

# Evaluating scalable supervised learning for synthesize-on-demand chemical libraries

Moayad Alnammi,<sup>†,‡,¶</sup> Shengchao Liu,<sup>†,‡,@</sup> Spencer S. Ericksen,<sup>§</sup> Gene E.

Ananiev,<sup>§</sup> Andrew F. Voter,<sup>||</sup> Song Guo,<sup>§</sup> James L. Keck,<sup>||</sup> F. Michael

Hoffmann,<sup>§,⊥</sup> Scott A. Wildman,<sup>§</sup> and Anthony Gitter<sup>\*,†,‡,#</sup>

<sup>†</sup>*Department of Computer Sciences, University of Wisconsin-Madison, Madison, WI  
53706, United States*

<sup>‡</sup>*Morgridge Institute for Research, Madison, WI 53715, United States*

<sup>¶</sup>*Department of Information and Computer Science, King Fahd University of Petroleum &  
Minerals, Dhahran 31261, Saudi Arabia*

<sup>§</sup>*Small Molecule Screening Facility, University of Wisconsin Carbone Cancer Center,  
Madison, WI 53792, United States*

<sup>||</sup>*Department of Biomolecular Chemistry, University of Wisconsin School of Medicine and  
Public Health, Madison, WI 53706, United States*

<sup>⊥</sup>*McArdle Laboratory for Cancer Research, University of Wisconsin-Madison, Madison,  
WI 53705, United States*

<sup>#</sup>*Department of Biostatistics and Medical Informatics, University of Wisconsin-Madison,  
Madison, WI 53792, United States*

<sup>@</sup>*Current address: Mila - Quebec AI Institute, Montreal, Quebec H2S 3H1, Canada*

E-mail: gitter@biostat.wisc.edu

## Abstract

Traditional small molecule drug discovery is a time consuming and costly endeavor. High-throughput chemical screening can only assess a tiny fraction of drug-like chemical space. The strong predictive power of modern machine learning methods for virtual chemical screening enables training models on known active and inactive compounds and extrapolating to much larger chemical libraries. However, there has been limited experimental validation of these models’ extensibility toward practical applications on large commercially-available or synthesize-on-demand chemical libraries. Through a prospective evaluation with the bacterial protein-protein interaction PriA-SSB, we demonstrate that ligand-based virtual screening can identify many chemically-diverse active compounds in a large commercial library. We use cross validation to compare many supervised learning models and select a random forest classifier as the best model for this target. When predicting the activity of more than 8 million compounds from Aldrich Market Select, the random forest substantially outperforms a naïve baseline based on chemical structure similarity. 48% of the random forest’s 701 selected compounds are active. The random forest model easily scales to score one billion compounds from the synthesize-on-demand Enamine REAL database. We tested 68 chemically-diverse top predictions from Enamine REAL and observed 31 hits (46%), including one with an IC<sub>50</sub> value of 1.3  $\mu$ M.

## 1 Introduction

Access to very large chemical libraries opens new opportunities in drug discovery but presents major scalability challenges for existing chemical screening workflows. Commercial libraries, which are the primary source of compounds in academic screening efforts, can be grouped into two categories: “in-stock” and “on-demand.” As of this writing, in-stock libraries comprise over 12 million molecules previously synthesized and physically stored by vendors. On-demand libraries are databases containing virtual molecules that a vendor considers to be readily accessible given stocks of available building blocks and established synthetic routes.<sup>1</sup>

Such libraries now measure in the billions of compounds: WuXi’s GalaXi contains 2.1 billion and Enamine’s REAL Space spans 17 billion.<sup>2,3</sup> On-demand libraries have superior chemical scaffold diversity and coverage of chemical shape characteristics,<sup>2</sup> unlocking new possibilities to manipulate biological targets. The growth of on-demand libraries has been disruptive, requiring new scalable strategies for enumerating, storing, and searching them. However, the question remains as to how to effectively prioritize molecules from on-demand libraries for acquisition and testing in the hit identification phase of drug discovery projects.

Virtual screening is essential for guiding the selection of compounds from vast on-demand chemical resources. Virtual screening uses computational methods to select compounds to test experimentally against a target of interest.<sup>4-7</sup> An efficient virtual screening algorithm can exhaustively assess compounds in a very large library so that only the most promising compounds are screened experimentally. Two broad classes of virtual screening approaches, structure-based and ligand-based, have had initial application to large libraries. Structure-based virtual screening, which includes docking,<sup>8,9</sup> uses the three-dimensional shape of the target protein structure to evaluate candidate ligands. Despite recent examples that successfully use docking to filter  $10^8$  to  $10^9$  compounds down to hundreds of interesting compounds,<sup>10,11</sup> there are inherent drawbacks to this approach. Structure-based screening is limited to targets having reasonably accurate protein structure models, and these methods are far more computationally expensive than ligand-based algorithms with respect to compound throughput.<sup>11</sup>

Ligand-based virtual screening evaluates each compound solely based on its chemical properties, which can provide higher throughput. These approaches are applicable to a broader range of targets, including proteins with unknown structures or assay endpoints reflecting perturbations to pathways, phenotypes, or cell populations. Ligand shape matching algorithms can run faster than traditional docking and scale to current on-demand chemical libraries.<sup>12</sup> However, implementations like FastROCS score compounds only based on similarity to active reference compounds. Unlike supervised machine learning, FastROCS cannot

eliminate candidate compounds based on their relationships to known inactive compounds.

Other recent ligand-based virtual screening models are often formulated as a supervised learning problem.<sup>13,14</sup> Models, such as random forests, neural networks, and support vector machines,<sup>15-17</sup> are trained on examples of active and inactive compounds. The requirement for adequate initial assay data is a limitation. However, once trained on sufficient data, these models can evaluate a new compound in milliseconds, making them attractive for very large libraries. Initial prospective evaluations of ligand-based models applied to large libraries have been promising but rare.<sup>18-21</sup> In general, the performance of such models in practical applications with on-demand libraries is still poorly understood.

Here, we demonstrate a supervised learning approach for ligand-based virtual screening that is highly scalable and capable of strong prospective performance. We build upon our prior virtual screening effort to find small molecule inhibitors of a bacterial protein-protein interaction between the DNA helicase, PriA, and single-stranded DNA binding protein (SSB).<sup>22</sup> The PriA-SSB interaction is critical for maintaining prokaryotic genome stability and as such represents a potential antibiotic target. Our previous effort involved the prospective evaluation of a random forest model on a library of 22,434 compounds. The random forest’s top 250 predictions identified 37 of the 54 active compounds in the library. Here, we examine the applicability and performance of such models when operating on much larger libraries. Our machine learning pipeline selects a random forest model that successfully prioritizes compounds from over 8 million compounds from Aldrich Market Select (AMS). Then, we successfully apply the model on over one billion compounds in the Enamine REAL database, yielding a small, chemically-diverse, highly hit-enriched compound subset. These prospective tests demonstrate a cost-effective approach for navigating very large chemical spaces in the search for active compounds. The random forest model is easy to train and readily scales to current and future on-demand libraries, unlike structure-based approaches.

## 2 Results

Our primary goals were to identify the best supervised learning model for predicting PriA-SSB inhibitors and then prospectively test the model in a virtual screen on the AMS and Enamine REAL libraries that contain 8,187,682 and 1,077,562,987 compounds, respectively. As a first step, we use a training dataset from previous PriA-SSB experimental screening with 427,300 compounds and 554 actives to compare multiple types of supervised learning algorithms, optimize their hyperparameters, and select the top-performing model, a random forest classifier (RF-C). We include classification models (denoted with -C) trained to predict binary activity as well as regression models (denoted with -R) trained to predict continuous % inhibition. Next, we assess the RF-C model’s prospective performance. We compare prioritized compounds from the RF-C model and a Similarity Baseline model based on chemical structural similarity using the AMS library. Approximately 700 prioritized compounds are ordered and tested from each model. After finding the RF-C model recovers more active compounds and more chemically-diverse actives than the baseline, we assess its scalability to the billion compound Enamine REAL library. The RF-C model again achieves a high hit rate of 45.6% and identifies potent compounds that have notable structural differences from those in the training set.

### 2.1 Supervised Learning Hyperparameter Tuning

To determine which supervised learning model would be most effective on large chemical libraries, we systematically explored model architectures in five different model classes to find effective hyperparameter combinations for each class based on cross validation performance. In this stage, we pruned the hyperparameter sets for each model class down to the top 20 performers (Appendix A.7). We split the training dataset into 10 folds and used only folds 0 through 7 in this stage, reserving the last two folds (folds 8 and 9) for the next stage (Table S1). In total, we trained 3,080 models and selected 20 top performers from each of

the five model classes. Because our focus is on early retrieval of actives, we chose these top models based on mean normalized enrichment factor at 1% ( $\text{NEF}_{1\%}$ ) performance (Section 4.4). Figure 1 shows the mean performance for these top 20 models (with ties) for each class. The top RF-C models consistently perform better than all other model classes on the three evaluation metrics. The classification versions of the extreme gradient boosting (XGB) and neural network (NN) models outperformed their regression-based counterparts.

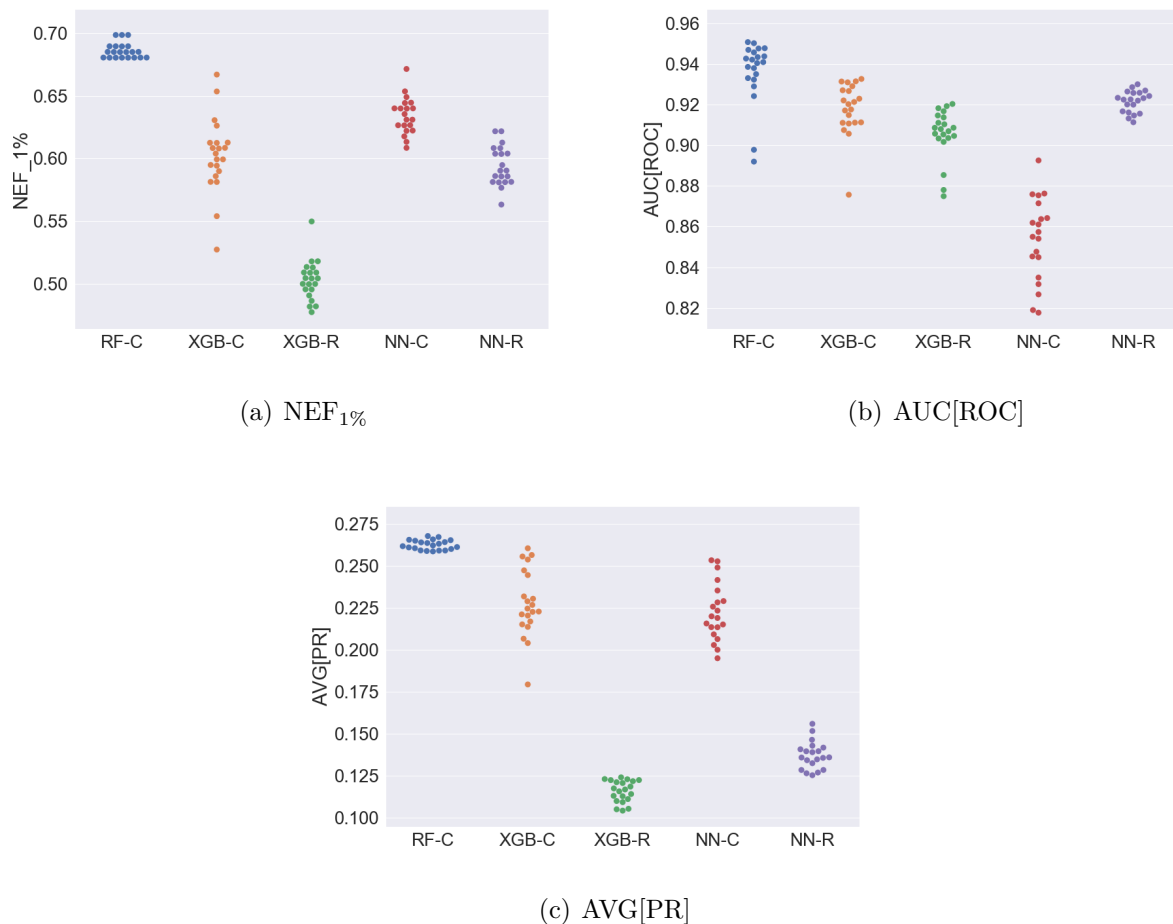


Figure 1: Cross validation mean performance on the top 20 hyperparameter sets from each model class. For the RF-C, XGB-C, and XGB-R model classes, 21 hyperparameter sets are shown due to ties. The -C and -R suffixes denote classification and regression variants, respectively.

## 2.2 Supervised Learning Model Selection

Next, we compared the performance of the top models from each model class on a reserved test fold. Here, the top 20 (with ties) hyperparameter configurations per model class were trained on folds 0-8 (fold 8 used as validation) and tested on fold 9. The results for the top model in each class are illustrated in Table 1. In addition to the individual models, we also examined two ensemble methods: Model-Based Ensemble and Max Vote Ensemble. Across classes, the optimal model is an RF-C model that achieved the best performance on two of the three metrics, NEF<sub>1%</sub> and average precision-recall (AVG[PR]). For prioritizing compounds in a large on-demand library, these two metrics are more relevant than the area under the receiver operating characteristic curve (AUC[ROC]). NEF<sub>1%</sub> explicitly focuses on early hit retrieval, which is the goal when selecting a small fraction of compounds from a large library. AUC[ROC] can provide an overly-optimistic summary of performance when there are far more inactive than active compounds,<sup>23</sup> which is the case in high-throughput screening. Therefore, we selected the RF-C model and the Similarity Baseline for prospective testing. Although the Similarity Baseline model was outperformed by most of the supervised learning approaches, we included it in the prospective testing as a control. It represents a typical strategy for prioritizing compounds to test from chemical libraries. This "hit expansion" strategy simply prioritizes the closest analogs of known actives.

Table 1: Top performance across model classes on the test fold. Each model was trained on folds 0-8 (fold 8 used as validation) and tested on fold 9. The best model is an RF-C model in the two most relevant performance metrics AVG[PR] and NEF<sub>1%</sub>.

Model	Hyperparameter ID	AUC[ROC]	AVG[PR]	NEF <sub>1%</sub>
RF-C	14	0.89	0.19	0.64
XGB-C	140	0.92	0.17	0.58
XGB-R	81	0.88	0.05	0.42
NN-C	47	0.83	0.13	0.58
NN-R	191	0.90	0.07	0.49
Similarity Baseline	-	0.81	0.09	0.40
Model-Based Ensemble	0	0.94	0.17	0.62
Max Vote Ensemble	0	0.94	0.17	0.62

To further explore why the ensemble methods do not improve the performance over the best individual models (Figures S1 and S2), we examined the overlap in actives retrieved between the ensembles and top models (Tables S2-S6). Although ensembles find some actives not prioritized by the best RF-C model, the RF-C finds many more actives leading to better performance. We note several methodological improvements that could improve the ensemble performance in Appendix A.8.

## 2.3 AMS Prospective Screening

Next we applied the RF-C and Similarity Baseline models in a prospective test on Sigma-Aldrich’s AMS library, which consists of 8,187,682 mostly in-stock compounds. First, the RF-C and Similarity Baseline models were re-trained using all 10 data folds. Then, each model was applied to score the AMS library. The top 1,500 ranked compounds from each model were then filtered based on cost, delivery, and availability criteria (Section 4.8). The union of these two prioritized and filtered sets comprised 1,028 unique compounds, which we purchased from Sigma-Aldrich. The average cost, including dissolution in DMSO and plating, was approximately \$40 per compound.

Upon receipt of the plated compounds, we determined that four compounds were incompletely dissolved. These molecules were removed from further consideration, leaving 1,024 for testing. Among the 1,024 compounds, 701 compounds were in the top 1,500 from the RF-C and 705 were in the top 1,500 from the Similarity Baseline, including 382 compounds selected by both models. All 1,024 compounds were tested in duplicate to reveal 412 hits based on our established activity labeling conditions—namely that both replicate tests exceed 50% inhibition and the compound passes a pan-assay interference compounds (PAINS) filter (Section 4.8).



### 2.3.1 Prospective Hit Summary

Recall that the training set consisted of 427,300 compounds with 554 actives, a hit rate of only 0.13%. If the hit rate is similar in the AMS library, we would expect to find on average one hit in random selections of 1,024 compounds. Table 2 illustrates the far superior hit rates for the 1,024 compounds grouped by the selection method. Both models were given similar budgets: 701 and 705 compounds for RF-C and Similarity Baseline, respectively. RF-C outperformed the Similarity Baseline, finding 337 hits compared to the baseline’s 256. Both models prioritized a common set of 382 compounds and thus identified the same 181 hits, but the RF-C model recovers more hits (Table 2). Overall, the two models rank the 8,434,707 compounds differently with a Spearman’s rank order correlation of -0.083.

Table 2: The overlap and hit rates of the AMS compounds selected by the RF-C and Similarity Baseline models.

Selector	Count	Hits	Misses	Hit Rate (%)
RF-C or Similarity Baseline	1,024	412	612	40.23
RF-C	701	337	364	48.07
Similarity Baseline	705	256	449	36.31
RF-C and Similarity Baseline	382	181	201	47.38
RF-C but not Similarity Baseline	319	156	163	48.90
Similarity Baseline but not RF-C	323	75	248	23.22

Figure 2 showcases the hit accumulation as each model is provided a progressively larger screening budget. The screening budget is based on the compounds selected by each model in descending order of score. The RF-C model consistently outperforms the Similarity Baseline, and the gap widens as we increase the budget. It is expected that this gap eventually closes as the budget expands due to the finite active compounds in the AMS library. However, for practical, cost-effective virtual screening, we are most interested in early hit retrieval performance at small budgets.



Figure 2: Budget versus Total Hits for the RF-C and Similarity Baseline models on the 1,024 selected AMS compounds. The RF-C performance is better than the Similarity Baseline for all compound budgets, and the gap increases with the budget. The budget is the number of top-ranked compounds evaluated. The total hits are the actives that would be found by screening only those compounds.

### 2.3.2 Chemical Diversity of the AMS Active Compounds

The ideal virtual screening algorithm will not only prioritize many active compounds but also identify chemically-diverse or structurally-novel hits. Clustering compounds by chemical structure provides one way to assess chemical diversity. We clustered the union of the training set of 427,300 compounds and the 1,024 ordered AMS compounds with a custom implementation of the Taylor-Butina<sup>24,25</sup> method using Tanimoto distances between the compounds’ fingerprint representations, the same chemical features used by the models. We define two diversity metrics: unique cluster hits and novel cluster hits. Using the 412 hits from the AMS library, we counted the number of unique clusters with at least one hit regardless of whether or not these hits appeared in training set clusters. This cluster set is defined as the unique clusters hits metric, which gives a measure of the hit diversity. In addition, we calculated the novel cluster hits metric, which is the number of clusters that contain AMS hits but do not contain any training set hits. Novel cluster hits measures how

well a model was able to generalize to active chemotypes that were either unexplored or show no hits based on the training set labels; this is similar to a novelty measure used by Sturm et al.<sup>26</sup> Table 3 summarizes these cluster hit metrics at a distance threshold of 0.4 (see Table S7 for other thresholds). The RF-C model substantially outperforms the Similarity Baseline in both hit diversity metrics.

Table 3: Summary of cluster hit metrics on the 1,024 AMS selected compounds. Unique cluster hits denotes the number of clusters containing at least one hit. Novel cluster hits denotes the number of clusters containing AMS hits without an active cohort in the training set. Taylor-Butina clustering with a distance threshold of 0.4 was used to compute clusters.

Selector	Unique Cluster Hits	Novel Cluster Hits
RF-C or Similarity Baseline	169	72
RF-C	142	61
Similarity Baseline	115	30
RF-C and Similarity Baseline	88	19
RF-C but not Similarity Baseline	72	44
Similarity Baseline but not RF-C	40	13

There is an association between the clusters defined using chemical structures and a chemical’s % inhibition. We categorized compounds as weak, moderate, or strong actives based on % inhibition ranges (Appendix B.2 and Table S8) and counted the number of weak, moderate, and strong actives in each cluster. The different categories of active compounds do not distribute uniformly across the chemical clusters (Fisher’s exact test p-value 0.0005). Most strong and moderate actives concentrate in a few clusters.

In addition to identifying more active compounds than the Similarity Baseline, the RF-C model also prioritizes compounds that are less similar to the training set actives. The Tanimoto distances from the RF-C model’s prioritized compounds to the most similar training set active tend to be larger than those from the Similarity Baseline (Figure 3(a)). Inspecting the five least similar, experimentally-confirmed hits from the RF-C model (Table S9) and the Similarity Baseline (Table S10) illustrates these differences. The maximum distance from RF-C is 0.57 compared to a maximum distance of 0.32 for the Similarity Baseline. The Similarity Baseline is based solely on chemical similarity to the known actives so its prioritized

compounds are mostly minor variations on these actives. In contrast, the RF-C model uses information from both active and inactive training instances to rank compounds. Although there are common substructures with the known actives, the AMS hits from RF-C are more distant overall. The distance between the prioritized AMS compounds and their nearest known active is not predictive of whether the prioritized compound will be active. The distance distributions for AMS actives and inactives are similar when considering all 1,024 ordered compounds (Figure 3(b)), the compounds selected by the RF-C model (Figure 3(c)), or the compounds selected by the Similarity Baseline (Figure 3(d)).

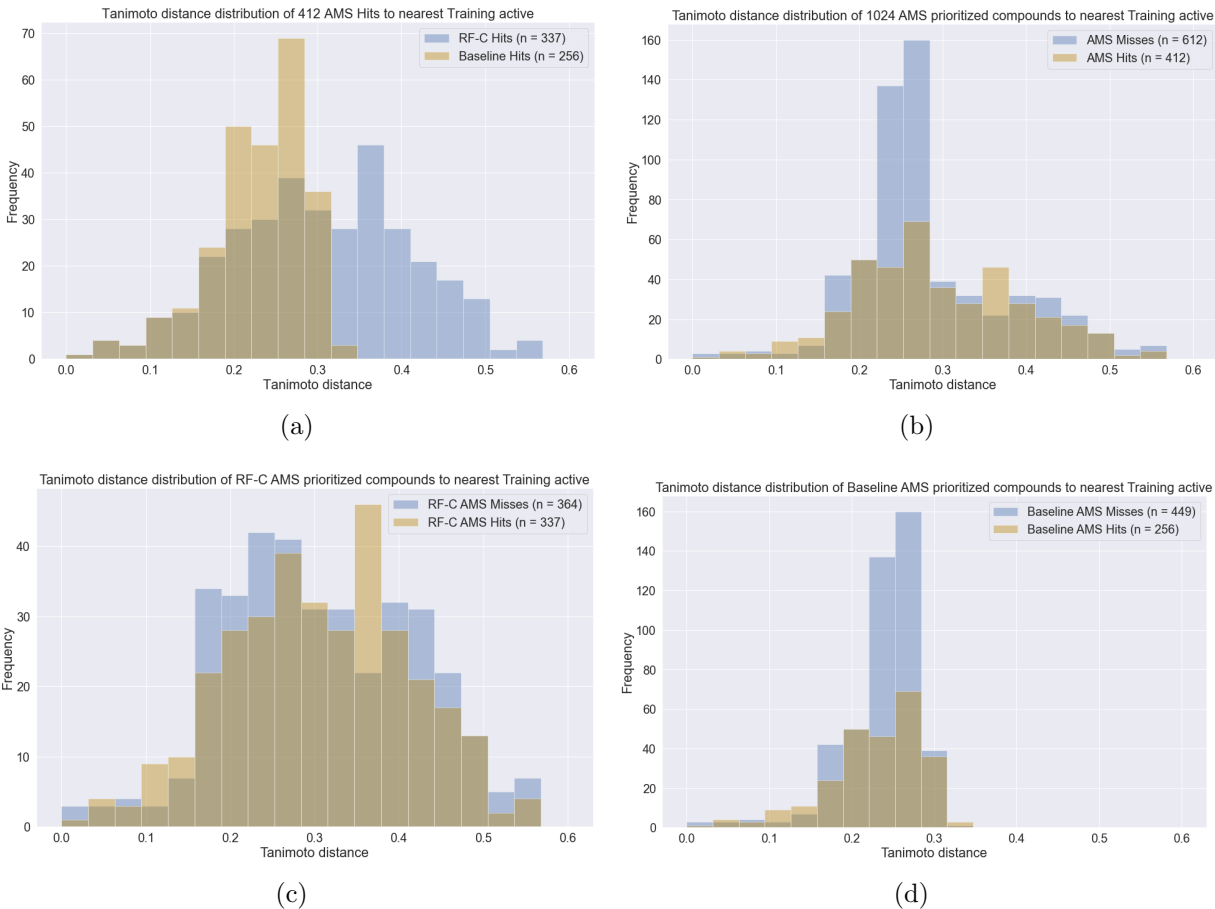


Figure 3: Tanimoto distance distributions of the 1,024 AMS compounds to nearest training set actives. Denotes distribution of (a) 412 AMS hits based on the model (RF-C or Similarity Baseline), (b) 1,024 AMS compounds based on hits and misses, (c) 701 RF-C AMS compounds based on hits and misses, and (d) 705 Similarity Baseline AMS compounds based on hits and misses.

## 2.4 Enamine REAL Prospective Screening

We used the Enamine REAL dataset to demonstrate how our supervised learning workflow can scale to much larger on-demand chemical libraries, maintain high predictive accuracy, and recover highly potent active compounds. Scoring all 1,077,562,987 compounds in the library with the RF-C model requires only tens of hours when parallelized on 18 servers with modest resources (Appendix B.3 and Table S11). We sorted all compounds by their RF-C score and requested a quote for the top-ranked 10,000 compounds. 5,620 of the 10,000 could be delivered in less than a month, and we discarded the rest. Figure 4 (top panel) shows the Tanimoto distance distribution of these 5,620 compounds to their nearest Training or AMS active. Several Enamine compounds closely resemble known active compounds from the training dataset with distances as low as 0.0. However, many compounds are far from their closest training or AMS active with distances exceeding 0.5. Because these compounds already represent the top 0.001% of predictions from the Enamine REAL pool, we next optimized the chemical diversity and distance from the known actives (Section 4.9) and selected 68 for screening. These 68 screened compounds have Tanimoto distances to the most similar active ranging from 0.35 to 0.62 (Figure 4).

### 2.4.1 Enamine REAL Hits

To determine the hits for the 68 Enamine REAL compounds, we generated dose-response curves. We first applied the same hit assignment criteria as the AMS screen and identified 31 initial hits, a 45.6% hit rate (Figure 5). The training set has a 0.13% hit rate, so a random selection of 68 compounds is expected to yield no hits. Our prioritization process selected top scoring compounds from different clusters in the prioritized Enamine set, which were also distinct from clusters with active training or AMS compounds, so that the 31 active compounds each represent a unique cluster by design. Similar to the AMS screen, the Enamine active and inactive compounds have no substantial difference in their Tanimoto distances to the most similar known active (Figure 4).

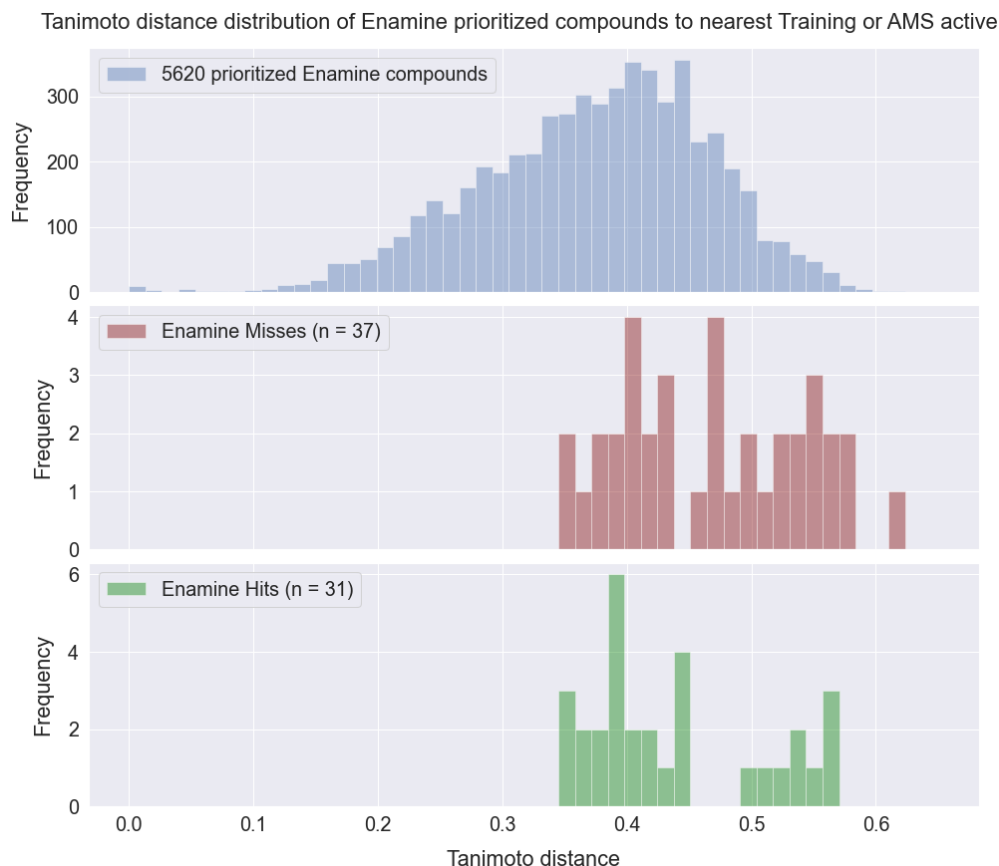


Figure 4: Tanimoto distance distribution of the 5,620 prioritized and 68 screened Enamine REAL compounds to their nearest Training or AMS active. Middle and bottom panels show the Tanimoto distance distributions of the inactive and active screened compounds using the initial hit criteria.

When we examined the full dose response curves, we confirmed 28 compounds as hits with IC<sub>50</sub> values ranging from 1.3 to 37.8  $\mu$ M (Table 4). Half of these hits had IC<sub>50</sub> values of 10.0  $\mu$ M or less. The three most potent compounds (Figure 6) have the same most similar known active compound, which is from the training set. These three and the nearest active training instance share a common scaffold comprised of pyridine and pyrimidine rings spanned by a hydrazone linker. Chemical differences reside on the 4- and 5-positions on the pyrimidine substituent, ortho and para to the linker. Whereas the training compound has 4-chloro and 5-amino substituents, Z1763598930 has only a 5-chloro substituent. In Z734854148 and Z50106757, these positions serve as bridge carbons in purine and thienopyrimidine bicycles, respectively. Several of the confirmed hits contain the 1,10-phenanthroline

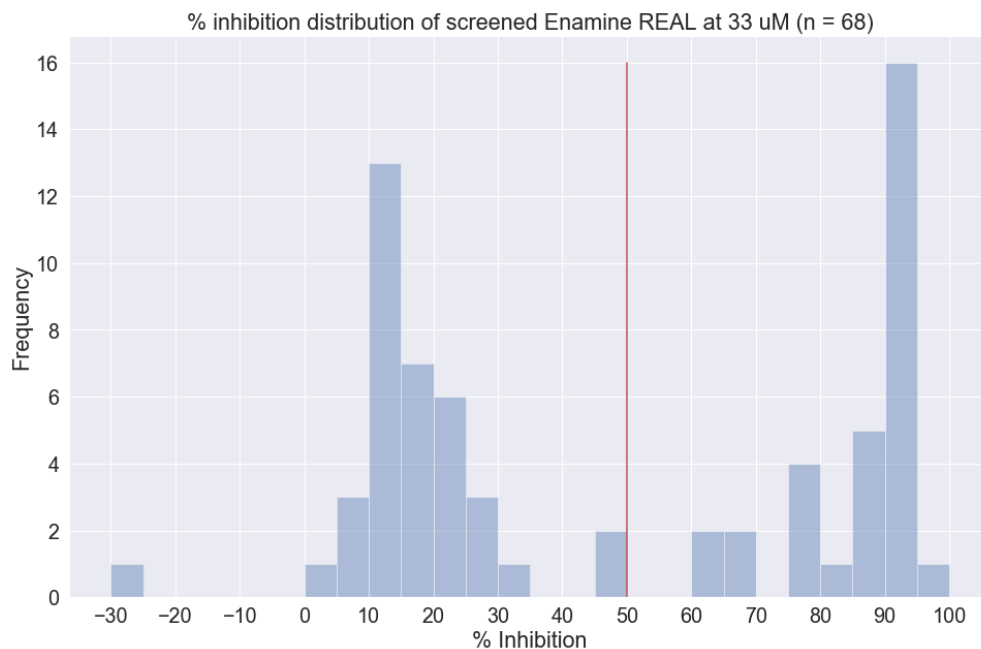


Figure 5: Median % inhibition distribution of the 68 screened Enamine REAL compounds at 33  $\mu$ M. Compounds with % inhibition of at least 50 (red line) are considered initial hits.

substructure, which we previously identified as an active compound. However, the new Enamine actives contain extensive modifications to this shared substructure with Tanimoto distances to the nearest known active that are greater than 0.5. Even though the RF-C model was only given a budget of 68 compounds to explore the billion compound Enamine library, it identified tens of active compounds that are non-trivially related to all of the known actives in the training data and AMS screen.

Table 4: The 28 confirmed hits from the Enamine REAL compounds based on dose response curves. The most similar known active compound from the training set or AMS compounds is shown along with its Tanimoto distance to the Enamine hit. The similarity maps from RDKit show similar substructures in green and dissimilar substructures in red. The IC<sub>50</sub> values are shown along with the 95% lower and upper confidence limits.

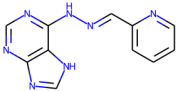
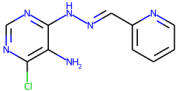
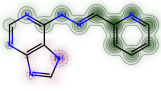
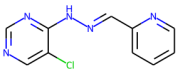
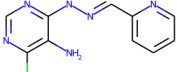
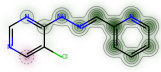
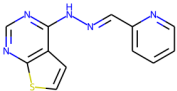
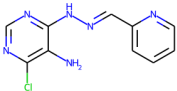
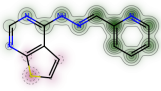
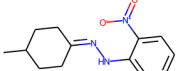
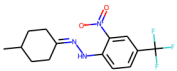
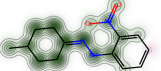
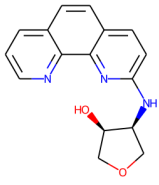
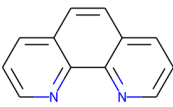
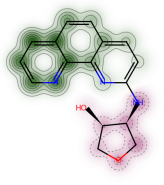
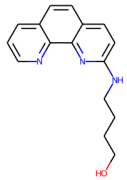
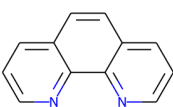
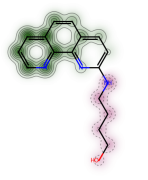
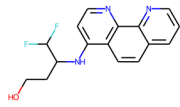
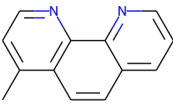
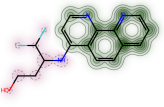
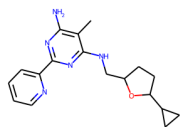
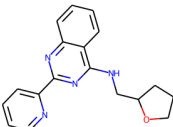
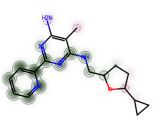
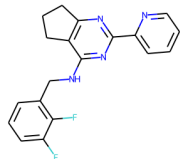
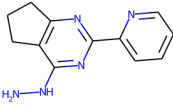
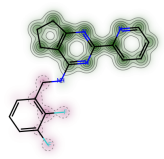
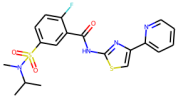
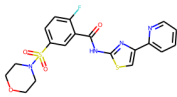
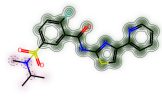
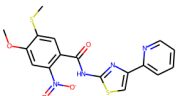
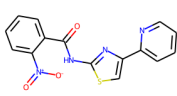
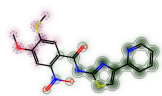
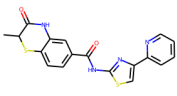
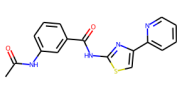
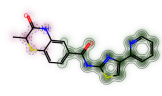
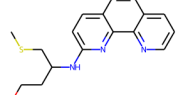
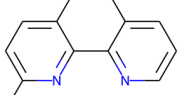
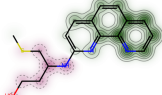
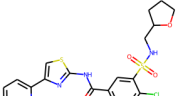
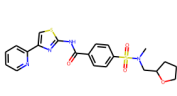
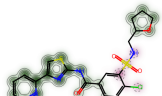
Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC <sub>50</sub> $\mu$ M (lower, upper)
 Z734854148			0.45	1.3 (1.2, 1.4)
 Z1763598930			0.38	1.7 (1.6, 1.8)
 Z50106757			0.45	2.7 (2.5, 2.9)
 Z49571468			0.35	3.6 (3.1, 4.2)
Continued on next page				



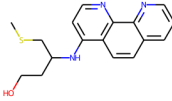
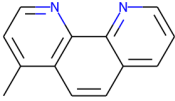
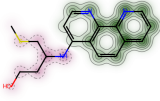
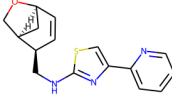
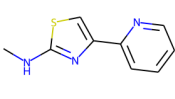
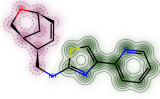
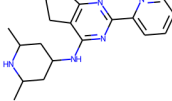
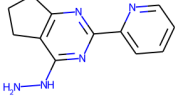
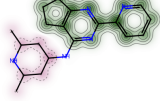
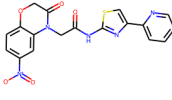
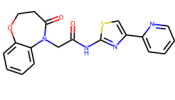
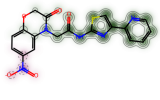
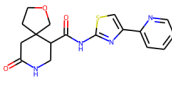
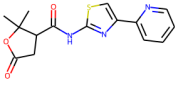
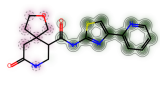
Table 4 – Continued from previous page.

Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC50 $\mu$ M (lower, upper)
 Z3559392588			0.54	4.3 (4.0, 4.5)
 Z3558428795			0.53	4.3 (4.1, 4.6)
 Z3559518523			0.57	4.5 (4.2, 4.8)
 Z2976359863			0.54	5.4 (5.1, 5.8)
 PV-001914484112			0.41	6.1 (5.2, 7.2)
Continued on next page				

**Table 4 – Continued from previous page.**

Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC50 $\mu$ M (lower, upper)
 Z339655406			0.36	6.8 (6.5, 7.2)
 Z1101543176			0.35	7.0 (6.0, 8.2)
 Z29466207			0.4	8.0 (6.6, 9.8)
 Z3557629875			0.57	9.1 (8.4, 9.8)
 Z240359708			0.39	10.0 (6.0, 16.8)
Continued on next page				

**Table 4 – Continued from previous page.**

Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC50 $\mu$ M (lower, upper)
 Z3557629872			0.56	10.8 (10.0, 11.7)
 Z3295209363			0.55	11.5 (10.7, 12.4)
 Z3395547714			0.38	12.1 (11.2, 13.0)
 Z109132782			0.36	12.4 (11.6, 13.3)
 Z3297024656			0.44	12.6 (11.8, 13.4)
Continued on next page				

**Table 4 – Continued from previous page.**

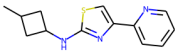
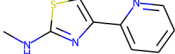
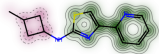
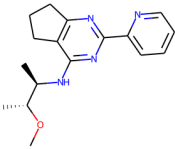
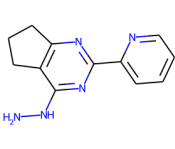
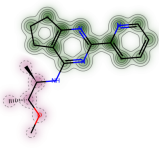
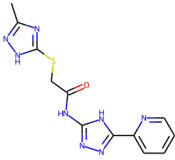
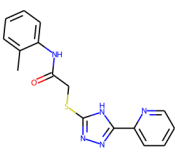
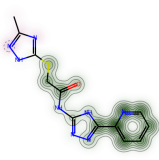
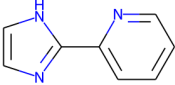
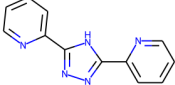
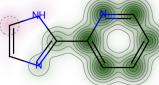
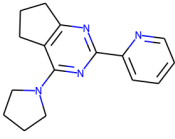
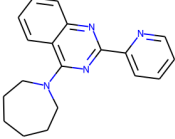
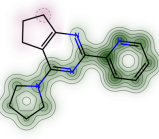
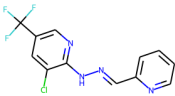
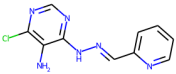
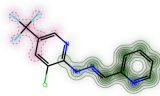
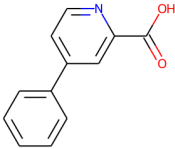
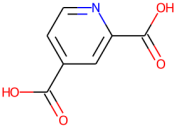
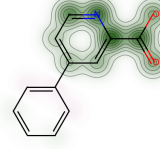
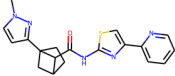
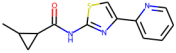
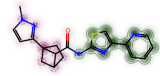
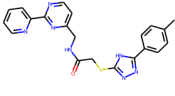
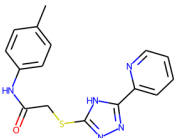
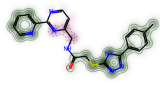
Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC <sub>50</sub> $\mu$ M (lower, upper)
 Z3295052620			0.39	13.1 (12.0, 14.3)
 Z3649501061			0.39	13.9 (13.2, 14.7)
 Z1313381195			0.4	15.5 (12.1, 19.9)
 Z1245633363			0.43	15.7 (14.3, 17.3)
 Z1172208679			0.41	17.6 (12.2, 25.4)
Continued on next page				

Table 4 – Continued from previous page.

Enamine Hit	Nearest Active	Similarity Map	Tanimoto Distance	IC50 $\mu$ M (lower, upper)
 Z49711282			0.5	20.2 (15.8, 26.0)
 Z1741972899			0.41	21.0 (16.6, 26.5)
 Z3414110258			0.49	22.2 (19.6, 25.0)
 PV-002421112068			0.4	37.8 (22.7, 63.0)

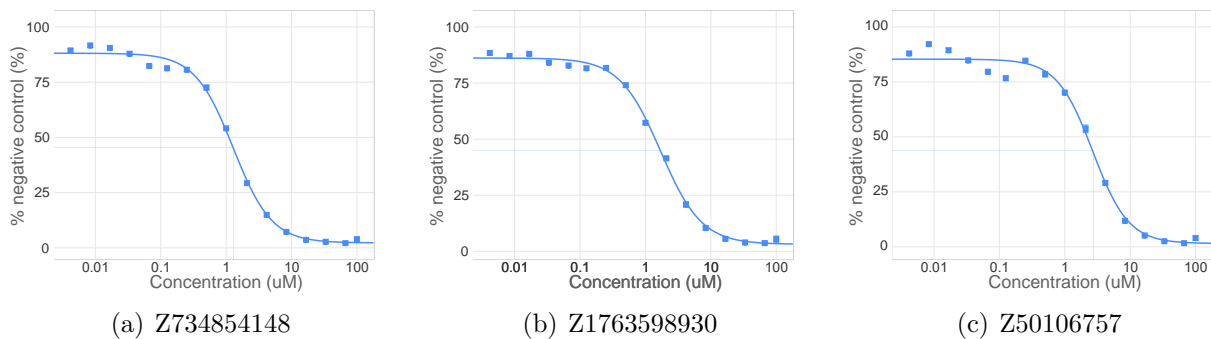


Figure 6: Dose response curves for the three Enamine compounds with the lowest IC<sub>50</sub> values.

### 3 Discussion

Despite the many advances in machine learning approaches for ligand-based virtual screening, few of these algorithms and pipelines have been validated with prospective chemical screens or compared with appropriate baseline methods.<sup>27</sup> Machine learning and virtual screening tools are starting to be applied to score and prioritize millions to billions of compounds.<sup>12,28–30</sup> There are some examples of experimentally evaluating those predictions,<sup>10,11,18–21,31–34</sup> but they remain rare. The strong results in our prospective evaluation of PriA-SSB inhibitor predictions support the potential for similar machine learning-based virtual screening in other drug discovery campaigns. The random forest model we selected generalized well to the AMS library of over 8 million compounds and the on-demand Enamine REAL library with over a billion compounds. These putative active compounds require further validation. The presence of false positive compounds in the training data, such as AlphaScreen frequent hitters,<sup>35</sup> could lead to prioritizing similar false positives in the AMS and Enamine REAL collections. Nevertheless, the high hit rates, potency, and chemical diversity of our prospective screens is quite encouraging.

Our PriA-SSB case study revealed some challenges of virtual screening on very large commercial chemical libraries and limitations of our current approach. Compound availability in

large libraries can change, especially when manually obtaining quotes for the desired compounds. Real-time, automated access to available compounds could improve the screening process. In addition, diversity of hits is important because it enables fruitful lead optimization. Our random forest was trained only to predict inhibitors and was not concerned with chemical diversity. For the AMS library, we did not filter the selected compounds based on diversity. Rather, we ranked them by predicted activity score and assessed chemical diversity post hoc. When moving to larger on-demand libraries like Enamine REAL, diversity requires more formal attention. There are many highly similar or redundant compounds, so prioritizing based on predicted activity alone can concentrate the selected compounds in an undesirably small number of chemical clusters. When selecting Enamine REAL compounds, we used heuristics such as cluster membership and Tanimoto distance from known actives to promote diversity. Related work prioritized the compound with the highest predicted score per cluster.<sup>20</sup>

We primarily used Taylor-Butina<sup>24,25</sup> clustering to quantify chemical diversity because it groups compounds with similar chemical structures, but there is no universally accepted way to calculate the diversity of a compound pool. Taylor-Butina clustering is susceptible to the order in which the compounds are processed and can result in different clusterings. For example, an active may have been clustered earlier than an inactive that is within cutoff distance. However, in three settings of the Taylor-Butina algorithm with varying distance thresholds, our conclusions about the diversity of actives are consistent. The RF-C model outperforms the Similarity Baseline in both unique and novel cluster hits (Tables 3 and S7), confirming it can prioritize novel chemical structures that are not present in the training data.

There are open questions about when our approach to virtual screening on very large compound libraries is applicable. In this study, we had access to a large training set with 427,300 screened compounds and 554 actives. A typical target will have far fewer screened compounds and known actives at the start of the drug discovery process. Virtual screen-

ing will have greater impact if it can succeed without requiring an initial high-throughput chemical screen to train models. Future work can explore the tradeoffs between training set size and prospective accuracy in compounds selected from large libraries. The amount of available training data may also affect the choice of machine learning model for virtual screening.

Accurate virtual screening algorithms are essential for expanding into on-demand chemical libraries in which purchasing and screening all compounds is impossible. With an efficient and accurate computational strategy to explore that large chemical space, the best case outcome is far superior to a traditional high-throughput chemical screen. The large libraries can provide access to many more hits and higher quality hits with respect to chemical diversity and other desirable properties, as exemplified by the potent PriA-SSB inhibitors we identify. Our simple supervised learning approach has a high hit rate, good chemical diversity, and can scale to on-demand libraries. Scoring over one billion compounds takes less than 1,000 CPU hours and can be trivially parallelized to score even larger chemical libraries in the future. In contrast, the computational costs of structure-based scoring of a library of this magnitude are prohibitive. One recent structure-based study required approximately 5 million CPU hours to evaluate 1.3 billion compounds.<sup>11</sup> Another used 27,612 GPUs to score 1.37 billion compounds in less than 24 hours.<sup>30</sup>

Virtual screening of ultra-large on-demand libraries presents both opportunities and new challenges. One concern is the possibility of getting buried by false positives due to biases or quirks in the scoring model that are not detected during model validation.<sup>36,37</sup> Models may erroneously recognize uncommon or exotic chemical features as important to activity due to limitations or mislabeled instances in a ligand-based model’s training set or unsuitable parameters in a structure-based model’s scoring function. The mishandled molecular features could be rare enough in smaller libraries that they account for only a small number of false positives and go unnoticed during model development. However, the sheer size and chemical diversity of ultra-large libraries could cause the top scoring compounds to



be dominated by such errors. We are encouraged that these potential issues did not arise in our large-scale ligand-based virtual screen or the largest structure-based virtual screens noted above. Although innovative algorithms will continue to advance virtual screening capabilities, probing the practical challenges of computationally-guided screening in ultra-large libraries necessitates prospective experimental testing.<sup>38</sup> The success of our two prospective screens showcases supervised learning as a powerful approach for navigating large on-demand chemical libraries in future drug discovery campaigns.

## 4 Methods

### 4.1 PriA-SSB Datasets

We trained virtual screening models on new and previously generated datasets for the PriA-SSB target.<sup>22,39</sup> This target is a protein-protein interaction that is involved in bacterial DNA replication and was considered as a candidate target for antibiotics development.<sup>40</sup> The compounds in these datasets come from four batches of a Life Chemicals Inc. (LC) library designated LC1-LC4 and the Molecular Libraries Probe Production Centers Network (MLPCN) library (Table 5). The compounds considered for prospective evaluation come from the in-stock AMS library and the on-demand Enamine REAL library.<sup>3</sup>

Table 5: Chemical libraries and compound counts. The training dataset is merged from the LC1234 and MLPCN libraries.

Stage	Library	# Compounds
Pre-processing	LC123 (Primary and Retest)	74,763
Pre-processing	LC4 (Primary)	25,278
Pre-processing	MLPCN (Primary and Retest)	337,104
Hyperparameter Tuning Model Selection Prospective	Training set	427,300
Prospective	AMS	8,187,682
Prospective	Enamine REAL	1,077,562,987

#### 4.1.1 Training Data Chemical Screening and Preprocessing

We screened the MLPCN compound library using an AlphaScreen (AS) assay at a single concentration (33.3  $\mu$ M) following the same protocol previously used to screen the LC1234 libraries.<sup>22,39</sup> Briefly, library compounds and controls were dispensed from 10 mM DMSO stocks into white 1,536-well plates with an Echo 550 acoustic liquid handler. Next, 3  $\mu$ L of a master mix was added to each well using a Mantis liquid handler with a high-volume silicone chip to yield a final reaction mixture containing 10 mM HEPES-HCl (pH 7.5), 150 mM NaCl, 1 mM MgCl<sub>2</sub>, 10 mM dithiothreitol, 1 mg/mL bovine serum albumin, 0.01% Triton X-100, 0.1  $\mu$ M PriA (with N-terminal 6XHis tag), 0.1  $\mu$ M Biotinylated-SSBct (N-Biotin-Trp-Met-Asp-Phe-Asp-Asp-Asp-Ile-Pro-Phe-C), 5  $\mu$ g/mL of both AS acceptor and donor beads, and 33.3  $\mu$ M of compound. For positive and negative control wells, the compound was replaced with 25  $\mu$ M SSBct peptide (N-Trp-Met-Asp-Phe-Asp-Asp-Asp-Ile-Pro-Phe-C) or 25  $\mu$ M  $\Delta$ FSSBct (N-Trp-Met-Asp-Phe-Asp-Asp-Asp-Ile-Pro-C), respectively. Plates were centrifuged briefly and rocked for an hour at room temperature. Then, the AS signal of each well was measured with a PheraStar plate reader using a 0.1 s settling time, 0.3 s excitation, a 0.04 s delay, and a 0.6 s integration time. AS signals for each well were adjusted as previously described to reduce the impact of plate and edge effects.<sup>39</sup> We calculated a Z' factor for each plate<sup>41</sup> and repeated plates with Z' < 0.5. In addition, we calculated the % inhibition relative to the controls for each compound. We retested compounds with  $\geq$  35% inhibition and that passed PAINS filters<sup>42,43</sup> with the same assay to confirm activity.

We compiled the training dataset by merging the MLPCN screening data with the LC1234 screening data (Table 5). We defined active compounds, also referred to as hits, by requiring the median % inhibition of the primary screens is  $\geq$  35%, the median % inhibition of the retest screens is  $\geq$  35%, and the compound does not match a PAINS filter (Appendix A.1). The merged training dataset contained 427,300 compounds with 554 actives. We split the training data into 10 folds for cross validation. The splitting method takes into account library information (LC1234 and MLPCN) by grouping the molecules based on their libraries

and then stratifying each of these groups into 10 folds that maintain roughly the same binary activity label distribution.

## 4.2 Feature Generation and Clustering

We used RDKit Morgan fingerprint features<sup>44,45</sup> for all virtual screening methods. We converted each compound SMILES to an RDKit mol object, removed salt counter ions using RDKit’s SaltRemover function, and generated a Morgan fingerprint with length 1,024 and radius 2. Using the fingerprints, we then defined molecule relationships based on Tanimoto distance, which is equivalent to Jaccard distance. To judge chemical diversity, we used Taylor-Butina<sup>24 25</sup> clustering at various Tanimoto distance thresholds (0.2, 0.3, and 0.4). Taylor-Butina takes a distance threshold as input that forces compounds within the same cluster to be within that threshold. Table S12 summarizes the number of compounds, number of clusters, and number of unique clusters that are in each cross-validation fold. The number of unique clusters that are in a given fold but not the others can measure a virtual screening model’s ability to generalize. Because we generated cross-validation folds by splitting on library and activity instead of cluster membership, the clusters are not uniformly distributed across folds.

## 4.3 Virtual Screening Methods

We considered a variety of supervised learning ligand-based virtual screening algorithms. Most of these could be trained in a classification setting to predict binary activity as well as a regression setting to predict the % inhibition, which is indicated by appending -C or -R, respectively, to the model name.

**Random Forest (RF):** A RF model consists of a collection of base learners, typically decision trees.<sup>16</sup> Each base learner is trained on random subsamples of the training data with replacement. This process is known as bootstrap aggregation (bagging), which helps to combat overfitting.<sup>46</sup> Furthermore, individual decision trees are split on random sub-

sets of the features, which further helps to combat overfitting.<sup>16,47</sup> Classification is done by aggregating the votes from the base learners. We used the scikit-learn random forest implementation.<sup>48</sup> The RF hyperparameters include the number of base learners, number of features, leaf node samples, and class weights (Table S13). Because RF-R regression models grew too large under some hyperparameter settings, we ultimately only considered RF-C classification models.

**eXtreme Gradient Boosting (XGB):** XGB is an ensembling method that is based on the concept of gradient boosting.<sup>49–51</sup> It builds the base learners sequentially, where each learner is built to reduce a loss function whose terms include the gradients (residuals) of previously built learners. As a result, each consecutively built learner aims to correct the mistakes of past learners using the gradients. We use the XGBoost Python package implementation.<sup>51</sup> The XGB hyperparameters involve varying the maximum depth, learning rate, and number of estimators (Tables S14 and S15).

**Neural Network (NN):** A NN consists of a series of hidden layers corresponding to weight matrices followed by non-linear activation functions. The process of forward propagating the input along the hidden layers via matrix multiplication is repeated until a final output layer is reached, which makes a prediction. The weight matrices are trained by gradient descent on the loss of the output; the technique is called backward propagation. Our implementation uses Keras<sup>52</sup> with the Theano backend<sup>53</sup> and the Adam optimizer.<sup>54</sup> The NN hyperparameters involve varying the learning rate, drop out, and multi-layer perceptron architecture (Tables S16 and S17).

**Ensembles:** The ensemble models combine predictions from multiple existing models, which are referred to as base models. Max Vote Ensemble applies the max function and outputs the largest value from the base models. The Model-Based Ensemble is a stacking ensemble method,<sup>55</sup> which consists of two layers: several base models as the first layer and another classifier as the second layer. In the first layer, multiple classification and regression models (the base models) output the predicted values for each molecule. In the second

layer, another classifier (the ensemble model) is trained to balance the output values from the base models. In this way, the simple classifier from the second layer can ensemble the performance from different base models. Finally, all the base models are retrained on the complete training data, passing through an ensemble model to get the final prediction value. Both ensemble methods incorporate a subset of the best models from the hyperparameter tuning stage and have 13 hyperparameter sets that vary the base models used (Table S18). Details of the ensemble training are described in Appendix A.8 and Figure S3.

**Similarity Baseline:** The baseline model represents a simple strategy for prioritizing compounds based on their similarity to known actives.<sup>56,57</sup> It ignores the inactive compounds in the training data. Given the known actives in the training data and a query molecule, the Similarity Baseline calculates the Tanimoto similarity between each active and the query. The maximum Tanimoto similarity is returned as the query molecule’s score. Therefore, query molecules whose fingerprints are most similar to those of known actives are prioritized.

## 4.4 Performance Metrics

The computational models generate different types of scores, such as the probability a compound is active or the predicted % inhibition. In all cases, compounds can be sorted by these scores so that those most likely to be active are first. We consider performance metrics that are based only on these compound ranks. For instance, one can move from most likely to be active to least likely, applying thresholds to compute the true-positive rate (TPR), false-positive rate (FPR), recall, and precision. If one plots the TPR versus FPR and computes the area under the curve (AUC), the resulting metric is the AUC of the Receiver Operating Characteristic curve (AUC[ROC]). The AUC[ROC] serves as a general metric for comparing

performance among models as it does not focus on early or late retrieval exclusively.

$$TPR = \frac{TP}{TP + FN}, \quad FPR = \frac{FP}{FP + TN} \quad (1)$$

$$Recall = \frac{TP}{TP + FN}, \quad Precision = \frac{TP}{TP + FP} \quad (2)$$

Similarly, the average precision (AVG[PR]) summarizes a precision-recall curve across different thresholds. Using scikit-learn’s<sup>48</sup> implementation, the AVG[PR] metric computes the precision  $P$  and recall  $R$  at each threshold  $k$ , then sums as follows:

$$AVG[PR] = \sum_k (R_k - R_{k-1}) P_k \quad (3)$$

This avoids interpolation issues with computing the AUC of the PR curve. The AVG[PR] gauges the overall retrieval and correctness of actives as we increase the threshold.

Enrichment factor (EF) computes the ratio between the number of actives found by a model in its top  $k$  selected compounds versus a random selection of  $k$  compounds. We normalize EF by the maximum EF possible for a perfect model. Given a fraction  $R \in [0\%, 100\%]$ , we compute the normalized enrichment factor ( $NEF_R$ ) as follows:

$$EF_R = \frac{\# \text{ actives in top } R \text{ ranked compounds}}{\# \text{ actives in entire library} \times R} \quad (4)$$

$$EF_{max,R} = \frac{\min\{\# \text{ actives, total } \# \text{ compounds} \times R\}}{\# \text{ actives in entire library} \times R} \quad (5)$$

$$NEF_R = \frac{EF_R}{EF_{max,R}} \quad (6)$$

We use  $NEF_{1\%}$  to assess the early retrieval performance in the top 1% of compounds ranked by a model.

## 4.5 Pipeline Overview

We follow a four stage pipeline: hyperparameter tuning, model selection, and two prospective screening stages. The hyperparameter tuning stage considers many hyperparameter combinations and filters them to the top 20 per model class (with ties). The model selection stage includes the best models from the hyperparameter tuning stage and introduces ensembles that combine these best models. This stage selects a single model for prospective evaluation. The AMS prospective stage uses the best selected model from the model selection stage and the Similarity Baseline to prioritize compounds from the AMS dataset. The Enamine REAL prospective stage uses that same best selected model from the model selection stage to prioritize compounds from the Enamine REAL dataset. Table 6 summarizes the hyperparameters per stage for each model class.

Table 6: The number of hyperparameter configurations evaluated for each model class at each stage. The pipeline culminates with a single top performing model. The Similarity Baseline was added as a control in the model selection and AMS prospective stages.

Model	Hyperparameter Tuning	Model Selection	AMS Prospective	Enamine Prospective
RF-C	216	21	1	1
XGB-C	1,000	21	-	-
XGB-R	1,000	21	-	-
NN-C	432	20	-	-
NN-R	432	20	-	-
Similarity Baseline	-	1	1	-
Model-Based Ensemble	-	13	-	-
Max Vote Ensemble	-	13	-	-
Total	3,080	130	2	1

## 4.6 Hyperparameter Tuning

The purpose of the hyperparameter tuning stage is to prune the large number of hyperparameter settings to the top 20 (with ties) for each of the base model classes: RF-C, XGB-C, XGB-R, NN-C, and NN-R. For this stage, we only use the first eight folds of the 10-fold training set. The last two folds are reserved for building ensemble models and assessing test

set performance in the subsequent model selection stage. For each hyperparameter setting, we conducted four cross-validation runs with different combinations of the first 8-folds (Table S1). For each cross-validation run, we record the test fold performance on AUC[ROC], AVG[PR], and NEF<sub>1%</sub>.

## 4.7 Model Selection

We consider the top 20 (with ties) hyperparameter sets from the hyperparameter tuning stage for each base model class based on mean performance of NEF<sub>1%</sub>. This gives a total of 103 selected models (3 are selected due to ties). We train all of the candidate models (103 base models + 1 Similarity Baseline + 26 ensembles for a total of 130 models) a single time. We use folds 0 through 7 for training, fold 8 for validation, and fold 9 for testing. This yields a total of 130 performance measures on AUC[ROC], AVG[PR], and NEF<sub>1%</sub>. The single top performing model is selected for the two prospective screening stages. In the RF model class, three models were tied in NEF<sub>1%</sub>. To break the tie, we used AVG[PR] to select the best RF model.

Our original intention was to select the final prospective model based on fold 9. From Table 1, this would be RF-C model with hyperparameter ID 14. However, the final prospective model was inadvertently selected based on hyperparameter tuning stage performance instead. This best model was also an RF-C model with hyperparameter ID 139 and fold 9 performance 0.94, 0.17, and 0.62 for AUC[ROC], AVG[PR], and NEF<sub>1%</sub>, respectively. This RF-C model is still better than the top performers from other model classes on the NEF<sub>1%</sub> metric that we used to select the model. Furthermore, selecting on the hyperparameter tuning stage performance is still a valid approach to select a model for the prospective evaluation because it uses cross validation.



## 4.8 AMS Compound Selection and Screening

The AMS prospective stage uses two models: the best performing model from the model selection stage (an RF-C model) and the Similarity Baseline. Both models are retrained on all 10 folds and then predict scores on the AMS library, which contains 8,187,682 compounds after removing compounds that are also in the training data. First, the RF-C and Similarity Baseline models each select the top 1,500 ranked compounds based on their prediction scores. This produces two lists with many compounds present in both lists. In an effort to maximize the number of compounds that could be purchased, we removed compounds that cost  $> \$75$  for the smallest sample, had delivery time  $> 21$  days, or came from vendors providing fewer than six compounds in our list. After filtering, we took the union of the remaining top 600 compounds from the RF-C model and the top 600 from the Similarity Baseline model. We ordered the 1,028 unique compounds in the union that were still available for purchase from AMS. When evaluating the hit rate for the RF-C and Similarity Baseline models, we considered not only their top 600 ranked predictions after filtering but rather all compounds we ordered that were in their original top 1,500 ranked compounds. For example, the compound ranked 73 by the RF-C and 986 by the Similarity Baseline is considered to be predicted by both models in Table 2.

After receiving the 1,028 compounds, we conducted two replicate screens following the same protocol described above for the MLPCN library. We excluded four compounds because they could not be dissolved and plated. We analyzed the % inhibition distributions for the 1,024 screened compounds for each replicate individually to determine the hit criteria (Figure S4). For a large random screen, we could set the % inhibition cutoff some standard deviations above the mean.<sup>41</sup> Thus, we could use the same MLPCN threshold of 35% inhibition. However, these AMS screens were targeted towards likely active compounds. Therefore, we applied a more stringent cutoff of 50%, which is the smaller % inhibition distribution mean from the two replicates (Figure S4). In addition to requiring at least 50% inhibition in both replicates, we also required active compounds to not match a PAINS filter.<sup>42,43</sup>

## 4.9 Enamine REAL Computational Scalability, Compound Selection, and Screening

The Enamine REAL database<sup>3</sup> consisted of 1,077,562,987 compounds when we downloaded it on October 11, 2019. To estimate computational scalability of scoring compounds with the RF-C model from the model selection stage, we split the REAL dataset into 18 batches of about 60.3 million compounds each. Each batch was run as a job on a generic CPU compute node at the University of Wisconsin-Madison Center for High-Throughput Computing using HTCondor.<sup>58</sup> Each job processes the batch by generating the chemical fingerprint features and then making activity predictions using the RF-C model. The model was trained only on the training set, not the 1,024 AMS compounds.

After scoring the entirety of Enamine REAL, we requested a quote for the top 10,000 ranked compounds. We then pruned the list to 5,620 compounds based on availability of starting materials and delivery constraints. We clustered the training set compounds, the 1,024 AMS compounds, and the 5,620 Enamine REAL compounds using Taylor-Butina with a 0.4 distance cutoff. We used these clusters to select a final list for purchase, seeking compounds that were predicted to be highly active, were chemically diverse, and were chemically dissimilar from known active compounds in the training set and AMS set.

To emphasize novel chemical structures, we retained only Enamine REAL clusters that did not contain an active compound from the training set or the AMS set. We were interested only in clusters that the model predicts to have actives despite not belonging to a cluster with known actives. This filter reduced the Enamine REAL list from 5,620 to 2,679 compounds.

Next, we retained only compounds that pass the PAINS<sup>42</sup> filter, Inpharmatica filter, and Lipinski’s rule of five. The PAINS filter used RDKit’s FilterCatalog. The Inpharmatica and Lipinski filters used `rd.filters` ([https://github.com/PatWalters/rd\\_filters](https://github.com/PatWalters/rd_filters)).<sup>59</sup> This filtering step reduced the Enamine REAL list from 2,679 to 1,604 compounds. Because over 1,000 compounds remained, we further emphasized chemical diversity by only retaining compounds that had a Tanimoto distance  $\geq 0.35$  from the closest training set active and

AMS active. This filter reduced the Enamine REAL list from 1,604 to 1,354 compounds.

For the remaining clusters, we selected the highest scoring predicted active as the representative. This selection reduced the Enamine REAL list from 1,354 to 311 compounds. Finally, we selected 100 diverse compounds from the remaining list via iterative Tanimoto distance selection. In the first iteration, the compound with highest activity prediction is selected. Then, the greedy method iteratively selects the candidate compound that is most distant to the already selected compounds until 100 compounds are selected.

We requested an updated quote for these 100 compounds. Only 90 of the 100 selected compounds were still available in stock or in the REAL database. The other 10 required custom chemistry, which is more expensive and has a longer delivery time. We ordered the 90 that did not require custom chemistry. Synthesis failed for 22 of the 90 compounds, so we screened 68 Enamine compounds against PriA-SSB.

All 68 compounds were initially screened in four replicates at eight concentrations ranging from 515.6 nM to 66  $\mu$ M. We defined compounds whose median % inhibition at 33  $\mu$ M was at least 50%, the same threshold used for the AMS screen, as initial hits. We repeated the dose response curve screens for two additional rounds of ten compounds each, expanding the range of concentrations tested to improve the quality of the curve fits (Appendix A.9). We defined confirmed hits as those with a dose response curve in curve class<sup>60</sup> 1.2 or in curve class 2.2 with an IC<sub>50</sub> 95% upper confidence limit within the tested range of concentrations. We used the Collaborative Drug Discovery Vault software<sup>61</sup> to fit dose response curves, calculate IC<sub>50</sub> values and the 95% upper and lower confidence limits, and define curve classes.

## 4.10 Software and Data Availability

Our Python implementation and conda environments with the required Python packages are available on GitHub (<https://github.com/gitter-lab/pria-ams-enamine>) and archived on Zenodo (doi:10.5281/zenodo.5551235). The chemical screening datasets are available at PubChem (AID: 1272365) and Zenodo (doi:10.5281/zenodo.5348290).

## 5 Acknowledgements

This research was supported by National Institutes of Health (NIH) awards R01GM135631 and U54AI117924, a scholarship from King Fahd University of Petroleum & Minerals through the Saudi Arabian Cultural Mission, the University of Wisconsin-Madison Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation, and the John W. and Jeanne M. Rowe Center for Research in Virology at the Morgridge Institute for Research. This research also benefited from GPU hardware from NVIDIA, the computing resources and assistance from the University of Wisconsin-Madison Center for High Throughput Computing, the University of Wisconsin Carbone Cancer Center Small Molecule Screening Facility (supported by NIH award P30CA014520), and credits from the NIH Cloud Credits Model Pilot, a component of the NIH Big Data to Knowledge program. We thank Daniel McNeela for helpful feedback on the manuscript and Cameron Scarlett and Xiaolei Li in the University of Wisconsin School of Pharmacy Analytical Instrumentation-Mass Spectrometry Facility for assistance in validation of compound structure and identity.

## References

- (1) Hoffmann, T.; Gastreich, M. The next level in chemical space navigation: going far beyond enumerable compound libraries. *Drug Discovery Today* **2019**, *24*, 1148 – 1156.
- (2) Irwin, J. J.; Tang, K. G.; Young, J.; Dandarchuluun, C.; Benjamin, W. R.; Khulrelbaatar, M.; Moroz, Y. S.; Mayfield, J.; Sayle, R. A. ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery. *Journal of Chemical Information and Modeling* **2020**, *60*, 6065–6073.
- (3) Enamine. Enamine REAL Database. <https://enamine.net/library-synthesis/real-compounds/real-database>, Accessed: 11 October 2019.

- (4) Krüger, D. M.; Evers, A. Comparison of Structure-and Ligand-Based Virtual Screening Protocols Considering Hit List Complementarity and Enrichment Factors. *ChemMedChem* **2010**, *5*, 148–158.
- (5) Gimeno, A.; Ojeda-Montes, M. J.; Tomás-Hernández, S.; Cereto-Massagué, A.; Beltrán-Debón, R.; Mulero, M.; Pujadas, G.; Garcia-Vallvé, S. The Light and Dark Sides of Virtual Screening: What Is There to Know? *International Journal of Molecular Sciences* **2019**, *20*, 1375.
- (6) Ekins, S.; Puhl, A. C.; Zorn, K. M.; Lane, T. R.; Russo, D. P.; Klein, J. J.; Hickey, A. J.; Clark, A. M. Exploiting machine learning for end-to-end drug discovery and development. *Nature Materials* **2019**, *18*, 435–441.
- (7) Singh, N.; Chaput, L.; Villoutreix, B. O. Virtual screening web servers: designing chemical probes and drug candidates in the cyberspace. *Briefings in Bioinformatics* **2021**, *22*, 1790–1818.
- (8) Cross, J. B.; Thompson, D. C.; Rai, B. K.; Baber, J. C.; Fan, K. Y.; Hu, Y.; Humblet, C. Comparison of Several Molecular Docking Programs: Pose Prediction and Virtual Screening Accuracy. *Journal of Chemical Information and Modeling* **2009**, *49*, 1455–1474.
- (9) Lionta, E.; Spyrou, G.; K Vassilatis, D.; Cournia, Z. Structure-based Virtual Screening for Drug Discovery: Principles, Applications and Recent Advances. *Current Topics in Medicinal Chemistry* **2014**, *14*, 1923–1938.
- (10) Lyu, J.; Wang, S.; Balius, T. E.; Singh, I.; Levit, A.; Moroz, Y. S.; O’Meara, M. J.; Che, T.; Algaa, E.; Tolmachova, K.; Tolmachev, A. A.; Shoichet, B. K.; Roth, B. L.; Irwin, J. J. Ultra-large library docking for discovering new chemotypes. *Nature* **2019**, *566*, 224–229.

- (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.; Fackeldey, K.; Hoffmann, M.; Iavniuk, I.; Wagner, G.; Arthanari, H. An open-source drug discovery platform enables ultra-large virtual screens. *Nature* **2020**, *580*, 663–668.
- (12) Grebner, C.; Malmerberg, E.; Shewmaker, A.; Batista, J.; Nicholls, A.; Sadowski, J. Virtual Screening in the Cloud: How Big Is Big Enough? *Journal of Chemical Information and Modeling* **2020**, *60*, 4274–4282.
- (13) Lavecchia, A. Machine-learning approaches in drug discovery: methods and applications. *Drug Discovery Today* **2015**, *20*, 318–331.
- (14) Carpenter, K. A.; Huang, X. Machine Learning-based Virtual Screening and Its Applications to Alzheimer’s Drug Discovery: A Review. *Current Pharmaceutical Design* **2018**, *24*, 3347–3358.
- (15) Mitchell, J. B. O. Machine Learning Methods in Chemoinformatics. *Wiley Interdisciplinary Reviews. Computational Molecular Science* **2014**, *4*, 468–481.
- (16) Breiman, L. Random Forests. *Machine Learning* **2001**, *45*, 5–32.
- (17) Mayr, A.; Klambauer, G.; Unterthiner, T.; Steijaert, M.; Wegner, J. K.; Ceulemans, H.; Clevert, D.-A.; Hochreiter, S. Large-scale comparison of machine learning methods for drug target prediction on ChEMBL. *Chemical Science* **2018**, *9*, 5441–5451.
- (18) Kutchukian, P. S.; Warren, L.; Magliaro, B. C.; Amoss, A.; Cassaday, J. A.; O’Donnell, G.; Squadroni, B.; Zuck, P.; Pascarella, D.; Culberson, J. C.; Cooke, A. J.; Hurzy, D.; Schlegel, K.-A. S.; Thomson, F.; Johnson, E. N.; Uebele, V. N.; Hermes, J. D.; Parmentier-Batteur, S.; Finley, M. Iterative Focused Screening with Biological Fingerprints Identifies Selective Asc-1 Inhibitors Distinct from Traditional High Throughput Screening. *ACS Chemical Biology* **2017**, *12*, 519–527.

- (19) Stokes, J. M.; Yang, K.; Swanson, K.; Jin, W.; Cubillos-Ruiz, A.; Donghia, N. M.; MacNair, C. R.; French, S.; Carfrae, L. A.; Bloom-Ackerman, Z.; Tran, V. M.; Chiappino-Pepe, A.; Badran, A. H.; Andrews, I. W.; Chory, E. J.; Church, G. M.; Brown, E. D.; Jaakkola, T. S.; Barzilay, R.; Collins, J. J. A Deep Learning Approach to Antibiotic Discovery. *Cell* **2020**, *180*, 688–702.e13.
- (20) McCloskey, K.; Sigel, E. A.; Kearnes, S.; Xue, L.; Tian, X.; Moccia, D.; Gikunju, D.; Bazzaz, S.; Chan, B.; Clark, M. A.; Cuzzo, J. W.; Guie, M.-A.; Guilinger, J. P.; Huguet, C.; Hupp, C. D.; Keefe, A. D.; Mulhern, C. J.; Zhang, Y.; Riley, P. Machine Learning on DNA-Encoded Libraries: A New Paradigm for Hit Finding. *Journal of Medicinal Chemistry* **2020**, *63*, 8857–8866.
- (21) Glaab, E.; Manoharan, G. B.; Abankwa, D. Pharmacophore Model for SARS-CoV-2 3CLpro Small-Molecule Inhibitors and in Vitro Experimental Validation of Computationally Screened Inhibitors. *Journal of Chemical Information and Modeling* **2021**, *61*, 4082–4096.
- (22) Liu, S.; Alnammi, M.; Ericksen, S. S.; Voter, A. F.; Ananiev, G. E.; Keck, J. L.; Hoffmann, F. M.; Wildman, S. A.; Gitter, A. Practical model selection for prospective virtual screening. *Journal of Chemical Information and Modeling* **2019**, *59*, 282–293.
- (23) Lever, J.; Krzywinski, M.; Altman, N. Classification evaluation. *Nature Methods* **2016**, *13*, 603–604.
- (24) Taylor, R. Simulation Analysis of Experimental Design Strategies for Screening Random Compounds as Potential New Drugs and Agrochemicals. *Journal of Chemical Information and Computer Sciences* **1995**, *35*, 59–67.
- (25) Butina, D. Unsupervised Data Base Clustering Based on Daylight’s Fingerprint and Tanimoto Similarity: A Fast and Automated Way To Cluster Small and Large Data Sets. *Journal of Chemical Information and Computer Sciences* **1999**, *39*, 747–750.

- (26) Sturm, N.; Sun, J.; Vandriessche, Y.; Mayr, A.; Klambauer, G.; Carlsson, L.; Engkvist, O.; Chen, H. Application of Bioactivity Profile-Based Fingerprints for Building Machine Learning Models. *Journal of Chemical Information and Modeling* **2019**, *59*, 962–972.
- (27) Bender, A.; Cortés-Ciriano, I. Artificial intelligence in drug discovery: what is realistic, what are illusions? Part 1: Ways to make an impact, and why we are not there yet. *Drug Discovery Today* **2021**, *26*, 511–524.
- (28) Irwin, J. J.; Gaskins, G.; Sterling, T.; Mysinger, M. M.; Keiser, M. J. Predicted Biological Activity of Purchasable Chemical Space. *Journal of Chemical Information and Modeling* **2018**, *58*, 148–164.
- (29) Koerstz, M.; Christensen, A. S.; Mikkelsen, K. V.; Nielsen, M. B.; Jensen, J. H. High Throughput Virtual Screening of 230 Billion Molecular Solar Heat Battery Candidates. *ChemRxiv* **2020**,
- (30) Glaser, J.; Vermaas, J. V.; Rogers, D. M.; Larkin, J.; LeGrand, S.; Boehm, S.; Baker, M. B.; Scheinberg, A.; Tillack, A. F.; Thavappiragasam, M.; Sedova, A.; Hernandez, O. High-throughput virtual laboratory for drug discovery using massive datasets. *The International Journal of High Performance Computing Applications* **2021**, *35*, 452–468.
- (31) Adeshina, Y. O.; Deeds, E. J.; Karanickolas, J. Machine learning classification can reduce false positives in structure-based virtual screening. *Proceedings of the National Academy of Sciences* **2020**, *117*, 18477–18488.
- (32) 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.; Stauch, B.; Johansson, L. C.; Cherezov, V.; Kenakin, T.; Irwin, J. J.; Shoichet, B. K.; Roth, B. L.; Dubocovich, M. L.



- Virtual discovery of melatonin receptor ligands to modulate circadian rhythms. *Nature* **2020**, *579*, 609–614.
- (33) Hughes, T. E.; Del Rosario, J. S.; Kapoor, A.; Yazici, A. T.; Yudin, Y.; Fluck, E. C., III; Filizola, M.; Rohacs, T.; Moiseenkova-Bell, V. Y. Structure-based characterization of novel TRPV5 inhibitors. *eLife* **2019**, *8*, e49572.
- (34) 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.; Cherezov, V.; Sarret, P.; Katritch, V. Structure-Based Virtual Screening of Ultra-Large Library Yields Potent Antagonists for a Lipid GPCR. *Biomolecules* **2020**, *10*, 1634.
- (35) Schorpp, K.; Rothenaigner, I.; Salmina, E.; Reinshagen, J.; Low, T.; Brenke, J. K.; Gopalakrishnan, J.; Tetko, I. V.; Gul, S.; Hadian, K. Identification of Small-Molecule Frequent Hitters from AlphaScreen High-Throughput Screens. *Journal of Biomolecular Screening* **2014**, *19*, 715–726.
- (36) Walters, W. P. Virtual Chemical Libraries. *Journal of Medicinal Chemistry* **2019**, *62*, 1116–1124.
- (37) Jain, A. N. Scoring noncovalent protein-ligand interactions: A continuous differentiable function tuned to compute binding affinities. *Journal of Computer-Aided Molecular Design* **1996**, *10*, 427–440.
- (38) Kearnes, S. Pursuing a Prospective Perspective. *Trends in Chemistry* **2021**, *3*, 77–79.
- (39) Voter, A. F.; Killoran, M. P.; Ananiev, G. E.; Wildman, S. A.; Hoffmann, F. M.; Keck, J. L. A High-Throughput Screening Strategy to Identify Inhibitors of SSB Protein–Protein Interactions in an Academic Screening Facility. *SLAS DISCOVERY: Advancing Life Sciences R&D* **2017**, 94–101.

- (40) Nordmann, P.; Cuzon, G.; Naas, T. The Real Threat of *Klebsiella Pneumoniae* Carbapenemase-producing Bacteria. *The Lancet Infectious Diseases* **2009**, *9*, 228–236.
- (41) Zhang, J.-H.; Chung, T. D. Y.; Oldenburg, K. R. A Simple Statistical Parameter for Use in Evaluation and Validation of High Throughput Screening Assays. *Journal of Biomolecular Screening* **1999**, *4*, 67–73.
- (42) Baell, J. B.; Holloway, G. A. New Substructure Filters for Removal of Pan Assay Interference Compounds (PAINS) from Screening Libraries and for Their Exclusion in Bioassays. *Journal of Medicinal Chemistry* **2010**, *53*, 2719–2740.
- (43) Lagorce, D.; Sperandio, O.; Baell, J. B.; Miteva, M. A.; Villoutreix, B. O. FAF-Drugs3: a Web Server for Compound Property Calculation and Chemical Library Design. *Nucleic Acids Research* **2015**, *43*, W200–W207.
- (44) Landrum, G. RDKit: Open-Source Cheminformatics Software. **2018**,
- (45) Rogers, D.; Hahn, M. Extended-Connectivity Fingerprints. *Journal of Chemical Information and Modeling* **2010**, *50*, 742–754.
- (46) Breiman, L. Bagging predictors. *Machine Learning* **1996**, *24*, 123–140.
- (47) Amit, Y.; Geman, D. Shape Quantization and Recognition with Randomized Trees. *Neural Computation* **1997**, *9*, 1545–1588.
- (48) Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; Duchesnay, E. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* **2011**, *12*, 2825–2830.
- (49) Schapire, R. E. The strength of weak learnability. *Machine Learning* **1990**, *5*, 197–227.
- (50) Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* **2001**, *29*, 1189 – 1232.

- (51) Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA, 2016; pp 785–794.
- (52) Chollet, F. Keras. <https://github.com/fchollet/keras> Accessed 2016-12-20.
- (53) The Theano Development Team,; Al-Rfou, R.; Alain, G.; Almahairi, A.; Angermueller, C.; Bahdanau, D.; Ballas, N.; Bastien, F.; Bayer, J.; Belikov, A.; Belopolsky, A.; Bengio, Y.; Bergeron, A.; Bergstra, J.; Bisson, V.; Snyder, J. B.; Bouchard, N.; Boulanger-Lewandowski, N.; Bouthillier, X.; de Brébisson, A.; Breuleux, O.; Carrier, P.-L.; Cho, K.; Chorowski, J.; Christiano, P.; Cooijmans, T.; Côté, M.-A.; Côté, M.; Courville, A.; Dauphin, Y. N.; Delalleau, O.; Demouth, J.; Desjardins, G.; Dieleman, S.; Dinh, L.; Ducoffe, M.; Dumoulin, V.; Kahou, S. E.; Erhan, D.; Fan, Z.; Firat, O.; Germain, M.; Glorot, X.; Goodfellow, I.; Graham, M.; Gulcehre, C.; Hamel, P.; Harlouchet, I.; Heng, J.-P.; Hidasi, B.; Honari, S.; Jain, A.; Jean, S.; Jia, K.; Korobov, M.; Kulkarni, V.; Lamb, A.; Lamblin, P.; Larsen, E.; Laurent, C.; Lee, S.; Lefrancois, S.; Lemieux, S.; Léonard, N.; Lin, Z.; Livezey, J. A.; Lorenz, C.; Lowin, J.; Ma, Q.; Manzagol, P.-A.; Mastropietro, O.; McGibbon, R. T.; Memisevic, R.; van Merriënboer, B.; Michalski, V.; Mirza, M.; Orlandi, A.; Pal, C.; Pascanu, R.; Pezeshki, M.; Raffel, C.; Renshaw, D.; Rocklin, M.; Romero, A.; Roth, M.; Sadowski, P.; Salvatier, J.; Savard, F.; Schlüter, J.; Schulman, J.; Schwartz, G.; Serban, I. V.; Serdyuk, D.; Shabanian, S.; Simon, É.; Spieckermann, S.; Subramanyam, S. R.; Sygnowski, J.; Tanguay, J.; van Tulder, G.; Turian, J.; Urban, S.; Vincent, P.; Visin, F.; de Vries, H.; Warde-Farley, D.; Webb, D. J.; Willson, M.; Xu, K.; Xue, L.; Yao, L.; Zhang, S.; Zhang, Y. Theano: A Python Framework for Fast Computation of Mathematical Expressions. *arXiv:1605.02688* **2016**,
- (54) Kingma, D. P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980* **2014**,

- (55) Wolpert, D. H. Stacked generalization. *Neural networks* **1992**, *5*, 241–259.
- (56) Willett, P. Similarity-based virtual screening using 2D fingerprints. *Drug Discovery Today* **2006**, *11*, 1046 – 1053.
- (57) Geppert, H.; Vogt, M.; Bajorath, J. Current Trends in Ligand-Based Virtual Screening: Molecular Representations, Data Mining Methods, New Application Areas, and Performance Evaluation. *Journal of Chemical Information and Modeling* **2010**, *50*, 205–216.
- (58) Thain, D.; Tannenbaum, T.; Livny, M. Distributed computing in practice: the Condor experience. **2005**, *17*, 323–356.
- (59) Ramsundar, B.; Eastman, P.; Walters, P.; Pande, V.; Leswing, K.; Wu, Z. *Deep Learning for the Life Sciences: Applying Deep Learning to Genomics, Microscopy, Drug Discovery, and More*; O'Reilly Media, 2019.
- (60) Inglese, J.; Auld, D. S.; Jadhav, A.; Johnson, R. L.; Simeonov, A.; Yasgar, A.; Zheng, W.; Austin, C. P. Quantitative high-throughput screening: A titration-based approach that efficiently identifies biological activities in large chemical libraries. *Proceedings of the National Academy of Sciences* **2006**, *103*, 11473–11478.
- (61) Ekins, S.; Bunin, B. A. In *In Silico Models for Drug Discovery*; Kortagere, S., Ed.; Methods in Molecular Biology; Humana Press: Totowa, NJ, 2013; pp 139–154.
- (62) R Core Team, R: A Language and Environment for Statistical Computing. 2013; <http://www.R-project.org/>, ISBN 3-900051-07-0.

## A Supplementary Methods

### A.1 Training Data Preprocessing Overview

Here we describe in detail how we preprocessed the LC and MLPCN screening data in Table 5 to create the 427,300 compound training dataset. At the time this dataset was assembled, the LC4 retest dataset was not available so hits were defined using only the LC4 primary screen. The first preprocessing step involved merging the three primary and two retest screens.

1. **Primary** screen for 74,896 LC123 compounds.
2. **Retest** screen for 2,698 LC123 compounds.
3. **Primary** screen for 25,290 LC4 compounds.
4. **Primary** screen for 337,990 MLPCN compounds.
5. **Retest** screen for 1,400 MLPCN compounds.

Merging these datasets yields a total of 442,274 entries; a compound can have multiple readings. This dataset is then processed to determine the binary activity of each compound. Specifically, for each compound, the scores are grouped, and three criteria are assessed: the median % inhibition of primary screens is  $\geq 35\%$ , the median % inhibition of retest screens is  $\geq 35\%$ , and the compound does not match a PAINS filter.<sup>42,43</sup> A compound is deemed active if it meets all three conditions. Finally, the training dataset is constructed such that it contains one row per compound that includes its binary activity label and its median inhibition score based on all recorded primary inhibition scores.

### A.2 Merged Dataset Structure

Each of the original screening data files have eight columns: **SMSSF ID**, **CDD SMILES**, **Batch Name**, **Library ID**, **Plate Name**, **Plate Well**, **Run Date**, **PriA-SSB AS %**

**inhibition.** We created a combined dataset with the following columns:

1. **Molecule ID:** uniquely identifies a molecule. A non-negative integer.
2. **Duplicate ID:** denotes the duplicate number (replicate). This is done so that the same molecule can have multiple % inhibition readings; allows grouping by ‘Molecule ID’.
3. **SMSSF ID:** molecule ID used by the screening facility.
4. **Batch Name:** used at the screening facility for screening batch identification. Not used in this project.
5. **Library ID:** the libraries that contain the molecule: LC1, LC2, LC3, LC4, and MLPCN. Note that a molecule can be in multiple libraries.
6. **Plate Name:** specifies the plate identifier for this molecule. Can help identify retests if **Plate Name** contains the text **CP**.
7. **Plate Well:** specifies the well row-column location of the molecule on the plate.
8. **Run Date:** specifies the date the data was entered into Collaborative Drug Discovery (CDD) software.
9. **CDD SMILES:** the CDD software SMILES string of the molecule.
10. **rdkit SMILES:** rdkit canonical smiles of the molecule. This is used for uniqueness identification of molecules.
11. **MorganFP:** rdkit Morgan fingerprint of the molecule.
12. **PriA-SSB AS % inhibition:** the % inhibition score of the molecule.
13. **PriA-SSB AS Activity:** the binary activity  $\{0, 1\}$  of the molecule.

14. **Primary Filter**: is active (denoted as 1) if the median of the primary screens for this molecule is greater than or equal to **binary\_threshold**.
15. **Retest Filter**: is active (denoted as 1) if the median of the retest screens for this molecule is greater than or equal to **binary\_threshold**.
16. **PAINS Filter**: is active (denoted as 1) if the molecule is not reported as a PAINS compound by the rdkit PAINS filter.

### A.3 Ensuring Uniqueness

There are two types of repeated molecules for which we retain all measurements. The first is molecules that were experimentally tested more than once, which we generally refer to as replicates. Molecules can be tested more than once because they belong to multiple libraries (e.g., LC1 and MLPCN) or they were included in a primary and a retest screen. The other type of repeated molecules has different **SMSSF IDs** but the same **rdkit SMILES** due to salts. These molecules with the same **SMSSF ID** or the same **rdkit SMILES** are grouped together. The grouping operation assigns them the same **Molecule ID** and different **Duplicate IDs**. This allows us to compute aggregate % inhibition scores by grouping on **Molecule ID**.

In addition to grouping replicate measurements, we also ensure that a single experimental measurement is not erroneously recorded twice. Entries in the combined dataset should not match in the following columns: **SMSSF ID**, **Plate Name**, **Plate Well**, **Run Date**, **PriA-SSB AS % inhibition**. If two entries match on these columns, the same data has been copied so one entry is removed.

### A.4 Preprocessing Steps

The preprocessing steps for the combined dataset are as follows:

1. Read in the merged five individual measurement files.

2. Remove 140 outliers with % inhibition  $\leq -100.0$  based on inspection of the % inhibition distribution.
3. Remove rows with missing values. Some molecules had **SMSSF ID** present, but other entries like **CDD SMILES**, **Plate Name**, etc. missing.
4. Define unique identifiers for each row to [**SMSSF ID**, **Plate Name**, **Plate Well**, **Run Date**, **PriA-SSB AS % inhibition**]. Assert that there are no duplicates on the uniqueness columns.
5. Add **rdkit SMILES** and fingerprints. Remove salts using rdkit’s **SaltRemover** and **Salts.txt** (<https://github.com/rdkit/rdkit/blob/master/Data/Salts.txt>).
6. Add **Molecule ID** and **Duplicate ID** placeholders. Group molecules that have the same **SMSSF ID** or **rdkit SMILES** giving them the same **Molecule ID** and increasing **Duplicate ID**.
7. Generate binary labels **PriA-SSB AS Activity** according to binarization rules below.
8. Save the combined dataset containing 441,900 molecules.

## A.5 Binary Activity Rules

Molecules can have up to four % inhibition score measurements (same **Molecule ID** but different **Duplicate ID**). We defined the following rules for creating binary activity labels for training classifiers:

1. The median % inhibition value over all **primary** screens of the molecule is  $\geq 35\%$ .
2. If one or more retest screens were conducted, the median % inhibition value over all **retest** screens of the molecule is  $\geq 35\%$ .
3. The molecule does not match a PAINS filter per rdkit’s **FilterCatalog**.



These rules are applied to all molecules individually by grouping using the **Molecule ID**. The binary activity label is assigned to all instances of the molecule identified by **Molecule ID**.

## A.6 Generating the Training Dataset

The combined dataset can have many % inhibition readings for a single molecule. The training dataset will contain a single entry for each unique molecule identified by its **Molecule ID**. It is generated as follows:

1. Read in the combined dataset.
2. Remove retests, leaving the primary screens. Recall that the binary activity is still recorded in the primary entries.
3. Standardize **Library IDs** to support stratifying by library.
4. Group by **Molecule ID** and compute the median % inhibition for the primary screens in the **PriA-SSB AS % inhibition (Primary Median)** column.
5. Save this as the training dataset.

Each entry in the training dataset has a unique **Molecule ID** and is thus regarded as a unique molecule. The method to generate folds for cross validation starts by grouping molecules based on the **Library ID** column. The next step performs stratified sampling on each **Library ID** value, grouping it into 10 folds based on the binary activity labels. This ensures that each library is well represented in each fold and the activity ratios are similar. For example, if LC1 has 1,000 total molecules with 10 actives, then each of the 10 folds will have 1 LC1 active and 99 LC1 inactives.

## A.7 Hyperparameters

Here we enumerate the hyperparameters for the STNN-C, STNN-R, RF-C, XGB-C, and XGB-R models. We performed grid searches over the hyperparameter values in the tables below. Each combination of hyperparameters in the grid search was assigned a numeric ID. The mapping from numeric IDs to hyperparameter combinations is available in our GitHub repository.

For each model, we report the top-performing hyperparameter IDs sorted by performance. Some training jobs took over a week to complete. We recorded results for all the jobs that terminated within a week and randomly picked additional long jobs to run to termination. The remaining long jobs are reported below as missing hyperparameter IDs because they did not generate results.

Table S13 shows the 216 hyperparameters for Random Forest-Classification. Top 20 (with ties) hyperparameter IDs: 139, 69, 111, 210, 148, 212, 28, 61, 124, 130, 131, 141, 14, 38, 165, 65, 123, 94, 3, 88, 72. Missing hyperparameter IDs: 5, 6, 19, 23, 24, 26, 27, 29, 37, 47, 52, 64, 71, 75, 91, 101, 113, 114, 126, 133, 136, 137, 145, 153, 156, 158, 160, 167, 168, 179, 187, 189, 191, 193, 201, 208.

Table S14 shows the XGBoost-Classification hyperparameters. There are 1,866,240 hyperparameter combinations from which we randomly select 1,000. Top 20 (with ties) hyperparameter IDs : 140, 967, 807, 263, 694, 440, 47, 116, 792, 663, 32, 564, 950, 84, 364, 605, 431, 55, 388, 960, 735. No missing hyperparameter IDs.

Table S15 shows the XGBoost-Regression hyperparameters. There are 829,440 hyperparameter combinations from which we randomly select 1,000. Top 20 (with ties) hyperparameter IDs are: 187, 880, 514, 605, 754, 6, 321, 753, 294, 718, 280, 214, 545, 507, 81, 65, 440, 409, 586, 721, 911. Missing hyperparameter IDs are: 56, 120, 434, 589, 708, 874.

Table S16 shows the 432 hyperparameters for Single-task Neural Network-Classification. Top 20 hyperparameter IDs: 356, 88, 180, 5, 416, 183, 47, 124, 93, 385, 54, 423, 363, 370, 215, 415, 29, 132, 254, 404. Missing hyperparameter IDs: 7, 35, 48, 69, 73, 79, 91, 92, 95,

97, 113, 120, 122, 126, 137, 138, 142, 143, 166, 225, 231, 234, 237, 239, 251, 261, 275, 281, 284, 294, 297, 301, 303, 314, 317, 320, 336, 338, 340, 341, 342, 346, 368, 369, 372, 376, 378, 380, 382, 384, 389, 391, 395, 398, 399, 403, 420, 429, 431.

Table S17 shows the 432 hyperparameters for Single-task Neural Network-Regression. Top 20 hyperparameter IDs: 47, 199, 175, 27, 178, 20, 114, 106, 157, 270, 123, 191, 323, 369, 265, 17, 45, 115, 108, 67. Missing hyperparameter IDs: 7, 15, 26, 69, 92, 93, 137, 138, 166, 231, 234, 237, 281, 297, 303, 336, 338, 341, 346, 380, 398, 403.

## A.8 Ensemble Model Training

The Model-Based Ensemble and Max Vote Ensemble models are introduced in the model selection stage. Both ensemble models make use of  $N$  base model predictions as input and produce a single activity probability prediction as output. The Max Vote Ensemble uses a heuristic strategy that takes the maximum of the base model predictions. On the other hand, the Model-Based Ensemble trains an XGB model with the base model predictions as input.

The steps for training the ensemble models (depicted in Figure S3) are as follows:

- **Step 1:** The base models are trained on folds 0-8 with fold 8 being used as the validation fold. The RF and Similarity Baseline models do not require a validation set, so they do not make use of fold 8.
- **Step 2:** The trained base models from step 1 are then used to generate predictions for folds 0-7 and fold 9.
- **Step 3:** All  $N$  base model predictions from step 2 for folds 0-7 are concatenated (stacked) to construct ensemble features. The Model-Based Ensemble is then trained with these ensemble features as input and the true labels as output for folds 0-7. For the Max Vote Ensemble, no training is needed as the output is just the maximum of the ensemble features.

- **Step 4:** All  $N$  base model predictions from step 2 for fold 9 are concatenated. The trained ensemble from step 3 is then used to generate the final fold 9 prediction with the concatenated features as input.

There are several ways in which our ensembling approach could be improved. In a typical stacking ensemble,<sup>55</sup> the ensemble is trained on predictions that the base models did not train on. If the training set is denoted as  $A$ , then one common stacking method is to do  $K$ -fold cross validation with the base models and then use the stacked  $K$  held-out fold predictions to train the ensemble. Another strategy is to split  $A$  into two subsets, one for training the base models, and one for generating base model predictions to train the ensemble. However, we train the base models on a dataset  $A$ , generate predictions on  $A$ , and train the ensemble on these predictions. This can introduce a bias in a parameter-based ensemble like Model-Based Ensemble towards favoring base models that have a better fit on  $A$ . The Max Vote Ensemble does not have this potential bias because it is parameter free.

In addition, we could have used a more flexible and thorough strategy to select the base model mixtures and excluded ensemble configurations with a single base model (Table S18). In retrospect, we observed that including regression models in the ensembles hurt performance (Figures S1 and S2). Furthermore, the regression base model outputs were not scaled, which could be detrimental to the Max Vote Ensemble when combining classification and regression models. The Model-Based Ensemble is less affected by the scale of the regression model outputs because its XGB model can account for different score distributions provided by each base model.

## A.9 Enamine REAL Dose Response Curves

We generated three sets of dose response curves: eight-point curves with four replicates for all 68 compounds, 16-point curves with eight replicates for 10 compounds, and another set of 16-point curves with six replicates for 10 compounds. The initial eight-point dose response curves started with an initial concentration of 66  $\mu\text{M}$  and progressively divided the

concentration in half until reaching 515.6 nM. Both 16-point dose response curves started with an initial concentration of 100  $\mu$ M, then 66.7  $\mu$ M, and then progressively divided the concentration in half until reaching 4.2 nM. All dose response screens followed the same protocol used for the MLPCN library.

We generated the 16-point dose response curves because we were unable to reliably establish IC<sub>50</sub> values for some compounds in the eight-point screen. The minimum concentrations tested were not low enough. When reporting IC<sub>50</sub> values and bounds in Table 4 we used the screening data from the highest-quality dose response curve. The second set of 16-point curve were the highest quality because they had 16 concentrations and were unaffected by any spatial plate effects. The first set of 16-point curve were the next highest quality because they also had 16 concentrations and the most replicates. However, some of the readings at the lowest concentration in these curves were skewed due to assay spatial effects at the plate edge. The eight-point curves were used only if the compound was not tested in a 16-point curve. Some of the replicates at the lowest concentration in the eight-point curves were also affected by the spatial effects.

The Collaborative Drug Discovery Vault software<sup>61</sup> modifies the original dose response classification criteria.<sup>60</sup> It defines curve class 1 as a “complete curve, showing multiple points near both asymptotes” and curve class 2 as a “partial curve, showing multiple points near only one asymptote.” The 1.2 and 2.2 subclasses indicate that the curve fit has  $R^2 > 0.9$  and maximum activity  $\leq 80\%$ .

## B Supplementary Results

### B.1 Base Model versus Ensemble Model Actives

In the model selection stage, fold 9 was used for performance comparison among models. The motivation for using ensembles was to pool the predictions of various models and thus promote diversity in the top predicted compounds. We compared the prediction overlap

between the ensembles and the best base models in the model selection stage (Table 1). Specifically, for each top base model, we compared the top 1% predicted compounds from fold 9 against the top 1% predicted by the ensembles (Tables S2, S3, S4, S5, and S6). This examines the similarity and difference in the actives retrieved by the top base models and the ensembles.

Note that the predictions for the Model-Based Ensemble were regenerated using a newer version of Keras. The performance of the regenerated ensemble and the ensemble referred to in Table 1 differs by no more than  $10^{-3}$ .

## B.2 AMS Activity Level versus Cluster Membership

The 1,024 AMS prospective compounds were delineated into three activity level categories based on their % inhibition values in the two replicates:

- **Weak Actives:** Compounds with % inhibition in [35.0, 50.0) in both replicates. The value 35.0 is close to the 40th percentile in replicate 1.
- **Normal Actives:** Compounds with % inhibition between [50.0, 72.0) in both replicates. The value 50.0 is the threshold used to define hits.
- **Strong Actives:** Compounds with % inhibition  $\geq 72.0$  in both replicates. The value 72.0 is close to the 75th percentile in replicate 1.

Table S8 summarizes the AMS prospective compound counts by activity level. Note that the sum of normal and strong actives is equal to the total hits. A Jupyter notebook in our GitHub repository shows the cluster membership versus activity level counts. Fisher’s exact test was performed using the stats package in R<sup>62</sup> with the following arguments: `workspace=2e8`, `hybrid=FALSE`, `alternative="two.sided"`, `conf.level=0.95`, `simulate.p.value=TRUE`.

### B.3 Scaling to a Billion Compounds with Enamine REAL

Using the top RF-C model from the model selection stage trained on the training set, we scored the entirety of the 1,077,562,987 Enamine REAL compounds. A total of 18 jobs, each processing about 60.3 million compounds, were processed on generic compute nodes with a single CPU, 6GB RAM, and 30GB disk space. Table S11 summarizes the processing time in hours and number of compounds per node. The mean time was 53.21 hours with a 6.40 hour standard deviation. Because each compound is scored independently, we could easily reduce the total wall time needed to score over a billion compounds by distributing the compounds across hundreds or thousands of compute nodes.

## C Supplementary Figures

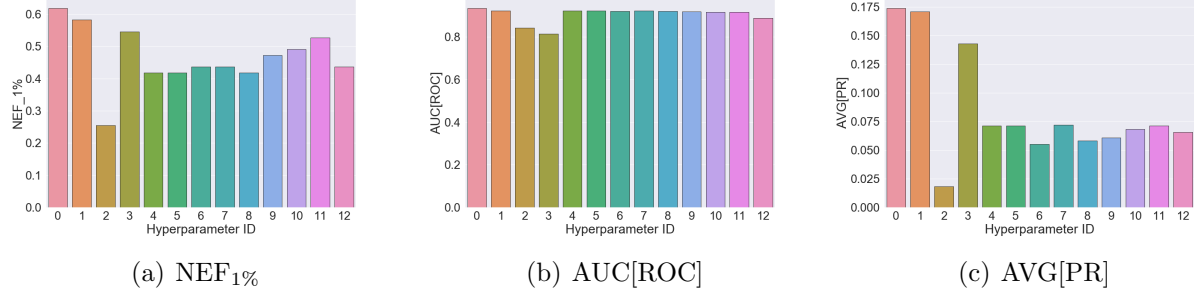


Figure S1: Model-Based Ensemble results on fold 9. Table S18 lists the base models associated with each ensemble hyperparameter ID.

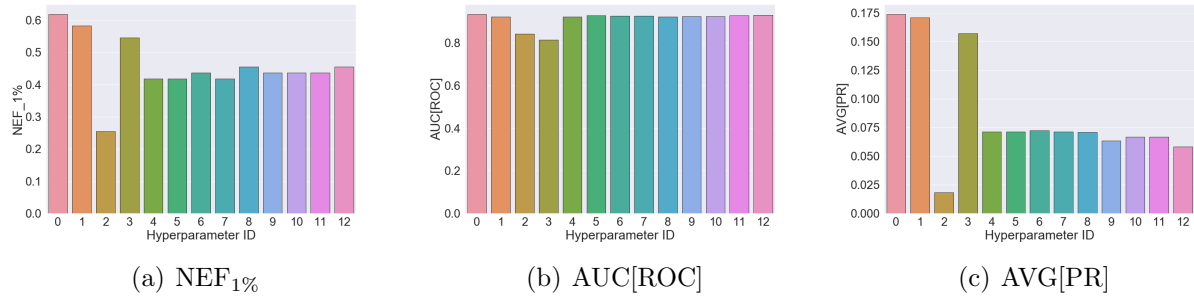


Figure S2: Max Vote Ensemble results on fold 9. Table S18 lists the base models associated with each ensemble hyperparameter ID.



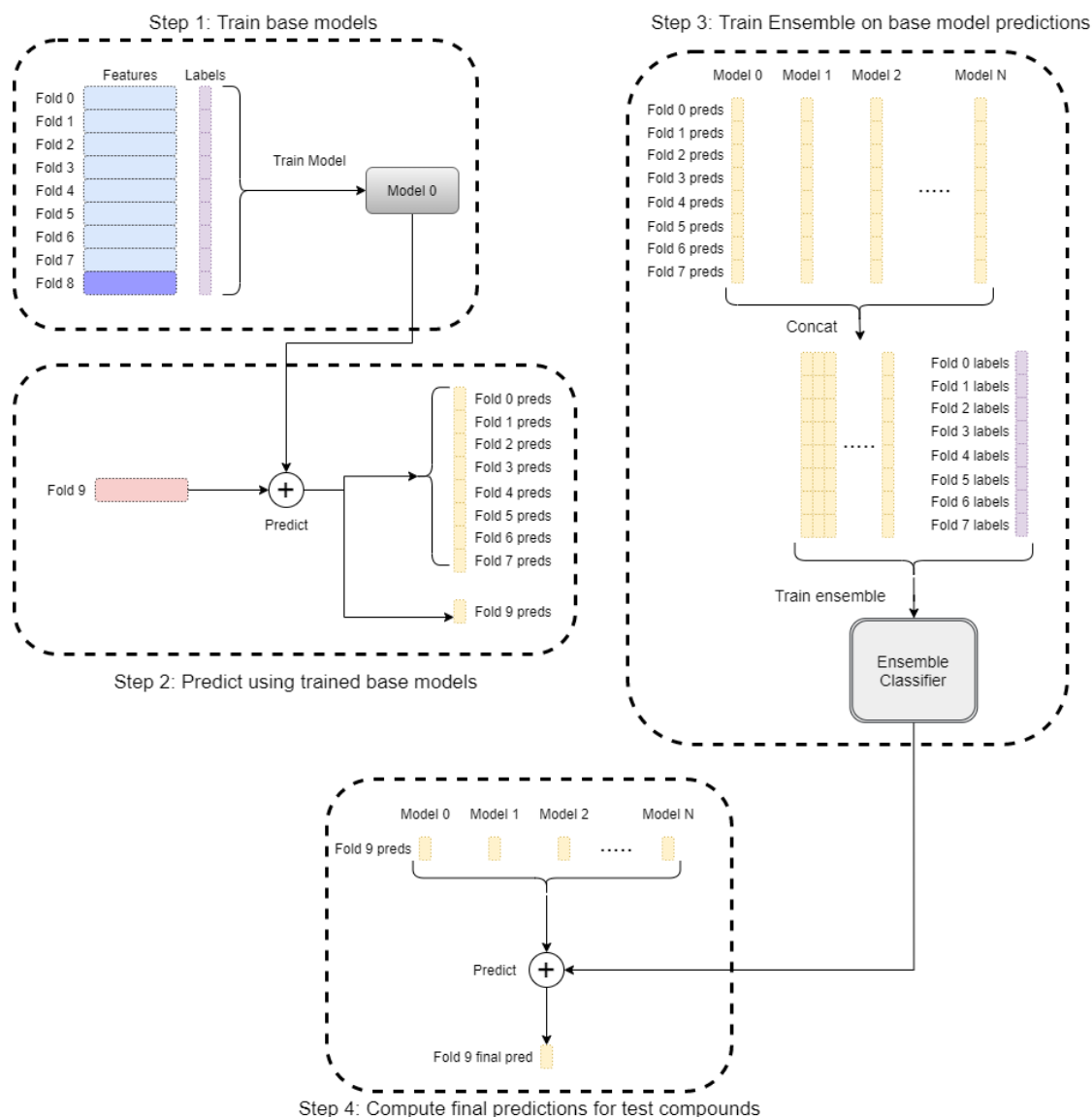
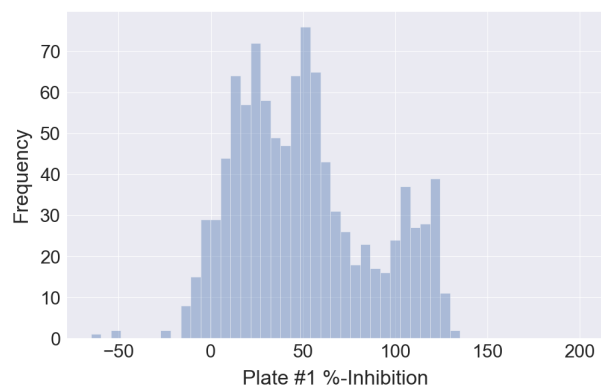
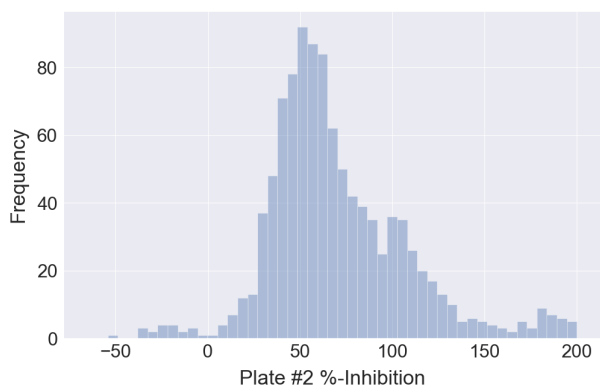


Figure S3: Illustration of ensemble training steps in the context of the model selection stage. Step 1 trains base models on folds 0-8 with fold 8 being used as the validation fold. Step 2 generates predictions on folds 0-7 and fold 9. Step 3 trains the ensemble classifier on base model predictions for folds 0-7. Step 4 uses the trained ensemble to generate final predictions on fold 9.



(a) Replicate 1



(b) Replicate 2

Figure S4: % inhibition distributions of the 1,024 AMS compounds for both replicates.

## D Supplementary Tables

Table S1: Cross validation (CV) training, validation, and test fold splits for each run ID. A total of four runs were done for each hyperparameter setting in the hyperparameter tuning stage. Note that Ensemble models follow a different splitting scheme. The RF-C and Similarity Baseline models do not use the validation folds. They are merged into the training folds.

CV run ID	Validation fold	Test fold	Training folds
0	0	1	2, 3, 4, 5, 6, 7
1	2	3	0, 1, 4, 5, 6, 7
2	4	5	0, 1, 2, 3, 6, 7
3	6	7	0, 1, 2, 3, 4, 5

Table S2: Top RF-C model versus the ensembles. Set difference of actives when filtering for the top 427 predictions on fold 9 in the model selection stage.

Model	Ensemble	Ensemble \ Model	Model \ Ensemble
RF-C	Model-Based #0	0	0
RF-C	Model-Based #1	1	3
RF-C	Model-Based #2	2	22
RF-C	Model-Based #3	0	4
RF-C	Model-Based #4	2	13
RF-C	Model-Based #5	2	13
RF-C	Model-Based #6	3	13
RF-C	Model-Based #7	2	12
RF-C	Model-Based #8	2	13
RF-C	Model-Based #9	3	11
RF-C	Model-Based #10	2	8
RF-C	Model-Based #11	2	7
RF-C	Model-Based #12	1	11
RF-C	Max Vote #0	0	0
RF-C	Max Vote #1	1	3
RF-C	Max Vote #2	2	22
RF-C	Max Vote #3	0	4
RF-C	Max Vote #4	2	13
RF-C	Max Vote #5	2	13
RF-C	Max Vote #6	2	12
RF-C	Max Vote #7	2	13
RF-C	Max Vote #8	2	11
RF-C	Max Vote #9	2	12
RF-C	Max Vote #10	2	12
RF-C	Max Vote #11	2	12
RF-C	Max Vote #12	1	10

Table S3: Top XGB-C model versus the ensembles. Set difference of actives when filtering for the top 427 predictions on fold 9 in the model selection stage.

Model	Ensemble	Ensemble \ Model	Model \ Ensemble
XGB-C	Model-Based #0	3	1
XGB-C	Model-Based #1	0	0
XGB-C	Model-Based #2	2	20
XGB-C	Model-Based #3	2	4
XGB-C	Model-Based #4	2	11
XGB-C	Model-Based #5	2	11
XGB-C	Model-Based #6	4	12
XGB-C	Model-Based #7	3	11
XGB-C	Model-Based #8	3	12
XGB-C	Model-Based #9	4	10
XGB-C	Model-Based #10	4	8
XGB-C	Model-Based #11	4	7
XGB-C	Model-Based #12	3	11
XGB-C	Max Vote #0	3	1
XGB-C	Max Vote #1	0	0
XGB-C	Max Vote #2	2	20
XGB-C	Max Vote #3	2	4
XGB-C	Max Vote #4	2	11
XGB-C	Max Vote #5	2	11
XGB-C	Max Vote #6	3	11
XGB-C	Max Vote #7	2	11
XGB-C	Max Vote #8	4	11
XGB-C	Max Vote #9	3	11
XGB-C	Max Vote #10	3	11
XGB-C	Max Vote #11	3	11
XGB-C	Max Vote #12	2	9

Table S4: Top XGB-R model versus the ensembles. Set difference of actives when filtering for the top 427 predictions on fold 9 in the model selection stage.

Model	Ensemble	Ensemble \ Model	Model \ Ensemble
XGB-R	Model-Based #0	22	2
XGB-R	Model-Based #1	20	2
XGB-R	Model-Based #2	0	0
XGB-R	Model-Based #3	18	2
XGB-R	Model-Based #4	14	5
XGB-R	Model-Based #5	14	5
XGB-R	Model-Based #6	13	3
XGB-R	Model-Based #7	15	5
XGB-R	Model-Based #8	12	3
XGB-R	Model-Based #9	15	3
XGB-R	Model-Based #10	18	4
XGB-R	Model-Based #11	19	4
XGB-R	Model-Based #12	14	4
XGB-R	Max Vote #0	22	2
XGB-R	Max Vote #1	20	2
XGB-R	Max Vote #2	0	0
XGB-R	Max Vote #3	18	2
XGB-R	Max Vote #4	14	5
XGB-R	Max Vote #5	14	5
XGB-R	Max Vote #6	15	5
XGB-R	Max Vote #7	14	5
XGB-R	Max Vote #8	16	5
XGB-R	Max Vote #9	15	5
XGB-R	Max Vote #10	15	5
XGB-R	Max Vote #11	15	5
XGB-R	Max Vote #12	15	4

Table S5: Top NN-C model versus the ensembles. Set difference of actives when filtering for the top 427 predictions on fold 9 in the model selection stage.

Model	Ensemble	Ensemble \ Model	Model \ Ensemble
NN-C	Model-Based #0	4	0
NN-C	Model-Based #1	4	2
NN-C	Model-Based #2	2	18
NN-C	Model-Based #3	0	0
NN-C	Model-Based #4	2	9
NN-C	Model-Based #5	2	9
NN-C	Model-Based #6	3	9
NN-C	Model-Based #7	2	8
NN-C	Model-Based #8	2	9
NN-C	Model-Based #9	3	7
NN-C	Model-Based #10	2	4
NN-C	Model-Based #11	2	3
NN-C	Model-Based #12	1	7
NN-C	Max Vote #0	4	0
NN-C	Max Vote #1	4	2
NN-C	Max Vote #2	2	18
NN-C	Max Vote #3	0	0
NN-C	Max Vote #4	2	9
NN-C	Max Vote #5	2	9
NN-C	Max Vote #6	2	8
NN-C	Max Vote #7	2	9
NN-C	Max Vote #8	2	7
NN-C	Max Vote #9	2	8
NN-C	Max Vote #10	2	8
NN-C	Max Vote #11	2	8
NN-C	Max Vote #12	1	6

Table S6: Top NN-R model versus the ensembles. Set difference of actives when filtering for the top 427 predictions on fold 9 in the model selection stage.

Model	Ensemble	Ensemble \ Model	Model \ Ensemble
NN-R	Model-Based #0	13	2
NN-R	Model-Based #1	11	2
NN-R	Model-Based #2	5	14
NN-R	Model-Based #3	9	2
NN-R	Model-Based #4	0	0
NN-R	Model-Based #5	0	0
NN-R	Model-Based #6	3	2
NN-R	Model-Based #7	1	0
NN-R	Model-Based #8	3	3
NN-R	Model-Based #9	4	1
NN-R	Model-Based #10	5	0
NN-R	Model-Based #11	6	0
NN-R	Model-Based #12	4	3
NN-R	Max Vote #0	13	2
NN-R	Max Vote #1	11	2
NN-R	Max Vote #2	5	14
NN-R	Max Vote #3	9	2
NN-R	Max Vote #4	0	0
NN-R	Max Vote #5	0	0
NN-R	Max Vote #6	1	0
NN-R	Max Vote #7	0	0
NN-R	Max Vote #8	2	0
NN-R	Max Vote #9	1	0
NN-R	Max Vote #10	2	1
NN-R	Max Vote #11	2	1
NN-R	Max Vote #12	3	1

Table S7: Summary of cluster hit metrics for the 1,024 AMS compounds. Unique cluster (U. Cl.) hits denotes the number of clusters containing at least one hit. Novel cluster (Nov. Cl.) hits denotes the number of clusters with AMS hits but no training set actives. Taylor-Butina (TB) clustering with the specified distance thresholds is used to define clusters.

Selector	U. Cl. Hits (TB 0.2)	Nov. Cl. Hits (TB 0.2)	U. Cl. Hits (TB 0.3)	Nov. Cl. Hits (TB 0.3)	U. Cl. Hits (TB 0.4)	Nov. Cl. Hits (TB 0.4)
RF-C or Baseline	351	304	242	138	169	72
RF-C	295	258	217	132	142	61
Baseline	200	153	124	21	115	30
RF-C and Baseline	144	107	99	15	88	19
RF-C but not Baseline	151	151	130	120	72	44
Baseline but not RF-C	63	48	43	7	40	13

Table S8: The 1,024 AMS compounds grouped by activity level. Only Strong Actives and Normal Actives were considered to be hits in all primary analyses. Weak Actives fell below the % inhibition threshold.

Selector	Count	Hits	Strong Actives	Normal Actives	Weak Actives
RF-C or Baseline	1,024	412	221	191	178
RF-C	701	337	208	129	110
Baseline	705	256	111	145	146
RF-C and Baseline	382	181	98	83	78
RF-C but not Baseline	319	156	110	46	32
Baseline but not RF-C	323	75	13	62	68



Table S9: Illustration of the five AMS hits selected by RF-C that are the farthest from their nearest training set active.

Prospective Hit	Closest Training Active	Similarity Map	Tanimoto Distance
			0.57
			0.56
			0.55
			0.55
			0.53

Table S10: Illustration of the five AMS hits selected by the Similarity Baseline that are the farthest from their nearest training set active.

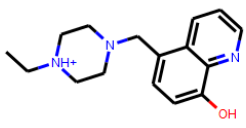
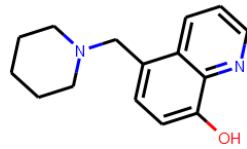
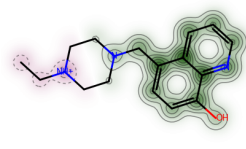
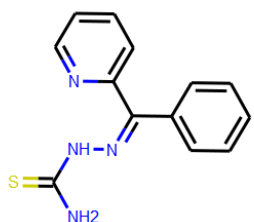
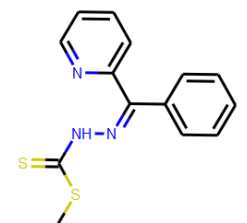
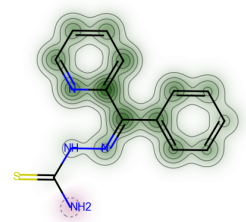
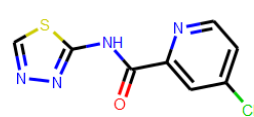
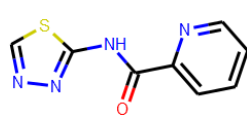
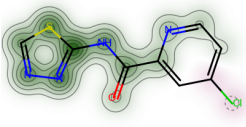
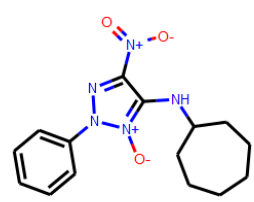
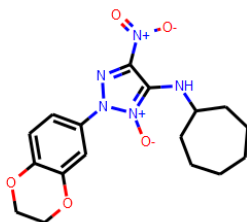
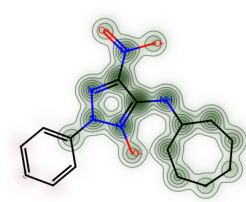
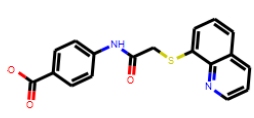
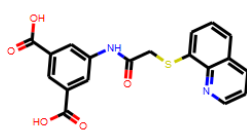
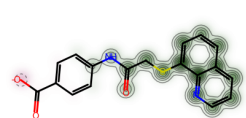
Prospective Hit	Closest Training Active	Similarity Map	Tanimoto Distance
			0.32
			0.32
			0.32
			0.31
			0.31

Table S11: Timing estimates in hours for RF-C prediction on the Enamine REAL library consisting of 1,077,562,987 compounds. The library processing was split among 18 compute nodes.

Part	Hours	# compounds
0	50.83	60262531
1	63.71	60262531
2	46.62	60262531
3	45.81	60262531
4	63.43	60262531
5	63.70	60262531
6	51.81	60262531
7	64.35	60262531
8	49.12	60262531
9	49.86	60262531
10	52.74	60262531
11	46.51	60262530
12	54.52	60263531
13	49.47	60263531
14	47.12	60263531
15	55.65	60263531
16	55.88	60263531
17	46.63	53094961
Mean	53.21	59864610
Std	6.40	1641881

Table S12: Summary of fold information denoting the number of compounds, number of clusters, and number of clusters not present in other folds. Compounds were clustered using Taylor-Butina at the 0.4 distance threshold.

Fold ID	# Compounds	# Clusters	# Clusters not in other folds
0	42729	26753	3377
1	42730	26734	3392
2	42737	26696	3276
3	42734	26709	3326
4	42730	26708	3352
5	42729	26772	3342
6	42730	26746	3397
7	42727	26717	3345
8	42727	26783	3442
9	42727	26667	3386
All folds	427300	88324	33635

Table S13: Random Forest-Classification. 216 hyperparameter combinations.

Hyperparameter	Hyperparameter values
number of estimators	50; 250; 1000; 4000; 8000; 16000
max features	None; Sqrt; Log2
min sample leaf	1; 10; 100; 1000
class weight	None; balanced subsample, balanced

Table S14: XGBoost-Classification. 1,866,240 hyperparameter combinations from which we randomly select 1,000.

Hyperparameter	Hyperparameter values
max depth	5; 10; 50; 100
learning rate	1; 3e-1; 1e-1; 3e-2
number of estimators	30; 100; 300; 1000; 3000
subsample	1; 0.5
colsample bylevel	1; 0.5; 0.2
colsample bytree	1; 0.5; 0.2
min child weight	1; 5; 20
reg alpha	0; 0.3
reg lambda	1; 0.7
scale pos weight	1; 0.1; 0.001
max delta step	0; 1; 3
eval metric	AVG[PR]; AUC[ROC]
eval round	100; 300; 1000
booster	gbtree; gblinear

Table S15: XGBoost-Regression. 829,440 hyperparameter combinations from which we randomly select 1,000.

Hyperparameter	Hyperparameter values
max depth	5; 10; 50; 100
learning rate	1; 3e-1; 1e-1; 3e-2
number of estimators	30; 100; 300; 1000; 3000
subsample	1; 0.5
colsample bylevel	1; 0.5; 0.2
colsample bytree	1; 0.5; 0.2
min child weight	1; 5; 20
reg alpha	0; 0.3
reg lambda	1; 0.7
scale pos weight	1; 0.3; 0.1; 0.003
eval metric	AVG[PR]; AUC[ROC]
eval round	100; 300; 1000
booster	gbtree; gblinear

Table S16: Single-task Neural Network-Classification. 432 hyperparameter combinations.

Hyperparameter	Hyperparameter values
learning rate	0.00003; 0.0001; 0.003
weighted schema	no weight; weighted sample
epoch patience	epoch size: 200, patience: 50; epoch size: 1000, patience: 200
early stopping	AUC[ROC]; AVG[PR]
drop out	0.1, 0.25
layers	500(ReLU), 200(Sigmoid), 1(Sigmoid); 1000(ReLU), 500(Sigmoid), 1(Sigmoid); 2000(ReLU), 2000(Sigmoid), 1(Sigmoid); 2000(ReLU), 2000(ReLU), 1(Sigmoid); 2000(Sigmoid), 2000(Sigmoid), 1(Sigmoid); 2000(ReLU), 4000(Sigmoid), 2000(Sigmoid), 1(Sigmoid); 2000(ReLU), 1000(Sigmoid), 1(Sigmoid); 2000(ReLU), 4000(Sigmoid), 1000(Sigmoid), 1(Sigmoid); 2000(ReLU), 4000(Sigmoid), 8000(Sigmoid), 1000(Sigmoid), 1(Sigmoid)

Table S17: Single-task Neural Network-Regression. 432 hyperparameter combinations.

Hyperparameter	Hyperparameter values
learning rate	0.00003; 0.0001; 0.003
weighted schema	no weight; weighted sample
epoch patience	epoch size: 200, patience: 50; epoch size: 1000, patience: 200
early stopping	AUC[ROC]; AVG[PR]
drop out	0.1, 0.25
layers	500(ReLU), 200(Sigmoid), 1(Linear); 1000(ReLU), 500(Sigmoid), 1(Linear); 2000(ReLU), 2000(Sigmoid), 1(Linear); 2000(ReLU), 2000(ReLU), 1(Linear); 2000(Sigmoid), 2000(Sigmoid), 1(Linear); 2000(ReLU), 4000(Sigmoid), 2000(Sigmoid), 1(Linear); 2000(ReLU), 1000(Sigmoid), 1(Linear); 2000(ReLU), 4000(Sigmoid), 1000(Sigmoid), 1(Linear); 2000(ReLU), 4000(Sigmoid), 8000(Sigmoid), 1000(Sigmoid), 1(Linear)

Table S18: The 13 different variants of the ensemble models. An ensemble model combines the top  $N$  models from each type of base machine learning model, where  $N$  is listed below.

ID	RF-C	XGB-C	XGB-R	NN-C	NN-R
0	1	0	0	0	0
1	0	1	0	0	0
2	0	0	1	0	0
3	0	0	0	1	0
4	0	0	0	0	1
5	1	1	1	1	1
6	2	2	2	2	2
7	2	2	1	2	1
8	5	5	3	5	3
9	10	10	5	10	5
10	5	5	10	5	10
11	20	20	10	20	10
12	10	10	20	10	20