

---

# CARE: a Benchmark Suite for the Classification and Retrieval of Enzymes

---

**Jason Yang**

Chemistry and Chemical Engineering  
California Institute of Technology

**Ariane Mora**

Chemistry and Chemical Engineering  
California Institute of Technology

**Shengchao Liu**

Computing and Mathematical Sciences  
California Institute of Technology

**Bruce J. Wittmann**

Office of the Chief Scientific Officer  
Microsoft Corporation

**Anima Anandkumar**

Computing and Mathematical Sciences  
California Institute of Technology

**Frances H. Arnold**

Chemistry and Chemical Engineering  
Biology and Biological Engineering  
California Institute of Technology

**Yisong Yue \***

Computing and Mathematical Sciences  
California Institute of Technology

## Abstract

Enzymes are important proteins that catalyze chemical reactions. In recent years, machine learning methods have emerged to predict enzyme function from sequence; however, there are no standardized benchmarks to evaluate these methods. We introduce CARE, a benchmark and dataset suite for the Classification And Retrieval of Enzymes (CARE). CARE centers on two tasks: (1) classification of a protein sequence by its enzyme commission (EC) number and (2) retrieval of an EC number given a chemical reaction. For each task, we design train-test splits to evaluate different kinds of out-of-distribution generalization that are relevant to real use cases. For the classification task, we provide baselines for state-of-the-art methods. Because the retrieval task has not been previously formalized, we propose a method called Contrastive Reaction-Enzyme Pretraining (CREEP) as one of the first baselines for this task and compare it to the recent method, CLIPZyme. CARE is available at <https://github.com/jsunn-y/CARE/>.

## 1 Introduction

Proteins, which are sequences of amino acid building blocks, are not only integral components of all living organisms, but also important for a myriad of commercial applications spanning from the health domain to the bio-economy. Enzymes are a subclass of proteins that can catalyze chemical reactions, and they have many applications in areas such as bioremediation, plastic degradation, gene editing, and drug synthesis [1, 2, 3, 4].

Identifying the specific chemical reactions that an enzyme is capable of performing (*i.e.* the enzyme’s function) is a key first step for many applications [5]. While hundreds of millions of proteins

---

\*Correspondance: yyue@caltech.edu

have been sequenced, less than 1% are annotated with function [6]. For instance, during standard metagenomic analyses, enzyme genes are annotated for their functions, which enables determination of an organism’s critical metabolic pathways and specialization [7, 8, 9]. When applied to chemical synthesis, enzyme annotations are needed to identify catalysts that can replace existing step(s) in drug synthesis procedures—known as retrobiosynthesis [10]. Likewise, enzyme engineering for new-to-nature function [11] involves discovering an enzyme starting point with a desired function before improving activity for that function using protein engineering techniques [12, 13]. Historically, similarity search algorithms, most notably BLAST, have been the most common methods used to assign function to protein sequences [14, 15]. These methods work by finding similar sequences in annotated reference databases, as similar sequences are likely to share function. More recently, searching based on structural similarity (Foldseek) has also demonstrated utility for assigning function [16, 17]. However, up to one third of genes from bacterial genomes cannot be assigned function using existing methods [7]. As such, there is a need for more abundant, high-quality annotations of enzymes and automated workflows to identify enzymes with desired functions.

In recent years, there has been increasing interest in using machine learning (ML) for a broad range of applications related to the functional prediction and design of enzymes [13, 18, 19]. In particular, ML models have emerged to classify protein sequences based on their function, which are reviewed here [20] with a few examples listed here [21, 9, 22, 23, 24, 25, 26]. Despite these advances, there is no standard benchmark or dataset for evaluating computational models for enzyme function prediction [27]. A challenge associated with classification is that a given enzyme is often able to perform multiple reactions [28], and many reactions are not annotated. Moreover, complex tasks such as extrapolation to unannotated reactions [29] have yet to be evaluated.

In this work, we present a benchmark suite for the **Classification And Retrieval of Enzymes (CARE)**. Our contributions can be summarized as: **(1)** formalizing model evaluation into two tasks that encompass applications relevant to scientists and engineers: classification of an enzyme by function (Task 1), and retrieval of enzyme sequences based on a reaction (Task 2); **(2)** curating high-quality and easy-to-use datasets; **(3)** providing train-test splits that mimic challenging extrapolations in real-world use cases; and **(4)** benchmarking state-of-the-art models for Task 1, and providing a new method that serves as a baseline for Task 2. Because Task 2 has been minimally explored, we introduce a model called **Contrastive Reaction-Enzyme Pretraining (CREEP)** to serve as a baseline for text, reaction, and sequence integration and compare it to an existing approach (**CLIPZyme**) that uses sequence and reaction, in addition to other retrieval approaches. CREEP can perform contrastive learning across three different modalities (protein, reaction, and textual description), and the learned representations are then used for retrieval. Overall, we anticipate that CARE will be a useful and easy-to-use resource for ML researchers to benchmark their enzyme function prediction models.

## 2 Related Work

**Datasets.** Various databases have emerged to help researchers store, share, and identify functionally annotated enzymes. Protein sequence databases such as UniProt [6] and Pfam [30] are catalogs of annotated protein sequences. While most databases reference protein sequences, increasingly, these sequences can be linked to protein structures, either experimentally validated, as in the Protein Data Bank (PDB), or via structural prediction tools [31] in databases such as the AlphaFold Database [32]. BRENDA [33] is a curated database specific for reaction and enzyme sequence information. Rhea [34] consolidates information from BRENDA, and other sources, such as pathway databases including KEGG [35]. There is ongoing work to compile and clean/standardize reactions from multiple databases, namely ECRreact [36] and EnzymeMap [37]. Related to these databases, there are retrobiosynthesis planning tools [38, 39, 40], and Selenzyme is tool to retrieve enzymes to perform a target reaction [41].

**Protein Benchmarks.** Our work takes inspiration from existing protein fitness prediction benchmarks, where fitness is a quantification of some function. TAPE [42] and FLIP [43] evaluate representations from protein language models for the prediction of a broad range of general and specific protein properties (stability, secondary structure, binding etc.) ProteinGym considers sequence variant effect prediction by using likelihoods from these language models [44]. While benchmarks for protein fitness prediction tasks are well defined, there are no standardized benchmarks for protein function prediction. Fitness is a numerical quantification of protein function (stability, enzyme activity

level, etc.), but function is more qualitative/categorical, e.g. a description of an enzymatic reaction associated with a protein. Here, we focus on proteins that perform chemical reactions, enzymes.

**Classification of Enzyme Function.** ML models have emerged to predict the outcomes of enzymatic reactions [45, 46, 47] and for classification of enzyme function. Enzyme function is usually expressed using enzyme commission (EC) numbers, which is a hierarchical scheme for classifying enzyme function into classes (families) and consists of four levels of descriptions (Figure 1A). Some classification models are general protein function prediction models, which encompass all proteins, not just enzymes, such as ProtCNN/ENN [24] and ProteInfer [25]. Many models utilize representations from protein language models [48, 49, 50], and others incorporate protein structure as information, such as DeepFRI [23] and BioCLIP [51]. Methods related to supervised contrastive learning [52] have been particularly useful here, such as CLEAN, HiFi-NN, and Enzhier [22, 9, 53], likely by reducing imbalances in the number of sequences representing each EC number. Recently, enhanced approaches have enabled function prediction in ProtEx[26], PhiGnet [54], and others [55, 56, 57]. Other retrieval tools enable more sensitive detection of homologs such as DHR [58], ProfTrek [59], and using structure [17, 60, 61]. Finally, there are related models that predict substrates for enzymes [62, 63] and that aim to learn connections between chemical space and protein space [64].

**Large Language Models.** Recently, there has been an explosion in pretrained models in the biological and chemical domains, particularly large language models (LLMs) [65, 66]. For example, ChatGPT is capable of answering questions related to general scientific knowledge. These language models can be further finetuned for applications such as answering questions about protein sequences (including Pika [67], InstructProtein [68], InstructBioMol [69], ProteinGPT [70], and ProteinChat [71]) and for reaction synthesis planning (ChemCrow [72]), among others. LLMs present important benchmarks for enzyme functional classification and retrieval, given the widespread adoption of LLMs as science facilitators and their ease of use (in particular webserver-based approaches), compared to domain-specific methods [73].

**Multimodal Contrastive Learning.** Contrastive learning is an efficient and effective pretraining paradigm that aligns positive pairs and contrasts negative pairs simultaneously. The design of these pairs depends on the specific tasks, such as using data augmentations of the same image [74] or considering the topology and geometry of molecules [75]. More recently, contrastive learning has shown success in aligning the representation space of different biological and chemical modalities, e.g., text and chemical structure alignment in MoleculeSTM [76], text and protein sequence alignment in ProteinDT [77] and ProteinCLIP [78], reaction structure and protein structure alignment in CLIPZyme [79], and protein sequence and structure alignment in BioCLIP [51] and with text in ProTrek [59], among others [80, 65]. Cross-modal alignment in the representation space has been shown to improve generalizability and improve performance on challenging tasks, such as out-of-distribution learning, zero-shot learning, and text-guided molecule design and optimization [77]. Consideration of multiple modalities may be especially important for the prediction of qualitative functions.

### 3 Overview of CARE

Though there are many studies using ML models to perform enzyme classification based on EC numbers (Figure 1A), there is no standardized benchmark to evaluate how well these models generalize to unseen protein sequences. To address this need, we present a benchmark suite for the classification and retrieval of enzymes (CARE, Figure 1B). CARE formalizes classification of an enzyme sequence by EC number as "Task 1" (Figure 1C). For this task, we design train-test splits of protein sequences to test out-of-domain generalizations that are relevant to real-world use cases. In addition, CARE addresses another key limitation of current studies: classification is limited to EC numbers, which is a closed vocabulary of functions (reactions), so existing models cannot generalize to unannotated reactions. Thus, we introduce an entirely new task, retrieval of an EC number given a reaction, which we call "Task 2" (Figure 1D). For this task, we design train-test splits to evaluate how well models can generalize to out-of-domain reactions, ensuring that the splits pose different levels of difficulty.

To streamline benchmarking, we curate a dataset of enzymes, reactions, and their associated EC numbers for CARE. At a high level, we build two datasets, one that links protein sequence to EC and one that links reaction to EC (Figure 1A). The former is processed from Swiss-Prot, the validated portion of UniProt [?] and filtered to protein sequences between length 100 and 1024 with annotated

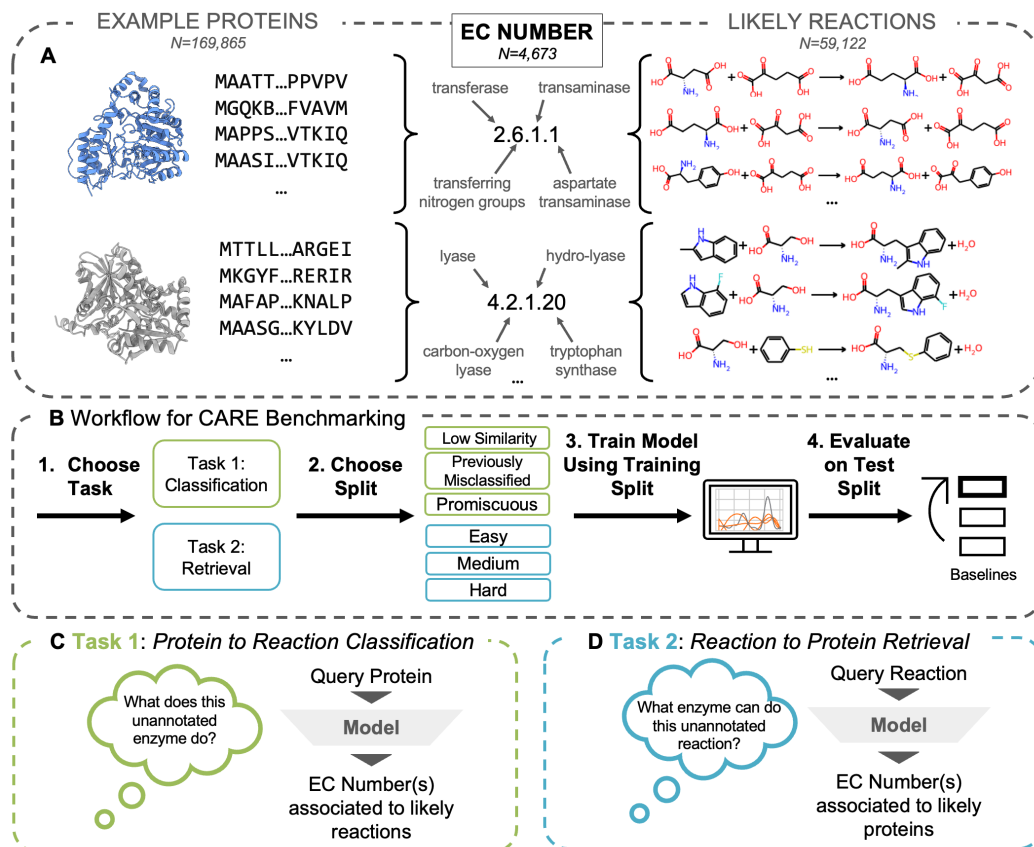


Figure 1: **Overview of CARE.** (A) Dataset format for CARE, showing examples of enzymes and their associated reactions. The EC number acts as a bridge between a protein sequence and the reactions it is likely to perform. The EC number is a hierarchical classification scheme for enzyme function with four levels of description, with increasing specificity from left to right. (B) General workflow for CARE benchmarking. (C) Task 1 is a real-world use case for enzyme classification based on a protein sequence. (D) Task 2 is a real-world use case for enzyme retrieval based on a reaction.

EC number(s). The latter is formed as a combination of EnzymeMap [37] and ECReact [36], where ECReact is only used to supplement EC numbers that are missing in EnzymeMap. Our workflow for generating the datasets used in this work is explained in detail in Appendix A.1 and shown visually in Appendix Figure A.1.

The overall workflow for benchmarking using CARE is shown in Figure 1B. For each task, domain-specific train-test splits are provided from the processed datasets. Model training can use any of the data in the train split, and each model is evaluated on the associated test split. In the rest of this study, we explain the specific design choices used to generate train-test splits and analyze benchmarking results of state-of-the-art methods on these splits. The curated datasets and splits used in CARE can be accessed at <https://github.com/jsunn-y/CARE/>.

## 4 Task 1: Enzyme Classification

Task 1, classification of an enzyme sequence, tests the ability of a model to extrapolate to unseen protein sequences. Task 1 is a fairly well studied task [20], but model evaluation has not been previously standardized as there are many factors to consider, such as the distribution of sequences and functions in the test sets. Task 1 applies to use cases where a scientist is given an unannotated enzyme sequence and seeks to understand the enzymatic function associated with that sequence (Figure 1C), for example for metagenomic analysis or finding new enzymes for retobiosynthesis. With the emergence of conditional generative models for protein sequences, it is also important to

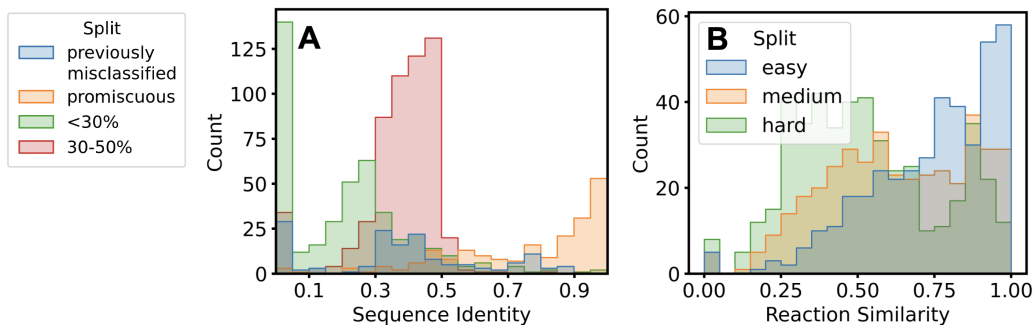


Figure 2: **Distribution of similarities between samples in each test set and the corresponding train set.** (A) Protein sequence identity (Task 1) was measured to the closest hit in the train set using BLASTp. Sequence identity can be thought of as normalized Levenshtein distance. (B) Reaction similarity (Task 2) was measured as cosine similarity with each reaction represented using DRFP [86]. Reaction similarity was measured from the test set to each train-set cluster center of reactions belonging to an EC number.

Table 1: **Summary of train-test splits used in Task 1.** For Task 1, certain protein sequences are held out. Train "samples" refers to the number of unique protein-EC data pairs, and test "samples" refers to the number of protein sequences. All splits for Task 1 share the same protein train set.

Split name	Description	Train samples	Test samples
<30% Identity	Held-out sequences with approximately < 30% identity to the training set	184,529	432
30-50% Identity	Held-out sequences with approximately 30-50% identity to the training set	184,529	560
Previously Misclassified (Price)	Previously misclassified enzyme sequences from Price et al.	184,529	148
Promiscuous	Held-out sequences with multiple EC numbers	184,529	209

have high-throughput computational methods that can predict the function of generated sequences [81, 82, 83, 84, 85]. For this task, a query protein is passed through a trained model to predict an EC number. The EC number is then associated with likely reactions that the protein will be able to perform.

**Splits for Task 1.** Task 1 can be framed as evaluating how models generalize toward unseen sequences with different types of difficulty, visualized in Appendix Figure A.3A. The train-test splits for Task 1 are summarized in Table 1.

- *<30% and 30-50% identity splits*: these two test splits contain sequences with sequence identities falling in the respective range, to sequences in the training set by using clustering (Section A.2). Sequence identity is related to the normalized Levenshtein distance between two sequences. Natural sequences with high (>40%) sequence identity at the protein sequence level are likely to share function [87]. It is expected that the lower the sequence identity, the more difficult it is to assign functional annotation.
- *previously misclassified (Price et al.) [7] split*: challenging to assign because some have low sequence identity to other proteins, and many may lie near "activity cliffs" (Figure A.3), a region where function can change sharply in sequence space.
- *promiscuous split*: in this study, we define promiscuous enzymes as those mapping to multiple EC numbers (thus lying on multiple activity peaks). These enzymes are particularly interesting for enzyme engineers, as new-to-nature activity can be found in between activity peaks [13].

For Task 1, the training split is constructed by holding out all of the pooled sequences in the test splits. The classification output vocabulary (EC numbers) is closed rather than open, so EC numbers in the

Table 2: **Performance of various methods on Task 1.** Performance is measured as k=1 classification accuracy (%). However, for the promiscuous split, we use k=(number of true ECs) for Random and CLEAN and report the average accuracy across all true EC numbers. For this split, we did not evaluate ChatGPT and Pika, as we prompted for only a single EC in the response. For BLAST and Foldseek, we use all EC numbers associated to the top single hit. More details are provided in Section A.4. CLEAN is a state-of-the-art method at the time of publication. Other methods such as ProteInfer, HiFi-NN, Enzhier, ProtEx, PhiGnet, etc. could also be benchmarked here.

Split	Method	Level 4 Accuracy (X.X.X.X)	Level 3 Accuracy (X.X.X.-)	Level 2 Accuracy (X.X.-.-)	Level 1 Accuracy (X.-.-.-)
<30% Identity	Random	0.0	1.2	3.2	19.4
	BLASTp	51.4	60.0	62.5	65.7
	Foldseek	54.9	68.3	72.9	80.6
	ChatGPT	0.0	0.0	1.6	28.9
	Pika	20.6	37.7	46.1	61.6
	CLEAN	<b>55.1</b>	<b>68.8</b>	<b>74.8</b>	<b>84.5</b>
30-50% identity	Random	0.0	0.7	3.6	22.5
	BLASTp	<b>81.1</b>	87.9	90.7	92.3
	Foldseek	79.8	87.1	90.5	95.0
	ChatGPT	0.0	1.4	3.0	34.8
	Pika	37.7	50.2	60.0	73.8
	CLEAN	80.2	<b>88.0</b>	<b>91.6</b>	<b>95.5</b>
Previously Misclassified (Price)	Random	0.0	0.7	4.7	22.3
	BLASTp	35.1	70.9	78.4	78.4
	Foldseek	<b>41.2</b>	<b>82.4</b>	<b>93.2</b>	<b>96.6</b>
	ChatGPT	0.0	9.5	17.6	37.2
	Pika	4.1	50.7	64.9	82.4
	CLEAN	31.8	74.3	81.8	85.8
Promiscuous	Random	0.5	4.1	9.0	41.1
	BLASTp	<b>93.7</b>	<b>94.8</b>	<b>95.2</b>	<b>95.9</b>
	Foldseek	88.0	90.9	92.4	94.4
	CLEAN	69.4	77.9	81.5	87.0

test sets are present in the train sets. We verify that sequences in the test sets generally fall within the expected sequence identities, relative to the training set (Figure 2A). Notably, the *Price* test set has a wide distribution of sequence identities, while the *promiscuous* test set has high sequence similarity to the training set. The sequences in the test splits are distributed generally evenly across all different EC numbers (Figure A.4 in Appendix). More details can be found in Appendix A.2.

**Task 1 benchmarking results.** Benchmarking results for Task 1 are summarized in Table 2. We start with two methods as baselines, classification using a random order of EC numbers (Random), and Diamond BLAST at the protein sequence level, herein referred to as BLASTp, which is a workhorse bioinformatics tool that performs local-alignment to determine the most similar sequence(s) given a target query and a database [14]. Additionally, we include search by structural similarity using Foldseek [16].

While there exist many ML tools to directly perform EC number classification, CLEAN [22] and several others [53, 9, 26, 54, 57] are a few that seem to report the current state-of-the-art with comparable performance. Some of the latter methods were not yet publicly available at the time of this study, and other recent retrieval methods would also be promising approaches to test, but we opted to focus on benchmarking on CLEAN here. CLEAN generally performs the best for enzymes with low sequence identity to known sequences, but the BLASTp and Foldseek baselines perform similarly. ChatGPT was also tested, but it appeared to often hallucinate as it was forced to provide an answer, with results similar to the random baseline. Pika seems to bridge the gap between standard LLMs and enzyme classification, but its performance is not as good as standard tools like BLASTp [67].

Table 3: **Summary of train-test splits used in Task 2.** Certain reactions are held out, and "samples" refers to the number of reaction-EC pairs.

Split name	Description	Train ECs	Train samples	Test samples
Easy	Certain reactions are held out, sampled uniformly across ECs, but no EC numbers are held out. The test set is the same as the holdout set.	4,960	61,373	393
Medium	All reactions corresponding to certain ECs are held out, at EC level 4 (X.X.X.X). Test set reactions are sampled uniformly across ECs from the holdout set.	4,748	57,691	393
Hard	All reactions corresponding to certain ECs are held out, at EC level 3 (X.X.X.-). Test set reactions are sampled uniformly across ECs from the holdout set.	3,052	35,252	460

Overall, these results suggest that there is still room to improve classification of protein sequences with low sequence identity (<30%) and that lie near multiple activity peaks (*Price*). In fact, Foldseek has the best performance on the *Price* split, suggesting that structure can be conserved, even in scenarios where sequence is misleading. The *promiscuous* split is not easy for all methods; even though the test sequences have high sequence identity to the train set, CLEAN sometimes misses EC numbers. Interestingly, BLASTp and Foldseek perform close to the state-of-the-art, even when compared to more complex ML models. Future models could take advantage of this finding to augment training, as ProtEx does [26]. Additional details on the implementation of each method can be found in Appendix A.4.

## 5 Task 2: Enzyme Retrieval

Task 2, enzyme retrieval from a query reaction, tests the ability of a model to extrapolate to unseen reactions. Task 2 has not been formalized or explored in previous studies, but it is equally as important as Task 1, as it applies to a use case where a scientist or engineer is seeking to identify a previously characterized enzyme sequence that can perform a novel (unannotated) reaction (Figure 1D). Typical applications include: an environmental engineer looking for an enzyme to degrade a toxic pollutant [88], an enzyme engineer looking for an enzyme to catalyze a selective reaction for drug synthesis [89, 90], or a gene annotator identifying the gene for an "orphan" enzyme with known function but unknown sequence [91]. For Task 2, a query reaction is passed through a trained model to perform retrieval to an EC number and its associated proteins that are likely to be able to perform that reaction.

**Splits for Task 2.** Task 2 aims to evaluate how well a model generalizes to unseen reactions with different levels of difficulty. The train-test splits for Task 2 (*easy*, *medium*, and *hard*) are summarized in Table 3 and visualized in Appendix Figure A.3B. We equate greater difficulty with a more challenging train-test split; in a harder set, the test reactions are less similar to reactions in the corresponding train set. We decide similarity based on the amount of overlap in EC number (e.g., a reaction from 4.2.1.20 is considered more similar to another one in 4.2.1.20 than one from 4.2.1.1).

- *easy* split: EC numbers are randomly sampled at EC level 4 (X.X.X.X) and then randomly mapped to reactions, which are held out as the test set.
- *medium* split: the same reactions are used for testing as the easy set, but all other reaction-EC pairs which share the same EC level 4 (X.X.X.X) are held out from training.
- *hard* split: random EC numbers are sampled at EC level 3 (X.X.X.-) and all reactions under that EC3 are held out from training, while a subset of reactions from the held-out EC numbers are used for testing.

The sequences in the test splits are distributed generally evenly across different EC numbers (Appendix Figure A.4). From *easy* to *medium* to *hard*, the test set reactions also become more dissimilar to their respective training sets (Figure 2B). Reaction similarity was quantified by DRFP, which is a reaction representation that uses set differences between product and reactant fingerprints and has demonstrated solid performance without requiring model training [86]. Overall, Task 2 is more

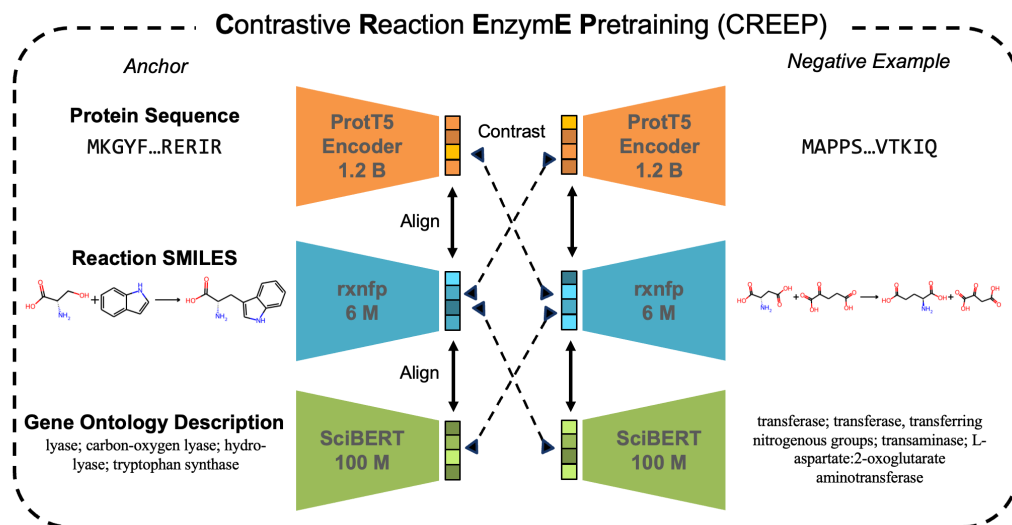


Figure 3: **Model architecture for CREEP.** CREEP aligns reaction and protein sequence using contrastive learning to perform downstream retrieval between domains. Optionally, CREEP training can be augmented by using textual description as a third modality to bridge the other two. Model sizes by number of parameters are also listed.

complex than Task 1, because unlike Task 1, entire EC numbers are held out from the training set (i.e., the classification output vocabulary is open rather than closed), so multiple modalities must be considered to link the unseen reactions to their respective EC numbers. More details on splitting can be found in Appendix A.2.

**CREEP baseline method.** There is a lack of models that have been tested for their ability to generalize beyond annotated reactions and none that have considered reaction, text and sequence together, so we develop a contrastive learning method for this task, called **Contrastive Reaction-Enzyme Pretraining (CREEP)** (Figure 3). Our approach is related to CLIPZyme [79], which uses contrastive alignment of reactions represented as 2D graphs and protein structures represented as 3D graphs. Somewhat differently, CREEP leverages finetuning of pretrained language models, rxnfp [92] and ProtT5 [50] to learn aligned representations of reactions and proteins, respectively. Rxnfp is a BERT-style [93] language model that was trained on reactions represented as SMILES/SMARTS strings and has demonstrated state-of-the-art performance on reaction type classification [92]. ProtT5 is a T5 protein language model that has been used for prediction of various protein properties and demonstrates similar performance to ESM [50, 49]. We used these language models for simplicity and their ease of finetuning. Uniquely, CREEP can learn on a third modality (CREEP w/ Text): textual description of the EC number based on gene ontology [94], which is encoded using SciBERT [95]. The use of text has recently been shown to boost function prediction tasks [96]. More details on CREEP training can be found in Appendix A.3.

**Task 2 benchmarking results.** Benchmarking results for Task 2 are summarized in Table 4. We start with two methods as baselines: randomly guessing a ranking of EC numbers (Random) and finding the most similar reaction in the train set to the test query (Similarity Baseline). For the Similarity Baseline, we used DRFP to represent chemical reactions [86]. Details on the downstream multimodal retrieval process can be found in Appendix A.3 and Figure A.4. Methods like CREEP and CLIPZyme retrieve reference protein sequences based on a query reaction, and those proteins are mapped to a ranking of retrieved EC numbers; thus, they can generalize to EC numbers that are not linked to reactions in (i.e. missing from) the training set.

Overall, the Similarity Baseline is quite strong and performs much better than Random. ChatGPT was prompted with reactions written using compound IUPAC names rather than SMILES strings, both with (ChatGPT w/ Text) and without (ChatGPT) textual descriptions from gene ontology [94]. We only show ChatGPT performance for the easy split, as ChatGPT is not a good measure of generalization on the medium and hard splits. These both require "leaving out" entire EC numbers before training to ensure there is no data leakage.



Table 4: **Performance of various methods on Task 2.** Performance is measured as k=1 retrieval accuracy (%). \*denotes that there may be data leakage causing performance to be inflated. Bolded accuracy is the best model.

Split	Method	Level 4 Accuracy (X.X.X.X)	Level 3 Accuracy (X.X.X.-)	Level 2 Accuracy (X.X.-.-)	Level 1 Accuracy (X.-.-.-)
Easy	Random	0.0	1.0	4.6	22.9
	Similarity Baseline	59.3	77.1	85.2	90.6
	ChatGPT*	4.8	22.6	43.5	71.0
	ChatGPT (w/ Text)*	13.7	56.7	81.4	<b>98.5</b>
	CLIPZyme	12.2	39.9	61.8	79.9
	CREEP	39.4	66.4	79.9	92.9
	CREEP (w/ Text)	<b>60.3</b>	<b>89.3</b>	<b>93.9</b>	96.7
Medium	Random	0.0	0.5	4.1	17.0
	Similarity Baseline	0.0	40.2	55.7	73.3
	CLIPZyme	2.0	26.0	46.6	69.0
	CREEP	4.1	44.3	63.1	86.5
	CREEP (w/ Text)	<b>7.4</b>	<b>59.5</b>	<b>75.6</b>	<b>92.1</b>
Hard	Random	0.0	0.9	1.5	18.3
	Similarity Baseline	0.0	0.0	13.5	42.0
	CLIPZyme	1.1	4.1	13.5	46.7
	CREEP	1.3	4.8	18.7	<b>57.6</b>
	CREEP (w/ Text)	<b>1.3</b>	<b>9.8</b>	<b>22.2</b>	57.2

On the more difficult test sets, CREEP offers an advantage compared to the Similarity Baseline, and particularly when combined with textual description: CREEP (w/ Text). This suggests that utilizing protein sequence information as a modality is useful. Still, performance on the harder splits is weak across the board, which suggests that there is significant room for future improvement, although retrieval of EC class may not be the ideal metric for out-of-distribution reaction classification. We anticipate that contrastive alignment with textual descriptions will play an increasingly important role in enzyme retrieval [78, 77, 65], and there is opportunity for better curation of these descriptions. Alternatively, CLIPZyme [79] represents reactions and proteins as graphs, but performance is lower, potentially due to some sample loss from some data being unable to be processed to graphs during training and inference. Here, performance could likely be improved by optimizing training hyperparameters. Additional details on the implementation of each method are in Appendix A.4, and additional benchmarking results are presented in Figure A.6.

## 6 Discussion

We made an important design choice in the CARE benchmarks: to perform classification and retrieval at the coarse-grained level of EC numbers, which could be a limitation. Enzymes can often perform many reactions, meaning it is an acceptable assumption that enzymes belonging to the same EC number will share the capacity to perform similar reactions, even if they are not directly annotated for all reactions. The ultimate task in this domain is to perform direct reaction to protein sequence retrieval and vice-versa (as in CLIPZyme [79] and Reactzyme [97]). However, currently the data for validation is limited, and we believe that many negative examples in those datasets may actually be feasible protein-reaction pairs. As experimentalists obtain higher resolution annotations of proteins and their associated reactions, this will become more realistic. Furthermore, our goal was to provide benchmarks for a broad range of functions, but model evaluation could also be performed on more specific classes of enzymes with direct protein-reaction pairs, as explored by EnzymeCAGE [98]. Other ways to hold out proteins with difficult-to-predict functions could also be explored. Some EC numbers are also incorrectly annotated, which are discussed in more detail here [62].

In the future, the proposed train-test splits could be refined for both tasks. For the Task 1 splits, there is some leakage in the sequence identity, with some sequences in test sets lying outside of the enforced sequence identity ranges, likely due to the different sequence similarity algorithms used

(MMseqs2 vs Diamond BLAST). MMseqs2 utilizes cascaded clustering which pre-filters based on initial clusters in the target set [99], while BLAST attempts to identify the closest sequence within the specified set based on local similarity. Another limitation is that while there are no duplicate reactions present in both the train and test sets, some of the test reactions are very similar to reactions in the training set despite having different ECs. It would be beneficial to do a more detailed analysis of reaction similarities and explore other representations of reactions to understand which reactions can be considered equivalent. Many EC classes also involve multi-complex enzymes; in other words, certain subunits of these enzyme are not actually performing catalytic activity. Future work could filter out some of the non catalytically active subunits or act to specifically predict the catalytic subunit, or the entire complex based on a reaction. Over time, the train-test splits should be updated as additional functional annotations are acquired and compiled in databases. For example, while over 36 million sequences in BRENDA/UniProt have EC numbers associated with them, these are often detected using homology based models, which may incorrectly assign EC numbers. A more detailed analysis of the specific failure modes of these ML models could be valuable future work [29].

We opted to assess performance using accuracy due to its simplicity and interpretability, but other retrieval and virtual screening metrics such as BEDROC and enrichment could also be explored. For the promiscuous enzymes with multiple EC numbers, we reported accuracy averaged across all of the true labels, but other classification metrics such as precision and recall should be considered in future work. The number of retrieved ECs could also be chosen using strategies like max separation and p-value as implemented in CLEAN [22].

There is also significant opportunity to use other modalities such as textual description and protein structure for more effective representations of enzymes and reactions, or to improve the text-based annotations. Gene context is also useful for annotating protein function and could be incorporated into future benchmarks [100]. The textual annotations used in this work are direct textual references to the EC class, meaning that the models in task 2 that incorporate text (e.g. CREEP), are effectively learning the relationship between the EC as a word and the EC as a number, in addition to the similarity of the reaction. Other tools such as Pika which use multiple types of textual description extend this to learn relations about the enzyme, and could be used in future iterations of CREEP. Future work will involve incorporating structure and graph based representations [101] into CREEP, similar to those used in CLIPZyme [79]. We also plan to do a more detailed analysis of the representations learned by CREEP. The addition of textual description in CREEP potentially introduces indirect data leakage between the train and test sets, so future iterations of CARE will need to consider this. Future work should also consider how to include protein function prediction models that go beyond enzymes—or models that would like to use additional data/modalities [59, 102]—into the CARE evaluation framework. LAB-Bench provides a framework for evaluating scientific reasoning using language, which will increasingly intersect with CARE [73]. A major bottleneck we encountered was that many models, including language models, were not available or difficult to use, limiting our ability to include them as benchmarks. There is also room to improve the prompt engineering of language models which could be further explored to enhance the performance of these models.

We finally note that methods used to retrieve dangerous proteins that could be used as bioweapons or to synthesize dangerous chemicals is a concern. The implications of this are discussed here [103].

## 7 Conclusion

Predicting the functions of enzymes is important for many applications ranging from gene annotation to enzyme engineering. While many models exist to classify enzyme function via EC numbers, there are no standardized benchmarks for evaluation of these models. Furthermore, no existing models have been tested for generalization beyond annotated reactions. To address this need, we introduce CARE, which is a benchmarking suite to formalize model evaluation for these two tasks. We also present CREEP, a model which uses multimodal contrastive learning and is one of the first models that can perform the latter task. We encourage developers to integrate their current and future methods or benchmarking results into the CARE Github repository (<https://github.com/jsunn-y/CARE/>). Overall, CARE is an important tool for encouraging progress in enzyme functional annotation. We believe that we are just seeing the beginning of the widespread adoption of multimodal models for protein functional prediction, and we expect that many researchers will find CARE useful for formulating and evaluating their models.

## Acknowledgments and Disclosure of Funding

This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Basic Energy Sciences, under Award Number DE-SC0022218. This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof. J.Y. is partially supported by the National Science Foundation Graduate Research Fellowship. A.M. is supported by the Schmidt Science Fellows, in partnership with the Rhodes Trust. The authors thank Peter Mikhael and Itamar Chinn for helpful discussions on CLIPZyme implementation.

## References

- [1] R. Buller, S. Lutz, R. J. Kazlauskas, R. Snajdrova, J. C. Moore, and U. T. Bornscheuer. From nature to industry: Harnessing enzymes for biocatalysis. *Science*, 382(6673):eadh8615, November 2023. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.adh8615.
- [2] Frances H. Arnold. Directed Evolution: Bringing New Chemistry to Life. *Angewandte Chemie International Edition*, 57(16):4143–4148, April 2018. ISSN 1433-7851, 1521-3773. doi: 10.1002/anie.201708408.
- [3] Kai Chen and Frances H. Arnold. Engineering new catalytic activities in enzymes. *Nature Catalysis*, 3(3):203–213, January 2020. ISSN 2520-1158. doi: 10.1038/s41929-019-0385-5.
- [4] Julia C. Reisenbauer, Kathleen M. Sicinski, and Frances H. Arnold. Catalyzing the future: recent advances in chemical synthesis using enzymes. *Current Opinion in Chemical Biology*, 83:102536, December 2024. ISSN 13675931. doi: 10.1016/j.cbpa.2024.102536.
- [5] Predrag Radivojac, Wyatt T Clark, Tal Ronnen Oron, Alexandra M Schnoes, Tobias Wittkop, Artem Sokolov, Kiley Graim, Christopher Funk, Karin Verspoor, Asa Ben-Hur, Gaurav Pandey, Jeffrey M Yunes, Ameet S Talwalkar, Susanna Repo, Michael L Souza, Damiano Piovesan, Rita Casadio, Zheng Wang, Jianlin Cheng, Hai Fang, Julian Gough, Patrik Koskinen, Petri Törönen, Jussi Nokso-Koivisto, Liisa Holm, Domenico Cozzetto, Daniel W A Buchan, Kevin Bryson, David T Jones, Bhakti Limaye, Harshal Inamdar, Avik Datta, Sunitha K Manjari, Rajendra Joshi, Meghana Chitale, Daisuke Kihara, Andreas M Lisewski, Serkan Erdin, Eric Venner, Olivier Lichtarge, Robert Rentzsch, Haixuan Yang, Alfonso E Romero, Prajwal Bhat, Alberto Paccanaro, Tobias Hamp, Rebecca Kaßner, Stefan Seemayer, Esmeralda Vicedo, Christian Schaefer, Dominik Achten, Florian Auer, Ariane Boehm, Tatjana Braun, Maximilian Hecht, Mark Heron, Peter Hönigschmid, Thomas A Hopf, Stefanie Kaufmann, Michael Kiening, Denis Krompass, Cedric Landerer, Yannick Mahlich, Manfred Roos, Jari Björne, Tapio Salakoski, Andrew Wong, Hagit Shatkay, Fanny Gatzmann, Ingolf Sommer, Mark N Wass, Michael J E Sternberg, Nives Škunca, Fran Supek, Matko Bošnjak, Panče Panov, Sašo Džeroski, Tomislav Šmuc, Yiannis A I Kourmpetis, Aalt D J Van Dijk, Cajo J F Ter Braak, Yuanpeng Zhou, Qingtian Gong, Xinran Dong, Weidong Tian, Marco Falda, Paolo Fontana, Enrico Lavezzo, Barbara Di Camillo, Stefano Toppo, Liang Lan, Nemanja Djuric, Yuhong Guo, Slobodan Vucetic, Amos Bairoch, Michal Linial, Patricia C Babbitt, Steven E Brenner, Christine Orengo, Burkhard Rost, Sean D Mooney, and Iddo Friedberg. A large-scale evaluation of computational protein function prediction. *Nature Methods*, 10(3):221–227, March 2013. ISSN 1548-7091, 1548-7105. doi: 10.1038/nmeth.2340.
- [6] The UniProt Consortium, Alex Bateman, Maria-Jesus Martin, Sandra Orchard, Michele Magrane, Shadab Ahmad, Emanuele Alpi, Emily H Bowler-Barnett, Ramona Britto, Hema Bye-A-Jee, Austra Cukura, Paul Denny, Tunca Dogan, ThankGod Ebenezer, Jun Fan, Penelope Garmiri, Leonardo Jose Da Costa Gonzales, Emma Hatton-Ellis, Abdulrahman Hussein,

Alexandr Ignatchenko, Giuseppe Insana, Rizwan Ishtiaq, Vishal Joshi, Dushyanth Jyothi, Swaathi Kandasamy, Antonia Lock, Aurelien Luciani, Marija Lugaric, Jie Luo, Yvonne Lussi, Alistair MacDougall, Fabio Madeira, Mahdi Mahmoudy, Alok Mishra, Katie Moulang, Andrew Nightingale, Sangya Pundir, Guoying Qi, Shriya Raj, Pedro Raposo, Daniel L Rice, Rabie Saidi, Rafael Santos, Elena Speretta, James Stephenson, Prabhat Tootoo, Edward Turner, Nidhi Tyagi, Preethi Vasudev, Kate Warner, Xavier Watkins, Rossana Zaru, Hermann Zellner, Alan J Bridge, Lucila Aimò, Ghislaine Argoud-Puy, Andrea H Auchincloss, Kristian B Axelsen, Parit Bansal, Delphine Baratin, Teresa M Batista Neto, Marie-Claude Blatter, Jerven T Bolleman, Emmanuel Boutet, Lionel Breuza, Blanca Cabrera Gil, Cristina Casals-Casas, Kamal Chikh Echioukh, Elisabeth Coudert, Beatrice Cuche, Edouard De Castro, Anne Estreicher, Maria L Famiglietti, Marc Feuermann, Elisabeth Gasteiger, Pascale Gaudet, Sebastien Gehant, Vivienne Gerritsen, Arnaud Gos, Nadine Gruaz, Chantal Hulo, Nevila Hyka-Nouspikel, Florence Jungo, Arnaud Kerhornou, Philippe Le Mercier, Damien Lieberherr, Patrick Masson, Anne Morgat, Venkatesh Muthukrishnan, Salvo Paesano, Ivo Pedruzzi, Sandrine Pilbout, Lucille Pourcel, Sylvain Poux, Monica Pozzato, Manuela Pruess, Nicole Redaschi, Catherine Rivoire, Christian J A Sigrist, Karin Sonesson, Shyamala Sundaram, Cathy H Wu, Cecilia N Arighi, Leslie Arminski, Chuming Chen, Yongxing Chen, Hongzhan Huang, Kati Laiho, Peter McGarvey, Darren A Natale, Karen Ross, C R Vinayaka, Qinghua Wang, Yuqi Wang, and Jian Zhang. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1): D523–D531, January 2023. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkac1052.

- [7] Morgan N. Price, Kelly M. Wetmore, R. Jordan Waters, Mark Callaghan, Jayashree Ray, Hualan Liu, Jennifer V. Kuehl, Ryan A. Melnyk, Jacob S. Lamson, Yumi Suh, Hans K. Carlson, Zuelma Esquivel, Harini Sadeeshkumar, Romy Chakraborty, Grant M. Zane, Benjamin E. Rubin, Judy D. Wall, Axel Visel, James Bristow, Matthew J. Blow, Adam P. Arkin, and Adam M. Deutschbauer. Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature*, 557(7706):503–509, May 2018. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-018-0124-0.
- [8] Georgios A. Pavlopoulos, Fotis A. Baltoumas, Sirui Liu, Oguz Selvitopi, Antonio Pedro Camargo, Stephen Nayfach, Ariful Azad, Simon Roux, Lee Call, Natalia N. Ivanova, I. Min Chen, David Paez-Espino, Evangelos Karatzas, Novel Metagenome Protein Families Consortium, Silvia G. Acinas, Nathan Ahlgren, Graeme Attwood, Petr Baldrian, Timothy Berry, Jennifer M. Bhatnagar, Devaki Bhaya, Kay D. Bidle, Jeffrey L. Blanchard, Eric S. Boyd, Jennifer L. Bowen, Jeff Bowman, Susan H. Brawley, Eoin L. Brodie, Andreas Brune, Donald A. Bryant, Alison Buchan, Hinsby Cadillo-Quiroz, Barbara J. Campbell, Ricardo Cavicchioli, Peter F. Chuckran, Maureen Coleman, Sean Crowe, Daniel R. Colman, Cameron R. Currie, Jeff Dangl, Nathalie Delherbe, Vincent J. Deneff, Paul Dijkstra, Daniel D. Distel, Emiley Eloë-Fadrosh, Kirsten Fisher, Christopher Francis, Aaron Garoutte, Amelie Gaudin, Lena Gerwick, Filipa Godoy-Vitorino, Peter Guerra, Jiarong Guo, Mussie Y. Habteselassie, Steven J. Hallam, Roland Hatzenpichler, Ute Hentschel, Matthias Hess, Ann M. Hirsch, Laura A. Hug, Jenni Hultman, Dana E. Hunt, Marcel Huntemann, William P. Inskeep, Timothy Y. James, Janet Jansson, Eric R. Johnston, Marina Kalyuzhnaya, Charlene N. Kelly, Robert M. Kelly, Jonathan L. Klassen, Klaus Nüsslein, Joel E. Kostka, Steven Lindow, Erik Lilleskov, Mackenzie Lynes, Rachel Mackelprang, Francis M. Martin, Olivia U. Mason, R. Michael McKay, Katherine McMahon, David A. Mead, Monica Medina, Laura K. Meredith, Thomas Mock, William W. Mohn, Mary Ann Moran, Alison Murray, Josh D. Neufeld, Rebecca Neumann, Jeanette M. Norton, Laila P. Partida-Martinez, Nicole Pietrasiak, Dale Pelletier, T. B. K. Reddy, Brandi Kiel Reese, Nicholas J. Reichart, Rebecca Reiss, Mak A. Saito, Daniel P. Schachtman, Rekha Seshadri, Ashley Shade, David Sherman, Rachel Simister, Holly Simon, James Stegen, Ramunas Stepanauskas, Matthew Sullivan, Dawn Y. Sumner, Hanno Teeling, Kimberlee Thamatrakoln, Kathleen Treseder, Susannah Tringe, Parag Vaishampayan, David L. Valentine, Nicholas B. Waldo, Mark P. Waldrop, David A. Walsh, David M. Ward, Michael Wilkins, Thea Whitman, Jamie Woollet, Tanja Woyke, Ioannis Iliopoulos, Konstantinos Konstantinidis, James M. Tiedje, Jennifer Pett-Ridge, David Baker, Axel Visel, Christos A. Ouzounis, Sergey Ovchinnikov, Aydin Buluç, and Nikos C. Kyrpides. Unraveling the functional dark matter through global metagenomics. *Nature*, 622(7983):594–602, October 2023. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-023-06583-7.

- [9] Gavin Ayres, Geraldene Munsamy, Michael Heinzinger, Noelia Ferruz, Kevin Yang, and Philipp Lorenz. HiFi-NN annotates the microbial dark matter with Enzyme Commission numbers. 2024.
- [10] Tianhao Yu, Aashutosh Girish Boob, Michael J. Volk, Xuan Liu, Haiyang Cui, and Huimin Zhao. Machine learning-enabled retrobiosynthesis of molecules. *Nature Catalysis*, 6(2): 137–151, February 2023. ISSN 2520-1158. doi: 10.1038/s41929-022-00909-w.
- [11] Shubhanshu Jain, Felipe Ospina, and Stephan C. Hammer. A New Age of Biocatalysis Enabled by Generic Activation Modes. *JACS Au*, page jacsau.4c00247, May 2024. ISSN 2691-3704, 2691-3704. doi: 10.1021/jacsau.4c00247.
- [12] Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nature Reviews Molecular Cell Biology*, 10:866–876, 2009. doi: 10.1038/nrm2805.
- [13] Jason Yang, Francesca-Zhoufan Li, and Frances H. Arnold. Opportunities and Challenges for Machine Learning-Assisted Enzyme Engineering. *ACS Central Science*, page acscentsci.3c01275, February 2024. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.3c01275.
- [14] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 1990.
- [15] Robert D. Finn, Jody Clements, William Arndt, Benjamin L. Miller, Travis J. Wheeler, Fabian Schreiber, Alex Bateman, and Sean R. Eddy. HMMER web server: 2015 update. *Nucleic Acids Research*, 43(W1):W30–W38, July 2015. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkv397.
- [16] Michel Van Kempen, Stephanie S. Kim, Charlotte Tumescheit, Milot Mirdita, Jeongjae Lee, Cameron L.M. Gilchrist, Johannes Söding, and Martin Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42:243–246, 2023. doi: 10.1101/2022.02.07.479398.
- [17] Cameron L M Gilchrist, Milot Mirdita, and Martin Steinegger. Multiple Protein Structure Alignment at Scale with FoldMason. 2024.
- [18] Petr Kouba, Pavel Kohout, Faraneh Haddadi, Anton Bushuiev, Raman Samusevich, Jiri Sedlar, Jiri Damborsky, Tomas Pluskal, Josef Sivic, and Stanislav Mazurenko. Machine Learning-Guided Protein Engineering. *ACS Catalysis*, 13(21):13863–13895, October 2023. ISSN 2155-5435, 2155-5435. doi: 10.1021/acscatal.3c02743.
- [19] Kadina E. Johnston, Clara Fannjiang, Bruce J. Wittmann, Brian L. Hie, Kevin K. Yang, and Zachary Wu. Machine Learning for Protein Engineering, May 2023. arXiv:2305.16634 [q-bio].
- [20] Prabakaran Ramakrishnan and Yana Bromberg. Functional profiling of the sequence stockpile: a review and assessment of in silico prediction tools, July 2023.
- [21] Jae Yong Ryu, Hyun Uk Kim, and Sang Yup Lee. Deep learning enables high-quality and high-throughput prediction of enzyme commission numbers. *Proceedings of the National Academy of Sciences*, 116(28):13996–14001, July 2019. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1821905116.
- [22] Tianhao Yu, Haiyang Cui, Jianan Canal Li, Yunan Luo, Guangde Jiang, and Huimin Zhao. Enzyme function prediction using contrastive learning. 379(6639):1358–1363, 2023. doi: 10.1126/science.adf2465.
- [23] Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolk, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, 12(1):1–14, May 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23303-9.

- [24] Maxwell L. Bileschi, David Belanger, Drew H. Bryant, Theo Sanderson, Brandon Carter, D. Sculley, Alex Bateman, Mark A. DePristo, and Lucy J. Colwell. Using deep learning to annotate the protein universe. *Nature Biotechnology*, 40(6):932–937, June 2022. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-021-01179-w.
- [25] Theo Sanderson, Maxwell L Bileschi, David Belanger, and Lucy J Colwell. ProteInfer, deep neural networks for protein functional inference. *eLife*, 12:e80942, February 2023. ISSN 2050-084X. doi: 10.7554/eLife.80942.
- [26] Peter Shaw, Bhaskar Gurram, David Belanger, Andreea Gane, Maxwell L Bileschi, Lucy J Colwell, Kristina Toutanova, and Ankur P Parikh. ProtEx: A Retrieval-Augmented Approach for Protein Function Prediction. 2024.
- [27] Judith Bernett, David B. Blumenthal, Dominik G. Grimm, Florian Haselbeck, Roman Joeres, Olga V. Kalinina, and Markus List. Guiding questions to avoid data leakage in biological machine learning applications. *Nature Methods*, 21(8):1444–1453, August 2024. ISSN 1548-7091, 1548-7105. doi: 10.1038/s41592-024-02362-y.
- [28] Olga Khersonsky and Dan S. Tawfik. Enzyme Promiscuity: A Mechanistic and Evolutionary Perspective. *Annual Review of Biochemistry*, 79(1):471–505, June 2010. ISSN 0066-4154, 1545-4509. doi: 10.1146/annurev-biochem-030409-143718.
- [29] Valerie De Crecy-Lagard, Raquel Dias, Iddo Friedberg, Yifeng Yuan, and Manal Swairjo. Limitations of Current Machine-Learning Models in Predicting Enzymatic Functions for Uncharacterized Proteins. *bioRxiv*, July 2024. doi: 10.1101/2024.07.01.601547.
- [30] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik L L Sonnhammer, Silvio C E Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, Robert D Finn, and Alex Bateman. Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419, January 2021. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa913.
- [31] Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O’Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Židek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, May 2024. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-024-07487-w.
- [32] Mihaly Varadi, Damian Bertoni, Paulyna Magana, Urmila Paramval, Ivanna Pidruchna, Malarvizhi Radhakrishnan, Maxim Tslenkov, Sreenath Nair, Milot Mirdita, Jingsi Yeo, Oleg Kovalevskiy, Kathryn Tunyasuvunakool, Agata Laydon, Augustin Židek, Hamish Tomlinson, Dhavanthi Hariharan, Josh Abrahamson, Tim Green, John Jumper, Ewan Birney, Martin Steinegger, Demis Hassabis, and Sameer Velankar. AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research*, 52(D1):D368–D375, January 2024. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkad1011.
- [33] Antje Chang, Lisa Jeske, Sandra Ulbrich, Julia Hofmann, Julia Koblitz, Ida Schomburg, Meina Neumann-Schaal, Dieter Jahn, and Dietmar Schomburg. BRENDA, the ELIXIR core data resource in 2021: new developments and updates. *Nucleic Acids Research*, 49(D1): D498–D508, January 2021. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkaa1025.
- [34] Parit Bansal, Anne Morgat, Kristian B Axelsen, Venkatesh Muthukrishnan, Elisabeth Coudert, Lucila Aimò, Nevila Hyka-Nouspikel, Elisabeth Gasteiger, Arnaud Kerhornou, Teresa Batista Neto, Monica Pozzato, Marie-Claude Blatter, Alex Ignatchenko, Nicole Redaschi, and Alan Bridge. Rhea, the reaction knowledgebase in 2022. *Nucleic Acids Research*, 50(D1):D693–D700, January 2022. ISSN 0305-1048, 1362-4962. doi: 10.1093/nar/gkab1016.

- [35] Minoru Kanehisa, Miho Furumichi, Yoko Sato, Masayuki Kawashima, and Mari Ishiguro-Watanabe. Kegg for taxonomy-based analysis of pathways and genomes. *Nucleic Acids Research*, 51(D1):D587–D592, January 2023. ISSN 0305-1048. doi: 10.1093/nar/gkac963.
- [36] Daniel Probst, Matteo Manica, Yves Gaetan Nana Teukam, Alessandro Castrogiovanni, Federico Paratore, and Teodoro Laino. Biocatalysed synthesis planning using data-driven learning. *Nature Communications*, 13(1):964, February 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-28536-w.
- [37] Esther Heid, Daniel Probst, William H. Green, and Georg K. H. Madsen. EnzymeMap: curation, validation and data-driven prediction of enzymatic reactions. *Chemical Science*, 14(48):14229–14242, 2023. ISSN 2041-6520, 2041-6539. doi: 10.1039/D3SC02048G.
- [38] William Finnigan, Lorna J. Hepworth, Sabine L. Flitsch, and Nicholas J. Turner. RetroBioCat as a computer-aided synthesis planning tool for biocatalytic reactions and cascades. *Nature Catalysis*, 4(2):98–104, January 2021. ISSN 2520-1158. doi: 10.1038/s41929-020-00556-z.
- [39] William Finnigan, Max Lubberink, Lorna J. Hepworth, Joan Citoler, Ashley P. Matthey, Grayson J. Ford, Jack Sangster, Sebastian C. Cosgrove, Bruna Zucoloto Da Costa, Rachel S. Heath, Thomas W. Thorpe, Yuqi Yu, Sabine L. Flitsch, and Nicholas J. Turner. RetroBioCat Database: A Platform for Collaborative Curation and Automated Meta-Analysis of Biocatalysis Data. *ACS Catalysis*, 13(17):11771–11780, September 2023. ISSN 2155-5435, 2155-5435. doi: 10.1021/acscatal.3c01418.
- [40] Itai Levin, Mengjie Liu, Christopher A. Voigt, and Connor W. Coley. Merging enzymatic and synthetic chemistry with computational synthesis planning. *Nature Communications*, 13(1):7747, December 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-35422-y.
- [41] Pablo Carbonell, Jerry Wong, Neil Swainston, Eriko Takano, Nicholas J Turner, Nigel S Scrutton, Douglas B Kell, Rainer Breitling, and Jean-Loup Faulon. Selenzyme: enzyme selection tool for pathway design. *Bioinformatics*, 34(12):2153–2154, June 2018. ISSN 1367-4803, 1367-4811. doi: 10.1093/bioinformatics/bty065.
- [42] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating Protein Transfer Learning with TAPE. June 2019.
- [43] Christian Dallago, Jody Mou, Kadina E. Johnston, Bruce J. Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K. Yang. FLIP: Benchmark tasks in fitness landscape inference for proteins. Technical report, November 2021.
- [44] Pascal Notin, Aaron W Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Hansen Spinner, Nathan Rollins, Ada Shaw, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Rose Orenbuch, Yarin Gal, and Debora S Marks. ProteinGym: Large-Scale Benchmarks for Protein Fitness Prediction and Design. 2023.
- [45] Peter G Mikhael, Itamar Chinn, and Regina Barzilay. Graph-based forward synthesis prediction of biocatalyzed reactions. 2024.
- [46] Philippe Schwaller, Teodoro Laino, Théophile Gaudin, Peter Bolgar, Christopher A. Hunter, Costas Bekas, and Alpha A. Lee. Molecular Transformer: A Model for Uncertainty-Calibrated Chemical Reaction Prediction. *ACS Central Science*, 5(9):1572–1583, September 2019. ISSN 2374-7943, 2374-7951. doi: 10.1021/acscentsci.9b00576.
- [47] David Kreutter, Philippe Schwaller, and Jean-Louis Reymond. Predicting enzymatic reactions with a molecular transformer. *Chemical Science*, 12(25):8648–8659, 2021. ISSN 2041-6520, 2041-6539. doi: 10.1039/D1SC02362D.
- [48] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), April 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2016239118.

- [49] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: <https://doi.org/10.1126/science.ade2574>.
- [50] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life’s Code Through Self-Supervised Learning. 14(8):29, 2021.
- [51] Louis Callum Butler Robinson, Timothy Atkinson, Liviu Copoiu, Patrick Bordes, Thomas Pierrot, and Thomas Barrett. Contrasting Sequence with Structure: Pre-training Graph Representations with PLMs. preprint, Bioinformatics, December 2023.
- [52] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised Contrastive Learning, March 2021. arXiv:2004.11362 [cs, stat].
- [53] Hongyu Duan, Ziyang Li, Yixuan Wu, Wen Chen, and Li C Xia. Predicting Enzyme Functions Using Contrastive Learning with Hierarchical Enzyme Structure Information. 2024.
- [54] Yaan J Jang. Accurate prediction of protein function using statistics-informed graph networks. *Nature Communications*, 15:6601, 2024.
- [55] Yidong Song, Qianmu Yuan, Sheng Chen, Yuansong Zeng, Huiying Zhao, and Yuedong Yang. Accurately predicting enzyme functions through geometric graph learning on ESMFold-predicted structures. *Nature Communications*, 15(1):8180, September 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-52533-w.
- [56] Lei Zheng, Bowen Li, Siqi Xu, Junnan Chen, and Guanxiang Liang. FEDKEA: Enzyme function prediction with a large pretrained protein language model and distance-weighted k-nearest neighbor. *bioRxiv*, 2024.
- [57] Ron S. Boger, Seyone Chithrananda, Anastasios N. Angelopoulos, Peter Hyungjun Yoon, Michael I Jordan, and Jennifer A. Doudna. Functional protein mining with conformal guarantees. June 2024. doi: 10.1101/2024.06.27.601042.
- [58] Liang Hong, Zhihang Hu, Siqi Sun, Xiangru Tang, Jiuming Wang, Qingxiong Tan, Liangzhen Zheng, Sheng Wang, Sheng Xu, Irwin King, Mark Gerstein, and Yu Li. Fast, sensitive detection of protein homologs using deep dense retrieval. *Nature Biotechnology*, August 2024. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-024-02353-6.
- [59] Jin Su, Xibin Zhou, Xuting Zhang, and Fajie Yuan. ProTrek: Navigating the Protein Universe through Tri-Modal Contrastive Learning. 2024.
- [60] Tymor Hamamsy, Meet Barot, James T. Morton, Martin Steinegger, Richard Bonneau, and Kyunghyun Cho. Learning sequence, structure, and function representations of proteins with language models. November 2023. doi: 10.1101/2023.11.26.568742.
- [61] Tymor Hamamsy, James T. Morton, Robert Blackwell, Daniel Berenberg, Nicholas Carriero, Vladimir Gligorijevic, Charlie E. M. Strauss, Julia Koehler Leman, Kyunghyun Cho, and Richard Bonneau. Protein remote homology detection and structural alignment using deep learning. *Nature Biotechnology*, September 2023. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-023-01917-2.
- [62] Max James Campbell. VIPER: A General Model for Prediction of Enzyme Substrates. June 2024. doi: 10.1101/2024.06.21.599972.
- [63] Alexander Kroll, Sahasra Ranjan, Martin K. M. Engqvist, and Martin J. Lercher. A general model to predict small molecule substrates of enzymes based on machine and deep learning. *Nature Communications*, 14(1):2787, May 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-38347-2.



- [64] Alexandra Paton, Daniil Boiko, Jonathan Perkins, Nicholas Cemalovic, Thiago Reschützegeger, Gabe Gomes, and Alison Narayan. Generation of connections between protein sequence space and chemical space to enable a predictive model for biocatalysis. *chemRxiv*, October 2024. doi: 10.26434/chemrxiv-2024-w4dtr.
- [65] Qiang Zhang, Keyang Ding, Tianwen Lyv, Xinda Wang, Qingyu Yin, Yiwen Zhang, Jing Yu, Yuhao Wang, Xiaotong Li, Zhuoyi Xiang, Xiang Zhuang, Zeyuan Wang, Ming Qin, Mengyao Zhang, Jinlu Zhang, Jiyu Cui, Renjun Xu, Hongyang Chen, Xiaohui Fan, Huabin Xing, and Huajun Chen. Scientific Large Language Models: A Survey on Biological & Chemical Domains, January 2024. arXiv:2401.14656 [cs].
- [66] Mayk Caldas Ramos, Christopher J. Collison, and Andrew D. White. A Review of Large Language Models and Autonomous Agents in Chemistry, June 2024. arXiv:2407.01603 [physics].
- [67] Eli M. Carrami and Sahand Sharifzadeh. PQA: Zero-shot Protein Question Answering for Free-form Scientific Enquiry with Large Language Models, February 2024. arXiv:2402.13653 [cs].
- [68] Zeyuan Wang, Qiang Zhang, Keyan Ding, Ming Qin, Xiang Zhuang, Xiaotong Li, and Huajun Chen. InstructProtein: Aligning Human and Protein Language via Knowledge Instruction, October 2023. arXiv:2310.03269 [cs, q-bio].
- [69] Xiang Zhuang, Keyan Ding, Tianwen Lyu, Yinuo Jiang, Xiaotong Li, Zhuoyi Xiang, Zeyuan Wang, Ming Qin, Kehua Feng, Jake Wang, Qiang Zhang, and Huajun Chen. InstructBioMol: Advancing Biomolecule Understanding and Design Following Human Instructions. *bioRxiv*, October 2024. arXiv:2410.07919 [cs, q-bio].
- [70] Yijia Xiao, Edward Sun, Yiqiao Jin, Qifan Wang, and Wei Wang. ProteinGPT: Multimodal LLM for Protein Property Prediction and Structure Understanding. *arXiv*, August 2024. arXiv:2408.11363 [cs, q-bio].
- [71] Mingjia Huo, Han Guo, Xingyi Cheng, Digvijay Singh, Hamidreza Rahmani, Shen Li, Philipp Gerlof, Trey Ideker, Danielle A Grotjahn, Elizabeth Villa, Le Song, and Pengtao Xie. Multi-Modal Large Language Model Enables Protein Function Prediction. *arXiv*, 2024.
- [72] Andres M. Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D. White, and Philippe Schwaller. ChemCrow: Augmenting large-language models with chemistry tools, October 2023. arXiv:2304.05376 [physics, stat].
- [73] Jon M. Laurent, Joseph D. Janizek, Michael Ruzo, Michaela M. Hinks, Michael J. Hammerling, Siddharth Narayanan, Manvitha Ponnampati, Andrew D. White, and Samuel G. Rodrigues. LAB-Bench: Measuring Capabilities of Language Models for Biology Research, July 2024. arXiv:2407.10362 [cs].
- [74] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A Simple Framework for Contrastive Learning of Visual Representations. *arXiv:2002.05709 [cs, stat]*, June 2020. arXiv: 2002.05709.
- [75] Shengchao Liu, Weitao Du, Zhiming Ma, Hongyu Guo, and Jian Tang. A Group Symmetric Stochastic Differential Equation Model for Molecule Multi-modal Pretraining. 2023.
- [76] Shengchao Liu, Weili Nie, Chengpeng Wang, Jiarui Lu, Zhuoran Qiao, Ling Liu, Jian Tang, Chaowei Xiao, and Animashree Anandkumar. Multi-modal molecule structure–text model for text-based retrieval and editing. *Nature Machine Intelligence*, 5(12):1447–1457, December 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00759-6.
- [77] Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, Jian Tang, Hongyu Guo, and Anima Anandkumar. A text-guided protein design framework, December 2023. arXiv preprint arXiv:2302.04611.
- [78] Kevin E Wu, Howard Chang, and James Zou. ProteinCLIP: enhancing protein language models with natural language. 2024.

- [79] Peter G. Mikhael, Itamar Chinn, and Regina Barzilay. CLIPZyme: Reaction-Conditioned Virtual Screening of Enzymes, February 2024. arXiv:2402.06748 [q-bio].
- [80] Rohit Singh, Samuel Sledzieski, Bryan Bryson, Lenore Cowen, and Bonnie Berger. Contrastive learning in protein language space predicts interactions between drugs and protein targets. *Proceedings of the National Academy of Sciences*, 120(24):e2220778120, June 2023. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2220778120.
- [81] Sarah Alamdari, Nitya Thakkar, Alex X Lu, Nicolo Fusi, Ava P Amini, and Kevin K Yang. Protein generation with evolutionary diffusion: sequence is all you need. 2023.
- [82] Geraldene Munsamy, Ramiro Illanes-Vicioso, Silvia Funcillo, Sebastian Lindner, Gavin Ayres, Lesley S Sheehan, Steven Moss, Ulrich Eckhard, Philipp Lorenz, and Noelia Ferruz. Conditional language models enable the efficient design of proficient enzymes. 2024.
- [83] Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, January 2023. ISSN 1087-0156, 1546-1696. doi: 10.1038/s41587-022-01618-2.
- [84] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, July 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32007-7.
- [85] Jason Yang, Aadyot Bhatnagar, Jeffrey A. Ruffolo, and Ali Madani. Conditional Enzyme Generation Using Protein Language Models with Adapters. *arXiv*, October 2024. arXiv:2410.03634 null.
- [86] Daniel Probst, Philippe Schwaller, and Jean-Louis Reymond. Reaction classification and yield prediction using the differential reaction fingerprint DRFP. *Digital Discovery*, 1(2):91–97, 2022. ISSN 2635-098X. doi: 10.1039/D1DD00006C.
- [87] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, 02 1999. ISSN 1741-0126. doi: 10.1093/protein/12.2.85.
- [88] Tess Richman, Elyssa Arnold, and Antony J. Williams. Curation of a list of chemicals in biosolids from EPA National Sewage Sludge Surveys & Biennial Review Reports. *Scientific Data*, 9(1):180, April 2022. ISSN 2052-4463. doi: 10.1038/s41597-022-01267-9.
- [89] Raman Samusevich, Teo Hebra, Roman Bushuiev, Anton Bushuiev, Tereza Čalounová, Helena Smrčková, Ratthachat Chatpatanasiri, Jonáš Kulhánek, Milana Perković, Martin Engst, Adéla Tajovská, Josef Sivic, and Tomáš Pluskal. Highly accurate discovery of terpene synthases powered by machine learning reveals functional terpene cyclization in Archaea, January 2024.
- [90] Raman Samusevich, Téo Hebra, Roman Bushuiev, Anton Bushuiev, Jonáš Kulhánek, Tereza Čalounová, Milana Perković, Adéla Tajovská, Josef Sivic, and Tomáš Pluskal. Discovery and Characterization of Terpene Synthases Powered by Machine Learning. 2024.
- [91] Keisuke Hirota, Felix Salim, and Takuji Yamada. DeepES: Deep learning-based enzyme screening to identify orphan enzyme genes. 2024.
- [92] Philippe Schwaller, Daniel Probst, Alain C. Vaucher, Vishnu H. Nair, David Kreutter, Teodoro Laino, and Jean-Louis Reymond. Mapping the space of chemical reactions using attention-based neural networks. *Nature Machine Intelligence*, 3(2):144–152, January 2021. ISSN 2522-5839. doi: 10.1038/s42256-020-00284-w.
- [93] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv:1810.04805 [cs]*, May 2019. arXiv: 1810.04805.

- [94] Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, Midori A. Harris, David P. Hill, Laurie Issel-Tarver, Andrew Kasarskis, Suzanna Lewis, John C. Matese, Joel E. Richardson, Martin Ringwald, Gerald M. Rubin, and Gavin Sherlock. Gene Ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29, May 2000. ISSN 1061-4036, 1546-1718. doi: 10.1038/75556.
- [95] Iz Beltagy, Kyle Lo, and Arman Cohan. SciBERT: A Pretrained Language Model for Scientific Text, September 2019. arXiv:1903.10676 [cs].
- [96] Haonan Duan, Marta Skreta, Leonardo Cotta, Ella Miray Rajaonson, Nikita Dhawan, Alán Aspuru-Guzik, and Chris J Maddison. Boosting the Predictive Power of Protein Representations with a Corpus of Text Annotations. 2024.
- [97] Chenqing Hua, Bozita Zhong, Sitao Luan, Liang Hong, Guy Wolf, Doina Precup, and Shuangjia Zheng. Reactzyme: A Benchmark for Enzyme-Reaction Prediction. *NeurIPS*, December 2024.
- [98] Yong Liu, Chenqing Hua, Tao Zeng, Jiahua Rao, Zhongyue Zhang, Ruibo Wu, Connor W Coley, and Shuangjia Zheng. EnzymeCAGE: A Geometric Foundation Model for Enzyme Retrieval with Evolutionary Insights. *bioRxiv*, 2024.
- [99] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, November 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. Number: 11 Publisher: Nature Publishing Group.
- [100] Aditi T Merchant, Samuel H King, Eric Nguyen, and Brian L Hie. Semantic mining of functional de novo genes from a genomic language model. *bioRxiv*, 2024.
- [101] Mohammadamin Tavakoli, Alexander Shmakov, Francesco Ceccarelli, and Pierre Baldi. Rxn Hypergraph: a Hypergraph Attention Model for Chemical Reaction Representation, January 2022. arXiv:2201.01196 [physics].
- [102] Samir Char, Nathaniel Corley, Sarah Alamdari, Kevin K Yang, and Ava P Amini. ProtNote: a multimodal method for protein-function annotation. *bioRxiv*, 2024.
- [103] David Baker and George Church. Protein design meets biosecurity. *Science*, 383(6681): 349–349, January 2024. ISSN 0036-8075, 1095-9203. doi: 10.1126/science.ado1671.
- [104] Shengchao Liu, Hanchen Wang, Weiyang Liu, Joan Lasenby, Hongyu Guo, and Jian Tang. Pre-training molecular graph representation with 3d geometry. In *International Conference on Learning Representations*, 2022.

## Checklist

1. For all authors...
  - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
  - (b) Did you describe the limitations of your work? [Yes] In the Appendix 6
  - (c) Did you discuss any potential negative societal impacts of your work? [Yes] The potential negative societal impacts are also discussed more extensively elsewhere
  - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
  - (a) Did you state the full set of assumptions of all theoretical results? [N/A] we did not have theoretical results
  - (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments (e.g. for benchmarks)...
  - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Github link was provided
  - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
  - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] Models we trained do not have high inherent stochasticity
  - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] in the github link
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
  - (a) If your work uses existing assets, did you cite the creators? [Yes] all datasets were cited
  - (b) Did you mention the license of the assets? [No] Data we used is open sourced
  - (c) Did you include any new assets either in the supplemental material or as a URL? [Yes] In the provided github link
  - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [No] Data we used is open sourced
  - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No] The data does not contain any personal information
5. If you used crowdsourcing or conducted research with human subjects...
  - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
  - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
  - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

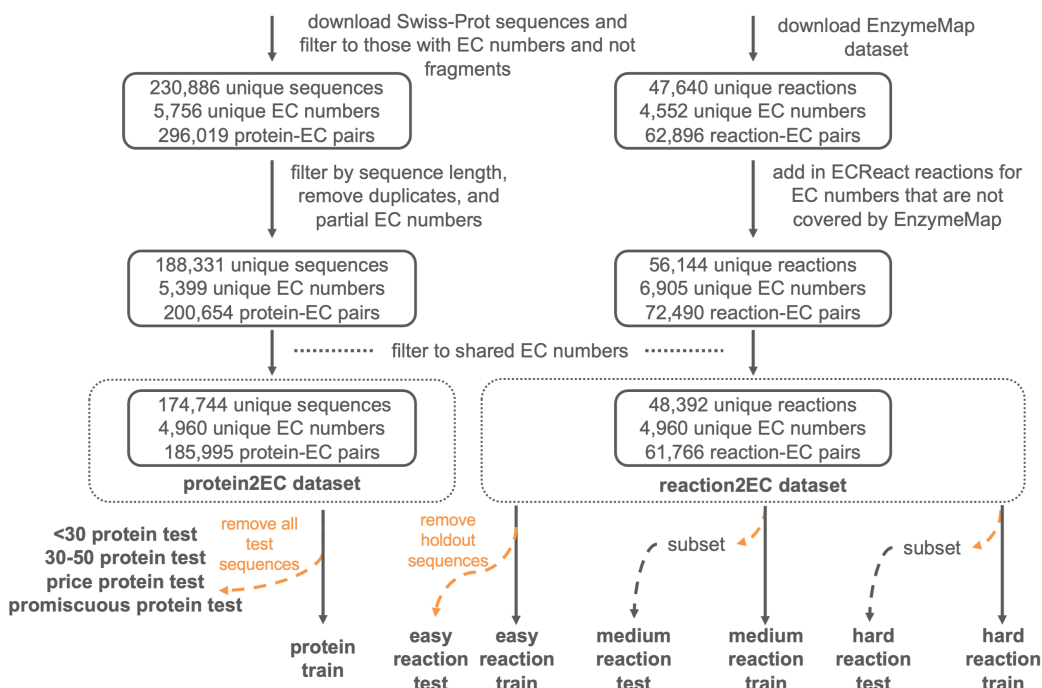


Figure A.1: Workflow used to process the datasets containing EC numbers, protein sequences, and reactions used in this study.

## A Appendix

### A.1 Dataset Processing

Our data processing and splitting workflow is shown in Figure A.1. Protein sequence data, paired to EC number(s), were downloaded from UniProt, selecting only reviewed sequences in Swiss-Prot, resulting in a total of 571,282 sequences on the 13<sup>th</sup> of May 2024. These were filtered to only retain sequences with an EC number that were not annotated as fragments. After, sequences were filtered to length between 100 and 1024, partial EC numbers (containing a dash) were removed, duplicate entries (based on EC and sequence) were dropped, and only EC numbers with associated reactions were kept, leaving 200,654 sequence-EC pairs. We call this the protein2EC dataset.

Reaction data were downloaded from ECRreact and EnzymeMap. ECRreact contains 62,222 entries of reaction in SMILES as SMARTS strings format coupled with EC number from a range of data sources. EnzymeMap had 349,458 entries containing sequence-reaction-EC triplets with the reaction in several formats, SMILES as SMARTS string, an atom-level mapping of bond formation and breaking (mapped reaction), and reactions written as text based on IUPAC compound names. Starting with EnzymeMap, protein sequence information was dropped (leaving reaction-EC pairs), and duplicate reaction-EC pairs were removed leaving 62,896 reaction-EC pairs. EC numbers that existed in ECRreact but not in Enzyme map were added to EnzymeMap providing our reaction dataset of 72,490 reaction-EC pairs. We did this because EnzymeMap is a higher quality reaction dataset, with fewer incorrect entries. Note the ECRreact only contains reactions formatted as SMILES/SMARTS, not as mapped reactions. Together, we call this the reaction2EC dataset.

Finally, we filtered the protein2EC and reaction2EC datasets, such that only EC numbers present in both datasets are retained, leaving 61,766 reaction-EC pairs and 185,995 sequence-EC pairs, with 4,960 unique EC numbers. Clustering using MMseqs2 [99] was performed at 30%, 50%, 70%, and 90% sequence identity and included in the protein2EC dataset for downstream use. The distributions of the curated datasets are shown in Figure A.2.

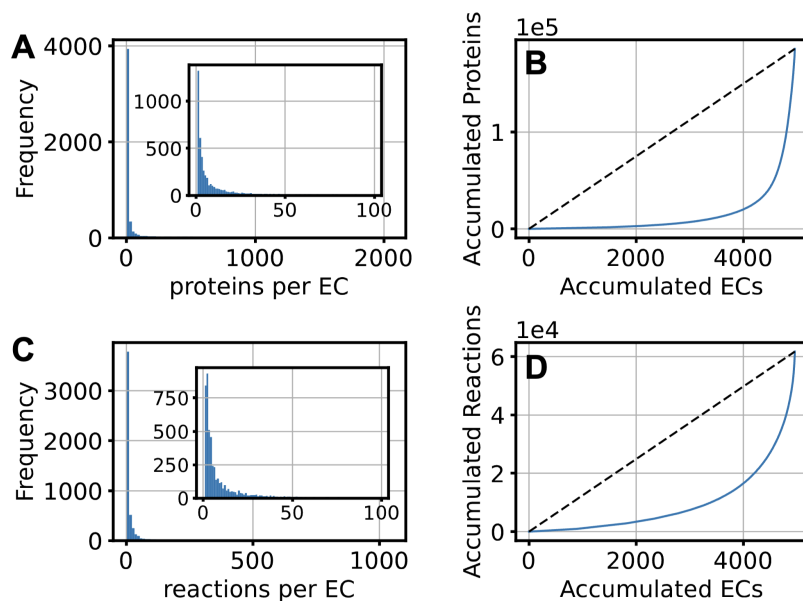


Figure A.2: Distribution of the protein2EC dataset, shown as (A) a histogram of proteins per EC and as (B) a cumulative distribution plot, analogous to a Lorenz curve. Distribution of the reaction2EC dataset, shown as (C) a histogram of reactions per EC and as (D) a cumulative distribution plot, analogous to a Lorenz curve. Both datasets are heavily skewed with most examples belonging to a few EC numbers, and many EC numbers only having one example. If samples were evenly distributed across ECs, they would follow the dashed line in (B) and (D).

## A.2 Dataset Splitting Details

First, we sought to create test sets capturing different types of out-of-domain generalization for Task 1, EC classification (visualized in Figure A.3A).

Using the results from clustering, we generated two test sets, with approximately <30% and 30-50% sequence identity, respectively, to sequences in the training set. We also only considered ECs at level 4 which had isolated clusters, in other words, an MMseqs2 cluster with only a single member. For every EC level 3 (X.X.X.-), we sampled up to three random ECs at level 4 (X.X.X.X), and then randomly selected a single isolated sequence from each class to add to the test set. This provided balanced test datasets across the functions at EC level 3. For this set, we also enforced that the sequences did not map to multiple EC numbers. Test sets were balanced across functions (e.g. EC number), rather than across their prevalence as protein sequences in nature, based on the idea that most synthesis planners and enzyme engineers would find more value in predictive ability across a broad range of functions.

Next, we looked for "promiscuous" enzymes, which in this work, we define as those annotated with multiple EC numbers. For our promiscuous test set, we only considered using promiscuous enzymes that had a combination of ECs that occurred at least twice. From this set we only took a single sample from each EC combination, and finally randomly selected a single entry from those which have ECs with high surprise levels (i.e. X.X.-.-, where variation occurs at the third EC level or higher). From the test sets, we did not include sequences that were labeled as belonging to part of a heterogeneous multi-complex protein.

Finally, we compiled a commonly misclassified test set from Price et al., filtered to sequences of length 100-1024, and dropped test sequences that are present in our protein2EC dataset. The pooled set of all proteins in the test sets were removed from our training data leaving 184,529 sequence-EC pairs for training. As a result of the holdout process, a few of the 4,960 EC numbers are missing from the train set, which should minimally affect performance on the test sets.

For Task 2, we sought to test reaction extrapolation, namely, retrieval of enzymes based on unseen reactions (visualized in Figure A.3B). For the easy and medium test sets, we randomly sampled an

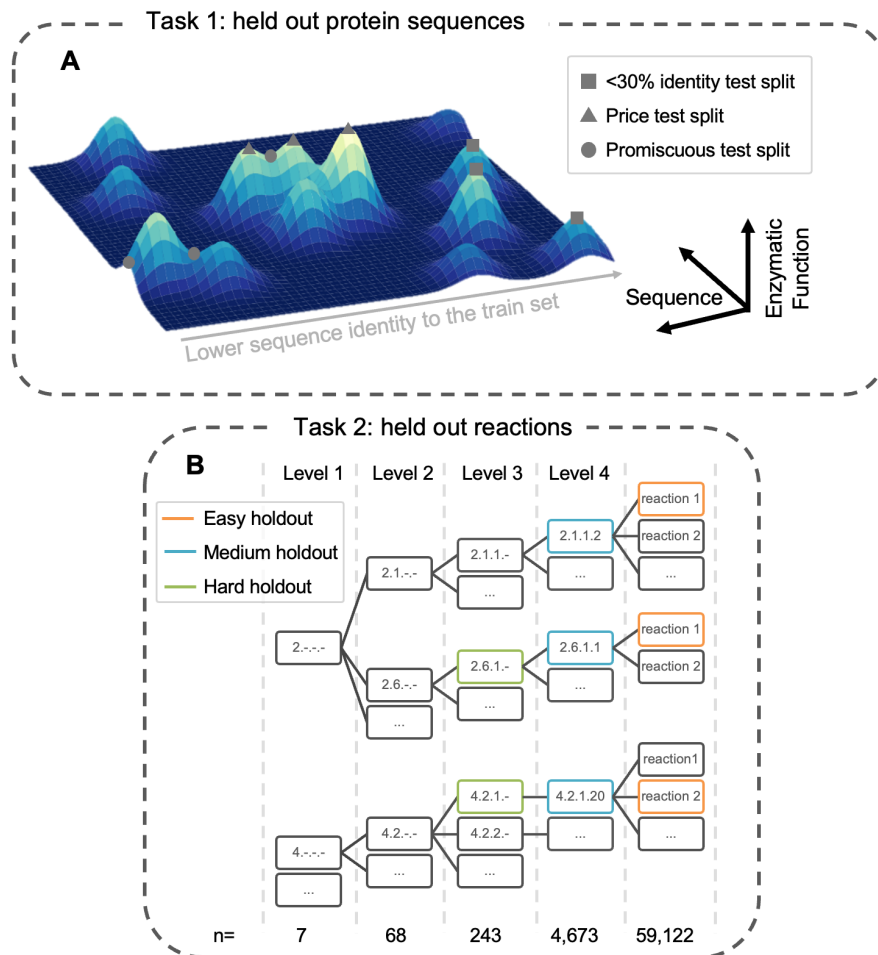


Figure A.3: (A) Visualization of the types of extrapolation found in the test splits for Task 1. (B) Visualization of how different subsets of reaction space are held out in the splits for Task 2.

EC level 4 EC (X.X.X.X) from every level three EC (X.X.X.-). From each of the held-out level 4 ECs, three random reactions were sampled to be in the test set. For the test set, we only considered (1) reactions that do not map to multiple EC numbers, (2) EC numbers with at least four reactions, and (3) reactions from EnzymeMap, which are higher fidelity and consequently have fewer misannotations.

For the easy set, only the test set reactions were held out. For the medium set, all reactions under the corresponding level 4 ECs were held out. Note that the easy and medium test sets are exactly the same, but the easy set has fewer held out reactions compared to the medium set, thus their training sets are different.

The hard set was evenly balanced across a random sample from EC level 2 (X.X.-.-) to ensure there was an even distribution across level 3 (EC3, X.X.X.-), from which 53 random EC3s are sampled, with all reactions under the 53 EC3s held out. 460 reactions associated with the 53 ECs at level 3 were used as the test set, which are evenly spread across the ECs at level 3, with up to three reactions from each EC number at level 4. Note we ensure the hard test reactions are shared with the easy and medium ones, when possible, to correlate the performance of our test splits. It should be noted that the hard train-test split is different from the easy and medium splits. For Task 2, all of the protein-EC pairs can be used during training.

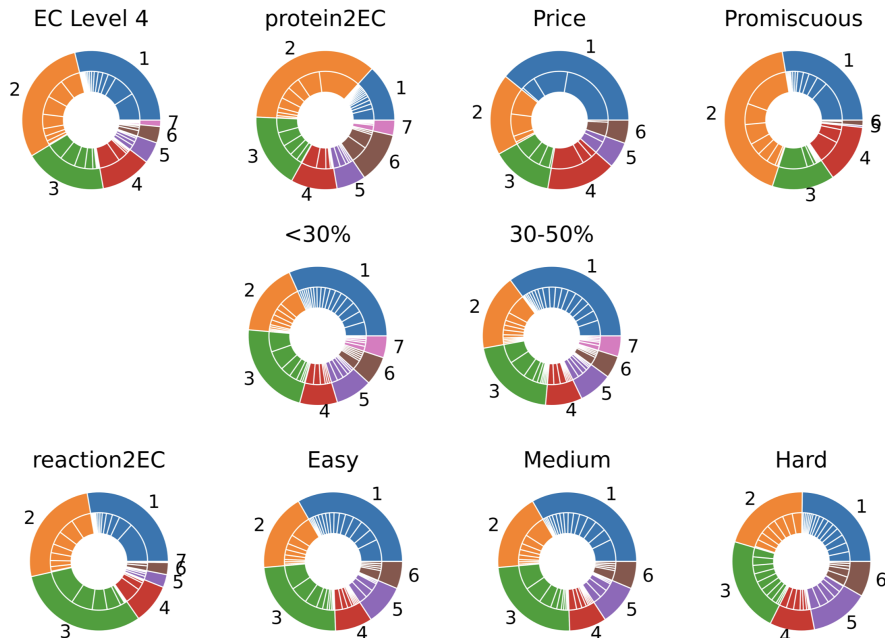


Figure A.4: Distribution of the EC numbers found in the test splits, and compared to a few reference datasets. Outer ring shows the level 1 category (X.-.-), and the inner ring shows the level 2 category (X.X.-). EC Level 4 refers the distribution of unique ECs at level 4 (X.X.X.X). protein2EC shows the distribution of all protein sequences in protein2EC across ECs, and reaction2EC shows the distribution of all reactions in reaction2EC across ECs.

### A.3 CREEP Model Details

All steps needed to reproduce CREEP training and downstream retrieval can be found at <https://github.com/jsunn-y/CARE/>.

During the pretraining stage, we follow the contrastive learning paradigm that maximizes mutual information between two views [104]. For each enzyme family (EC number)  $x$ , we extract three views: protein sequence  $x_p$  (protein), reactions smiles  $x_r$  (reaction), and textual description of EC number from gene ontology  $x_t$  (text) [94]. In CREEP, the goal is to maximize the mutual information between all three modalities. For illustration purposes, we use the example of protein-reaction contrastive learning, but the derivations for reaction-text and text-protein contrastive learning follow a similar approach.

We follow the method proposed in GraphMVP by adopting EBM-NCE to estimate mutual information between our modalities [104]. In our case, EBM-NCE aligns the protein sequence and reaction SMILES pairs for the same enzyme family and contrasts the pairs for different enzyme families simultaneously. The objective function is

$$\mathcal{L}_{\text{EBM-NCE}} = -\frac{1}{2} \left( \mathbb{E}_{\mathbf{x}_p, \mathbf{x}_r} [\log \sigma(E(\mathbf{x}_p, \mathbf{x}_r))] + \mathbb{E}_{\mathbf{x}_p, \mathbf{x}'_r} [\log(1 - \sigma(E(\mathbf{x}_p, \mathbf{x}'_r)))] \right) + \mathbb{E}_{\mathbf{x}_p, \mathbf{x}_r} [\log \sigma(E(\mathbf{x}_p, \mathbf{x}_r))] + \mathbb{E}_{\mathbf{x}'_p, \mathbf{x}_r} [\log(1 - \sigma(E(\mathbf{x}'_p, \mathbf{x}_r)))] \quad (1)$$

where  $\mathbf{x}_p$  and  $\mathbf{x}_r$  form the (protein sequence, reaction SMILES) pair for each reaction, and  $\mathbf{x}_{p'}$  and  $\mathbf{x}_{r'}$  are the negative samples which are produced by randomly sampling from a Gaussian distribution, which we use as an approximation for the empirical data distribution.  $E(\cdot)$  is the energy function with a flexible formulation, and we use the dot product as a metric for similarity within the latent space.

For our default CREEP model, we project representations to 256 dimensions in the shared latent space for all modalities. We train for 40 epochs, and in each epoch, we loop over each EC number and mine each batch such that all protein-reaction-text triplets come from different EC numbers. Note that we also use 50% sequence clustering to help select protein sequences with increased diversity.



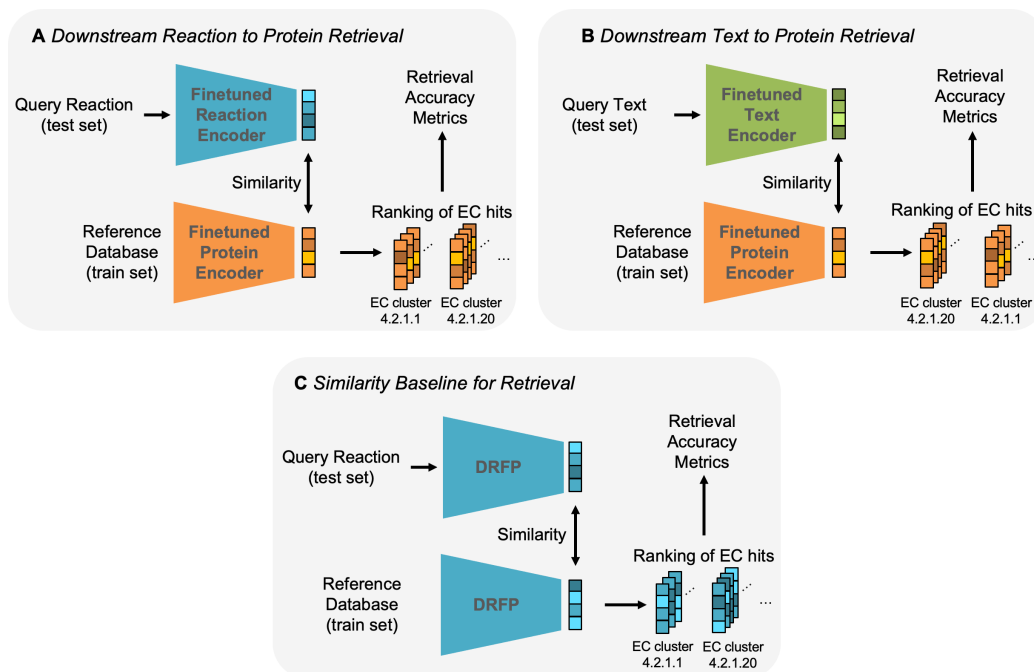


Figure A.5: Downstream retrieval using contrastively pretrained models. CREEP and CLIPZyme both utilize (A) reaction to protein retrieval. CREEP (w/ Text) can incorporate (B) text to protein retrieval, and consider the rankings from both (A) and (B). (C) The similarity baseline involves reaction to reaction retrieval using a fixed representation from DRFP [86].

Downstream retrieval using models such as CREEP and CLIPZyme is shown in Figure A.5. The standard workflow starts with a query reaction and ranks ECs by sorting based on distance from the reaction representation to the EC cluster centroids of protein sequence representations belonging to each EC number (Figure A.5A). For CREEP (w/ Text), a similar ranking can be determined by using text as a query (Figure A.5B), and the overall ranking can be calculated as an average of the two rankings. To reduce inference time, we used protein sequences clustered at 50% identity to calculate the protein representation centroids clustered by EC. Finally, the similarity baseline can be understood as sorting based on distance from a query reaction to reaction cluster centroids of reactions in the train set, clustered by EC (Figure A.5C).

#### A.4 Implementation Details for Benchmarking Methods

All implementation details for methods used in the study can be found at <https://github.com/jsunn-y/CARE/>. We focus on evaluating the accuracy of retrieval for  $k=1$ . A level 1 accuracy means the top level EC class has to be correct, for level 2 accuracy, both level 1 and level 2 have to be correct, and so on. Accuracy is calculated as the number of true answers over the total number of examples. An invalid EC number from language models (ChatGPT and Pika) is considered incorrect on every level. For the non-promiscuous splits, the accuracy for each entry in the test set is 1 or 0. For the promiscuous split in Task 1, for each entry, we enumerate over each true EC to find the maximum accuracy within the pool of predicted ECs, and the overall accuracy for the entry is the average of the accuracies for each true EC.

**ChatGPT.** ChatGPT was used for both Task 1 and Task 2, using the API and gpt-4o-mini model. We performed minimal prompt engineering, with our initial prompt being the most standard initial question: "give me the top N EC numbers associated with this amino acid sequence:" for Task 1. See Table A.1 for example responses. The query was modified to: "You are protein engineer capable of predicting EC numbers from the sequence alone. You are also a skilled programmer and able to execute the code necessary to predict an EC number when you can't use reason alone. Given a protein sequence you are able to determine the most likely enzyme class for a sequence because you

are that skilled. You don't give up when faced with a sequence you don't know, you will use tools to resolve the most likely enzyme sequence. You only return enzyme commission numbers in a comma separated list, no other text is returned, you have failed if you do not return the EC numbers. You only return the most likely EC number." This resulted in a  $k=1$  response, we found this more consistent then asking for a list of  $k$  results. Note that we do not do any filtering or data cleaning on the output of ChatGPT. Doing this could improve the accuracy, for example, in one case we identified in the hard set the output was EC4.6.1.18 which we scored as 0 despite the true EC being 4.6.1.2 for the reaction **UTP = 3',5'-cyclic UMP + diphosphate**. There are possibly other edge cases that we were unable to identify.

For Task 2, the query was maintained to be relatively consistent with the Task 1 query, and modified to: "Return the most likely EC number for this reaction: **naloxone + NAD(P)H = 6alpha-naloxol + NADP+**, which associates with the following text: **oxidoreductase; oxidoreductase, acting on CH-OH group of donors; oxidoreductase, acting on the CH-OH group of donors, NAD or NADP as acceptor; morphine 6-dehydrogenase**"., where a sample reaction has been bolded. Using the system prompt of: "You are protein engineer capable of predicting EC numbers from a combination of textual information and a reaction that corresponds to a specific protein. You are also a skilled programmer and able to execute the code necessary to predict an EC number when you can't use reason alone. Given a reaction and text information of an EC you are able to determine the most likely enzyme class for a reaction. You don't give up when faced with a reaction you don't know, you will use tools to resolve the most likely enzyme number. You only return enzyme commission numbers in a comma separated list, no other text is returned, you have failed if you do not return the EC numbers. You only return the most likely EC number." For Task 2 without the reaction text, we used the same prompt as above except with the reaction text components removed. Compared with the other tools, ChatGPT was provided with the reaction as a text string rather than a smiles string.

**BLASTp.** BLAST (Basic Alignment Search Tool) is an efficient algorithm for finding similar sequences in a reference database, to a query sequence [14]. We opted to use Diamond BLAST at the protein sequence level. For each test set, the training dataset was provided as the fasta file reference database, and the test set was provided as the query fasta.  $k = 1$  hits were returned for each query, sorted based on similarity, using default parameters. The EC number classification was then inferred from the returned reference sequence(s).

**Foldseek.** Structures for entries in the test sets were obtained from the AF2 database. Missing structures were folded with ESMFold, but a few rare cases ( 5) could not be successfully folded. Structures for the Price dataset were obtained with ESMFold. Structures for sequences in the training set were obtained from the AF2 database. While a small fraction of structures in the training set could not be retrieved from the AlphaFold database, we decided to proceed without these. Structure search was performed with Foldseek with default parameters, and only the top hit was used to assign function.

**CLEAN.** CLEAN is a supervised contrastive model, which trains a classification head using embeddings from the ESM protein language model [48] by aligning embeddings from the same EC category and contrasting embeddings from different EC categories [22]. For Task 1, we retrained CLEAN using our training set with only one example from each cluster at 50% identity, following the instructions provided in their codebase. We did not perform any clustering before training. Because our training set size was similar in size to that used in the original model, we used the recommended 7000 epochs of training. We used the default script with triplet margin loss. After, we performed inference but added our own code into the original CLEAN code to output the classification results into a format compatible with our downstream analysis.

**Pika.** Pika is a finetuned LLM, which accepts protein sequence as an auxiliary input and can perform reasoning and protein function prediction [67]. We retrained Pika using the training set and evaluated it with the "qa" functionality by creating a new annotation set. The annotation set is an entry for each enzyme with the question: "What is the EC number of this protein?" and the answer of the EC number for that protein. For training, we use 70% of the training data for training, and 15% for testing and validation respectively (the "test" datasets are completely held out of this process). The model was trained for a maximum of 1000 epochs, with a maximum batch size of 100. We used the default models from Pika: esm2\_t6\_8M\_UR50D as the protein model and gpt2 as the text model. All parameters were unchanged from the default apart from increasing the batch size and number of epochs, though hyperparameter tuning could improve the model. To evaluate the ability of Pika

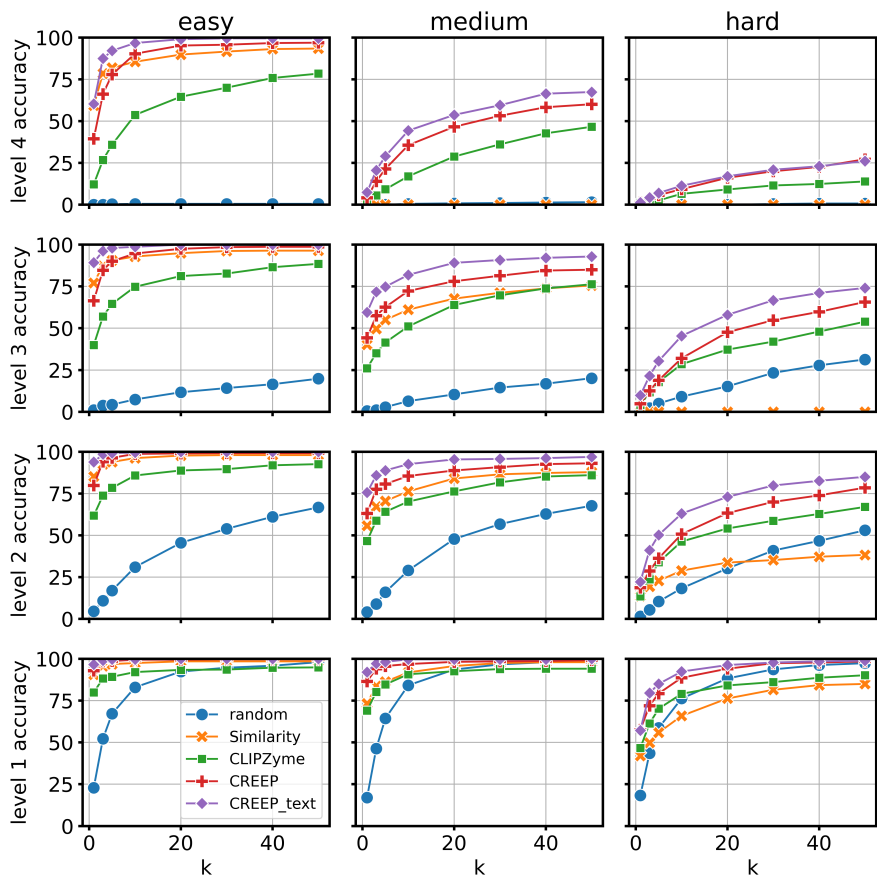


Figure A.6: Additional benchmarking results on Task 2, showing the increase in retrieval accuracy for  $k = 1, 3, 5, 10, 20, 30, 40$ .

to infer the EC numbers, we then query the trained model for each enzyme in the test set with the question: "What is the EC number of this protein?".

**CLIPZyme.** CLIPZyme is a multimodal contrastive model, between reactions represented as 2D graphs and protein structures represented as 3D graphs [79]. CLIPZyme was trained on direct reaction-protein pairs from the EnzymeMap dataset [37]. In this study, we retrained CLIPZyme model using default parameters and the EGNN protein structure representation for retrieval on the Task 2 train-test splits. We modified the CLIPZyme code to loop over EC numbers and randomly sample pairs of protein sequences and reaction pairs associated with that EC number, much like CREEP, for 40 epochs. In this case, we only provided one protein sequence from each cluster at 50% identity, and we only trained with mapped reactions that had conserved atom numbers on both sides of the reaction, thus reducing the total number of available EC numbers. Furthermore, we dropped several protein sequences with structures that did not work with the EGNN encoder. In the future, the hyperparameters and training procedure could likely be optimized to yield improved performance.

**ChemCrow.** ChemCrow is an LLM for chemistry problems, primarily synthesis planning [72]. It was unable to answer the results for Task 1, responding with the answers in Table A.1. For Task 2, we again tested a similar query to that input into ChatGPT, and received the responses in Table A.1. It should be noted, we use the public version so the private model may provide answers. Future work could involve building a biosynthesis-focused version of ChemCrow.

## A.5 Extended Benchmarking Results

Additional benchmarking results can be found at <https://github.com/jsunn-y/CARE/>.

Table A.1: Large language model responses to queries

Tool	Query and Response
ChemCrow	<p><i>Return the EC number associated with this reaction: C=CCNI@J2..</i> → Entering new RetryAgentExecutor chain... Thought: The question asks for the EC number associated with a given reaction. However, none of the provided tools can directly provide the EC number for a reaction. The EC number is typically associated with enzymes that catalyze reactions, and it's not something that can be derived from the reactants or products alone. Therefore, it's not possible to provide an answer using the available tools. Final Answer: I'm sorry, but with the tools available, I can't provide the EC number associated with the given reaction. The EC number is typically associated with enzymes that catalyze reactions, and it's not something that can be derived from the reactants or products alone. Thought: The question asks for the EC number associated with a given reaction. However, none of the provided tools can directly provide the EC number for a reaction. The EC number is typically associated with enzymes that catalyze reactions, and it's not something that can be derived from the reactants or products alone. Therefore, it's not possible to provide an answer using the available tools. Final Answer: I'm sorry, but with the tools available, I can't provide the EC number associated with the given reaction. The EC number is typically associated with enzymes that catalyze reactions, and it's not something that can be derived from the reactants or products alone.</p> <p>&gt; Finished chain. "I'm sorry, but with the tools available, I can't provide the EC number associated with the given reaction. The EC number is typically associated with enzymes that catalyze reactions, and it's not something that can be derived from the reactants or products alone.</p>
ChatGPT	<p><i>Examples of failed responses from ChatGPT: Example 1:</i> → Due to the limitations of this text-based platform, it's not possible to execute a program code here. In real conditions, I would use bioinformatics tools or a machine learning system to predict the Enzyme Commission (EC) numbers from the given protein sequence. However, in this case, I cannot complete this task. <b>Example 2:</b> → Due to the constraints of the current AI model, it's not able to computationally predict Enzyme Commission (EC) numbers from protein sequences directly. It's recommended to use specialized protein sequence analysis tools or machine learning algorithms designed for this purpose. For reliable and accurate results, databases like BRENDA, the Universal Protein Resource (UniProt), or tools like BLASTP, ECemblem, DeepEC could be utilized.</p>

## A.6 License

The codebase is open source under the MIT license.