PROTMAMBA: A HOMOLOGY-AWARE BUT ALIGNMENT-FREE PROTEIN STATE SPACE MODEL

Anonymous authors
Paper under double-blind review

ABSTRACT

Protein design has important implications for drug discovery, personalized medicine, and biotechnology. Models based on multiple sequence alignments efficiently capture the evolutionary information in homologous protein sequences, but multiple sequence alignment construction is imperfect. We present ProtMamba, a homology-aware but alignment-free protein language model based on the Mamba architecture. In contrast with attention-based models, ProtMamba efficiently handles very long context, comprising hundreds of protein sequences. We train ProtMamba on a large dataset of concatenated homologous sequences, using two GPUs. We combine autoregressive modeling and masked language modeling through a fill-in-the-middle training objective. This makes the model adapted to various protein design applications. We demonstrate ProtMamba's usefulness for the generation of novel sequences and for fitness prediction. ProtMamba reaches competitive performance with other protein language models despite its smaller size, which sheds light on the importance of long-context conditioning.

1 INTRODUCTION

Proteins are essential building blocks of life, serving vital roles in metabolic processes, cellular transport, structural integrity, and immune responses. Composed of long chains of amino acids (polypeptides), proteins fold into specific three-dimensional structures critical for their biological functions. One of the key challenges in biology is protein engineering and design: conceiving protein sequences to exhibit enhanced or novel functions. While experimental approaches like directed evolution and mutational scanning are effective in this regard, they only allow exploring the neighbors of existing sequences. However, the recent growth of extensive databases has opened up new avenues for computational methods that exploit the breadth of biological evolution. For instance, UniProt (The UniProt Consortium, 2021) contains more than two hundreds of millions of protein sequences. Biological functions exert evolutionary constraints on protein sequences, which can be probed by considering families of homologous proteins (i.e. proteins that share an evolutionary history) and analyzing this data through statistical methods and, more recently, through deep learning methods.

Protein language models rely on recurrent (Bepler & Berger, 2019), transformer (Rives et al., 2021) or convolutional (Yang et al., 2024) architectures, and are trained through masked language modeling, autoregressive modeling, or discrete diffusion techniques (Alamdari et al., 2023), on large ensembles of single protein sequences (Khakzad et al., 2023). The representations learned by these models correlate with biochemical properties of proteins (such as function, structure, contacts) (Elnaggar et al., 2021; Vig et al., 2021; Rives et al., 2021; Madani et al., 2023), and can be used to generate protein sequences or to evaluate the fitness of variants. The vast majority of these methods are trained on non-structured ensembles of single protein sequence and do not have direct access to homology, or to conservation and variability within protein families. Models trained on multiple sequence alignments (MSAs) of homologous sequences have also been introduced, despite raising memory challenges and potentially suffering from the imperfections of MSAs (Thompson et al., 2011). Successful MSA-based transformer models, such as MSA Transformer (Rao et al., 2021) or the EvoFormer module of AlphaFold2 (Jumper et al., 2021) alternate attention along protein sequences and across homologs. More recently, PoET (Truong Jr & Bepler, 2024) was trained on concatenations of non-aligned homologous sequences, offering a promising autoregressive alternative to MSA Transformer for protein fitness prediction and design.

State space models such as S4 (Gu et al., 2021), Hyena (Poli et al., 2023) and Mamba (Gu & Dao, 2023b) are catching up with transformers thanks to their ability to efficiently handle very long sequences of tokens. These models were quickly adapted to work with biological data. Approaches such as HyenaDNA (Nguyen et al., 2023) or Evo (Nguyen et al., 2024) were trained on long DNA sequences and capture regulatory mechanics. Meanwhile, PTM-Mamba addresses post-translational modifications of protein sequences (Peng et al., 2024).

In this paper, we present ProtMamba, a novel homology-aware but alignment-free protein language model, trained on concatenated sequences of homologous proteins. Based on the Mamba architecture (Gu & Dao, 2023b), ProtMamba is able to handle extremely long contexts (unlimited lengths during inference). Trained to autoregressively predict the next amino acid, but also with a fill-in-the-middle (FIM) objective, it can be used for multiple different tasks. First, ProtMamba can autoregressively generate novel sequences without contextual information. Second, by providing ProtMamba with sequences from a specific protein family or subfamily as context, users can prompt it to generate sequences tailored to their specifications. This conditional generation approach is a key strength of the model (see also Truong Jr & Bepler (2024)), and could become an alternative to fine-tuning. Third, ProtMamba supports sequence inpainting, i.e., filling specific masked regions with the desired number of amino acids. For this, along with homologous sequences (used as context), the model is provided with a target sequence to be modified. This generation mode opens novel methods of designing specific parts of protein sequences. Furthermore, ProtMamba is useful for fitness prediction tasks. Users can input a sequence with specific masked positions, prompting the model to output the probability distribution of all mutations in each variant with a single forward pass. Across these various tasks, we obtain competitive results with larger protein language models and task-specific methods.

2 METHODS

2.1 KEY TECHNICAL CONTRIBUTIONS

1. To harness the evolutionary information present in homologous sequences without relying on multiple sequence alignments (MSAs), we use as input a concatenation of homologous sequences for each protein family. In each of these long arrays, sequences are separated with a specific token. The motivation is that evolutionary information is extremely useful for protein modeling Jumper et al. (2021); Rao et al. (2021); Abramson et al. (2024), but MSAs can be inaccurate. This approach is similar to that used recently in the autoregressive transformer PoET (Truong Jr & Bepler, 2024).
2. We develop an architecture based on Mamba blocks, an alternative to attention recently proposed by (Gu & Dao, 2023b) that relies on state space models. In Mamba, which is a recurrent neural network, memory scales linearly in sequence length, bypassing the quadratic memory constraints of transformers. This allows handling significantly longer input sequences, in addition to being faster to train and to use at inference. This is a key asset here, as concatenating homologous sequences results in long inputs. Note that Truong Jr & Bepler (2024) employed attention matrix chunking to address this issue, but this results in potential losses of statistical dependence signals, and only partially solves the memory limit.
3. We combine elements of both autoregressive modeling and masked language modeling (MLM), by training our model using the fill-in-the-middle (FIM) objective (Bavarian et al., 2022; Fried et al., 2022). The model learns to predict masked patches extracted randomly from a sequence and positioned at the end of it, and can therefore leverage the full sequence context, while being trained autoregressively. This is of particular interest for biological sequences, because preceding and subsequent tokens can all be informative to predict a new token. While autoregressive models are generative by definition, they yield the probability of each new token conditioned on previous ones (ignoring subsequent ones). Besides, MLM can be productively used for protein sequence generation (Sgarbossa et al., 2023).
4. To promote the model's ability to reason over in-sequence positions, which is particularly useful for the FIM task, we modify the original Mamba implementation by introducing sequence-level positional embeddings. This enables the model to pay attention to relative positions inside each sequence. In inference and generation, it opens the possibility of controlling the number of amino-acids to generate.

2.2 MODEL ARCHITECTURE AND TRAINING STRATEGY

ProtMamba's architecture is adapted from Mamba (Gu & Dao, 2023a). An important modification is that we introduce learned positional embeddings for the input tokens. Among different variants, we observed that the most effective and stable method to integrate positional embeddings is to concatenate them with the input token embeddings into a single vector. Specifically, we allocated half of the embedding dimension d to token information and the other half to positional information.

We trained a 107 million parameters model with 16 layers, embedding dimension $d = 1024$, and hidden state dimension equal to embedding dimension. We started with a maximal total input sequence length of $2^{11} = 2048$ amino acids (recall that input sequences are concatenated homologous protein sequences). The model was trained following (Gu & Dao, 2023a) with some minor modifications. We used the AdamW optimizer with the following parameter values: weight decay $w = 0.1$ and $(\beta_1, \beta_2) = (0.9, 0.95)$. We scheduled the learning rate to increase from zero to 6×10^{-4} with a linear warm-up of 500 steps followed by a constant learning rate. To optimize memory usage, we trained the model using the `bfloat16` format.

To avoid training instabilities observed in (Nguyen et al., 2024), we implemented a callback mechanism to revert to a previous checkpoint if the loss never assumed values below a threshold for 10 successive evaluation steps. The threshold value was chosen as the lowest training loss increased by 0.5%. This ensures that the loss decreases overall, while allowing it to transiently increase. We also prevented gradient explosions by clipping the gradient norm to 1.0.

The model was trained by scheduling the context length of the input using sequence length warm up (SLW) (Nguyen et al., 2023). Initially, we used inputs of length $L = 2^{11}$ tokens with a batch size of 64. We doubled input length each time the loss reached a plateau, simultaneously reducing batch size to maintain a fixed total number of tokens per batch. In case of memory constraints, we decrease the batch size and use gradient accumulation. This heuristic approach is based on the idea that a longer context should provide more information. It is useful because of training instabilities for long contexts (Nguyen et al., 2023; 2024). Note that we did not start training the model with a long context to benefit from a larger batch size, which helps to approximate the loss landscape more efficiently. Finally, once we reached a context length of $L = 2^{17}$, we implemented gradient checkpointing to minimize memory consumption. This allowed us to increase the batch size for the final part of the training and obtain a better approximation of the loss landscape, see (Nguyen et al., 2023; 2024).

Figure 4 reports the loss and perplexity during training, computed on the training set and on a validation set of 192 held-out OpenProteinSet sequence clusters (see Section 2.3). The model was trained on one NVIDIA RTX A6000 for 35 days, and then on two of them for 15 days. This allowed us to keep the batch size large enough when the context size increased. In total, the model was trained on 1.95×10^{11} tokens (approximately 1.5 epochs) and used 2.0×10^{20} FLOPs during training. These numbers show the huge improvements that the Mamba architecture has in terms of training speed with respect to transformers. As a comparison, the smallest ESM3 model (Hayes et al., 2024) was trained with 0.8×10^{11} tokens using 6.72×10^{20} FLOPs, which means that given a fixed amount of compute, ProtMamba can see 8.5 times the tokens seen by ESM3.

We consider two different ProtMamba versions that were obtained by saving checkpoints at different moments of the training. Our model *ProtMamba, Foundation* was trained on a maximum context length of 2^{15} tokens. Our model *ProtMamba Long, Foundation* was trained until the context length reached 2^{17} tokens. Both models were fine-tuned for 2 days on predicting only the FIM amino acids to improve inpainting capabilities, yielding the models *ProtMamba/ProtMamba Long, Fine-tuned*.

2.3 DATASET CONSTRUCTION

We trained ProtMamba on OpenProteinSet (Ahdritz et al., 2024), a dataset which comprises 16 millions MSAs, one for each sequence cluster within Uniclust30 (Mirdita et al., 2017). This dataset was curated to train OpenFold (Ahdritz et al., 2022). We used a filtered subset of the full dataset, consisting of maximally diverse representative MSA clusters, built by iteratively eliminating redundant clusters whose representative sequences appeared in other clusters' MSAs (Ahdritz et al., 2024). This ensures that each representative sequence is only present in its cluster, as detailed in (Ahdritz et al., 2024). This dataset comprises 268,000 clusters including a total of 508 million sequences and 110 billion residues, see Figure 5 for additional statistics. A validation set and a testing set are formed by

holding out respectively 192 and 500 randomly chosen clusters from the training set. Importantly, our use of the filtered version of OpenProteinSet (Ahdritz et al., 2024) ensures that overlap between clusters in the training, validation and test set is strongly minimized. Indeed, this filtering is based on selecting only MSAs of maximal diversity and ensuring that the reference sequences used to build each cluster are not present in any other cluster.

Figure 1 illustrates the construction of a training example. First, a cluster is randomly selected from the filtered OpenProteinSet database described in Section 2.3. As OpenProteinSet uses MSAs, we restore the original unaligned sequences by removing gaps and converting all lowercase insertion residues to uppercase. Each amino acid is tokenized using a unique token. Then, N sequences are sampled uniformly at random and concatenated into a single array, with a **<cls>** token separating each sequence from the next one. The value of N is chosen for the total length of the concatenated sequence to exceed the desired training context length L (e.g. $L = 2^{11}$ at the beginning of training), and the input is then cropped precisely at L. Next, the sequences are prepared for the FIM task. For each sequence, some patches of consecutive tokens are randomly sampled (see below) and masked by replacing them with a mask token **<mask i>**, with one such token representing patch i. For each patch, we append to the sequence another mask token followed by the corresponding masked amino acids (which are unmasked). An **<eos>** token is used to separate the main (masked) sequence from its unmasked patches.

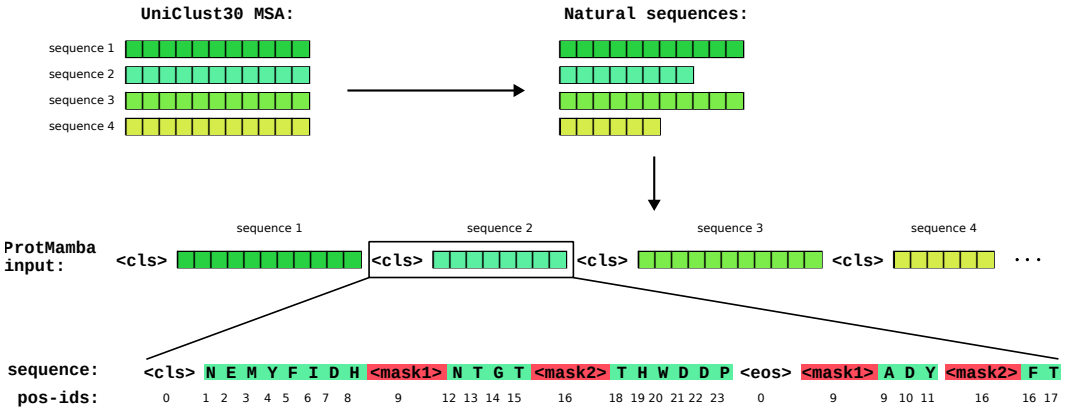

Figure 1: **Input to ProtMamba.** Each element of the input is a concatenation of unaligned homologous sequences separated by **<cls>** tokens. Each sequence starts with a **<cls>** token and ends with an **<eos>** token. Masked segments are replaced by numbered mask tokens, **<mask1>**, ..., **<mask5>**. The masked tokens are appended to the sequence, after the **<eos>** token, each masked segment being preceded by its associated mask token. The position indices ("pos-ids") follow the succession of tokens in the natural sequence. Thus, the masked tokens have their initial position indices in the natural sequence. The position index of each mask is set to that of the first associated masked token. In this particular example we sampled two masks $i = 1, 2$ with length $P_1 = 3$ and $P_2 = 2$.

The following rules are applied when masking each sequence:

1. The number of masked patches in a sequence is sampled from a Poisson distribution with $\lambda = 1$, and capped at 5 (by resampling in case values above 5 are obtained). This yields no mask in 36% of sequences, one mask in 36% of sequences, and more in 28% of sequences.
2. The starting position of each patch is sampled uniformly (without replacement) from all possible positions in the sequence.
3. The length P_i of each patch i is sampled uniformly in $[1, \max(P_i)]$, where $\max(P_i)$ is 0.2 times the distance from the start point of patch i to the start point of patch $i + 1$ (or to the end of the sequence for the last patch). This ensures that no more than 20% of all tokens in each sequence are masked, in line with masking fractions of similar models (Rao et al., 2021; Rives et al., 2021).

Finally, each token is allocated a position index (used to obtain the associated positional embedding) that tracks its position in the original sequence. The position indices of **<cls>** and **<eos>** are set to zero, while the mask tokens **<mask i>** have the same position indices as the first token they are masking, see Figure 1.

3 RESULTS

3.1 PROTMAMBA BENEFITS FROM LONG CONTEXT

To evaluate the effectiveness of incorporating context information in ProtMamba, we examine the scaling of the model's perplexity with context length for natural sequences. Perplexity is commonly used to evaluate autoregressive models and assesses how uncertain they are about a sequence. It is the exponential of the cross entropy loss. Figure 2 shows the scaling of perplexity for the masked parts of the sequences as a function of the number of context sequences, when using the FIM objective. ProtMamba Long (Fine-tuned) achieves remarkably low values of perplexity for small numbers N_m of masked tokens. Furthermore, perplexity decreases when increasing the number of context sequences, revealing the positive impact of richer context on model performance. This decrease tends to be steeper for larger N_m, suggesting that these difficult tasks particularly benefit from richer context. Given the diverse lengths of sequences across protein families, we report perplexity versus the number of sequences in the context rather than versus the total length of the context. Indeed, there can be different amounts of information in contexts of similar lengths but composed of sequences of varying lengths.

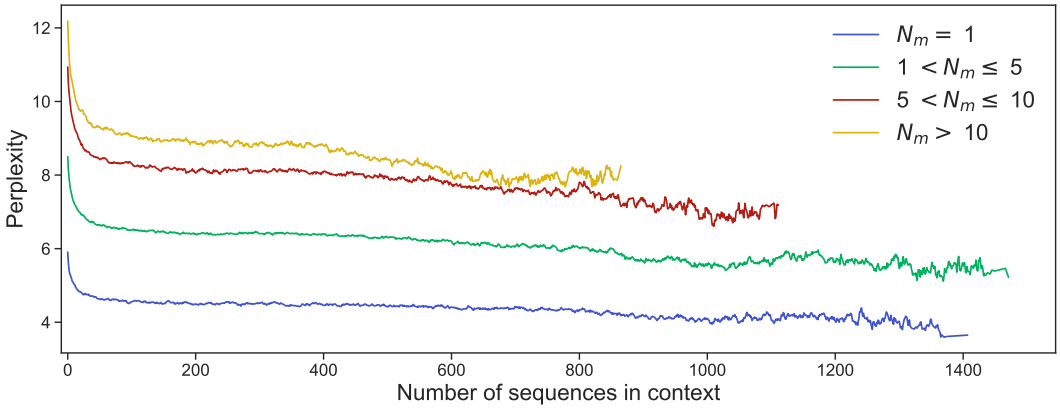

Figure 2: **Scaling of the FIM perplexity with the number of context sequences.** We show the perplexity of the FIM part for different number N_m of masked amino acids versus the number of context sequences. Results are averaged over all 500 clusters of the test set and 100 replicates for each cluster (differing by the random sampling of context sequences). Context sizes go up to 2^{17} amino acids. To reduce noise, we take the exponential moving average, and we restrict to cases where the count of samples is at least 100.

Figure 6 shows the scaling of the per-sequence perplexity (i.e. the standard autoregressive perplexity of the full non-masked sequence) computed on the test set using ProtMamba Long (Foundation). Initially, we notice a decrease of perplexity to a minimum of 7.70 as the number of sequences in the context increases, with lower perplexity values for shorter individual sequences, but this reduction plateaus after a certain point. We attribute this behavior to the finite size ($d = 1024$, see Section 2) of the hidden state of the model, which limits its capacity to effectively leverage context information at each step. We hypothesize that a larger model with a higher-dimensional hidden state could increase the amount of information transferred from the context to the next predicted token. For completeness, we also report perplexity versus context length measured in tokens in Figure 7. There, we observe a rise in perplexity when the context sizes reaches $2^{17} = 131,072$ tokens, which is the highest context length seen during training. We expect that further training the model for longer contexts could lead to lower perplexity values, yet ultimately reaching a lower bound due to the limitations imposed by the hidden state dimension and model size.

3.2 PROTMAMBA PREDICTS MUTATIONAL EFFECTS IN DIFFERENT PROTEIN FAMILIES

Next, we evaluate ProtMamba's ability to predict mutational effects, leveraging its inpainting capabilities arising from the FIM training objective. Indeed, by masking specific amino acids in the wild-type sequence of interest, we can predict the fitness of all variants at these sites. Our first step

to evaluate variant fitness is to collect a context of homologs to the wild-type sequence. We use the ColabFold protocol (Mirdita et al., 2022) for this, ensuring that diverse sequences are found in a few minutes. Then, we randomly subsample 200 sequences among those that have between 30% and 98% similarity to the wild type to construct the context, and we sort these sequences by increasing similarity to the wild type, as in Truong Jr & Bepler (2024).

To evaluate the effect of a variant with a single mutated site, we append the wild-type sequence to the context, mask the mutated residue in it, and predict this residue using the FIM method. Let $\mathcal{C}$ denote the union of the context sequences and of the wild-type sequence masked at the mutated position i. We evaluate the effect of mutations at position i by their fitness score $\mathcal{F}$, defined as:

$$\mathcal{F}(i, x_i, \mathcal{C}) = \log p(x_i|\mathcal{C}) - \log p(x_i^{WT}|\mathcal{C}) \,,$$

for all residues x_i different from the wild-type residue x_i^{WT}. Using this method based on the FIM objective allows us to evaluate the effects of all mutations at position i, decreasing 20-fold the number of passes through the model needed to evaluate all mutations with respect to the typical method used with autoregressive protein language models. To predict the fitness effects of variants involving mutations at multiple sites, we add all the single mutation likelihoods. This approximate, but fast method avoids computing the complete likelihood for all variants, thus reducing the number of calls to ProtMamba. It is accurate when the mutations can be considered independent.

We consider the ProteinGym benchmark (Notin et al., 2023), which contains 217 datasets of substitutions in protein sequences (both single and multiple) and allows comparing to state-of-the-art methods. We evaluate our model's performance using different context sizes and using the 4 different ProtMamba versions (see Section 2). In Figure 8, we report the mean Spearman correlation for each of these models using different context lengths. Models fine-tuned on the FIM task clearly outperform foundation models. Furthermore, ProtMamba Long performs better than ProtMamba, confirming the importance of training the model with a long context. In Figure 9(a), we break down the performance of ProtMamba Long for different context lengths and different protein sequences lengths. We observe that variants with long sequences particularly benefit from long contexts, as they allow including more sequences. This interpretation is supported by Figure 9(b), which shows that this dependence on context length is weaker when considering context length in terms of number of sequences. Based on performance on a validation set (see supplementary Section B and Figure 10), we chose to use a context of 200 sequences to predict fitness using ProtMamba Long (fine-tuned).

In Table 1, we compare the performance of ProtMamba to other models, namely: ESM2-150M (Lin et al., 2023), a transformer protein language model trained on single protein sequences that has a similar size as ProtMamba; ESM-IF1 (Hsu et al., 2021), a structure-based model of similar size; MSA Transformer (Rao et al., 2021), an MSA-based transformer protein language model; Tranception models of different sizes (Notin et al., 2023), single sequence models that are both used on their own and ensembled with independent-site models or variational autoencoders to improve their fitness prediction accuracy. The state-of-the-art model on the ProteinGym benchmark is TranceptEVE L, which combines Tranception L with EVE, a variational autoencoder trained on each specific MSA (Notin et al., 2022). Table 1 is divided in two parts. The first part shows only single-sequence models which do not use alignment information and have not been ensembled with other models. The second part shows models that use MSA information, either by incorporating it directly in the input like MSA Transformer or by ensembling with other models (in practice, either an independent-site model based on first order MSA statistics or EVE (Frazer et al., 2021)).

The first part of Table 1 shows that ProtMamba outperforms all single-sequence models of the same size, except ESM-IF1, which uses structural information. Moreover, ProtMamba outperforms even larger models like Tranception L (which has 7 times more parameters), showing the importance of the context.

Since MSA information remains useful in scoring variants, in the second part of Table 1, we show results where explicit use of MSAs was made via retrieval, i.e. ensembling the models with an independent-site model, as in Notin et al. (2023) (denoted by "R"). Using retrieval, ProtMamba obtains the same performance as Tranception L and as MSA Transformer, which also leverages MSA information. Both of these models were trained using at least one order of magnitude more FLOPs than ProtMamba. We observe that for datasets with more than one mutation (last column in Table 1), ProtMamba with retrieval slightly outperforms the overall state-of-the-art model TranceptEVE L and reaches performance close to the structure-based model ESM-IF1. However, averaging

Model	Par.	Spearman correlation by MSA depth				by mutations	
		All depths	Deep	Medium	Shallow	1	2+
ESM-2	150M	0.387	0.497	0.358	0.306	0.367	0.379
ESM-IF1	142M	0.422	0.544	0.431	0.300	0.413	0.471
Tranception S (w/o R)	85M	0.303	0.320	0.295	0.258	0.293	0.262
Tranception L (w/o R)	700M	0.374	0.419	0.371	0.358	0.358	0.390
ProtMamba (w/o R)	107M	0.406	0.465	0.411	0.391	0.376	0.444
MSA Transformer	100M	0.421	0.473	0.435	0.393	0.392	0.435
Tranception S (w/ R)	85M	0.418	0.444	0.415	0.428	0.389	0.409
Tranception L (w/ R)	700M	0.434	0.473	0.438	0.432	0.404	0.463
TranceptEVE L	>700M	0.456	0.492	0.467	0.451	0.426	0.467
ProtMamba (w/ R)	107M	0.432	0.472	0.438	0.448	0.404	0.469

Table 1: **Performance of different models on the ProteinGym benchmark.** We report Spearman correlation values obtained both based on retrieval (w/ R) and non-retrieval (w/o R) methods, and parameter count for each model. We report results divided according to MSA depth and number of mutations in the benchmark dataset. Results for benchmark models were obtained from `https://proteingym.org/`. Note that PoET-205M (Truong Jr & Bepler, 2024) reports an overall Spearman correlation of 0.474 (Truong Jr & Bepler) on ProteinGym, but it is not yet on the ProteinGym website, and no information is given about the training time or resources.

over all datasets, ProtMamba does not reach the same performance as TranceptEVE L. But since ProtMamba performs better than Tranception L, ensembling ProtMamba and EVE predictions might yield comparable performance.

Finally, in Figure 11, we break down these comparisons between models by category of experiment (panel (a)), taxonomic category (panel (b)) and sequence length (panel (c)). We also show scores for different models on randomly selected example experimental datasets in Figure 12.

3.3 ProtMamba accurately predicts the activity of chorismate mutase enzyme variants

Next, we evaluate ProtMamba, and in particular the power of the FIM objective, on a dataset of experimentally tested natural and *in silico* generated sequences from the chorismate mutase family from Russ et al. (2020). Chorismate mutase functions as an enzyme involved in the catalysis of synthesis of amino acids, and is a domain of the bifunctional chorismate mutase/prephenate dehydratase. We use ProtMamba to evaluate the activity of experimentally studied variants of this enzyme. For this, we sample 100 sequences, either randomly among the natural sequences that were experimentally studied, or randomly among the subset of those that were experimentally shown to be active in *E. coli*. For these two types of context, we test three different protocols to predict the activity of the other variants in the dataset of Russ et al. (2020) with ProtMamba. First, we use only the chorismate mutase domains (cropped sequences) as context, and autoregressively evaluate the likelihood of the full sequence ("from left to right"). Second, we use the full sequences (chorismate mutase/prephenate dehydratase) as context and we evaluate the perplexity of the full sequence autoregressively from left to right. Third, we use the full sequences (chorismate mutase/prephenate dehydratase) as context and evaluate the perplexity of the chorismate mutase domain using the FIM objective. In Figure 13, we report the ROC curve for the two different context types and the three different protocols. We observe that focusing on active variants in the context consistently improves the discrimination power of ProtMamba. We further observe that the quality of our activity predictions increases with context quality. For both context types, using full sequences improves the prediction over using only domains, and using FIM improves accuracy. Moreover, using FIM reduces the computation time per variant compared to autoregressively scoring the full sequence (from 1.1 second per variant to 3.4 seconds per variant, in line with the domain corresponding to a third of the protein total length).

In Figure 14 **a**, we show the impact of context size on the performance of our best activity predictor (using only active variants in context, full domains and FIM). The accuracy of this predictor increases with context length and plateaus at the longest context length seen in training (i.e. 2^{17} residues). In Figure 14 **b** and **c**, we compare the perplexity of the variants when using only active variants as context and when using both inactive and active variants as context. We observe that the perplexity of inactive variants is often higher when using a context of active variants, which shows a better ability to predict inactivity in this case. Furthermore, the perplexity of active variants is often lower in this case, also showing better ability to predict activity with high-quality context. Finally, we display the distribution of the perplexity for ProtMamba using FIM and a context composed of active variants, compared to the experimental activity in Figure 15, and the same perplexity versus the model score from Russ et al. (2020) in Figure 16.

3.4 PROTMAMBA AUTOREGRESSIVELY GENERATES PROMISING NOVEL SEQUENCES

Finally, we evaluate ProtMamba on the autoregressive generation of novel protein sequences given a context of known homologs, corresponding to members of a given cluster of sequences. We generate sequences from 19 randomly selected clusters in the test set, varying the following parameters: temperature (T), top-k number, and top-p fraction, following the approach proposed by (Ferruz et al., 2022). These parameters are commonly employed to control the output of autoregressive models. At each step, top-k limits their output to the top-k most probable tokens, while top-p only includes the top tokens reaching a cumulative probability p. Meanwhile, temperature T adjusts the randomness of sampling. Additionally, we vary the number of sequences in the context to assess the impact of different levels of conditioning on the generated sequences. Specifically, for each cluster, we perform generation using context lengths of $n = 10,\ 100,\ 500,\ 1000$ and N sequences, where N is the total number of sequences in the cluster. For each value of n, we consider the following $(T, \text{top-}k, \text{top-}p)$ triplets: $(0.8, 10, 0.9)$, $(0.9, 10, 0.95)$, $(1, 10, 0.95)$, $(1, 10, 1)$, $(1, 15, 1)$. We generate 100 sequences for each $(n, T, \text{top-}k, \text{top-}p)$, obtaining a total of 2500 sequences per family. As expected, we observed that the parameters which promote higher sampling variability tend to yield sequences with higher perplexity. Note that sequences with more than 750 amino acids, i.e. longer than the longest natural sequence considered here, were discarded from further analysis. They represented $\sim 5\%$ of the generated sequences.

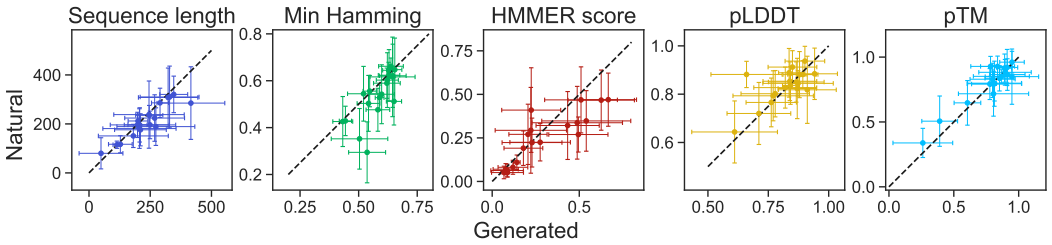

Figure 3: **Comparison of low-perplexity generated sequences with natural ones.** We report the median and the standard deviation of sequence length, Hamming distance to the closest natural neighbor in the sequence cluster from which the context is drawn ("Min Hamming"), HMMER score (rescaled), pLDDT and pTM scores from ESMFold. For each of 19 test clusters, we compare the 100 sequences with lowest perplexity values out of 2500 generated sequences (x-axis) with a randomly chosen subset of 100 natural sequences in the sequence cluster (y-axis). Dashed black lines: $y = x$.

We compare the sampled sequences (aggregated across all parameter sets mentioned above) with natural sequences from the cluster used as context for generation using various scores evaluating novelty, homology, and structure.

1. Novelty is assessed by computing the pairwise Hamming distance (using pairwise Smith-Waterman alignment) with each natural sequence in the cluster, after which it is possible to focus on distance to the closest natural neighbor if desired.
2. Homology evaluation involves training an HMM (using HMMER (Eddy, 2020)) on the cluster's MSA, obtained from OpenProteinSet, and computing the scores it gives to generated sequences.

3. Structure is assessed by predicting the structure of each sampled sequence using ESMFold (Lin et al., 2023). As ESMFold is a single sequence model, it provides predictions that are less biased by MSAs than those of MSA-based models. Futhermore, it is faster than AlphaFold2. ESMFold's confidence measures, both global with pTM scores and local with pLDDT scores, allow for a precise comparison of different sequences sampled from the same cluster.

Figure 17 shows that ProtMamba's estimated sequence perplexity correlates well with HMMER scores, Hamming distance to the closest natural neighbor in the cluster and structural scores. Thus, ProtMamba assigns lower perplexity values to sequences that are more likely to be part of the cluster. The absolute Pearson correlation value averaged over all clusters and scores is above 0.57. Detailed results for each family and each score are presented in Figure 18. Figure 3 shows that the median scores of our generated sequences that have low perplexity are comparable to those of natural ones. Overall, these results are promising for protein design applications.

4 Discussion

Here, we presented ProtMamba, a homology-aware but alignment-free generative protein language model. ProtMamba leverages the long-context capabilities of state space models, allowing it to handle concatenated sequences of homologous proteins. It also benefits from their faster speed compared to attention-based models (Gu & Dao, 2023b), allowing fast sequence generation. ProtMamba was trained using a hybrid strategy combining autoregressive modeling and masked language modeling via the FIM objective. This allows ProtMamba to efficiently predict the next amino acid in a protein sequence as well as to inpaint masked regions.

Our results demonstrate ProtMamba's versatility across multiple tasks, including conditioned generation and protein fitness prediction, both for close and for distant variants. For the latter task, the sequence inpainting abilities of ProtMamba, via the FIM objective, proved to be particularly useful. Overall, ProtMamba benefits from capturing signal across multiple scales. In particular, it is able to predict fitness by exploiting constraints shared broadly across the proteome via its pre-training, but also specific constraints shared between homologs via the context, and it can exploit the full context of a given protein sequence when predicting only part of it.

Limitations. So far, ProtMamba did not reach perplexity values as low as those of larger transformer models like PoET (Truong Jr & Bepler, 2024) for full sequences. However, it can handle longer context sizes and much shorter training time, which is extremely beneficial for the sequence inpainting task. We believe that scaling the model to larger sizes and training times (comparable to PoET) may result in comparable performance while retaining ProtMamba's assets of lower memory cost and inference time.

We did not provide a direct test of the generative ability of ProtMamba for protein sequence inpainting. Indeed, this is a highly specific task lacking clear benchmarks so far. However, we believe that our two analyses on fitness prediction constitute a convincing indirect proof of the usefulness of ProtMamba's inpainting ability. It would be very interesting to experimentally test ProtMamba's inpainting ability, as well as its de novo sequence generation ability (Verkuil et al., 2022).

Perspectives. Our results demonstrate ProtMamba's flexibility, as it allows for precise conditioning by carefully choosing the context information (e.g. restricting to active sequences). Thus, ProtMamba responds very well to prompt engineering. We propose that this could become an alternative or complement to fine-tuning of language models. ProtMamba is also naturally designed to take advantage of retrieval augmented generation (RAG) techniques (Lewis et al., 2021), as it allows for using retrieved protein sequences from any external database, to condition the generation process.

Furthermore, we envision the possibility to use the model for homology search, by scoring sequences within specific contexts. This would be very fast, because only one forward pass would be required.

An interesting further extension of ProtMamba would be to make it explicitly structure-aware, e.g. using a structural alphabet (van Kempen et al., 2023), along the lines of SaProt (Su et al., 2023) or ProstT5 (Heinzinger et al., 2023). Another possible extension would be to include Gene Ontology (GO) terms to condition sequence generation (Madani et al., 2023; Nijkamp et al., 2023).

5 Reproducibility Statement

We describe in details all the steps to reproduce our work in Section 2 and we provide all the code in the supplementary material attached to the submission.

References

J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C. C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. ė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. R. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. dek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, and J. M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, May 2024.

Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Qinghui Xia, William Gerecke, Timothy J O'Donnell, Daniel Berenberg, Ian Fisk, Niccolò Zanichelli, Bo Zhang, et al. OpenFold: Retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Biorxiv*, pp. 2022–11, 2022.

Gustaf Ahdritz, Nazim Bouatta, Sachin Kadyan, Lukas Jarosch, Dan Berenberg, Ian Fisk, Andrew Watkins, Stephen Ra, Richard Bonneau, and Mohammed AlQuraishi. OpenProteinSet: Training data for structural biology at scale. *Advances in Neural Information Processing Systems*, 36, 2024.

Sarah Alamdari, Nitya Thakkar, Rianne van den Berg, Alex X. Lu, Nicolo Fusi, Ava P. Amini, and Kevin K. Yang. Protein generation with evolutionary diffusion: sequence is all you need. *bioRxiv*, 2023. doi: 10.1101/2023.09.11.556673. URL https://www.biorxiv.org/content/early/2023/09/12/2023.09.11.556673.

Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. Efficient training of language models to fill in the middle. *arXiv*, 2022.

Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=SygLehCqtm.

Sean R. Eddy. HMMER: biosequence analysis using profile hidden Markov models, 2020. URL http://hmmer.org.

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 deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/TPAMI.2021.3095381.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348, 2022.

Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

Daniel Fried, Armen Aghajanyan, Jessy Lin, Sida Wang, Eric Wallace, Freda Shi, Ruiqi Zhong, Wen-tau Yih, Luke Zettlemoyer, and Mike Lewis. Incoder: A generative model for code infilling and synthesis. *arXiv preprint arXiv:2204.05999*, 2022.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv*, (arXiv:2312.00752), 2023a. doi: 10.48550/arXiv.2312.00752. URL http://arxiv.org/abs/2312.00752.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023b.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *bioRxiv*, 2024. doi: 10.1101/2024.07.01.600583. URL https://www.biorxiv.org/content/early/2024/07/02/2024.07.01.600583.

Michael Heinzinger, Konstantin Weissenow, Joaquin Gomez Sanchez, Adrian Henkel, Milot Mirdita, Martin Steinegger, and Burkhard Rost. Bilingual language model for protein sequence and structure. *bioRxiv*, 2023. doi: 10.1101/2023.07.23.550085.

Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Combining evolutionary and assay-labelled data for protein fitness prediction. *bioRxiv*, pp. 2021.03.28.437402, 2021.

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*, 596(7873):583–589, 2021. doi: 10.1038/s41586-021-03819-2.

Hamed Khakzad, Ilia Igashov, Arne Schneuing, Casper Goverde, Michael Bronstein, and Bruno Correia. A new age in protein design empowered by deep learning. *Cell Systems*, 14(11): 925–939, 2023. ISSN 2405-4712. doi: 10.1016/j.cels.2023.10.006. URL https://www.sciencedirect.com/science/article/pii/S2405471223002983.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574.

A. Madani, B. Krause, E. R. Greene, S. Subramanian, B. P. Mohr, J. M. Holton, J. L. Olmos, C. Xiong, Z. Z. Sun, R. Socher, J. S. Fraser, and N. Naik. Large language models generate functional protein sequences across diverse families. *Nat. Biotechnol.*, 41(8):1099–1106, 2023.

Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J. Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Research*, 45(D1):D170–D176, 2017. ISSN 0305-1048. doi: 10.1093/nar/gkw1081.

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. ColabFold: making protein folding accessible to all. *Nat Methods*, 19(6):679–682, 2022. doi: 10.1038/s41592-022-01488-1.

Eric Nguyen, Michael Poli, Marjan Faizi, Armin W Thomas, Michael Wornow, Callum Birch-Sykes, Stefano Massaroli, Aman Patel, Clayton M. Rabideau, Yoshua Bengio, Stefano Ermon, Christopher Re, and Stephen Baccus. HyenaDNA: Long-range genomic sequence modeling at single nucleotide resolution. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=ubzNoJjOKj.

Eric Nguyen, Michael Poli, Matthew G. Durrant, Armin W. Thomas, Brian Kang, Jeremy Sullivan, Madelena Y. Ng, Ashley Lewis, Aman Patel, Aaron Lou, Stefano Ermon, Stephen A. Baccus, Tina Hernandez-Boussard, Christopher Ré, Patrick D. Hsu, and Brian L. Hie. Sequence modeling and design from molecular to genome scale with Evo. *bioRxiv*, 2024. doi: 10.1101/2024.02.27.582234. URL https://www.biorxiv.org/content/10.1101/2024.02.27.582234v1.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978, 2023.

Pascal Notin, Lood Van Niekerk, Aaron W Kollasch, Daniel Ritter, Yarin Gal, and Debora S. Marks. TranceptEVE: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, 2022. doi: 10.1101/2022.12.07.519495. URL https://www.biorxiv.org/content/early/2022/12/27/2022.12.07.519495.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 64331–64379. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/cac723e5ff29f65e3fcbb0739ae91bee-Paper-Datasets_and_Benchmarks.pdf.

Zhangzhi Peng, Benjamin Schussheim, and Pranam Chatterjee. PTM-Mamba: A PTM-aware protein language model with bidirectional gated Mamba blocks. *bioRxiv*, 2024. doi: 10.1101/2024.02.28.581983. URL https://www.biorxiv.org/content/early/2024/02/29/2024.02.28.581983.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. MSA Transformer. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 8844–8856. PMLR, 2021. URL https://proceedings.mlr.press/v139/rao21a.html.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl. Acad. Sci. U.S.A.*, 118(15), 2021. ISSN 0027-8424. doi: 10.1073/pnas.2016239118.

William P. Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, and Rama Ranganathan. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, 2020. ISSN 10959203. doi: 10.1126/science.aba3304.

D. Sgarbossa, U. Lupo, and A.-F. Bitbol. Generative power of a protein language model trained on multiple sequence alignments. *Elife*, 12:e79854, 2023. doi: 10.7554/eLife.79854.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. SaProt: Protein language modeling with structure-aware vocabulary. *bioRxiv*, 2023. doi: 10.1101/2023.10.01.560349.

The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100.

J. D. Thompson, B. Linard, O. Lecompte, and O. Poch. A comprehensive benchmark study of multiple sequence alignment methods: current challenges and future perspectives. *PLoS One*, 6(3): e18093, Mar 2011.

Timothy Truong Jr and Tristan Bepler. PoET: A high-performing protein language model for zero-shot prediction. https://www.openprotein.ai/poet-a-high-performing-protein-language-model-for-zero-shot-prediction. Accessed: 2024-05-21.

Timothy Truong Jr and Tristan Bepler. PoET: A generative model of protein families as sequences-of-sequences. *Advances in Neural Information Processing Systems*, 36, 2024.

M. van Kempen, S. S. Kim, C. Tumescheit, M. Mirdita, J. Lee, C. L. M. Gilchrist, J. Söding, and M. Steinegger. Fast and accurate protein structure search with Foldseek. *Nat. Biotechnol.*, 2023.

Robert Verkuil, Ori Kabeli, Yilun Du, Basile I. M. Wicky, Lukas F. Milles, Justas Dauparas, David Baker, Sergey Ovchinnikov, Tom Sercu, and Alexander Rives. Language models generalize beyond natural proteins. *bioRxiv*, 2022. doi: 10.1101/2022.12.21.521521. URL https://www.biorxiv.org/content/early/2022/12/22/2022.12.21.521521.

Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Rajani. BERTology meets biology: Interpreting attention in protein language models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=YWtLZvLmud7.

K. K. Yang, N. Fusi, and A. X. Lu. Convolutions are competitive with transformers for protein sequence pretraining. *Cell Syst*, 15(3):286–294, Mar 2024.

A Appendix

B Supplementary data

ProteinGym assays used in validation

Here, we list the 20 assays we extracted from the ProteinGym benchmark to choose some hyperparameters (see Figure 10).

A0A2Z5U3Z0_9INFA_Wu_2014
AMFR_HUMAN_Tsuboyama_2023_4G3O
CAR11_HUMAN_Meitlis_2020_lof
CBS_HUMAN_Sun_2020
CUE1_YEAST_Tsuboyama_2023_2MYX
DYR_ECOLI_Nguyen_2023
GDIA_HUMAN_Silverstein_2021
HIS7_YEAST_Pokusaeva_2019
HXK4_HUMAN_Gersing_2023_abundance
KCNE1_HUMAN_Muhammad_2023_expr
KKA2_KLEPN_Melnikov_2014
PITX2_HUMAN_Tsuboyama_2023_2L7M
PPM1D_HUMAN_Miller_2022
R1AB_SARS2_Flynn_2022
RDRP_I33A0_Li_2023
S22A1_HUMAN_Yee_2023_abundance
SCN5A_HUMAN_Glazer_2019
SHOC2_HUMAN_Kwon_2022
TRPC_SACS2_Chan_2017
VILI_CHICK_Tsuboyama_2023_1YU5

C Supplementary figures

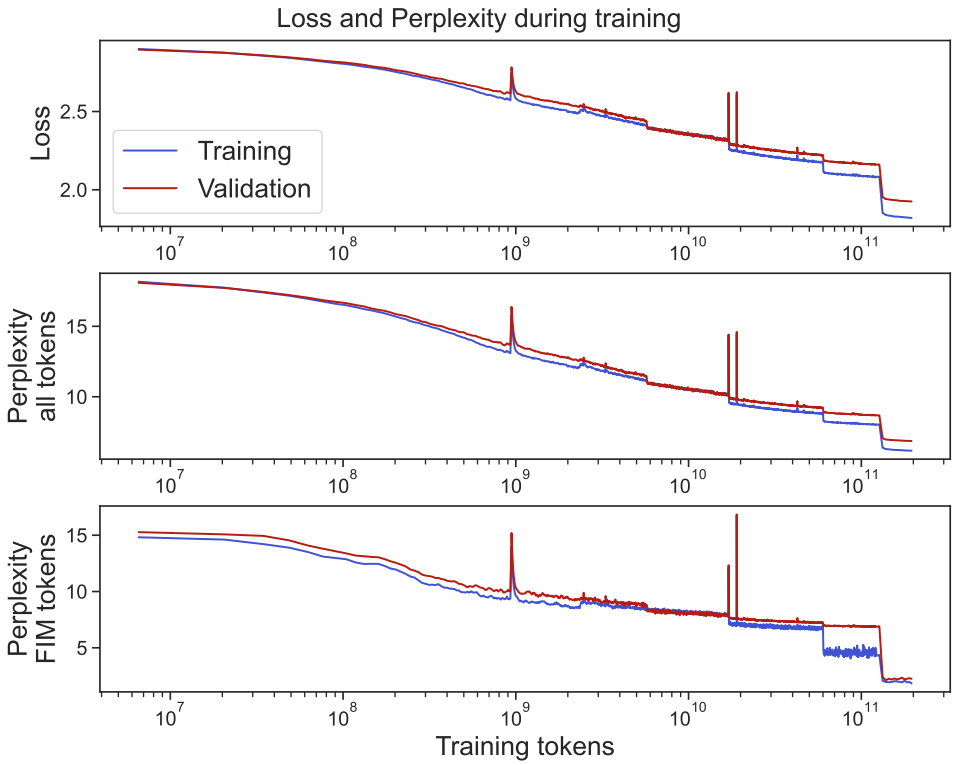

Figure 4: **Loss and perplexity during training.** Cross entropy loss and perplexity computed for both the full non-masked sequences and the FIM tokens. We show them as a function of the number of tokens processed during the training of ProtMamba.

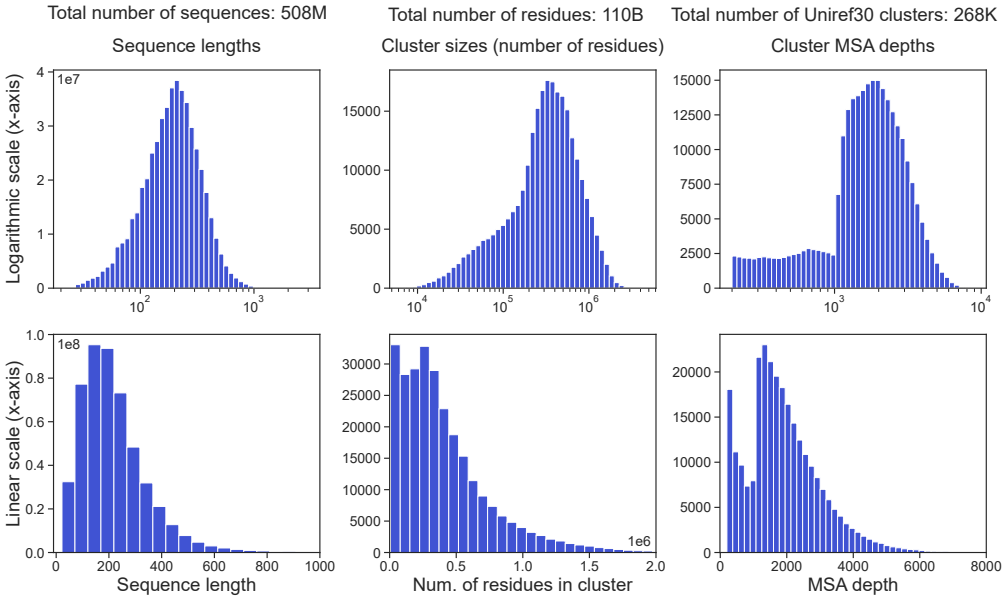

Figure 5: **Datasets statistics.** We show the x axis both in log scale (first row) and in linear scale (second row) to have a better grasp of the distributions.

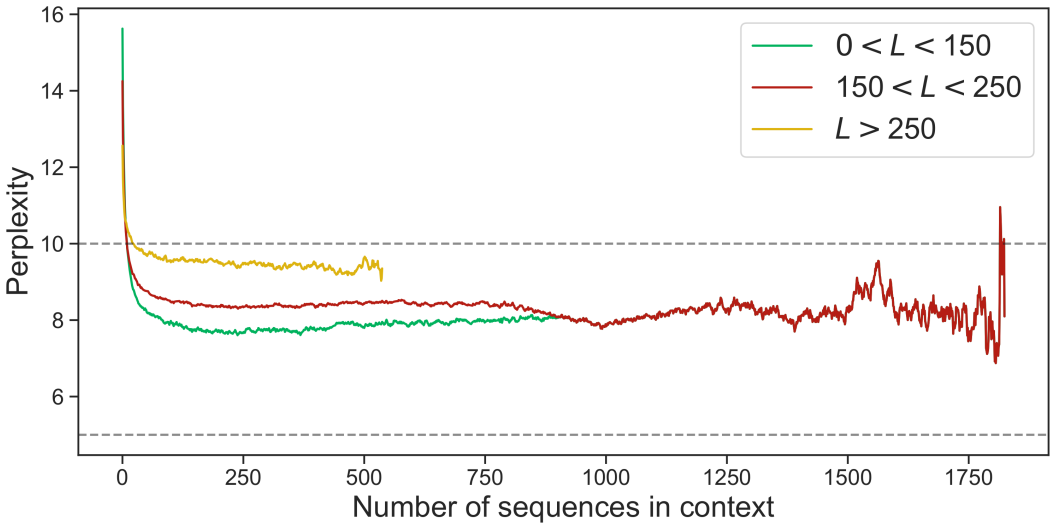

Figure 6: **Loss and perplexity of the full sequences vs. number of sequences in the context.** Scaling of the per-sequence perplexity (i.e. the standard autoregressive perplexity of the full non-masked sequence) versus the number of context sequences. Results are averaged over all 500 clusters of the test set and 20 replicates for each cluster (differing by the random sampling of context sequences). Context sizes go up to 2^{17} amino acids.

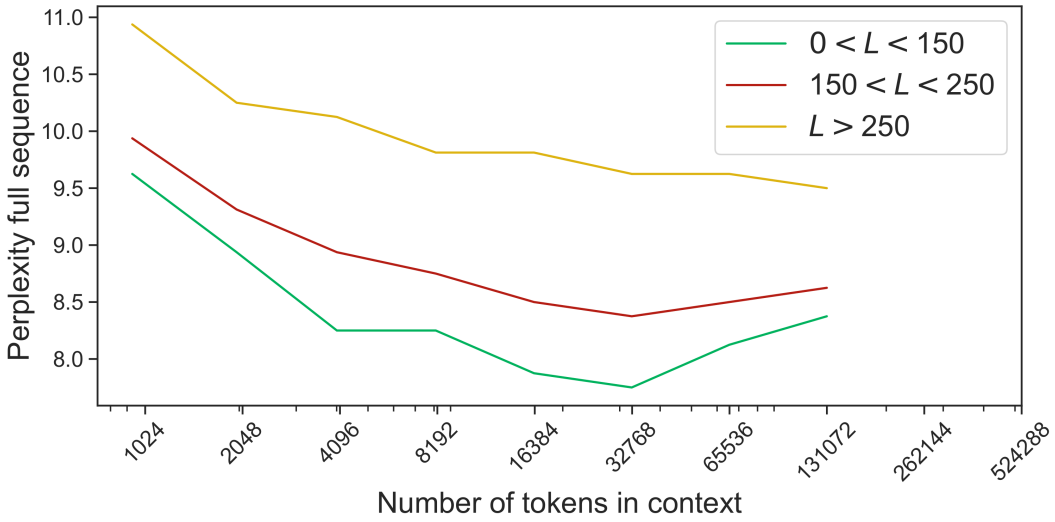

Figure 7: **Loss and perplexity of the vs. number of sequences in the context.** Scaling of the per-sequence perplexity (i.e. the standard autoregressive perplexity of the full non-masked sequence) versus the size of the context (i.e. the number of preceeding tokens). Results are averaged over all 500 clusters of the test set and 20 replicates for each cluster (differing by the random sampling of context sequences). Context sizes go up to 2^{17} amino acids.

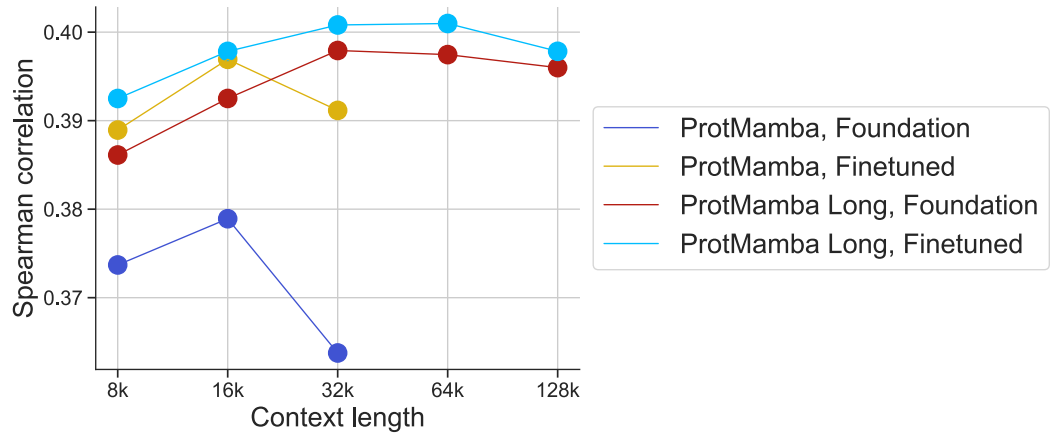

Figure 8: **Comparison of 4 ProtMamba variants on the ProteinGym benchmark.** We show the predictive power for variant effect on the ProteinGym benchmark, via the Spearman correlation between predictions and experimental results, for "ProtMamba, Foundation" ($2^{15} = 32768$ tokens context seen in training), "ProtMamba, Fine-tuned" (fine-tuned on predicting only FIM tokens), ProtMamba Long, Foundation" ($2^{17} = 131072$ tokens context seen in second phase of training) and "ProtMamba Long, Fine-tuned" (fine-tuned on predicting only FIM tokens). We notice that models fine-tuned only on the FIM objective outperform the foundation models. ProtMamba Long is overall performing better than ProtMamba and its performance does not decrease as sharply as ProtMamba for longer context.

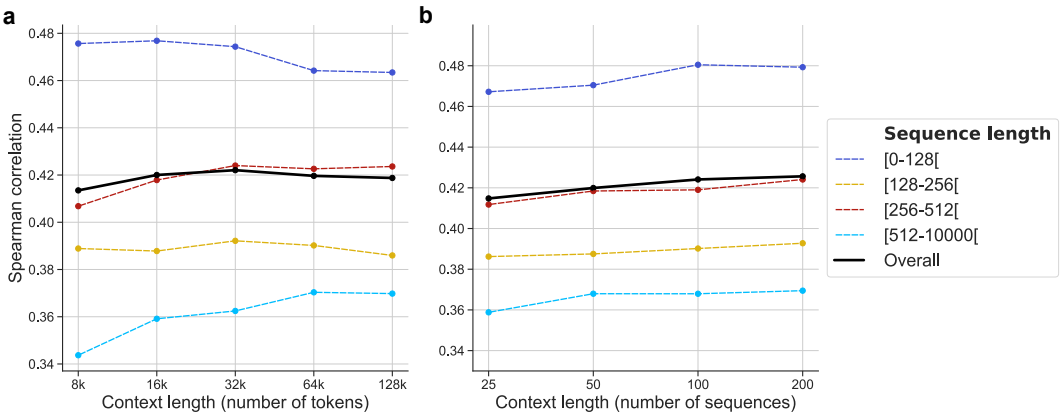

Figure 9: **Impact of context length on results on the ProteinGym benchmark.** **(a)** We run ProtMamba Long on the ProteinGym dataset, building contexts of different sizes in terms of numbers of tokens (from 8,000 to 128,000). We see that the increase in performance is more important for long sequences, which highlights the benefit of long context to model long protein sequences. **(b)** We also run ProtMamba Long on the ProteinGym dataset, building contexts of different sizes in terms of numbers of sequences (from 25 to 200). Overall, we notice a rise in the Spearman correlation, showing that prediction benefits from longer context.

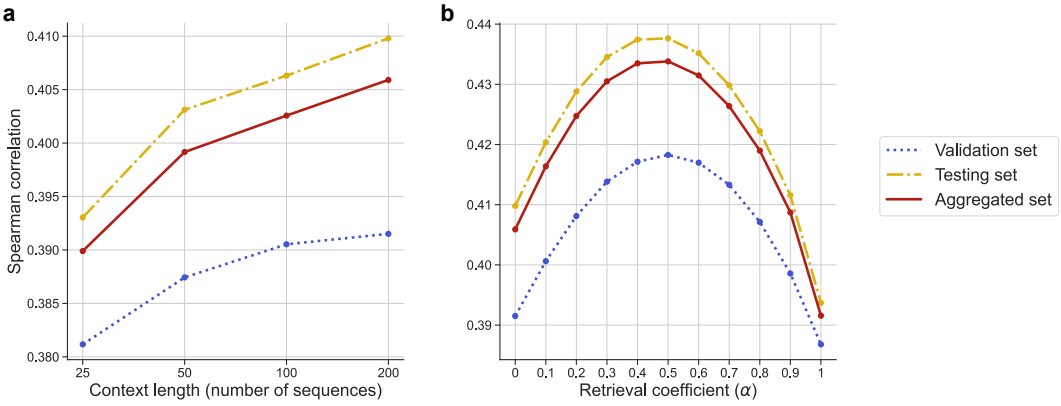

Figure 10: **Choice of context length and retrieval coefficient using a validation set.** We randomly extracted a validation set of 20 datasets (see supplementary Section B) to select the best context length and retrieval coefficient. **(a)** The prediction improves with the context size in the validation set. This trend was later observed in the rest of the benchmark (testing set) too. **(b)** Retrieval requires mixing the fitness score $\mathcal{F}_m$ obtained from ProtMamba and the fitness score obtained from the independent-site model $\mathcal{F}_i$ through the retrieval fitness score $\mathcal{F}_r = \alpha\mathcal{F}_i + (1-\alpha)\mathcal{F}_m$. The best model on the validation set was obtained for a retrieval coefficient $\alpha = 0.5$, which was later verified on the rest of the dataset.

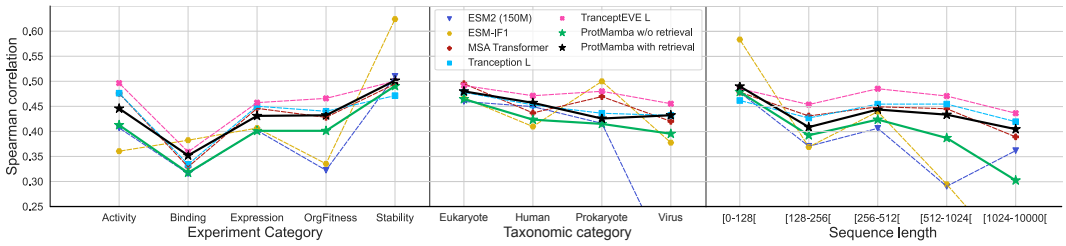

Figure 11: **Breakdown by categories of results on the ProteinGym benchmark.** Results of ProtMamba Long and of existing specialized models on the ProteinGym benchmark, averaged over all datasets, are shown broken down by category of experiments (left), taxonomic category (middle) and wild-type sequence length (right). ProtMamba is fairly competitive with these models. We note that the inverse folding model ESM-IF1 outperforms sequence-based models for stability assessment, as expected (see left panel).

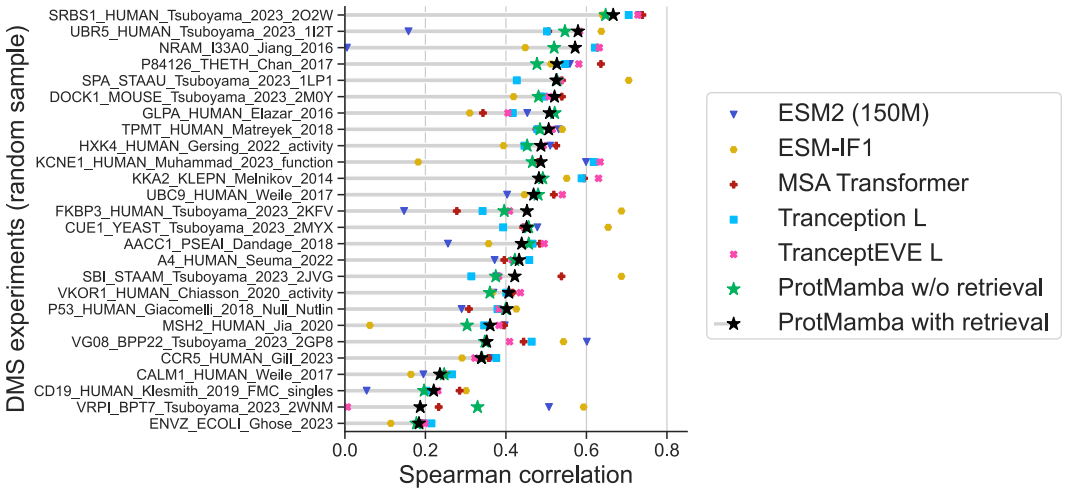

Figure 12: **Example results on the ProteinGym benchmark.** Results of ProtMamba Long are shown on 25 randomly sampled deep mutational scan (DMS) experimental datasets from ProteinGym, and are compared to existing methods (see main text). The score shown is the Spearman correlation between predictions and experimental results.

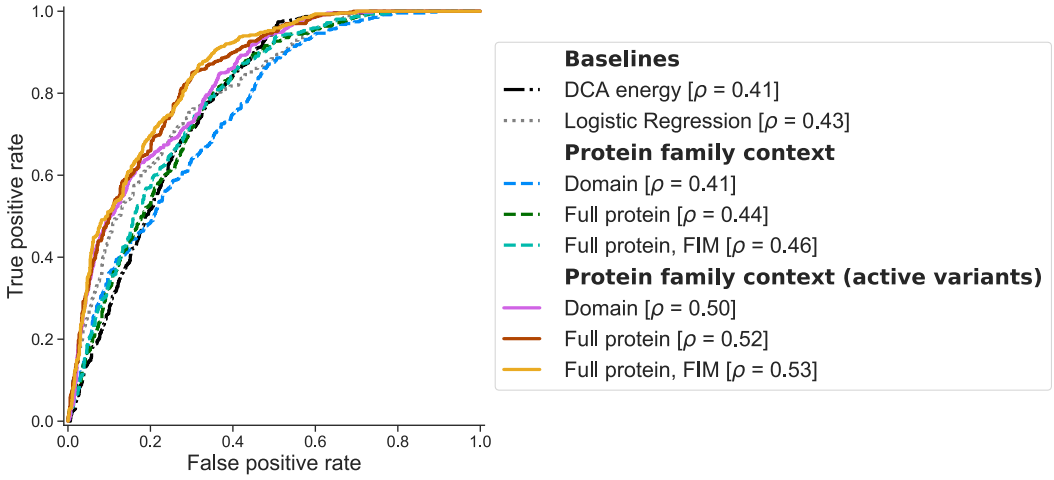

Figure 13: **Impact of various context construction methods on results on chorismate mutase activity.** The ROC curve is shown for various context construction methods (see main text) for predicting active variants in the chorismate mutase dataset, and for baseline methods from Russ et al. (2020). Overall, we observe that restricting to active variants in context helps improving prediction quality (Spearman correlation ρ going from 0.41-0.46 to 0.50-0.53). Giving full proteins instead of restricting to the chorismate mutase domain also improves the results. Using FIM to condition the domain to score using the rest of the protein also improves performance. ProtMamba also outperforms the baselines provided in Russ et al. (2020), namely the Potts or DCA energy and the logistic regression trained directly on amino-acid sequences.

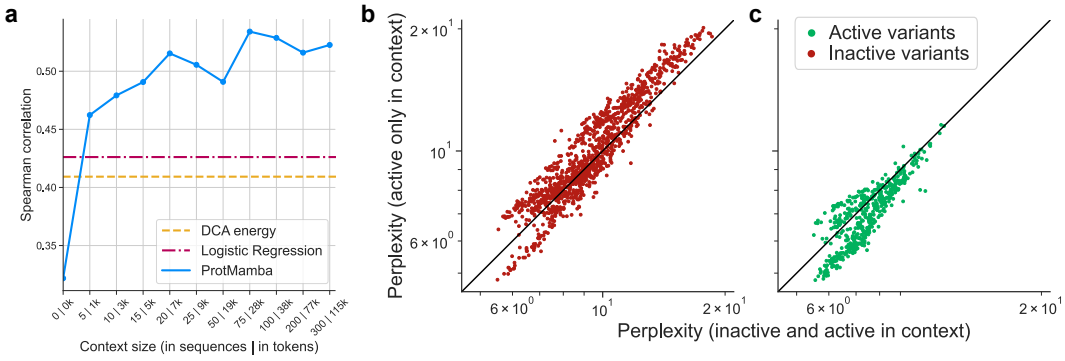

Figure 14: (**a**) **Impact of context length on results on chorismate mutase activity.** Spearman correlation between experimental activity and predictions from ProtMamba is shown using a different number of active sequences in the context (using FIM and full active proteins as context to score variants using ProtMamba). The Spearman correlation quickly increases with the number of proteins sequences given in context, especially from 0 to 25 sequences (or 10,000 tokens) before slowly increasing with context size. (**b**) and (**c**) **Perplexity of generated variants when using only active variants in context (b) or using active and inactive variants in context (c)**. Inactive variants tend to have higher perplexity (implying lower fitness score) when the context contains only active variants (**b**) while active variants have lower perplexity (implying higher fitness score) when the context contains only active variants (**c**) .

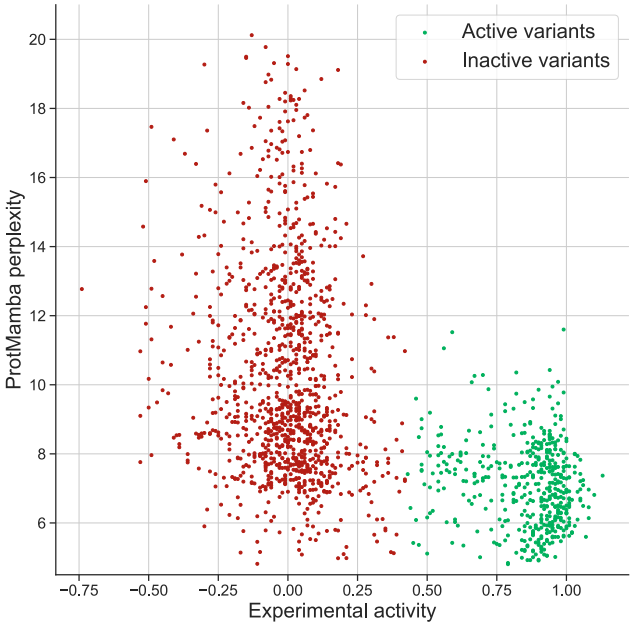

Figure 15: **ProtMamba captures chorismate mutase activity.** Experimental activity of chorismate mutase enzyme variants from Russ et al. (2020) is shown versus ProtMamba per-token perplexity, determined using FIM and full active proteins as context. The per-token perplexity is a good proxy of the activity. We obtain a Spearman correlation of 0.53 between this score and experimental activity, and it yields an AUC of 0.84 to discriminate active from inactive sequences.

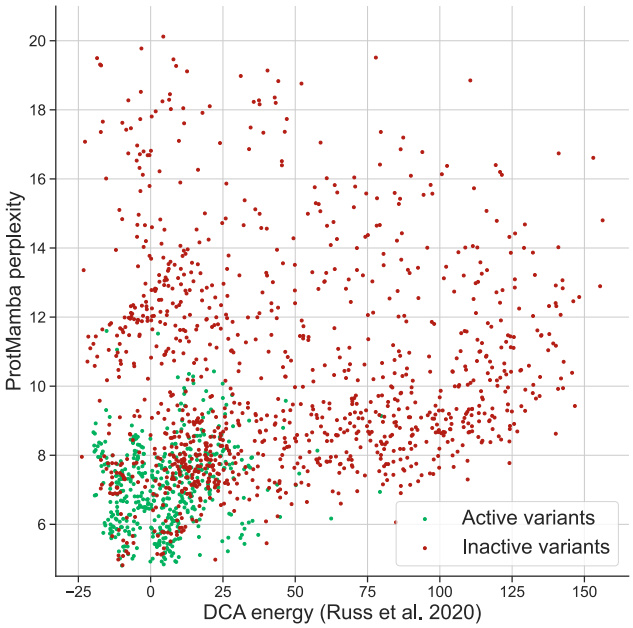

Figure 16: **ProtMamba perplexity versus DCA energy for chorismate mutase variants.** ProtMamba perplexity is evaluated using full sequences, FIM and only active variants in the context, and is shown versus the Potts or DCA energy from Russ et al. (2020). Active variants are in green, while inactive variants are in red. We observe that most of the variants that are active have low perplexity, and that many inactive variants that were not discriminated as inactive by DCA are labelled as such by ProtMamba (bottom right part of the plot).

Cluster	Hamming	HMMER	pLDDT	pTM
A0A2H9MP70	0.45	−0.44	−0.77	−0.54
A0A135YUE9	0.76	−0.68	−0.61	−0.58
G4ZH78	0.23	−0.42	−0.47	−0.47
A0A1A8YWK1	0.79	−0.7	−0.84	−0.81
A0A0A0HZM8	0.46	−0.47	−0.69	−0.63
A0A091TDH7	0.31	−0.32	−0.46	−0.46
A0A2N1P554	0.18	−0.35	−0.37	−0.4
A0A1C5UJ41	0.81	−0.78	−0.81	−0.77
A0A194V424	0.66	−0.71	−0.7	−0.54
S7UZ45	0.56	−0.57	−0.74	−0.66
F2CV06	0.89	−0.85	−0.81	−0.79
A0A146ZGL6	0.72	−0.75	−0.56	−0.64
D8SD16	0.65	−0.74	−0.71	−0.61
A0A139IN77	0.18	−0.2	−0.15	−0.27
A0A1C6Q5J2	0.55	−0.22	−0.45	−0.27
I4B642	0.44	−0.52	−0.61	−0.52
A0A2X4BAY2	0.27	−0.63	−0.63	−0.54
A0A241VGM5	0.44	−0.59	−0.62	−0.55
A0A1S3G530	0.88	−0.72	−0.8	−0.74
Mean	0.54	−0.56	−0.62	−0.57

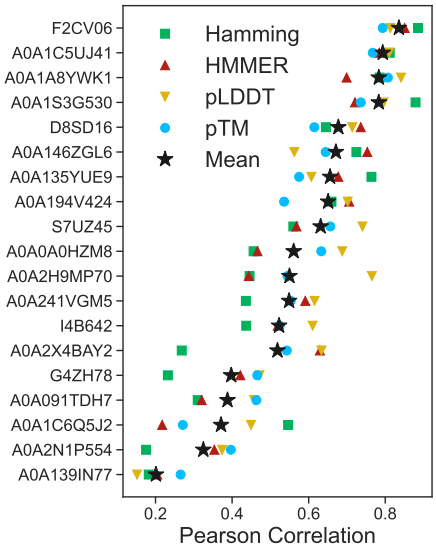

Figure 17: **Pearson correlation between ProtMamba perplexity and scores for generated sequences.** For each of 19 test clusters, we used all the sequences generated by ProtMamba to compute the Pearson correlation between the model perplexity and the Hamming distance to the closest natural neighbor, the HMMER score, the pLDDT and pTM scores from ESMFold.

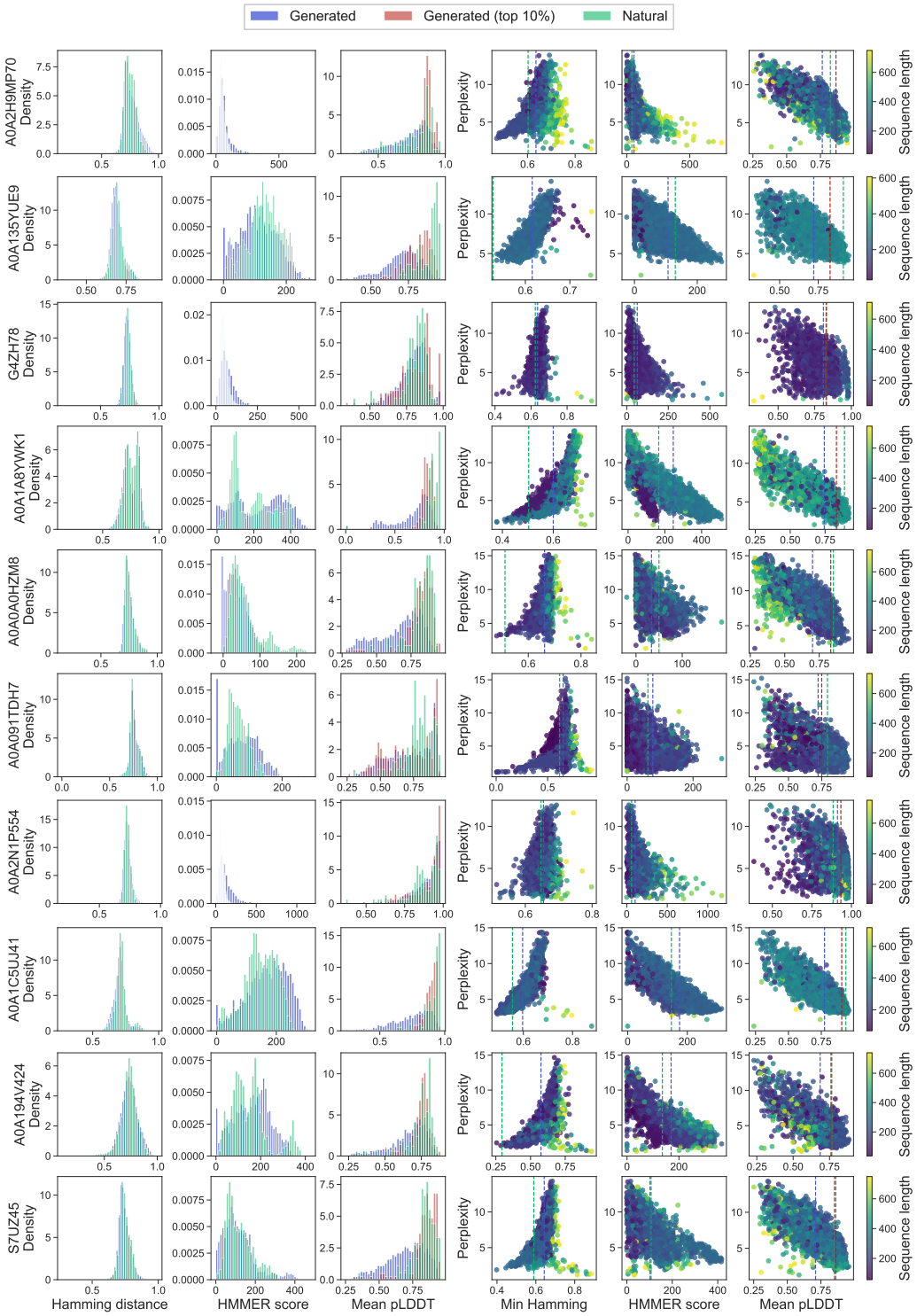

Figure 18: **Properties of generated sequences.** Left panels: histograms of Hamming distances, HMMER scores and mean pLDDT scores from ESMFold of generated sequences for 10 example test clusters (10 rows). Right panels: scatter plots of ProtMamba perplexity versus the Hamming distance to the closest natural neighbor, the HMMER score and mean pLDDT score from ESMFold for all generated sequences from each of 10 example clusters (10 rows). Dashed vertical lines: median of the generated sequences (blue), median of the natural sequences (green) and pLDDT value of the reference structure of the cluster (red). The last one is shown only for the rightmost plot.