some had multiple copies of the complex in the asymmetric unit. These copies have slight differences in the coordinates relative to one another, and thus differences in the calculated features as well. As a form of data augmentation, we retained these multiple copies of the same complex: we expected that this would help make vScreenML 2.0 resistant to noise in the training set. Thus, our set of actives comprised 1806 instances, of which 1321 are unique protein–ligand complexes. Each structure was minimized using PyRosetta prior to calculating vScreenML 2.0 features.

From each of the 1806 actives, 3 property-matched decoys were also generated. To do so, the DUD-E server was first applied to each ligand in the collection of actives, leading to 50 candidate decoy ligands. These were then filtered based on 2D structural considerations used for the active ligands (Table S1) to ensure that there would be no systematic differences between active versus decoy ligands that could lead to spurious model performance. For each of the candidate decoys that passed the filters, 500 low-energy conformers were generated using OpenEye OMEGA (v. 5.0.0.3, Santa Fe, NM, USA) [39], and these were aligned to the corresponding active structure using ROCS (v. 3.6.1.3, Santa Fe, NM, USA) [40]. The three best-matched decoys (evaluated using the TanimotoCombo score) were selected, because these would be most likely to fit with the protein pocket in a compelling manner. For certain actives, there were not three candidate decoys from DUD-E that passed these filters; in those cases, fewer than three decoys were generated. The alignment from ROCS was used to place the decoy ligands into the protein structure from the active ligand, and then the model was refined using Protoss [41,42] and minimized with PyRosetta. Ultimately, this procedure led to a total of 4475 decoys, reflecting the fact that some of the 1806 actives produced fewer than three decoys.

4.1.1. Model Training and Feature Selection

As with the original vScreenML approach [13], numeric descriptors (features) were calculated from the structure of each protein–ligand complex. These were used in training the XGBoost (v.1.7.6) [43] machine learning (ML) model and when applying the model in practice. As noted earlier, some elements of the original vScreenML features were not particularly user-friendly: these dependencies were eliminated in vScreenML 2.0, and a completely new Python-based framework was developed for vScreenML 2.0. Additionally, several new features are also included in this new implementation. A total of 166 features are calculated for each protein–ligand complex: these are reported in Table S2.

To prevent potential information leakage during training and testing, the dataset was divided into distinct groups by clustering protein sequences with MMSeq2 [44]. By ensuring that related proteins are always placed together in either the training set or the test set, we avoid potential overtraining in which the model sees a very similar complex in both training and testing. It is important to avoid this scenario, because it can lead to artificially inflated performance of the model and disappointing results in future prospective applications [45].

Clustering on protein sequences yielded 269 clusters that were used for cross-validation by splitting the data into multiple sets for training and testing across several iterations. In each iteration, proteins from all clusters except one were used to train the model, while the remaining cluster was used for testing. This process was repeated until each cluster had been used as a test set. The final performance score of the model was obtained by averaging the performance metrics from all iterations, providing a more robust and reliable assessment of the model's ability to generalize to unseen data.

To promote model generalization and avoid potential overfitting associated with using 166 features, we sought to eliminate features not contributing to model performance. We applied sequential backward floating feature selection, as implemented in the mlxtend [46] package in combination with cross-validation on the 269 protein sequence clusters. We found that the model which maximized the Matthews correlation coefficient required 49 features (Figure 1C): the features included in the final model are indicated in Table S2.

As with the original vScreenML implementation, we used the binary Extreme Gradient Boosting (XGB) framework [43] for our model. In model training, hyper-parameters of the

XGB model ("n_estimators", "max_depth", and "learning_rate") were optimized using Optuna [47] to find the set of parameters that gave the best AUC upon 10-fold cross-validation.

4.1.2. Benchmarking with DEKOIS2

The performance of vScreenML 2.0 was evaluated using DEKOIS2 [19,20], a collection of active and decoy compounds for 81 protein targets. Models for the actives and the decoys in complex with their cognate target proteins were obtained from a docked set created to evaluate the docking tool KarmaDock [21,22]. The structures were minimized using PyRosetta to make them more energetically consistent with the vScreenML 2.0 features.

To prevent any potential information leakage from the training set, we clustered the protein targets from DEKOIS2 along with our training set, and excluded from the training set any targets with high sequence similarity to the DEKOIS2 protein targets.

The performance of vScreenML 2.0 was compared with other virtual screening tools that allow scoring in place of pre-generated poses, specifically AA-Score [14] and GNINA [15]. As a performance metric, we used the enrichment factor of actives in the top 1% (EF1%) of predicted scores. A statistical comparison of methods was carried out by removing "near ties" (absolute difference in EF1% less than 3) and then using the binomial test implemented in the SciPy package [48].

4.2. Data Availability

Minimized 3D structures of proteins in complex with actives and decoys, along with calculated features, are available from Zenodo (https://doi.org/10.5281/zenodo.10819385). The vScreenML 2.0 package and source code are located on GitHub (https://github.com/gandrianov/vScreenML2 , accessed on 1 September 2024).

5. Conclusions

vScreenML 2.0 represents a significant advancement in machine learning-based virtual screening, specifically aimed at reducing the number of false positive predictions. By overcoming the usability challenges of the previous version, vScreenML 2.0 provides a more streamlined and accessible implementation, capable of functioning both as a standalone application and as part of a complex virtual screening pipeline. Training on a newly compiled dataset with carefully selected features improved the tool's accuracy and generalizability across various targets. In benchmark experiments, we found that vScreenML 2.0 outperformed tools such as AA-Score and AutoDock Vina; we expect these improvements to translate into enhanced effectiveness in real-world applications as well.

Supplementary Materials: The following supporting information can be downloaded at https://www.mdpi.com/article/10.3390/ijms252212350/s1.

Author Contributions: Conceptualization, G.V.A. and J.K.; Methodology, G.V.A. and J.K.; Software, G.V.A. and E.H.; Formal analysis, G.V.A.; Investigation, G.V.A. and E.H.; Data curation, G.V.A. and E.H.; Writing—original draft, G.V.A. and J.K.; Writing—review & editing, J.K.; Supervision, J.K.; Funding acquisition, J.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the W.M. Keck Foundation and by the NIH National Institute of General Medical Sciences (R01GM141513). This research was also funded in part through the NIH/NCI Cancer Center Support Grant P30CA006927. This work used the Extreme Science and Engineering Discovery Environment (XSEDE) allocation MCB130049, which is supported by the National Science Foundation, grant number 1548562. This work also used computational resources through allocation MCB130049 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants 2138259, 2138286, 2138307, 2137603, and 2138296.

Data Availability Statement: The original contributions presented in the study are included in the article/Supplementary Material, further inquiries can be directed to the corresponding author.

Acknowledgments: We thank ChemAxon for providing an academic research license.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Grygorenko, O.O.; Radchenko, D.S.; Dziuba, I.; Chuprina, A.; Gubina, K.E.; Moroz, Y.S. Generating Multibillion Chemical Space of Readily Accessible Screening Compounds. *iScience* **2020**, *23*, 101681. [CrossRef]
2. Kaplan, A.L.; Confair, D.N.; Kim, K.; Barros-Álvarez, X.; Rodriguiz, R.M.; Yang, Y.; Kweon, O.S.; Che, T.; McCorvy, J.D.; Kamber, D.N.; et al. Bespoke library docking for 5-HT2A receptor agonists with antidepressant activity. *Nature* **2022**, *610*, 582–591. [CrossRef]
3. Stein, R.M.; Kang, H.J.; McCorvy, J.D.; Glatfelter, G.C.; Jones, A.J.; Che, T.; Slocum, S.; Huang, X.-P.; Savych, O.; Moroz, Y.S.; et al. Virtual discovery of melatonin receptor ligands to modulate circadian rhythms. *Nature* **2020**, *579*, 609–614. [CrossRef]
4. Lyu, J.; Wang, S.; Balius, T.E.; Singh, I.; Levit, A.; Moroz, Y.S.; O'Meara, M.J.; Che, T.; Algaa, E.; Tolmachova, K.; et al. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224–229. [CrossRef]
5. Alon, A.; Lyu, J.; Braz, J.M.; Tummino, T.A.; Craik, V.; O'Meara, M.J.; Webb, C.M.; Radchenko, D.S.; Moroz, Y.S.; Huang, X.-P.; et al. Structures of the σ2 receptor enable docking for bioactive ligand discovery. *Nature* **2021**, *600*, 759–764. [CrossRef]
6. Fink, E.A.; Xu, J.; Hübner, H.; Braz, J.M.; Seemann, P.; Avet, C.; Craik, V.; Weikert, D.; Schmidt, M.F.; Webb, C.M.; et al. Structure-based discovery of nonopioid analgesics acting through the α 2A-adrenergic receptor. *Science* **2022**, *377*, eabn7065. [CrossRef]
7. Sadybekov, A.A.; Brouillette, R.L.; Marin, E.; Sadybekov, A.V.; Luginina, A.; Gusach, A.; Mishin, A.; Besserer-Offroy, É.; Longpré, J.-M.; Borshchevskiy, V.; et al. Structure-Based Virtual Screening of Ultra-Large Library Yields Potent Antagonists for a Lipid GPCR. *Biomolecules* **2020**, *10*, 1634. [CrossRef]
8. Grotsch, K.; Sadybekov, A.V.; Hiller, S.; Zaidi, S.; Eremin, D.; Le, A.; Liu, Y.; Smith, E.C.; Illiopoulis-Tsoutsouvas, C.; Thomas, J.; et al. Virtual Screening of a Chemically Diverse "Superscaffold" Library Enables Ligand Discovery for a Key GPCR Target. *ACS Chem. Biol.* **2024**, *19*, 866–874. [CrossRef]
9. Luttens, A.; Gullberg, H.; Abdurakhmanov, E.; Vo, D.D.; Akaberi, D.; Talibov, V.O.; Nekhotiaeva, N.; Vangeel, L.; De Jonghe, S.; Jochmans, D.; et al. Ultralarge Virtual Screening Identifies SARS-CoV-2 Main Protease Inhibitors with Broad-Spectrum Activity against Coronaviruses. *J. Am. Chem. Soc.* **2022**, *144*, 2905–2920. [CrossRef]
10. Gahbauer, S.; Correy, G.J.; Schuller, M.; Ferla, M.P.; Doruk, Y.U.; Rachman, M.; Wu, T.; Diolaiti, M.; Wang, S.; Neitz, R.J.; et al. Iterative computational design and crystallographic screening identifies potent inhibitors targeting the Nsp3 macrodomain of SARS-CoV-2. *Proc. Natl. Acad. Sci. USA* **2023**, *120*, e2212931120. [CrossRef]
11. Gorgulla, C.; Boeszoermenyi, A.; Wang, Z.-F.; Fischer, P.D.; Coote, P.W.; Padmanabha Das, K.M.; Malets, Y.S.; Radchenko, D.S.; Moroz, Y.S.; Scott, D.A.; et al. An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **2020**, *580*, 663–668. [CrossRef]
12. Ackloo, S.; Al-Awar, R.; Amaro, R.E.; Arrowsmith, C.H.; Azevedo, H.; Batey, R.A.; Bengio, Y.; Betz, U.A.K.; Bologa, C.G.; Chodera, J.D.; et al. CACHE (Critical Assessment of Computational Hit-finding Experiments): A public-private partnership benchmarking initiative to enable the development of computational methods for hit-finding. *Nat. Rev. Chem.* **2022**, *6*, 287–295. [CrossRef]
13. Adeshina, Y.O.; Deeds, E.J.; Karanicolas, J. Machine learning classification can reduce false positives in structure-based virtual screening. *Proc. Natl. Acad. Sci. USA* **2020**, *117*, 18477–18488. [CrossRef]
14. Pan, X.; Wang, H.; Zhang, Y.; Wang, X.; Li, C.; Ji, C.; Zhang, J.Z.H. AA-Score: A New Scoring Function Based on Amino Acid-Specific Interaction for Molecular Docking. *J. Chem. Inf. Model.* **2022**, *62*, 2499–2509. [CrossRef]
15. McNutt, A.T.; Francoeur, P.; Aggarwal, R.; Masuda, T.; Meli, R.; Ragoza, M.; Sunseri, J.; Koes, D.R. GNINA 1.0: Molecular docking with deep learning. *J. Cheminform.* **2021**, *13*, 43. [CrossRef]
16. 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*, 455–461. [CrossRef]
17. Francoeur, P.G.; Masuda, T.; Sunseri, J.; Jia, A.; Iovanisci, R.B.; Snyder, I.; Koes, D.R. Three-Dimensional Convolutional Neural Networks and a Cross-Docked Data Set for Structure-Based Drug Design. *J. Chem. Inf. Model.* **2020**, *60*, 4200–4215. [CrossRef]
18. Sunseri, J.; Koes, D.R. Virtual Screening with Gnina 1.0. *Molecules* **2021**, *26*, 7369. [CrossRef]
19. Bauer, M.R.; Ibrahim, T.M.; Vogel, S.M.; Boeckler, F.M. Evaluation and optimization of virtual screening workflows with DEKOIS 2.0—A public library of challenging docking benchmark sets. *J. Chem. Inf. Model.* **2013**, *53*, 1447–1462. [CrossRef]
20. Boeckler, F.M.; Bauer, M.R.; Ibrahim, T.M.; Vogel, S.M. Use of DEKOIS 2.0 to gain insights for virtual screening. *J. Cheminform.* **2014**, *6*, O24. [CrossRef]
21. Zhang, X. *DEKOIS2.0 for KarmaDock*; Zenodo: Geneve, Switzerland, 2023.
22. Zhang, X.; Zhang, O.; Shen, C.; Qu, W.; Chen, S.; Cao, H.; Kang, Y.; Wang, Z.; Wang, E.; Zhang, J.; et al. Efficient and accurate large library ligand docking with KarmaDock. *Nat. Comput. Sci.* **2023**, *3*, 789–804. [CrossRef] [PubMed]
23. Shen, C.; Zhang, X.; Deng, Y.; Gao, J.; Wang, D.; Xu, L.; Pan, P.; Hou, T.; Kang, Y. Boosting Protein-Ligand Binding Pose Prediction and Virtual Screening Based on Residue-Atom Distance Likelihood Potential and Graph Transformer. *J. Med. Chem.* **2022**, *65*, 10691–10706. [CrossRef] [PubMed]

24. Baldwin, J.J.; Ponticello, G.S.; Anderson, P.S.; Christy, M.E.; Murcko, M.A.; Randall, W.C.; Schwam, H.; Sugrue, M.F.; Gautheron, P. Thienothiopyran-2-sulfonamides: Novel topically active carbonic anhydrase inhibitors for the treatment of glaucoma. *J. Med. Chem.* **2002**, *32*, 2510–2513. [CrossRef] [PubMed]
25. Tsai, J.; Lee, J.T.; Wang, W.; Zhang, J.; Cho, H.; Mamo, S.; Bremer, R.; Gillette, S.; Kong, J.; Haass, N.K.; et al. Discovery of a selective inhibitor of oncogenic B-Raf kinase with potent antimelanoma activity. *Proc. Natl. Acad. Sci. USA* **2008**, *105*, 3041–3046. [CrossRef]
26. Nascimento, M.; Moura, S.; Parra, L.; Vasconcellos, V.; Costa, G.; Leite, D.; Dias, M.; Fernandes, T.V.A.; Hoelz, L.; Pimentel, L.; et al. Ponatinib: A Review of the History of Medicinal Chemistry behind Its Development. *Pharmaceuticals* **2024**, *17*, 1361. [CrossRef]
27. Nar, H. The role of structural information in the discovery of direct thrombin and factor Xa inhibitors. *Trends Pharmacol. Sci.* **2012**, *33*, 279–288. [CrossRef]
28. Venkatraman, S. Discovery of boceprevir, a direct-acting NS3/4A protease inhibitor for treatment of chronic hepatitis C infections. *Trends Pharmacol. Sci.* **2012**, *33*, 289–294. [CrossRef]
29. Gentile, F.; Agrawal, V.; Hsing, M.; Ton, A.T.; Ban, F.; Norinder, U.; Gleave, M.E.; Cherkasov, A. Deep Docking: A Deep Learning Platform for Augmentation of Structure Based Drug Discovery. *ACS Cent. Sci.* **2020**, *6*, 939–949. [CrossRef]
30. Yang, Y.; Yao, K.; Repasky, M.P.; Leswing, K.; Abel, R.; Shoichet, B.K.; Jerome, S.V. Efficient Exploration of Chemical Space with Docking and Deep Learning. *J. Chem. Theory Comput.* **2021**, *17*, 7106–7119. [CrossRef]
31. Chaudhury, S.; Lyskov, S.; Gray, J.J. PyRosetta: A script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **2010**, *26*, 689–691. [CrossRef]
32. Morris, G.M.; Huey, R.; Lindstrom, W.; Sanner, M.F.; Belew, R.K.; Goodsell, D.S.; Olson, A.J. AutoDock4 and AutoDockTools4: Automated docking with selective receptor flexibility. *J. Comput. Chem.* **2009**, *30*, 2785–2791. [CrossRef] [PubMed]
33. Wójcikowski, M.; Zielenkiewicz, P.; Siedlecki, P. Open Drug Discovery Toolkit (ODDT): A new open-source player in the drug discovery field. *J. Cheminform.* **2015**, *7*, 26. [CrossRef] [PubMed]
34. RDKit: Open-Source Cheminformatics. Available online: https://www.rdkit.org (accessed on 1 March 2024).
35. *SZYBKI 2.7.0.3*; OpenEye, Cadence Molecular Sciences: Santa Fe, NM, USA. Available online: https://www.eyesopen.com/szybki (accessed on 1 March 2024).
36. Berman, H.M.; Westbrook, J.; Feng, Z.; Gilliland, G.; Bhat, T.N.; Weissig, H.; Shindyalov, I.N.; Bourne, P.E. The Protein Data Bank. *Nucleic Acids Res.* **2000**, *28*, 235–242. [CrossRef] [PubMed]
37. Burley, S.K.; Bhikadiya, C.; Bi, C.; Bittrich, S.; Chen, L.; Crichlow, G.V.; Christie, C.H.; Dalenberg, K.; Di Costanzo, L.; Duarte, J.M.; et al. RCSB Protein Data Bank: Powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.* **2021**, *49*, D437–D451. [CrossRef]
38. Mysinger, M.M.; Carchia, M.; Irwin, J.J.; Shoichet, B.K. Directory of useful decoys, enhanced (DUD-E): Better ligands and decoys for better benchmarking. *J. Med. Chem.* **2012**, *55*, 6582–6594. [CrossRef]
39. Hawkins, P.C.D.; Skillman, A.G.; Warren, G.L.; Ellingson, B.A.; Stahl, M.T. Conformer generation with OMEGA: Algorithm and validation using high quality structures from the Protein Databank and Cambridge Structural Database. *J. Chem. Inf. Model.* **2010**, *50*, 572–584. [CrossRef]
40. Hawkins, P.C.D.; Skillman, A.G.; Nicholls, A. Comparison of shape-matching and docking as virtual screening tools. *J. Med. Chem.* **2007**, *50*, 74–82. [CrossRef]
41. Bietz, S.; Urbaczek, S.; Schulz, B.; Rarey, M. Protoss: A holistic approach to predict tautomers and protonation states in protein-ligand complexes. *J. Cheminform.* **2014**, *6*, 12. [CrossRef]
42. Lippert, T.; Rarey, M. Fast automated placement of polar hydrogen atoms in protein-ligand complexes. *J. Cheminform.* **2009**, *1*, 13. [CrossRef]
43. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13 August 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794.
44. Steinegger, M.; Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **2017**, *35*, 1026–1028. [CrossRef]
45. Ong, W.J.G.; Kirubakaran, P.; Karanicolas, J. Poor Generalization by Current Deep Learning Models for Predicting Binding Affinities of Kinase Inhibitors. *bioRxiv.* **2023**. [CrossRef]
46. Raschka, S. MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *J. Open Source Softw.* **2018**, *3*, 638. [CrossRef]
47. Akiba, T.; Sano, S.; Yanase, T.; Ohta, T.; Koyama, M. Optuna: A Next-generation Hyperparameter Optimization Framework. *arXiv* **2019**, arXiv:1907.10902.
48. Virtanen, P.; Gommers, R.; Oliphant, T.E.; Haberland, M.; Reddy, T.; Cournapeau, D.; Burovski, E.; Peterson, P.; Weckesser, W.; Bright, J.; et al. SciPy 1.0: Fundamental algorithms for scientific computing in Python. *Nat. Methods* **2020**, *17*, 261–272. [CrossRef]