ProGen: Language Modeling for Protein Generation

Ali Madani[1]*, **Bryan McCann**[1], **Nikhil Naik**[1], **Nitish Shirish Keskar**[1], **Namrata Anand**[2], **Alexander Chu**[2], **Raphael Eguchi**[2], **Possu Huang**[2], **Richard Socher**[1]

[1]Salesforce Research, [2]Stanford University

Abstract

Generative modeling for protein engineering has the potential to solve fundamental problems in synthetic biology, medicine, and material science. We pose protein engineering as an unsupervised sequence generation problem in order to leverage the exponentially growing set of proteins that lack costly, structural annotations. We train a 1.2B-parameter language model, ProGen, on $\sim$280M protein sequences conditioned on taxonomic and keyword tags such as molecular function and cellular component. This provides ProGen with a large range of evolutionary sequence diversity, allowing it to generate realistic proteins according to primary sequence similarity, secondary structure accuracy, and conformational energy metrics. Further analysis reveals that ranking mutated sequences by ProGen perplexity provides a zero-shot method for assessing functional fitness and ProGen may be used to construct novel sequence libraries that resemble natural proteins.

1 Introduction

Generating proteins with desired properties is one of the most challenging yet impactful problems in biology. However, leading experimental techniques for protein engineering such as directed evolution [4] still rely on heuristics and random mutations to select initial candidate sequences.

The raw amino acid sequence encodes a protein, and during synthesis, this chain of amino acids folds in ways that exhibit local (secondary) and global (tertiary) structure. These structural properties then directly determine a unique function, which is of ultimate interest to protein engineers. Unfortunately, obtaining three-dimensional structural information for proteins is expensive and time consuming. Consequently, there are three orders of magnitude more raw sequences than there are sequences with structural annotations, and protein sequence data is growing at a near exponential rate.

Recent research [2, 33, 31] has begun to capitalize on the much larger set of raw protein sequences by adapting state-of-the-art representation learning techniques [14] from natural language processing (NLP) to classification of protein properties. However, there has been no attempt to adapt state-of-the-art methods for artificial text generation [30], and in particular the kind of controllable generation [23] that would be most useful for protein engineering.

We introduce ProGen for controllable protein generation (Figure 1). ProGen is a 1.2 billion parameter conditional language model trained on a dataset of 280 million protein sequences together with conditioning tags that encode annotations such as taxonomic, functional, and locational information. ProGen is a powerful language model as determined by perplexity and accuracy metrics. ProGen's performance improves in settings with larger amino acid and conditioning tag context, which highlights its potential for applications in generating viable, starting sequences for directed evolution or *de novo* protein design [19]. ProGen also performs well when used to model unseen protein families, but it is even more effective when fine-tuned for those unseen families. These results inspire the use of ProGen to generate candidate sequences in challenging, low-homology applications.

[*] Correspondence to: Ali Madani <amadani@salesforce.com>

Machine Learning for Structural Biology Workshop, NeurIPS 2020, Vancouver, Canada.

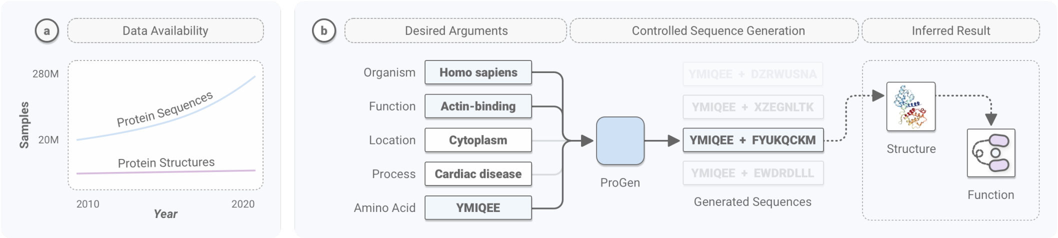


Figure 1: a) Protein sequence data is growing exponentially as compared to structural data. b) We use protein sequences and metadata (tags) to train a conditional language model.

Proteins generated by ProGen satisfy desired structural properties when evaluated with metrics for sequence similarity, secondary structure accuracy, and conformational energy—from lower level structure to higher level structure. Generation performance is judged higher quality by higher level metrics, which suggests that ProGen has learned invariances to mutations at the sequence level that conserve structure and inferred function. At the highest level, conformational energy analysis reveals that generated proteins exhibit energy levels near that of native proteins, providing our strongest evidence that these proteins satisfy the desired structure and potentially its inferred function. Case studies with held-out proteins for sequence completion, zero-shot fitness selection, and protein library construction provide further evidence that ProGen may be used to generate and screen proteins that may be structurally and functionally valid.

2 Related Work

Protein representation learning. Recent methods for contextualized representations [25, 28, 14] in NLP have been demonstrated to work well for contextual protein representation learning. Structural information about a protein can be extracted from such representations using linear methods, and the representations themselves can be adapted to improve performance on other tasks [33]. Similarly, UniRep [2] showed that such representations could be used to predict stability of natural and *de novo* designed proteins as well as quantitative function of molecularly diverse mutants. TAPE [31] is a new benchmark consisting of five tasks for assessing such protein embeddings. While this body of prior work along with [15] focuses on transferable representation learning using bidirectional models, our work demonstrates controllable protein engineering with generative, unidirectional models.

Generative models for protein engineering. Recent generative modeling work [34, 22] has shown promising directions such as Ingraham et al. [20] which extends the transformer to condition it on a graph-structured specification of a desired target. Anand & Huang [3] utilizes generative adversarial networks to produce 2D pairwise distance map for given protein structural fragments, essentially in-painting missing residues. The aforementioned work, along with O'Connell et al. [27], Boomsma & Frellsen [11], and Greener et al. [17], all utilize explicit structural information for generative modeling, thereby are unable to fully capture the number and diversity of sequence-only data available. Meanwhile sequence-only generative modeling have been attempted recently through residual causal dilated convolutional neural networks [32] and variational autoencoders [12]. In contrast, our work on generative modeling focuses on a high-capacity, controllable language models that scale well with sequence data.

Language models and controllable generation. Large Transformer architectures [36, 30] represent the state-of-the-art in unconditional language modeling that perform impressive text generation [40] when trained on large unsupervised text corpora. CTRL [23] trained a similar architecture for generation by conditioning on properties of the text easily extracted at scale, e.g. domain, style, and associated URL.

3 Methods

Let $a = (a_1, \ldots, a_{n_a})$ be a sequence of amino acids—a protein. In protein engineering, there is typically a set of desired properties such as function or target organism. Following recent work on controllable, conditional language modeling [23], we refer to these properties generally as 'conditioning tags'. Let $c = (c_1, \ldots, c_{n_c})$ be a sequence of such conditioning tags and $x = [c; a]$ the sequence formed by prepending conditioning tags to an amino acid sequence. $p(x)$ is the probability over combined sequences of length $n = n_a + n_c$. We factorize this distribution using the chain rule of probability [9] and train a neural network with parameters θ to minimize negative log-likelihood over a dataset D where each sample sequence, x^k, has length n_k:

$$p(x) = \prod_{i=1}^{n} p(x_i | x_{<i}) \qquad \mathcal{L}(D) = -\frac{1}{|D|} \sum_{k=1}^{|D|} \frac{1}{n_k} \sum_{i=1}^{n_k} \log p_\theta(x_i^k | x_{<i}^k) \tag{1}$$

$p(a|c)$, the distribution over proteins conditioned on conditioning tags, is one of many conditional distributions that can be recovered from a model trained in this way. A new protein $\tilde{a}$ of length m_a with desired properties encoded by conditioning tag sequence $\tilde{c}$ of length m_c can be generated by sequentially sampling: $p_\theta(a_0|\tilde{c}), p_\theta(a_1|\tilde{a}_0, \tilde{c}), \ldots, p_\theta(a_p|\tilde{a}_{<p}, \tilde{c})$.

We train a Transformer [36] variant to learn these conditional distributions over amino acids and conditioning tags. A sequence containing n tokens is embedded as a sequence of n corresponding vectors in $\mathbb{R}^d$. This sequence of vectors is stacked into a matrix $X_0 \in \mathbb{R}^{n \times d}$ so that it can be processed by l attention layers. It then proceeds through pre-activation layer-normalized Transformer layers exactly as defined in Radford et al. [30] and Keskar et al. [23]. Only the embedding layers (determined by the training data) and hyperparameters differ between these architectures. Scores for each token in the vocabulary are computed from each output of the last layer. During training, these scores input to a cross-entropy loss. During generation, the scores for the final token in the sequence are normalized with a softmax to get a distribution for sampling a new token.

3.1 Data

We curate the protein sequences and associated tags available in Uniparc [24], UniprotKB [7], SWISS-PROT [6], TrEMBL [10], Pfam [8], and NCBI taxonomic information [16]. The aggregated dataset contains over 280M proteins—a comprehensive, non-redundant, annotated database. We followed the UniParc definition of non-redundancy, identical subsequences with varying context (previous amino acids) are non-redundant, as a generative model will significantly alter prediction of next tokens based on previous context, as shown below. For the amino acid vocabulary, we use the standard 25 amino acids designations in IUPAC [29]. The conditioning tags are divided into 2 categories: (1) keyword tags and (2) taxonomic tags. Following the definitions laid out in the UniprotKB controlled, hierarchical vocabulary of keywords (many of which are derived from Gene Ontology (GO) terms) [5], the conditioning keyword tags included 1100 terms ranging from cellular component, biological process, and molecular function terms. The taxonomic tags include 100k terms from the NCBI taxonomy across the eight standard taxonomic ranks. The aggregated dataset was split into a training set of size 280M, an out-of-distribution protein family[2] test set (OOD-test) of size 100k, and a randomly sampled, in-domain test set (ID-test) of size 1M. OOD-test contains 20 protein families, as defined in Pfam, that were entirely excluded from the training data. Performance on OOD-test measures ability to model samples from unseen protein families, whereas performance on ID-test measures ability to model samples from a wider range of protein families that more closely match the distribution of the training set as is quantitatively measured in the Supplemental Material.

3.2 Training and Generation

Training. For training, we include each sequence and its reverse[3], as the protein sequence itself is invariant to the temporal notion of sequence generation. We prepend each sequence with a corresponding subset of conditioning tags. For a given sequence, there can be multiple versions across databases, each with their own associated conditioning tags. We randomly sample which set of conditioning tags to utilize but bias toward SWISSPROT tags as they are manually verified. Additionally, we always include a sample with the sequence alone without conditioning tags so that ProGen can be used to complete proteins using sequence data alone. We truncate all sequences

[2] Protein families are groups of evolutionarily-related proteins that have similar structure, function, and sequence similarity as defined by Pfam [8]

[3] As ProGen is a unidirectional language model, the reversed sequence assists in data augmentation and learning dependencies from either direction.

to a maximum length of 512. Sequences of length less than 512 were padded, but no loss was backpropagated through the network for padding tokens. The model has dimension $d = 1028$, inner dimension $f = 512$, 36 layers, and 8 heads per layer. Our model was implemented in TensorFlow [1] and trained with a global batch size of 64 distributed across 256 cores of a Cloud TPU v3 Pod for 1M iterations. Training took approximately two weeks using Adagrad with linear warmup from 0 to $1e^{-2}$ over 40k steps. The model was initialized with pretrained weights of Keskar et al. [23].

Generation. ProGen generates proteins one amino acid at a time, by sampling from output probability distribution over amino acids using a context sequence, until a desired protein length is reached. We compare different combinations of top-k sampling [30] with a repetition penalty that reduces the probability of amino acids that have been generated within 4 prior tokens. We report results for top-k values of $k = 1$ and $k = 3$ with repetition penalties of 0 and 1.2.

3.3 Evaluation Details

We evaluate the model's generative quality through a series of computational-only means: from language modeling metrics to sequence- and structure-based calculations along with pertinent case studies. As opposed to low-throughput, resource-intensive wet lab experiments, our computational criterion provides a scalable framework for generative model evaluation and serves as a necessary precursor to future wet lab work to acquire gold-standard measurements. To assess how well ProGen models the training and test distributions, we rely on perplexity as the standard metric for language models, a mean hard accuracy over each token to strictly assess each amino acid error, and a mean soft accuracy defined by incorporating BLOSUM62 [18], a standard amino acid substitution matrix.

To assess the quality of generation, we evaluate across three levels of structure: (1) primary sequence similarity, (2) secondary structure accuracy, and (3) conformational energy analysis. Primary sequence similarity is defined by a length-normalized global, pairwise sequence alignment score. Secondary structure accuracy was computed per-residue for predicted secondary structures by PSIPRED[4] with greater than 0.5 confidence. Experiments reporting sequence similarity and secondary structure accuracy are limited to test samples with a form of experimental evidence of X-ray/NMR crystallography, mass spectrometry, or existence in cDNA or RT-PCR to indicate transcript existence. See discussion on UniprotKB existence scores[5] for details. Conformational energy uses the Rosetta-RelaxBB protocol[6], which performs a Monte Carlo optimization of the Rosetta energy function over the space of amino acid types and rotamers. Experiments that report conformational energy are limited to test samples from SWISSPROT with associated 3D structures in RCSB PDB.

To provide an intuition for generation quality across each metric, we provide baselines for different levels of random mutation. For a given sequence, a proportion $(25 - 100\%)$ of amino acids in the sequence is randomly substituted within one of the other 20 standard amino acids. For conformational energy, we also include an all-alanine baseline (i.e. a sequence with only the amino acid alanine), as it is a non-bulky, chemically-inert amino acid that mimics the existing secondary structure well when substituted. A particular random mutation may or may not have any effects on protein structure or function. But in aggregate, the performance of the 100% mutation baseline for any metric indicates failed generation. As performance approaches 0%, generation statistically indicates a closer reflection to desired structural and functional properties.

4 Results and Analysis

4.1 Evaluating ProGen as a language model

We first show that ProGen is a high-quality language model according to per-token metrics.

ProGen generalizes to the full test set and achieves perplexities representative of a high-quality language model. ProGen is significantly better than the baselines: a Uniform baseline, with amino acids sampled according to a uniform distribution, a third-order markov model derived from the training distribution, and UniRep [2], a state-of-the-art unidirectional language model (Table 1).

[4] http://bioinf.cs.ucl.ac.uk/psipred/

[5] https://www.uniprot.org/help/protein_existence

[6] https://www.rosettacommons.org/

Table 1: ProGen outperforms uniform, 3-order markov model, and UniRep [2] baselines on the full test set, which includes ID- and OOD-test. OOD-test results reveal that ProGen also performs well on protein families unseen during training. Fine-tuning ProGen dramatically improves performance over training from random initialization. PPL: Perplexity, HARD ACC: Hard Accuracy.

MODEL	UNIFORM BASELINE	3-ORDER MARKOV	UNIREP	PROGEN	ID-TEST	OOD-TEST	OOD-TEST-20 (RAND. INIT.)	OOD-TEST-20 (FINE-TUNED)
PPL	25	16.68	10.94	8.56	8.17	13.34	17.78	7.45
HARD ACC.	4	11	34	45	45	22	9	50

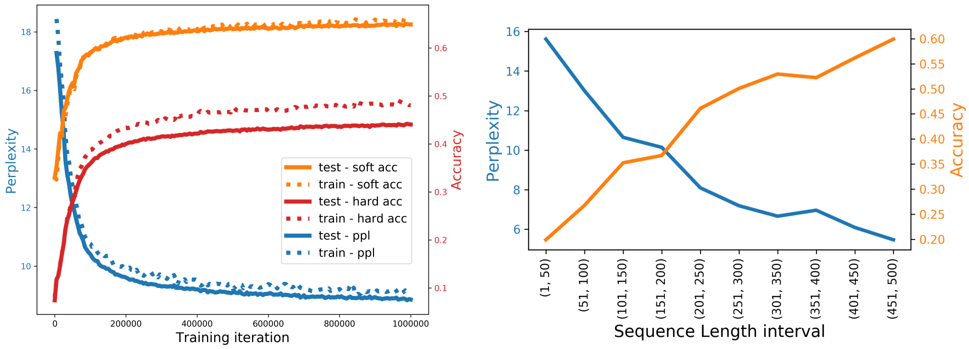


Figure 2: (Left) Large model capacity is warranted as ProGen has yet to overfit. BLOSUM62-informed soft accuracy shows no gap between train and test performance, suggesting hard accuracy hides the possibility that ProGen errors often correspond to amino acid substitutions found in nature. (Right) Full test set performance is better for later segments of sequences in keeping with intuition that additional context supports better predictions.

.

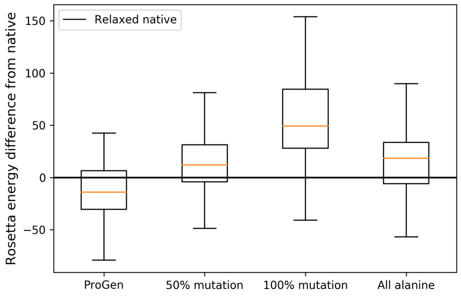
ProGen generalizes to unseen protein families. We break down ProGen's performance into perplexities over the ID-test and OOD-test sets separately (Table 1-second section). Results on ID-test confirm that ProGen generalizes well to sequences that belonged to randomly sampled protein families. As expected, ProGen performance drops for the difficult out-of-distribution sequences but still outperforms markov model performance on in-distribution samples.

Fine-tuning ProGen on unseen protein families improves over training from random initialization. We further split OOD-test into OOD-test-80 and OOD-test-20, finetune ProGen on OOD-test-80 until convergence (5 epochs; Adam; linear learning rate warmup to 1k iterations), and retest on OOD-test-20. Fine-tuning from ProGen improves over training the same architecture with randomly initialized weights (Table 1-third section).

Training curves suggest that protein generation would benefit from even larger models and longer training. With 1B parameters, ProGen is comparable in size to the largest language models that have been publicly released for any modality, and, to the best of our knowledge, it is the largest model trained on amino acid sequences. Figure 2-(Left) shows that despite its size and the amount of compute used to train, ProGen has yet to overfit the training data. This suggests that models for protein generation could still benefit from even larger models and additional compute.

ProGen performance improves with increased amino acid and conditioning tag context. In Figure 2-(Right), we examine the mean perplexity and per-token hard accuracy over different portions of proteins. Perplexity decreases and hard accuracy increases for later portions of a protein, in keeping with the intuition that additional amino acid context narrows down the possibilities for future tokens.

BLOSUM62 soft accuracy reveals that ProGen prediction errors often follow natural amino acid substitutions that likely conserve higher level structure. Though ProGen models proteins as pure sequences, protein function is more directly determined by the secondary and tertiary structures that these sequences encode in three-dimensional space. Model performance based on BLOSUM62 soft accuracy (Section 3.3) is more than 20% higher than using hard accuracy, which indicates



	20-60%	60-90%
	Max Identity	Max Identity
ProGen	**11.06**	**5.11**
50% Mutation	26.10	13.42
All Alanine	18.71	20.36
100% Mutation	54.81	59.72

Figure 3: (Left) Conformational energies for ProGen generated proteins surpasses all baselines and adheres closely to the energy of the native template. (Right) For test subsets that exhibit low similarity to the training data, ProGen still exhibits close to native energy as compared to baselines. Mean Rosetta energy difference from native are displayed binned by max identity of test sample to train set.

that when ProGen errors may often be substitutions that are acceptable in nature because they still reflect the proper higher-level properties. This suggests that ProGen has learned how to work within function-preserving mutational invariances. We continue to validate this finding in the next section.

4.2 Generating with ProGen

Generation quality is directly correlated with evolutionary viability and functional qualities, which can be inferred through protein structure. For this reason, we assess generation quality by using metrics for primary sequence similarity, secondary structure accuracy, and conformational energy (Section 3.3). We also include mutation baselines to compare the similarity of generated proteins to a target, reference protein across all metrics. In reference to these mutation baselines, ProGen quality improves as we move from primary sequence to full conformational structure metrics, thereby suggesting the model has learned mutational invariances in structure which present as errors in lower-level metrics.

ProGen achieves higher sequence similarity scores with an amino acid repetition penalty. In experiments with top-k sampling, ProGen performs best with $k = 1$ and the repetition penalty applied to recently generated amino acids, over all context lengths. Consequently, we use these settings for all following experiments. With this nearly greedy sampling, ProGen manages to generate proteins with sequence similarity comparable to randomly mutating 50% of the amino acids that are not seen in the given context.

Sequence similarity suggests that ProGen merely approaches the 25% mutation baseline, but secondary structure accuracy suggests that ProGen surpasses it. We analyze sequence similarity across differing numbers of conditioning tags. Sequences associated with at least 3 conditioning tags begin to exceed the 50% mutation baseline and sequences with at least 8 conditioning tags approach the 25% mutation baseline. Even in the best case, according to sequence similarity, ProGen doesn't surpass the 25% mutation baseline. By contrast, according to secondary structure accuracy, sequences with at least 8 conditioning tags surpass the 25% mutation baseline. This discrepancy between sequence similarity and secondary structure accuracy further corroborates our claim that errors registered by lower-level metrics often correspond to acceptable substitutions according to higher-level metrics that more directly correspond to functional viability.

After threading and relaxation, samples generated by ProGen are likely to exhibit desired structure and function. As a measure of generation quality, we thread ProGen sequences through known structures to examine if they exhibit favorable, low energy states. Figure 3-(Left) shows the differences between the energy levels of native proteins, ProGen samples, the native proteins with 50% and 100% of amino acids randomly mutated, and the all-alanine baseline. Proteins completed by ProGen are much closer to the energy levels of the native protein than all baselines, with energy levels near or even below their associated relaxed native templates.

Data splitting presents a limitation of our study as it likely led to highly similar sequences found across the train and test splits. To examine the interpolative (in-distribution) vs extrapolative (out-of-

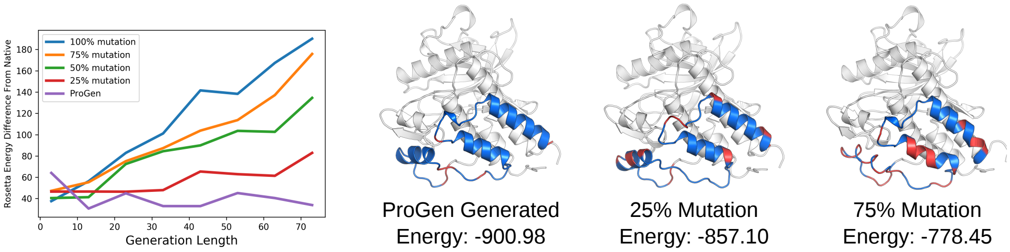


Figure 4: (Left) ProGen's VEGFR2 completion quality is near native conformational energy levels across generation lengths. (Right) ProGen makes fewer mistakes and prioritizes conservation of secondary structure as compared to baselines. Blue: Low energy (stable). Red: high energy (unstable).

distribution) generation ability, we examined performance based on increasingly more challenging test subsets in Figure 3-(Right). Each sequence is searched within the entire training database to find the maximum sequence identity. While performance degrades with decreasing sequence similarity as expected, ProGen still maintains a closer energy level to native proteins than all baselines.

4.3 Case Studies

Completing VEGFR2 kinase domain. We study how well ProGen generates in the context of a protein completion task using VEGFR2, which was excluded from training ProGen. We consider the amino acid sequence beginning at residue 806 and ending at residue 1168 of VEGFR2 (PDB ID: 2XIR). For different generation lengths, we sample from ProGen to complete the sequence up to residue 1168 with the remainder of the sequence provided as context. Figure 4-(Left) shows that the conformational energy calculated after threading and relaxation of ProGen samples is lower than baselines, indicating better structural conservation. Generation quality remains near the native relaxed protein independent of generation length. The generated samples exhibit a mean sequence identity of 73.1% with the native sequence. This correlates to a lower sequence identity than the 25% mutation baseline (74% identity) but with better Rosetta energies. This suggests meaningful deviation from the native protein while achieving the ultimate goal of preserving low energy. Figure 4-(Right) shows one sample from ProGen as well as one from each of the 25% and 75% mutation baselines. The ProGen sample exhibits lower energy overall, and energy is highest for amino acids that do not have secondary structure. This suggests that ProGen learned to prioritize the most structurally important segments of the protein.

Zero-shot fitness selection for protein GB1. The ultimate goal of protein engineering is to design *functional* proteins. Ideally, a generative model, such as ProGen, that has learned the distribution of evolutionarily-relevant proteins can directly generate high-fitness proteins. We examine the empirical fitness landscape of protein G domain B1 (GB1) binding to an antibody [38]. We would want the ability to generate GB1 proteins with high binding affinity and stability. The data includes 149,361 of a total 160,000 possible variants from NNK/NNS saturation mutagenesis at four positions known to interact epistatically. Reported fitness values correspond to a measure of both stability (i.e. the fraction of folded proteins) and function (i.e. binding affinity to IgG-Fc). Without supervised training of ProGen on the GB1 data or unsupervised fine-tuning of ProGen on a subset of similar immunoglobulin-binding proteins, we pass each variant through ProGen and select the top hundred variants with the lowest perplexity values. In Figure 5, we demonstrate ProGen is effective in zero-shot selection of high-fitness protein sequences. In comparison, random mutation, which is the main technique used by state-of-the-art methods such as directed evolution [4] and ML-assisted directed evolution [39], statistically generates samples

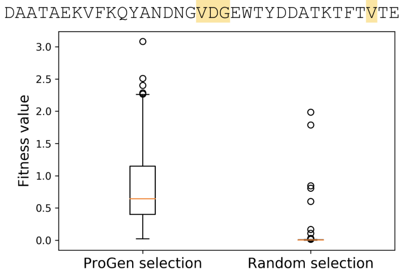


Figure 5: Without training on the GB1 dataset [38] and with no exposure to functional data, ProGen can identify protein variants with high fitness.

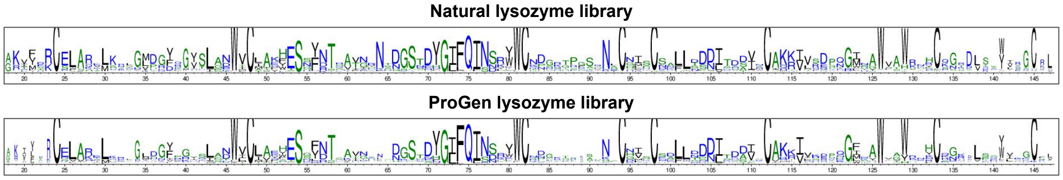


Figure 6: ProGen-generated lysozyme sequences resemble the diversity of lysozymes found in nature. The sequence logo [13] between a wild hen lysozyme multiple sequence alignment (MSA) resembles the sequence logo of a ProGen-generated lysozyme library.

with low or zero fitness. With effective sampling techniques, ProGen could be utilized to generate a spread of samples that are statistically high fitness. While further evidence (particularly in the wet lab) is required, these results may indicate that ProGen has not only learned the distribution of structurally-relevant proteins, but also functionally-relevant proteins.

Library construction of novel ProGen lysozyme sequences. To examine ProGen's ability to end-to-end generate a library of sequences that are not found in nature, we chose the lysozyme protein family, an examplar protein responsible for antibiotic defense. We selected the T4-bacteriophage (Uniprot: P00720) and wild hen (Uniprot:P00698) variants as reference proteins along with their respective multiple sequence alignments (MSAs). Each sequence is then prepended with a new conditioning tag, appended with a stop token, and padded to the maximum sequence length. We finetune ProGen for 5 epochs (Adam, $1e-4$ learning rate with linear warm-up). The finetuned model is used to generate protein sequences with only the conditioning tag for a wild hen lysozyme as input. Generation continued iteratively with top-k (k=1 to 6) greedy sampling until a stop token was predicted.

The library is then pruned to ensure no samples are found in the training datasets. The sequence logo of the ProGen generated library has high concordance with the sequence logo of the natural wild hen lysozyme (Figure 6), indicating that ProGen has learned the natural distribution of sequences by conserving important residues and maintaining relative frequencies. Next, we use the online server of GREMLIN [21] to extract a $HH\Delta$ score. It provides a quality metric where a value of 1 indicates no homolog with a known structure and a value of 0 indicates when the query and template alignments are identical. ProGen's generated lysozyme library demonstrates a $HH\Delta$ score of 0.15 to the PDB:2NWD lysozyme which is similar to a $HH\Delta$ of 0.10 for an MSA of natural wild hen lysozyme proteins. These results imply that ProGen may be utilized in novel discovery of previously-unseen proteins.

5 Conclusion

We introduced ProGen, a controllable protein generation language model trained on the full evolutionary diversity of a comprehensive sequence database. The model generates proteins that exhibit near native structure energies which likely implies functional viability. ProGen has the potential to play a new, complementary role alongside other state-of-the-art methods in protein engineering. For example, in directed evolution, initial sequences may be sampled from ProGen according to desired conditioning tags. In later rounds of evolution, protein completion with context for particular residue spans, or hotspots, may provide higher fitness samples. In *de novo* protein design, using ProGen with conditioning tags may allow for designing new proteins with existing folding motifs in new protein families or host organisms.

6 Broader Impact

We develop a language model that aims to generate structurally and functionally viable protein sequences to enable machine learning guided protein engineering. Proteins are responsible for almost all biological processes critical to life. They are workhorses that maintain our health and the generation of proteins can enable therapeutics development for human disease. Proteins are also widely used in industrial settings (e.g., to break down plastic waste and create laundry detergents). A tool that could controllably generate new functional proteins would have a transformative impact on advancing our understanding of science, curing a vast array of diseases, and cleaning our planet. There are uncertainties in the path forward; wet-lab testing and validation is necessary before ProGen-like algorithms could be commonplace in protein engineering and research. The resources and expertise needed to perform wet-lab experiments in addition to training of large-scale transformers adds an entry cost that may be palatable only to well-funded players, possibly contributing to inequity in how ProGen is extended in the future. Lastly, any technology that enables the discovery of new proteins faces the risk of being abused for nefarious purposes such as biological weapons. ProGen is no exception.

7 Acknowledgements

We would like to thank Melvin Gruesbeck for figure design, Subu Subramanian for discussions regarding lysozymes, Jesse Vig for visualizing the attention heads of ProGen.

References

[1] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al. Tensorflow: A system for large-scale machine learning. In *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)*, pp. 265–283, 2016.

[2] Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12): 1315–1322, 2019.

[3] Anand, N. and Huang, P. Generative modeling for protein structures. In *Advances in Neural Information Processing Systems*, pp. 7494–7505, 2018.

[4] Arnold, F. H. Design by directed evolution. *Accounts of chemical research*, 31(3):125–131, 1998.

[5] Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[6] Bairoch, A., Boeckmann, B., Ferro, S., and Gasteiger, E. Swiss-prot: juggling between evolution and stability. *Briefings in bioinformatics*, 5(1):39–55, 2004.

[7] Bairoch, A., Apweiler, R., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. The universal protein resource (uniprot). *Nucleic acids research*, 33(suppl_1):D154–D159, 2005.

[8] Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffiths-Jones, S., Khanna, A., Marshall, M., Moxon, S., Sonnhammer, E. L., et al. The pfam protein families database. *Nucleic acids research*, 32(suppl_1):D138–D141, 2004.

[9] Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

[10] Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.-C., Estreicher, A., Gasteiger, E., Martin, M. J., Michoud, K., O'Donovan, C., Phan, I., et al. The swiss-prot protein knowledgebase and its supplement trembl in 2003. *Nucleic acids research*, 31(1):365–370, 2003.

[11] Boomsma, W. and Frellsen, J. Spherical convolutions and their application in molecular modelling. In *Advances in Neural Information Processing Systems*, pp. 3433–3443, 2017.

[12] Costello, Z. and Martin, H. G. How to hallucinate functional proteins. *arXiv preprint arXiv:1903.00458*, 2019.

[13] Crooks, G. E., Hon, G., Chandonia, J.-M., and Brenner, S. E. Weblogo: a sequence logo generator. *Genome research*, 14(6):1188–1190, 2004.

[14] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[15] Elnaggar, A., Heinzinger, M., Dallago, C., and Rost, B. End-to-end multitask learning, from protein language to protein features without alignments. *bioRxiv*, pp. 864405, 2019.

[16] Federhen, S. The ncbi taxonomy database. *Nucleic acids research*, 40(D1):D136–D143, 2012.

[17] Greener, J. G., Moffat, L., and Jones, D. T. Design of metalloproteins and novel protein folds using variational autoencoders. *Scientific reports*, 8(1):1–12, 2018.

[18] Henikoff, S. and Henikoff, J. G. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[19] Huang, P.-S., Boyken, S. E., and Baker, D. The coming of age of de novo protein design. *Nature*, 537(7620):320–327, 2016.

[20] Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. In *Advances in Neural Information Processing Systems*, pp. 15794–15805, 2019.

[21] Kamisetty, H., Ovchinnikov, S., and Baker, D. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679, 2013.

[22] Karimi, M., Zhu, S., Cao, Y., and Shen, Y. De novo protein design for novel folds using guided conditional wasserstein generative adversarial networks (gcwgan). *bioRxiv*, pp. 769919, 2019.

[23] Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*, 2019.

[24] Leinonen, R., Diez, F. G., Binns, D., Fleischmann, W., Lopez, R., and Apweiler, R. Uniprot archive. *Bioinformatics*, 20(17):3236–3237, 2004.

[25] McCann, B., Bradbury, J., Xiong, C., and Socher, R. Learned in translation: Contextualized word vectors. In *Advances in Neural Information Processing Systems*, pp. 6294–6305, 2017.

[26] Needleman, S. B. and Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[27] O'Connell, J., Li, Z., Hanson, J., Heffernan, R., Lyons, J., Paliwal, K., Dehzangi, A., Yang, Y., and Zhou, Y. Spin2: Predicting sequence profiles from protein structures using deep neural networks. *Proteins: Structure, Function, and Bioinformatics*, 86(6):629–633, 2018.

[28] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[29] Pettit, L. D. and Powell, K. The iupac stability constants database. *Chemistry international*, 2006.

[30] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019.

[31] Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, P., Canny, J., Abbeel, P., and Song, Y. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pp. 9686–9698, 2019.

[32] Riesselman, A. J., Shin, J.-E., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., and Marks, D. S. Accelerating protein design using autoregressive generative models. *bioRxiv*, pp. 757252, 2019.

[33] Rives, A., Goyal, S., Meier, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., and Fergus, R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, pp. 622803, 2019.

[34] Strokach, A., Becerra, D., Corbi-Verge, C., Perez-Riba, A., and Kim, P. M. Fast and flexible protein design using deep graph neural networks. *Cell Systems*, 11(4):402–411, 2020.

[35] Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H., and Consortium, U. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[36] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 5998–6008. Curran Associates, Inc., 2017. URL `http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[37] Vig, J. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.

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

[39] Wu, Z., Kan, S. J., Lewis, R. D., Wittmann, B. J., and Arnold, F. H. Machine learning-assisted directed protein evolution with combinatorial libraries. *Proceedings of the National Academy of Sciences*, 116(18):8852–8858, 2019.

[40] Zellers, R., Holtzman, A., Rashkin, H., Bisk, Y., Farhadi, A., Roesner, F., and Choi, Y. Defending against neural fake news. *arXiv preprint arXiv:1905.12616*, 2019.

[41] Zimmermann, L., Stephens, A., Nam, S.-Z., Rau, D., Kübler, J., Lozajic, M., Gabler, F., Söding, J., Lupas, A. N., and Alva, V. A completely reimplemented mpi bioinformatics toolkit with a new hhpred server at its core. *Journal of molecular biology*, 430(15):2237–2243, 2018.

A Appendix

A.1 Measuring out-of-distribution

The objective of our work is to enable high-quality protein generation. To test the effectiveness of our trained model, we had two test subsets: ID-Test and OOD-Test. ID-Test is a random split of the non-redundant sample database and can be viewed as a typical *in-distribution* test set of held-out samples.

In contrast, OOD-Test represents an *out-of-distribution* set. OOD-Test consists samples that contained a matching sub-sequence residing in one of twenty Pfam protein families that were held out of Train and ID-Test.

	3-GRAM SAE	5-GRAM SAE
TRAIN AND ID-TEST	0.027	0.095
TRAIN AND OOD-TEST	0.399	1.112
ID-TEST AND OOD-TEST	0.387	1.104

Table 2: The training data and ID-Test data seem to be drawn from a similar distribution, but OOD-Test is markedly different from the others. SAE refers to the sum of absolute errors for normalized 3-gram and 5-gram histograms. If two histograms were entirely divergent, the SAE would yield a value of 2.

To quantify the out-of-distribution nature of OOD-Test, we computed a normalized histogram of 3-grams and 5-grams across samples in the Train, ID-Test, and OOD-Test datasets. The sum of absolute errors (SAE) was computed for a pair of histograms as shown in Table 2. Two normalized histograms that align perfectly would have an SAE of 0 and two normalized histograms that are completely divergent would have an SAE of 2. The results imply that the OOD-Test is drawn from a significantly different distribution.

The held-out protein families included PF18369, PF04680, PF17988, PF12325, PF03272, PF03938, PF17724, PF10696, PF11968, PF04153, PF06173, PF12378, PF04420, PF10841, PF06917, PF03492, PF06905, PF15340, PF17055, PF05318.

A.2 Evaluation metrics details

Perplexity is the exponentiated cross-entropy loss computed over each token in a dataset. Thus, high quality language models are expected to have low perplexities. Mean per-token hard accuracy over the tokens in a sequence judges a prediction incorrect for any amino acid that is not the ground truth. Mean per-token soft accuracy relies on BLOSUM62, a block substitution matrix that specifies which amino acid substitutions are more or less acceptable according to their frequency in known well-formed proteins. BLOSUM62 is widely used across adopted alignment software (e.g., BLAST[7]). Our mean per-token soft accuracy uses BLOSUM62 to penalize incorrect amino acid predictions (negative BLOSUM values) according to the frequency of that substitution in the matrix. In this way, if the substitution is likely in nature, soft accuracy penalizes the model less.

To assess the quality of generation, we evaluate across three levels of structure: (1) primary sequence similarity, (2) secondary structure accuracy, and (3) conformational energy analysis.

Primary sequence similarity is defined by a global, pairwise sequence alignment score computed with the Biopython package[8]. This score is based on the Needleman-Wunsch algorithm [26] informed by the BLOSUM62 substitution matrix. The resulting score is then normalized by the length of the protein.

Secondary structure accuracy was computed per-residue for predicted secondary structures by PSIPRED with greater than 0.5 confidence. PSI-BLAST was performed on each generated sample to

[7] https://blast.ncbi.nlm.nih.gov/Blast.cgi

[8] https://biopython.org/

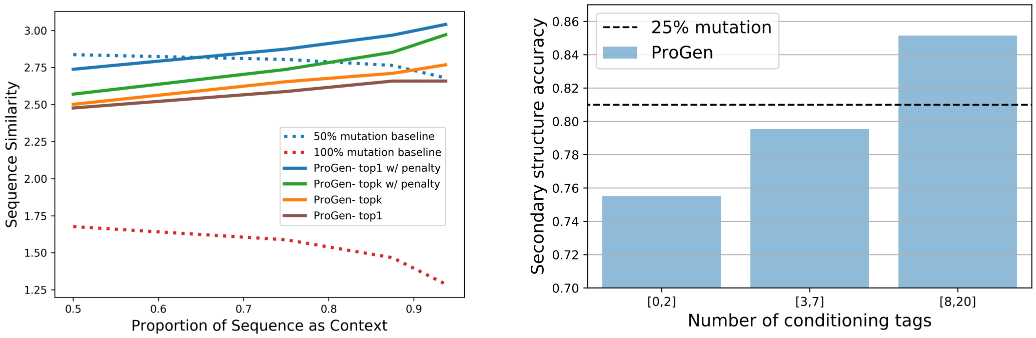


Figure 7: Generation quality improves as we move to higher level evaluation metrics. (Left) Sampling technique affects the generation quality. Among greedy techniques, an applied repetition penalty enables higher quality generation as determined by sequence similarity. (Right) ProGen generates sequences that conserve secondary structure. Sequences with increasing conditioning tags approach the 25% mutation baseline for secondary structure metrics.

extract the Multiple Sequence Alignments (MSAs) with respect to the UniRef90 database [35]. These MSAs were provided to PSIPRED for higher quality secondary structure prediction. Experiments reporting secondary structure accuracy were limited to test samples with high UniprotKB existence scores as described in the previous paragraph.

Conformational energy uses the Rosetta-RelaxBB protocol. Rosetta-RelaxBB performs a Monte Carlo optimization of the Rosetta energy function over the space of amino acid types and rotamers. The Rosetta energy is based on biophysical laws and constraints. Between each design round, amino acid side-chains are replaced, while the carbon backbone torsions are kept fixed. Energy minimization/relaxation is performed after threading the amino acid sequence through the known structure. This allows the backbone to move, possibly into a lower energy state. A lower resulting Rosetta energy correlates to a more relaxed-state and viable conformation for a given protein sequence. Before applying the procedure above, we relax the native template first. Experiments that report conformational energy are limited to test samples from SWISSPROT with associated 3D structures in RCSB PDB [9].

A.3 Generation quality as determined by primary sequence similarity and secondary structure accuracy

The percent random mutation baselines provide an intuitive barometer for generation quality. For differing sequence completion proportions as shown in Figure 7, we observe that ProGen's generation quality improves as we move toward higher level evaluation metrics.

The choice of sampling strategy can have meaningful effects on generation quality. We limit our study to two greedy sampling techniques: topK sampling to ensure sampling diversity and an applied repetition penalty to avoid characteristic mode collapse. It has been observed empirically in NLP and with ProGen that the model can fall into repeating patterns referred to as mode collapse. Based on sequence similarity results, we observe that the repetition penalty maintains the highest performance.

For secondary structure, we observe that performance approaches the 25% mutation baseline. While there are limitations to the underlying secondary structure prediction, the subsequent conformational energy analysis (which accounts for full 3D structure) does not present the same biases.

A.4 Influence of conditioning tags

In Figure 8, we observe how conditioning tags play a role in generative modeling. There seems to be a combination of two phenomenon present. First, the sample perplexity and per-token accuracy

[9] https://www.rcsb.org/

performance increases for sequences with more conditioning tags. This may be due to a correlation of manual curation of protein properties to sampling representation in protein databases. Second, the ablation study reveals that conditioning tags influence the generative model in predicting the initial amino acids. Once the model is conditioned with a sufficient number of amino acid tokens, the next-token prediction is more heavily influenced by previous amino acids than conditioning tags. Conditioning tags still provide ProGen with useful information and enable a form of control for sequence generation (as evidenced by Section A.5). Future work delving into more explicit forms of control with conditioning tags would be of great interest.

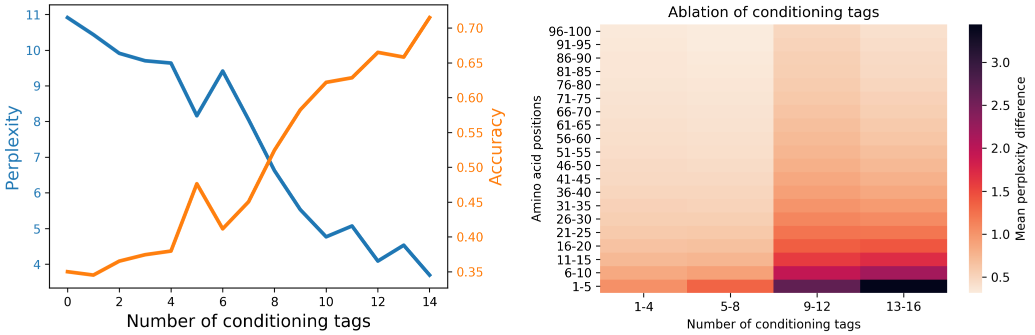


Figure 8: (Left) Test sequences with increasing conditioning tags demonstrate higher per-token performance. We examined proteins with up to 14 conditioning tags to ensure a minimum of 3k samples per category. (Right) Ablation of conditioning tags reveals tags influence the initial amino acids greatly. Prediction of later amino acid positions are more influenced by previous amino acid context as opposed to conditioning tags.

A.5 Generation with only conditioning tags

We observe that ProGen can be used to generate proteins with only conditioning tags and no initial amino acid context. For the following example, we prompt ProGen to greedily generate a protein sequence with the tags `Flavoprotein` and `FMN`. As defined by the UniprotKB keyword, the `FMN` tag refers to "a protein involved in flavin adenine mononucleotide (FMN) synthesis or protein which contains at least one FMN as prosthetic group/cofactor (flavoproteins) or cosubstrate, such as many oxidation-reduction enzymes".

The generated sequence of length 400 is then passed to the HHblits package by Zimmermann et al. [41] to search for a multiple sequence alignment (MSA). As shown in Figure 10, there are multiple sequences that align well with the ProGen sequence. Figures 11-13 demonstrate the alignments have high E-values and have related properties. The lower the E-value, the lower the probability of a random match and the higher the probability that the alignment match is related to the searched sequence.

A.6 Model visualizations

ProGen was trained from a randomly initialized embedding layer with no prior knowledge of residue biochemical properties. Through per-token training on millions of protein sequences, ProGen seems to have inherently learned the natural clustering of amino acids that align with our understanding of biophysicochemical properties. In Figure 9, the trained embedding weights for the standard amino acids tokens are reduced to three dimensions with principle component analysis (PCA).

Using Vig [37], we visualize the attention head patterns of ProGen. For both Figure 14 and Figure 15, we are visualizing the attention weight patterns in each head of ProGen for α-actinin protein (PDB: 4D1E) residues 510 to 528, which exhibits an alpha helical structure. In Figure 14, we visualize layers 1 to 3 and attention heads 1 to 12 of ProGen. The attention mechanism exhibits well-differentiated local and global patterns which may indicate specialization of each head on different tasks.

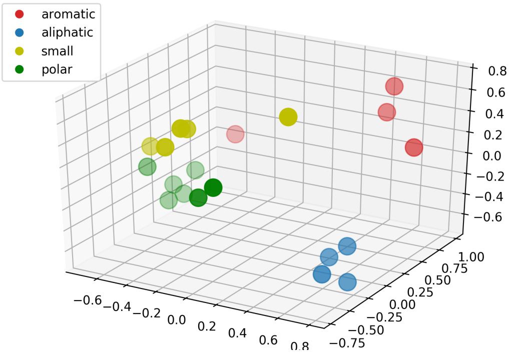


Figure 9: Principle component analysis (PCA) of the ProGen's amino acid embeddings aligns with our intuition of amino acid properties.

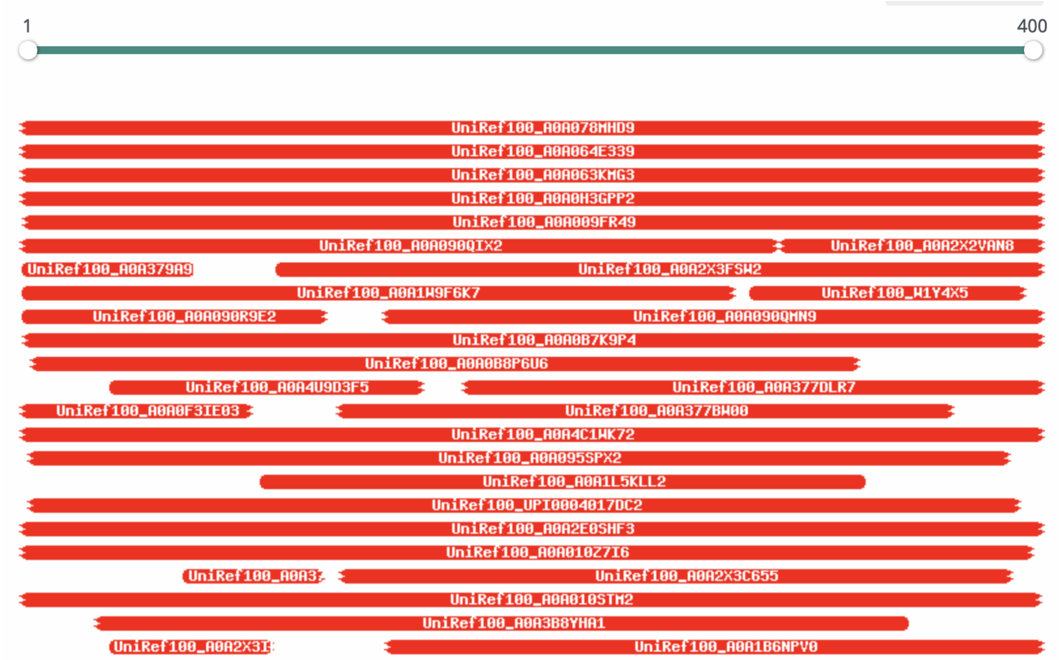
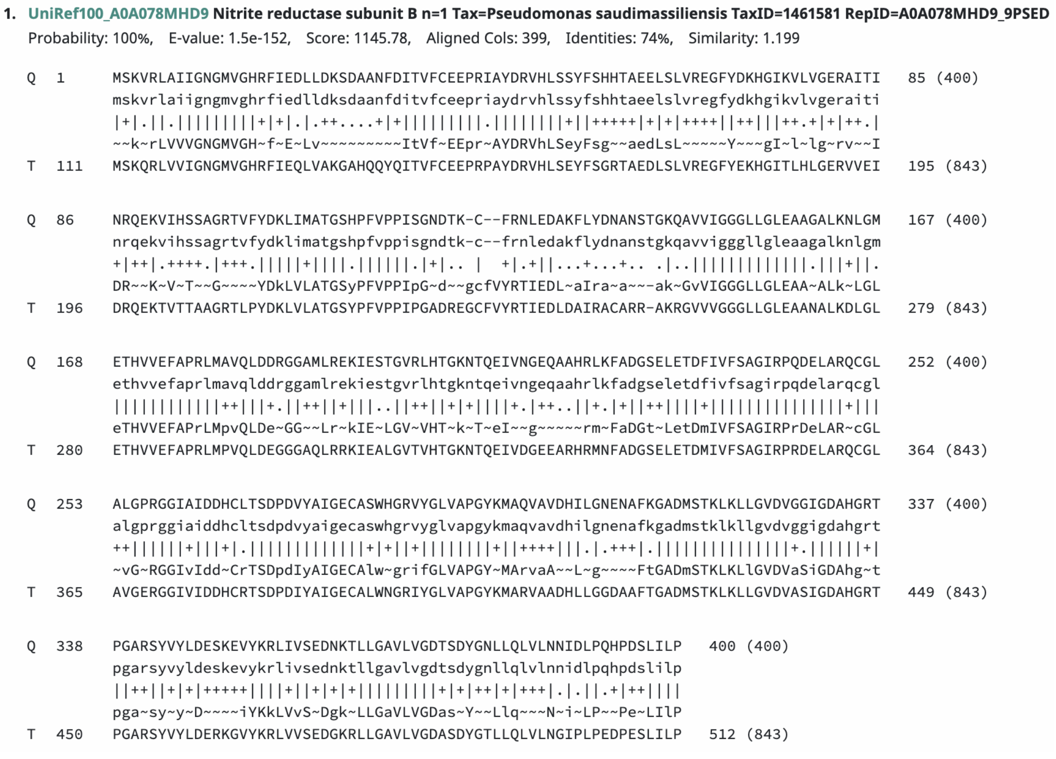
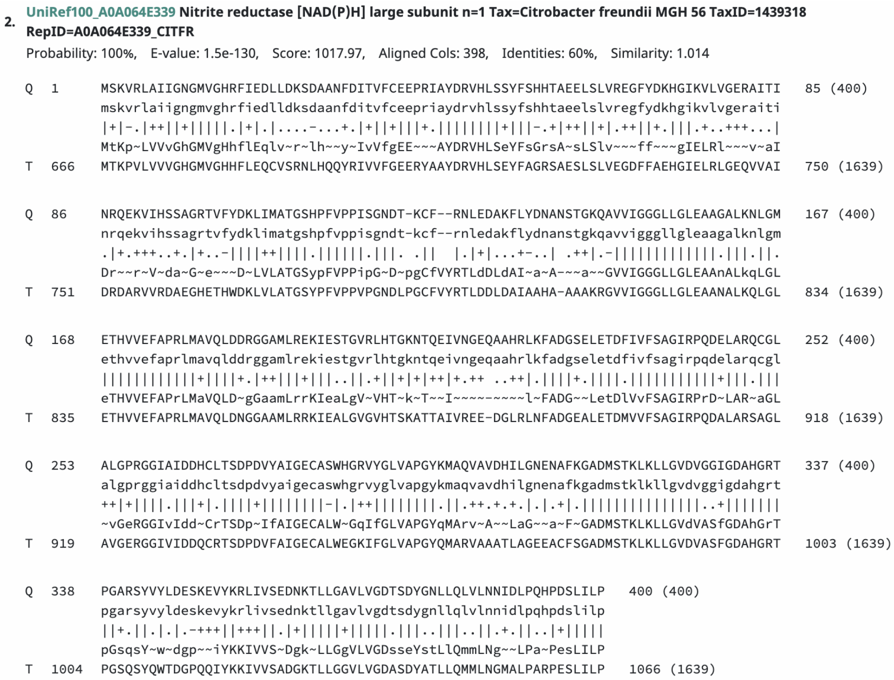


Figure 10: There are multiple sequences that align well with the ProGen generated FMN sequence from only conditioning tags. Many of the matching alignments have properties reflective of FMN proteins (e.g. oxidoreductases). A red color corresponds to a significantly low E-value, implying a matching homolog. The MSA was directly taken using HHblits.



```
1.  UniRef100_A0A078MHD9 Nitrite reductase subunit B n=1 Tax=Pseudomonas saudimassiliensis TaxID=1461581 RepID=A0A078MHD9_9PSED
    Probability: 100%,   E-value: 1.5e-152,   Score: 1145.78,   Aligned Cols: 399,   Identities: 74%,   Similarity: 1.199

  Q   1     MSKVRLAIIGNGMVGHRFIEDLLDKSDAANFDITVFCEEPRIAYDRVHLSSYFSHHTAEELSLVREGFYDKHGIKVLVGERAITI   85 (400)
            mskvrlaiigngmvghrfiedlldksdaanfditvfceepriaydrvhlssyfshhtaeelslvregfydkhgikvlvgeraiti
            |+|.||.|||||||||+|+|.|.++....+|+|||||||||.||||||||+||+++++|+|+|++++||++|||++.+|+|+++.|
            ~~k~rLVVVGNGMVGH~f~E~Lv~~~~~~~~~~ItVf~EEpr~AYDRVhLSeyFsg~~aedLsL~~~~~Y~~~gI~l~lg~rv~~I
  T   111   MSKQRLVVIGNGMVGHRFIEQLVAKGAHQQYQITVFCEEPRPAYDRVHLSEYFSGRTAEDLSLVREGFYEKHGITLHLGERVVEI   195 (843)


  Q   86    NRQEKVIHSSAGRTVFYDKLIMATGSHPFVPPISGNDTK-C--FRNLEDAKFLYDNANSTGKQAVVIGGGLLGLEAAGALKNLGM   167 (400)
            nrqekvihssagrtvfydklimatgshpfvppisgndtk-c--frnledakflydnanstgkqavviggglleaagalknlgm
            +|++|.++++.|+++.|||||+||||.||||||.|+|.. |  +|.+||...+...+.. .|..||||||||||||.|||+||.
            DR~~K~V~T~~G~~~~YDkLVLATGSyPFVPPIpG~d~~gcfVYRTIEDL~aIra~a~~-ak~GvVIGGGLLGLEAA~ALk~LGL
  T   196   DRQEKTVTTAAGRTLPYDKLVLATGSYPFVPPIPGADREGCFVYRTIEDLDAIRACARR-AKRGVVVGGGLLGLEAANALKDLGL   279 (843)


  Q   168   ETHVVEFAPRLMAVQLDDRGGAMLREKIESTGVRLHTGKNTQEIVNGEQAAHRLKFADGSELETDFIVFSAGIRPQDELARQCGL   252 (400)
            ethvvefaprlmavqlddrggamlrekiestgvrlhtgkntqeivngeqaahrlkfadgseletdfivfsagirpqdelarqcgl
            |||||||||||++|||+.||++||+|||..||++||+|+||||+.|++..||+.|+||++||||+||||||||||||||+|||
            eTHVVEFAPrLMpvQLDe~GG~~Lr~kIE~LGV~VHT~k~T~eI~~g~~~~~rm~FaDGt~LetDmIVFSAGIRPrDeLAR~cGL
  T   280   ETHVVEFAPRLMPVQLDEGGGAQLRRKIEALGVTVHTGKNTQEIVDGEEARHRMNFADGSELETDMIVFSAGIRPRDELARQCGL   364 (843)


  Q   253   ALGPRGGIAIDDHCLTSDPDVYAIGECASWHGRVYGLVAPGYKMAQVAVDHILGNENAFKGADMSTKLKLLGVDVGGIGDAHGRT   337 (400)
            algprggiaiddhcltsdpdvyaigecaswhgrvyglvapgykmaqvavdhilgnenafkgadmstklkllgvdvggigdahgrt
            ++||||||+|||+|.|||||||||||||+|+||+||||||||+||++++|||.|.+++|.||||||||||||||+.||||||+|
            ~vG~RGGIvIdd~CrTSDpdIyAIGECAlw~grifGLVAPGY~MArvaA~~L~g~~~~FtGADmSTKLKLlGVDVaSiGDAhg~t
  T   365   AVGERGGIVIDDHCRTSDPDIYAIGECALWNGRIYGLVAPGYKMARVAADHLLGGDAAFTGADMSTKLKLLGVDVASIGDAHGRT   449 (843)


  Q   338   PGARSYVYLDESKEVYKRLIVSEDNKTLLGAVLVGDTSDYGNLLQLVLNNIDLPQHPDSLILP   400 (400)
            pgarsyvyldeskevykrlivsednktllgavlvgdtsdygnllqlvlnnidlpqhpdslilp
            ||++||+|+|+++++|||+||+|+|+|||||||||+|+|++|+|+++|.|.||.+|++||||
            pga~sy~y~D~~~~iYKkLVvS~Dgk~LLGaVLVGDas~Y~~Llq~~~N~i~LP~~Pe~LIlP
  T   450   PGARSYVYLDERKGVYKRLVVSEDGKRLLGAVLVGDASDYGTLLQLVLNGIPLPEDPESLILP   512 (843)
```

Figure 11: First alignment (ranked by E-value) of a ProGen generated FMN protein. An E-value less than $1e^{-4}$ and identity greater than 40% is desired to consider the match as potentially homologous. The sequence labeled as Q is the ProGen protein and the sequence labeled as T is the matched sequence.

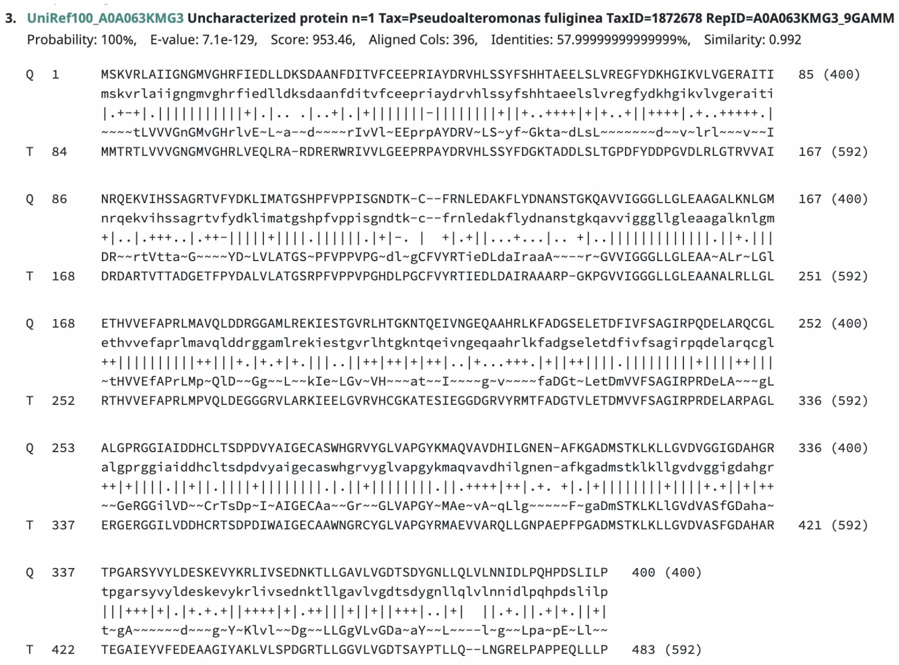


2. UniRef100_A0A064E339 Nitrite reductase [NAD(P)H] large subunit n=1 Tax=Citrobacter freundii MGH 56 TaxID=1439318 RepID=A0A064E339_CITFR

Probability: 100%, E-value: 1.5e-130, Score: 1017.97, Aligned Cols: 398, Identities: 60%, Similarity: 1.014

```
Q   1      MSKVRLAIIGNGMVGHRFIEDLLDKSDAANFDITVFCEEPRIAYDRVHLSSYFSHHTAEELSLVREGFYDKHGIKVLVGERAITI   85 (400)
           mskvrlaiigngmvghrfiedlldksdaanfditvfceepriaydrvhlssyfshhtaeelslvregfydkhgikvlvgeraiti
           |+|-.|++||+|||||.|+|.|....-...+.|+||+|||+.||||||||+|||-.+|++||+|.++||+.|||.+..+++...|
           MtKp~LVVvGhGMVgHhflEqlv~r~lh~~y~IvVfgEE~~~AYDRVHLSeYFsGrsA~sLSlv~~~ff~~~gIELRl~~~v~aI
T   666    MTKPVLVVVGHGMVGHHFLEQCVSRNLHQQYRIVVFGEERYAAYDRVHLSEYFAGRSAESLSLVEGDFFAEHGIELRLGEQVVAI   750 (1639)


Q   86     NRQEKVIHSSAGRTVFYDKLIMATGSHPFVPPISGNDT-KCF--RNLEDAKFLYDNANSTGKQAVVIGGGLLGLEAAGALKNLGM   167 (400)
           nrqekvihssagrtvfydklimatgshpfvppisgndt-kcf--rnledakflydnanstgkqavvigggllgleaagalknlgm
           .|+.+++..+.|+..-||||++||||.||||||.|||. .||  |.|+|...+-..| .++|.-||||||||||||.|||.||.
           Dr~~r~V~da~G~e~~~D~LVLATGSypFVPPipG~D~pgCfVYRTLdDLdAI~a~A~~~a~~GVVIGGGLLGLEAAnALkqLGL
T   751    DRDARVVRDAEGHETHWDKLVLATGSYPFVPPVPGNDLPGCFVYRTLDDLDAIAAHA-AAAKRGVVIGGGLLGLEAANALKQLGL   834 (1639)


Q   168    ETHVVEFAPRLMAVQLDDRGGAMLREKIESTGVRLHTGKNTQEIVNGEQAAHRLKFADGSELETDFIVFSAGIRPQDELARQCGL   252 (400)
           ethvvefaprlmavqlddrggamlrekiestgvrlhtgkntqeivngeqaahrlkfadgseletdfivfsagirpqdelarqcgl
           |||||||||||+||||+.|++|||+|||..||.+||+|+|++|+.++ ..++|.||||+.||||.|||||||||||+|||.|||
           eTHVVEFAPrLMaVQLD~gGaamLrrKIeaLgV~VHT~k~T~~I~~~~~~~~~l~FADG~~LetDlVvFSAGIRPrD~LAR~aGL
T   835    ETHVVEFAPRLMAVQLDNGGAAMLRRKIEALGVGVHTSKATTAIVREE-DGLRLNFADGEALETDMVVFSAGIRPQDALARSAGL   918 (1639)


Q   253    ALGPRGGIAIDDHCLTSDPDVYAIGECASWHGRVYGLVAPGYKMAQVAVDHILGNENAFKGADMSTKLKLLGVDVGGIGDAHGRT   337 (400)
           algprggiaiddhcltsdpdvyaigecaswhgrvyglvapgykmaqvavdhilgnenafkgadmstklkllgvdvggigdahgrt
           ++|+||||.|||+|.||||+||||||||-|.|++||||||||.||.++.+.+.|.|.+|.|||||||||||||||..+|||||||
           ~vGeRGGIvIdd~CrTSDp~IfAIGECALW~GqIfGLVAPGYqMArv~A~~LaG~~a~F~GADMSTKLKLLGVdVASfGDAhGrT
T   919    AVGERGGIVIDDQCRTSDPDVFAIGECALWEGKIFGLVAPGYQMARVAAATLAGEEACFSGADMSTKLKLLGVDVASFGDAHGRT   1003 (1639)


Q   338    PGARSYVYLDESKEVYKRLIVSEDNKTLLGAVLVGDTSDYGNLLQLVLNNIDLPQHPDSLILP   400 (400)
           pgarsyvyldeskevykrlivsednktllgavlvgdtsdygnllqlvlnnidlpqhpdslilp
           ||+.||.|.|.-+++||+++||.|+|||||+|||||.|||..|||..||.+.||..|+|||||
           pGsqsY~w~dgp~~iYKKIVVS~Dgk~LLGgVLVGDsseYstLlQmmLNg~~LPa~PesLILP
T   1004   PGSQSYQWTDGPQQIYKKIVVSADGKTLLGGVLVGDASDYATLLQMMLNGMALPARPESLILP   1066 (1639)
```

Figure 12: Second alignment (ranked by E-value) of a ProGen generated FMN protein. An E-value less than $1e^{-4}$ and identity greater than 40% is desired to consider the match as potentially homologous. The sequence labeled as Q is the ProGen protein and the sequence labeled as T is the matched sequence.



3. UniRef100_A0A063KMG3 Uncharacterized protein n=1 Tax=Pseudoalteromonas fuliginea TaxID=1872678 RepID=A0A063KMG3_9GAMM

Probability: 100%, E-value: 7.1e-129, Score: 953.46, Aligned Cols: 396, Identities: 57.99999999999999%, Similarity: 0.992

```
Q   1      MSKVRLAIIGNGMVGHRFIEDLLDKSDAANFDITVFCEEPRIAYDRVHLSSYFSHHTAEELSLVREGFYDKHGIKVLVGERAITI   85 (400)
           mskvrlaiigngmvghrfiedlldksdaanfditvfceepriaydrvhlssyfshhtaeelslvregfydkhgikvlvgeraiti
           |.+-+|.||||||||||+|.|.. .|..+|.|+||||||-||||||+||+..++++|+|+..+||++++|.+..+++++.|
           ~~~~tLVVVGnGMvGHrlvE~L~a-~d~~~~rIvVl~EEprpAYDRV~LS~yf~Gkta~dLsL~~~~~~~~d~~v~lrl~~~v~~I
T   84     MMTRTLVVVGNGMVGHRLVEQLRA-RDRERWRIVVLGEEPRPAYDRVHLSSYFDGKTADDLSLTGPDFYDDPGVDLRLGTRVVAI   167 (592)


Q   86     NRQEKVIHSSAGRTVFYDKLIMATGSHPFVPPISGNDTK-C--FRNLEDAKFLYDNANSTGKQAVVIGGGLLGLEAAGALKNLGM   167 (400)
           nrqekvihssagrtvfydklimatgshpfvppisgndtk-c--frnledakflydnanstgkqavvigggllgleaagalknlgm
           +|..|.+++..|.++-||||+||||.||||||.|+|-. |  +|.+||...+...|.. +|..|||||||||||||.||+.|||
           DR~~rtVtta~G~~~~YD~LVLATGS~PFVPPVPG~dl~gCFVYRTieDLdaIraaA~~-~r~GVVIGGGLLGLEAA~ALr~LGl
T   168    DRDARTVTTADGETFPYDALVLATGSRPFVPPVPGHDLPGCFVYRTIEDLDAIRAAARP-GKPGVVIGGGLLGLEAANALRLLGL   251 (592)


Q   168    ETHVVEFAPRLMAVQLDDRGGAMLREKIESTGVRLHTGKNTQEIVNGEQAAHRLKFADGSELETDFIVFSAGIRPQDELARQCGL   252 (400)
           ethvvefaprlmavqlddrggamlrekiestgvrlhtgkntqeivngeqaahrlkfadgseletdfivfsagirpqdelarqcgl
           ++|||||||||++|||+.|+.+|+.|||..||++|++|+|++|..|+...+++.|+||++||||.|||||||||+||||++|||
           ~tHVVEfAPrLMp~QlD~~Gg~~L~~kIe~LGv~VH~~~at~~I~~~~g~v~~~~~faDGt~LetDmVVFSAGIRPRDeLA~~~gL
T   252    RTHVVEFAPRLMPVQLDEGGGRVLARKIEELGVRVHCGKATESIEGGDGRVYRMTFADGTVLETDMVVFSAGIRPRDELARPAGL   336 (592)


Q   253    ALGPRGGIAIDDHCLTSDPDVYAIGECASWHGRVYGLVAPGYKMAQVAVDHILGNEN-AFKGADMSTKLKLLGVDVGGIGDAHGR   336 (400)
           algprggiaiddhcltsdpdvyaigecaswhgrvyglvapgykmaqvavdhilgnen-afkgadmstklkllgvdvggigdahgr
           ++|+||||.||+||.||||+||||||||.|.||+||||||||.||.++++|++|.+. +|.|+||||||||+||||+.+||+|++
           ~~GeRGGilVD~~CrTsDp~I~AIGECAa~~Gr~~GLVAPGY~MAe~vA~qLlg~~~~~F~gaDmSTKLKLlGVdVASfGDaha~
T   337    ERGERGGILVDDHCRTSDPDIWAIGECAAWNGRCYGLVAPGYRMAEVVARQLLGNPAEPFPGADMSTKLKLLGVDVASFGDAHAR   421 (592)


Q   337    TPGARSYVYLDESKEVYKRLIVSEDNKTLLGAVLVGDTSDYGNLLQLVLNNIDLPQHPDSLILP   400 (400)
           tpgarsyvyldeskevykrlivsednktllgavlvgdtsdygnllqlvlnnidlpqhpdslilp
           |||+++|+|.|+.+.+||++++|+|.++|||+||+||+++|..|+|  ||.+.||.+|+.||+|
           t~gA~~~~~~d~~~g~Y~Klvl~~Dg~~LLGgVLvGDa~aY~~L~~~-l~g~~Lpa~pE~Ll~~
T   422    TEGAIEYVFEDEAAGIYAKLVLSPDGRTLLGGVLVGDTSAYPTLLQ--LNGRELPAPPEQLLLP   483 (592)
```

Figure 13: Third alignment (ranked by E-value) of a ProGen generated FMN protein. An E-value less than $1e^{-4}$ and identity greater than 40% is desired to consider the match as potentially homologous. The sequence labeled as Q is the ProGen protein and the sequence labeled as T is the matched sequence.

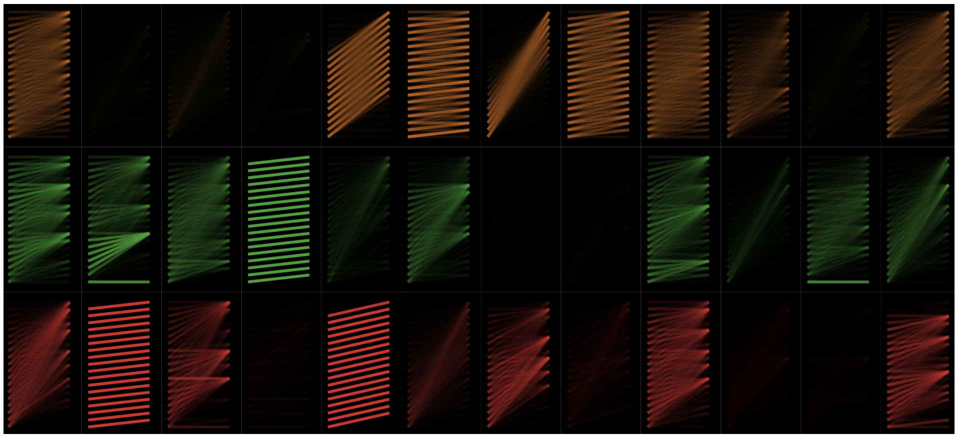


Figure 14: Attention patterns of ProGen for a given sequence. Layers 1-3 (rows) and attention heads 1-12 (columns) are displayed. The attention mechanism exhibits well-differentiated local and global patterns which may indicate specialization of each head on different tasks. Two corresponding attention heads from this visualization are shown in Figure 15.

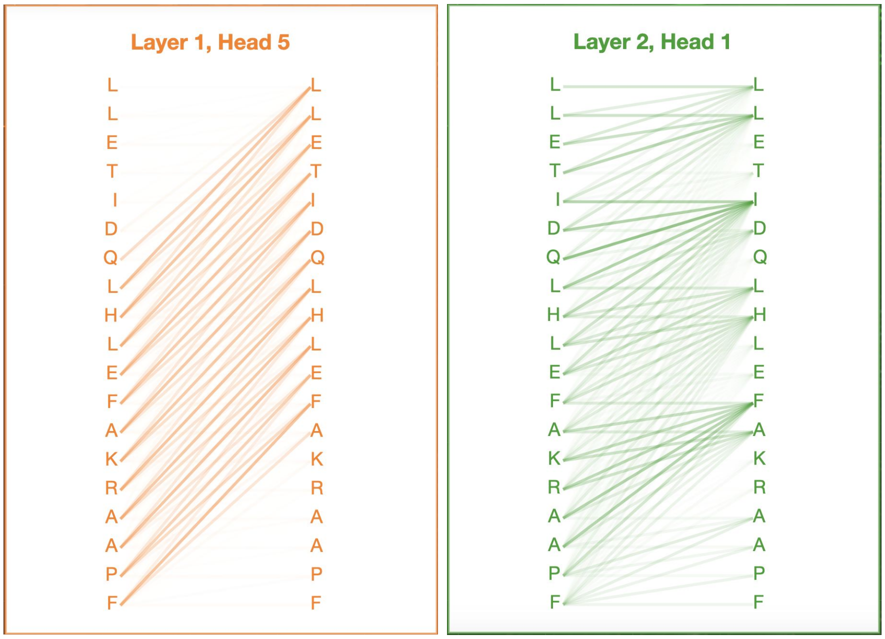

Layer 1, Head 5
L L E T I D Q L H L E F A K R A A P F
L L E T I D Q L H L E F A K R A A P F
Layer 2, Head 1
L L E T I D Q L H L E F A K R A A P F
L L E T I D Q L H L E F A K R A A P F

Figure 15: Local attention pattern for two example attention heads. Lines indicate attention to previous tokens for a given predicted token.