EVALUATING PROMPT TUNING FOR CONDITIONAL PROTEIN SEQUENCE GENERATION

Andrea Nathansen* **& Kevin Klein***
Hasso Plattner Institute, Digital Engineering Faculty
University of Potsdam
14482 Potsdam, Germany
{Andrea.Nathansen,Kevin.Klein}@student.hpi.de

Bernhard Y. Renard, Melania Nowicka† & Jakub M. Bartoszewicz†
Hasso Plattner Institute, Digital Engineering Faculty
University of Potsdam
14482 Potsdam, Germany
{Bernhard.Renard,Melania.Nowicka,Jakub.Bartoszewicz}@hpi.de

ABSTRACT

Text generation models originally developed for natural language processing have proven to be successful in generating protein sequences. These models are often finetuned for improved performance on more specific tasks, such as generation of proteins from families unseen in training. Considering the high computational cost of finetuning separate models for each downstream task, prompt tuning has been proposed as an alternative. However, no openly available implementation of this approach compatible with protein language models has been previously published. Thus, we adapt an open-source codebase designed for NLP models to build a pipeline for prompt tuning on protein sequence data, supporting the protein language models ProtGPT2 and RITA. We evaluate our implementation by learning prompts for conditional sampling of sequences belonging to a specific protein family. This results in improved performance compared to the base model. However, in the presented use case, we observe discrepancies between text-based evaluation and predicted biological properties of the generated sequences, identifying open problems for principled assessment of protein sequence generation quality.

1 INTRODUCTION

Protein design has been an important topic in biological research, motivated by high potential benefits from use cases as diverse as cancer immunotherapy (Silva et al., 2019), more effective antibodies against SARS-CoV-2 (Shan et al., 2022), and enzymatic degradation of plastic waste (Lu et al., 2022). Inspired by advances in natural language processing (NLP) methods, researchers have successfully developed large language models for protein sequence generation, such as ProtGPT2 (Ferruz et al., 2022) based on GPT-2 (Radford et al., 2019), RITA (Hesslow et al., 2022), ProGen (Madani et al., 2023), ProGen2 (Nijkamp et al., 2022), and ZymCTRL (Munsamy et al., 2022) for enzyme generation.

Training these models from the ground up requires large datasets and vast computational resources, but pretrained models can be finetuned on specific protein families or properties. Dhodapkar (2023) introduced SpikeGPT2, a finetuned ProtGPT2 model that generates SARS-CoV-2 spike protein sequences to predict potential future mutations. Madani et al. (2023) have also finetuned ProGen on protein families unseen during training. In those cases, one model is trained and stored per task. For large language models with sizes up to billions of parameters, this requires a large amount of

*Equal contribution.
†Equal contribution; corresponding authors.

storage. However, NLP models like GPT-3 (Brown et al., 2020) can perform well on specific tasks without model finetuning if given a task-specific prompt in addition to the input. Prompt engineering is a method to manually tune the (natural language) prompt in order to further improve the model's performance on the task (Brown et al., 2020). Recent work in NLP enhanced this approach by introducing P-tuning (Liu et al., 2021), a method for automatically tuning the embeddings for a given prompt. P-tuning preserves the structure of a human-designed prompt with regard to the positions of trainable, continuous tokens. To create prompts that are more agnostic to the natural language structure of the task description, Lester et al. (2021) proposed learning soft prompts as a set of extra embeddings that are prepended to the embedded input of the model. While still including task-specific tuning, prompt tuning reduces the storage requirements per task from a billion-parameter model to a prompt of ten thousand up to a few million parameters. Lester et al. (2022) have further worked on transferring tuned prompts across models.

Although prompt tuning is a promising direction in NLP, only a few approaches have been proposed to make it available for protein language models and evaluate its potential for improving protein sequence generation. ProGen and ZymCTRL allow specifying control tags for conditional generation (Madani et al., 2023; Munsamy et al., 2022), but the model has to be pretrained or finetuned to understand a defined set of control tags. In contrast, prompt tuning is a method to tune new task specifications without having to finetune the model. Hesslow et al. (2022) report having tested prompt tuning for RITA, but the authors have not made their code available yet, and provide only a brief description that does not allow complete reproducibility. Wang et al. (2023) proposed a prompt-aware transformer for different tasks such as function prediction and masked language modeling. However, they incorporate prompts in pretraining, so the method cannot be applied to existing unconditional protein language models. Lester et al. (2021) have released an implementation of their NLP prompt tuning approach, but it is not immediately applicable to protein language models like ProtGPT2 or RITA.

In our work, we adapt an open-source implementation of prompt tuning for natural language to learn prompts for conditional protein sequence generation. Our pipeline is compatible with ProtGPT2 and the RITA models. In order to reproduce the experiments conducted by Hesslow et al. (2022), we tune prompts for the RITA models on a held-out protein family and evaluate the performance in comparison to the base model. The results show an improvement in perplexity similar to the effect reported in Hesslow et al. (2022). However, we observe that the computationally predicted protein family membership matches the target family for only a few sequences. This discrepancy shows the necessity of more extensive evaluations, as well as the need for further research on representative metrics measuring protein sequence generation quality. Considering the benefits of low-cost, task-specific tuning, we see our work as a promising starting point for an openly available prompt tuning pipeline for pretrained protein language models, facilitating future studies in this direction. The source code, preprocessing scripts, model checkpoints and data splits used are available at `https://gitlab.com/dacs-hpi/protein-prompt-tuning`.

2 Methods

2.1 Prompt Tuning for sequence generation

Prompt tuning, as introduced by Lester et al. (2021), is an automated approach for tuning task-specific soft prompts that are prepended to a model's input at the embedding level. A soft prompt is a continuous matrix $\boldsymbol{P} \in \mathbb{R}^{m \times e}$, with m being the number of tokens of the soft prompt and e the dimension of the model's embeddings.

In a typical sequence generation step i, the model computes the probability for the next token $\boldsymbol{x}_i$ conditioned on the previous tokens $\{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{i-1}\}$. When applying a prompt $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_m\}$, the probability for token $\boldsymbol{x}_i$ is calculated from $\{\boldsymbol{p}_1, ..., \boldsymbol{p}_m, \boldsymbol{x}_1, ..., \boldsymbol{x}_{i-1}\}$. At the embedding level, the soft prompt $\boldsymbol{P}$ is prepended to the matrix $\boldsymbol{X} \in \mathbb{R}^{(i-1) \times e}$ consisting of the embedded previous tokens for step i. This results in an input matrix $\boldsymbol{X}_P \in \mathbb{R}^{(m+i-1) \times e}$ ($\boldsymbol{X}_P = [\boldsymbol{P}; \boldsymbol{X}]$) that is fed into the model. Soft prompts can be initialized in various ways, such as a concatenation of randomly sampled elements of the model's vocabulary or as a matrix filled with random numbers. During training, $\boldsymbol{P}$ receives gradient updates while $\boldsymbol{X}$ and the model's weights remain unchanged. The specific training objective and loss function is determined by the model implementation.

2.2 IMPLEMENTATION

Lester et al. (2021) published an implementation of prompt tuning[1] dependent on Jax, Flax, Flaxformer and T5X. However, we employ a codebase relying on Huggingface and PyTorch, as we deem it to be more easily accessible for the wider protein design community. We adapt the independently developed prompt tuning implementation mkultra[2], built for the NLP language models GPT-2 and GPT-Neo, to build a training pipeline for protein sequences supporting ProtGPT2. Additionally, we add support for the RITA language models to enable a direct comparison with Hesslow et al. (2022). Adaptation to more protein language models like ProGen (Madani et al., 2023) and ProGen2 (Nijkamp et al., 2022) is a subject of future improvement. The pipeline is implemented in Python 3.10 using PyTorch 1.13.1 and Huggingface transformers 4.20.1.

2.3 DATASET

Following Hesslow et al. (2022), we conduct our experiments on protein sequences of the Pfam family `PF03272` that was not included in the pretraining of the RITA models. As the specific dataset used in the experiments of Hesslow et al. (2022) is not available, we construct an analogous dataset consisting of the proteins associated with this family in InterPro[3] (Paysan-Lafosse et al., 2022). As in Hesslow et al. (2022), we preprocess our data by leaving out sequences containing `X` (uncharacterized amino acids), and replace the ambiguous characters `B`, `J` and `Z` with a random choice between `D` and `N`, `I` and `L`, and `E` and `Q`, respectively. We create training, validation and test splits at the ratio of $80:10:10$.

To avoid duplicate sequences between the splits, we cluster the dataset beforehand with MMseqs2 (Steinegger & Söding, 2017; 2018). For each cluster, we select the representative member determined by MMseqs2. Our main dataset is clustered at a 100% sequence similarity threshold to resemble the UniRef100 conditions. It contains 1536 sequences for training, 193 for validation and 193 for testing. We create further datasets by clustering at lower sequence identity thresholds: 95%, (611, 77, and 77 sequences for training, validation and test, respectively), 65% (283 / 36 / 36), and 35% (191 / 24 / 24). This allows us to evaluate the model's performance in generating novel protein sequences that are less similar to the sequences seen in training. On the other hand, the datasets become smaller for lower thresholds, which is likely to influence the results as well. More details about the clustering decisions are described in Appendix A.1.

2.4 EXPERIMENTAL SETUP

We evaluate our prompt tuning pipeline applied on the RITA models (Hesslow et al., 2022). RITA is an autoregressive decoder-only transformer, trained with the objective of minimizing the cross-entropy loss in next-token prediction. Hesslow et al. (2022) provide four models of different sizes: RITA-S (85M parameters, embedding size 768), RITA-M (300M / 1024), RITA-L (680M / 1536), and RITA-XL (1.2B / 2048) [4]. We choose RITA-S as the default model size. Similar to the experiments of Hesslow et al. (2022), we run further studies with RITA-M and RITA-L to examine the effect of prompt tuning in relation to model size. At the moment, we exclude RITA-XL due to its higher computational cost. While in this work we focus on the RITA model family, we also perform a proof-of-principle experiment using ProtGPT2 (see Appendix A.2).

Initializing the prompt weights from the model's vocabulary embeddings was shown to be superior to random uniform initialization from range $[-0.5, 0.5]$ for NLP models (Lester et al., 2021). We confirm this in a small ablation study (see Appendix A.3 and Table A1) and conduct our experiments with prompts sampled from the vocabulary. We set the prompt length to 10 tokens, as suggested by

[1] `https://github.com/google-research/prompt-tuning`

[2] `https://github.com/corolla-johnson/mkultra/commit/a25c72d47980a767b6178861a436900fd83c058f`

[3] `https://www.ebi.ac.uk/interpro/entry/pfam/PF03272/protein/UniProt/`, downloaded on January 5, 2023

[4] `https://huggingface.co/lightonai/RITA_s`, `https://huggingface.co/lightonai/RITA_m`, `https://huggingface.co/lightonai/RITA_l`, `https://huggingface.co/lightonai/RITA_xl`

the authors of Hesslow et al. (2022)[5]. We train the prompts using the Adam optimizer (Kingma & Ba, 2014) with a learning rate of 0.001. Due to limitations in GPU memory, we choose a batch size of 2 in order to maintain the same configuration for all model sizes. The GPU setup is described in Appendix A.4. Each training runs for a maximum of 300 epochs with early stopping after 20 epochs of no decrease in validation loss. We run each training three times with a different random seed for prompt initialization. The context size is 1024 (as in RITA pretraining (Hesslow et al., 2022)), split into 10 tokens for the prompt and 1014 tokens for the protein sequence. When evaluating the performance of the pretrained model without a prompt, we set the same maximum amount of sequence tokens per context, leading to a context size of 1014 in that case.

To compare sequence generation performance, we measure model perplexity, a common metric for language models equal to the exponential of the model's cross-entropy loss. Therefore, we expect perplexity to drop for models achieving a better fit to the target task. Hesslow et al. (2022) report perplexity per amino acid to enable comparison between models with different vocabularies. For RITA models, this is equivalent to perplexity per token, as RITA's vocabulary consists of one token per amino acid.

Perplexity provides a metric of text generation performance related to the probability of a given sequence, as estimated by the model. However, another level of validation from the biological perspective is needed. To test whether the produced sequences indeed exhibit the desired biological properties, we predict their family membership with ProtCNN (Bileschi et al., 2022) and measure the percentage of sequences classified as members of the desired family. We set the maximum number of generated amino acids to 1014, which is the model's context size excluding the prompt tokens. Note that ProtCNN was trained on Pfam seed sequences rather than whole proteins, so a sliding window approach must be used to call individual domains in the inputs used here (Bileschi et al., 2022). The length of the `PF03272` seed sequences, as available in InterPro, ranges from 112 to 121 amino acids. Therefore, we set a fixed window size of 120, with a stride of 10. Since Pfam family membership is dependent on the presence of specific domains, we count the whole protein sequence as belonging to the target family if ProtCNN predicts the family `PF03272` for at least one window within a sequence.

3 RESULTS

First, we aim to reproduce the results reported by Hesslow et al. (2022) by tuning prompts for RITA-S, RITA-M and RITA-L and comparing the perplexity of each prompt-tuned model to its untuned counterpart on the main test set (clustered at the sequence identity threshold of 100%). Similarly to the values presented in Table 4 of Hesslow et al. (2022), prompt tuning leads to a decrease in perplexity (Table 1). This also holds for a smaller experiment using ProtGPT2 (see Appendix A.2). Overall, larger models perform better, and the large prompt-tuned model has the lowest perplexity among all tested models.

Next, we tune prompts for datasets clustered with different sequence identity thresholds (Appendix A.1) to evaluate the performance of RITA-S for larger differences between training and test data (Table 2). As expected, the perplexity of the base model does not fluctuate much between the test sets. The slight differences can be explained by the fact that the clustered datasets contain differ-

[5] `https://github.com/lightonai/RITA/issues/3#issuecomment-1131381833`

Table 1: Perplexity comparison between prompt-tuned model and base model (Hesslow et al., 2022) for different model sizes (small, medium and large), measured on the main test set (clustered at the sequence identity threshold of 100%). For each prompt-tuned model, we report the mean and standard deviation over the three training runs.

	RITA Small	**RITA Medium**	**RITA Large**
Prompt-tuned model (ours)	8.61 ± 0.09	7.28 ± 0.05	6.3 ± 0.44
Base model	14.01	12.38	9.7

Table 2: Perplexity comparison between prompt-tuned model and base model for RITA-S on datasets clustered with different sequence identity thresholds (100%, 95%, 65% and 35%), measured on the respective test set. For each prompt-tuned model, we report mean and standard deviation over the three training runs.

	100	**95**	**65**	**35**
Prompt-tuned model (ours)	8.61 ± 0.09	9.78 ± 0.05	12.71 ± 0.06	13.39 ± 0.02
Base model	14.01	13.04	13.9	14.28

Table 3: Percentage of generated sequences classified as belonging to `PF03272`, compared between the prompt-tuned model and the base model (Hesslow et al., 2022) for different model sizes (small, medium and large). For each prompt-tuned model, we report mean and standard deviation over the three training runs.

	RITA Small	**RITA Medium**	**RITA Large**
Prompt-tuned model (ours)	2.9 ± 0.9	2.6 ± 0.7	1.2 ± 0.2
Base model	0.0	0.0	1.0

Table 4: Percentage of generated sequences classified as belonging to `PF03272`, compared for tuning a prompt on datasets clustered with different sequence identity thresholds (100%, 95%, 65% and 35%). We report the mean and standard deviation each over the three training runs. Base model performance (see RITA-S base model in Table 3) is not influenced by the datasets and therefore not shown.

	100	**95**	**65**	**35**
Prompt-tuned model (ours)	2.9 ± 0.9	2.8 ± 1.3	2.2 ± 1.1	1.9 ± 0.6

ent samples from the original dataset. The prompt-tuned models show consistent improvement over the base models, although the performance gain decreases noticeably for lower clustering thresholds. This could indicate that learning a prompt successfully generalizing to novel data becomes more challenging with decreased similarity to the training set. However, as the perplexity on the training set follows a similar trend (Table A2), smaller training set sizes for lower thresholds could have influenced the overall fit of the models as well.

ProtCNN (Bileschi et al., 2022) correctly identifies 80.8% of the sequences in the main test set as belonging to the protein family `PF03272`. This estimation of ProtCNN performance on natural sequences enables further comparisons with the sequences generated by both prompt-tuned and untuned models. For each RITA model size, we generate 193 sequences with a prompt tuned on the main dataset, and further 193 without a prompt. Strikingly, among all prompt-tuned models, only 1%-3% of the generated sequences are classified as members of the `PF03272` family (Table 3 and Table 4). We observe that for this use case the perplexity metric may imply a higher suitability of the models than we can confirm on the selected biological data. While prompt tuning allows generating at least some sequences with target biological properties, overall recovery rates are extremely small. In stark contrast with the perplexity evaluation, larger models do not show further improvement.

4 Discussion

We observe a decrease in perplexity for the prompt-tuned models. While reproducing the exact setup used by Hesslow et al. (2022) is not possible as the specific design choices, code, and data have not

been made publicly available at the time of writing, our results show improvements consistent with those reported. Hence, we have shown that prompt tuning can be used to tune RITA to improve perplexity for a given protein family. We share the pipeline with the wider research community for easy reproducibility and reuse in other protein generation tasks. Further, we plan to extend our approach to be compatible with other protein language models, such as ProGen (Madani et al., 2023) and ProGen2 (Nijkamp et al., 2022). Future work could also include optimization of continuous, rather than categorical properties, recycling prompts between different protein language models (Lester et al., 2022), or combining prompts learned for multiple optimization goals.

We also show that while untuned RITA-S and RITA-M models generate no sequences recognized as members of the target family, prompt tuning enables the design of at least some potential candidates. However, only a small fraction of the sequences contains the desired domains, and the effect is not noticeable for the largest of the evaluated architectures. This shows that the effects of prompt tuning on the predicted biological properties of designed proteins are more complex than suggested by the perplexity analysis alone. The gap between the text-based metrics and the biologically-informed evaluation highlights the need for more research on assessing the quality of conditional sampling from protein language models. Low numbers of promising candidates could indicate problems with the prompt tuning procedure or the RITA models themselves for this use case. However, it is also possible that the observed effect is an artifact of the domain-calling procedure used – low recall could be caused by unsuccessful generalization of ProtCNN to synthetic sequences, the choice of the sliding window width, or lower performance on this particular family. Therefore, biological properties of the generated sequences should be evaluated more comprehensively in future work, including extended benchmarking on more target families and functions. A comparison to analogous finetuned models would allow estimating the effects of a particular tuning procedure on both text-based and biology-informed metrics.

5 Conclusion

We present a prompt tuning pipeline for the protein generation models ProtGPT2 (Ferruz et al., 2022) and RITA (Hesslow et al., 2022), adapted from a prompt tuning codebase for NLP. Our results show that learning a prompt on a specific protein family improves perplexity, which is consistent with the results reported by Hesslow et al. (2022). Although this approach can also improve the rates of predicted target family membership for the designed proteins, desired domains can be recognized in only a small fraction of the generated sequences. For this use case, higher performance in terms of perplexity alone reported by Hesslow et al. (2022) does not necessarily translate to the desired gain of biological properties. The full scope and consequences of the observed discrepancy should be investigated further to better characterize the generative capabilities of large protein language models.

Acknowledgments

This work was supported by the de.NBI Cloud within the German Network for Bioinformatics Infrastructure (de.NBI) funded by Bundesministerium für Bildung und Forschung [031A537B, 031A533A, 031A538A, 031A533B, 031A535A, 031A537C, 031A534A, 031A532B].

References

Maxwell L. Bileschi, David Belanger, Drew H. Bryant, Theo Sanderson, Brandon Carter, D. Sculley, Alex Bateman, Mark A. DePristo, and Lucy J. Colwell. Using deep learning to annotate the protein universe. *Nature Biotechnology*, 40(6):932–937, Jun 2022. ISSN 1546-1696. doi: 10.1038/s41587-021-01179-w. URL https://doi.org/10.1038/s41587-021-01179-w.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris 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. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Rahul M. Dhodapkar. A deep generative model of the sars-cov-2 spike protein predicts future variants. *bioRxiv*, 2023. doi: 10.1101/2023.01.17.524472. URL https://www.biorxiv.org/content/early/2023/01/18/2023.01.17.524472.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. Protgpt2 is a deep unsupervised language model for protein design. *Nature Communications*, 13(1):4348, Jul 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-32007-7. URL https://doi.org/10.1038/s41467-022-32007-7.

Daniel Hesslow, Niccoló Zanichelli, Pascal Notin, Iacopo Poli, and Debora Marks. Rita: a study on scaling up generative protein sequence models, 2022. URL https://arxiv.org/abs/2205.05789.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL https://arxiv.org/abs/1412.6980.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243.

Brian Lester, Joshua Yurtsever, Siamak Shakeri, and Noah Constant. Reducing retraining by recycling parameter-efficient prompts, 2022. URL https://arxiv.org/abs/2208.05577.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too, 2021. URL https://arxiv.org/abs/2103.10385.

Hongyuan Lu, Daniel J. Diaz, Natalie J. Czarnecki, Congzhi Zhu, Wantae Kim, Raghav Shroff, Daniel J. Acosta, Bradley R. Alexander, Hannah O. Cole, Yan Zhang, Nathaniel A. Lynd, Andrew D. Ellington, and Hal S. Alper. Machine learning-aided engineering of hydrolases for pet depolymerization. *Nature*, 604(7907):662–667, Apr 2022. ISSN 1476-4687. doi: 10.1038/s41586-022-04599-z. URL https://doi.org/10.1038/s41586-022-04599-z.

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. Large language models generate functional protein sequences across diverse families. *Nature Biotechnology*, Jan 2023. ISSN 1546-1696. doi: 10.1038/s41587-022-01618-2. URL https://doi.org/10.1038/s41587-022-01618-2.

Geraldene Munsamy, Sebastian Lindner, Philipp Lorenz, and Noelia Ferruz. Zymctrl: a conditional language model for the controllable generation of artificial enzymes. *NeurIPS Machine Learning in Structural Biology Workshop*, 2022.

Erik Nijkamp, Jeffrey Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. Progen2: Exploring the boundaries of protein language models, 2022. URL https://arxiv.org/abs/2206.13517.

Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, Julian Gough, Daniel H Haft, Ivica Letunić, Aron Marchler-Bauer, Huaiyu Mi, Darren A Natale, Christine A Orengo, Arun P Pandurangan, Catherine Rivoire, Christian J A Sigrist, Ian Sillitoe, Narmada Thanki, Paul D Thomas, Silvio C E Tosatto, Cathy H Wu, and Alex Bateman. InterPro in 2022. *Nucleic Acids Research*, 51(D1):D418–D427, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac993. URL https://doi.org/10.1093/nar/gkac993.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Sisi Shan, Shitong Luo, Ziqing Yang, Junxian Hong, Yufeng Su, Fan Ding, Lili Fu, Chenyu Li, Peng Chen, Jianzhu Ma, Xuanling Shi, Qi Zhang, Bonnie Berger, Linqi Zhang, and Jian Peng. Deep learning guided optimization of human antibody against SARS-CoV-2 variants with broad neutralization. *Proceedings of the National Academy of Sciences*, 119(11):e2122954119, March 2022. doi: 10.1073/pnas.2122954119. URL https://www.pnas.org/doi/10.1073/pnas.2122954119. Publisher: Proceedings of the National Academy of Sciences.

Daniel-Adriano Silva, Shawn Yu, Umut Y. Ulge, Jamie B. Spangler, Kevin M. Jude, Carlos Labão-Almeida, Lestat R. Ali, Alfredo Quijano-Rubio, Mikel Ruterbusch, Isabel Leung, Tamara Biary, Stephanie J. Crowley, Enrique Marcos, Carl D. Walkey, Brian D. Weitzner, Fátima Pardo-Avila, Javier Castellanos, Lauren Carter, Lance Stewart, Stanley R. Riddell, Marion Pepper, Gonçalo J. L. Bernardes, Michael Dougan, K. Christopher Garcia, and David Baker. De novo design of potent and selective mimics of il-2 and il-15. *Nature*, 565(7738):186–191, Jan 2019. ISSN 1476-4687. doi: 10.1038/s41586-018-0830-7. URL https://doi.org/10.1038/s41586-018-0830-7.

Martin Steinegger and Johannes Söding. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology*, 35(11):1026–1028, Nov 2017. ISSN 1546-1696. doi: 10.1038/nbt.3988. URL https://doi.org/10.1038/nbt.3988.

Martin Steinegger and Johannes Söding. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9(1):2542, Jun 2018. ISSN 2041-1723. doi: 10.1038/s41467-018-04964-5. URL https://doi.org/10.1038/s41467-018-04964-5.

Zeyuan Wang, Qiang Zhang, Shuang-Wei HU, Haoran Yu, Xurui Jin, Zhichen Gong, and Huajun Chen. Multi-level protein structure pre-training via prompt learning. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=XGagtiJ8XC.

A Appendix

A.1 Clustering the dataset

The PF03272 dataset consists of all protein sequences assigned to this family deposited in InterPro. Here, we address the possibility of duplicate sequences in the data, as well as investigate the effects of similarities between the sequences within a single family. We employ MMSeqs2 (Steinegger & Söding, 2017; 2018) to cluster the raw dataset by sequence identity, setting the thresholds between 0% and 100% with a step size of 1%. The number of clusters and the size of the largest cluster for different thresholds are plotted in Figure A1. As expected, the number of clusters drops for a lower threshold, which is particularly visible between thresholds 100% and 90%. Thus, to avoid a radical reduction in dataset size, we employ a threshold of 100% for the data used in the main experiments to remove duplicates while retaining a substantial amount of sequences. We additionally create datasets with thresholds of 35%, 65% and 95% to assess the influence of sequence similarity on the model's performance. Due to high discrepancies between the cluster sizes, we select the representative member determined by MMseqs2 to form new, balanced datasets. As figure A1 shows, at the thresholds of 35%, the size of the largest cluster drops from 489 to 329, while the number of clusters barely increases. A similar drop happens at 95%, although to a smaller extent. Threshold 65% corresponds to the mean of 35% and 95%.

A.2 Prompt tuning for ProtGPT2

To showcase the compatibility of our pipeline with ProtGPT2, we conduct a small-scale prompt tuning experiment on the main dataset (clustered at the sequence identity threshold of 100%). Note that the sequences of this dataset might not have been withheld from ProtGPT2 pretraining. However, we deem this setup sufficient for a proof-of-principle. As described in Ferruz et al. (2022), we set a context size of 512 (10 prompt tokens, 502 protein sequence tokens). We use the Adam (Kingma & Ba, 2014) optimizer with a learning rate of 0.0001. The batch size is set to 2, and the prompt size to 10 tokens. The prompt is initialized from the model's vocabulary embeddings. We run training for a single random seed for 100 epochs with the patience of 20 epochs for early stopping.

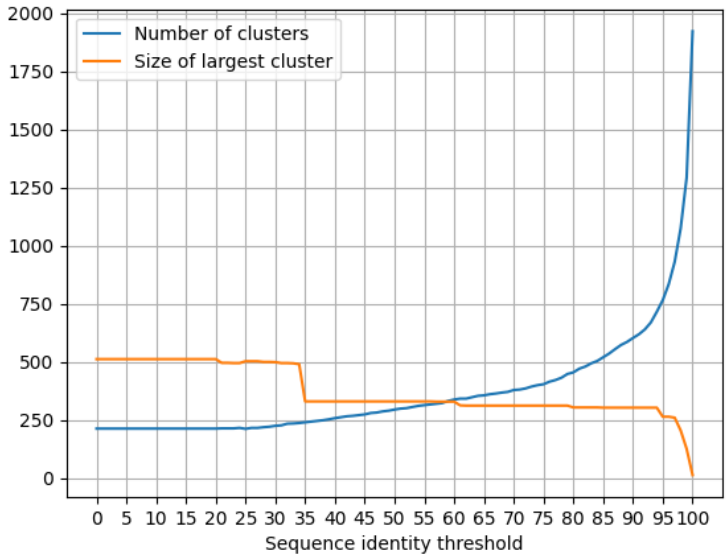

Figure A1: Number of clusters and size of the largest cluster for different sequence similarity thresholds.

Table A1: Perplexity comparison between soft prompts with random uniform initialization and soft prompts sampled from the model's vocabulary for different model sizes (small and medium), measured on the main test set.

	RITA Small	**RITA Medium**
Sampled Vocab	8.11 ± 0.08	6.81 ± 0.03
Random Uniform	8.53 ± 0.6	7.07 ± 0.57

Results of this training run show a decrease in perplexity to 1529.71, compared to 2999.05 for the base model. Note that these numbers are not comparable to those reported for the RITA models due to differences in perplexity computation. ProtGPT2 uses a different vocabulary where most tokens consist of more than one amino acid, whereas in RITA, each token corresponds to exactly one amino acid.

A.3 Ablation study results

To examine whether the findings of Lester et al. (2021) regarding soft prompt initialization hold true for protein language models, we evaluate the performance of soft prompts with random uniform initialization compared to soft prompts sampled from the model's vocabulary in a small ablation study. We report the perplexity on the main test set for RITA-S and RITA-M with the different initialization methods in Table A1. The results back the discovery that sampled embeddings form better starting points for soft prompts than randomly generated vectors.

A.4 GPU setup

Each training and evaluation run was conducted on one GPU at a time. We used a set of NVIDIA A100 GPUs with 40 and 80GB memory for training, which took about 10h, 30h and 40h per run for RITA-S, RITA-M and RITA-L, respectively. Perplexity was evaluated on the same set of GPUs, with an exception of some model S runs, where we used an NVIDIA Tesla T4 with 16 GB memory.

Table A2: Perplexity comparison for the prompt-tuned RITA-S model on datasets clustered with different sequence identity thresholds (100%, 95%, 65% and 35%), measured on the respective training, validation and test sets. For each prompt-tuned model, we report mean and standard deviation over the three training runs.

	100	**95**	**65**	**35**
Train	8.17 ± 0.1	10.27 ± 0.05	11.76 ± 0.08	11.59 ± 0.03
Validation	8.11 ± 0.08	9.79 ± 0.06	13.15 ± 0.14	12.12 ± 0.02
Test	8.61 ± 0.09	9.78 ± 0.05	12.71 ± 0.06	13.39 ± 0.02

ProtCNN inference and generation of sequences evaluated with ProtCNN were run on NVIDIA Tesla V100 GPUs with 16GB memory. Prompt tuning for ProtGPT2 was conducted and evaluated on the NVIDIA Tesla V100 GPUs.