Squidly: Enzyme Catalytic Residue Prediction Harnessing a Biology-Informed Contrastive Learning Framework

William JF Rieger[1], Mikael Boden[1], Frances Arnold[2], Ariane Mora[2]*

[1]School of Chemistry and Molecular Biology, University of Queensland, Cooper Road, 4072, Queensland, Australia
[2]Division of Chemistry and Chemical Engineering, California Institute of Technology, California Blvd., 91125, California, USA

*Corresponding author: amora@caltech.edu

Abstract
Enzymes present a sustainable alternative to traditional chemical industries, drug synthesis, and bioremediation applications. Because catalytic residues are the key amino acids that drive enzyme function, their accurate prediction facilitates enzyme function prediction. Sequence similarity-based approaches such as BLAST are fast but require previously annotated homologs. Machine learning approaches aim to overcome this limitation; however, current gold-standard machine learning (ML)-based methods require high-quality 3D structures limiting their application to large datasets. To address these challenges, we developed Squidly, a sequence-only tool that leverages contrastive representation learning with a biology-informed, rationally designed pairing scheme to distinguish catalytic from non-catalytic residues using per-token Protein Language Model embeddings. Squidly surpasses state-of-the-art ML annotation methods in catalytic residue prediction while remaining sufficiently fast to enable wide-scale screening of databases. We ensemble Squidly with BLAST to provide an efficient tool that annotates catalytic residues with high precision and recall for both in- and out-of-distribution sequences.

Introduction

Enzymes are efficient, non-toxic and sustainable alternatives to traditional chemical catalysts in applications such as bio-remediation, pharmaceutical and biofuel production[1]. Catalytic residues refer to the amino acids in the protein sequence directly involved in the mechanism of action. Catalytic residues are involved both directly and indirectly in catalysis; for example, they can stabilize transition states [2], act as direct electron donors, or interact with secondary residues, water molecules, or cofactors that are then directly involved in the catalytic mechanism. As catalytic activity depends on specific structural conformations and sequence context, identifying catalytic residues remains a challenge.

Catalytic residue annotations can be used to infer protein function prediction and enable insights into mechanisms of action and evolutionary relationships [3–5]. Annotations are used to facilitate enzyme engineering methods such as directed evolution and rational design [6], and can help construct the theoretical catalytic sites known as theozymes that are used to condition *de novo* enzyme design [7]. Experimental methods to determine catalytic residues are time-consuming, labour-intensive, and costly, as they often require experimentally determined enzyme-substrate structures. To expedite annotation, computational approaches have been developed to predict catalytic residues from protein sequences or predicted structures [8–11], using annotations in databases such as UniProt [12] and M-CSA, a manually curated enzyme mechanism database [13]. Existing computational methods can be broadly grouped into three categories: 1) similarity-based methods and sequence alignment, 2) structural methods, or 3) statistical and machine learning approaches.

Similarity-based approaches are effective when homologous sequences have been previously annotated.

Structure-based approaches use a variety of geometric search algorithms to find potential binding pockets and then map sequence conservation scores to these regions to predict catalytic residues with higher specificity [14–16]. Geometric methods improve the accuracy of approaches reliant on homology, while the limitation of requiring homologous sequences remains. Data-driven approaches such as statistical models or machine learning algorithms offer a more flexible framework [9,11,17] but are restricted by the quality and diversity of datasets available for training. The current machine learning (ML) state-of-the-art (SOTA) tools apply multimodal deep learning strategies and often emphasise the value of the structural modality. Shen et al. [10] combined an adaptive edge-gated graph attention neural network (AEGAN) with graphs constructed using the 3D structure and sequence-derived positional-specific scoring matrices. SCREEN leverages protein sequence and structure by combining graphical convolutional neural networks and contrastive learning on Protein Language Models (PLMs) [9]. EasIFA, developed by Wang et al. [11], predicts both catalytic residues and binding sites by incorporating graph attention representations of chemical reactions with PLM information and structure representations. Despite the reported improvements on sequence based methods by multi-modal approaches, current benchmarks inherently obscure the reliance on structural prediction as they often contain sequences with known structures, which are more likely to be predicted accurately by tools such as AlphaFold [18]. This bias may limit the evaluation of generalisability to unseen data, i.e. enzymes without experimentally derived structures. Furthermore computing structures remains computationally intensive.

In comparison to structural models PLMs learn the distribution of amino acids across protein sequences. PLMs embed biologically meaningful representations that capture complex dependencies between residues within a protein sequence [19–21]. These models are trained in an unsupervised manner on large numbers (billions) of diverse sequences [22,23]. Remote homology search and alignment algorithms based on PLM embeddings have been reported to outperform sequence similarity-based methods at low sequence identities [24,25]. In enzyme function prediction, models highlight the utility of PLM embeddings in combination with contrastive learning, to achieve SOTA ML performance in the prediction of enzyme function (by proxy of enzyme classification numbers) and associated reactions catalysed [26–28]; however, similarity-based approaches (e.g. BLAST or reaction distance) perform comparably [28]. We hypothesised that using PLM embeddings to predict catalytic residues would complement sequence similarity approaches in low homology settings.

Contrastive learning is a discriminative modelling approach that facilitates learning by distinguishing between paired observations in data that are “similar” or “different”, thus enabling information to be extracted from dense PLM embeddings [29]. Data engineering, such as choosing a dataset to enhance specific relationships [30], enables domain expertise to inform pair schemes for training. For example, it is common to explicitly incorporate and maximise “hard negatives” (samples that are dissimilar but close in the latent space) to refine the model’s ability to discriminate subtle differences in the feature space [31–33].

Although pair schemes can be generated automatically in contrastive learning frameworks [30], we posited that the physical principles of enzyme functions could be used to maximise hard negatives and create an accurate catalytic residue prediction algorithm, as in ref [28]. Enzyme function can be grouped into different classes using the domain standard Enzyme Class (EC) ontology, which classifies enzymes into four levels based on the reactions they catalyse. The first level represents the general type of reaction (e.g., oxidoreductases), while subsequent levels add specificity, capturing information about substrates, cofactors, and products [34]. This hierarchical structure provides as basis for relationships between enzyme functions and mechanisms that can be used to create biologically meaningful pairs.

To overcome the compute limitations with structural methods and existing benchmarks, we developed

Squidly and a new benchmark for low sequence identity catalytic residue predictions. We combined a contrastive learning framework on PLM per-token embeddings with a rationally designed hierarchical pair scheme to create a sequence-based catalytic residue predictor that is accurate, scalable, and able to generalise in the absence of sequence similarity. Squidly exceeds the performance of both BLAST and SOTA ML tools in existing benchmarks achieving a F1 of greater than 0.85 across enzyme families, in addition to generalising to low sequence identity enzymes, with an F1 of 0.64 on sequences with less than 30% identity. Squidly is over 50x faster than folding a structure, enabling it to be used to annotate existing databases, in metagenomics pipelines, and to filter *de novo* designed proteins. Finally, owing to the added interpretability of sequence similarity-based approaches we provide Squidly as an ensemble with BLAST, and suggest using Squidly where the sequence identity is less than 50%. We envision Squidly will be broadly applicable in the enzyme engineering field and will complement existing sequence homology and structure-based methods.

Methods

AEGAN training and test datasets

To ensure comparability with the SOTA tools for catalytic residue prediction, the same training, test, and benchmark datasets (Uni14230, Uni3750) were utilised as described in prior work [10]. These were originally sourced from UniProt and M-CSA databases. Uni14230 and Uni3750 datasets also include computational and manually curated predictions from SwissProt. To reduce redundancy, the data was filtered to retain only sequences with less than 60% sequence identity, resulting in 8,784 sequences in the training set and 1,955 sequences in the test set. Additionally, Squidly was evaluated against six previously reported benchmark datasets [35–38]. The EF family, superfamily, fold, and HA superfamily datasets were specifically designed to include one representative sequence from each enzyme family, fold, or superfamily present in existing databases, and are thus not designed to predict "novel" active sites [35,36]. The NN and PC datasets were both constructed to be structurally and functionally heterogeneous based on SCOP and CATH classifications [37,38]. These datasets were generated prior to 2007 and are therefore limited in their catalytic diversity compared to the current databases. Both AEGAN and SCREEN were originally benchmarked against distinct subsets of these datasets, however, we found that they contained sequences with high identity to their respective training data, as detailed in Supplementary Table 2. Squidly's reported performance with these datasets is based on the Uni14230 training set to ensure a fair comparison with AEGAN and given the lower overall sequence similarity between the training and test sets.

New benchmark: CataloDB

To address the shortcomings of existing benchmark datasets for the evaluation of catalytic residue predictions, we introduce a benchmarking dataset named CataloDB. CataloDB collects the annotation data available for enzymes in the SwissProt database. The total set of enzymes includes experimentally determined annotations in addition to reviewed sequences with known structures, see Appendix A.1 for more details. Sequences with greater than 90% sequence similarity, based on shared overlap, were removed using mmSeqs2 [39]. Structural and sequence-based clustering was performed on the filtered dataset to create a test set of 232 sequences with less than 30% sequence and structural identity to the training set (the remaining 5357 sequences). Structural identity was calculated using FoldSeek [40]. Sequences that were in any of the six commonly used benchmarks were omitted from the training data to ensure compatibility between CataloDB and existing benchmarks. Additional detailed descriptions of the data are available in the Appendix A.1, and see Supplementary Figures S1-S6 for distribution of amino acids and EC classes in the benchmark dataset.

To enable direct comparison with the SOTA tool SCREEN, we retrained a lightweight version of the

model using the CataloDB benchmark training data. This version of SCREEN, provided by the original authors, omits costly evolutionary features which are cumbersome for larger datasets. The authors report that these features have minimal impact on performance [9]. Minor changes were made to the source code to improve model training stability. Notably, we allowed optimal stopping to occur after 5 epochs of training, rather than after 200 epochs in the original source code as we observed exploding gradients at later epochs when trained on CataloDB.

ESM2 embeddings

Each sequence was encoded by either the 3 billion parameter or 15 billion ESM2 model [23]. Pre-trained model weights were used without fine-tuning and per-token embeddings were extracted from the final layer of the encoder. These embeddings contain a vector of length 2560 (3B) or 5120 (15B) for each amino acid in the sequence. No normalization was applied to the embeddings before downstream use.

Contrastive model

The supervised contrastive model is a network that takes per-residue embeddings as input (2560 or 5120 length vectors) and is trained to output a new embedding (N=128) of the residue. The model contains an initial dropout layer connected to the inputs, with a dropout rate of 10%, and two subsequent hidden layers of 1280 and 640 size each. During training, batches of 16,000 pairs are passed through the model and the distance between the output representations of each pair is calculated. The following cosine embedding loss function implemented by PyTorch is used to calculate the loss based on the distances:

$$loss(x, y) = \begin{cases} 1 - \cos(x_1, x_2), & if\ y = 1 \\ max(0, \cos(x_1, x_2) - margin), & if\ y = -1 \end{cases}$$

Where x_1 and x_2 are input vectors and y is the label, with $y = 1$ indicating a positive pair and $y = -1$ indicating a negative pair. When $y = 1$, the loss function rewards the model for maximising the cosine similarity, $\cos(x_1, x_2)$. The loss function minimises the cosine similarity when $y = -1$. Importantly, the pair selection scheme optimises for hard-negative pairs, and thus the margin is set equal to 0 when defining the loss function.

Reaction informed pair-mining

Positive and negative pairs were created to enable contrastive comparisons where in the simplest form positives are pairs of exclusively catalytic or exclusively non-catalytic residues while negatives include a catalytic and a non-catalytic residue. As all possible pair combinations of amino acids becomes intractable we are required to reduce this to make training feasible, hence we test three methods to subsample from this pool of possible pairs. Specifically, we used information about the EC number of the sequence and amino acid characters to refine the pair selection process. For further information, see Appendix A.2.

Three pair schemes were created to test for the effect of the pair scheme, herein referred to as "reaction-informed pair-mining". See Figure 1 for a visual description.

Scheme 1 samples pairs across all sequences and residues such that positive pairs are pairs of residues that are either both catalytic or both non-catalytic.

Scheme 2 samples positive and negative pairs much like in scheme 1, except that each pair of residues must be the same amino acid and must have come from a sequence which has the same EC number.

Scheme 3 performs the same pair sampling as in scheme 2; however, scheme 3 also includes additional negative pairs that are either exclusively catalytic residues or exclusively non-catalytic and come from different EC numbers or amino acids.

To combat the inherent bias in the data towards certain EC numbers, sequences were grouped by EC

number at the 2nd level, and an upper limit of 600 catalytic residues and 600 non-catalytic residues were sampled from each EC number. An equal number of positive and negative pairs were generated for each EC number. A parameter called the sample limit was set to limit the total number of pairs made for each group. The sample limit chosen was 16,000, providing a total of 8-10 million pairs for schemes 2 and 3. Scheme 1 was then trained with an equal total number of pairs.

LSTM classifier

All LSTM classification models were trained using the same parameters. The models are bidirectional LSTM networks designed for binary classification of catalytic versus non-catalytic residues in protein sequences. The model takes as input 128-dimensional per-residue embeddings (from the contrastive model) that are processed by a 2-layer LSTM (hidden size: 128) with a dropout rate of 0.2, followed by a linear layer mapping the output to a single logit score, see Figure 2. Class imbalance is addressed by weighting the misclassification of catalytic residues 100 times higher during loss computation. The models were trained using a 90:10 train-validation split, with a batch size of 400 sequences, a learning rate of 0.01, and early stopping of 5 epochs was used. These parameters were not optimised. See Figure 2 for a depiction of the model architecture.

Squidly ensemble and uncertainty quantification

For each training dataset, five models were ensembled by taking the per-residue output from the LSTM and then combining these to compute the mean score, along with the predictive entropy for each residue [41]. Users can specify a threshold for classifying a residue as catalytic (between 0 to 1.0): if the mean score exceeds this threshold, the residue is designated as "catalytic". Users can optionally use the uncertainty to further filter for confidence in predictions.

BLAST similarity-based search

BLAST is a local alignment search tool able to identify homologous sequences from a reference database [42]. For our study, we utilised diamond BLAST to search the training set for similar sequences to each test set [43]. The ultra-sensitive flag was used to identify sequences with low sequence similarities. Upon retrieval of results, the sequence with the highest sequence similarity was retained. This sequence was then aligned against the query using the alignment tool ClustalOmega [44]. The gapped alignment was used to infer the catalytic residues of the query sequence.

Results

ESM2 per-token embeddings contain catalytic residue specific variation

To illustrate the information content of per-residue embeddings for catalytic residue differentiation, we performed principal component analysis (PCA) on equally sampled catalytic and non-catalytic residues from EC 3.1 and 2.7, Figure 3. For EC 3.1, the first two principal components explained 5.24% and 1.86% of the variance for all residues, respectively. Although the explained variance in PCs 1 and 2 is minimal, the separation in PCA space indicates that the largest two signatures of linear variation are related to catalytic roles, with visible clusters of catalytic (red-coloured) and non-catalytic (blue-coloured) residues, Figure 3. The separability from a linear transformation suggests that ESM2 per-residue embeddings could be used to capture catalytic-specific signals.

Reaction informed pair-mining improves upon contrastive learning and state of the art prediction performance

We hypothesised that contrastive learning would present an effective approach to leverage the rich feature space of ESM2 embeddings to distinguish between catalytic and non-catalytic roles. We implemented a biologically informed hierarchical contrastive learning pair selection framework to select pairs which are

informative at the per-residue level. Three datasets were created to compare the performance of different pair scheme selection criteria (see Methods for details).

Squidly performance on the Uni3175 dataset

Scheme 1, see Figure 1, is a "control" scheme and uses a standard contrastive loss function with random pair selection. We observed that scheme 1 performs poorly overall on the Uni3175 dataset, Figure 4. Across all datasets, Scheme 1 achieved relatively low F1 scores, with a mean F1 score of 0.49 and 0.43 for the 3B and 15B ESM2 models. Scheme 1 failed to surpass the LSTM classification model using unchanged ESM2 embeddings, which had a mean F1 score of 0.73, Figure 4. Schemes 2 and 3, both of which employ reaction-informed pair-mining, exhibited improved performance relative to both Scheme 1 and the ESM2 baseline. Scheme 2 achieved mean F1 scores of 0.80 and 0.82 using the 3B and 15B ESM2 models, respectively. Scheme 3 improves upon scheme 2 by contrasting catalytic and non-catalytic sites based on distinct EC and amino acid labels, Figure 1. This resulted in mean F1 scores of 0.79 and 0.84. The best performing model had an F1 score of 0.86 achieved using Scheme 3 and the 15B ESM2 model.

The results from Uni3175 benchmark show that Squidly improved marginally upon the performance of previous work and the current SOTA ML prediction model AEGAN by Shen et al. [10] Squidly uses only sequence through ESM2 embeddings for prediction, which, compared to the homology and structural modalities included in AEGAN, makes for a more flexible predictor without compromising performance. Overall, the results on the Uni3175 dataset indicate that the rationally informed pair-mining strategy is an effective method for discriminative tasks using per-token ESM2 embeddings. It also indicates that sequence-based models using PLM embeddings can compete with models using structural modalities. See Methods for details on each scheme design.

Squidly outperforms other ML-based methods yet works best in conjunction with sequence similarity methods

Squidly was evaluated against six previously reported benchmark datasets, namely the EF fold, family, and superfamily benchmarks, as well as the HA superfamily, NN, and PC benchmarks, see Supplementary Table 1 and Appendix 3 for description of each dataset, see Figure 5 for sequence identity between test and train sets. Squidly's performance exceeded those of SOTA models AEGAN and SCREEN, Table 1. When compared to the baseline of BLAST, we find that BLAST outperforms all other ML models except Squidly, likely due to the homologous sequences in the test and train sets, see Appendix A3, and Supplementary Table 2 for details.

We hypothesised that BLAST would work best in instances where a similar sequence existed while Squidly would excel in low sequence similarity situations. To test for the optimal conditions for integration we varied the rate at which either predictor was used, Figure 6. For sequences with low sequence identity (<50%), using Squidly improves the F1 score. However, Squidly has a higher likelihood than BLAST to record false positives where similar sequences exist while BLAST is more precise. The cutoff of 50% sequence identity was used to compare the performance of the ensemble method to existing tools. On all datasets using this cutoff an ensemble of BLAST and Squidly performs SOTA, Table 1.

Squidly is scalable and fast

Squidly's sequence-based approach reduces the overall computational cost of prediction to enable the screening of large databases. SOTA ML models rely on accurate structures as input, and thus structural prediction must be done prior to inference when screening databases such as metagenomic samples. Squidly's smaller 3B model can be run locally and can predict roughly 6 sequences a second in our tests using an NVIDIA H100 GPU. In comparison to the current widely used structural generation tool Chai [45],

Squidly is approximately 50-fold faster (e.g. N=108, Squidly=108s, Chai=5757s). Note this under-represents the time for the structural tools to run, as they also require predictions which also add time, Table 2.

Squidly generalises to low identity sequences

To evaluate the capacity of Squidly to generalise to low identity sequences, we developed a benchmark dataset, CataloDB with low structural and sequence identity, see Methods and Figure 7.A, B for similarities between the test and train sets. This benchmark serves as an alternative test set that takes structural and sequence similarity into account, allowing for fair comparison between future structural and sequence-based tools and testing in low identity settings. Squidly 3B and 15B models showed similar performance on the benchmark with most sequences (N=148/232, N=154/232), recording at least one correct catalytic residue. For comparison, BLAST identifies only 68 sequences that have a catalytic residue recovered correctly. Both Squidly models performed similarly with F1 scores of (0.64, 0.68), precision of (0.80, 0.82), and recall of (0.53, 0.58) for the 15B and 3B models (respectively), while BLAST records an F1 of 0.37, recall of 0.24, and precision of 0.72, see Figure 8. We also retrain SCREEN, with performances recorded above BLAST however still below Squidly, Figure 7.C, see Methods for details. Reflecting the bias of public data, CataloDB has a bias towards well-represented EC numbers and amino acids. For example, EC 6 was represented by only three sequences in the test data; and EC 7 had no representation. Overall, these results suggest that Squidly can generalise to low identity settings and be used in predictions where traditional homology transfer is not possible. However, caution should be taken when applied to poorly represented enzyme classes such as EC 6 and 7.

Discussion

Rationally constraining pair generation in contrastive learning can improve model performance on biological data. Using publicly available EC number annotations and amino acid sequences, we categorised pairs of catalytic residues into distinct groups enriched with hard negatives to enable sequence-only prediction of catalytic residues. This approach led to substantial improvements compared to a standard LSTM classifier and typical pair-mining strategies. Squidly significantly outperformed other ML-based sequence methods and showed marginal improvement to SOTA ML-based catalytic residue prediction models that use structural data while running in a fraction of the time. Furthermore, Squidly performs comparably to BLAST when homologous sequences exist, yet also shows good performance in low-identity settings, where BLAST is unable to identify any similar sequences. Given the complementarity of BLAST and Squidly, we provide Squidly as an ensemble with sequence similarity, leveraging the interpretability of sequence similarity-based methods where homologous sequences exist, and Squidly, in out-of-distribution settings. The focus of this work was catalytic site prediction; however, of similar utility are combined binding and catalytic prediction models, such as EasIFA [11], and therefore future work should additionally consider binding sites.

A limitation of this study is the inherent bias in publicly available data towards well studied enzyme classes. The bias likely confounds the perceived performance on less studied mechanisms, and in these instances the results should be considered with caution, for example, in EC classes 6 and 7. A further challenge is the lack of a consolidated benchmark for catalytic residue prediction, making direct and fair comparison between tools difficult. Notably, the multiclass classification objective and benchmarks used to evaluate EasIFA [11] made it infeasible to compare performance for the binary catalytic residue prediction task. Existing benchmark datasets do not capture the diversity of enzyme catalysis. Although we have provided a new benchmark that improves upon previous datasets by using structural identity filtering, the core issues remain. CataloDB and current benchmarks are small and include entire enzyme families. While low sequence and structural similarity samples are held out from training, they may not accurately

represent the generalisability of these methods in out-of-distribution settings. Furthermore, the datasets often consist of sequences with known structures in the PDB, many of which were likely part of the AlphaFold training set or structurally similar clusters represented in the set. Tools also evaluate their performance using known, rather than predicted structures [10]. Consequently, SOTA tools that rely on AlphaFold predicted structures tend to perform better in these benchmarks but are likely to underperform in practical applications involving truly unique and uncharacterised sequences. This issue also applies, albeit to a lesser extent, to the method proposed in this study, which utilises ESM2 representations. While ESM2's training set is more diverse and comprehensive, it may not generalise to distant out-of-distribution sequences compared to the well studied sequences used in the benchmarks.

Squidly and other data-driven approaches to catalytic residue prediction should be used with caution when applied to retrieve understudied catalytic mechanisms. Currently, there exists a limited set of experimentally validated annotations [13]. Datasets such as Uni14230 and CataloDB used for training in this study include SwissProt sequences with reviewed computational annotations to increase the size of the training data. However, this does not improve the diversity of catalytic mechanisms and enzyme sequences, as these methods reflect the inherent biases in databases towards well represented EC classes. Squidly, having learnt from this biased data, may indeed replicate this bias during inference and is therefore less likely to provide insights into novel or understudied catalytic mechanisms. Despite the limited data for catalytic residue prediction, we show that an ensemble approach of machine learning and sequence-similarity-based methods performs SOTA across multiple benchmark sets. By leveraging contrastive learning, we find that ML sequence-based predictions can generalize to low sequence/structural similarity settings. Finally, the methods reported in this paper that leverage rational data engineering for contrastive learning have broad implications for learning across biological data. Biological sequences are generated through evolutionary processes occurring over billions of years, resulting in hierarchically structured data that has significant variance between evolutionarily distant sequences. Moreover, mechanisms in biochemistry are often similar for related chemical reactions. By leveraging ontologies such as EC numbers, pairs can be selected rationally to improve training on proteins that are not only evolutionarily related but also share broader biochemical functions. A wide range of alternative ontologies exist that can capture different structures within biological data, suggesting that this approach is likely to be applicable across other biological classification tasks, particularly in studies utilising contrastive frameworks for per-token in-sequence classification [46]. Future studies should explore and compare additional ontologies to better understand the utility of such data structures.

Code and data availability

All source code is available at Github (https://github.com/WRiegs/Squidly). All associated data including CataloDB is available at Zenodo (https://doi.org/10.5281/zenodo.15541320).

Competing interests

No competing interest is declared.

Author contributions statement

W.R. and A.M. conceived the experiment(s), W.R. designed and implemented the model, W.R. and A.M. ran the experiment(s), analysed the results, and wrote the manuscript. M.B. and F.H.A reviewed the manuscript and provided input to the research.

Acknowledgments

A.M. is supported by the Schmidt Science Fellows, in partnership with the Rhodes Trust. W.R. is supported by the Australian Government Research Training Program (RTP) Scholarship. The authors

thank Charlie Trimble for his generous contributions to the project.

References

(1) Reisenbauer, J. C.; Sicinski, K. M.; Arnold, F. H. Catalyzing the Future: Recent Advances in Chemical Synthesis Using Enzymes. *Current Opinion in Chemical Biology* **2024**, *83*, 102536. https://doi.org/10.1016/j.cbpa.2024.102536.

(2) Ribeiro, A. J. M.; Tyzack, J. D.; Borkakoti, N.; Holliday, G. L.; Thornton, J. M. A Global Analysis of Function and Conservation of Catalytic Residues in Enzymes. *Journal of Biological Chemistry* **2020**, *295* (2), 314–324. https://doi.org/10.1074/jbc.REV119.006289.

(3) Precord, T. W.; Ramesh, S.; Dommaraju, S. R.; Harris, L. A.; Kille, B. L.; Mitchell, D. A. Catalytic Site Proximity Profiling for Functional Unification of Sequence-Diverse Radical S-Adenosylmethionine Enzymes. *ACS Bio & Med Chem Au* **2023**, *3* (3), 240–251. https://doi.org/10.1021/acsbiomedchemau.2c00085.

(4) Martínez-Martínez, M.; Coscolín, C.; Santiago, G.; Chow, J.; Stogios, P. J.; Bargiela, R.; Gertler, C.; Navarro-Fernández, J.; Bollinger, A.; Thies, S.; Méndez-García, C.; Popovic, A.; Brown, G.; Chernikova, T. N.; García-Moyano, A.; Bjerga, G. E. K.; Pérez-García, P.; Hai, T.; Del Pozo, M. V.; Stokke, R.; Steen, I. H.; Cui, H.; Xu, X.; Nocek, B. P.; Alcaide, M.; Distaso, M.; Mesa, V.; Peláez, A. I.; Sánchez, J.; Buchholz, P. C. F.; Pleiss, J.; Fernández-Guerra, A.; Glöckner, F. O.; Golyshina, O. V.; Yakimov, M. M.; Savchenko, A.; Jaeger, K.-E.; Yakunin, A. F.; Streit, W. R.; Golyshin, P. N.; Guallar, V.; Ferrer, M.; The INMARE Consortium. Determinants and Prediction of Esterase Substrate Promiscuity Patterns. *ACS Chem. Biol.* **2018**, *13* (1), 225–234. https://doi.org/10.1021/acschembio.7b00996.

(5) Wang, Z.; Yin, P.; Lee, J. S.; Parasuram, R.; Somarowthu, S.; Ondrechen, M. J. Protein Function Annotation with Structurally Aligned Local Sites of Activity (SALSAs). *BMC Bioinformatics* **2013**, *14* (3), S13. https://doi.org/10.1186/1471-2105-14-S3-S13.

(6) Yu, H.; Ma, S.; Li, Y.; Dalby, P. A. Hot Spots-Making Directed Evolution Easier. *Biotechnology Advances* **2022**, *56*, 107926. https://doi.org/10.1016/j.biotechadv.2022.107926.

(7) Ahern, W.; Yim, J.; Tischer, D.; Salike, S.; Woodbury, S. M.; Kim, D.; Kalvet, I.; Kipnis, Y.; Coventry, B.; Altae-Tran, H. R.; Bauer, M.; Barzilay, R.; Jaakkola, T. S.; Krishna, R.; Baker, D. Atom Level Enzyme Active Site Scaffolding Using RFdiffusion2.

(8) Zhang, T.; Zhang, H.; Chen, K.; Shen, S.; Ruan, J.; Kurgan, L. Accurate Sequence-Based Prediction of Catalytic Residues. *Bioinformatics* **2008**, *24* (20), 2329–2338. https://doi.org/10.1093/bioinformatics/btn433.

(9) Pan, T.; Bi, Y.; Wang, X.; Zhang, Y.; Webb, G. I.; Gasser, R. B.; Kurgan, L.; Song, J. SCREEN: A Graph-Based Contrastive Learning Tool to Infer Catalytic Residues and Assess Enzyme Mutations. *Genomics, Proteomics & Bioinformatics* **2024**, qzae094. https://doi.org/10.1093/gpbjnl/qzae094.

(10) Shen, X.; Zhang, S.; Long, J.; Chen, C.; Wang, M.; Cui, Z.; Chen, B.; Tan, T. A Highly Sensitive Model Based on Graph Neural Networks for Enzyme Key Catalytic Residue Prediction. *Journal of Chemical Information and Modeling* **2023**, *63* (14), 4277–4290. https://doi.org/10.1021/acs.jcim.3c00273.

(11) Wang, X.; Yin, X.; Jiang, D.; Zhao, H.; Wu, Z.; Zhang, O.; Wang, J.; Li, Y.; Deng, Y.; Liu, H.; Luo, P.; Han, Y.; Hou, T.; Yao, X.; Hsieh, C.-Y. Multi-Modal Deep Learning Enables Efficient and Accurate Annotation of Enzymatic Active Sites. *Nat Commun* **2024**, *15* (1), 7348. https://doi.org/10.1038/s41467-024-51511-6.

(12) Consortium, T. U. UniProt: The Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* **2022**, *51* (D1), D523–D531. https://doi.org/10.1093/nar/gkac1052.

(13) Ribeiro, A. J. M.; Holliday, G. L.; Furnham, N.; Tyzack, J. D.; Ferris, K.; Thornton, J. M. Mechanism and Catalytic Site Atlas (M-CSA): A Database of Enzyme Reaction Mechanisms and

Active Sites. *Nucleic Acids Research* **2018**, *46* (D1), D618–D623. https://doi.org/10.1093/nar/gkx1012.

(14) Glaser, F.; Pupko, T.; Paz, I.; Bell, R. E.; Bechor-Shental, D.; Martz, E.; Ben-Tal, N. ConSurf: Identification of Functional Regions in Proteins by Surface-Mapping of Phylogenetic Information. *Bioinformatics* **2003**, *19* (1), 163–164. https://doi.org/10.1093/bioinformatics/19.1.163.

(15) Mayrose, I.; Graur, D.; Ben-Tal, N.; Pupko, T. Comparison of Site-Specific Rate-Inference Methods for Protein Sequences: Empirical Bayesian Methods Are Superior. *Molecular Biology and Evolution* **2004**, *21* (9), 1781–1791. https://doi.org/10.1093/molbev/msh194.

(16) del Sol Mesa, A.; Pazos, F.; Valencia, A. Automatic Methods for Predicting Functionally Important Residues. *Journal of Molecular Biology* **2003**, *326* (4), 1289–1302. https://doi.org/10.1016/S0022-2836(02)01451-1.

(17) Song, J.; Li, F.; Takemoto, K.; Haffari, G.; Akutsu, T.; Chou, K.-C.; Webb, G. I. PREvaIL, an Integrative Approach for Inferring Catalytic Residues Using Sequence, Structural, and Network Features in a Machine-Learning Framework. *Journal of Theoretical Biology* **2018**, *443*, 125–137. https://doi.org/10.1016/j.jtbi.2018.01.023.

(18) Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; others. Highly Accurate Protein Structure Prediction with AlphaFold. *nature* **2021**, *596* (7873), 583–589.

(19) Tule, S.; Foley, G.; Boden, M. Do Protein Language Models Learn Phylogeny? bioRxiv September 25, 2024, p 2024.09.23.614642. https://doi.org/10.1101/2024.09.23.614642.

(20) Rao, R.; Meier, J.; Sercu, T.; Ovchinnikov, S.; Rives, A. Transformer Protein Language Models Are Unsupervised Structure Learners. *bioRxiv* **2020**, 2020.12.15.422761. https://doi.org/10.1101/2020.12.15.422761.

(21) Vig, J.; Madani, A.; Varshney, L. R.; Xiong, C.; Socher, R.; Rajani, N. F. BERTology Meets Biology: Interpreting Attention in Protein Language Models, 2021. https://arxiv.org/abs/2006.15222.

(22) Heinzinger, M.; Weissenow, K.; Sanchez, J. G.; Henkel, A.; Mirdita, M.; Steinegger, M.; Rost, B. Bilingual Language Model for Protein Sequence and Structure. *bioRxiv* **2024**, 2023.07.23.550085. https://doi.org/10.1101/2023.07.23.550085.

(23) Lin, Z.; Akin, H.; Rao, R.; Hie, B.; Zhu, Z.; Lu, W.; Smetanin, N.; Verkuil, R.; Kabeli, O.; Shmueli, Y.; dos Santos Costa, A.; Fazel-Zarandi, M.; Sercu, T.; Candido, S.; Rives, A. Evolutionary-Scale Prediction of Atomic-Level Protein Structure with a Language Model. *Science* **2023**, *379* (6637), 1123–1130. https://doi.org/10.1126/science.ade2574.

(24) Kulikova, A. V.; Parker, J. K.; Davies, B. W.; Wilke, C. O. Semantic Search Using Protein Large Language Models Detects Class II Microcins in Bacterial Genomes. *bioRxiv* **2023**, 2023.11.15.567263. https://doi.org/10.1101/2023.11.15.567263.

(25) McWhite, C. D.; Armour-Garb, I.; Singh, M. Leveraging Protein Language Models for Accurate Multiple Sequence Alignments. *Genome Research* **2023**, *33* (7), 1145–1153. https://doi.org/10.1101/gr.277675.123.

(26) Mikhael, P. G.; Chinn, I.; Barzilay, R. CLIPZyme: Reaction-Conditioned Virtual Screening of Enzymes. arXiv February 9, 2024. https://doi.org/10.48550/arXiv.2402.06748.

(27) Yu, T.; Cui, H.; Li, J. C.; Luo, Y.; Jiang, G.; Zhao, H. Enzyme Function Prediction Using Contrastive Learning. *Science* **2023**, *379* (6639), 1358–1363. https://doi.org/10.1126/science.adf2465.

(28) Yang, J.; Mora, A.; Liu, S.; Wittmann, B. J.; Anandkumar, A.; Arnold, F. H.; Yue, Y. CARE: A Benchmark Suite for the Classification and Retrieval of Enzymes. arXiv January 6, 2025. https://doi.org/10.48550/arXiv.2406.15669.

(29) Chopra, S.; Hadsell, R.; LeCun, Y. Learning a Similarity Metric Discriminatively, with Application

to Face Verification. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*; IEEE, 2005; Vol. 1, pp 539–546.

(30) Le-Khac, P. H.; Healy, G.; Smeaton, A. F. Contrastive Representation Learning: A Framework and Review. *IEEE Access* **2020**, *8*, 193907–193934. https://doi.org/10.1109/ACCESS.2020.3031549.

(31) Kalantidis, Y.; Sariyildiz, M. B.; Pion, N.; Weinzaepfel, P.; Larlus, D. Hard Negative Mixing for Contrastive Learning. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H., Eds.; Curran Associates, Inc., 2020; Vol. 33, pp 21798–21809.

(32) Khosla, P.; Teterwak, P.; Wang, C.; Sarna, A.; Tian, Y.; Isola, P.; Maschinot, A.; Liu, C.; Krishnan, D. Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2020; Vol. 33, pp 18661–18673.

(33) Lin, N.; Qin, G.; Wang, J.; Yang, A.; Zhou, D. An Effective Deployment of Contrastive Learning in Multi-Label Text Classification. *arXiv e-prints* **2022**, arXiv:2212.00552. https://doi.org/10.48550/arXiv.2212.00552.

(34) McDonald, A. G.; Tipton, K. F. Enzyme Nomenclature and Classification: The State of the Art. *The FEBS Journal* **2023**, *290* (9), 2214–2231. https://doi.org/10.1111/febs.16274.

(35) Youn, E.; Peters, B.; Radivojac, P.; Mooney, S. D. Evaluation of Features for Catalytic Residue Prediction in Novel Folds. *Protein Science* **2007**, *16* (2), 216–226. https://doi.org/10.1110/ps.062523907.

(36) Chea, E.; Livesay, D. R. How Accurate and Statistically Robust Are Catalytic Site Predictions Based on Closeness Centrality? *BMC Bioinformatics* **2007**, *8* (1), 153. https://doi.org/10.1186/1471-2105-8-153.

(37) Bartlett, G. J.; Porter, C. T.; Borkakoti, N.; Thornton, J. M. Analysis of Catalytic Residues in Enzyme Active Sites. *Journal of Molecular Biology* **2002**, *324* (1), 105–121. https://doi.org/10.1016/S0022-2836(02)01036-7.

(38) Petrova, N. V.; Wu, C. H. Prediction of Catalytic Residues Using Support Vector Machine with Selected Protein Sequence and Structural Properties. *BMC Bioinformatics* **2006**, *7* (1), 312. https://doi.org/10.1186/1471-2105-7-312.

(39) Steinegger, M.; Söding, J. MMseqs2 Enables Sensitive Protein Sequence Searching for the Analysis of Massive Data Sets. *Nature Biotechnology* **2017**, *35* (11), 1026–1028. https://doi.org/10.1038/nbt.3988.

(40) van Kempen, M.; Kim, S. S.; Tumescheit, C.; Mirdita, M.; Lee, J.; Gilchrist, C. L. M.; Söding, J.; Steinegger, M. Fast and Accurate Protein Structure Search with Foldseek. *Nat Biotechnol* **2024**, *42* (2), 243–246. https://doi.org/10.1038/s41587-023-01773-0.

(41) Lakshminarayanan, B.; Pritzel, A.; Blundell, C. Simple and Scalable Predictive Uncertainty Estimation Using Deep Ensembles. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc., 2017; Vol. 30.

(42) Altschul, S. F.; Gish, W.; Miller, W.; Myers, E. W.; Lipman, D. J. Basic Local Alignment Search Tool. *J Mol Biol* **1990**, *215* (3), 403–410. https://doi.org/10.1016/S0022-2836(05)80360-2.

(43) Buchfink, B.; Xie, C.; Huson, D. H. Fast and Sensitive Protein Alignment Using DIAMOND. *Nat Methods* **2015**, *12* (1), 59–60. https://doi.org/10.1038/nmeth.3176.

(44) Sievers, F.; Wilm, A.; Dineen, D.; Gibson, T. J.; Karplus, K.; Li, W.; Lopez, R.; McWilliam, H.; Remmert, M.; Söding, J.; Thompson, J. D.; Higgins, D. G. Fast, Scalable Generation of High-quality Protein Multiple Sequence Alignments Using Clustal Omega. *Molecular Systems Biology* **2011**, *7* (1), 539. https://doi.org/10.1038/msb.2011.75.

(45) Chai Discovery. Chai-1: Decoding the Molecular Interactions of Life. *bioRxiv* **2024**. https://doi.org/10.1101/2024.10.10.615955.

(46) Wang, J.; Liu, Y.; Tian, B. Protein-Small Molecule Binding Site Prediction Based on a Pre-Trained

Protein Language Model with Contrastive Learning. *Journal of Cheminformatics* **2024**, *16* (1), 125. https://doi.org/10.1186/s13321-024-00920-2.

Tables:

Table 1. F1 scores of tools on 6 common benchmark datasets.

Tool	EF-fold	EF-superfamily	EF-family	HA-superfamily	NN	PC
PREvaIL [17]	0.2579	0.2575	0.2585	0.2483	0.2537	0.2582
CRpred [8]	0.2613	0.2643	0.2651	0.2628	0.2638	0.2630
AEGAN* [10]	0.8369	0.8687	0.8552	0.8259	0.8177	0.8736
SCREEN* [9]	0.645	0.615	-	0.72	0.738	0.741
Squidly (ESM2 3B)	0.8794	0.8985	0.8876	0.8532	0.8698	0.8791
Squidly (ESM2 15B)	0.8852	0.8927	0.8875	0.8538	0.8705	0.8524
BLAST (ultra-sensitive)	0.8939	0.8814	0.8526	0.8382	0.8324	0.8854
BLAST + Squidly (ESM2 3B)	0.9108	**0.9212**	**0.9049**	0.8786	**0.8961**	**0.9130**
BLAST + Squidly (ESM2 15B)	**0.9161**	0.9140	0.9027	**0.8790**	0.8943	0.8817

Benchmark results for PREvaIL, CRpred and AEGAN were sourced from [10]. Results from Screen were sourced from [9]. BLAST results were performed using the ultra-sensitive flag and aligned to the closest recorded enzyme with an active site in the training dataset (Uni14230). Squidly results shown for models trained on the Uni14230 dataset (same as AEGAN). * Indicates structure-based tools.

Table 2. Resource usage measurements for Squidly models when predicting 1000 sequences with an average length of 401 residues.

Squidly	Wall Clock (s)	VRAM (GB)	RAM (GB)
ESM2 3B	170.76	10.98	44.93
ESM2 15B	653.30	56.91	110.89

Benchmarked using an Nvidia H100 GPU with 80Gb of VRAM.

Figures:

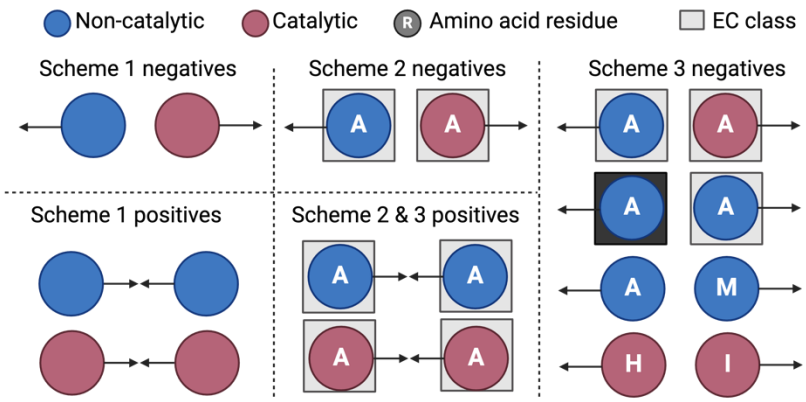

Fig. 1: Representation of the pair-mining schemes depicting the negative and positive classes for each scheme. Scheme 1 is the most permissive and is amino acid and EC *unaware*. Schemes 2 and 3 are more restrictive, requiring both the amino acids and the EC classes to be shared or different in the pair-mining scheme. Scheme 3 included additional negative pairs to separate EC numbers in the latent space.

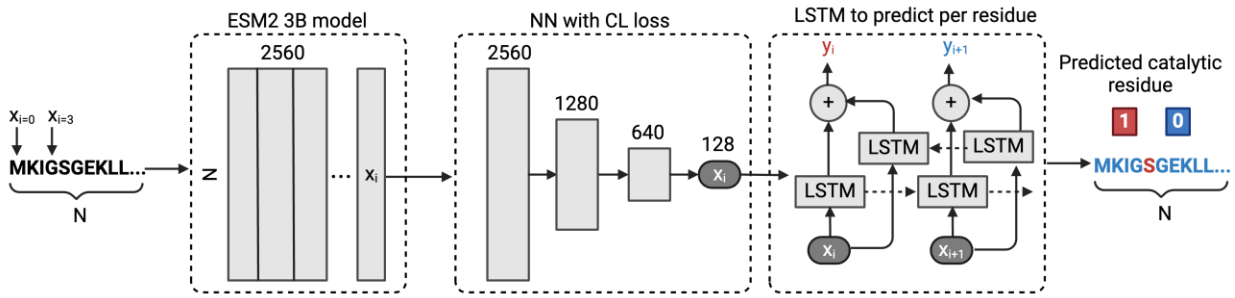

Fig. 2: Overview of the model, depicting a sequence being translated into per-token embeddings via ESM2, each per-token embedding is then translated into a smaller representation space by the MLP trained with contrastive loss (CL). Each residue from the CL model is then predicted as catalytic or not catalytic via a bidirectional LSTM network.

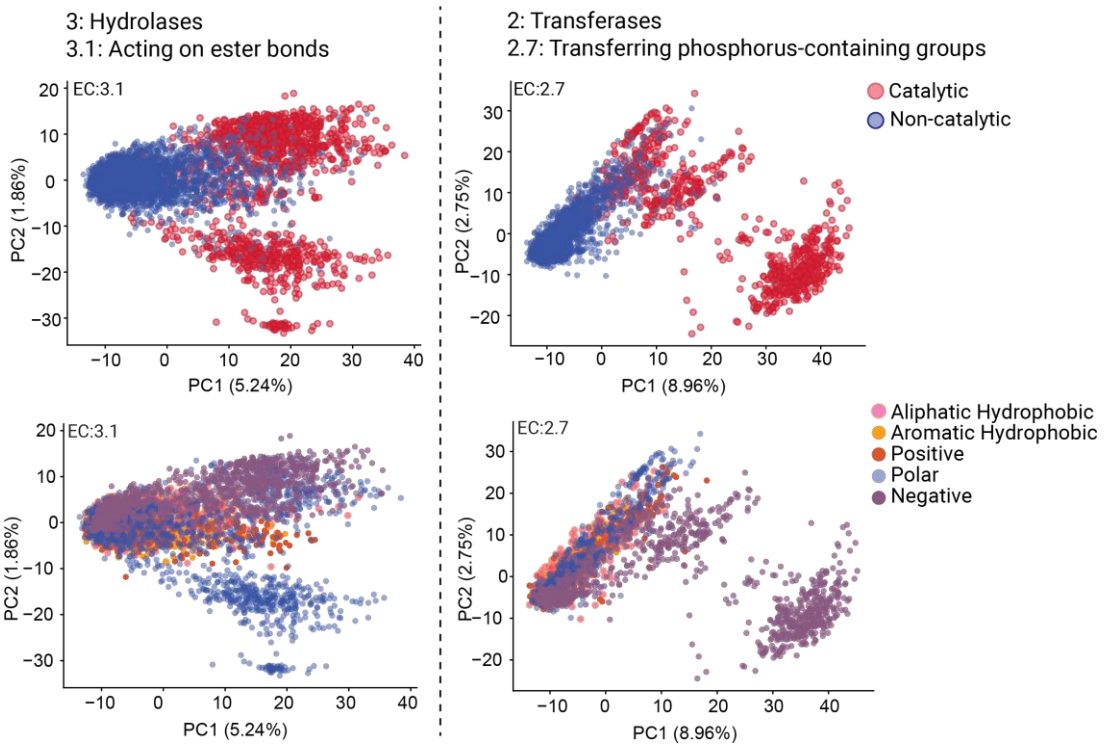

Fig. 3: Principal component analysis of per-residue embeddings from EC numbers 3.1 and 2.7. In the upper row, red colours indicate a catalytic residue, and blue, non-catalytic. In the lower row, residues are coloured by amino acid class which represent their structural and chemical properties. The x and y axes represent the first and second principal components in each subplot. A clear separation of catalytic and non-catalytic residues is seen in the principal components which explain the most variance in our data.

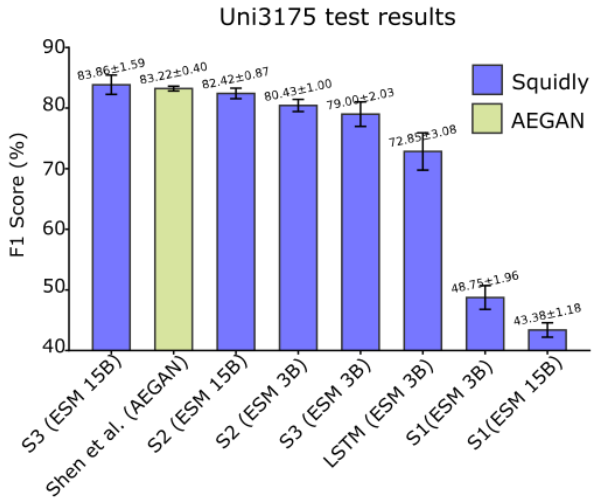

Fig. 4: Comparison of the Squidly contrastive learning framework pair schemes on the Uni3175 dataset. We directly compare the schemes with AEGAN, the tool by Shen et al., the authors of this dataset [10]. Schemes 2 and 3 (S1 and S2) using the largest ESM2 model (15 billion parameters) embeddings slightly improved upon AEGAN's performance. The smaller ESM2 (3 billion parameters) model still performs competitively, at a heavily reduced computation cost. In comparison to the rationally informed pair schemes, the Scheme 1 and raw embedding LSTM models performed poorly.

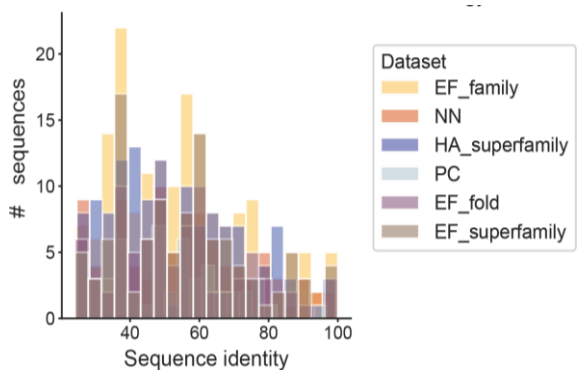

Fig. 5: Sequence similarity based on sequence identity to the closest sequence in the training set between each family.

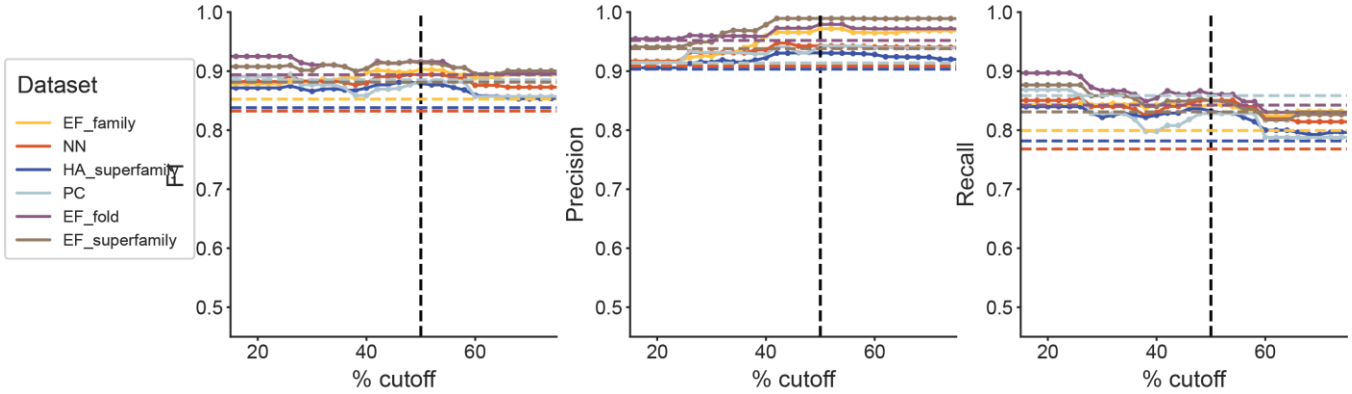

Fig. 6: F1, recall, and precision for the ensemble of Squidly and BLAST with different sequence identity cut-offs. The cut-offs indicate the point at which to transition from a Squidly to a BLAST prediction. At 100% only Squidly is used for predictions, and at 0%, BLAST is used, unless there are no homologous sequences. For sequences with less than 50% sequence identity (dashed black line), Squidly predictions are used. The dashed horizontal line represents the total score for BLAST on the specific dataset, while the dots represent the combined BLAST and Squidly ensemble approach. Squidly reports higher scores for recall, while BLAST is more precise. The performance is best when used as an ensemble.

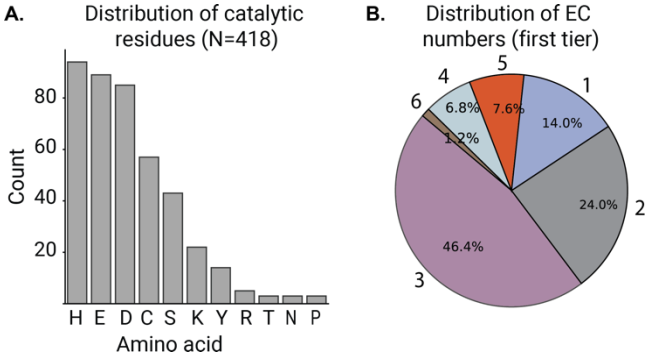

Fig. 7: Distribution of the catalytic residues in the proposed benchmark dataset. A. The distribution of the identity of the catalytic residues is not even and represents the uneven distribution of the catalytic mechanism within the benchmark dataset. B Almost 50% of the data points are in the well characterised class of EC 3, which are hydrolases. Neither EC 7 nor EC 6 is represented, which are translocases and ligases, respectively.

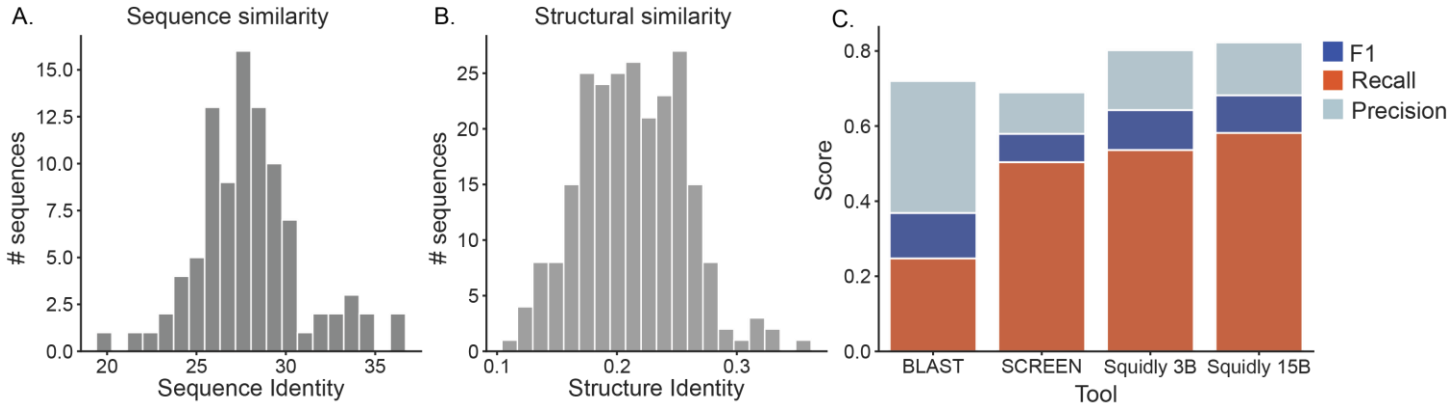

Fig. 8: A. BLAST sequence similarity to the training set for low identity CataloDB test set sequences. B. Structural similarity for the final test set, filtered on both structure and sequence. C. F1, recall, and precision for Squidly, SCREEN and BLAST on CataloDB.

APPENDIX

A.1 Rationale for developing CataloDB

To enable comparisons to existing work, we utilized the AEGAN training and test datasets as described above. However, the limitations identified motivated us to create a new benchmark, CataloDB, for future evaluations. The pre-existing benchmarks were published prior to 2007 and were developed by selecting proteins with active site annotations representing unique protein families, superfamilies, or folds. Their construction lacked standardized metrics for ensuring appropriate similarity separation between training and test data. The continued use of these historical benchmarks likely persists to maintain backward compatibility in comparisons with earlier methods. However, these six benchmarks predominantly represent well-studied folds and families, making it challenging to create properly separated training and test sets.

Our results demonstrate that BLAST already performs comparably or better when predicting catalytic residues in sequences with greater than 0.35 identity. Therefore, we believe machine learning methods should aim to develop tools that generalize effectively to low-identity sequences. CataloDB addresses this challenge, and the concerns by containing test sequences with strictly less than 0.3 sequence and structure identity to the training set. This benchmark is more extensive than previous collections while still allowing for a larger training set with a higher redundancy threshold (90%), enabling machine learning methods to better learn from the underlying data distribution.

Evaluation of Swissprot data for use in CataloDB

To overcome the limited availability of experimentally validated catalytic residue annotations, annotations from sequences which have been reviewed by Swissprot's expert review panel have been included in training and test data in recent machine learning tools [9,10]. However, the assumption that these annotations are valid is not a given. Swissprot provides limited information to clarify how their review process for catalytic mechanisms is conducted [12]. These annotations may have been generated manually or by computational predictions and likely contain the same bias present in the original experimentally validated set. Here we present initial analysis done, using an earlier version of Squidly, to determine whether reviewed sequences improved performance, and thus, the quality of training data.

Three datasets were curated to evaluate the impact of reviewed sequences on training catalytic residue prediction models. Dataset 1 contained 2,209 sequences with only experimentally validated catalytic residue annotations. Dataset 2 included 6,437 sequences, including experimentally validated annotations and sequences with catalytic residues supported by known structures. Dataset 3 included 99,926 sequences comprising all catalytic residue annotations in the SwissProt database. Redundancy reduction removed sequences with over 90% sequence identity. After filtering, Dataset 1 was reduced to 2,030 sequences, Dataset 2 to 5,921 sequences, and Dataset 3 had the highest proportion of redundant sequences with only 57,698 sequences left after filtering. See Table 2 for details.

Contrary to the expectation that increasing data size would improve model performance, we found that increasing the available training data by including reviewed sequences from SwissProt does not improve the overall bias or variance of the available experimental data. Figures S2, S4 and S6 show the EC distributions (EC numbers up to the second tier) across the datasets. All three datasets are biased toward hydrolases (EC 3), followed by classes 2, 1, 4, 5, 6, and 7. The relative abundance of EC class 2 increases between dataset 1, dataset 2, and dataset 3, while other classes remain relatively constant. This trend suggests that the additional sequences in the reviewed datasets likely originate from similarity-based prediction methods that incorporate experimental data, thus maintaining the relative abundance in EC

representation despite increasing dataset size.

Figures S1, S3 and S5 compare the distributions of amino acids acting as catalytic residues across the datasets. The distributions are dominated by histidine, aspartic acid, glutamic acid, cysteine, and serine, in line with the dominant classes being hydrolases. Dataset 2 and dataset 3 show a relative increase in aspartic acid abundance compared to dataset 1. Despite this, and the much larger size of datasets 2 and 3, their amino acid distributions closely resemble dataset 1, indicating that reviewed datasets contribute little diversity in rare catalytic residues, which may correspond to mechanisms that are less studied. Overall, these findings suggest that training the models using the additional data will not greatly improve the performance of the models when generalising to sequences with low identity in the experimental ground truth set. Based on our analyses we use dataset 2 for CataloDB.

A.2 Reaction-informed pair schemes

By performing contrastive learning with pairs, we move from having too few samples to having too many samples. The large pool of amino acid pairs in our dataset varies in how informative each pair is for the model. Therefore, we must find the most effective ways to down sample and select the most informative pairs for our model to learn from. Particularly, we want to maximise the proportion of hard negative pairs in the training data.

The Uni14320 training dataset contains 8,735 non-redundant sequences from Dataset 1 (experimentally validated) with an average sequence length of 407 residues. Given the total pool of $3{,}555{,}145$ residues, a total number of 6.32×10^{12} unique pairwise comparisons can be made, as determined by the binomial coefficient $\binom{n}{2}$. We developed pair schemes 2 and 3 to capture the most informative pairs from this available data. Maximising hard negatives improves the model's ability to discriminate catalytic and non-catalytic residues in a more generalisable way. If two pairs of residue embeddings are inherently very distinct from one another, the model will be unlikely to learn much from them. This is not just because our loss function will penalise the model less during training, but because the relationships extracted from the feature space that are used to separate these two inputs might not necessarily be related to our primary objective.

Reaction informed pair-mining assumes that there must be key biological contexts which create harder negative pairs for the model to differentiate. First, we make the basic assumption that the model will be unable to correctly discriminate between similar amino acids pairs. It is likely that a contrastive model will find pairs individual amino acids, such as histidine, harder to contrast, because the unique structure and properties of each amino acid determine the catalytic or non-catalytic roles in which these amino acids play within enzymes. Therefore, we decided to implicitly include the amino acid character of each catalytic site into the pair sampling scheme.

Another assumption we made was inspired by the use of contrastive learning in another catalytic site prediction tool developed by Tong Pan et al. [9]. Their contrastive model creates EC specific representations of sequences for further use in a convolutional graph neural network classification model. Their contrastive model, however, did not try to use these contexts to better differentiate between catalytic and non-catalytic sites in their model, but as a support to ensure the information about the sequence EC is propagated through their multimodal system. EC numbers separate enzymes based on the reactions they perform. Not only that, but lower-level EC numbers group enzymes by the types of substrate bonds they cleave, or mechanisms by which they catalyse reactions. Inspired by this idea, we leveraged EC numbers to sample hard negative pairs. We specifically chose to represent sequences with only the first two levels of their EC-numbers so that there is sufficient variety in the data that represents each label.

Table S1. Brief description of methods compared to Squidly in common benchmarks.

Study	Input	Method
SCREEN [9]	Structure, sequence embeddings, and evolutionary conservation.	Graph Neural Network which employs contrastive learning on EC numbers.
AEGAN [10]	Structure, Atchley Factors, PSSM conservation scores	Integrated graphical representations are generated using local protein networks and an amino acid spatial model with conservation scores and Atchley Factors included. A Graph Neural Network is trained to predict Catalytic residues.
PREvaIL [17]	Structure and amino acid sequence	Generates a range of sequence- and structure-based features representing conservation, structural descriptions, and network properties of the enzyme, and then performs feature selection and trains Random Forest models for prediction.
CRpred [8]	Sequence	Generates and predicts sequence-based features, including PSSM conservation scores and trains an SVM for prediction.

A.3 Evaluation of Existing Benchmarks:
The six standard benchmarks traditionally used for catalytic residue prediction exhibit methodological inconsistencies in the literature. In our comparative analysis, we discovered that state-of-the-art methods benchmarked in this study – specifically SCREEN and AEGAN utilized different independent subsets of these benchmarks, complicating direct performance comparisons. According to the SCREEN authors, data preparation involved clustering the EF family and M-CSA training data to less than 0.4 sequence identity, selecting unique cluster representatives for training and validation. The authors reported that they "excluded enzymes from these five test datasets from our training/validation dataset."

Table S2. Benchmark datasets in prior work.

Dataset	Year of publication	Size (before sub-setting)	SCREEN Subset	SCREEN >90% identity	AEGAN Subset	AEGAN >90% identity
EF family [35]	2007	293	NA	NA	196	3.6%
EF superfamily [35]	2007	189	49	14.3%	93	2.2%
EF fold [35]	2007	236	60	10.0%	123	3.3%
HA superfamily [36]	2007	282	191	77.0%	152	2.6%
NN [37]	2002	163	95	83.1%	108	2.7%
PC [38]	2006	81	48	85.4%	55	1.8%
SCREEN [9] training	2024	1055	1055	NA	NA	NA
AEGAN [10] training (Uni14230)	2023	8435	NA	NA	8435	NA
AEGAN test (Uni3175)	2023	1955	NA	NA	1955	0.00%

Our analysis identified that up to 85.4% of the test sequences had greater than 0.9 identity to the training set in the already filtered subsets provided by the authors. Notably, the high sequence similarity appears reduced in the EF superfamily and EF fold datasets, likely resulting from the authors' use of the closely related EF family dataset during training set curation. The test results published by SCREEN align with our observations, showing a marked increase in F1 score when comparing the HA, NN, and PC datasets with the EF datasets, corresponding to their differing levels of data leakage.

Additionally, the curation process employed in SCREEN for the training data appears unnecessarily stringent, with the average closest training sequence similarity measured at only 0.21. This sparse training data likely impacts machine learning models attempting to fit such data distributions effectively. We hypothesize this factor may explain why AEGAN reportedly performed poorly when retrained using this data in the SCREEN authors' experiments.

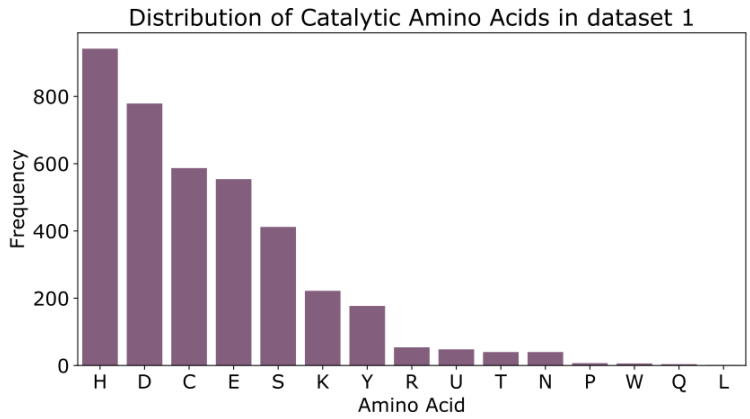

Figure S1: Amino acid distribution of Catalytic Residues in Dataset 1. Dataset 1 contains 3,873 catalytic residues, with 15 possible amino acids. The database is skewed towards certain amino acids which are commonly involved in catalysis.

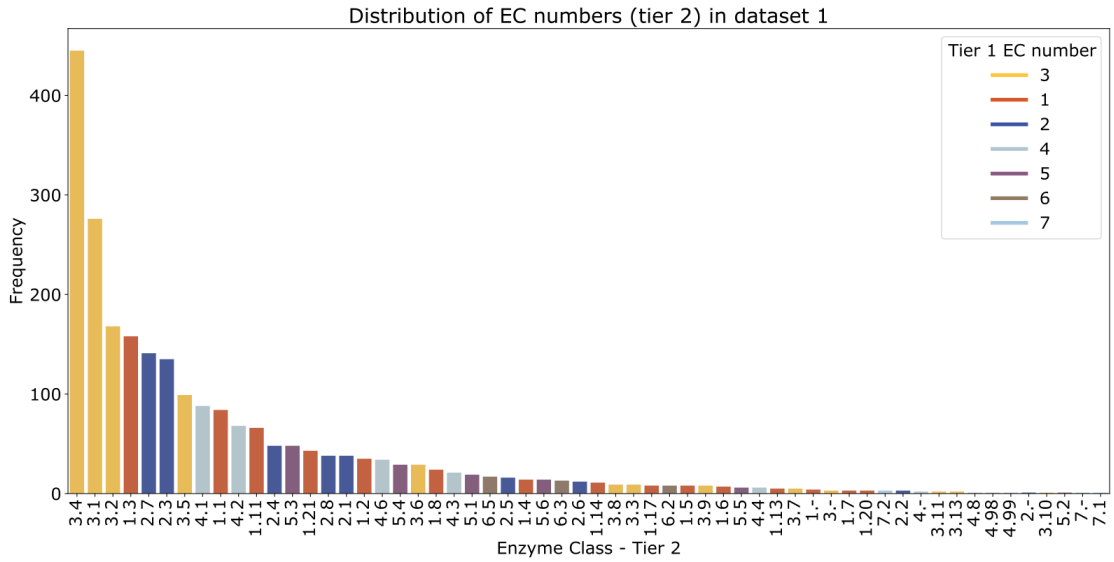

Figure S2: Dataset 1 EC Distribution. Dataset 1 is made up of 2,210 sequences from Swissprot. The sequences have experimental validation for the catalytic residue annotations. Validation sets are derived exclusively from this distribution. A clear bias exists for hydrolases (EC 3.X).

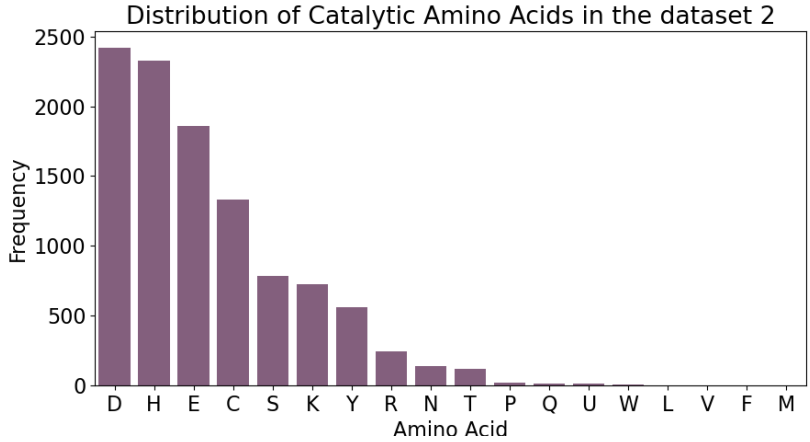

Figure S3: Amino acid distribution of Catalytic Residues in Dataset 2. Catalytic residue distribution. Dataset 2 contains 10,564 catalytic residues, with 18 possible amino acids. Amino acids M, V and F are additions not seen in dataset 1, although they exist in very few numbers.

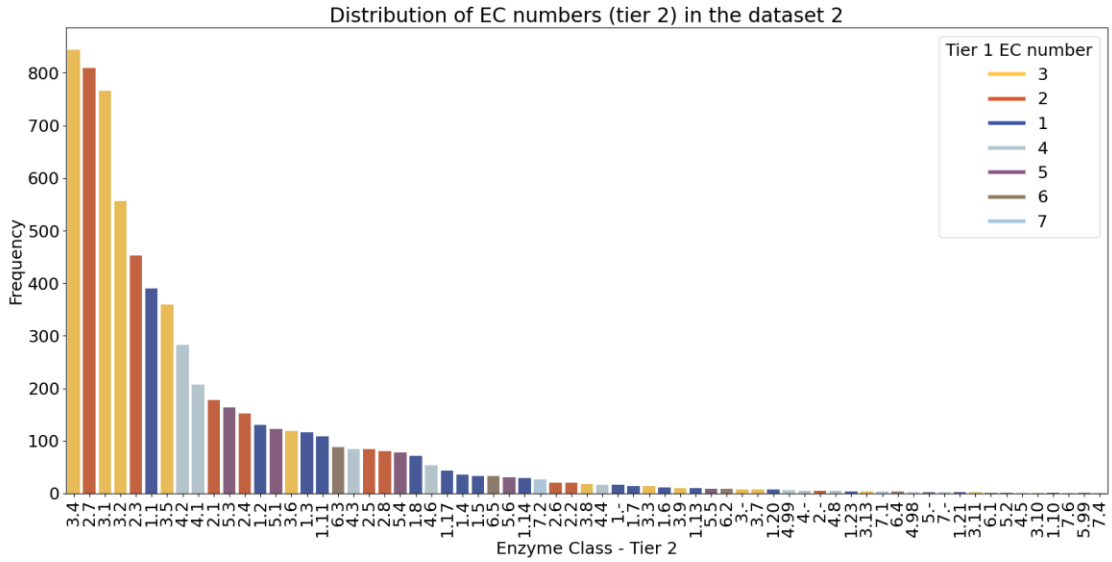

Figure S4: Dataset 3 EC Distribution. Dataset 3 is made up of 48,625 sequences. The sequences are taken from all the available proteins with catalytic site annotations in Swissprot. A very similar distribution is seen between datasets 3 and 2, with a notable increase in EC 2 again.

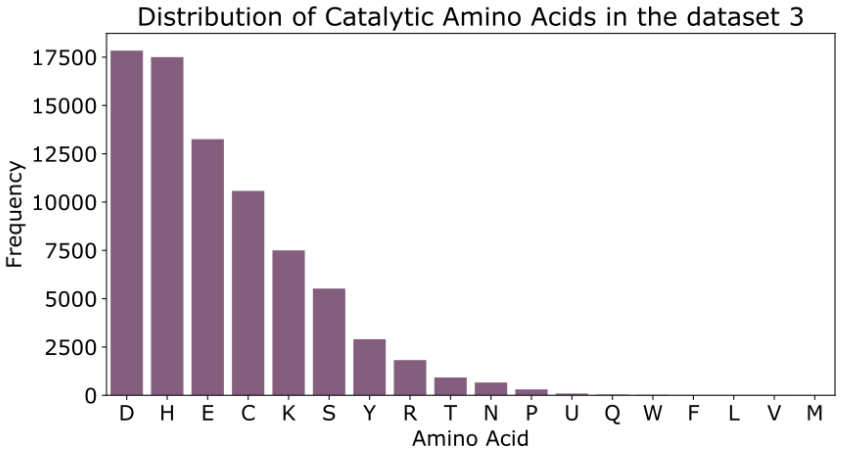

Figure S5: Amino acid distribution of Catalytic Residues in Dataset 3. Dataset 3 contains 78,966 catalytic residues, with 19 possible catalytic amino acids. Isoleucine is an additional amino acid not seen in dataset 1 or 2. The distribution seen here is very similar to that of dataset 1 and 2.

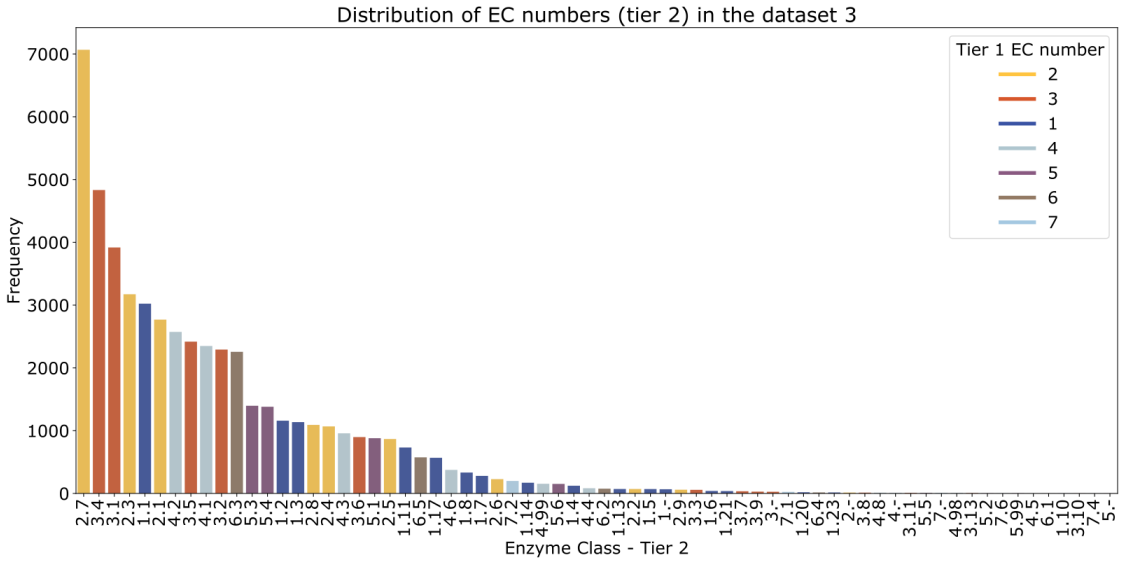

Figure S6: Dataset 3 EC Distribution. Dataset 3 is made up of 48,625 sequences. The sequences are taken from all the available proteins with catalytic site annotations in Swissprot. A very similar distribution is seen between datasets 3 and 2, with a notable increase in EC 2 again.