ZymCTRL: a conditional language model for the controllable generation of artificial enzymes

Geraldene Munsamy*
Basecamp Research Ltd.
London, United Kingdom

Sebastian Lindner*
Department of Biology
Friedrich-Alexander-Universität
Erlangen, Germany

Philipp Lorenz
Basecamp Research Ltd.
London, United Kingdom
phil@basecamp-research.com

Noelia Ferruz
University of Girona
University of Bayreuth
Girona, Spain
noelia.ferruz@udg.edu

Abstract

The design of custom-tailored proteins has the potential to provide novel and groundbreaking solutions in many fields, including molecular medicine or environmental sciences. Among protein classes, enzymes are particularly attractive because their complex active sites can accelerate chemical transformations by several orders of magnitude. Since enzymes are biodegradable nanoscopic materials, they hold an unmatched promise as sustainable, large-scale industrial catalysts. Motivated by the enormous success of language models in designing novel yet nature-like proteins, we hypothesised that an enzyme-specific language model could provide new opportunities to design purpose-built artificial enzymes. Here, we describe ZymCTRL, a conditional language model trained on the BRENDA database of enzymes, which generates enzymes of a specific enzymatic class upon a user prompt. ZymCTRL generates artificial enzymes distant from natural ones while their intended functionality matches predictions from orthogonal methods. We release the model to the community.

1 Introduction

The design of proteins with tailored properties would open the door to novel approaches to address many global challenges, such as pandemics or environmental pollution. The protein class of enzymes is particularly attractive, given their capability to accelerate chemical transformations by several orders of magnitude while being biodegradable nanoscopic materials. Enormous advances have been reported in the field of enzyme design in the last two decades [24]. For example, protein engineers have successfully controlled several chemical transformations, including the Kemp elimination [35, 31], ester hydrolysis [33], Diels-Alder [37], retro-aldol [21, 4], and Morita-Baylis-Hilman [5] reactions. Despite these impressive advances, these enzymes often do not achieve the catalytic rates of their natural counterparts, with k_{cat}/K_M values being several orders of magnitude lower [22].

*These authors contributed equally

Machine Learning for Structural Biology Workshop, NeurIPS 2022.

With their precisely arranged catalytic residues, enzymes bind their substrates in well-defined catalytic cavities. Slight deviations from the atomic-level interplay among catalytic and other influencing residues and cofactors might deplete enzymatic activity entirely, significantly hindering the optimisation process and our understanding thereof. This atomic interplay, however, also opens opportunities to design improved or novel enzyme functions through targeted mutations; and although the combinatorial space of possible mutations is astronomically large, recent advances in artificial intelligence (AI) may allow us to find better solutions at a higher throughput.

AI has opened a new era in protein research. Besides the significant advances in structure prediction methods [2, 46, 23], neural networks are also greatly impacting protein design. To put progress in the field into context, in just the last three years, over 40 new neural-based protein design methods have been reported [14], meeting unprecedented success in many cases [11, 44, 43, 9]. Many of these new models come from breakthroughs in the Natural Language Processing (NLP) field [15], where much of this revolution can be attributed to the Transformer, a powerful modular architecture behind widely used applications such as Google Translator or GPT3 [7]. In the protein research realm, Transformers have met two main applications. First, trained as denoising autoencoding models [12], protein language models are efficient at embedding sequence representations, which then can be coupled to various protein downstream tasks [23, 34, 32, 6]. Second, trained to meet an autoregressive objective, i.e., predicting the next word or amino acid given a previous context, Transformers produce models capable of generating novel protein sequences, a compelling application for protein design that has met wide success [26, 16, 29, 18, 30]. One disadvantage of generativ Transformers, however, is that these usually exert little control over the properties that the generated sequences will feature when the models are used in a zero-shot fashion. There are, however, specific ways to filter or control the generated properties, e.g., by 1) fine-tuning specific families [25], 2) prompt-engineering [18], or 3) generating sequences in a high-throughput fashion and filtering those with the desired predicted properties [14].

While these are all powerful techniques, we ideally also want an end-to-end model that generates sequences upon a given user-defined prompt. An example in this area in the NLP field is CTRL, a language model trained with control tags, such as 'style' and 'topic,' which define the direction the generated text takes. This concept was applied to protein sequences in ProGen, a protein language model trained with labels defining the biological process, cellular component, function, or taxonomy [26]. While a large repertoire of ProGen models has been made public [29], the conditional models with control tags have to date not been released.

Motivated by these exciting opportunities, we trained ZymCTRL, a language model that generates enzymes upon a prompt that defines a specific catalytic reaction. ZymCTRL was trained on the BRENDA database, a dataset of 37M enzyme sequences classified according to their enzymatic class (EC). During training, we linked every sequence to its associated EC class, and the model learned the specific sequence features that define each catalytic reaction. To avoid the known issue of lack of representativeness for some families, we tokenized the labels in such a way that notions learned from class EC: 1.1.1.1 ('alcohol dehydrogenase') can be transferred to EC: 1.22.1.1 'iodotyrosine deiodinase' because they belong to the same group 'EC1 'oxydoreductases.' ZymCTRL can be used after a user-defined EC class to generate novel enzymes that catalyze that reaction. Our analysis shows that ZymCTRL generates globular, stable enzymes whose alleged functions match orthogonal function prediction methods such as ProteInfer [36]. ZymCTRL contains 36 layers totalling 738M parameters. We make this model freely available at (`https://huggingface.co/hzgz/ZymCTRL`) for the benefit of the entire community.

2 Results

We have trained a conditional language model to generate enzyme sequences that fulfil a user-defined catalytic reaction (Fig. 1). To this end, we used the Transformer architecture's decoder module [3] and trained it on the BRENDA database [8] with an autoregressive objective to obtain an enzyme language model (Methods). Language models assign probabilities to sentences *p(x)* and the tokens that compose them *(x)*, defining the problem as a next-word (or next-amino acid) prediction:

$$p(x) = \prod_{i=1}^{n} p(x \mid x_{<i}) \tag{1}$$

We trained ZymCTRL to generate sequences conditioned on a control tag, defined as the enzymatic commission number (also enzyme class) (EC) assigned to each enzyme (Fig. 1a). To promote transfer

learning among subclasses belonging to the same class, we also tokenized the EC numbers (Methods). The probability distribution is decomposed as:

$$p(x \mid EC) = \prod_{i=1}^{n} p(x \mid x_{<i}, EC) \tag{2}$$

We trained ZymCTRL by minimizing the negative log-likelihood over the entire dataset (D). This way, the model must learn the relationships between the EC control tag and the amino acids that follow:

$$L(D) = -\sum_{k=1}^{D} p(x^k \mid x^k_{<i}, EC^k) \tag{3}$$

After training, the ZymCTRL can generate enzyme sequences in a zero-shot fashion, or that match a user-defined EC number (Fig. 1b).

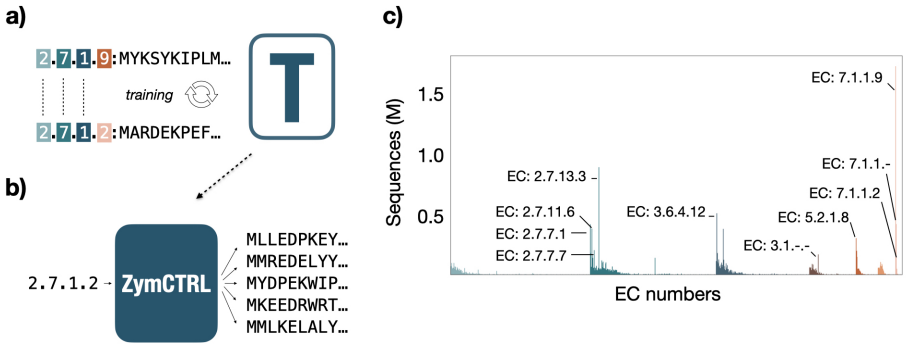

Figure 1: Training and generation process and sequences in the training database. (a) During pre-training, a Transformer (T) model learns the relationship among sequences and their tokenized labels in an iterative process. (b) At inference time, users can specify a target catalytic reaction as a condition for the model's generation. (c) Sequences in the BRENDA database classified per EC number. Depicted are the ten largest classes.

2.1 The BRENDA dataset: Implications and redundancy

The BRENDA database is a collection of enzyme sequences with annotated EC numbers [8]. We removed sequences with more than one label, giving a dataset with over 36M sequences (Methods), which is the one we refer to throughout this manuscript. BRENDA's EC numbers feature a four-level hierarchy, with each successive number defining the catalytic activity more precisely. For example, enzymes classified as EC: 2.1.1.13 are transferases (first level), transferring one-carbon groups (second level), such as methyl (third level), to specifically regenerate methionine from homocysteine (fourth level). Dashes in the hierarchy (e.g., EC: 2.1.1.-) point out a lack of specificity of functional annotation at that level. Besides the annotation disparity among classes, there are also large deviations in representation, with some classes significantly more populated than others. While the top 100 most populated classes encompass 37% of the sequences in the dataset, 9% of the 6,062 classes only include one sequence (Fig. 2c). This fact is mainly reinforced by the definition of the fourth-level class '1', which in each case comprises enzymes that are non-specific or whose specificity has not been analyzed to date (such as EC: 2.7.11.1). There are a few possibilities to partly alleviate representation bias during training. One strategy would be to ensure an equivalent number of members per class. While this approach would deplete representation biases during training, it would also not exploit a significant proportion of this valuable annotated data. In contrast, we envisioned a training strategy where the model could transfer learning from populated to underrepresented classes. In particular, we tokenized the EC labels by subclasses (Methods) for two reasons. First, to promote that the model transfers notions from populated to underrepresented subclasses within the same group (e.g. EC: 7.1.1.9 and 7.1.1.2) and second, to understand what are the minimal requisites for a good-quality per-class generation.

2.2 General quality assessment of the generated sequences

Language models assign a next-token probability given a series of previous tokens. Consequently, different generation approaches produce text with various degrees of success [19]. We have explored the models' capability to emit sequences with different parameters, by measuring the distance to the observed natural amino acid frequencies (Methods). We observe the best accuracy when sampling with top_p = 1, top_k = 9, temperature = 1, and repetition_penalty = 1.2 (Fig. S1) and use these parameters throughout this work. To assess the quality of the generated sequences at a zero-shot level, we use a set of tasks inspired by the GLUE benchmark [30, 42, 10], evaluating the predicted properties of the sequences in comparison with the training set. To this end, we crafted two datasets: the first one, which we refer to as 'natural', by sampling two random sequences per class, when possible. The second, which we denote as 'generated', by generating the same number of sequences per class with ZymCTRL (Methods). The datasets contains 11,439 sequences.

We first evaluated the percentage of sequences that are predicted to be globular using IUPRED3 [13]. Since our goal is to provide high-level comparisons between the two sets, we focused on the globular prediction model. Our analysis shows similar globularity levels between the two sets, with 97.7% of the ZymCTRL-generated sequences predicted to be globular, versus 99.3% for the natural dataset (Table 1).

Next, we focused on comparing the predicted structure for the two datasets. The rise of structure prediction methods such as AlphaFold [2], OmegaFold [46] or ESMfold [23] has enabled predicting thousands of variants in a few hours. In this case, we used OmegaFold and ESMfold to predict the structure of the two sets given as input of a single sequence (Methods). These methods produce a per-residue estimate of its confidence (plDDT) with values ranging from 0 to 100. This score has been shown to correlate with protein order for AlphaFold predictions [40]: Low scores (pLDDT > 50) tend to appear in disordered regions, while high ones (pLDDT > 90) appear in ordered ones. The mean LDDT of the generated dataset -averaging over each predicted residue- is 60.01. 38.4% of the sequences in the dataset show values over 70, in line with previously artificially generated datasets [16, 27]. For the natural dataset, the average LDDT is 84.78, with 88.9% of sequences with values over 70.

Besides, we computed structure predictions using ESMfold for the set of sequences below 400 amino acids (Methods). The results are in line with OmegaFold, with plDDT averages of 60.2 and 84.9 for the generated and natural datasets, respectively, after transferring to the 0-100 scale. These results are concordant with our previous analysis on ProtGPT2-generated and natural sequences using AlphaFold [2]. We previously reported a plDDT average of 75.3 for the natural dataset, these differences are however understandable based on the nature of the two datasets; the BRENDA dataset, comprising only functional enzymes, will on average include more globular and ordered proteins than Uniref50.

Table 1: Comparison between the structural features of natural and the generated dataset

Program	Natural Dataset	Generated Dataset
IUPRED3 (globular)	99.3%	97.7%
OmegaFold (LDDT)	84.78	60.01
ESMFold (plDDT) (<400 aa.)	84.94	61.04

2.3 The generated enzymes are distant from the natural space

One of the critical properties of language models is that they can generalize on the training set and infer novel, unseen, yet coherent texts. This is a particularly interesting property for protein design since we are interested in designing plausible, functional sequences that are, however, distant from natural ones, and hence have the potential to constitute novel solutions to established problems. To understand the extent to which ZymCTRL explores novel sequences, we ran MMseqs2 [38] searches on the generated dataset versus the BRENDA training set (Fig. 2a) and BLASTP searches against the non-redundant protein sequence database (Fig. S4).

The sequences are distant from the training set, with alignments that show average identities and lengths of 53.1 ± 23.2% and 337.9 ± 151.2 amino acids. This is a remarkable feature since we aim to obtain a model that generates solutions to chemical transformations that are distant from the protein space. Nevertheless, we observe a non-negligible set of sequences with identities over 90%;

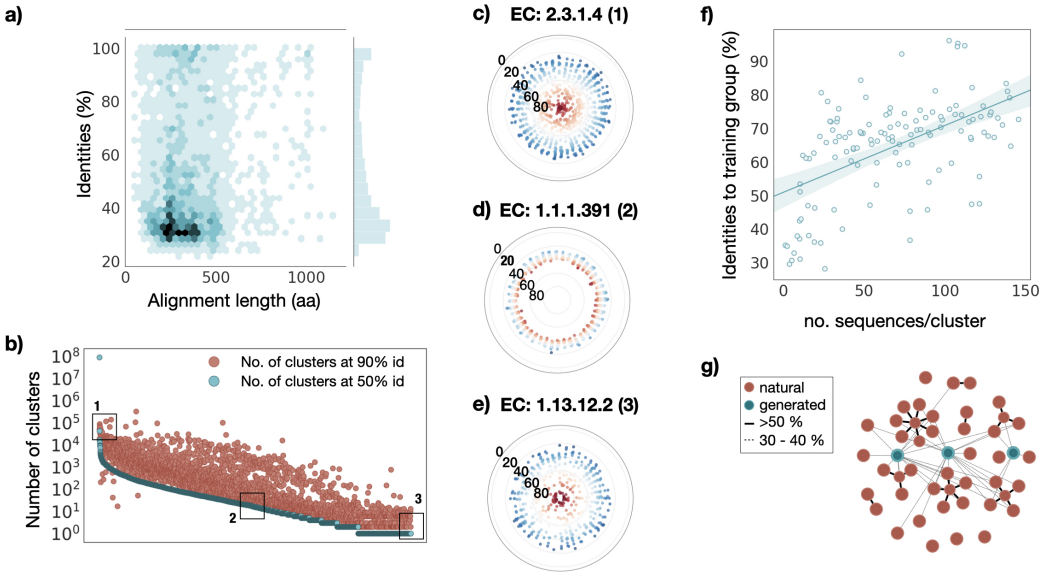

Figure 2: Distance of the generated sequences to the training set. a) Identities and lengths for the alignment with the lowest E-value found with MMseqs. b) Number of clusters at 90 and 50% redundancy per sequence label, with three distant regions depicted and extended in c-e. c) Distance in % identity to the sequences in the dataset for the three EC classes 2.3.1.4, 1.1.1.391 d), and 3.1.1.55 e). f) Scatter plot between the identities to training set groups for generated sequences and the average number of sequences per cluster, for 100 EC classes. g) Network visualization of natural and three generated sequences in EC class 1.1.1.391. The generated sequences show low identities to several clusters.

12.5% of the sequences in the set (Fig. 2a). These sequences span all major enzymatic classes and belong to groups with diverse numbers of members, spanning six orders of magnitude (Fig. S4). We hypothesized that those sequences might belong to specific classes with high internal redundancy, i.e., EC classes whose sequences are homologous and within 90% identities to themselves. Hence, we looked at possible determinants of generation redundancy at the EC class level. To this end, we first clustered each of the training set's EC class sequences at 90 and 50% identity (which we denote as cluster_{90} and cluster_{50}, Fig. 2b). Then, we generated 1000 sequences of each EC class in the training set. Panels 2c-e showcase the distance among generated and training set sequences for three EC class examples. In principle, we did not observe differences depending on the region of the clustering scale, for example, the model produces sequences at all identity ranges for classes 2.3.1.4 and 1.13.12.2, even if they are at the two axis extremes. Surprisingly, EC class 1.1.1.391 shows remarkably distant identities to the natural sequences for all its generated sequences (Fig. 2d). A more detailed feature comparison against the other EC classes, such as 2.3.1.4, revealed that the main difference between the two sets is the average number of members per cluster50.

To identify whether this difference is an example of a more general trend, we (1) computed identities for each of the 1,000 generated sequences per group against the natural sequences and computed their averages and (2) plotted these values against the number of sequences per cluster_{50} (Fig. 2f). Interestingly, the analysis revealed that there is a relationship (p-value = 6.04e-11, R=0.54) between these two variables. We had also envisioned relationships to the number of clusters or the total number of sequences within groups; these trends were however not observed (Fig. S5). To better exemplify the connections among generated and natural sequences within groups, we looked at the particular example EC: 1.1.1.391, where the natural sequences form 19 clusters50 (Fig. 2g). The generated sequences show identities in the long-range (30-40%) to many clusters, as opposed to showing higher identities to a single cluster. Fig. 2g depicts three example sequences (blue). In other words, rather than extending natural clusters, the model interpolates among all clusters to generate new ones. These findings have implications for protein design, since specific user cases may require closer (i.e 90%) or more distant (i.e <40%) sequences to the natural ones. Besides, it points towards

the model capturing general features of the different clusters and generating new, distant groups with interpolated properties across them.

2.4 ZymCTRL sequences catalyze their expected reactions

ZymCTRL's key component is its ability to generate sequences with the potential to catalyse a user-defined chemical reaction. Before proceeding to experimental characterization, it is paramount to characterize to what extent other methods agree on the predicted functionality. To this end, we predicted ZymCTRL sequences' function with ProteInfer [36], a convolutional neural network for functional annotation using unaligned amino acid sequences as input. While the use of other models to predict protein function is not without its limitations, ProteInfer was trained with a different neural architecture and training objective and constitutes an orthogonal method to our predictions.
We computed ProteInfer predictions for a subset of one sequence per label in the natural and generated datasets, totalling 6,000 sequences. ProteInfer generates a set of predicted GO terms given a sequence, and an EC number for sequences with predicted catalytic capabilities. For the natural dataset, 45.9% of the sequences were returned with an EC prediction. Regarding the generated dataset, 30.2% of the sequences had an EC label assigned (Table 2). Discrepancies between assignments in generated and natural dataset could be indeed due to a lack of functionality for some of the sequences, but also to the distance of these sequences to the natural dataset, in particular to the training set that Proteinfer learned during training.
Out of the confidently-predicted sequences, 81.2% and 80.9% of sequences were predicted with their correct top-level EC class (e.g. EC:1: oxydoreductases), for the natural and generated datasets, respectively. Successive levels in the hierarchy are more challenging to predict due to the increasing specificity of the described chemical reactions, and hence it is expected that fewer sequences will match the predictions. In this regard, 79.6% and 76.5% of sequences were assigned to their correct second-level class, 76.1% and 73.5% their third-level class, and 62.0% and 54.0% their complete EC class, for the natural and generated dataset (Table 2). These results are remarkable, taking into account that the sequences were generated in a zero-shot fashion, with no previous fine-tuning on the specific classes. Besides, the results suggest that a significant amount of sequences have the potential to catalyze their intended reactions while conferring distant, novel solutions in the sequence space.

Table 2: Accuracy of ProteInfer predictions for the natural and generated datasets at each EC level.

	1st EC level	2nd EC level	3rd EC level	4th EC level	EC labels assigned
Natural	81.2%	79.6%	76.1%	62.0%	45.9%
Generated	80.9%	76.5%	73.5%	54.0%	30.2%

2.5 The generated sequences feature complex, non-idealized structures

The protein design field has provided a wealth of *de novo* structures, especially in the last few years [20]. While these proteins pose a significant advance in the field, they are often idealized structures with minimal loops and often lack the necessary structural embodiments and dynamic properties to interact with other molecules such as substrates. The emergence of alternative end-to-end protein design methods in the last two years has shown new designs with natural-like structures. Recently, we trained a protein language model termed ProtGPT2, and observed that the generated sequences feature large loops and embodiments capable of accommodating binding partners, paving the way for functionalization (manuscript in preparation). To characterize ZymCTRL's capabilities at generating structures with large loops and embodiments -a critical property in enzyme design, which require flexible regions that specifically accommodate substrates and products- we attempted to visualize its coverage of the enzyme space while parsing its structural predictions. Studies to reduce the large dimensionality of protein sequences in a few human-understandable dimensions have focused on hierarchical characterizations [17], cartesian representations, or similarity networks [28]. Recently, manifold learning techniques, such as tSNE and UMAP, have emerged as powerful dimensionality reduction and visualization tools.
Motivated by these advances, we generated ZymCTRL representations for both natural and generated datasets and reduced their dimensionality using UMAP. Figure 3 shows the first two UMAP projections, revealing that the model has learned to capture differences among the main catalytic classes. We besides observed large structural diversity across the generated examples and show one

example per major catalytic class. In particular, we report an angelicin synthase (1.14.14.148), a demethylmacrocin O-methyltransferase (2.1.1.102), a 4-chlorobenzoyl-CoA dehalogenase (3.8.1.7), a pyridinium-3,5-bisthiocarboxylic acid mononucleotide nickel chelatase (4.99.1.12), an isopenicillin-N epimerase (5.1.1.17), a cholate-CoA ligase (6.2.1.7), and an ABC-type methionine transporter (7.4.2.11). For each case, we searched the most similar structure using Foldseek, and in all cases, the best hits returned structures of the same fold. However, the selected examples reveal distant sequences to the best hits, sometimes reaching values as low as 26.5% identity. As evidenced in the examples, the sequences show predicted complex structures with natural-like surfaces and multiple cavities that, in principle have the potential to adjust to incoming substrates and release reaction products.

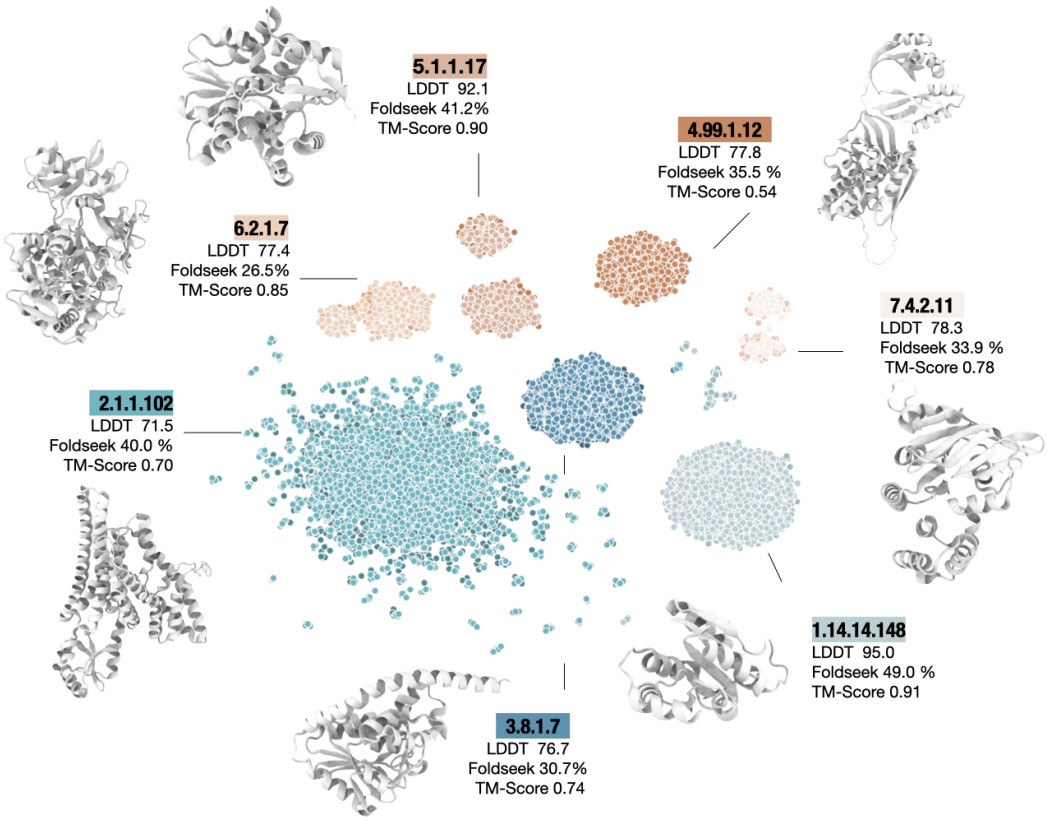

Figure 3: UMAP projection into two dimension of the generated and natural datasets and examples of ZymCTRL proteins. The model discerns among categories. We show an example per class with the structures predicted with OmegaFold. We indicate the pLDDT value, and the identity and ™-score to the best hit obtained with Foldseek.

2.6 The model transfers learning to underrepresented groups

Since ZymCTRL has been trained conditionally on very differently populated EC classes - ranging from one sequence per group to more than a million-, we attempted to understand the general transfer capabilities of the model and the effect of our label tokenization. To this end, we first computed the perplexity of the generated dataset and visualized these results by EC class, plotting against the total number of sequences present in the training set per EC class (Fig. 4a). When analyzing the entire dataset, the average perplexity of the generated sequences is 6.17 ± 2.94. We observed an increase in perplexity for generated sequences belonging to less-populated classes, with a Spearman rank correlation coefficient of -0.64. To exemplify this trend, 50.4% of the dataset contain classes with less than 100 sequences, and these sequences show an average perplexity of 7.94 ± 1.89. Consequently,

these classes would benefit from fine-tuning with different sequences to produce higher-confidence values. However, despite this perplexity-frequency correlation, the model can still produce sequences with low perplexity values. Even in highly underrepresented classes (<30 sequences), the model generates sequences with perplexity scores comparable to classes trained with more than 1000 sequences (Fig. 4). These results indicate that in scenarios where no fine-tuning is possible, users could still potentially generate and filter for high-confidence sequences in a zero-shot fashion.
As a second analysis towards characterizing transfer learning, we computed perplexity for sequences in training and generated datasets. To this end, we generated 50 sequences for 20 classes with more than 1,000 members in the training sets. Besides, we also computed perplexity for labels in the validation dataset, corresponding to labels that the model had not seen during training but still had plausible nomenclatures. Lastly, we manually created labels with the right amount of tokens but with not-seen numbers during training, such as '9.9.9.9' (Table S1). We computed perplexity for 50 sequences per label, totaling 20 labels and giving 1000 sequences per dataset (Fig. 4b). The average perplexity of the training set is 2.34 ± 1.37, followed by the generated dataset with 2.17 ± 1.00. These values reflect the capacity of the model in capturing the training distribution and generating sequences that are distinct from natural sequences yet comparable to naturally functional sequences. The perplexity for out-of-dataset labels (validation set) showed an increased mean perplexity of 4.02 ± 1.3. In contrast, the mean perplexity of the invented labels is 7.94 ± 2.42.

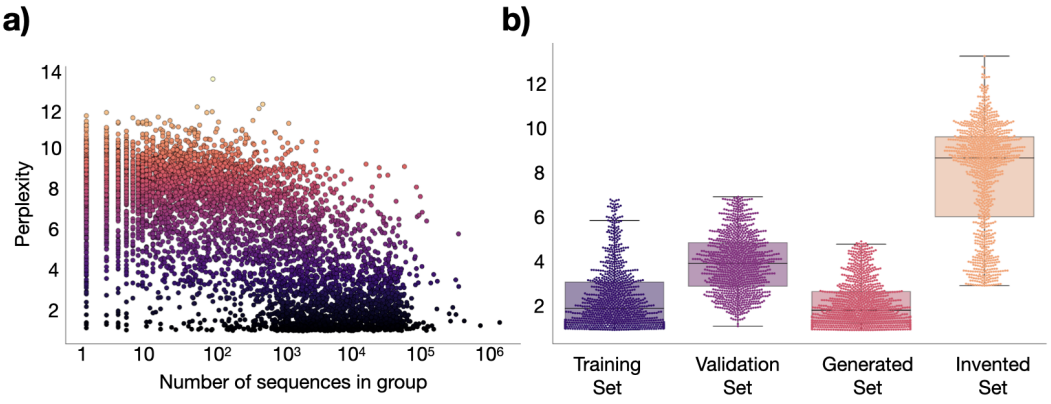

Figure 4: Perplexity of sequences depending on group representativeness and dataset type. (a) Relationship between perplexity of the generated sequences per class and the number of members in their training set groups. (b) Perplexities for sequences in several training groups. Validation set comprises labels not seen during training. Invented labels are shown in Table S1.

3 Discussion

The use of deep learning for the design of novel proteins has exploded in the last two years.13 Motivated by the recent advances in NLP, and in particular, in text generation, we recently trained ProtGPT2, a language model that generates ordered, globular sequences in an unconditioned fashion. This work has aimed to advance significantly forward, i.e., by pairing sequences to their functions to provide a conditional language model. In particular, we have trained ZymCTRL, a conditional language model trained on the currently known enzyme space. ZymCTRL produces globular, ordered proteins whose predicted reactions with orthogonal methods match those intended by a user-defined prompt. We tokenized labels during training to alleviate the lack of representativeness for some specific enzyme classes, permitting the model to transfer knowledge for mechanistically similar enzymes. Our results show that the model can generate high-confidence sequences even for highly underrepresented classes provided sufficient inferred data. Fine-tuning on specific classes with added labels will further increase confidence for those cases, while zero-shot generation of populated classes may provide fit results. Near-future efforts include fine-tuning on specific classes and experimental testing of several variants.

4 Code Availability

The model is available at https://huggingface.co/nferruz/ZymCTRL

5 Acknowledgments

The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High-Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b114cb (UID 210235). NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. We thank Thomas Zeiser for his considerate support. N.F. acknowledges support from an AGAUR Beatriu de Pinós MSCA-COFUND Fellowship (project 2020-BP-00130). We thank Florian Grün for stimulating discussions. We thank Marc Garcia-Borràs for his helpful feedback and inspiring discussions.

6 Methods

6.1 Dataset preparation and vocabulary encoding

We downloaded the BRENDA database [8] through the UniProt web interface [39] (on July 2022) giving a total of 37,624,812 sequences. To avoid multi-chain sequences with multiple EC assignments, we removed sequences with several EC labels, giving a total of 36,276,604 sequences. We split the database into training (90%) and evaluation (10%) datasets. We used a block size of 1024, separated control tags and sequences with a separator token, and further specified the boundaries of sequences below 1024 amino acids with start and end tokens. We fit as many complete sequences as possible in the 1024 window, provided that sequences are not split across blocks. The sequences will follow this schema if their length fits in the 1024 window: <control tag><sep><start><ENZYME SEQUENCE><end><|endoftext|>, and the following scheme otherwise: <control tag><sep><ENZYME SEQUENCE><|endoftext|>. Sequences over 1024 amino acids (3%) were truncated to the N-terminal part.

6.2 Model finetuning

Additional metagenomic sequences for finetuning were derived from Basecamp Research's knowledge graph. Environmental samples subjected to metagenomic sequencing were collected after receiving landowner's permission and entering access-benefit-sharing agreements with the relevant local or national authority, following Nagoya protocol guidelines. All samples were sequenced with both long-read (Oxford Nanopore GridION) and short-read (Illumina NovaSeq 6000) sequencing methods applied to each sample after extraction. Following standard sequencing QC, an assembly-based approach was followed, generating *de novo* assemblies that were subjected to polishing and open-reading frame annotation. Each gene was functionally annotated (including KEGG, COG, and EC number). Translated protein sequences alongside functional, genomic and sample information were inserted into Basecamp's graph database.

6.3 Vocabulary encoding

We train our model with an associated label (control tag) per sequence. Following recent studies [18, 30], we tokenized our enzyme sequences using amino acid encoding. We further tokenized the labels in the dataset, to account for similarities among sub-classes in the same classes and help the model generalize in lower-populated catalytic reactions. This way, the control tag '1.1.1.1' is split into its categories ('1' + '.' + '1' + '.' + '1' + '.' + '1') and shares 6 tokens with '1.1.1.2'.

6.4 Model pre-training

We use a Transformer decoder model as architecture for our training. The model uses the original dot-scale self-attention [41]. The architecture uses that of the CTRL/GPT2 Transformer, which was downloaded from HuggingFace [45]. ZymCTRL consists of 36 layers, a model dimensionality of 1260, and 16 attention heads. The model was optimized using Adam $\beta 1 = 0.9, \beta 2 = 0.999$ with

a learning rate of 0.8e-04, following previous works [29]. A batch size of 4 per device was used accumulating 4 gradient steps, totaling a total batch size of 768. We trained for 179,000 steps on 48 NVIDIA A100s 80GB for about 15,000 GPU hours. Parallelism of the model was handled with DeepSpeed [1].

6.5 Dataset creation

We randomly sampled two sequences per EC number for multi-sequence classes, one otherwise. For the generated dataset, we generated 20 sequences per EC number and selected the best or two best perplexity-scoring sequences depending on the number of sequences in the natural dataset's equivalent classes. Each dataset contained 11,439 sequences. We ensured that the generated sequences followed the natural dataset length distribution (Fig. S2), by applying a length limit of 600 to all labels, except when no sequence could be generated at that length, hence the limit was extended to 1024. In all cases, the sequences were only selected if they had been finished and not truncated by the model. For the analysis of function prediction with ProteInfer, we selected and generated one sequence per EC number, totaling 6,000 sequences per dataset.

6.6 ProteInfer analysis

We sent natural and generated datasets to the ProteInfer web interface (`https://google-research.github.io/proteinfer/`) [36] and extracted the predicted EC numbers. Subsequently, we computed the accuracy by comparing the ground truth EC numbers with the predicted labels, also in the case of multi-labeled predictions.

6.7 IUPRED3 analysis

IUPRED3 [13] was run on both datasets using the 'glob' option.

6.8 Structure prediction

We computed the OmegaFold structure prediction for each sequence in the two datasets using the max –subbatch_size that fit into memory for each sequence length range [46]. We computed the ESMfold [23] predictions for sequences with less than 400 amino acids. In total the datasets contained 6753 and 6435 for natural and generated datasets, respectively.

6.9 Amino acid propensities

We computed the 'natural' amino acid propensities by taking all sequences in the BRENDA database. The 'generated' dataset was created using as input to the model the ten largest EC classes in BRENDA (Fig. 1b) and generating 20 sequences per parameter set. We tested a sampling generative procedure [19], with a temperature of 1, max_length of 1024, top_p of 1, repetition penalty 1.2 and 1.3, and top_k for the values from 5 to 20, and 30, 50, 100, 200, and 458. Accuracy to match the natural distribution was computed as the sum of the absolute differences between all amino acid pairs. Repetition penalty of 1.2 provided better results in all cases. Top_k=9 gave the closest distribution to the target propensities (Fig. S1).

References

[1] DeepSpeed | proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining.

[2] Highly accurate protein structure prediction with AlphaFold | nature.

[3] Papers with code - language models are unsupervised multitask learners.

[4] Eric A. Althoff, Ling Wang, Lin Jiang, Lars Giger, Jonathan K. Lassila, Zhizhi Wang, Matthew Smith, Sanjay Hari, Peter Kast, Daniel Herschlag, Donald Hilvert, and David Baker. Robust design and optimization of retroaldol enzymes. 21(5):717–726. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/pro.2059.

[5] Sinisa Bjelic, Lucas G. Nivón, Nihan Çelebi Ölçüm, Gert Kiss, Carolyn F. Rosewall, Helena M. Lovick, Erica L. Ingalls, Jasmine Lynn Gallaher, Jayaraman Seetharaman, Scott Lew, Gaetano Thomas Montelione, John Francis Hunt, Forrest Edwin Michael, K. N. Houk, and David Baker. Computational design of enone-binding proteins with catalytic activity for the morita–baylis–hillman reaction. 8(4):749–757. Publisher: American Chemical Society.

[6] Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. ProteinBERT: a universal deep-learning model of protein sequence and function. 38(8):2102–2110.

[7] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners.

[8] Antje Chang, Lisa Jeske, Sandra Ulbrich, Julia Hofmann, Julia Koblitz, Ida Schomburg, Meina Neumann-Schaal, Dieter Jahn, and Dietmar Schomburg. BRENDA, the ELIXIR core data resource in 2021: new developments and updates. 49:D498–D508.

[9] A. Courbet, J. Hansen, Y. Hsia, N. Bethel, Y.-J. Park, C. Xu, A. Moyer, S. E. Boyken, G. Ueda, U. Nattermann, D. Nagarajan, D.-A. Silva, W. Sheffler, J. Quispe, A. Nord, N. King, P. Bradley, D. Veesler, J. Kollman, and D. Baker. Computational design of mechanically coupled axle-rotor protein assemblies. 376(6591):383–390. Publisher: American Association for the Advancement of Science.

[10] 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.

[11] J. Dauparas, I. Anishchenko, N. Bennett, H. Bai, R. J. Ragotte, L. F. Milles, B. I. M. Wicky, A. Courbet, R. J. de Haas, N. Bethel, P. J. Y. Leung, T. F. Huddy, S. Pellock, D. Tischer, F. Chan, B. Koepnick, H. Nguyen, A. Kang, B. Sankaran, A. K. Bera, N. P. King, and D. Baker. Robust deep learning–based protein sequence design using ProteinMPNN. 0(0):eadd2187. Publisher: American Association for the Advancement of Science.

[12] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, 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.

[13] Gábor Erdős, Mátyás Pajkos, and Zsuzsanna Dosztányi. IUPred3: prediction of protein disorder enhanced with unambiguous experimental annotation and visualization of evolutionary conservation. 49:W297–W303.

[14] Noelia Ferruz, Michael Heinzinger, Mehmet Akdel, Alexander Goncearenco, Luca Naef, and Christian Dallago. From sequence to function through structure: deep learning for protein design. Pages: 2022.08.31.505981 Section: Confirmatory Results.

[15] Noelia Ferruz and Birte Höcker. Controllable protein design with language models. 4(6):521–532. Number: 6 Publisher: Nature Publishing Group.

[16] Noelia Ferruz, Steffen Schmidt, and Birte Höcker. ProtGPT2 is a deep unsupervised language model for protein design. 13(1):4348. Number: 1 Publisher: Nature Publishing Group.

[17] Naomi K. Fox, Steven E. Brenner, and John-Marc Chandonia. SCOPe: Structural classification of proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. 42:D304–D309.

[18] Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. RITA: a study on scaling up generative protein sequence models.

[19] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration.

[20] Po-Ssu Huang, Scott E. Boyken, and David Baker. The coming of age of de novo protein design. 537(7620):320–327.

[21] Lin Jiang, Eric A. Althoff, Fernando R. Clemente, Lindsey Doyle, Daniela Röthlisberger, Alexandre Zanghellini, Jasmine L. Gallaher, Jamie L. Betker, Fujie Tanaka, Carlos F. Barbas, Donald Hilvert, Kendall N. Houk, Barry L. Stoddard, and David Baker. De novo computational design of retro-aldol enzymes. 319(5868):1387–1391. Publisher: American Association for the Advancement of Science.

[22] Horst Lechner, Noelia Ferruz, and Birte Höcker. Strategies for designing non-natural enzymes and binders. 47:67–76.

[23] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, and Alexander Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction. Pages: 2022.07.20.500902 Section: New Results.

[24] Sarah L. Lovelock, Rebecca Crawshaw, Sophie Basler, Colin Levy, David Baker, Donald Hilvert, and Anthony P. Green. The road to fully programmable protein catalysis. 606(7912):49–58.

[25] Ali Madani, Ben Krause, Eric R. Greene, Subu Subramanian, Benjamin P. Mohr, James M. Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z. Sun, Richard Socher, James S. Fraser, and Nikhil Naik. Deep neural language modeling enables functional protein generation across families. Pages: 2021.07.18.452833 Section: New Results.

[26] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language modeling for protein generation. Pages: 2020.03.07.982272 Section: New Results.

[27] Lewis Moffat, Shaun M. Kandathil, and David T. Jones. Design in the DARK: Learning deep generative models for de novo protein design.

[28] Sergey Nepomnyachiy, Nir Ben-Tal, and Rachel Kolodny. Global view of the protein universe. 111(32):11691–11696.

[29] Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models.

[30] Pascal Notin, Mafalda Dias, Jonathan Frazer, Javier Marchena-Hurtado, Aidan Gomez, Debora S. Marks, and Yarin Gal. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval.

[31] Heidi K. Privett, Gert Kiss, Toni M. Lee, Rebecca Blomberg, Roberto A. Chica, Leonard M. Thomas, Donald Hilvert, Kendall N. Houk, and Stephen L. Mayo. Iterative approach to computational enzyme design. 109(10):3790–3795. Publisher: Proceedings of the National Academy of Sciences.

[32] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with TAPE.

[33] Florian Richter, Rebecca Blomberg, Sagar D. Khare, Gert Kiss, Alexandre P. Kuzin, Adam J. T. Smith, Jasmine Gallaher, Zbigniew Pianowski, Roger C. Helgeson, Alexej Grjasnow, Rong Xiao, Jayaraman Seetharaman, Min Su, Sergey Vorobiev, Scott Lew, Farhad Forouhar, Gregory J. Kornhaber, John F. Hunt, Gaetano T. Montelione, Liang Tong, K. N. Houk, Donald Hilvert, and David Baker. Computational design of catalytic dyads and oxyanion holes for ester hydrolysis. 134(39):16197–16206. Publisher: American Chemical Society.

[34] 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. 118(15):e2016239118. Publisher: Proceedings of the National Academy of Sciences.

[35] Daniela Röthlisberger, Olga Khersonsky, Andrew M. Wollacott, Lin Jiang, Jason DeChancie, Jamie Betker, Jasmine L. Gallaher, Eric A. Althoff, Alexandre Zanghellini, Orly Dym, Shira Albeck, Kendall N. Houk, Dan S. Tawfik, and David Baker. Kemp elimination catalysts by computational enzyme design. 453(7192):190–195. Number: 7192 Publisher: Nature Publishing Group.

[36] Theo Sanderson, Maxwell L. Bileschi, David Belanger, and Lucy J. Colwell. ProteInfer: deep networks for protein functional inference. Publisher: Cold Spring Harbor Laboratory _eprint: https://www.biorxiv.org/content/early/2021/10/06/2021.09.20.461077.full.pdf.

[37] Justin B. Siegel, Alexandre Zanghellini, Helena M. Lovick, Gert Kiss, Abigail R. Lambert, Jennifer L. St.Clair, Jasmine L. Gallaher, Donald Hilvert, Michael H. Gelb, Barry L. Stoddard, Kendall N. Houk, Forrest E. Michael, and David Baker. Computational design of an enzyme catalyst for a stereoselective bimolecular diels-alder reaction. 329(5989):309–313. Publisher: American Association for the Advancement of Science.

[38] Martin Steinegger and Johannes Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. 35(11):1026–1028.

[39] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. 49:D480–D489.

[40] Kathryn Tunyasuvunakool, Jonas Adler, Zachary Wu, Tim Green, Michal Zielinski, Augustin Žídek, Alex Bridgland, Andrew Cowie, Clemens Meyer, Agata Laydon, Sameer Velankar, Gerard J. Kleywegt, Alex Bateman, Richard Evans, Alexander Pritzel, Michael Figurnov, Olaf Ronneberger, Russ Bates, Simon A. A. Kohl, Anna Potapenko, Andrew J. Ballard, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Ellen Clancy, David Reiman, Stig Petersen, Andrew W. Senior, Koray Kavukcuoglu, Ewan Birney, Pushmeet Kohli, John Jumper, and Demis Hassabis. Highly accurate protein structure prediction for the human proteome. 596(7873):590–596. Number: 7873 Publisher: Nature Publishing Group.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need.

[42] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding.

[43] Jue Wang, Sidney Lisanza, David Juergens, Doug Tischer, Joseph L. Watson, Karla M. Castro, Robert Ragotte, Amijai Saragovi, Lukas F. Milles, Minkyung Baek, Ivan Anishchenko, Wei Yang, Derrick R. Hicks, Marc Expòsit, Thomas Schlichthaerle, Jung-Ho Chun, Justas Dauparas, Nathaniel Bennett, Basile I. M. Wicky, Andrew Muenks, Frank DiMaio, Bruno Correia, Sergey Ovchinnikov, and David Baker. Scaffolding protein functional sites using deep learning. 377(6604):387–394. Publisher: American Association for the Advancement of Science.

[44] B. I. M. Wicky, L. F. Milles, A. Courbet, R. J. Ragotte, J. Dauparas, E. Kinfu, S. Tipps, R. D. Kibler, M. Baek, F. DiMaio, X. Li, L. Carter, A. Kang, H. Nguyen, A. K. Bera, and D. Baker. Hallucinating symmetric protein assemblies. 0(0):eadd1964. Publisher: American Association for the Advancement of Science.

[45] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45. Association for Computational Linguistics.

[46] Ruidong Wu, Fan Ding, Rui Wang, Rui Shen, Xiwen Zhang, Shitong Luo, Chenpeng Su, Zuofan Wu, Qi Xie, Bonnie Berger, Jianzhu Ma, and Jian Peng. High-resolution de novo structure prediction from primary sequence. Pages: 2022.07.21.500999 Section: New Results.

7 Appendix

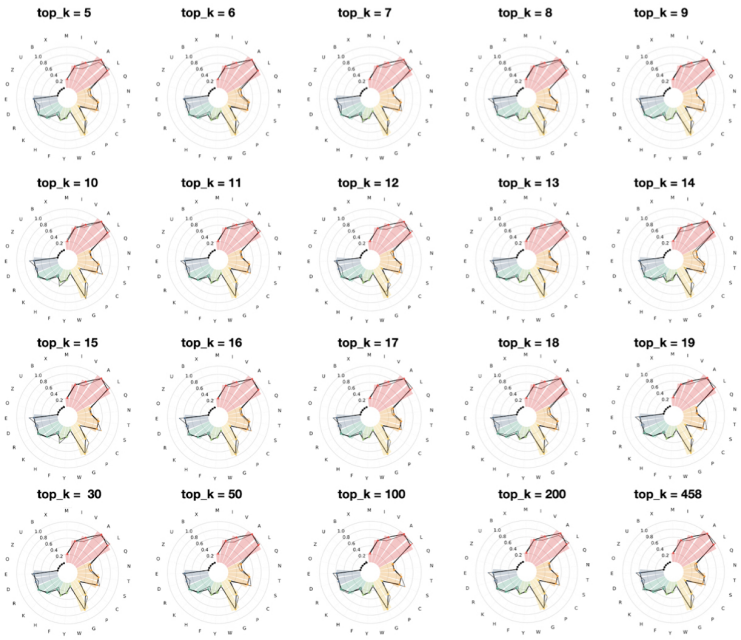

Figure 5: Comparison of different parameter sets at recapitulating the natural sequences' amino acid propensity. The amino acids in the dataset are normalised (0,1) and shown in wider lines are the natural propensities. Different parameters approximate to different extents this distribution, with top_k = 9 being the closest.

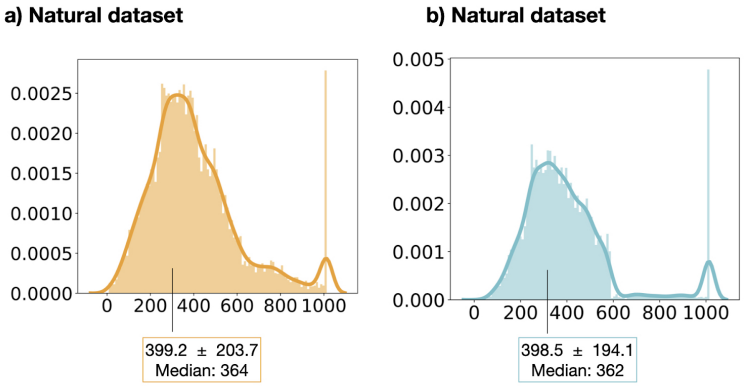

Figure 6: Length distribution for the natural (a) and generated (b) datasets.

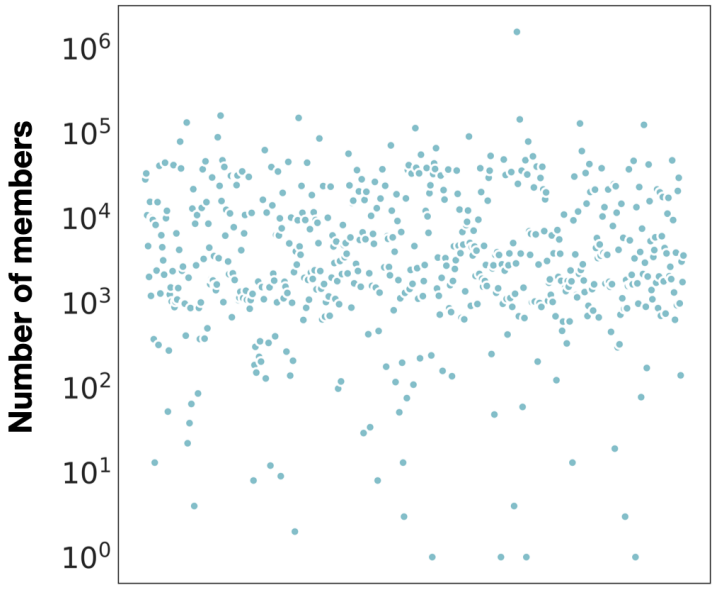

Figure 7: Number of members for each of the classes with hits over 90% in Figure 3 in the main text. The classes are shown in random order on the x-axis.

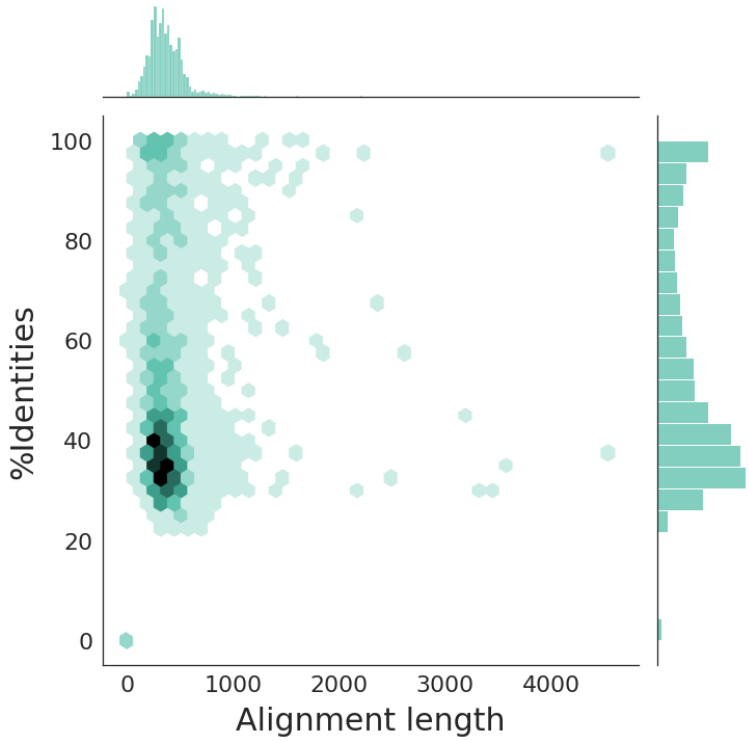

Figure 8: Identities and lengths for the best alignment found according to the E-value with BLASTp against the non-redundant protein sequence (nr) database

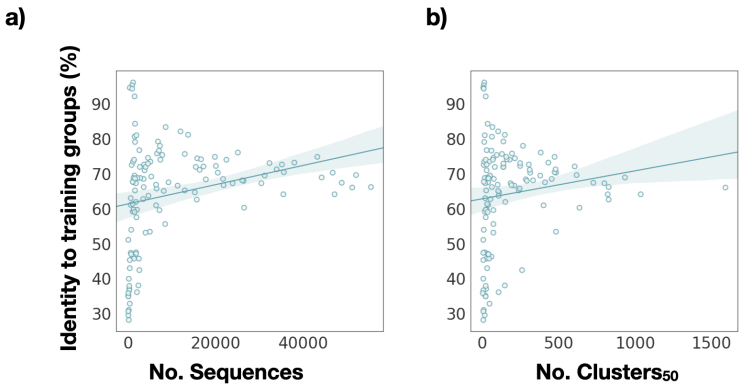

Figure 9: Mean identities for generated sequences to their training set groups as a function of (a) number of sequences and (b) number of clusters at 50%