VIPER: A General Model for Prediction of Enzyme Substrates

Maxwell J. Campbell
max@hearthindustries.io

Jun 20, 2024

Abstract

Enzymes, nature's catalysts, possess remarkable properties such as high stereo-, regio-, and chemo-specificity. These properties allow enzymes to greatly simplify complex synthetic processes, resulting in improved yields and reduced manufacturing costs compared to traditional chemical methods. However, the lack of experimental characterization of enzyme substrates, with only a few thousand out of tens of millions of known enzymes in Uniprot having annotated substrates, severely limits the ability of chemists to repurpose enzymes for industrial applications. Previous machine learning models aimed at predicting enzyme substrates have been hampered by poor generalization to new substrates. Here, we introduce VIPER (Virtual Interaction Predictor for Enzyme Reactivity), a model that achieves an average 34% improvement over the previous state-of-the-art model (ProSmith) in reaction prediction for unseen substrates. Furthermore, we present a novel benchmarking methodology for assessing the out-of-distribution generalization capabilities of enzyme-substrate prediction models. VIPER represents a significant advance towards the in silico prediction of enzyme-substrate compatibility, paving the way for the discovery of novel biocatalytic routes for the sustainable synthesis of high-value chemicals.

Keywords Bioinformatics · Machine Learning · Enzyme Substrate Prediction

1 Introduction

Enzymes, nature's catalysts, possess unique catalytic properties, including high stereospecificity, regiospecificity, and chemospecificity [1]. These properties allow for the simplification of complex synthetic processes including ones used in the synthesis of several FDA-approved drugs [2, 1]. One recent example is the biosynthesis of QS-21 [3], a potent immunostimulatory agent used in several FDA-approved vaccines [4]. While the chemical synthesis of the Api and Xyl components of QS-21 requires 76 steps [5–7], the biosynthesis only requires 20 steps [3], significantly improving the yield and potentially significantly decreasing the cost of manufacturing QS-21. Additionally, enzymes have great potential for use as eco-friendly catalysts as they operate under mild reaction conditions, such as ambient temperature and pressure making them an eco-friendly choice for industrial processes [8].

However, very few enzyme substrates have been experimentally characterized, with only a few thousand out of tens of millions of enzymes in Uniprot [9] having experimentally confirmed

substrate annotations, making repurposing previously discovered enzymes for use in industrial reactions difficult, as the enzyme substrates must be manually characterized in high-throughput screens, a time-consuming and expensive endeavor. This limitation hampers the ability of chemists to select an enzyme off the shelf as they would a chemical reagent for a traditional reaction, making it difficult to fully realize the potential of enzymes in industry.

Previous works have attempted to address this issue by introducing machine learning models for the prediction of enzyme substrates. Three models for this task have been introduced previously: ESP [10], ProSmith [11], and Ridge Regression [12]. ESP is based on XGBoost [13] and uses data scraped from GO annotations [14]. It is trained on fine-tuned ESM-1b embeddings [15] and Extended-connectivity fingerprints (ECFPs) [16]. While ESP is capable of learning to predict enzyme-substrate reactions for the substrates included in its training set, as we will discuss later it performs poorly for distant substrates achieving an MCC of 0.219. ProSmith also uses data scraped from GO annotations, it is a hierarchical combination of a transformer model [17] that takes in ChemBERTa2 [18] and ESM-1b embeddings, then feeds the output into an ensemble of XGBoost heads, along with the original ChemBERTa2 and ESM-1b embeddings. Although ProSmith outperforms ESP it still only achieves an MCC of 0.291 on our unseen substrate benchmark. Finally, the Ridge Regression model uses data from high-throughput screens with ESM-1b and ECFP embeddings. The Ridge Regression model performs better than ESP but slightly worse than ProSmith achieving an MCC score of 0.287.

Other work has introduced models for the prediction of enzyme substrates within singular families. Two models have been developed for this purpose. Mou et al. developed a decision-tree-based model that predicts the substrates of bacterial nitrilases using substrate-protein 3D model-based descriptors trained on 12 nitrilases and 20 nitriles [19]. The other model proposed by Yang et al. also uses a decision tree approach with manual chemical and enzyme features trained on 54 glycosyltransferases and 94 substrates [20]. Unfortunately, like the previously discussed models, these models only achieve good performance within the enzyme family they are trained on, and do not learn general enzymatic properties or reaction schemes [10].

There have been some inroads made into the prediction of Enzyme Commission (EC) numbers for enzymes. Models such as CLEAN [21] and DEEPre [22] have had great success in grouping unannotated enzymes into families. Unfortunately, they are not useful for profiling the activity of enzymes toward specific substrates beyond knowing what family the enzyme lies in.

While previous models have taken significant steps towards achieving the ideal of enzyme-substrate prediction, as we will show later they either do not generalize well to new substrates [12, 10, 11], or are too broad to be of practical use when the family an enzyme lies in is already known [22, 21]. A model's practical use is dependent on its ability to be interpreted from a physicochemical standpoint, rather than relying on hidden patterns in the training data or rote memorization [23]; therefore, we validated that our model learns generalizable enzyme-substrate reaction mechanics by testing it's out-of-distribution generalization capabilities.

In this work, we introduce VIPER (Virtual Interaction Predictor for Enzyme Reactivity), a machine learning (ML) model that outperforms previous models in predicting enzyme-substrate reactions on unseen substrates and exhibits superior out-of-distribution generalization. In addition to evaluating VIPER on unseen substrate performance we conduct a thorough investigation of VIPER's out-of-distribution generalization capabilities and compare its performance to previously introduced enzyme-substrate prediction models. Furthermore, we introduce an enhanced dataset for enzyme-substrate interactions, building upon the work of

Goldman et al. [12] by incorporating additional data cleaning and standardization techniques to ensure high-quality data.

2 Methods

2.1 Software

ML models were implemented in PyTorch [24] in Python [25]. XGBoost [13] was used for gradient-boosting tree models.

2.2 Acquisition of data

We acquired the sets: phosphatases [26], esterases [27], glycosyltransferases [20], beto-keto acid cleavage enzymes [28], thiolases [29], halogenases [30], aminotransferases [31], and nitrilases [32] from the pre-processed data made available by Goldman et al. [12].

2.3 Preprocessing of data

We found that the data for the BKACE set [28] made available by Goldman et al. [12] did not include the full enzyme sequence, unlike the other sets but instead just the results of a multi-sequence alignment. To resolve this we acquired the raw sequences with Uniprot IDs from the source paper's supplementary [28] and the raw MSA data with Uniprot IDs from the unprocessed data made available by Goldman et al. [12]. We then used this information to map the correct amino acid sequences onto the BKACE dataset.

Some of the sets used isomeric SMILES strings while others used canonical SMILES strings [33]. Additionally, the sets with isomeric SMILES strings were of dubious quality. To resolve this we used RDKit [34] with the Merck molecular force field optimization [35] to generate an initial geometry for the molecule which was then saved to an xyz file. We then used the initial structures as input for xTB [36] with the GFN2 model [37] to perform a semi-empirical geometry optimization in gas-phase with the energy convergence limit set to 5×10^{-8} hartrees and the gradient convergence set to 5×10^{-5} $\mathrm{Eh} \cdot \alpha^{-1}$. We then extracted SMILES strings from these molecular structures using RDKit. Some molecules were excluded as RDKit was unable to determine the appropriate bond order. These SMILES strings were then standardized using the PubChem standardization service via their API [38]. We did not process the nitrilase set in the same manner as RDKit was not able to extract an appropriate bond order for a large number of molecules in this set instead the SMILES strings provided by Goldman et al. [12] were canonicalized with RDKit and then standardized with the PubChem standardization service via their API [38]. This pre-processing pipeline was applied to all external models in the same manner as it was applied to VIPER.

Activity data for the phosphatases [26], esterases [27], glycosyltransferases [20], beto-keto acid cleavage enzymes [28], thiolases [29], halogenases [30], aminotransferases [31], and nitrilases [32] were binarized according to cutoff values provided by Goldman et al. [12].

2.4 Creating enzyme representations

To create the enzyme representations, we used the Ankh base model (450M parameters), a protein language model based on the transformer architecture introduced by Elnaggar et al. [39]. Ankh takes the protein amino acid sequence as input and passes it through a modified T5

transformer architecture [40], producing embeddings of shape [Sequence Length, 768] where 768 is the hidden size of the model. We decided to use Ankh because of its state-of-the-art (SOTA) performance and its computational efficiency. Ankh was pre-trained on the UniRef50 database [41] with a masking objective. We used the Ankh implementation provided by Elnaggar et al. on GitHub [39].

2.5 Creating molecule representations

To generate molecular representations, we employed Molformer, a chemical language model developed by Ross et al. [42] that utilizes a 12 layer transformer architecture. Molformer takes a SMILES string [33] as input and generates a pooled embedding of shape [768] for each molecule, where 768 is the hidden size of the model. The model was pre-trained on 10% of the ZINC database [43] and 10% of the PubChem [44] database using a masking objective. We used the Molformer implementation provided by Ross et al. on GitHub [42].

2.6 Validating external models

We acquired the source code for external models from their public GitHub repos [10–12]. The Goldman et al. Ridge Regression [12] was implemented in Sci-Kit learn [45]. ESP [10] was implemented in XGBoost [13]. ProSmith [11] was implemented with Pytorch [24] and XGBoost [13]. All models were trained with the same dataset used with VIPER and with the same testing methodology. The performance reported for ProSmith [11] is the best-weighted combination of ESM-1b+ChemBERTa2+cls outputs from the gradient boosting ensemble. Additionally, ProSmith was run without pretraining for the out-of-distribution benchmark as this may bias our benchmark due to data leakage between the pre-training dataset and the test dataset. ProSmith was pretrained for the unseen molecule benchmark.

For ESP [10] the implementation provided by Kroll et al. had to be modified as the default implementation was fully deterministic, which proved to be problematic, as the model requires some randomness to be able to measure the standard error for the creation of error bars. To resolve this issue we set the subsample parameter, which controls how much data is sampled for each tree in the network [13], to 0.5 instead of the default setting of 1.0. We subsequently validated that our adjustment has minimal impact on the performance of ESP. Finally, the K-Nearest Neighbors (KNN) model was implemented with Scikit-Learn [45]. As input, we used a 1024 dimensional molecular ECFP fingerprint [16] with a radius of three together with the enzyme ESM-1b embedding [15].

2.7 Graphing

The Python [25] library Vaex [46] was used to plot Figure 2. Vaex was chosen over other graphing libraries due to its ability to handle large datasets.

3 Results

3.1 Our model architecture

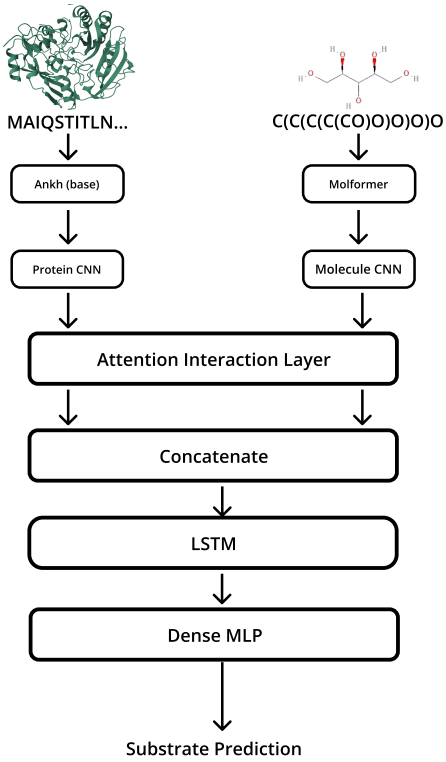

Figure 1: VIPER Model architecture displayed as pictogram.

Enzyme-substrate prediction is a challenging task due in part to the limited amount of available data. To address this issue, VIPER employs a transfer learning approach by utilizing pre-trained transformer models to generate embeddings for both the input molecule and protein. This allows the model to leverage knowledge about proteins and molecules acquired during pre-training and apply it to enzyme-substrate prediction. For protein embedding, VIPER utilizes Ankh base [39], a 450 million parameter protein language model. Ankh base was pre-trained on the UniRef50 [9] database, which consists of 45 million protein sequences, enabling it to capture a wide range of protein characteristics and features. For molecule embedding, VIPER employs Molformer [42], a 44 million parameter chemical transformer model pre-trained on a dataset consisting of 10% of ZINC [43] and 10% of Pubchem [44], amounting to roughly 110 million molecules. This extensive pre-training allows Molformer to learn a rich representation of molecular structures and properties. Our model then takes the Ankh and Molformer embeddings as input, which are subsequently processed by separate Convolutional Neural Networks (CNNs). The CNNs serve two key purposes in the model architecture: first, they extract meaningful features from the input embeddings; second, they transform the embedding tensors to have consistent dimensions, which is necessary for the embeddings to be effectively utilized by the remainder of the model. The CNN outputs are then fed into a cross-attention layer [17]; the cross-attention mechanism helps VIPER learn interactions between the molecule and protein. After cross-attention is applied, the resulting protein and molecule embeddings are concatenated together. This combined representation is then fed into a bidirectional Long Short-Term Memory (LSTM) layer. The LSTMs ability to capture long-range dependencies allows the model to learn interactions between distant protein and molecule features. The output of the LSTM is then fed into a Multi-Layer Perceptron (MLP). The final output is a prediction score between 0 and 1, indicating the likelihood of interaction.

To optimize the model, VIPER combines the AdamW optimizer [47] with Sharpness Aware Minimization (SAM) [48]. SAM is used in conjunction with AdamW to encourage the model to converge to flatter minima in the loss landscape. Flatter minima are associated with better generalization performance, as they are less sensitive to small perturbations in the input data [49, 48]. This is particularly important for enzyme-substrate prediction as the data is inherently noisy, as some degree of experimental variance is always expected. To handle the imbalanced nature of the dataset, VIPER uses a weighted binary cross-entropy loss function. The weights are set to a 0.3:1 ratio, giving more importance to the minority class (Positive reactions). This helps prevent the model from being biased towards the majority class and ensures that it learns to correctly classify both positive and negative samples.

3.2 Our dataset

Our dataset consisted of nine high-throughput enzyme substrate screens extracted from the literature. This dataset consisted of a relatively diverse set of enzyme families, including phosphatases [26], esterases [27], glycosyltransferases [20], beta-keto acid cleavage enzymes [28], thiolases [29], halogenases [30], aminotransferases [31], and nitrilases [32]. Using this experimental data provided several key advantages over relying on datasets derived from GO annotations [14] or the BRENDA database [50].

One of the most significant advantages of our dataset lies in its ability to avoid misannotations, which are pervasive in human-curated annotation databases. Numerous studies have documented the prevalence of misannotations across various databases, including KEGG, BRENDA, UniProtKB, and GenBank NR [51, 52]. In some cases, misannotation rates have been reported to exceed 75% across different EC numbers and enzyme families [51, 52]. Previous studies focused on enzyme-substrate prediction have sought to address these issues by implementing heuristics and rule-based systems to clean the data [53]. However, the effectiveness of such methods is limited when dealing with data characterized by such high error rates. A particularly striking example is presented by Rembeza et al. [52], where they analyzed 122 representative sequences of the EC number 1.1.3.15 in the BRENDA database and discovered that at least 78% of the data in this family was incorrectly annotated. By directly utilizing data obtained from high-throughput enzyme-substrate screens, our approach effectively circumvents these challenges.

Our dataset offers another key advantage in that it includes negative enzyme-substrate pair reactions, owing to its direct acquisition from high-throughput enzyme-substrate screens. Such information is frequently absent in annotation databases, which has resulted in previous work [10, 11] artificially creating negative samples resulting in reduced data quality. Moreover, high-throughput screens generate data with substantially lower sparsity (i.e., fewer missing enzyme-substrate pairs) and decrease experimental variance by assessing enzymes against a wide range of substrates under controlled conditions. In contrast, annotation datasets often suffer from high sparsity and high experimental variance.

The selection of enzyme families for inclusion in our dataset was based on several criteria: data accessibility, the presence of non-proprietary enzymes, and overall data quality. This approach ensured a relatively diverse representation of enzyme classes and reaction types, thereby enhancing the broad applicability of our model. The final dataset comprised approximately 51,000 experimental observations. It is worth noting that the data for the phosphatases set was significantly more extensive than data for other sets, constituting 44.4% of the dataset.

3.3 Improved generalization to unseen substrates with VIPER

ESP	ProSmith	Ridge	VIPER
0.219±0.001	0.291±0.017	0.287±0.01	**0.390±0.01**

Table 1: Our model compared with previously introduced models on MCC metric for unseen substrates. The best MCC score is bolded. All models were run in triplicate to compute the standard error.

One common application of enzyme-substrate prediction models is where a chemist aims to replace a chemical catalyst with an enzymatic alternative. While enzyme-substrate prediction models have made significant progress, their performance on unseen enzyme families remains a challenge. Because of this, VIPER is intended to be used in scenarios where the target enzyme family is included in the model's training data, but the molecule is not. To evaluate VIPER's performance in this setting, we employed a sphere exclusion split on molecular extended connectivity fingerprints (ECFPs) [16] to create a training set, validation set, and testing set with dissimilar molecules. The sphere exclusion method ensures that the test set molecules are sufficiently different from the training set molecules by enforcing a minimum Euclidean distance between their ECFP representations. For our experiments, we set the Euclidean distance threshold to 0.25.

For VIPER, the validation set was used to select the best performing model out of ten training epochs, while for ProSmith, it was used to determine the optimal weighting between three gradient boosting tree models. The validation set was not utilized for the Ridge and ESP models, as these models do not utilize early stopping, hyperparameter tuning, or best epoch selection.

As shown in Table 1, VIPER surpasses the performance of the previous state-of-the-art model, ProSmith, by 34%, achieving a Matthews Correlation Coefficient (MCC) of 0.390 on unseen substrates (McNemar's test P-value: <0.0001 [54]). The Ridge Regression model proposed by Goldman et al. [12] achieves an MCC of 0.287, followed by ESP [10] with an MCC of 0.219.

3.4 Out-of-distribution benchmarking methodology

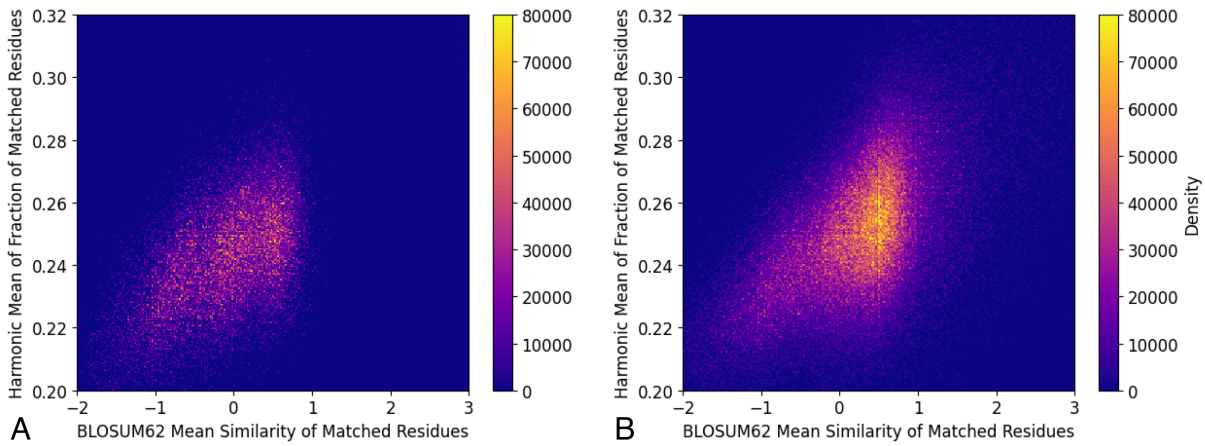

Figure 2: **A.** Enzyme family splitting methodology on Esterases (Our splitting methodology) **B.** 80/20 train-test split with 40% sequence similarity cutoff (Previous splitting methodology).

Every enzyme sequence in the test set was compared to every enzyme sequence in the train set using the Needleman–Wunsch global alignment algorithm [55].

To evaluate the out-of-distribution generalization performance of our model we broke down our dataset into k folds, where k is the number of enzyme families; we then trained the model on k - 1 folds where the fold subtracted was then used as the test set, in a manner similar to a typical KFold cross-validation. The performance reported for each fold is the model's best performance out of ten epochs. In addition to evaluating our model on out-of-distribution generalization without a test set, we also test our model using two other validation techniques. The first is to exclude 10% of the training set and use that as a validation set, and the second is to exclude a second unseen out-of-distribution enzyme family and use that as the validation set. These additional validation techniques aid in confirming the out-of-distribution generalization capabilities of VIPER. This methodology ensures minimal data similarity between the folds, as almost no substrates are shared between the groups and the catalytic mechanism of the enzymes is inherently different. This is exemplified by Figure 2 where we find that our benchmarking methodology significantly reduces sequence similarity between the train set and test set enzymes with an average 601.312% decrease in BLOSUM62 Mean Similarity of Matched Residues (Mann-Whitney U test: p-value <0.0001) and a 3.753% decrease in Harmonic Mean of Fraction of Matched Residues (Mann-Whitney U Test: p-value <0.0001). Additionally, we find that our splitting methodology lowers the mean Tanimoto coefficient by 45.3% (Supplementary Table 2) showing that our splitting methodology significantly reduces both enzyme similarity and molecule similarity. To further ensure that there is minimal data similarity we used a K-Nearest Neighbors (KNN) model to see how well a purely memorization-driven approach performs. Additionally, all models were run in triplicate to be able to calculate standard error for the creation of error bars. We employed Matthews correlation coefficient (MCC) to evaluate our model, as it integrates information about all four quadrants of the confusion matrix. This makes it a significantly more robust measure of performance than AUPRC or AUROC [56].

3.5 VIPER exhibits superior generalization performance on out-of-distribution data compared to prior models

Dataset	KNN	ESP	ProSmith	Ridge	VIPER
Phosphatases	0.02±0.001	0.012±0.0	−0.009±0.008	**0.026±0.002**	0.002±0.004
Esterases	0.094±0.004	−0.014±0.0	0.06±0.021	−0.015±0.019	**0.129±0.019**
GT Acceptors	0.071±0.0	−0.08±0.001	0.001±0.0	0.076±0.005	**0.229±0.018**
GT Donors	−0.058±0.0	0.001±0.022	0.0863±0.028	0.001±0.0	**0.191±0.025**
BKACE	0.086±0.009	−0.012±0.007	0.001±0.051	0.052±0.016	**0.104±0.024**
Thiolases	−0.04±0.066	0.001±0.0	0.069±0.023	0.001±0.008	**0.305±0.01**
Halogenases	−0.045±0.008	0.001±0.002	0.001±0.014	0.06±0.02	**0.17±0.007**
Aminotransferases	0.056±0.0	0.001±0.0	−0.183±0.065	0.001±0.023	**0.098±0.037**
Nitrilases	−0.034±0.004	0.042±0.0	0.001±0.023	0.084±0.035	**0.087±0.003**

Table 2: Performance of our model and previously introduced models across different unseen enzyme subsets. Best MCC score is bolded.

As can be seen in Table 2, VIPER performs substantially better than ProSmith, ESP, KNN, and Ridge in our test suite. On average, VIPER performs 352.492% better (McNemar's test P-value: <0.0001 [54]) than the next best model (Ridge). ESP [10] is nearly equivalent to random chance on most folds, the Ridge Regression and ProSmith [11] models performed poorly on all folds albeit better than ESP. The KNN model produces roughly random predictions across all enzyme families further validating that our benchmark significantly mitigates data leakage and provides a more realistic evaluation of model performance. It is of note that all five models perform quite poorly on the phosphate enzyme family; this is due to this subset of the data composing 44.4% of our training dataset thus when it is removed so it can be evaluated the models perform worse this can be seen in Supplementary Table 1.

Dataset	**Ridge**	**VIPER**	**VIPER ID**	**VIPER OOD**
Phosphatases	**0.026**	0.002	−0.010	−0.038
Esterases	−0.015	(0.129)	(0.047)	(**0.146**)
GT Acceptors	0.076	(**0.229**)	(0.088)	(0.089)
GT Donors	0.001	(**0.191**)	(0.138)	(0.057)
BKACE	0.052	(**0.104**)	−0.04	−0.064
Thiolases	0.001	(**0.305**)	(0.210)	(0.153)
Halogenases	0.06	(**0.170**)	(0.061)	0.025
Aminotransferases	0.001	(**0.098**)	−0.081	0.000
Nitrilases	0.084	(**0.087**)	0.059	0.074

Table 3: VIPER without a validation set compared to viper with an in-distribution (ID) validation set and VIPER with an out-of-distribution (OOD) validation set. The best performing model on the previous test (Ridge) was also included. The best test-set MCC score is bolded. If a VIPER model acheives a better score than the score of the best previous model (Ridge) it is placed in (brackets).

To test the robustness of VIPER on out-of-distribution data, we compared two different validation methodologies in addition to testing the model without a validation set. First, we selected 10% of the training set and used it as an in-distribution (ID) validation set; second, we excluded a second unseen enzyme family to use as an out-of-distribution (OOD) validation set. The validation set is then used to select the best performing epoch out of ten. As can be seen in Table 3, when the in-distribution validation set is used the test set performance is 44.4% better than the next best model (Ridge), and when the out-of-distribution validation set is used the test set performance is 54.5% better than the next best model (Ridge).

While VIPER with the in-distribution validation set and out-of-distribution validation set still perform better on this benchmark than previous models, we can see that VIPER's performance on the out-of-distribution test set is unstable compared to the baseline VIPER model which is evaluated without a validation set. The instability of the out-of-distribution test set generalization performance is most likely due to a number of factors afflicting our dataset, such as, a lack of diversity in enzyme families, which could cause the model to learn more specialized solutions rather than broad reaction kinetics [57, 23]; and high enzyme-substrate pair sparsity, which could impair the model's ability to learn all possible reactions given the substrates and enzymes in the dataset [23]. These factors could introduce hidden biases that

push the model away from out-of-distribution generalization and incentivizes memorization and interpolation over extrapolation.

Examining the out-of-distribution validation set performance compared to the in-distribution validation set performance, we can see that when the out-of-distribution validation set is used the observed test set performance of the model is more stable. But despite this, the per epoch MCC for the out-of-distribution validation set and test set do not always align. While good out-of-distribution validation performance has more correlation to good out-of-distribution test set performance than for the in-distribution validation set, there are still instances where validation performs well and test performs poorly and vice versa; as an example, the per epoch validation and test set performance for the esterase enzyme family can be found in Supplementary Table 2. This instability points towards limitations of the existing data, causing an inability for enzyme-substrate models to learn solutions that can generalize well to out-of-distribution data.

3.6 Pre-training VIPER

Dataset	VIPER	VIPER P	VIPER FP
Phosphatases	**0.001**	0.0	0.0
Esterases	0.128	0.042	**0.192**
GT Acceptors	**0.229**	0.168	0.201
GT Donors	0.190	**0.275**	0.136
BKACE	0.104	**0.111**	0.056
Thiolases	**0.304**	0.292	0.221
Halogenases	**0.169**	0.168	0.146
Aminotransferases	0.097	**0.173**	0.059
Nitrilases	**0.087**	0.031	0.067
Unseen substrates	**0.390**	0.383	0.170

Table 4: Performance comparison of VIPER variants on unseen enzyme families and unseen substrates. VIPER represents the base model without pre-training, while VIPER P and VIPER FP denote the pre-trained model and the pre-trained model with frozen top layers, respectively. The values reported for Phosphatases - Nitrilases are for out-of-distribution generalization, and 'Unseen Substrates' represents the in-distribution test discussed earlier. Performance is reported as test set MCC score.

We hypothesized that pre-training VIPER on a ligand-protein binding task would improve performance due to the similarity between enzyme-substrate prediction and molecule-protein binding prediction, and the fact that there is substantially more high-quality data available for molecule-protein binding prediction than is available for enzyme-substrate prediction. We theorized that this would allow VIPER to learn more complex interactions between proteins and molecules. To pretrain VIPER, we used the high-quality subset of the Papyrus dataset [58] which consisted of ~550,000 rows with ~272,000 molecules and ~3,200 proteins. We then trained VIPER on Ki, Kd, and IC50 values using multi-task learning. VIPER reached a minimum test set RMSE of 0.349 (Random split) on the Ki prediction task. We then loaded these saved weights into VIPER for our enzyme-substrate prediction task. We attempted fine-tuning VIPER both with fully unfrozen layers and by freezing the top CNN layers, the cross-attention layer, and the

LSTM. However, as evident in Table 4, we did not observe any performance improvement from pre-training VIPER, and in some cases, pretraining with the top layers frozen even reduced performance on tasks such as unseen substrates. The reason for this lack of improvement remains unclear to us, but it may suggest that we have reached a limit in the generalization performance attainable from our current training scheme and features.

3.7 VIPER can express confidence

Discriminating between results with high and low confidence is crucial for a model intended for practical use. While there are various techniques to assess the uncertainty of a deep learning model such as Monte Carlo Dropout [59], and Bayesian Neural Networks [60], we opted to use deep ensembles [61], as Monte Carlo dropout can produce overconfident results compared to deep ensembles [59], and Bayesian approximation of uncertainty would prove too computationally intensive for this study.

Dataset	**LOW**	**MEDIUM**	**HIGH**
Phosphatases	0.001	0.001	0.001
Esterases	0.065	0.129	**0.228**
GT Acceptors	0.005	0.186	**0.31**
GT Donors	0.001	**0.17**	0.157
BKACE	0.004	−0.002	**0.179**
Thiolases	0.055	**0.414**	0.342
Halogenases	**0.24**	0.194	0.101
Aminotransferases	0.001	0.001	**0.071**
Nitrilases	**0.184**	0.158	−0.028
Unseen substrates	0.281	0.324	**0.495**

Table 5: VIPER Performance segmented by confidence across unseen enzyme families and unseen substrates. The best test set MCC score is bolded.

We created an ensemble of three VIPER models and calculated the standard deviation for their predictions to represent the uncertainty of the model, then grouped predictions into three categories: LOW, MEDIUM, and HIGH based on the standard deviation. We set the cutoff for HIGH confidence predictions as a prediction with a standard deviation less than 0.04 and the cutoff for LOW confidence predictions as any prediction with a standard deviation greater than 0.08; predictions that fall in between 0.04 and 0.08 were classified as MEDIUM confidence predictions. Using the Mann-Whitney U test between the LOW and HIGH classes, we found that VIPER performs better on HIGH confidence predictions for sets one through six with a p-value of 0.0079. Most of the sets (one through six) and the unseen substrate test show strong correlation but for some sets, this correlation is murkier; for example, the glycosyltransferase donor set has an average MEDIUM MCC; higher than the HIGH MCC class, but they are both still higher than the LOW class indicating that the model still maintains some correlation between confidence and performance. For sets seven through ten, the model confidence does not correlate with the prediction quality (Mann-Whitney U test LOW-HIGH p-value: 0.339). This issue might be alleviated with an increase in the size of the ensemble or with a full Bayesian approach, but we will leave this to future research.

3.8 A web server for easy use of VIPER

We implemented a web server for easy use of VIPER. It is available at https://viperwebserver.com/; the web server takes a SMILES string [33] and a protein amino acid sequence as input, and produces a prediction as well as a confidence value (LOW, MEDIUM, or HIGH). Additionally, the server allows users to upload a CSV with up to 100 rows for bulk predictions. We recommend that you utilize our open-source scripts for large bulk runs.

4 Discussion

Here we introduce VIPER, a novel machine learning model for enzyme-substrate prediction that outperforms the previously proposed models: ESP [10], ProSmith [11], and Ridge Regression [12]. Our model achieves an average MCC of 0.390 on our unseen substrate benchmark, representing a 34% improvement over the next best model, ProSmith. This substantial performance improvement represents a significant step toward practical in-silico prediction of enzyme substrates.

We have also introduced a novel benchmarking methodology to better understand the out-of-distribution generalization capabilities of enzyme-substrate prediction models. We achieved this by stratifying enzymes in our dataset into families, which is significantly more effective at minimizing enzyme-substrate pair similarity than sequence similarity, due to the inherent differences in catalytic function between enzyme families. Additionally this method significantly reduces molecule similarity where we observed a 45.3% drop in the mean Tanimoto coefficient.

Despite the significant performance improvement VIPER provides over previous models, the model's performance is still suboptimal for practical use. To further improve model performance, future work could explore the use of semi-supervised learning to increase the available training data or the extraction of data from databases such as BRENDA [50] or Sabio-RK [62]. Acquiring more diverse data is likely to lead to a large improvement in performance as our current dataset contains only a very narrow selection of enzymatic reactions and enzyme families, one key area where our dataset is deficient is coupling reactions performed by enzymes like ligases, synthetases, and polymerases. VIPER is unlikely to generalize well to these enzymes as the reactions they catalyze are significantly different from anything in our training data. The downside to acquiring data from Sabio-RK or BRENDA is it will have to be cleaned extensively as this data is of quite low quality due to misannotations, human error in curation, and high experimental variance [53, 51, 52]. Another possible approach for improving model performance is to integrate physical principles of molecule-protein interaction into enzyme-substrate prediction models, which could lead to further improvements in model performance [63, 64]. The addition of 3D protein information could also be explored as this is critical for determining substrate preferences in many enzyme-catalyzed reactions [65–68].

The successful development of a model that can generalize to new enzyme families and substrates has great potential for industrial synthesis. By utilizing the ability of enzymes to highly regiospecific, stereospecific, and chemospecific complex synthetic processes can be greatly simplified [2, 1], resulting in higher yields and fewer synthesis steps compared to traditional chemical synthesis. This is exemplified by the biosynthesis of QS-21, an FDA-approved immunostimulatory adjuvant commonly used in vaccines [4]. The biosynthesis of QS-21 requires 20 steps [3] in contrast to the chemical synthesis, which requires 76 steps [5–7], significantly improving the yield and potentially decreasing the manufacturing cost. Enzymes

have also been used to simplify the synthesis of several FDA-approved pharmaceuticals, such as Montelukast [69], Sitagliptin [70, 71], Pregabalin [72], and many other pharmaceuticals and high-value compounds [1]. Furthermore, VIPER can aid in understanding biological systems by increasing our understanding of substrates for uncharacterized enzymes in various organisms, providing new insights into metabolic pathways and disease mechanisms.

5 Data & Code Availability

All data and code is available at the GitHub repo: https://github.com/maxall41/VIPER-Code. Model weights can be found on Zenodo: https://zenodo.org/records/12151573.

Bibliography

[1] U. T. Bornscheuer, G. W. Huisman, R. J. Kazlauskas, S. Lutz, J. C. Moore, and K. Robins, "Engineering the third wave of biocatalysis," *Nature*, vol. 485, no. 7397, pp. 185–194, May 2012, doi: 10.1038/nature11117.

[2] S. Simić, E. Zukić, L. Schmermund, K. Faber, C. K. Winkler, and W. Kroutil, "Shortening Synthetic Routes to Small Molecule Active Pharmaceutical Ingredients Employing Biocatalytic Methods," *Chemical Reviews*, vol. 122, no. 1, pp. 1052–1126, 2022, doi: 10.1021/acs.chemrev.1c00574.

[3] Y. Liu *et al.*, "Complete biosynthesis of QS-21 in engineered yeast," *Nature*, May 2024, doi: 10.1038/s41586-024-07345-9.

[4] X. Lv *et al.*, "Chemical and biological characterization of vaccine adjuvant QS-21 produced via plant cell culture," *iScience*, vol. 27, no. 3, p. 109006–109007, 2024, doi: https://doi.org/10.1016/j.isci.2024.109006.

[5] P. Wang, Y.-J. Kim, M. Navarro-Villalobos, B. D. Rohde, and D. Y. Gin, "Synthesis of the Potent Immunostimulatory Adjuvant QS-21A," *Journal of the American Chemical Society*, vol. 127, no. 10, pp. 3256–3257, 2005, doi: 10.1021/ja0422007.

[6] Y.-J. Kim, P. Wang, M. Navarro-Villalobos, B. D. Rohde, DerryBerry, and D. Y. Gin, "Synthetic Studies of Complex Immunostimulants from Quillaja saponaria: Synthesis of the Potent Clinical Immunoadjuvant QS-21Aapi," *Journal of the American Chemical Society*, vol. 128, no. 36, pp. 11906–11915, 2006, doi: 10.1021/ja062364i.

[7] K. Deng, M. M. Adams, P. Damani, P. O. Livingston, G. Ragupathi, and D. Y. Gin, "Synthesis of QS-21-Xylose: Establishment of the Immunopotentiating Activity of Synthetic QS-21 Adjuvant with a Melanoma Vaccine," *Angewandte Chemie International Edition*, vol. 47, no. 34, pp. 6395–6398, 2008, doi: https://doi.org/10.1002/anie.200801885.

[8] E. P. Cipolatti *et al.*, "Chapter 5 - Enzymes in Green Chemistry: The State of the Art in Chemical Transformations," *Advances in Enzyme Technology.* in Biomass, Biofuels, Biochemicals. Elsevier, pp. 137–151, 2019. doi: https://doi.org/10.1016/B978-0-444-64114-4.00005-4.

[9] T. U. Consortium, "UniProt: the Universal Protein Knowledgebase in 2023," *Nucleic Acids Research*, vol. 51, no. D1, p. D523–D531, 2022, doi: 10.1093/nar/gkac1052.

[10] A. Kroll, S. Ranjan, M. K. M. Engqvist, and M. J. Lercher, "A general model to predict small molecule substrates of enzymes based on machine and deep learning," *Nature Communications*, vol. 14, no. 1, p. 2787–2788, May 2023, doi: 10.1038/s41467-023-38347-2.

[11] A. Kroll, S. Ranjan, and M. J. Lercher, "A multimodal Transformer Network for protein-small molecule interactions enhances drug-target affinity and enzyme-substrate predictions," *bioRxiv*, 2023, doi: 10.1101/2023.08.21.554147.

[12] R. A. Y. K. K. A. C. C. W. Goldman Samuel AND Das, "Machine learning modeling of family wide enzyme-substrate specificity screens," *PLOS Computational Biology*, vol. 18, no. 2, pp. 1–20, 2022, doi: 10.1371/journal.pcbi.1009853.

[13] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," Aug. 2016, doi: 10.1145/2939672.2939785.

[14] T. G. O. Consortium *et al.*, "The Gene Ontology knowledgebase in 2023," *Genetics*, vol. 224, no. 1, p. iyad31, 2023, doi: 10.1093/genetics/iyad031.

[15] A. Rives *et al.*, "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences," *PNAS*, 2019, doi: 10.1101/622803.

[16] D. Rogers and M. Hahn, "Extended-Connectivity Fingerprints," *Journal of Chemical Information and Modeling*, vol. 50, no. 5, pp. 742–754, 2010, doi: 10.1021/ci100050t.

[17] A. Vaswani *et al.*, "Attention Is All You Need," 2023, doi: https://doi.org/10.48550/arXiv.1706.03762.

[18] W. Ahmad, E. Simon, S. Chithrananda, G. Grand, and B. Ramsundar, "ChemBERTa-2: Towards Chemical Foundation Models." 2022. doi: https://doi.org/10.48550/arXiv.2209.01712.

[19] Z. Mou *et al.*, "Machine learning-based prediction of enzyme substrate scope: Application to bacterial nitrilases," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 3, pp. 336–347, 2021, doi: https://doi.org/10.1002/prot.26019.

[20] M. Yang *et al.*, "Functional and informatics analysis enables glycosyltransferase activity prediction," *Nature Chemical Biology*, vol. 14, no. 12, pp. 1109–1117, Dec. 2018, doi: 10.1038/s41589-018-0154-9.

[21] T. Yu, H. Cui, J. C. Li, Y. Luo, G. Jiang, and H. Zhao, "Enzyme function prediction using contrastive learning," *Science*, vol. 379, no. 6639, pp. 1358–1363, 2023, doi: 10.1126/science.adf2465.

[22] Y. Li *et al.*, "DEEPre: sequence-based enzyme EC number prediction by deep learning," *Bioinformatics*, vol. 34, no. 5, pp. 760–769, 2017, doi: 10.1093/bioinformatics/btx680.

[23] M. Volkov *et al.*, "On the Frustration to Predict Binding Affinities from Protein–Ligand Structures with Deep Neural Networks," *Journal of Medicinal Chemistry*, vol. 65, no. 11, pp. 7946–7958, 2022, doi: 10.1021/acs.jmedchem.2c00487.

[24] A. Paszke *et al.*, "PyTorch: An Imperative Style, High-Performance Deep Learning Library," vol. 32, p. , 2019, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2019/file/bdbca288fee7f92f2bfa9f7012727740-Paper.pdf

[25] G. Van Rossum, F. L. Drake, and others, *Python reference manual*, vol. 111. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[26] H. Huang *et al.*, "Panoramic view of a superfamily of phosphatases through substrate profiling," *Proceedings of the National Academy of Sciences*, vol. 112, no. 16, Apr. 2015, doi: 10.1073/pnas.1423570112.

[27] M. Martínez-Martínez *et al.*, "Determinants and Prediction of Esterase Substrate Promiscuity Patterns," *ACS Chemical Biology*, vol. 13, no. 1, pp. 225–234, 2018, doi: 10.1021/acschembio.7b00996.

[28] K. Bastard *et al.*, "Revealing the hidden functional diversity of an enzyme family," *Nature Chemical Biology*, vol. 10, no. 1, pp. 42–49, Jan. 2014, doi: 10.1038/nchembio.1387.

[29] S. L. Robinson, M. D. Smith, J. E. Richman, K. G. Aukema, and L. P. Wackett, "Machine learning-based prediction of activity and substrate specificity for OleA enzymes in the thiolase superfamily," *Synthetic Biology*, vol. 5, no. 1, p. ysaa4, 2020, doi: 10.1093/synbio/ysaa004.

[30] B. F. Fisher, H. M. Snodgrass, K. A. Jones, M. C. Andorfer, and J. C. Lewis, "Site-Selective C–H Halogenation Using Flavin-Dependent Halogenases Identified via Family-Wide Activity Profiling," *ACS Central Science*, vol. 5, no. 11, pp. 1844–1856, 2019, doi: 10.1021/acscentsci.9b00835.

[31] T. Li *et al.*, "Exploration of Transaminase Diversity for the Oxidative Conversion of Natural Amino Acids into 2-Ketoacids and High-Value Chemicals," *ACS Catalysis*, vol. 10, no. 14, pp. 7950–7957, 2020, doi: 10.1021/acscatal.0c01895.

[32] G. W. Black, N. L. Brown, J. J. B. Perry, P. D. Randall, G. Turnbull, and M. Zhang, "A high-throughput screening method for determining the substrate scope of nitrilases," *Chem. Commun. (Camb.)*, vol. 51, no. 13, pp. 2660–2662, Feb. 2015.

[33] D. Weininger, "SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules," *Journal of Chemical Information and Computer Sciences*, vol. 28, no. 1, pp. 31–36, 1988, doi: 10.1021/ci00057a005.

[34] G. Landrum, "RDKit: Open-Source Cheminformatics Software," 2016, [Online]. Available: https://github.com/rdkit/rdkit/releases/tag/Release_2016_09_4

[35] T. A. Halgren, "Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94," *Journal of Computational Chemistry*, vol. 17, no. 5–6, pp. 490–519, 1996, doi: https://doi.org/10.1002/(SICI)1096-987X(199604)17:5/6<490::AID-JCC1>3.0.CO;2-P.

[36] C. Bannwarth *et al.*, "Extended tight-binding quantum chemistry methods," *WIREs Computational Molecular Science*, vol. 11, no. 2, p. e1493, 2021, doi: https://doi.org/10.1002/wcms.1493.

[37] C. Bannwarth, S. Ehlert, and S. Grimme, "GFN2-xTB—An Accurate and Broadly Parametrized Self-Consistent Tight-Binding Quantum Chemical Method with Multipole Electrostatics and Density-Dependent Dispersion Contributions," *Journal of Chemical Theory and Computation*, vol. 15, no. 3, pp. 1652–1671, 2019, doi: 10.1021/acs.jctc.8b01176.

[38] V. D. Hähnke, S. Kim, and E. E. Bolton, "PubChem chemical structure standardization," *Journal of Cheminformatics*, vol. 10, no. 1, p. 36–37, Aug. 2018, doi: 10.1186/s13321-018-0293-8.

[39] A. Elnaggar *et al.*, "Ankh: Optimized Protein Language Model Unlocks General-Purpose Modelling," *arXiv preprint arXiv:2301.06568*, 2023.

[40] A. Elnaggar *et al.*, "ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing," 2021, doi: https://doi.org/10.48550/arXiv.2007.06225.

[41] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and the UniProt Consortium, "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches," *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2014, doi: 10.1093/bioinformatics/btu739.

[42] J. Ross, B. Belgodere, V. Chenthamarakshan, I. Padhi, Y. Mroueh, and P. Das, "Large-scale chemical language representations capture molecular structure and properties," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1256–1264, 2022, doi: 10.1038/s42256-022-00580-7.

[43] J. J. Irwin *et al.*, "ZINC20—A Free Ultralarge-Scale Chemical Database for Ligand Discovery," *Journal of Chemical Information and Modeling*, vol. 60, no. 12, pp. 6065–6073, 2020, doi: 10.1021/acs.jcim.0c00675.

[44] S. Kim *et al.*, "PubChem 2023 update," *Nucleic Acids Research*, vol. 51, no. D1, p. D1373–D1380, 2022, doi: 10.1093/nar/gkac956.

[45] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[46] M. A. Breddels and J. Veljanoski, "Vaex: big data exploration in the era of Gaia," *Astronomy & Astrophysics*, vol. 618, p. A13, Oct. 2018, doi: 10.1051/0004-6361/201732493.

[47] I. Loshchilov and F. Hutter, "Decoupled Weight Decay Regularization," 2017, doi: https://doi.org/10.48550/arXiv.1711.05101.

[48] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, "Sharpness-aware Minimization for Efficiently Improving Generalization," 2021, [Online]. Available: https://openreview.net/forum?id=6Tm1mposlrM

[49] S. Hochreiter and J. Schmidhuber, "Flat Minima," *Neural Computation*, vol. 9, no. 1, pp. 1–42, 1997, doi: 10.1162/neco.1997.9.1.1.

[50] A. Chang *et al.*, "BRENDA, the ELIXIR core data resource in 2021: new developments and updates," *Nucleic Acids Research*, vol. 49, no. D1, p. D498–D508, 2020, doi: 10.1093/nar/gkaa1025.

[51] S. D. A. D. I. A. B. P. C. Schnoes Alexandra M. AND Brown, "Annotation Error in Public Databases: Misannotation of Molecular Function in Enzyme Superfamilies," *PLOS Computational Biology*, vol. 5, no. 12, pp. 1–13, 2009, doi: 10.1371/journal.pcbi.1000605.

[52] M. K. M. Rembeza Elzbieta AND Engqvist, "Experimental and computational investigation of enzyme functional annotations uncovers misannotation in the EC 1.1.3.15 enzyme class," *PLOS Computational Biology*, vol. 17, no. 9, pp. 1–22, 2021, doi: 10.1371/journal.pcbi.1009446.

[53] E. Heid, D. Probst, W. H. Green, and G. K. H. Madsen, "EnzymeMap: curation, validation and data-driven prediction of enzymatic reactions," *Chem. Sci.*, vol. 14, no. 48, pp. 14229–14242, 2023, doi: 10.1039/D3SC02048G.

[54] T. G. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895–1923, 1998, doi: 10.1162/089976698300017197.

[55] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970, doi: https://doi.org/10.1016/0022-2836(70)90057-4.

[56] D. Chicco and G. Jurman, "The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification," *BioData Mining*, vol. 16, no. 1, p. 4–5, Feb. 2023, doi: 10.1186/s13040-023-00322-4.

[57] Z. Gong, P. Zhong, and W. Hu, "Diversity in Machine Learning," *IEEE Access*, vol. 7, pp. 64323–64350, 2019, doi: 10.1109/access.2019.2917620.

[58] O. J. M. Béquignon, B. J. Bongers, W. Jespers, A. P. IJzerman, B. van der Water, and G. J. P. van Westen, "Papyrus: a large-scale curated dataset aimed at bioactivity predictions," *Journal of Cheminformatics*, vol. 15, no. 1, p. 3–4, Jan. 2023, doi: 10.1186/s13321-022-00672-x.

[59] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning," in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., in Proceedings of Machine Learning Research, vol. 48. New York, New York, USA: PMLR, 2016, pp. 1050–1059. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html

[60] D. J. C. MacKay, "A Practical Bayesian Framework for Backpropagation Networks," *Neural Computation*, vol. 4, no. 3, pp. 448–472, 1992, doi: 10.1162/neco.1992.4.3.448.

[61] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles," vol. 30, p. , 2017, [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf

[62] U. Wittig, M. Rey, A. Weidemann, R. Kania, and W. Müller, "SABIO-RK: an updated resource for manually curated biochemical reaction kinetics," *Nucleic Acids Research*, vol. 46, no. D1, p. D656–D660, 2017, doi: 10.1093/nar/gkx1065.

[63] A. Kadambi, "Blending physics with artificial intelligence," vol. 11396, p. 113960–113961, 2020, doi: 10.1117/12.2565099.

[64] S. Cain, A. Risheh, and N. Forouzesh, "A Physics-Guided Neural Network for Predicting Protein–Ligand Binding Free Energy: From Host–Guest Systems to the PDBbind Database," *Biomolecules*, vol. 12, no. 7, 2022, doi: 10.3390/biom12070919.

[65] J. D. Woodward, I. Trompetter, B. T. Sewell, and M. Piotrowski, "Substrate specificity of plant nitrilase complexes is affected by their helical twist," *Communications Biology*, vol. 1, no. 1, p. 186–187, Nov. 2018, doi: 10.1038/s42003-018-0186-4.

[66] Y. Ding *et al.*, "Mechanism and Structural Insights Into a Novel Esterase, E53, Isolated From Erythrobacter longus," *Frontiers in Microbiology*, vol. 12, 2022, doi: 10.3389/fmicb.2021.798194.

[67] H. Ludewig *et al.*, "Halogenases: structures and functions," *Current Opinion in Structural Biology*, vol. 65, pp. 51–60, 2020, doi: https://doi.org/10.1016/j.sbi.2020.05.012.

[68] C. Breton, L. Šnajdrová, C. Jeanneau, J. Koča, and A. Imberty, "Structures and mechanisms of glycosyltransferases," *Glycobiology*, vol. 16, no. 2, p. 29–30, 2005, doi: 10.1093/glycob/cwj016.

[69] J. Liang *et al.*, "Development of a Biocatalytic Process as an Alternative to the (−)-DIP-Cl-Mediated Asymmetric Reduction of a Key Intermediate of Montelukast," *Organic Process Research & Development*, vol. 14, no. 1, pp. 193–198, 2010, doi: 10.1021/op900272d.

[70] C. K. Savile *et al.*, "Biocatalytic Asymmetric Synthesis of Chiral Amines from Ketones Applied to Sitagliptin Manufacture," *Science*, vol. 329, no. 5989, pp. 305–309, 2010, doi: 10.1126/science.1188934.

[71] A. A. Desai, "Sitagliptin Manufacture: A Compelling Tale of Green Chemistry, Process Intensification, and Industrial Asymmetric Catalysis," *Angewandte Chemie International Edition*, vol. 50, no. 9, pp. 1974–1976, Feb. 2011, doi: 10.1002/anie.201007051.

[72] S. Hu, C. A. Martinez, J. Tao, W. E. Tully, P. Kelleher, and Y. Dumond, "Preparation of pregabalin and related compounds," Mar. 2012 [Online]. Available: https://ppubs.uspto.gov/dirsearch-public/print/downloadPdf/8134023