FLOP: Tasks for Fitness Landscapes Of Protein wildtypes

Peter Mørch Groth[1,2], **Richard Michael**[1],
Jesper Salomon[2], **Pengfei Tian**[2], **Wouter Boomsma**[1]
[1]Department of Computer Science, University of Copenhagen
[2]Bioinformatics & Design, Enzyme Research, Novozymes
{petergroth,richard.michael,wb}@di.ku.dk
{pmg,jrsx,pfi}@novozymes.com

Abstract

Protein engineering has the potential to create optimized protein variants with improved properties and function. An initial step in the protein optimization process typically consists of a search among natural (wildtype) sequences to find the naturally occurring proteins with the most desirable properties. Promising candidates from this initial discovery phase then form the basis of the second step: a more local optimization procedure, exploring the space of variants separated from this candidate by a number of mutations. While considerable progress has been made on evaluating machine learning methods on single protein datasets, benchmarks of data-driven approaches for global fitness landscape exploration are still lacking. In this paper, we have carefully curated a representative benchmark dataset, which reflects industrially relevant scenarios for the initial wildtype discovery phase of protein engineering. We focus on exploration within a protein family, and investigate the downstream predictive power of various protein representation paradigms, i.e., protein language model-based representations, structure-based representations, and evolution-based representations. Our benchmark highlights the importance of coherent split strategies, and how we can be misled into overly optimistic estimates of the state of the field. The codebase and data can be accessed via `https://github.com/petergroth/FLOP`.

1 Introduction

The goal of protein engineering is to optimize proteins towards a particular trait of interest. This has applications both for industrial purposes and drug design. There is clear potential for machine learning to aid in this process. By predicting which protein sequences are most promising for experimental characterization, we can accelerate the exploration of the "fitness landscape" of the protein in question [1]. Regression of functional landscapes is challenging for multiple reasons. Typically a data-scarce problem, careful considerations of the experimental setup are required to avoid inadvertent data leakage. Concerning the functional landscapes of naturally occurring (also known as *wildtype*) proteins, the pairwise amino acid sequence identities can often vary significantly with some proteins differing by only a single amino acid while others might be less than ten percent similar. High-throughput experimental techniques are improving the data scarcity issue, while underlying structure typically exists in the datasets allowing for supervised learning despite the intrinsic challenges.

The optimization process of proteins and enzymes can typically be divided into multiple stages. An often employed initial step is to search for promising candidates among wildtype proteins, resulting in a set of proteins with desirable properties. We will refer to this as the *wildtype discovery* phase [2]. Since we are typically optimizing for a specific trait, we may often limit this initial exploration to a

Preprint. Under review.

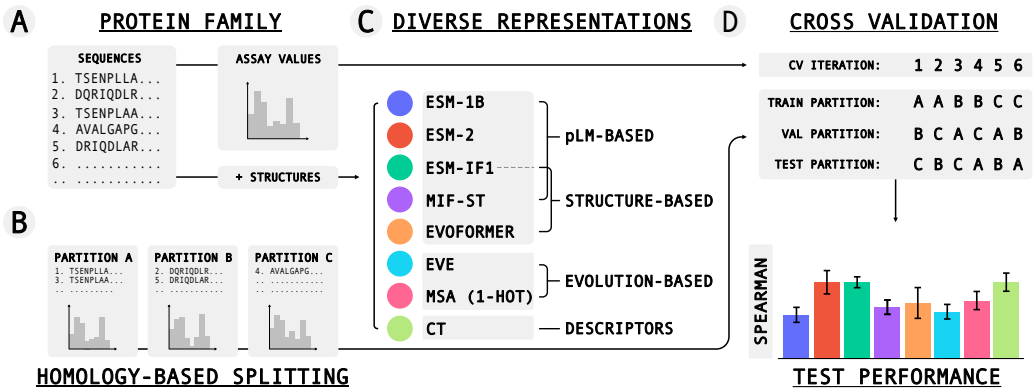

Figure 1: Schematic over dataset splitting, representations, and cross-validation process. A: A dataset with sequences from a single protein family and corresponding assay values is curated. B: A stratified sequence identity splitting procedure generates partitions A, B, and C, which are (1) homologically different from each other, (2) contain similar number of sequences, and (3) match the full dataset's target distribution. C: Eight types of protein representations are computed. D: Cross-validation using a random forest regressor is applied to obtain mean values and standard errors on the test partitions.

particular protein family, where the members share an evolutionary history which has resulted in a similar function. The selected set of wildtype proteins will then form the basis for a second phase in the engineering process: localized optimization, where novel variants of the wildtype proteins are examined through various assays [3]. Sometimes, the wildtype discovery phase is not only carried out once as several rounds might be required before an initial suitable candidate is found. Additionally, the resulting candidate might prove insufficient at a later stage of protein engineering, where conditions such as temperature are altered or where stress-factors are introduced.

In recent years, we have seen considerable efforts in defining benchmarks to help the machine learning community make progress in this field. However, these efforts have primarily focused on the second stage, i.e., variant effect prediction, where a dataset consist of thousands of variants from a single wildtype. In this paper, we argue for the importance of establishing well-defined benchmark tasks for the first stage as well. We present three challenging tasks and a careful analysis of the experimental design, demonstrating how poor choices can lead to dramatic overestimation of performance.

We conduct our experiments using a variety of fixed-size protein representations: sequence-based embeddings obtained through protein language models, structure-based representations from folding and inverse folding models, evolution-based representations obtained from multiple sequence alignments, as well as simple biologically-motivated sequence descriptors. In addition to the supervised approach, we include three unsupervised predictors to showcase a simpler approach to the task of identifying promising candidates. We show that the choice of representation can greatly affect the downstream predictive performance, and we therefore argue that more progress can be made by constructing meaningful representations and not solely in the construction of complex models. Given the oftentimes limited dataset sizes, we therefore rely on a random forest regressor.

2 Related work

Benchmarks play an important role in driving progress in protein-related prediction tasks. The most well-known is perhaps the rolling CASP benchmark, which is arguably responsible for the recent breakthroughs in protein structure prediction [4–6]. For the prediction of protein stability and function, several studies have curated relevant experimental datasets for use as benchmarks. The TAPE benchmark was an early such example designed to test protein sequence representations on a set of diverse downstream tasks [7]. Two of these tasks were related to protein engineering: stability prediction on variants of a set of 12 designed proteins [8] and characterization of the functional landscape of green fluorescent protein [9]. The PEER benchmark [10] expanded on the TAPE benchmark with many additional tasks. This included prediction of β-lactamase activity [11], and a binary solubility classification task on a diverse set of proteins. Focusing entirely on variant effects,

the recent ProteinGym benchmark has assembled a large set of Deep Mutational Scanning (DMS) assays and made them available as substitution and insertion-deletion prediction tasks [12]. While the above all consider protein sequence inputs, the recent Atom3D benchmark [13] presents various prediction tasks using 3D structure as input, including predicting amino acid identity from structural environments (for general proteins), and mutation effects on protein binding, using data originating from the SKEMPI database [14, 15].

Most closely related to this current paper is the FLIP benchmark, which dedicates itself to the prediction of functional fitness landscapes of proteins for protein engineering [16]. FLIP introduces three tasks: one on the prediction of protein stability of wildtype proteins (i.e., distributed over many families) using data from the Meltome Atlas [17], and two tasks focused on mutations at specific sites of proteins GB1 [18] and AAV [19]. While the FLIP benchmark is of great value for protein engineering, there are key characteristics which make it unsuitable for wildtype discovery, e.g., the use of the Meltome Atlas, which consists of thousands of sequences from different organisms spanning many different protein families. The sequences in the GB1 dataset only have mutations at four fixed positions while the sequences in the AAV dataset only contains 39 mutation sites, both of which corresponds to mutations at less than 10% of the full-length proteins. Such datasets with very local fitness landscapes are not generalizable enough for wildtype discovery.

Most of the functional tasks in current benchmarks are thus concerned with protein sequences that are derived from a single wildtype sequence by one or more mutations. Characterizing the functional effects of such variants is critical for protein engineering. However, before engaging in the optimization process itself, it is important to select meaningful starting points. As a natural complement to the FLIP benchmark, we therefore present a novel benchmark titled FLOP. The tasks we present are thus the characterization of functional landscapes of wildtype proteins within single families.

Our curated datasets all consist of functionally characterized wildtype sequences. For each dataset, we limit ourselves to a single family, and define our tasks as regression problems on the functional assay values. Given the low-data regime, the focus of our benchmark is to find representations of the protein input that makes few-shot or even zero-shot learning feasible. As a point of departure, we provide a set of state-of-the-art embeddings, reflecting different protein modalities.

3 Experimental setup

The domain we explore in this work is characterized by data scarcity, requiring special care in the design of the experimental setup. Figure 1 shows an overall schematic of the benchmarking process.

3.1 Dataset splitting

With the proliferation of large datasets and computationally demanding models, a common learning paradigm in machine learning is to rely on hold-out validation, whereby fixed training, validation, and testing sets are randomly generated. This method has several serious limitations when applied to biological datasets of limited sizes. Firstly, randomly splitting a dataset assumes that the data points are independent and identically distributed (i.i.d.). This is however not the case for members of a protein family which share common ancestors, leading to potential data leakage if protein sequences that are close in evolutionary space are placed in separate splits. Secondly, when splitting small datasets for a hold-out validation approach for supervised learning, the target values might not be well-balanced, resulting in dissimilar target distributions thus leading to bias and poor generalizability.

To handle these potential issues, we rely on a sequence identity-based, stratified cross-validation procedure ensuring that (1) partitions are generated such that any two proteins occurring in different partitions are guaranteed to be different at a pre-set homology cut-off, (2) cross-validation (CV) minimizes the potential bias which might occur during hold-out validation, (3) the target distribution is reflected by the generated partitions via stratification on discretized target values, and (4) the number of sequences in each partition is similar to reduce the variance.

To generate these high-quality data partitions, we use the four-phase procedure described in [20] to create three label-balanced dataset splits. We concretely use "GraphPart" [21]. We begin the procedure from an initial sequence identity threshold, and increase the threshold until the generated partitions are of sufficient sizes (i.e., at least 25 % of sequences in all three partitions). The stratification is

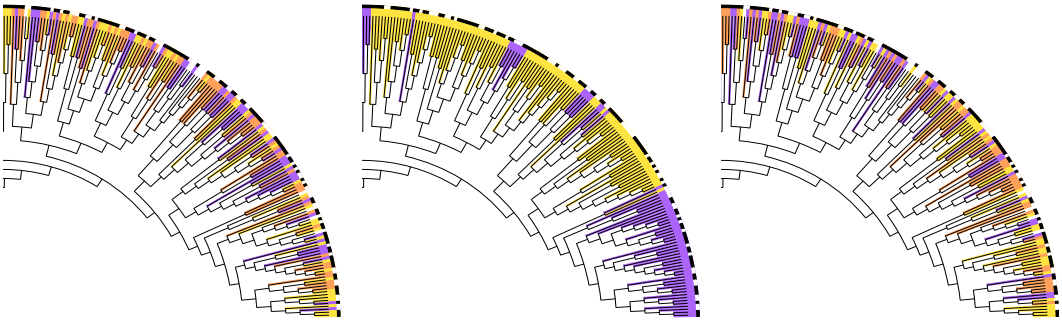

Figure 2: The same segment of a phylogenetic tree for the PPAT dataset. Branch color corresponds to its CV partition, while the outermost ring shows the target values (black indicates high and white indicates low values). The segments highlight the diversity found in wildtype protein families. Left: entries are colored according to the prescribed dataset splitting procedure which allows learning across subfamilies (indicated by the mix of colors). Middle: entries are colored by a clustering approach leading to wide regions, disallowing learning across subfamilies. Right: entries are randomly assigned a color. While similar to the leftmost scheme, the random coloring allows near identical sequences to be placed in separate partitions leading to excessive data-leakage.

achieved by creating a binary label which indicates whether a protein has low or high target value, e.g., by fitting a two-component Gaussian mixture model.

Figure 2 shows the same segment of a phylogenetic tree of the curated PPAT dataset, showing the evolutionary relationship between sequences. Large versions of the three trees can be found in Appendix E. The colors indicate which CV partition each sequence belongs to while the black and white squares in the outer ring indicate the stratification labels. The segments show the diversity encountered in wildtype protein families. The left segment is colored by our splitting procedure and shows that it manages to create diverse partitions spanning the entire evolutionary tree to allow learning across protein subfamilies. The middle segment is colored by an MMseqs [22] clustering approach, leading to wide areas disallowing learning across subfamilies. The entries in the rightmost segment are randomly assigned a color, corresponding to random splitting. While similar to the leftmost scheme, the random coloring allows near-identical sequences to be placed in separate partitions leading to excessive data-leakage.

3.2 Representations

To accurately reflect the current paradigms of state-of-the-art protein representations, we choose representatives from three main categories, the dimensionalities of which can be found in Appendix F.

Protein language models (pLMs) that are trained on hundreds of millions of protein sequences in an unsupervised fashion have been proven to be competitive representations for a multitude of tasks [23–25]. We here choose the popular ESM-1B [23] and the more recent ESM-2 models [26]. To fix the dimensionality for proteins of different lengths, we perform mean-pooling over the residue dimension.

The second category we include is structure-based. While folding proteins with AlphaFold2 [27], we extract embeddings from the Evoformer-modules, which have been shown to perform well for structure-related prediction tasks [28], using ColabFold [29]. Using the predicted structures, we then extract embeddings from the inverse-folding model ESM-IF1 [30] which also incorporates a pLM architecture. We similarly use embeddings from the MIF-ST model, which is an inverse folding model leveraging a pretrained pLM [31]. To fix the dimensionality of the variably-sized embeddings, we apply mean pooling across the amino acid dimension.

The third category is evolution-based. As a baseline, we will use a one-hot encoded multiple sequence alignment (MSA) over the proteins of interest [32–34]. Since the MSA is independent of labels, we enrich the unaligned sequence pools with additional members from the respective protein families using UniProt [35] and InterPro [36]. Given MSAs, models can be designed which leverage the evolutionary history of the protein family (e.g., EVE and related models [37–39, 12, 40]). For

Table 1: Summary of datasets and splits.

	N_{tot}	N_A	N_B	N_C	**Split %ID**	**Target**	**Median %ID**	**Avg. length**
GH114	53	20	18	17	0.55	Activity	0.46	268.8
CM	855	341	259	255	0.40	Activity	0.40	91.1
PPAT	615	182	234	199	0.55	Fitness	0.51	161.6

each curated dataset, we train EVE [37] on the corresponding protein family and extract the latent representations. Technical details on the training procedure can be found in Appendix G.

In addition to these groups of advanced representations, we include compositional and transitional (CT) physicochemical descriptors for each protein sequence as a simple baseline, which relate to overall polarizability, charge, hydrophobicity, polarity, secondary structure, solvent accessibility, and van der Waals volume of each sequence as predicted using the PyBioMed library [41].

3.3 Regression

The purpose of this benchmark is to provide a structured procedure to evaluate the predictive performance on downstream regression tasks given protein representations. We believe that larger prediction improvements can be achieved by focusing on developing novel protein representations rather than more complex regression models. Due to the low-N setting in which we operate, the training of large, complex models is practically inhibited, which is why we have chosen to rely on a random forest regressor. For each combination of the generated CV partitions, we perform a hyperparameter optimization on the current validation partition and evaluate the best-performing predictor on the current test partition. The experiments were also carried out using alternate regressors, the results of which can be found in Appendix L.1 with the hyperparameter grids in Appendix I.

3.4 Zero-shot predictors

To investigate the efficacy of unsupervised learning on the curated datasets, we evaluate three zero-shot predictors. Using EVE, we evaluate the evidence lower bound (ELBO) by sampling and obtain a proxy for sequence fitness, analogous to the evolutionary index in [37]. A second proxy is obtained by evaluating the log-likelihood of a sequence conditioned on its structure using the ESM-IF1 inverse folding model [30]. The third zero-shot estimator is obtained by using Tranception [42] to evaluate the log-likelihood of each sequence. Details can be found in Appendix H.

4 Datasets

The three curated datasets and the corresponding fitness landscapes are here motivated and described.

4.1 GH114

Motivation. Accurately identifying enzymes with the highest activities towards a specific substrate is of central importance during enzyme engineering. This requires that assay observations are directly comparable [43], which includes ensuring identical experimental assay conditions, such that enzymes are evaluated at identical concentrations and purity levels. However, purifying enzymes requires significant work, resulting in a limited number of tested sequences, but of higher experimental quality.

Landscape. This dataset includes purified and concentration normalized natural glycoside gydrolase 114 (GH114) alpha-1,4-polygalactosaminidase enzymes and corresponding catalytic activity values [44] which will act as the target of interest. GH114 enzymes degrade the exopolysaccharide PEL, which provides structure and protection in some biofilms [45]. Having measurements of purified enzymes avoids issues with background effects from other enzymes in the recombinant host background. We provide a curated version of the GH114 dataset which, to our knowledge, has not been used in previous work for function prediction purposes.

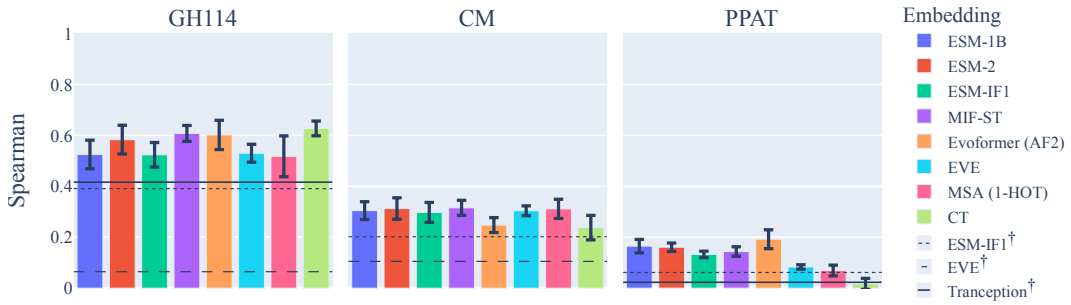

Figure 3: Benchmark results. Average Spearman's rank correlation (and standard error) between predictions and targets over test partitions. Higher is better. †: Zero-shot correlations.

4.2 CM

Motivation. Identification of enzymes with high catalytic activities is essential for enzyme engineering campaigns. However, predicting the activity level of enzymes using physics-based methods remains a great challenge [46]. Recent progress in high throughput screening allows the measurement of enzyme activity of sequences with high diversity, but with low experimental cost.

Landscape. This dataset contains the catalytic activity of natural chorismate mutase (CM) homologous proteins, as well as artificial sequences which follow the same pattern of variations (e.g., conservation and co-evolution) [47]. The artificial sequences generated by Monte Carlo simulations at low temperatures exhibit catalytic levels comparable to the natural homologs and have therefore been included. We perform an additional filtering of the dataset prior to the splitting procedure by removing sequences with target values less than 0.42, corresponding to inactive proteins [47]. This task thereby assumes that a preceding classification procedure has been carried out. For completeness, we include classification results before the filtering step in Appendix K, supporting this assumption.

4.3 PPAT

Motivation. PPAT (phosphopantetheine adenylyltransferase) is composed of around 160 amino acids and is an essential enzyme that catalyzes the second-to-last step in the CoA biosynthetic pathway. The target value for this prediction task is the fitness score, which reflects the ability of PPAT homologs to complement a knockout E. coli strain. The fitness of homologs can be affected by factors such as protein misfolding, mismatched metabolic flux, or environmental mismatches etc.

Landscape. This dataset contains fitness scores of 615 different PPAT homologs obtained by a novel DNA synthesis/assembly technology, DropSynth, followed by a multiplexed functional assay to measure how well each PPAT homolog can rescue a knockout phenotype [48].

4.4 Summary

A summary of the curated datasets including their total sizes, partition sizes, between-partition sequence identity threshold (Split %ID), target value, median pair-wise sequence identity, and average sequence length can be seen in Table 1. For comparison, in existing mutational fitness landscape datasets (i.e., for the local optimization phase), the median sequence identity is often close to 1.0 as all sequences are at pairwise Hamming distances of 1 (in the case of single mutants). The low median sequence identity observed here highlights the diversity – and difficulty – of wildtype datasets. Additional curation details are found in Appendix A. Histograms of the target values for the three datasets as well as partition histograms can be seen in Appendix C and D, respectively.

5 Results

Spearman's rank correlation between the predictions and targets over the three datasets can be seen in Figure 3 and Table 2. In protein engineering, the highest performing proteins are of interest making

Table 2: Benchmark results. Spearman's rank correlation and standard error using cross-validation. †: Zero-shot correlation on full datasets. Highest value and values within 1 SE are bold.

	GH114	CM	PPAT
ESM-1B	0.52 ± 0.06	$\mathbf{0.30} \pm 0.03$	$\mathbf{0.16} \pm 0.03$
ESM-2	$\mathbf{0.58} \pm 0.06$	$\mathbf{0.31} \pm 0.04$	$\mathbf{0.16} \pm 0.02$
ESM-IF1	0.52 ± 0.05	$\mathbf{0.30} \pm 0.04$	0.13 ± 0.01
MIF-ST	$\mathbf{0.61} \pm 0.03$	$\mathbf{0.32} \pm 0.03$	0.14 ± 0.02
Evoformer (AF2)	$\mathbf{0.60} \pm 0.06$	0.25 ± 0.03	$\mathbf{0.19} \pm 0.04$
EVE	0.53 ± 0.04	$\mathbf{0.30} \pm 0.02$	0.08 ± 0.01
MSA (1-HOT)	0.52 ± 0.08	$\mathbf{0.31} \pm 0.04$	0.07 ± 0.02
CT	$\mathbf{0.63} \pm 0.03$	0.24 ± 0.05	0.02 ± 0.02
ESM-IF1†	0.39	0.20	0.06
EVE†	0.06	0.11	-0.01
Tranception†	0.42	-0.05	0.02

it the ranking and not the absolute predictions that indicate the performance, despite the regressors being optimized using the mean squared error. The RMSE can found in Table A2 in the appendix.

GH114 The performance on the GH114 dataset highlights some of the peculiarities encountered with small wildtype protein datasets. The collection of physicochemical descriptors (CT), which is simpler than its competitors, achieves the highest score. Slightly below it are the structure-informed MIF-ST and Evoformer representations, indicating a structural signal which is however not picked up by the ESM-IF1 embeddings. While the CT representation achieves the highest mean value, several others are within one standard error, giving no clear advantage to neither complex nor simple models. All supervised approaches beat the zero-shot predictors, where both Tranception and ESM-IF1 likelihoods correlate fairly well with the targets, however.

CM The second prediction task can be considered more challenging given the results, despite the comparatively large size of the CM dataset. While similar to the first task, an abundance of data is not sufficient to increase the downstream capabilities if it comes at the cost of potentially noisier measurements, as compared to the concentration normalized GH114 dataset. Most representations fall within one standard error of the top performer such that, once again, no representation paradigm has a clear advantage.

PPAT The most challenging task of the datasets, the results on the PPAT task show different behaviour. The evolutionary signal, i.e., the amount of information which can be learned from evolutionary homologs, is weak as indicated by the low correlations from the one-hot encoded MSA and from EVE (both in the supervised and unsupervised settings as per Table 2). Furthermore, the physicochemical descriptors fail to correlate – as does the zero-shot predictors. The pLM and structure-based representations achieve the highest scores with the Evoformer embeddings coming out slightly ahead.

6 Ablation study

The dataset splitting procedure introduced and benchmark tasks have been carefully constructed to ensure reliable estimates of model performance. In this section we show three ablation studies – one for each dataset – whereby different choices of task-structuring might lead to great over-estimations of performance. The results can be seen in Figure 4 and Table 3. The Δ columns in the table indicate differences from the benchmark results, where a positive/green value indicates *better* performance during ablation, i.e., over-estimation.

Hold-out validation For GH114, we perform hold-out validation by arbitrarily designating the three generated partitions as training, validation, and test sets. The correlations are significantly different, with the ESM-2 representation decreasing by 50 % . With no systematic pattern and decreased nuance given the lack of errorbars, it is easy to draw incorrect conclusions. Choosing which

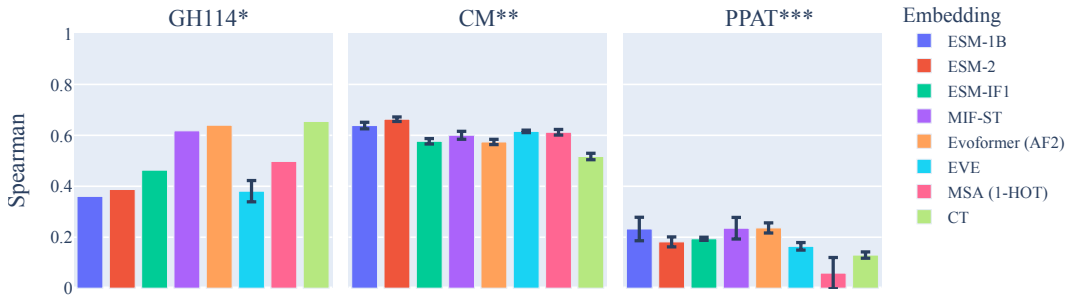

Figure 4: Ablation results. Spearman's correlation coefficient between predictions and targets over test partitions, grouped by dataset. Standard error is shown as vertical bars. *: Hold-out validation. **: Regression on both active and inactive proteins. ***: Repeated random splitting.

partitions to use for training, validation, and testing is (in this case) arbitrary and can evidently lead to biased results. An analogue ablation study for the CM and PPAT data can be found in Appendix J.1, where similar conclusions can be drawn.

Disregarding distinct target modalities For the CM dataset, we only included the active sequences in the benchmark. To demonstrate why, we have included the results of performing regression on both inactive and active sequences in the center of Table 3. These results are greatly overinflated, with some representations more than doubling the correlation values. Regression performed on a distinctly bimodal dataset (such as CM, see Figure C in the appendix) can inflate the results as the model is able to distinguish between the modalities, driving the ranking correlation to overly-optimistic values. The caveat here is knowing the labels a priori, which assumes a preceding classification-screening has been performed. The classification results of such a process can be seen in Appendix K.

Random partitioning for cross-validation For the PPAT dataset, we see the predictive performance when using repeated random splitting instead of stratified, homology-based splitting. While the results look similar to the benchmark results, we do see an increase in performance across the board. By randomly sampling, we risk placing very similar sequences in separate partitions, thereby allowing data-leakage, where we are essentially testing on training/validation data, thus overestimating the performance [49]. The results for this ablation study carried out on the GH114 and CM datasets can be found in Appendix J.2, where we can once again draw similar conclusions.

7 Discussion

We observe that the choice of representation can greatly affect the downstream predictive capabilities, with no consistent, clear edge given by any of the three protein representation paradigms. For CM, a simple one-hot encoded MSA acts as an impressive baseline proving difficult to convincingly beat. For GH114, physicochemical descriptors are sufficient to achieve top performance, while the PPAT dataset benefits from the complex, structure-informed Evoformer embeddings. While the specific top-scoring representation fluctuates, the ESM-2 embeddings are consistently within one standard error and can thus be considered a relatively consistent baseline for future experiments, where others occasionally underperform. For the three tasks, we see supervised learning outperforming unsupervised predictions, while the inverse-folding likelihood estimator however offers a decent zero-shot approach.

Despite similar overall patterns, some results stand out, e.g., the comparatively high performance on the GH114 dataset and the low performance on the PPAT dataset. Variations in experimental conditions and techniques can introduce different levels of noise. The CM and PPAT datasets are derived from tests on supernatants with complex backgrounds with potential side-activities from impurities, whereas the GH114 dataset uses purified samples with less expected noise. This can be a potential reason for the comparatively high performance of the latter. As for the low performance on the PPAT dataset, the reason might lie in the target values: the GH114 and CM datasets both measure enzymatic activities while the PPAT dataset measures fitness. The overall performance disparities

Table 3: Ablation results. Spearman correlation. *: Hold-out validation, **: Regression on both active and inactive proteins, ***: Repeated random splitting. Δ shows difference to benchmark results. Highest value and values within 1 SE are bold.

	GH114*	Δ	**CM****	Δ	**PPAT*****	Δ
ESM-1B	0.36	-0.16	0.64 ± 0.01	$+0.33$	$\mathbf{0.23} \pm 0.05$	$+0.07$
ESM-2	0.39	-0.20	$\mathbf{0.66} \pm 0.01$	$+0.35$	0.18 ± 0.02	$+0.02$
ESM-IF1	0.46	-0.06	0.58 ± 0.01	$+0.28$	0.19 ± 0.01	$+0.06$
MIF-ST	0.62	$+0.01$	0.60 ± 0.02	$+0.28$	$\mathbf{0.24} \pm 0.04$	$+0.09$
Evoformer (AF2)	0.64	$+0.04$	0.57 ± 0.01	$+0.33$	$\mathbf{0.24} \pm 0.02$	$+0.04$
EVE	0.38 ± 0.04	-0.15	0.62 ± 0.00	$+0.31$	0.16 ± 0.01	$+0.08$
MSA (1-HOT)	0.50	-0.02	0.61 ± 0.01	$+0.30$	0.06 ± 0.06	-0.01
CT	**0.65**	$+0.03$	0.52 ± 0.01	$+0.28$	0.13 ± 0.01	$+0.11$

suggest that enzyme activities, rather than a more complex and assay-specific fitness value, are easier to model given the available protein representation paradigms. The stark contrast in performance between the concentration normalized GH114 dataset and both the CM and PPAT datasets indicates that higher quality datasets are of central importance to learn accurate fitness landscapes – more so than the number of labelled sequences.

8 Conclusion

In this work we have presented a novel benchmark which investigates an unexplored domain of machine learning-driven protein engineering: the navigation of global fitness landscapes for single protein families. The curation of relevant labelled datasets proved challenging and the creation of new comprehensive family-wide datasets is bound to ease and improve future model development and applicability. We have highlighted the importance of careful learning schemes in the evaluation of our models to ensure reliable estimates of generalizability thus avoiding inadvertently overestimating the results. While both the sequence-based and structure-informed representations consistently yielded good results, the difference in performance was not significant. Often, a family-wide, one-hot encoded MSA proved competitive, highlighting the difficulty of creating efficient representations. We encourage the design of new representations which can avoid such pooling operations as were applied to the pretrained model embeddings, which are bound to filter out potentially important information.

The observed performance differences highlight the importance of the wildtype discovery phase of protein engineering, whereby more effective protein representations will allow for improved starting points for local optimization. In silico screenings of potential wildtype candidates can be scaled efficiently compared to expensive, time-consuming in vitro assays, significantly reducing the early costs of future protein engineering campaigns. Despite the sometimes low test-time performance observed in our study, even low correlations can provide useful guidance on selecting wildtype protein starting points.

We hope that FLOP will pave the way for modeling global fitness landscapes of protein families and accelerate the further development of novel protein representations which will drive protein engineering for drug discovery and sustainable biotechnology forward.

Acknowledgments and Disclosure of Funding

This work was funded in part by Innovation Fund Denmark (1044-00158A), the Novo Nordisk Foundation through the MLSS Center (Basic Machine Learning Research in Life Science, NNF20OC0062606), the Pioneer Centre for AI (DRNF grant number P1), and the Danish Data Science Academy, which is funded by the Novo Nordisk Foundation (NNF21SA0069429) and VILLUM FONDEN (40516).

References

[1] B. J. Wittmann, Y. Yue, and F. H. Arnold, "Informed training set design enables efficient machine learning-assisted directed protein evolution," *Cell systems*, vol. 12, no. 11, pp. 1026–1045, 2021.

[2] T. Davids, M. Schmidt, D. Böttcher, and U. T. Bornscheuer, "Strategies for the discovery and engineering of enzymes for biocatalysis," *Current Opinion in Chemical Biology*, vol. 17, no. 2, pp. 215–220, 2013.

[3] K. K. Yang, Z. Wu, and F. H. Arnold, "Machine-learning-guided directed evolution for protein engineering," *Nature methods*, vol. 16, no. 8, pp. 687–694, 2019.

[4] P. E. Bourne, "Casp and cafasp experiments and their findings," *Methods of Biochemical Analysis*, vol. 44, pp. 501–508, 2003.

[5] J. Moult, "A decade of casp: progress, bottlenecks and prognosis in protein structure prediction," *Current opinion in structural biology*, vol. 15, no. 3, pp. 285–289, 2005.

[6] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, "Critical assessment of methods of protein structure prediction (casp)—round xiv," *Proteins: Structure, Function, and Bioinformatics*, vol. 89, no. 12, pp. 1607–1617, 2021.

[7] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, X. Chen, J. Canny, P. Abbeel, and Y. S. Song, "Evaluating Protein Transfer Learning with TAPE," *arXiv:1906.08230 [cs, q-bio, stat]*, Jun. 2019, arXiv: 1906.08230. [Online]. Available: http://arxiv.org/abs/1906.08230

[8] G. J. Rocklin, T. M. Chidyausiku, I. Goreshnik, A. Ford, S. Houliston, A. Lemak, L. Carter, R. Ravichandran, V. K. Mulligan, A. Chevalier, C. H. Arrowsmith, and D. Baker, "Global analysis of protein folding using massively parallel design, synthesis, and testing," *Science (New York, N.Y.)*, vol. 357, no. 6347, pp. 168–175, Jul. 2017.

[9] K. S. Sarkisyan, D. A. Bolotin, M. V. Meer, D. R. Usmanova, A. S. Mishin, G. V. Sharonov, D. N. Ivankov, N. G. Bozhanova, M. S. Baranov, O. Soylemez *et al.*, "Local fitness landscape of the green fluorescent protein," *Nature*, vol. 533, no. 7603, pp. 397–401, 2016.

[10] M. Xu, Z. Zhang, J. Lu, Z. Zhu, Y. Zhang, C. Ma, R. Liu, and J. Tang, "PEER: A Comprehensive and Multi-Task Benchmark for Protein Sequence Understanding," Jun. 2022, number: arXiv:2206.02096 arXiv:2206.02096 [cs]. [Online]. Available: http://arxiv.org/abs/2206.02096

[11] V. E. Gray, R. J. Hause, J. Luebeck, J. Shendure, and D. M. Fowler, "Quantitative Missense Variant Effect Prediction Using Large-Scale Mutagenesis Data," *Cell Systems*, vol. 6, no. 1, pp. 116–124.e3, Jan. 2018.

[12] P. Notin, M. Dias, J. Frazer, J. Marchena-Hurtado, A. Gomez, D. S. Marks, and Y. Gal, "Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval," May 2022, number: arXiv:2205.13760 arXiv:2205.13760 [cs]. [Online]. Available: http://arxiv.org/abs/2205.13760

[13] R. J. L. Townshend, M. Vögele, P. Suriana, A. Derry, A. Powers, Y. Laloudakis, S. Balachandar, B. Jing, B. Anderson, S. Eismann, R. Kondor, R. B. Altman, and R. O. Dror, "ATOM3D: Tasks On Molecules in Three Dimensions," *arXiv:2012.04035 [physics, q-bio]*, Jan. 2022, arXiv: 2012.04035. [Online]. Available: http://arxiv.org/abs/2012.04035

[14] I. H. Moal and J. Fernández-Recio, "SKEMPI: a Structural Kinetic and Energetic database of Mutant Protein Interactions and its use in empirical models," *Bioinformatics*, vol. 28, no. 20, pp. 2600–2607, Oct. 2012. [Online]. Available: https://doi.org/10.1093/bioinformatics/bts489

[15] J. Jankauskaite, B. Jiménez-García, J. Dapkunas, J. Fernández-Recio, and I. H. Moal, "SKEMPI 2.0: an updated benchmark of changes in protein-protein binding energy, kinetics and thermodynamics upon mutation," *Bioinformatics (Oxford, England)*, vol. 35, no. 3, pp. 462–469, Feb. 2019.

[16] C. Dallago, J. Mou, K. E. Johnston, B. J. Wittmann, N. Bhattacharya, S. Goldman, A. Madani, and K. K. Yang, "FLIP: Benchmark tasks in fitness landscape inference for proteins," Jan. 2022, pages: 2021.11.09.467890 Section: New Results. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.11.09.467890v2

[17] A. Jarzab, N. Kurzawa, T. Hopf, M. Moerch, J. Zecha, N. Leijten, Y. Bian, E. Musiol, M. Maschberger, G. Stoehr, I. Becher, C. Daly, P. Samaras, J. Mergner, B. Spanier, A. Angelov, T. Werner, M. Bantscheff, M. Wilhelm, M. Klingenspor, S. Lemeer, W. Liebl, H. Hahne, M. M. Savitski, and B. Kuster, "Meltome atlas—thermal proteome stability across the tree of life," *Nature Methods*, vol. 17, no. 5, pp. 495–503, May 2020, number: 5 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41592-020-0801-4

[18] N. C. Wu, L. Dai, C. A. Olson, J. O. Lloyd-Smith, and R. Sun, "Adaptation in protein fitness landscapes is facilitated by indirect paths," *Elife*, vol. 5, p. e16965, 2016.

[19] D. H. Bryant, A. Bashir, S. Sinai, N. K. Jain, P. J. Ogden, P. F. Riley, G. M. Church, L. J. Colwell, and E. D. Kelsic, "Deep diversification of an aav capsid protein by machine learning," *Nature Biotechnology*, vol. 39, no. 6, pp. 691–696, 2021.

[20] M. H. Gíslason, H. Nielsen, J. J. Almagro Armenteros, and A. R. Johansen, "Prediction of gpi-anchored proteins with pointer neural networks," *Current Research in Biotechnology*, vol. 3, pp. 6–13, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2590262821000010

[21] M. H. Gíslason, F. Teufel, J. J. A. Armenteros, O. Winther, and H. Nielsen, "Protein dataset partitioning pipeline," 2021. [Online]. Available: https://github.com/graph-part/graph-part

[22] M. Steinegger and J. Söding, "MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature Biotechnology*, vol. 35, no. 11, pp. 1026–1028, Nov. 2017, number: 11 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/nbt.3988

[23] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," bioRxiv, Tech. Rep., Dec. 2020, section: New Results Type: article. [Online]. Available: https://www.biorxiv.org/content/10.1101/622803v4

[24] J. Meier, R. Rao, R. Verkuil, J. Liu, T. Sercu, and A. Rives, "Language models enable zero-shot prediction of the effects of mutations on protein function," bioRxiv, Tech. Rep., Jul. 2021, section: New Results Type: article. [Online]. Available: https://www.biorxiv.org/content/10.1101/2021.07.09.450648v1

[25] R. M. Rao, J. Liu, R. Verkuil, J. Meier, J. Canny, P. Abbeel, T. Sercu, and A. Rives, "MSA Transformer," in *Proceedings of the 38th International Conference on Machine Learning*. PMLR, Jul. 2021, pp. 8844–8856, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v139/rao21a.html

[26] Z. Lin, H. Akin, R. Rao, B. Hie, Z. Zhu, W. Lu, A. d. Santos Costa, M. Fazel-Zarandi, T. Sercu, S. Candido, and A. Rives, "Language models of protein sequences at the scale of evolution enable accurate structure prediction," Synthetic Biology, preprint, Jul. 2022. [Online]. Available: http://biorxiv.org/lookup/doi/10.1101/2022.07.20.500902

[27] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska, T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, and D. Hassabis, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, Aug. 2021, number: 7873 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-021-03819-2

[28] M. Hu, F. Yuan, K. K. Yang, F. Ju, J. Su, H. Wang, F. Yang, and Q. Ding, "Exploring evolution-based & -free protein language models as protein function predictors," Jun. 2022, number: arXiv:2206.06583 arXiv:2206.06583 [cs, q-bio]. [Online]. Available: http://arxiv.org/abs/2206.06583

[29] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger, "Colabfold: making protein folding accessible to all," *Nature Methods*, vol. 19, no. 6, pp. 679–682, Jun 2022. [Online]. Available: https://doi.org/10.1038/s41592-022-01488-1

[30] C. Hsu, R. Verkuil, J. Liu, Z. Lin, B. Hie, T. Sercu, A. Lerer, and A. Rives, "Learning inverse folding from millions of predicted structures," in *Proceedings of the 39th International Conference on Machine Learning*. PMLR, Jun. 2022, pp. 8946–8970, iSSN: 2640-3498. [Online]. Available: https://proceedings.mlr.press/v162/hsu22a.html

[31] K. K. Yang, H. Yeh, and N. Zanichelli, "Masked inverse folding with sequence transfer for protein representation learning," Mar. 2023, pages: 2022.05.25.493516 Section: New Results. [Online]. Available: https://www.biorxiv.org/content/10.1101/2022.05.25.493516v3

[32] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt, "Direct-coupling analysis of residue coevolution captures native contacts across many protein families," *Proceedings of the National Academy of Sciences*, vol. 108, no. 49, pp. E1293–E1301, 2011.

[33] D. S. Marks, L. J. Colwell, R. Sheridan, T. A. Hopf, A. Pagnani, R. Zecchina, and C. Sander, "Protein 3d structure computed from evolutionary sequence variation," *PloS one*, vol. 6, no. 12, p. e28766, 2011.

[34] P. Tian and R. B. Best, "How many protein sequences fold to a given structure? a coevolutionary analysis," *Biophysical journal*, vol. 113, no. 8, pp. 1719–1730, 2017.

[35] The UniProt Consortium, "UniProt: the Universal Protein Knowledgebase in 2023," *Nucleic Acids Research*, vol. 51, no. D1, pp. D523–D531, Jan. 2023. [Online]. Available: https://doi.org/10.1093/nar/gkac1052

[36] M. Blum, H.-Y. Chang, S. Chuguransky, T. Grego, S. Kandasaamy, A. Mitchell, G. Nuka, T. Paysan-Lafosse, M. Qureshi, S. Raj, L. Richardson, G. A. Salazar, L. Williams, P. Bork, A. Bridge, J. Gough, D. H. Haft, I. Letunic, A. Marchler-Bauer, H. Mi, D. A. Natale, M. Necci, C. A. Orengo, A. P. Pandurangan, C. Rivoire, C. J. A. Sigrist, I. Sillitoe, N. Thanki, P. D. Thomas, S. C. E. Tosatto, C. H. Wu, A. Bateman, and R. D. Finn, "The interpro protein families and domains database: 20 years on," *Nucleic acids research*, vol. 49, no. D1, p. D344—D354, January 2021. [Online]. Available: https://europepmc.org/articles/PMC7778928

[37] J. Frazer, P. Notin, M. Dias, A. Gomez, J. K. Min, K. Brock, Y. Gal, and D. S. Marks, "Disease variant prediction with deep generative models of evolutionary data," *Nature*, vol. 599, no. 7883, pp. 91–95, Nov. 2021, number: 7883 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41586-021-04043-8

[38] A. J. Riesselman, J. B. Ingraham, and D. S. Marks, "Deep generative models of genetic variation capture the effects of mutations," *Nature Methods*, vol. 15, no. 10, pp. 816–822, Oct. 2018, number: 10 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41592-018-0138-4

[39] T. A. Hopf, A. G. Green, B. Schubert, S. Mersmann, C. P. I. Schärfe, J. B. Ingraham, A. Toth-Petroczy, K. Brock, A. J. Riesselman, P. Palmedo, C. Kang, R. Sheridan, E. J. Draizen, C. Dallago, C. Sander, and D. S. Marks, "The EVcouplings Python framework for coevolutionary sequence analysis," *Bioinformatics (Oxford, England)*, vol. 35, no. 9, pp. 1582–1584, May 2019.

[40] D. Hesslow, N. Zanichelli, P. Notin, I. Poli, and D. Marks, "RITA: a Study on Scaling Up Generative Protein Sequence Models," arXiv, Tech. Rep. arXiv:2205.05789, May 2022, arXiv:2205.05789 [cs, q-bio] type: article. [Online]. Available: http://arxiv.org/abs/2205.05789

[41] J. Dong, Z.-J. Yao, L. Zhang, F. Luo, Q. Lin, A.-P. Lu, A. F. Chen, and D.-S. Cao, "PyBioMed: a python library for various molecular representations of chemicals, proteins and DNAs and their interactions," *Journal of Cheminformatics*, vol. 10, no. 1, p. 16, Mar. 2018.

[42] P. Notin, M. Dias, J. Frazer, J. M. Hurtado, A. N. Gomez, D. Marks, and Y. Gal, "Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval," in *International Conference on Machine Learning*. PMLR, 2022, pp. 16 990–17 017.

[43] R. Lonsdale, J. N. Harvey, and A. J. Mulholland, "A practical guide to modelling enzyme-catalysed reactions," *Chemical Society Reviews*, vol. 41, no. 8, pp. 3025–3038, Apr. 2012. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3371381/

[44] M. Li, J. Salomon, D. R. Segura, M. A. Stringer, R. M. Vejborg, D. M. K. Klitgaard, D. Nissen, W. Peng, and T. Sun, "Polypeptides," Patent WO/2019/228 448, Dec., 2019. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2019228448

[45] K. M. Colvin, V. D. Gordon, K. Murakami, B. R. Borlee, D. J. Wozniak, G. C. L. Wong, and M. R. Parsek, "The Pel Polysaccharide Can Serve a Structural and Protective Role in the Biofilm Matrix of Pseudomonas aeruginosa," *PLOS Pathogens*, vol. 7, no. 1, p. e1001264, Jan. 2011, publisher: Public Library of Science. [Online]. Available: https://journals.plos.org/plospathogens/article?id=10.1371/journal.ppat.1001264

[46] M. K. Tiwari, R. Singh, R. K. Singh, I.-W. Kim, and J.-K. Lee, "Computational approaches for rational design of proteins with novel functionalities," *Computational and structural biotechnology journal*, vol. 2, no. 3, p. e201204002, 2012.

[47] W. P. Russ, M. Figliuzzi, C. Stocker, P. Barrat-Charlaix, M. Socolich, P. Kast, D. Hilvert, R. Monasson, S. Cocco, M. Weigt, and R. Ranganathan, "An evolution-based model for designing chorismate mutase enzymes," *Science*, vol. 369, no. 6502, pp. 440–445, Jul. 2020, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/full/10.1126/science.aba3304

[48] C. Plesa, A. M. Sidore, N. B. Lubock, D. Zhang, and S. Kosuri, "Multiplexed gene synthesis in emulsions for exploring protein functional landscapes," *Science*, vol. 359, no. 6373, pp. 343–347, Jan. 2018, publisher: American Association for the Advancement of Science. [Online]. Available: https://www.science.org/doi/10.1126/science.aao5167

[49] P. Bork and E. V. Koonin, "Predicting functions from protein sequences–where are the bottlenecks?" *Nature Genetics*, vol. 18, no. 4, pp. 313–318, Apr. 1998.

[50] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, and M. Steinegger, "ColabFold: making protein folding accessible to all," *Nature Methods*, vol. 19, no. 6, pp. 679–682, Jun. 2022, number: 6 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/s41592-022-01488-1

[51] S. Deorowicz, A. Debudaj-Grabysz, and A. Gudyś, "FAMSA: Fast and accurate multiple sequence alignment of huge protein families," *Scientific Reports*, vol. 6, no. 1, p. 33964, Sep. 2016, number: 1 Publisher: Nature Publishing Group. [Online]. Available: https://www.nature.com/articles/srep33964

[52] M. N. Price, P. S. Dehal, and A. P. Arkin, "FastTree: Computing Large Minimum Evolution Trees with Profiles instead of a Distance Matrix," *Molecular Biology and Evolution*, vol. 26, no. 7, pp. 1641–1650, Jul. 2009. [Online]. Available: https://doi.org/10.1093/molbev/msp077

[53] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

Appendix

A Dataset details

The curated datasets are kept in csv-files with the following columns:

- index: Index for each protein.
- name: Unique name for each protein. This identifier maps directly to the file name for all representations. For example, the ESM-2 embedding for sequence <seq_id> from <dataset> can be found in representations/<dataset>/esm_2/<seq_id>.pt.
- sequence: Amino acid sequence.
- target_reg: The assay value/regression target.
- target_class: Binarized assay value for stratification.[1]
- part_0: 1 if sequence belongs to the first partition, 0 otherwise.
- part_1: 1 if sequence belongs to the second partition, 0 otherwise.
- part_2: 1 if sequence belongs to the third partition, 0 otherwise.

The curated file for <dataset> is placed in data/processed/<dataset>/<dataset>.csv. For details on data access, see Section A.1.

All structures were predicted with AlphaFold2 [27] using ColabFold [50] using five recycling runs with model version alphafold2_multimer_v3 with early stopping at pLDDT of 90.0. The predicted structures can be found in the data/raw/<dataset>/pdb directory for each dataset.

We ask that references to the tasks in this paper include references to the original dataset authors.

A.1 Dataset/code access

All code is accessible via the repository at https://github.com/petergroth/FLOP. The three curated dataset files can be found in the repository as three separate csv-files. All remaining files (including PDB-files, pre-computed representations, raw data files, etc.) can be found at https://sid.erda.dk/sharelink/HLXs3e9yCu. Additional details can be found in the repository.

A.2 GH114

The GH114 dataset was extracted from the WO2019228448 patent [44] which can be accessed at https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2019228448, or alternatively at https://patents.google.com/patent/WO2019228448A1/en. The assay values/protein pairs can be found in Table 1 in the main text (columns SEQ ID and Absorbance at 405 nm - blank) while the corresponding sequences can be found in the Sequence Listing document. Each protein sequence is encapsulated by <210>, where the number following <210> corresponds to a SEQ ID entry from patent Table 1. E.g., the sequence for protein SEQ ID 12 is found between <210> 12 and the next <210>. Each amino acid is described using 3-letter symbols (e.g., Ala for alanine). These have been processed into 1-letter symbols, and subsequently into the full sequence strings, which are collected in data/raw/gh114/gh114.fasta.

To strengthen the MSA, additional members from the GH114 family (PF03537) were added using the UniProt and InterPro databases [35, 36], where the sequence lengths of the added members were limited to 550 to limit the size of the final alignment, resulting in a sequence pool of 6507 sequences. The sequences are aligned using FAMSA [51].

While the data is publicly available, explicit permission to use the data for benchmarking purposes has been given by the patent's inventors, one of which is a coauthor of this paper.

[1] Tasks can alternatively be cast as classification problems by predicting this column instead, as was also done for the CM dataset.

A.3 CM

The CM dataset was extracted from the supplementary materials of [47] which can be accessed at `https://www.science.org/doi/full/10.1126/science.aba3304`.

While the data is publicly available, explicit consent to use the data for benchmarking purposes has been given by the authors.

The 2133 sequences used in this paper correspond to the sequences with "comments"

- `natural sequences`,
- `bmDCA designed sequences, T=0.33`,
- `bmDCA designed sequences, T=0.66`.

The sequences with comments `bmDCA designed sequences, T=1` and `designed sequences by profile model` were discarded as they where non-functional and deemed too distant from the wildtype set. These would furthermore have been filtered out at a later stage, since only sequences with values greater than 0.42 were included in the benchmark.

The used natural sequences are found in `aba3304_table_s1.xlsx` while the designed sequences are found in `aba3304_table_s2.xlsx`. The sequences are found aligned in the `Sequence` columns. These were stripped of the `-` token. The target values are found in the `norm r.e.` columns and correponds to the normalized activity relative to Escherichia coli. The proteins were named using the `No.` column while appending `seq_id_`.

To strengthen the MSA, additional members from the chorismate mutase family (IPR036979) were added using the UniProt and InterPro databases [35, 36], where the sequence lengths of the added members were limited to 600 to limit the size of the final alignment, resulting in a sequence pool of 49017 sequences. The sequences are aligned using FAMSA [51].

A.4 PPAT

The PPAT dataset was extracted from [48] and can be accessed at `https://www.science.org/doi/10.1126/science.aao5167`. The dataset file can be found in the supplementary materials in the `aao5167_plesa-sm-tables-s8-s14.xlsx` file, sheet name `S12_PPATdata`. The sequences and target values are in the `seq` and `globalfit14` columns, respectively.

While the data is publicly available, consent to use the data for benchmarking purposes was given by authors of [48].

To strengthen the MSA, additional members from the phosphopantetheine adenylyltransferase family (IPR001980) were added using the UniProt and InterPro databases [35, 36], where the sequence lengths of the added members were limited to 200 to limit the size of the final alignment, resulting in a sequence pool of 17891 sequences. The sequences are aligned using FAMSA [51].

B Reproducibility

All results can be reproduced using the provided shell scripts in the `scripts` directory in the code repository. A description of this process can be found in the repository's README.

Reproducing the main results (i.e., running the regression benchmark given the representations) is cheap and can be achieved in a few hours using multithreading by running the shell script `scripts/reproduce.sh`. The figures and tables can then be generated via `scripts/process_results.sh`. Generating structures and representations is more time consuming, and will be system specific. For further details, see Section B.1. We provide all used representations via the data link in Section A.1. The representations can be downloaded either in bulk with `representations.tar.gz` or individually via the `representations` directory.

All data (raw and curated) can be collected from the links provided in Section A.1. The data can be downloaded in bulk via `data.tar.gz` or individual files can be chosen through the file manager and the `data` directory.

Minor preprocessing (e.g., removing headers to make the Excel-files conform to a tabulated format) might be required before the compilation scripts in `src/data/` can be run. These preprocessed files can be found in the following files in the data repository (see Section A):

- GH114: `data/raw/gh114/gh114.csv`
- CM: `data/raw/cm/cm.csv`
- PPAT: `data/raw/ppat/ppat.xlsx`

Each dataset can then be compiled (i.e., processed and split according to the prescribed dataset splitting procedure) using `src/data/compile_<dataset>.py`. This yields the format described in Section A.

The final partitioning as determined using GraphPart [21] is dependent on the ordering of the input data. Shuffling the datasets, i.e., changing the order of the sequences, will thus slightly change the partitions. We observed only minor changes to the benchmark results given these slight differences.

The CT, ESM-1B, ESM-2, ESM-IF1, MIF-ST, MSA (1-HOT) as well as ESM-IF1 likelihoods can be generated using the `generate_representations.sh` script.

The Evoformer embeddings are extracted during folding using AlphaFold2 by using the `--save-single-representations` flag of ColabFold [50].

To generate the EVE embeddings, the model has to be trained. This can be handled via the `train_EVE_models.sh` script. EVE is trained on each dataset a total of three times using different seeds. The ELBO scores and embeddings are computed/extracted from each trained model. The embeddings are placed in the `0/1/2` subdirectories of `representations/<dataset>/EVE/`.

B.1 Computational resources

A system with an Intel Xeon E5-2680v4 CPU, NVIDIA RTX A5000 GPUs, and 512 GB of RAM was used for benchmarking, computing ESM/MIF-ST embeddings, and training EVE models (though the benchmarking process itself does not utilize GPUs). A system with an AMD EPYC 7642 CPU, NVIDIA A40 GPUs, and 1 TB of RAM was used for protein folding.

A conservative estimate puts the computational resources for each sequence at 4 minutes, which for 2804 sequences results in approximately 187 GPU hours. The majority of this time (>80 %) is spent folding the proteins using AlphaFold2. Running the regression benchmark takes approximately 3 hours using a multithreading-capable CPU.

C Dataset target histograms

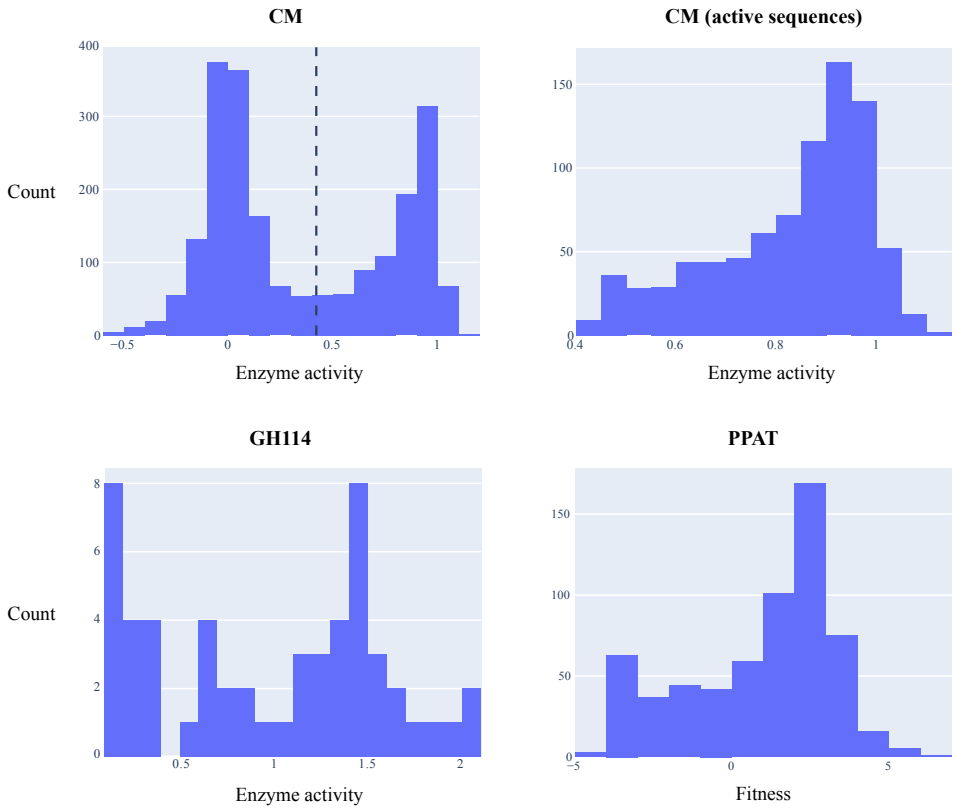

Figure A1: Target histograms of the datasets. CM dataset shows both full dataset prior to filtering and the subset of active sequences that is included in the benchmark. The subset includes only sequences with enzyme activities > 0.42.

D Histograms of cross-validation partitions

D.1 GH114

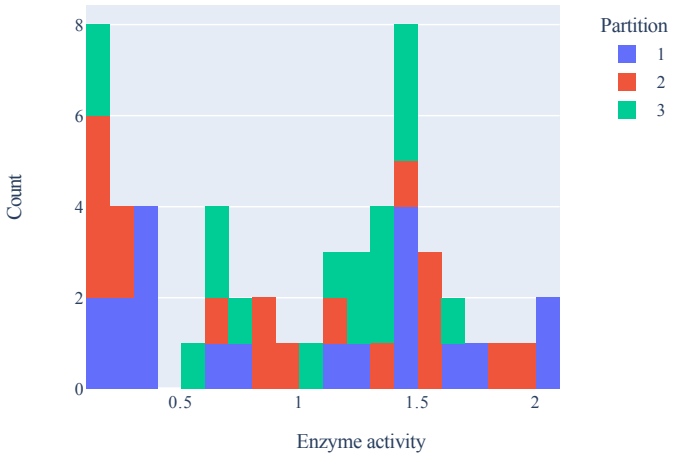

Figure A2: Stacked histogram over distribution of target values for GH114 dataset. Each color correspond to a partition.

D.2 CM

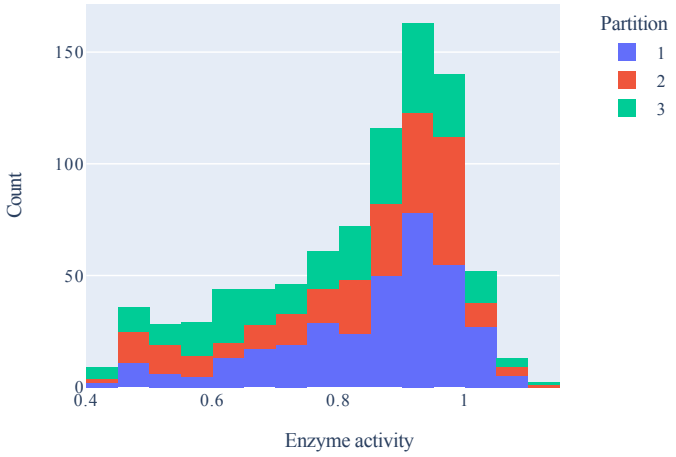

Figure A3: Stacked histogram over distribution of target values for CM dataset. Each color correspond to a partition.

D.3 PPAT

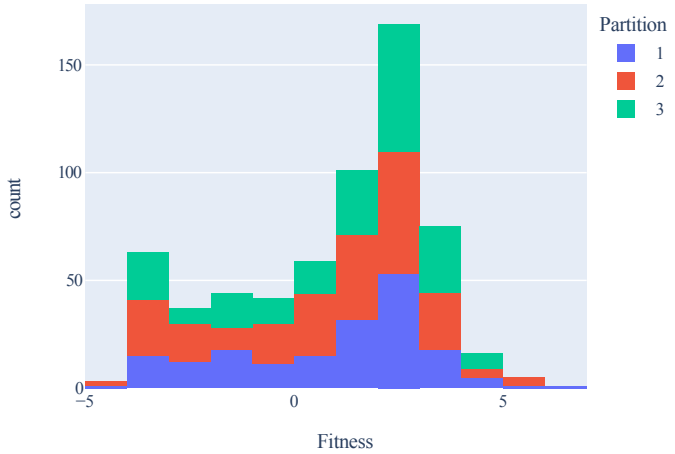

Figure A4: Stacked histogram over distribution of target values for PPAT dataset. Each color correspond to a partition.

E Phylogenetic trees for PPAT dataset

The phylogenetic tree in Figure 2 was constructed based on a family-wide multiple sequence alignment using FastTree [52]. The extracted segment corresponds to the top right quarter.

E.1 Phylogenetic tree colored by dataset partitioning scheme

The phylogenetic tree in Figure A5 is the full version of the leftmost segment in Figure 2.

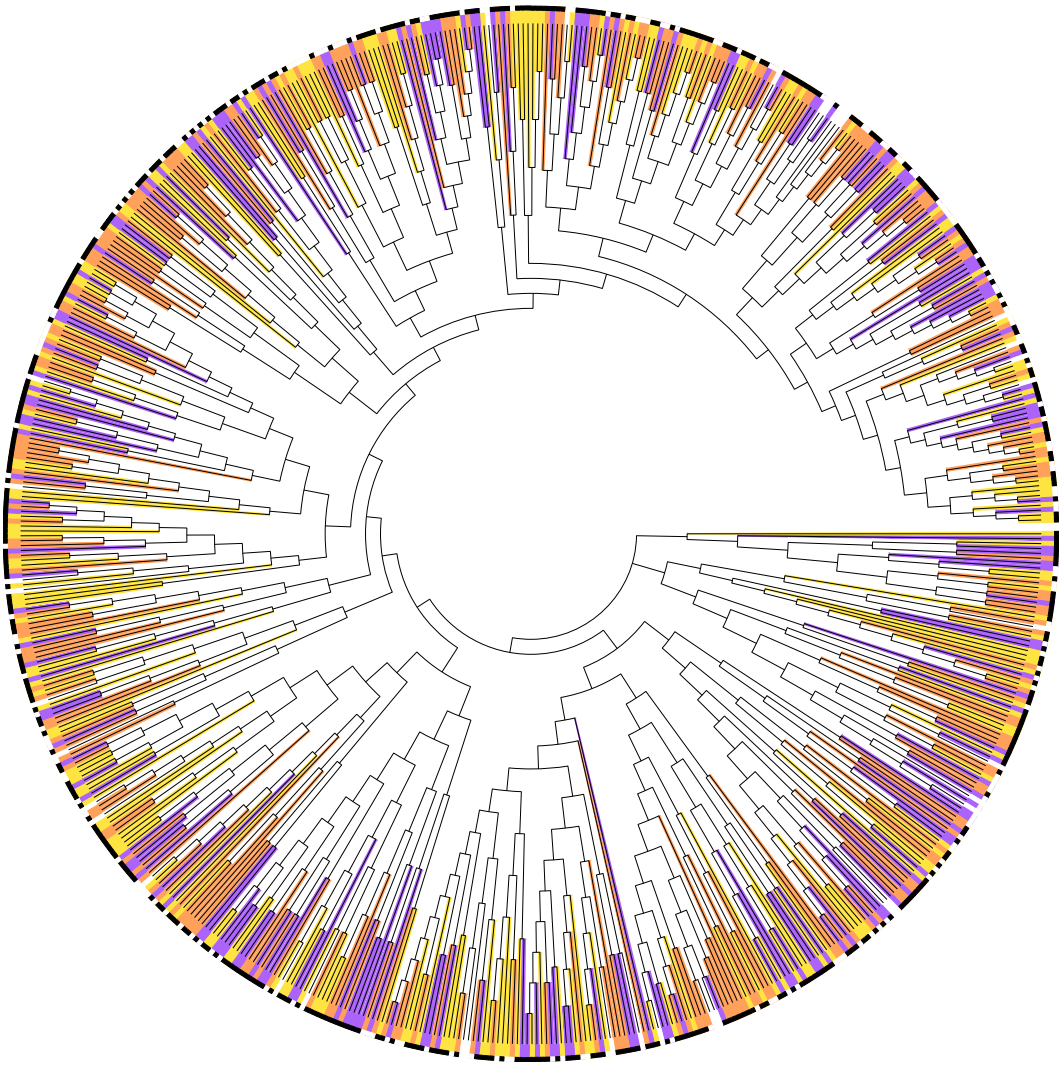

Figure A5: Phylogenetic tree for PPAT dataset. Each sequence is colored according to its partition as computed in the data splitting setup. Black squares indicate high target value while white squares indicate low target value.

E.2 Phylogenetic tree colored by MMseqs-based clustering scheme

The phylogenetic tree in Figure A6 is the same tree as in Figure A5 with a different coloring scheme. The protein sequences were clustered using MMseqs [22] such that at least two large clusters were created. These two large clusters get separate colors, while the remaining minor clusters get a shared color. This represents an alternative dataset splitting scheme. As is apparent from the figure, wide bands of uniformly colored (and thus partitioned) sequences appear. Large subfamilies are all placed in the same partition which means that learning across subfamilies is difficult. The partitioning is furthermore not stratified which might result in low-scoring partitions.

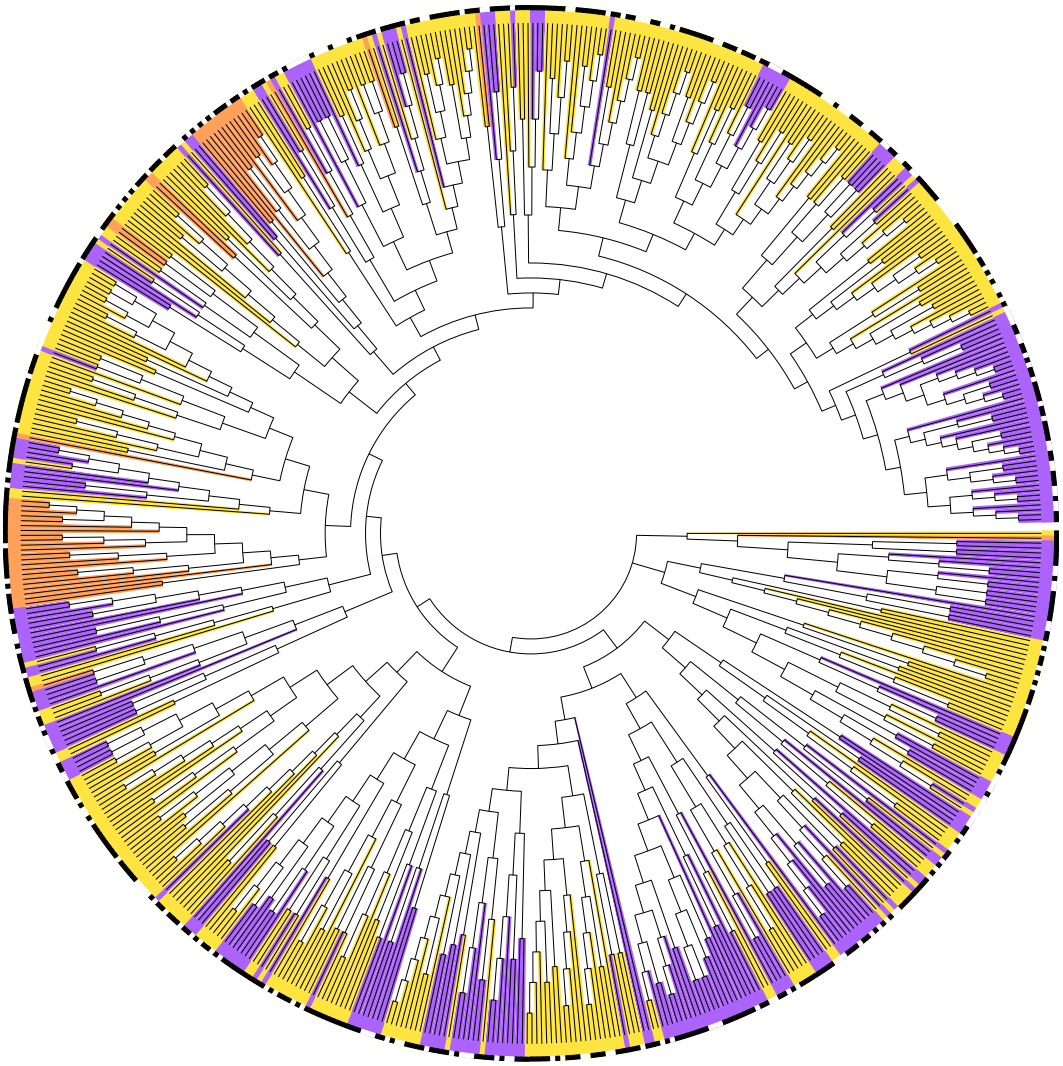

Figure A6: Phylogenetic tree for PPAT dataset. Each sequence is colored according to its partition as computed in the data splitting setup. Black squares indicate high target value while white squares indicate low target value.

E.3 Phylogenetic tree colored randomly

The phylogenetic tree in Figure A7 is the same tree as in Figure A5 with a different coloring scheme. Instead of relying on the prescribed partitioning strategy, each sequence is assigned one of the three colors randomly. This corresponds to generating three random partitions. While the tree looks similar to the one in Figure A5, there is no guarantee that nearly identical sequences are not placed in separate partitions thus allowing for data leakage. There is furthermore no mechanism to ensure properly stratified splits (although this can be handled in most machine learning frameworks).

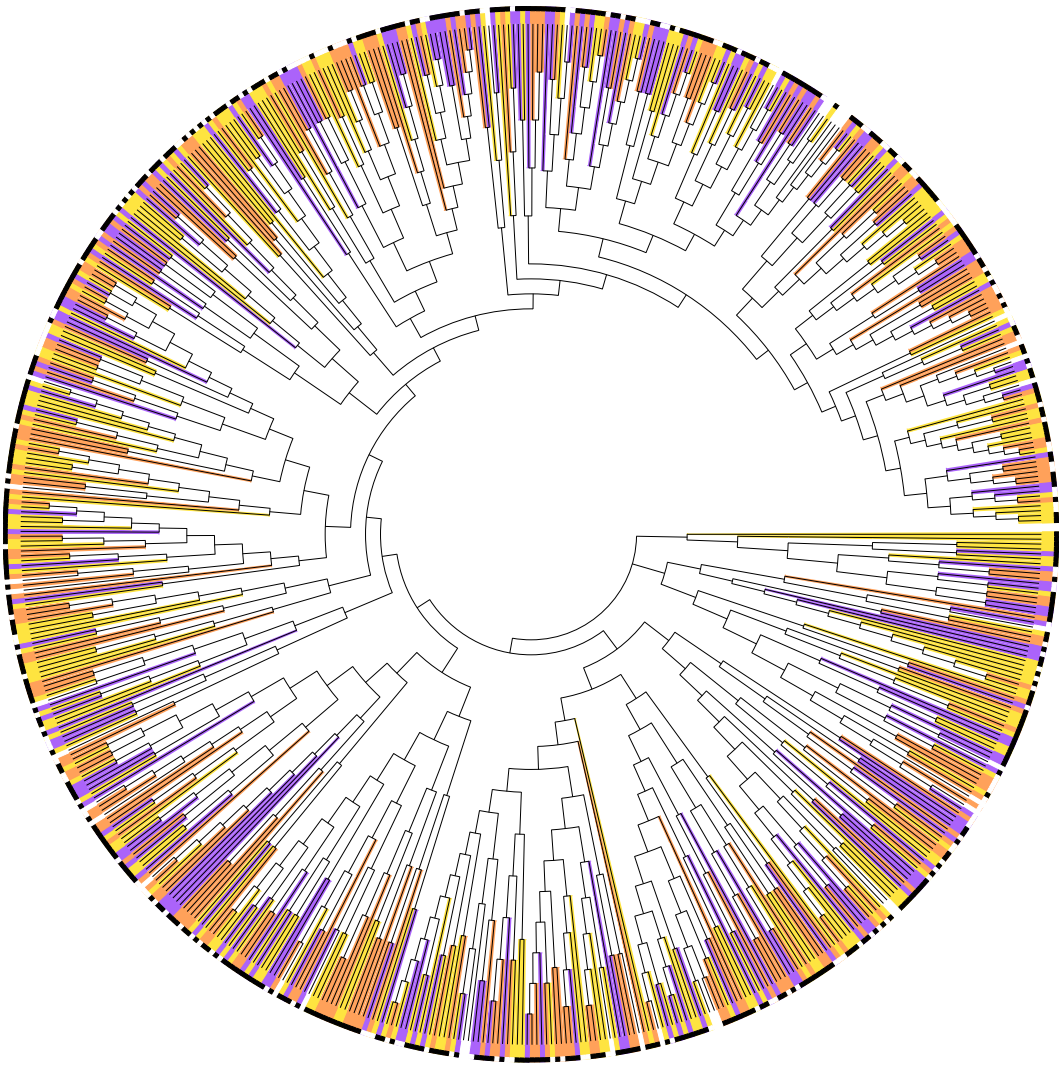

Figure A7: Phylogenetic tree for PPAT dataset. Each sequence is colored randomly corresponding to a random splitting procedure. Black squares indicate high target value while white squares indicate low target value.

F Representation dimensionalities

The dimensionalities of the different protein representations are shown in Table A1. The ESM, Evoformer, and MIF-ST embeddings are mean-pooled along the protein length dimension to obtain fixed inputs.

A multiple sequence alignment (MSA) is generated for each (enriched) protein family, resulting in different dimensionalities. The amino acids are then one-hot encoded to a `MSA_length` $\times$ 20 matrix for each protein, which is in turn flattened to a vector input.

The CT representation consists of two parts: *compositional* and *transitional* descriptors which are concatenated. Each of the two groups in turn consists of seven physicochemical descriptors, relating to overall polarizability, charge, hydrophobicity, polarity, secondary structure, solvent accessibility, and van der Waals volume of a sequence. Each descriptor is in turn represented by three numbers. This yields a total of $2 \times 7 \times 3 = 42$ dimensions. For desciptions of the various features, see `https://github.com/gadsbyfly/PyBioMed/blob/45440d8a70b2aa2818762ceadb499dd3a1df90bc/PyBioMed/PyProtein/CTD.py#L60` and [41].

Table A1: Dimensionalities of the different protein representations.

Representation	**D**	**Note**	**Model name**
CT	42	–	–
ESM-1B	1280	Mean-pooled	`esm1b_t33_650M_UR50S`
ESM-2	2560	Mean-pooled	`esm2_t36_3B_UR50D`
ESM-IF1	256	Mean-pooled	`esm_if1_gvp4_t16_142M_UR50`
MIF-ST	256	Mean-pooled	`mifst`
EVE	50	Seeds `0,1,2`	–
Evoformer (AF2)	256	Mean-pooled	`alphafold2_multimer_v3`
MSA (1-HOT, GH114)	88420	Flattened	6507 sequences in MSA.
MSA (1-HOT, CM)	109980	Flattened	49017 sequences in MSA.
MSA (1-HOT, PPAT)	10140	Flattened	17891 sequences in MSA.

G EVE for protein families

Due to the stochastic training process, we train EVE on each fitness landscape using three different random seeds (`0,1,2`). The reported performance will thus be the average over the predictions using the three different representations for each sequence. While EVE was originally used to predict variant effects of single wildtype proteins, it can be used on any multiple sequence alignment. The built-in preprocessing requires a reference wildtype (query) sequence. This query sequence is then used to trim and otherwise clean the remaining sequences in the MSA. Since no single wildtype is representative for entire protein families, we instead generate an artificial query sequence. Given the full-length MSA, we iterate through all of our labelled sequences (a minor part of the full MSA), and create a query sequence which has an amino acid (we arbitrarily chose '`A`') at any position in the MSA, where any of the labelled sequences also have an amino acid. The remaining positions are filled with gaps. For example, say that sequences `-A-T-H` and `-AT-J-` are two labelled sequences from the MSA. The corresponding query sequence would thus be `-AA-AAA`. The query sequence is only used in the preprocessing, e.g., to conserve the columns, where the labelled sequences have occupancy, and to remove columns where none do. The query sequence is not included in the model training itself. Alternative preprocessing is equally viable which can avoid the creation of the artificial query sequence.

H Tranception for wildtype proteins

We evaluate the fitness of the wildtype sequences using the bidirectional scoring with retrieval using the Tranception L (Large) as defined in the manuscript [42]. This utilises a multiple sequence alignment for each sequence during scoring.

I Regressor hyperparameters

In each cross validation iteration, the regressor is optimized via a grid search. The regressor is trained with all configurations on the training set, and the model providing the lowest mean squared error on the validation set is used to predict on the test set. In addition to the shown results from a random forest regressor, the results from K-nearest neighbour model, a ridge regressor, and a multilayer perceptron (MLP) are also computed. The following hyperparameter grids are used:

- `Ridge(random_state=0)`: Regularization strength was chosen among: 0.0001, 0.001, 0.01, 0.1, 0.2, 0.5, 1, 2, 10, 25, 50, 100.
- `KNeigborsRegressor()`: The number of neighbours was chosen among: 1, 2, 5, 10, 25. For the GH114 dataset, the 10 and 25 options were removed due to the small partition sizes.
- `RandomForestRegressor(random_state=0)`: Minimum samples to split was chosen among 2, 5. Maximum number of features was either `sqrt` or `log2`. Number of estimators was either 100 or 200.
- `MLPRegressor(random_state=0, max_iter=2000)`: hidden layer sizes was either 10 or 100, the L2 regularization strength was set to 0, 0.01, or 0.0001, while the optimizer was either Adam (with gradient descent) or L-BFGS.

We use the scikit-learn implementations of the regressors [53]. The parameters not explicitly defined above are the default parameters. Several other grids for the four models were examined but provided no significant performance increases. The MLP-regressor occasionally experienced convergence issues (with both optimizers).

J Ablation results figure

J.1 Hold-out ablation study on all datasets

The included ablation study shows the results if hold-out validation is applied to the GH114 dataset using a ridge regressor. In Figure A8 is shown the same ablation study on all three datasets using a K-nearest neighbour regressor, a ridge regressor, and a random forest regressor. For EVE, three models have been trained at different initializations thereby explaining the errors bars.

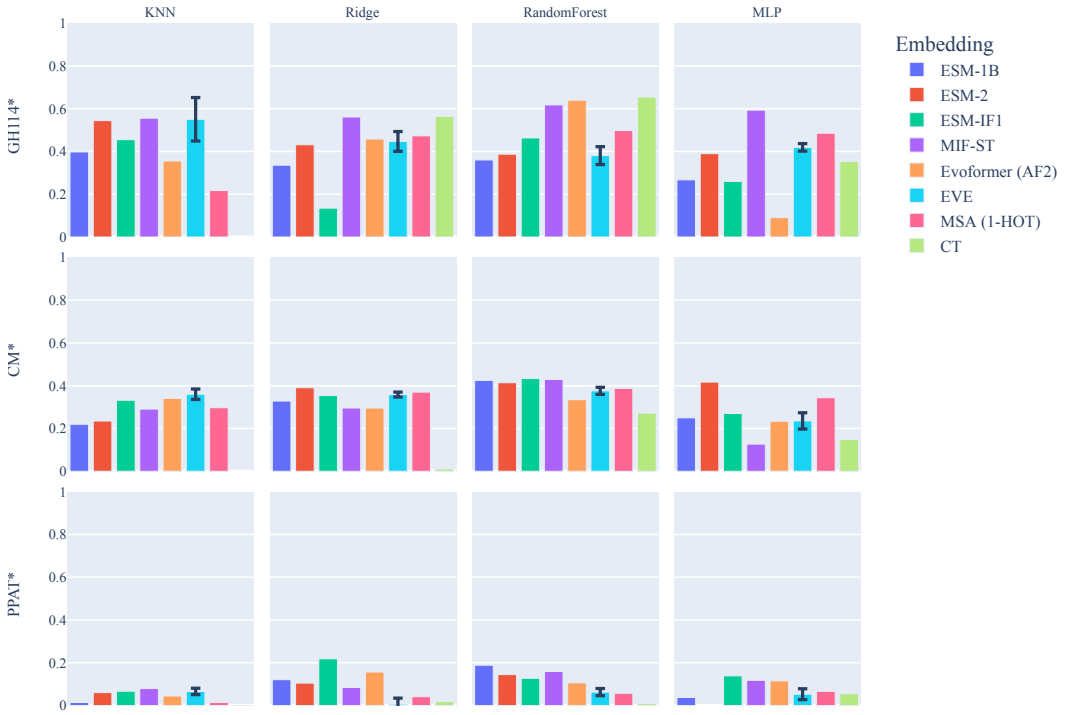

Figure A8: Spearman's rank correlation coefficient between predictions and targets using a hold-out validation approach, grouped by regressor and dataset.

J.2 Random splitting ablation study on all datasets

The included ablation study shows the results if splitting is applied to the PPAT dataset using a ridge regressor. In Figure A9 is shown the same ablation study on all three datasets using a K-nearest neighbour regressor, a ridge regressor, and a random forest regressor.

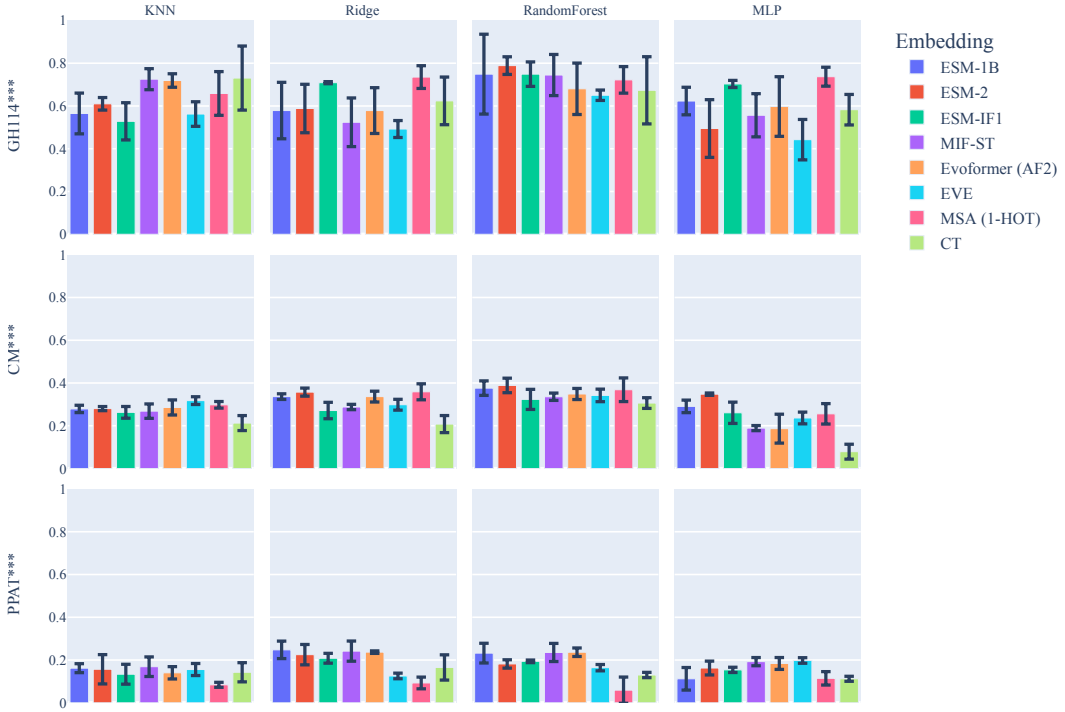

Figure A9: Spearman's rank correlation coefficient between predictions and targets over using a cross-validation approach with randomly sampled partitions repeated on on three random seeds, grouped by regressor and dataset.

K Classification results for CM dataset

Classification was carried out on a combined pool of inactive and active sequences for the CM dataset. The threshold between the two classes is set to 0.42 as described in [47]. The procedure was carried out just as described in Section 3.1 simply with alternative targets and objectives. The results using a K-nearest neighbour classifier, a logistic regression classifier, a random forest classifier, and a multi-layer perceptron are shown in Figure A10. The models were optimized using a binary cross-entropy loss function. The shown metric is Matthew's correlation coefficient. As can be seen from the results, the classification task is significantly easier than the proposed regression benchmark. This supports the notion of carrying out an initial classification prior to performing regression on the subset of active sequences.

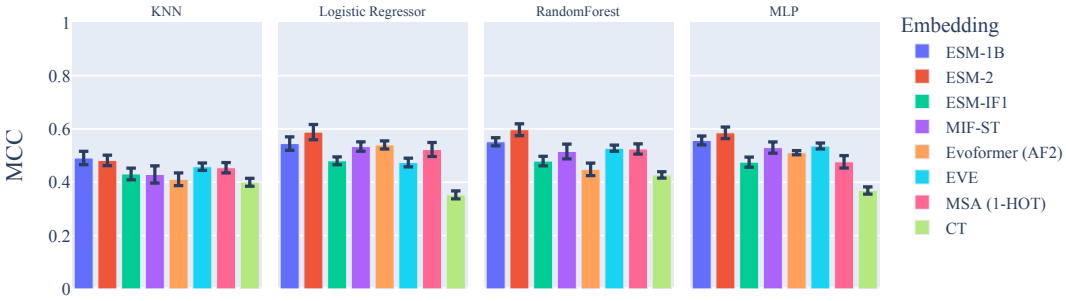

Figure A10: Average Matthew's correlation coefficient between predictions and targets over test partitions. Standard error is shown as vertical bars.

L Additional results

L.1 Results using additional regressors (Spearman)

Test results obtained using a K-nearest neighbour regressor, a ridge regressor (as shown in the main text), a random forest regressor, and an MLP are shown in Figure A11. We observe no systematic differences between the choice of regressor, other than the random forest consistently reaching high performance. This led us to include only the results from the random forest predictor in the main text.

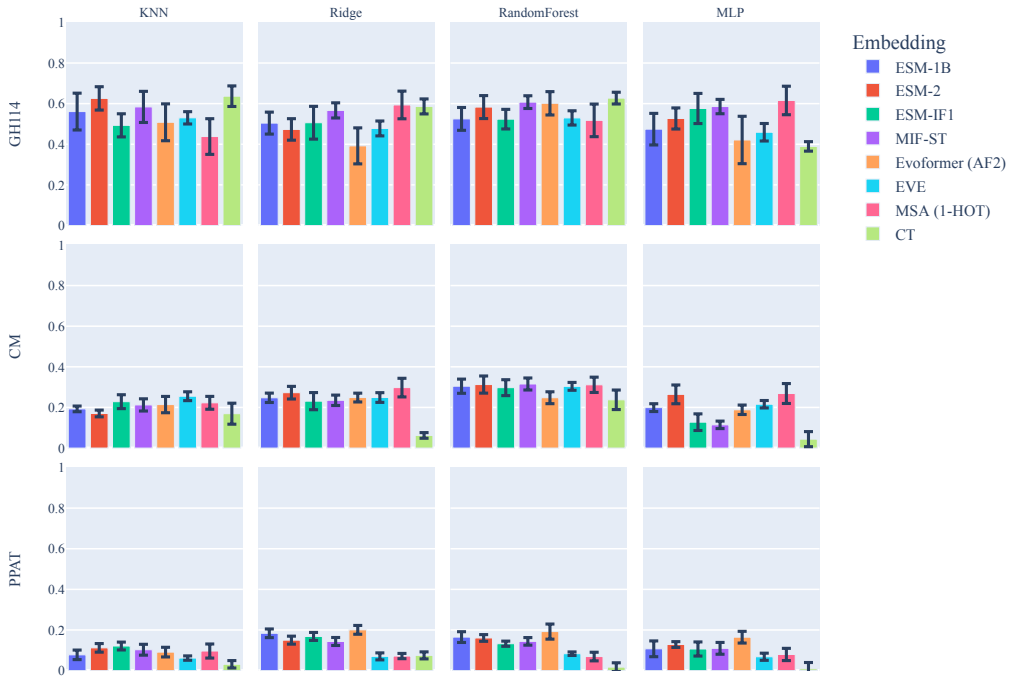

Figure A11: Average Spearman's correlation between predictions and targets over test partitions, grouped by regressor and dataset. Standard error is shown as vertical bars.

L.2 Benchmark results (RMSE)

Table A2: Benchmark results with random forest regressor. Mean RMSE and standard error using cross-validation. Lower is better.

	GH114	CM	PPAT
ESM-1B	**0.43** $\pm$ 0.04	**0.15** $\pm$ 0.0	**2.32** $\pm$ 0.03
ESM-2	**0.43** $\pm$ 0.04	**0.15** $\pm$ 0.0	**2.33** $\pm$ 0.03
ESM-IF1	0.48 $\pm$ 0.04	**0.15** $\pm$ 0.0	**2.33** $\pm$ 0.02
MIF-ST	**0.42** $\pm$ 0.04	**0.15** $\pm$ 0.0	**2.34** $\pm$ 0.03
Evoformer (AF2)	0.45 $\pm$ 0.05	0.16 $\pm$ 0.0	**2.32** $\pm$ 0.03
EVE	**0.44** $\pm$ 0.02	0.16 $\pm$ 0.0	2.41 $\pm$ 0.01
MSA (1-HOT)	**0.45** $\pm$ 0.04	**0.15** $\pm$ 0.0	**2.35** $\pm$ 0.02
CT	**0.45** $\pm$ 0.05	0.16 $\pm$ 0.0	2.41 $\pm$ 0.03

L.3 Results using additional regressors (RMSE)

Test RMSE obtained using a K-nearest neighbour regressor, a ridge regressor (as shown in the main text), a random forest regressor, and an MLP are shown in Figure A12. Note that the y-axes are not shared.

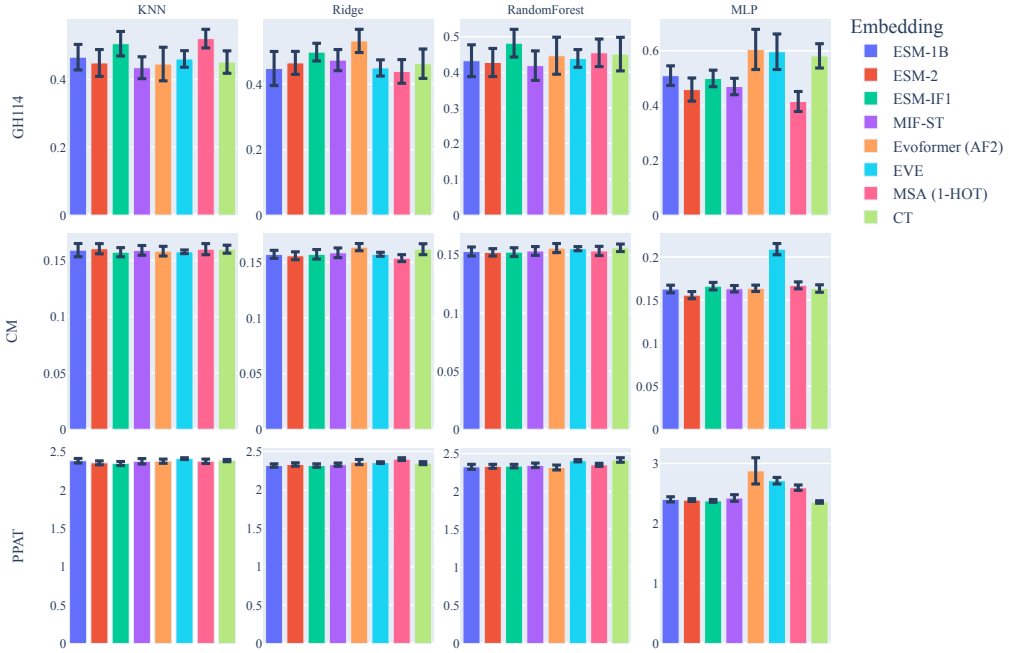

Figure A12: Average RMSE over test partitions, grouped by regressor and dataset. Standard error is shown as vertical bars.