A NEW ULTRA-HIGH-THROUGHPUT ASSAY FOR MEASURING PROTEIN FITNESS

Vikram Sundar
Computational and Systems Biology Program, MIT
Cambridge, MA 02139
vsundar@mit.edu

Boqiang Tu
Department of Biological Engineering, MIT
Cambridge, MA 02139
btu@mit.edu

Lindsey Guan
Computational and Systems Biology Program, MIT
Cambridge, MA 02139

Kevin Esvelt
MIT Media Lab
Cambridge, MA 02139
esvelt@mit.edu

ABSTRACT

Machine learning (ML) for protein design frequently requires large datasets of protein fitness measurements generated by high-throughput experiments; however, publicly available protein fitness datasets generated by deep mutational scanning are noisy and only include 10^3 to 10^5 data points. In this work, we present DHARMA, a new protein fitness assay using molecular recording via base editors and high-throughput sequencing to measure the fitness of up to 10^6 variants. To mitigate noise in DHARMA experiments, we design a Bayesian inference method FLIGHTED that denoises the output of a DHARMA experiment for downstream ML applications. Our results show that DHARMA and FLIGHTED can accurately measure protein fitness with calibrated errors. Using this technology, we generate a new fitness dataset of 160000 TEV protease variants and benchmark a number of standard ML models, including protein language model embeddings, on this dataset. We find that data size is the single most important factor in determining ML model performance and that scaling up protein language models does not currently improve performance. DHARMA and FLIGHTED can help generate more large protein fitness datasets for the ML community.

1 INTRODUCTION

Machine learning (ML) approaches have been remarkably successful in addressing a variety of protein design problems (Johnston et al., 2023; Frappier & Keating, 2021). Many of these approaches rely on data generated by high-throughput experiments (Wu et al., 2019; Johnston et al., 2023); unfortunately, large, reliable datasets of protein fitness measurements from high-throughput experiments are rare, limiting the power of these ML approaches (Fannjiang & Listgarten, 2023). Most such datasets are generated by deep mutational scanning, but these experiments are inherently noisy (Busia & Listgarten, 2023) and often quite small, generally only including 10^3 to 10^5 data points (Notin et al., 2023). In this paper, we present a new assay to generate large protein fitness datasets, new ML methods for denoising the outputs of this assay, and a new large fitness landscape along with a benchmarking study.

Key Contributions We present three key contributions:

1. DHARMA, a new molecular recording assay that can be used for high-throughput measurement of protein fitness and generation of large fitness datasets of up to 10^6 variants.
2. A denoising model FLIGHTED-DHARMA that uses Bayesian inference to estimate fitness error on DHARMA results, thereby making them reliable and trustworthy.
3. A new dataset of 160000 variants of TEV protease fitness, along with benchmarking 39 models on 3 supervised splits of this new dataset.

2 METHODS

2.1 DHARMA

We developed the novel platform DHARMA (direct high-throughput activity recording and measurement assay) to enable massively parallel quantification of protein fitness. In Figure 1a, we quantitatively couple fitness (e.g. enzyme activity) to the mutation rates of a targeted DNA sequence ("canvas") adjacent to the protein coding sequence through a base editor. This approach enables simultaneous retrieval of both variant sequence and corresponding fitness levels from high-throughput DNA sequencing (only), unlike conventional fitness methods such as fluorescent-activated cell sorting (FACS) or phage display (Smith & Petrenko, 1997; Sarkisyan et al., 2016). Such direct coupling allows DHARMA to quantify the fitness of 10^6 variants more easily and cheaply than conventional methods.

With sufficient optimization, any molecular fitness function can be measured using DHARMA as long as the fitness can be coupled to the production or activity of the base editor, a requirement fulfilled by many common functions such as polymerases, proteases, and binders (Miller et al., 2020). We have so far constructed and optimized biological circuits to reliably measure the activities of T7 RNA polymerase (RNAP) and TEV protease. For T7 RNAP, we measured the activities of T7 RNAP variants on the T3 promoter. In this circuit (in Figure 1c), the T7 RNAP is required for the transcription of the base editor, which is under the control of a T3 promoter. After careful tuning of the expression levels of T7 RNAP and base editor, we successfully linked the activities of T7 RNAP variants within a wide dynamic range to mutation rates in the canvas sequence.

By coupling the activity of TEV protease to base editor transcription, we generated a fitness dataset of 160000 TEV protease variants. In this circuit design (in Figure 1d), T7 RNAP, normally repressed by its natural inhibitor T7 lysozyme, becomes functional after the inhibitor is cleaved by active TEV protease variants, thus allowing the transcription of the T7 promoter-driven base editor to proceed.

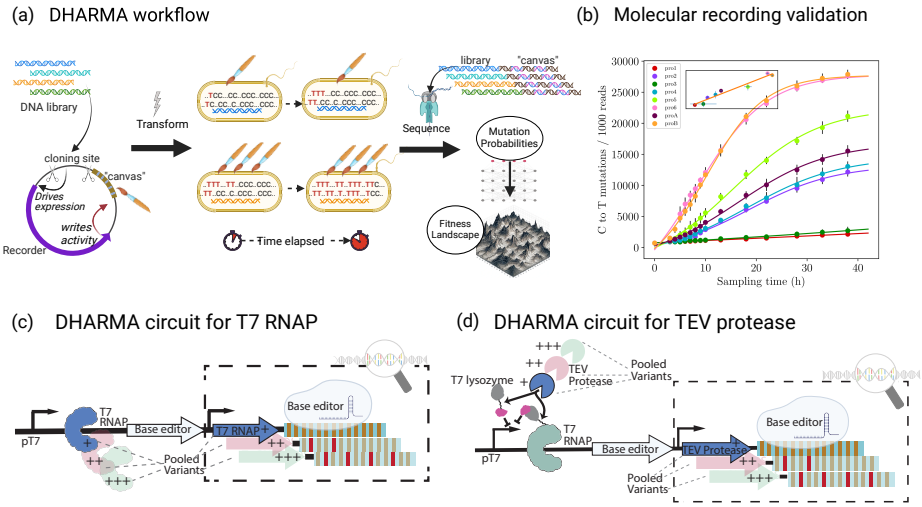

Figure 1: DHARMA Workflow and Circuit Design. (a) The general workflow of DHARMA. (b) Validation of DHARMA using a set of insulated promoters. Inset: Mutation rates correlate with promoter activity. (c) Circuit design for measuring T7 RNA polymerase activity. T7 RNAP variants directly drive the expression of the base editor under the control of a T3 promoter (d) Circuit design for measuring TEV protease activity. Base editor is under the control of a T7 promoter. T7 RNAP, normally inhibited by T7 lysozyme, becomes functional in the presence of active TEV protease.

2.2 FLIGHTED

To account for and reduce noise produced by a typical DHARMA experiment, we developed FLIGHTED (Fitness Landscape Inference Generated by High-Throughput Experimental Data), a Bayesian inference method to generate reliable fitness landscapes with calibrated errors from noisy high-throughput experiments. FLIGHTED is applicable to any high-throughput experiment; here we focus on FLIGHTED-DHARMA, a model trained for DHARMA experiments.

FLIGHTED uses stochastic variational inference (SVI) to model noise generation in a high-throughput experiment. The probabilistic graphical model is designed using knowledge of the noise generation process in a given experiment. The guide, or variational distribution, predicts fitness from the experimental readout using a neural network. SVI has two main advantages: first, model parameters can be fit once to a calibration dataset and then used to estimate a posterior given any subsequent experimental measurements; second, the model and guide can have arbitrary complexity.

The probabilistic graphical model for FLIGHTED-DHARMA is in Figure 2a. The guide predicts both the mean and variance of fitness given a single DHARMA read using a feedforward neural network. In order to train FLIGHTED, we generate a calibration dataset with ground-truth fitnesses and train FLIGHTED in a fully unsupervised fashion on the experimental readouts in the calibration dataset. Ground-truth fitnesses are only used for hyperparameter tuning and to evaluate model accuracy. Learning exclusively from experimental readouts is possible because the biological knowledge used to build the model constrains the space of possible models.

3 RESULTS

3.1 VALIDATION OF DHARMA

For preliminary validation of DHARMA, we cloned 24 previously characterized promoters (Davis et al., 2011) and linked their activities to base editor activity and fluorescence. Figure 1b shows that mutation rates obtained by sequencing correlate well with activity. Therefore, biological circuits that use base editing to record biomolecular fitness can be used for high-throughput profiling of large libraries. However, the inherent stochasticity in base editing rate contributed to high noise in the measurements, as shown in Figure 2b; we designed FLIGHTED to account for this noise.

3.2 VALIDATION OF FLIGHTED

We trained and evaluated FLIGHTED-DHARMA on the calibration dataset of T7 activity, comparing to the FACS readout of a subset of 119 variants as ground-truth fitnesses. Error was minimized in these FACS measurements by using a large number of cells per variant. To eliminate data leakage, FLIGHTED-DHARMA was not trained on any DHARMA read from any of the FACS variants. Figure 2c shows the predicted number of C→T edits with error given a particular fitness.

Fitnesses predicted by FLIGHTED-DHARMA can be related to the FACS readout by an arbitrary non-decreasing function. We used a validation set to fit this to a piece-wise linear function. The initial flat section of the piece-wise linear functions corresponds to low-activity variants where background fluorescence dominates the FACS measurement. We then evaluated model performance on the remaining test set, as shown in Figure 2d. FLIGHTED-DHARMA was compared to a baseline model that predicted the mean and variance of C→T mutation count. We found that FLIGHTED-DHARMA's MSE was 0.72, an improvement over the baseline MSE of 0.78.

We also evaluated the calibration of our predicted variances. In Figure 2e and f, we selected random subsets of DHARMA reads and measured the true and predicted error of FLIGHTED-DHARMA. We then computed z-scores to evaluate model calibration. The mean log likelihood was -3.93, suggesting that FLIGHTED-DHARMA is slightly overconfident but decently well-calibrated. Baseline model performance was substantially worse with a mean log likelihood of -8.00 (see Figure S3). Since our calibration tests included small subsets of DHARMA reads, we can be confident that FLIGHTED-DHARMA will produce reasonable errors even when given few DHARMA reads. This tells us whether a given number of DHARMA reads is sufficient for a fitness measurement.

Finally, since FLIGHTED uses a biological model of DHARMA, we can examine the parameters of that biological model directly. In Figure 2g, we compute the logit of the C→T edit probability at a

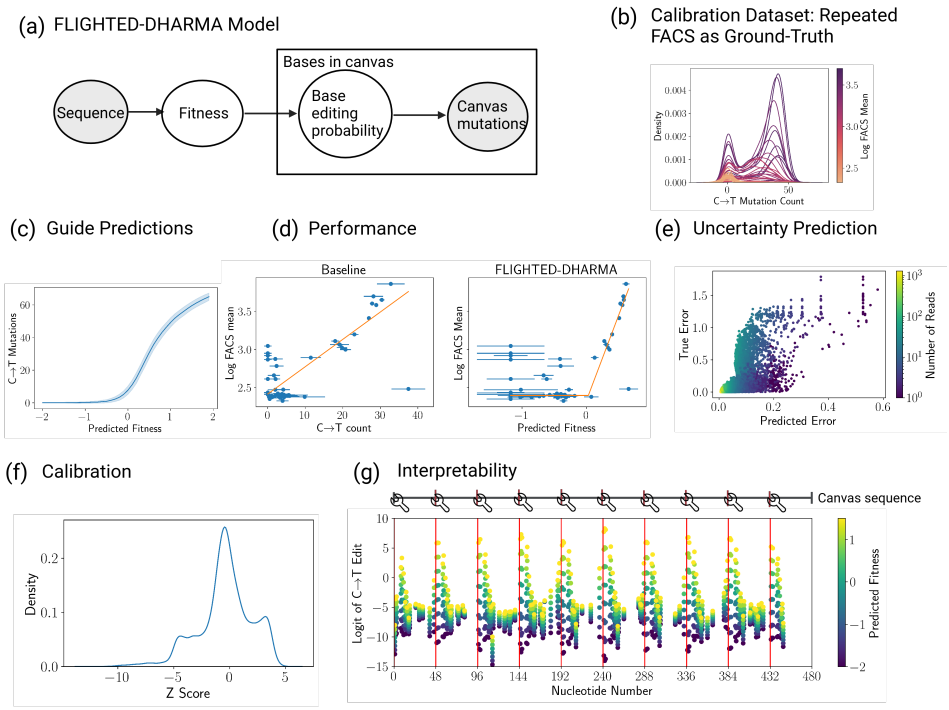

Figure 2: FLIGHTED-DHARMA Model Performance. (a) The probabilistic graphical model for FLIGHTED-DHARMA. (b) Noise in DHARMA as a function of true fitness. (c) FLIGHTED-DHARMA's prediction of number of C→T mutations as a function of fitness. (d) FLIGHTED-DHARMA improves performance of fitness predictions when compared to a baseline model. (e) Predicted and true uncertainty from FLIGHTED-DHARMA correlate well for even small numbers of reads. (f) FLIGHTED-DHARMA is well-calibrated. (g) Canvas nucleotides most likely to be edited according to FLIGHTED-DHARMA are found at guide RNA binding sites, as expected.

given canvas residue as a function of fitness. The probability of a C→T edit increases and is more fitness-dependent every 48 residues. These are precisely the locations targeted by the base editor.

3.3 Benchmarking on a DHARMA Dataset

Based on DHARMA and FLIGHTED as described above, we profiled the enzymatic activity of 160,000 variants of TEV protease on its wild type substrate and used FLIGHTED to generate the largest TEV protease fitness dataset to date. We performed site-saturation mutagenesis on 4 amino acid residues (T146, D148, H167, and S170) in the TEV protease S1 pocket, which is known to interact with the P1 residue on the substrate and determine substrate specificity. We found that 99.8% of the library members recorded fitness values on the wild-type substrate that were less than that of the wild-type enzyme (Figure 3a), as is expected given that most mutations are deleterious. See Supplementary Section B.3 for more discussion about the impact of read count on measured fitness.

Our experiments with DHARMA and FLIGHTED enabled us to reconstruct a complete 4-site local fitness landscape, which represents a new resource for benchmarking ML models for fitness prediction. We consequently benchmarked a number of publicly available ML models, focusing on the supervised problem of predicting fitness given a small amount of TEV protease fitness data. We accordingly split our dataset into three benchmarks: training sets of mutation distance ≤ 1 (one-vs-rest), ≤ 2 (two-vs-rest), and ≤ 3 (three-vs-rest) from the wild type. As shown in Figure 3b, each model consisted of an embedding generated by a one-hot baseline or by a protein language model, and a task-specific top model trained on the TEV data that was either a linear layer, an augmented model (Hsu et al., 2022), a feedforward neural network (FNN), a convolutional neural

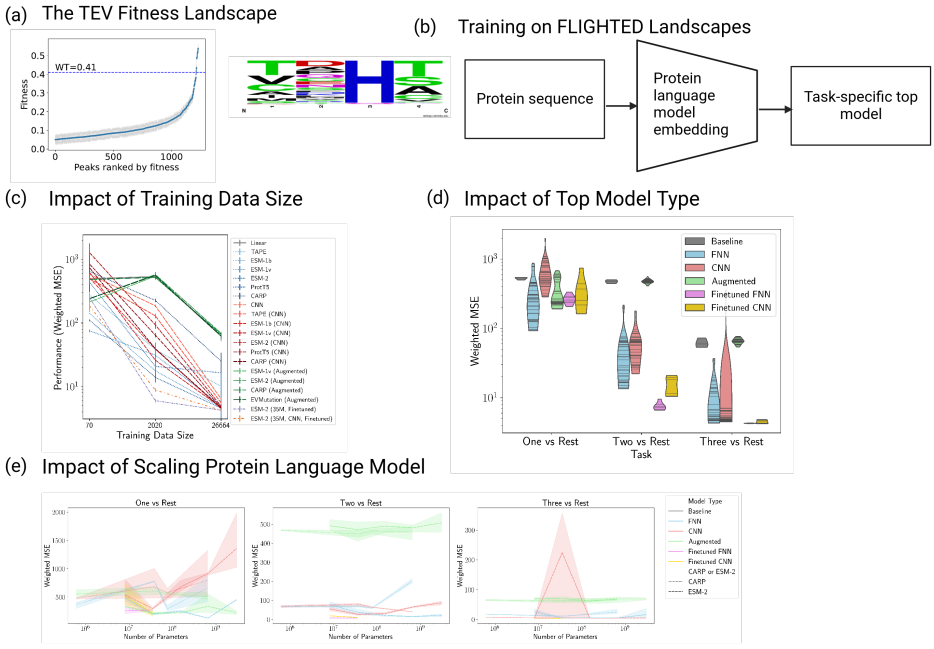

Figure 3: Benchmarking on a TEV Landscape produced by FLIGHTED and DHARMA. (a) Fitness peaks in reconstructed landscape; consensus sequence of the top 100 most active variants in the TEV landscape. (b) General model diagram used for benchmarking. (c) Increasing training data size substantially improves model accuracy on the same test set. (d) The best-performing top models are FNNs and CNNs, with CNNs being the very best on three-vs-rest. Fine-tuning generally improves performance. (e) Scaling up protein language models to generate embeddings does not substantially improve performance.

network (CNN), or a finetuned FNN or CNN. Model performance is measured as weighted MSE as a proxy for log likelihood.

We first examined the effect of training data size, controlling the test set to include only quadruple mutants in Figure 3c. As training data increases from single to double to triple mutants, model error decreases exponentially. This dramatic improvement in model accuracy indicates an increasing ability to generalize and highlights the importance of high-throughput experimental methods like DHARMA: with more data, our models become substantially more powerful.

We then examined what model architecture choices mattered the most for determining performance. We found in Figure 3d that the top model was the most impactful, with all models equally unable to learn from single mutants, FNNs and CNNs performing similarly on two-vs-rest, and CNNs outperforming FNNs on three-vs-rest. Fine-tuning generally results in an improvement in performance, especially for FNNs on small datasets. The choice of protein language model was substantially less important, as can be seen in the full performance table presented in the appendix. More surprisingly, increasing the number of parameters in the protein language model did not have a large impact on performance for most models on our dataset. In Figure 3e, we ran published variants of CARP and ESM-2 with smaller numbers of parameters (Yang et al., 2022; Lin et al., 2022). We found no substantial or consistent performance improvement, across models ranging from millions to billions of parameters. Our results suggest that scaling up protein language models will not improve performance in predicting protease activity given current data limitations.

4 Discussion

We have demonstrated that DHARMA can be used as a high-throughput measurement of protein fitness for a variety of applications, including T7 RNAP and TEV protease. DHARMA is in princi-

ple applicable to any biomolecular activity that can be linked to transcription, which also includes protein-protein binding and solubility (Miller et al., 2020). FLIGHTED-DHARMA enables the use of DHARMA as a data generation method for ML; we can provide calibrated error estimates that inform practitioners when they have enough DHARMA reads to make a reliable fitness estimate. DHARMA and FLIGHTED can be used to cheaply generate large datasets of up to 10^6 variants.

Our benchmarking results support two important conclusions about the development of future protein fitness ML models. First, data size is currently the most important factor, underscoring the importance of gathering large, high-quality datasets through methods such as DHARMA. Second, the top model architecture matters much more than the embedding, so practitioners should focus on development of top models and less on development of larger protein language models.

Author Contributions

V.S., B.T., and K.M.E. conceived the study. V.S. designed and implemented FLIGHTED and B.T. developed DHARMA and gathered the experimental data, both under the supervision of K.M.E. L.G. aided in benchmarking and optimization of FLIGHTED-DHARMA. V.S. wrote the first draft of the paper. All authors contributed to editing and approved the submission.

Acknowledgments

V.S. acknowledges funding from the Fannie and John Hertz Foundation. B.T. acknowledges funding from the National Science Foundation Graduate Research Fellowship under Grant No. 2141064. L.G. acknowledges funding from the National Institutes of Health under Grant. No. T32GM087237. We are grateful to gifts from Open Philanthropy Project and the Aphorism Foundation (to K.M.E.).

Data and Code Availability

Model weights for FLIGHTED-DHARMA have been released at this Zenodo repository. The FLIGHTED TEV landscape and all TEV models have been released at this Zenodo repository. All code necessary to run FLIGHTED and reproduce all figures in this paper has been released in at this Github repository.

References

Eli Bingham, Jonathan P Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D Goodman. Pyro: Deep Universal Probabilistic Programming. *Journal of Machine Learning Research*, 20:1–6, 2019.

Akosua Busia and Jennifer Listgarten. Model-based differential sequencing analysis, April 2023. URL http://biorxiv.org/lookup/doi/10.1101/2023.03.29.534803.

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, January 2022. URL https://www.biorxiv.org/content/10.1101/2021.11.09.467890v2. Section: New Results Type: article.

Joseph H. Davis, Adam J. Rubin, and Robert T. Sauer. Design, construction and characterization of a set of insulated bacterial promoters. *Nucleic Acids Research*, 39(3):1131–1141, February 2011. ISSN 1362-4962. doi: 10.1093/nar/gkq810.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(8), 2021. doi: 10.1101/2020.07.12.199554. URL http://biorxiv.org/lookup/doi/10.1101/2020.07.12.199554.

Clara Fannjiang and Jennifer Listgarten. Is novelty predictable?, June 2023. URL http://arxiv.org/abs/2306.00872. arXiv:2306.00872 [cs, q-bio].

Vincent Frappier and Amy E. Keating. Data-driven computational protein design. *Current Opinion in Structural Biology*, 69:63–69, August 2021. doi: 10.1016/J.SBI.2021.03.009. URL https://doi.org/10.1016/j.sbi.2021.03.009. Publisher: Elsevier Ltd.

Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P. I. Schärfe, Michael Springer, Chris Sander, and Debora S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, February 2017. ISSN 1546-1696. doi: 10.1038/nbt.3769.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Learning protein fitness models from evolutionary and assay-labeled data. *Nature Biotechnology*, 2022. URL https://www.nature.com/articles/s41587-021-01146-5.

Brendan J. Hussey and David R. McMillen. Programmable T7-based synthetic transcription factors. *Nucleic Acids Research*, 46(18):9842–9854, October 2018. ISSN 1362-4962. doi: 10.1093/nar/gky785.

Kadina E. Johnston, Clara Fannjiang, Bruce J. Wittmann, Brian L. Hie, Kevin K. Yang, and Zachary Wu. Machine Learning for Protein Engineering, May 2023. URL http://arxiv.org/abs/2305.16634. arXiv:2305.16634 [q-bio].

Eric Jones, Travis Oliphant, Pearu Peterson, and Others. SciPy: Open source scientific tools for Python, 2001. URL http://www.scipy.org/.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction, July 2022. URL https://www.biorxiv.org/content/10.1101/2022.07.20.500902v1.full.pdf.

Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *Proceedings of the 35th Conference on Neural Information Processing Systems*, 2021.

Shannon M. Miller, Tina Wang, and David R. Liu. Phage-assisted continuous and non-continuous evolution. *Nature Protocols 2020 15:12*, 15(12):4101–4127, November 2020. ISSN 1750-2799. doi: 10.1038/s41596-020-00410-3. URL https://www.nature.com/articles/s41596-020-00410-3. Publisher: Nature Publishing Group.

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 Design and Fitness Prediction. In *NeurIPS 2023 Track on Datasets and Benchmarks*, December 2023. doi: 10.1101/2023.12.07.570727. URL http://biorxiv.org/lookup/doi/10.1101/2023.12.07.570727.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. 2019.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. Evaluating protein transfer learning with TAPE. In *Advances in Neural Information Processing Systems*, 2019. doi: 10.1101/676825. URL https://www.biorxiv.org/content/10.1101/676825v1. arXiv: 1906.08230 ISSN: 1049-5258.

Alexander Rives, Siddharth Goyal, Joshua Meier, 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):622803, April 2021. doi: 10.1101/622803. URL https://doi.org/10.1101/622803. Publisher: bioRxiv.

Karen S. Sarkisyan, Dmitry A. Bolotin, Margarita V. Meer, Dinara R. Usmanova, Alexander S. Mishin, George V. Sharonov, Dmitry N. Ivankov, Nina G. Bozhanova, Mikhail S. Baranov, Onuralp Soylemez, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva, Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, May 2016. ISSN 14764687. doi: 10.1038/nature17995. URL https://www.nature.com/articles/nature17995. Publisher: Nature Publishing Group.

George P. Smith and Valery A. Petrenko. Phage Display. *Chemical Reviews*, 97(2):391–410, April 1997. ISSN 0009-2665. doi: 10.1021/cr960065d. URL https://doi.org/10.1021/cr960065d. Publisher: American Chemical Society.

Zachary Wu, S. B. Jennifer Kan, Russell D. Lewis, Bruce J. Wittmann, and Frances H. Arnold. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences of the United States of America*, 116(18):8852–8858, April 2019. ISSN 10916490. doi: 10.1073/PNAS.1901979116/SUPPL_FILE/PNAS.1901979116.SAPP.PDF. arXiv: 1902.07231 Publisher: National Academy of Sciences.

Kevin K Yang, Alex X Lu, and Nicolo Fusi. Convolutions are competitive with transformers for protein sequence pretraining, 2022. URL https://www.biorxiv.org/content/10.1101/2022.05.19.492714v1.full.pdf.

A Detailed Methods

A.1 DHARMA Circuit Design and Preliminary Validation

Molecular recording allows direct mapping of biomolecular activities to sequence identity by encoding activity information as mutation rate and patterns of the recording medium. In this method, the activity of interest is coupled to the expression of a base editor, which introduces mutations on a canvas sequence. The number of mutations accumulated on the canvas sequence is then used to infer the activity of the enzyme of interest. To test this concept and begin calibrating our method, we initially designed plasmid constructs consisting of a base editor expressed under the control of a constitutive promoter, a canvas with multiple target sites for activity recording, and a GFP or luxAB expressed under the control of the same promoter to serve as a signal reference for relative activity quantification. In the presence of sgRNA, such constructs simultaneously measure the promoter strength using both sequencing and either fluorescence or luminescence intensity.

A.2 T7 Polymerase Dataset Generation

We engineered a biological circuit to associate the enzymatic characteristics of T7 RNA polymerase variants with mutations accumulating on a designated DNA sequence, known as the canvas. T7 RNA polymerase is an enzyme responsible for transcribing DNA into RNA, and we constructed a library of variants with different recognition and activity profiles.

The transcription of the base editor, an enzyme that can induce mutations, is controlled by a T3 promoter. This promoter is selectively recognized by a subset of the T7 RNA polymerase variants. To moderate the expression levels of the T7 polymerase library and prevent rapid saturation of the canvas—which would compromise the collection of meaningful activity data—we used both a weak constitutive promoter and a weak ribosomal binding site.

We constructed a library of 8,000 T7 RNAP variants using a commercial libraries of ssDNAs with the objective of site-saturation mutagenesis of N748, R756 and Q758, which were shown to be key residues for T7 RNAP activity on T3 promoters (Hussey & McMillen, 2018). We incorporated this biological circuit into cells already containing plasmids that express the single guide RNA (sgRNA), through a technique called electroporation. The sgRNA serves to guide the base editor to the specific canvas sequence where mutations are to be introduced. After electroporation, the cells were grown continuously in a bioreactor. Following this growth period, the region of the T7 polymerase library and the canvas with mutations was selectively amplified as contiguous fragments of DNA. The amplified material was then subjected to long-read Nanopore sequencing for detailed analysis of the mutations and activities of different T7 RNA polymerase variants.

To facilitate data analysis, we implemented a data processing pipeline. Its core function is to identify, tabulate, and assign mutations present in each sequencing read to the corresponding library member. This assignment is based on internal barcodes represented as degenerate codons in the T7 RNA polymerase sequence. The pipeline accepts raw sequence reads in standard genomic formats and performs length-based filtering and optional sequence trimming. An algorithm incorporating local sequence alignment is used for barcode recognition and classification during the demultiplexing of reads. Additionally, each read is aligned to the reference sequence of the canvas to identify the location of each C→T mutation, which is then stored in a matrix for downstream ML model training.

A.3 THE FLIGHTED-DHARMA MODEL

All models were implemented using the probabilistic programming package Pyro (Bingham et al., 2019) alongside PyTorch (Paszke et al., 2019). Use of Pyro makes model development substantially easier due to rapid iteration of model architectures.

The model is an implementation of the probabilistic graphical model shown in Figure 2a in Pyro. The sequence-to-fitness function is simply a dictionary lookup, with the fitness of each variant i defined as a parameter to be optimized. Fitness means m_i and variances σ_i are predicted by the sequence-to-fitness function/dictionary, and the variance is transformed by a softplus function F to ensure it is positive. The fitness of each variant is then sampled from a normal distribution $\mathcal{N}(m_i, F(\sigma_i))$ and fitnesses are clamped at -2.

For each position i in the canvas, we set a learnable parameter r_i for a baseline rate of generic mutations (which accounts for most sequencing error from the long-read sequencing). For positions that are cytosines, we set a learnable parameter m_i for the fitness-dependent slope and b_i for the intercept. Then for all cytosines, given a fitness f of the variant, the logit of the C→T edit is set to $m_i + b_i f$ and the logit of any other mutation or deletion is r_i. For non-cytosine residues, we simply have a logit of any mutation set to r_i. We sample from the one-hot categorical distribution for each residue independently to get the output DHARMA read, i.e. from $\text{Cat}((1, m_i + b_i f, r_i))$ for cytosine residues and $\text{Cat}((1, -\infty, r_i))$ for non-cytosine residues.

The guide predicts fitness (with variance) from a single DHARMA read. We used a feedforward network with 2 layers with a hidden dimension of 10 and a ReLU activation between the two layers. To simplify the learning problem, only cytosine residues were fed into the guide and each position was featurized as either a C→T edit or a non-C→T edit (including both no mutation or other mutations or deletions). Variances were transformed under the softplus function. Fitnesses of variants with multiple reads can be predicted by multiplying the appropriate predicted Gaussians of the individual read.

The ELBO loss used 1 ELBO particle. The landscape model (sequence-to-fitness function) learning rate was 10^{-2}, the learning rate elsewhere was 10^{-4}, and the batch size was 2. We used a plateau learning rate scheduler with a patience of 1 epoch and a factor of 0.1, stepped based on the training loss (not validation loss). The model was trained for 25 epochs.

To minimize data leakage, all reads corresponding to sequences in the fluorescence dataset were eliminated from training. 10% of the remaining reads were held out as a validation set to pick the optimal epoch for the model.

Many of the hyperparameters were carefully tuned using a grid search. For tuning, we wanted to use the MSE with the FACS data, so we needed to fit the predicted fitnesses to the FACS data. We did not use Spearman correlation so we could measure true error (as opposed to rank-order error) and so we could account for situations where the function relating the FACS data to predicted fitness was not increasing (as happened here). For this, we split out a 50% of the FACS data to use as a fitness regression/validation set and evaluated the mean of the logarithm of all FACS samples. Each model was given the option of fitting using either a piecewise linear function or a linear regression. The linear regression was fit normally. The piecewise linear function corresponded to the functional form

$$f(x) = \begin{cases} a & x \leq x_c \\ m(x - x_c) + a & x > x_c. \end{cases}$$

This ensured that for low fitnesses, the predicted fluorescence was constant, as would be expected for background fluorescence. We then used orthogonal distance regression, as implemented in `scipy`,

to fit the piecewise linear function accounting for errors in both predicted fitness and the FACS data. (Jones et al., 2001) Between the linear regression and piecewise linear fit, the function with the lower MSE on the validation data was selected.

This validation set MSE was also used for selecting hyperparameters, leaving the remaining 50% of the FACS data as a test set used solely for evaluating model performance as shown in Figure 2d. Hyperparameters tuned included the batch size, learning rate, learning rate scheduler step, and number of layers in the guide model. We also evaluated the optimal number of hours to grow the cells in the bioreactor, as an example of an experimental parameter than can be tuned using FLIGHTED.

A.4 Model Evaluation

Model performance was measured with the 50% held-out test set of the FACS data, as described above, using the fitnesses predicted by the guide model in FLIGHTED-DHARMA and the fitness-to-fluorescence function fit on the validation set. The results of that performance evaluation are shown in Figure 2c.

We then evaluated calibration of the model. To increase the number of data points (since we only had 192 variants measured through FACS), we subsampled subsets of reads for each variant. Specifically, for all test set variants, we sampled 10 subsets each of sizes ranging from 1 to the maximum possible number of reads. We then predicted the fitness of each variant using the given subset of reads with the guide model. We computed the true fitness using the FACS data, eliminating any data points where the true fitness was below the baseline of the piecewise linear function. We then computed the z-score by comparing this predicted and true fitness. This generates the plots shown in Figure 2d.

A.5 TEV Dataset Generation

High-throughput quantification of TEV protease fitness using DHARMA was made possible by a biological circuit that couples the activity of TEV protease to base editor transcription driven by a T7 promoter. As part of the circuit, the T7 RNA polymerase, normally repressed by its natural inhibitor T7 lysozyme, becomes functional after the inhibitor is disabled by active TEV protease variants, thus allowing the transcription of base editor to proceed, which in turn introduce mutations to the canvas. We engineered a variant of T7 lysozyme in which the TEV protease substrate sequence is inserted in the middle of the coding sequence. This modified T7 lysozyme, expressed on a separate plasmid under the control of a medium constitutive promoter, together with the T7 lysozyme tethered to the T7 RNAP, provides an enhanced dynamic range of base editing coupled to the activity of TEV protease variants.

The library of TEV protease used in this study was obtained by performing site-saturation mutagenesis on 4 amino acid residues (T146, D148, H167, and S170) in the TEV protease S1 pocket, which is known to interact with P1 residue on the substrate and determine substrate specificity. Briefly, NNK degenerate primers was used to introduce mutations at the targeted residues in a PCR reaction. The pool of amplicons was then cloned into a Golden Gate vector comprising the rest of the TEV protease expression cassette, the sgRNA and the canvas sequence. Commercial electrocompetent cells were then transformed with the cloning reaction, selected with appropriate antibiotics on agar plate overnight and subjected to DNA extraction. The resulting library was then sequenced to assess for bias and coverage. After quality control, the TEV protease library was transformed into electrocompetent cells that already express the base editor, the T7 RNAP and the engineered T7 lysozyme. The transformants were then selected with appropriate animate and grown continuously in a bioreactor. Multiple time points were taken during this growth period to find the optimal incubation time. The region of the plasmid containing TEV protease library and the canvas sequence was selectively amplified as contiguous fragments of DNA. To minimize the amplicon size, nucleotides not of interest were removed via self-circularization and re-amplification. The final amplified material was then subjected to long-read nanopore sequencing to simultaneously retrieve both variant identity and the mutations on the canvas sequence for each individual library member. Sequencing data processing and analysis was performed as described in the T7 dataset generation section (A.2).

A.6 BENCHMARKING MODELS ON THE TEV DATASET

The TEV landscape was generated by processing the raw TEV data with FLIGHTED-DHARMA as trained on the T7 dataset. We split the dataset by mutation distance from the wild-type to create a one-vs-rest, two-vs-rest, and three-vs-rest data split. We then split out a random 10% of the training data to use as validation data. Models trained on the TEV dataset are trained with a weighted mean-squared-error (MSE) loss, weighted by the inverse variance. Weighted MSE is used to account for the variance, assuming that the likelihood of the data is the same as that of a normal distribution with the provided variance. Performance results are weighted MSEs for datasets with corrections. In Figure 3c, we use a separate test set consisting of only quadruple mutants (i.e. the test set in the three-vs-rest split) for all models.

Our models are inspired by but go beyond the models proposed in FLIP (Fitness Landscape Inference for Proteins) (Dallago et al., 2022). The linear regression model (labeled Linear) takes one-hot embeddings of the full sequence and runs them through 1 linear layer. It uses an Adam optimizer with a learning rate of 10^{-2}, a batch size of 256, and a weight decay of 1. The CNN model (labeled CNN) takes one-hot embeddings of the full sequence and has 1 convolutional layer with 1024 channels and filter size 5, and same padding. It then has a 1D batch normalization layer and a ReLU activation, followed by an embedding neural network consisting of a linear layer to 2048 dimensions and a ReLU activation. Then there is a max-pooling layer over residues, a dropout layer with probability 0.2, and a final linear layer for the output. The CNN is trained with a batch size of 256 and an Adam optimizer with a learning rate of 10^{-3} and weight decay of 0 for the convolutional layer, a learning rate of $5 * 10^{-5}$ and weight decay of 0.05 for the embedding layer, and a learning rate of $5 * 10^{-6}$ and weight decay of 0.05 for the output layer. Unlike the original FLIP paper, we did not use early stopping and trained all models for a full 500 epochs, using validation set performance to select the optimal model.

The models labeled TAPE, ESM-1b, ESM-1v, ESM-2, ProtT5, and CARP all use mean embeddings across the entire sequence that are fed into a feedforward neural network to compute the output (Rives et al., 2021; Meier et al., 2021; Rao et al., 2019; Elnaggar et al., 2021; Yang et al., 2022). The output feedforward neural network has 2 layers with a hidden dimension equal to the embedding dimension of the protein language model and ReLU activation. All models are trained with an Adam optimizer with batch size 256 and learning rate 10^{-3}. The TAPE model used was the transformer, the ESM-1v model used was version 1, the largest ESM-2 model used was the 3 billion parameter version (due to memory issues with the larger model), the ProtT5 model was the suggested ProtT5-XL-UniRef50 model, and the CARP model was the largest 640 million parameter version. For Figure 3e, we ran smaller versions of CARP and the ESM-2 models, as specified by their parameter number, with the same architectures as described above (Yang et al., 2022; Lin et al., 2022).

The models labeled TAPE (CNN), ESM-1b (CNN), ESM-1v (CNN), ESM-2 (CNN), ProtT5 (CNN), and CARP (CNN) use residue-level embeddings that are fed into a CNN, similar to the baseline CNN described above. The intermediate dimension output by the embedding neural network is set to be twice the dimension of the output of the protein language model. All other parameters remained the same.

The models labeled ESM-1v (Augmented), ESM-2 (Augmented), CARP (Augmented), and EVMutation (Augmented) used the zero-shot variant effect prediction from the model in question, combined with a one-hot encoding of the entire sequence that was fed into a linear layer (Hopf et al., 2017; Lin et al., 2022; Meier et al., 2021; Yang et al., 2022). These were trained with the same parameters as the baseline linear model. The ESM models used the masked marginal approach proposed in Meier et al. (2021) to compute zero-shot variant effect prediction. EVMutation used the default parameters (Hopf et al., 2017).

Fine-tuned models are trained as described above, but we also fine-tuned the underlying ESM model that generated the embeddings with a learning rate of 10^{-6}. Due to compute and cost limitations, fine-tuning was only done for the ESM models with 8 and 35 million parameters.

We observed high run-to-run variance in the performance of many models. As a result, all models were run in triplicate and we have reported both the mean and standard deviation of model performance.

B SUPPLEMENTARY RESULTS

B.1 FLIGHTED-DHARMA MODEL PERFORMANCE

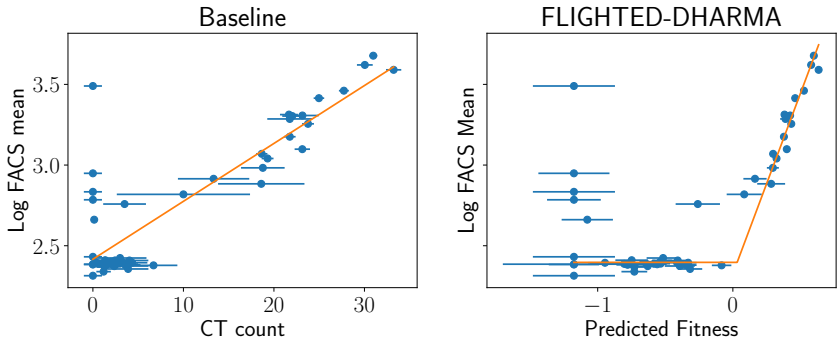

Figure S1: Accuracy on Validation set for (left) Baseline and (right) FLIGHTED-DHARMA. This shows the fit between the fitnesses predicted by each model and the log FACS mean, as done by either a piecewise linear function or a linear function.

To better contextualize the fitness-to-FACS fit, we show the fits on the validation set for both FLIGHTED-DHARMA and the baseline model in Figure S1. Generally, FLIGHTED-DHARMA's predicted fitnesses were fit using a piecewise linear function, while depending on the timepoint, the baseline model used either a linear or piecewise linear function.

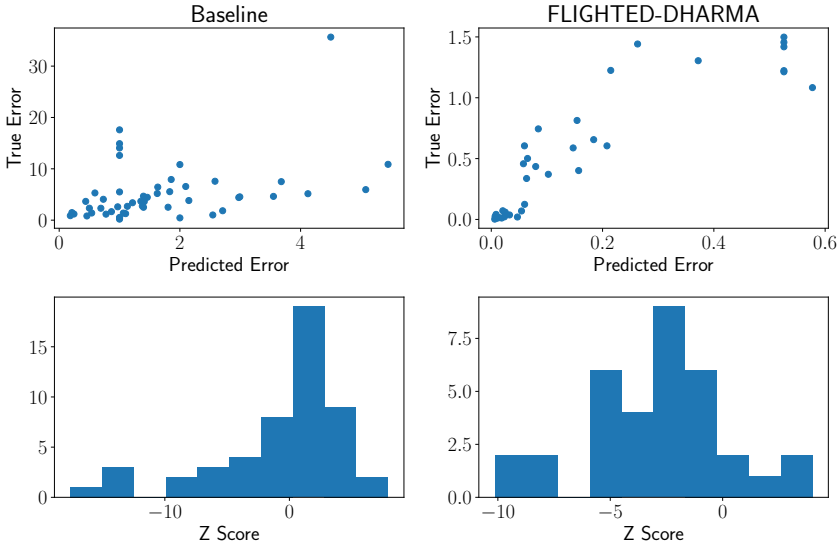

Figure S2: Calibration of FLIGHTED-DHARMA as Compared to Baseline. The baseline model's true errors are not related to the predicted error, while FLIGHTED-DHARMA has higher true error when predicted error is higher. As such, FLIGHTED-DHARMA is considerably better calibrated with many fewer z-score outliers.

We next examine the calibration of FLIGHTED-DHARMA as compared to the C→T baseline model. First, we directly examine calibration on the test data, using all reads available for each datapoint in the test set. In Figure S3, we see the true and predicted errors from FLIGHTED-DHARMA and the baseline model on the test set data, as well as the histogram of z scores for both models. The baseline model predicted errors are largely unrelated to true error, while the FLIGHTED-DHARMA model's predicted errors increase as true error increases. The log likelihood

of the baseline model on this dataset was -19.8 while the log likelihood of FLIGHTED-DHARMA was -10.0, so FLIGHTED-DHARMA shows considerable improvement.

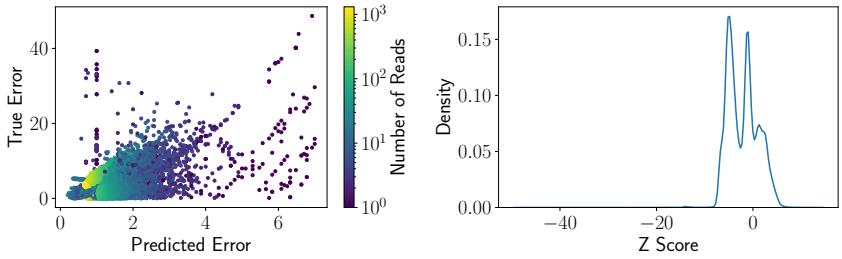

Figure S3: Calibration of Baseline C→T Model. The baseline model's calibration is considerably worse compared to FLIGHTED-DHARMA's in Figure 2d.

This is a relatively small dataset since we did not subsample subsets of reads as we did in Figure 2d. The subsampling process gives us a better understanding of how FLIGHTED-DHARMA behaves with very small subsets of reads. In Figure S3, we have the calibration of the baseline model as compared with that of FLIGHTED-DHARMA in Figure 2d. The calibration is considerably worse, with the z-score distribution looking very non-normal and a much lower log likelihood of -8.00 as compared to FLIGHTED-DHARMA's log likelihood at -3.93.

B.2 The TEV Dataset

Some histograms of basic properties of the TEV dataset may be seen in Figures S4, S5, and S6. Note that the distribution of read counts is very skewed, with some variants being read a very small number of times. We believe this is likely due to toxicity of the uncleaved T7 lysozyme present in the cell. This substantially changes the measured variance in Figure S6; fortunately, FLIGHTED is designed to handle this issue.

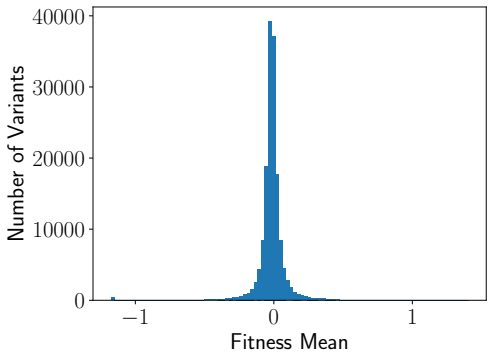

Figure S4: Distribution of fitnesses in the TEV dataset.

In Figures S7 and S8, we show the distribution of fitnesses in the training and test sets, respectively, for each split.

B.3 Benchmarking on the TEV Dataset

The complete performance of all models may be seen in Figure S9. Bolded models performed the best on a given task. We also ran a t-test comparing every model to the best model, and highlighted models were not significantly different from the best model with a p-value < 0.05, i.e. we cannot conclude that the best performing model on a given task outperformed any of the highlighted models.

We now examine model predictions for a single high-performing model (the ESM-1b CNN on three-vs-rest) in Figure S10; here, predicted fitness refers to the fitness predicted by the ESM-1b model,

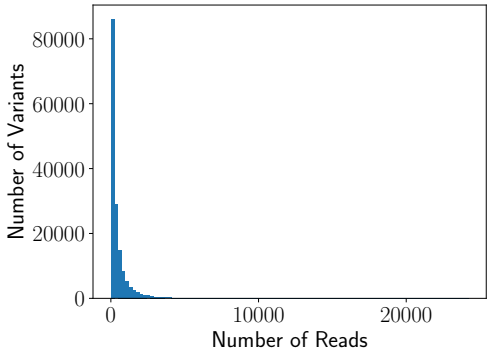

Figure S5: Distribution of read counts in the TEV dataset.

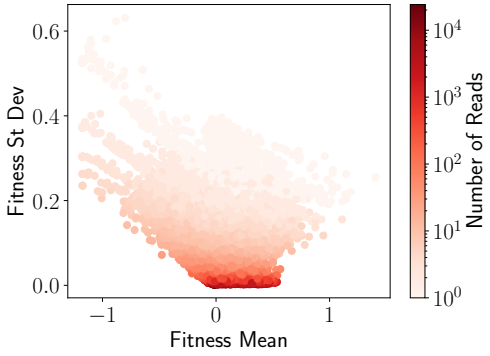

Figure S6: Fitness mean, variances, and read count in the TEV dataset.

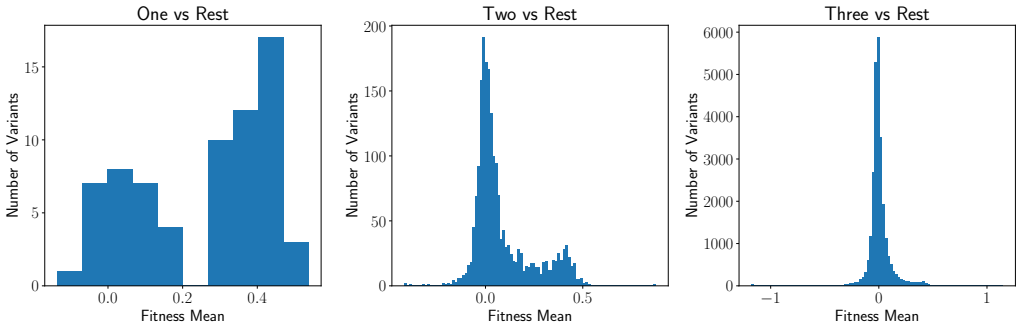

Figure S7: Distribution of fitnesses in the training sets.

and true fitness refers to the FLIGHTED-measured fitness with the error (variance) shown on the colorbar. The model has reasonably high accuracy on predictions with very low variance (< 0.01) and predicts most other data points with high variance to be inactive, suggesting the model has learned to filter by variance when making predictions. Figure S6 suggests that the primary determinant of variance is read count, so this implies the model has learned read count on the test set, potentially surprising.

We investigated this result further to ensure that there was no accidental data leakage at any point. There are two effects we identified that allow the model to learn read count on the test set. First, the T7 lysozyme is slightly toxic to the cell and when not cleaved (i.e. when the protease is inactive),

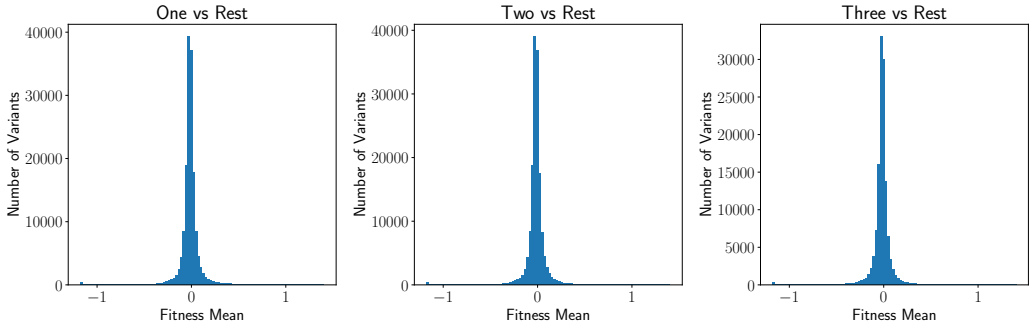

Figure S8: Distribution of fitnesses in the test sets.

results in a lower read count. In Figure S11, we compare the read count in libraries with and without the lysozyme and see that active variants have higher read count in the library with the lysozyme. Since the models are trained to learn fitness, they can also implicitly learn about this effect.

Second, the initial (no-lysozyme) library construction was generated by a potentially biased synthesis process, so the initial read count is learnable, i.e. similar variants have similar read counts. We tested this by training a model (the same ESM-1b CNN) on the initial read counts of a training set consisting of triple mutants and predicting read counts on the quadruple mutants, as shown in Figure S12. Since read count is learnable, we conclude that models are learning a combination of the initial read count from the biased library and TEV protease fitness when making predictions. This dataset and these models are still generally useful for predicting TEV protease activity and benchmarking ML models, but any active variant that had low read count in the initial library would likely be a false negative on any model trained on this dataset. To confirm that the models did learn TEV protease activity, we verified that the models predicted low activity for variants with high read count and low activity, as expected. In the future, we aim to reduce library bias to generate higher-quality TEV protease datasets and more accurate models to mitigate this issue.

Task Model	One vs Rest	Two vs Rest	Three vs Rest
Linear	526.0933 ± 22.7232	473.0123 ± 30.1776	61.9535 ± 9.9049
TAPE	297.5274 ± 16.9196	42.1769 ± 5.7037	10.1017 ± 2.9131
ESM-1b	165.9262 ± 0.8583	19.2814 ± 2.1384	4.7189 ± 0.0789
ESM-1v	**96.5753 ± 4.8990**	31.6219 ± 8.8151	5.0249 ± 0.2268
ESM-2 (8M)	487.1830 ± 67.0827	77.3104 ± 13.6258	26.0598 ± 11.2005
ESM-2 (35M)	201.0534 ± 14.7650	38.3270 ± 14.7239	5.3898 ± 0.0861
ESM-2 (150M)	253.0745 ± 17.7263	21.2191 ± 4.1567	5.8834 ± 0.7763
ESM-2 (650M)	129.0170 ± 3.0636	15.7421 ± 2.0325	5.0022 ± 0.5990
ESM-2	455.1378 ± 8.2257	21.5850 ± 7.1289	16.4862 ± 17.6678
ProtT5	128.9400 ± 5.0140	16.2017 ± 2.1542	4.5883 ± 0.2502
CARP (600k)	370.3106 ± 61.6088	67.1536 ± 5.6832	17.8028 ± 0.6229
CARP (38M)	779.2183 ± 6.6225	69.2424 ± 6.2080	11.5826 ± 1.0835
CARP (76M)	293.7587 ± 52.0933	59.6290 ± 2.5587	14.5622 ± 0.9285
CARP	661.4456 ± 227.2850	201.7350 ± 12.4057	25.1076 ± 8.3689
CNN	613.7185 ± 242.4505	169.1024 ± 7.8148	6.4333 ± 0.1398
TAPE (CNN)	535.2333 ± 132.6787	119.1340 ± 10.3364	5.5736 ± 0.2825
ESM-1b (CNN)	565.8093 ± 203.6468	40.1009 ± 9.9084	4.5211 ± 0.0431
ESM-1v (CNN)	678.3326 ± 164.8031	28.9555 ± 0.4254	4.7168 ± 0.1845
ESM-2 (8M, CNN)	537.5033 ± 238.1350	56.3152 ± 9.9305	5.3183 ± 0.1928
ESM-2 (35M, CNN)	299.9104 ± 14.8647	26.8592 ± 5.2674	224.8132 ± 191.9686
ESM-2 (150M, CNN)	676.5742 ± 103.0750	31.0988 ± 5.3971	4.7040 ± 0.1121
ESM-2 (650M, CNN)	915.3362 ± 23.6130	66.0526 ± 3.5709	4.8076 ± 0.0419
ESM-2 (CNN)	1361.5717 ± 539.0819	87.6785 ± 10.3623	4.9448 ± 0.1656
ProtT5 (CNN)	900.8455 ± 233.7611	61.6325 ± 3.8367	4.7374 ± 0.0591
CARP (600k, CNN)	479.3737 ± 9.2137	69.0239 ± 4.6387	6.8296 ± 0.4412
CARP (38M, CNN)	685.8954 ± 281.9056	79.6063 ± 9.2690	4.8927 ± 0.0288
CARP (76M, CNN)	581.1124 ± 86.5494	64.4380 ± 2.4957	4.7728 ± 0.1322
CARP (CNN)	797.2115 ± 459.2779	40.6028 ± 0.7333	4.6357 ± 0.1348
ESM-1v (Augmented)	238.4065 ± 5.2917	480.2571 ± 33.7659	63.6243 ± 3.8785
ESM-2 (8M, Augmented)	337.0354 ± 13.8129	492.2340 ± 28.7706	69.7858 ± 4.0155
ESM-2 (35M, Augmented)	216.1071 ± 23.8610	472.1725 ± 53.4941	71.2082 ± 6.5696
ESM-2 (150M, Augmented)	222.0371 ± 4.0988	490.2639 ± 30.5448	61.4350 ± 7.4693
ESM-2 (650M, Augmented)	338.1440 ± 190.0764	481.7004 ± 19.7602	69.2233 ± 11.1288
ESM-2 (Augmented)	226.2928 ± 33.0988	507.1823 ± 49.9358	69.0547 ± 3.8129
CARP (600k, Augmented)	555.9357 ± 65.4261	469.3298 ± 3.0498	65.6611 ± 1.8785
CARP (38M, Augmented)	608.4704 ± 69.0148	449.0583 ± 18.4007	61.7768 ± 5.5423
CARP (76M, Augmented)	562.6250 ± 15.5729	464.0757 ± 13.7924	64.1665 ± 7.0719
CARP (Augmented)	508.6772 ± 74.7882	459.4699 ± 36.2165	66.3554 ± 4.3394
EVMutation (Augmented)	247.1387 ± 20.9441	492.7133 ± 31.5477	62.2780 ± 6.3277
ESM-2 (8M, Finetuned)	260.1472 ± 66.3833	7.6464 ± 1.6408	4.2765 ± 0.0316
ESM-2 (35M, Finetuned)	273.4673 ± 15.2778	**7.5348 ± 0.3203**	4.2672 ± 0.0318
ESM-2 (8M, CNN, Finetuned)	517.6429 ± 199.1142	19.5290 ± 1.4134	4.7530 ± 0.1151
ESM-2 (35M, CNN, Finetuned)	211.5166 ± 46.2842	10.7844 ± 0.5173	**4.2564 ± 0.0468**

Figure S9: Summary Table of TEV Model Performance. Bolded models are the best-performing on a given dataset and highlighted models are statistically not significant from the best model by a t-test with a p-value < 0.05.

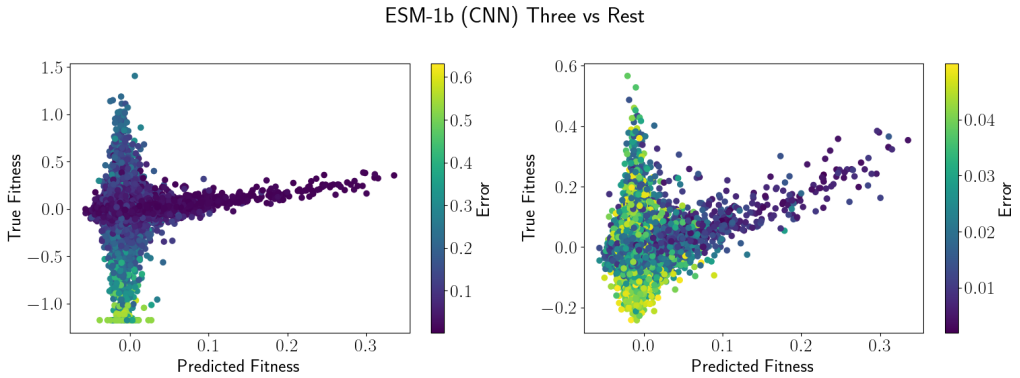

Figure S10: Sample Model Predictions on TEV Dataset, produced by ESM-1b CNN on three-vs-rest. Left includes all data points and right includes data points filtered with error < 0.05 to highlight most accurate model predictions.

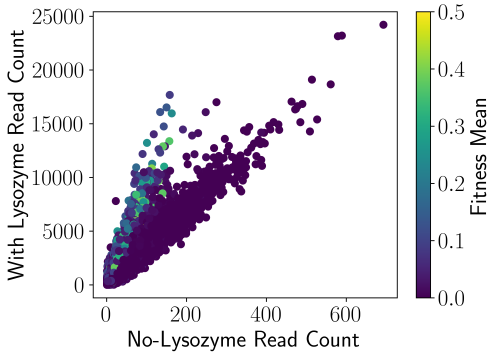

Figure S11: Read Count With and Without T7 Lysozyme. Active variants have higher read counts with the T7 lysozyme.

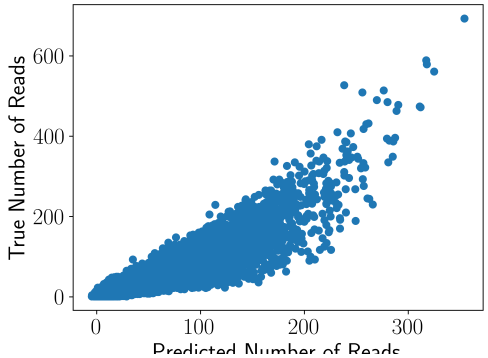

Figure S12: Control Model Trained on No-Lysozyme Read Count, showing that read count is learnable.