RITA: a Study on Scaling Up Generative Protein Sequence Models

Daniel Hesslow [1] **Niccolò Zanichelli** [1] **Pascal Notin** [2] **Iacopo Poli** [1] **Debora Marks** [3]

Abstract

In this work we introduce RITA: a suite of autoregressive generative models for protein sequences, with up to 1.2 billion parameters, trained on over 280 million protein sequences belonging to the UniRef-100 database. Such generative models hold the promise of greatly accelerating protein design. We conduct the first systematic study of how capabilities evolve with model size for autoregressive transformers in the protein domain: we evaluate RITA models in next amino acid prediction, zero-shot fitness, and enzyme function prediction, showing benefits from increased scale. We release the RITA models openly, to the benefit of the research community.

1. Introduction

The ability to reliably design new proteins to tackle specific problems would mark the beginning of a new age, comparable to the transition between Stone and Iron age, according to Huang et al. (2016). While significant progress has been achieved towards this goal, beginning with directed evolution for protein engineering (Arnold, 1998), much work remains to be done. Machine learning has been applied to a number of problems in computational biology in recent years, with promising results. A substantial fraction of recent advances in this area has been possible thanks to the application of techniques originally developed for natural language processing (NLP), in particular with the recent trend towards large language models, motivated by the discovery of scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). In the world of protein design, there have also been efforts to train large generative models. The largest protein language model, ProGen (Madani et al., 2020), has been shown to be capable of generating protein sequences characterized by some desired downstream functions (Madani et al., 2021), but unfortunately the model remains closed source. Downstream open-source experimentation is important to discover surprising and unpredictable capabilities that are hard to discern without large-scale experimentation (Ganguli et al., 2022). This was recently exemplified when independent researchers discovered that AlphaFold 2 (Jumper et al., 2021) could successfully predict multimer interactions, even though it had only been trained to predict the structure of single protein chains (Yoshitaka, 2021; Baek, 2021). In addition, there exists no systematic study about the evolution of capabilities with respect to model size in the protein domain: Rao et al. (2020) and Rives et al. (2021) provided such a study for bidirectional transformers, and Madani et al. (2020) simply noted that their largest model was still underfitting.

Our contributions are as follows:

- We introduce RITA[1], a family of generative protein sequence models for protein design with up to 1.2B parameters.
- We study the relationship between model size and downstream task performance, taking a first step towards establishing scaling laws for protein sequence modeling.
- We release RITA models on Hugging Face and make them available to the scientific community at `https://github.com/lightonai/RITA`.

2. Related Work

2.1. Language models for natural language processing

Large transformer models have grown to become the de facto standard in natural language processing. As model size increased, a new paradigm of in-context learning and zero-shot classification has emerged. Instead of finetuning language models on specific tasks, models are instead trained on massive unstructured pre-training corpora where they can learn to solve a wide variety of tasks without explicit dataset curation. An important factor in the explosion of work on increasingly large language models is the discovery of scaling laws (Kaplan et al., 2020; Hoffmann et al.,

[1]LightOn, Paris, France [2]Department of Computer Science, University of Oxford, Oxford, UK [3]Department of Systems Biology; Harvard Medical School, Boston, MA, USA. Correspondence to: Daniel Hesslow <{firstname}@lighton.ai>.

[1]This project is dedicated to the loving memory of Rita Guidi (1961-2022). May ingenuity, human or otherwise, rid us one day of the disease that too soon stole you away from your loved ones.

Table 1. **Perplexity evaluation:** We evaluate generative protein models on the upstream modeling perplexity on four different datasets. In all cases performance is correlated with model size and RITA-XL provides the best results, highlighted in **bold**.

	RITA				BASELINE
DATASET	SMALL	MEDIUM	LARGE	XLARGE	PROTGPT2
UNIREF-100	10.07	7.47	6.18	**5.48**	18.10
METACLUST	15.08	13.80	12.17	**11.53**	21.07
MGNIFY	13.57	12.12	10.72	**9.89**	21.10
PFAM HELDOUT	11.78	10.68	9.23	**7.95**	15.05

Table 2. **Fitness evaluation - ProteinGym substitution benchmark:** We compute the Spearman's rank correlation between the fitness value measured experimentally and the predicted fitness value across the 87 substitution DMS assays from ProteinGym, and report the average values. RITA models approach the performance of specialized models with increasing parameter count, exceeding that of ESM-1v. Baselines results are based on a single seed. Results provided in full in Table 5. Best performance is in **bold**.

	RITA				BASELINES			
	SMALL	MEDIUM	LARGE	XLARGE	ESM-1V	MSA TRANSFORMER	TRANCEPTION	EVE
AVG FITNESS	0.330	0.370	0.381	0.387	0.371	0.422	**0.451**	0.448

2022), guiding decisions on optimal model and dataset size for a given compute budget. They allow the *a priori* estimation of the expected language modeling loss, reducing the risks associated with training such large models.

While massive unstructured pre-training corpora are readily available for protein sequences, there has only been limited work in both scaling up generative protein sequence models to the sizes seen in NLP, and in studying the effect of scaling on relevant downstream tasks. For these reasons, we explore the capabilities of generative protein models as model size is increased, facilitating future work on further scaling.

2.2. Protein Sequence Models

Much work has gone into exploring the potential of protein sequence models. UniRep (Alley et al., 2019) demonstrated that the internal representation learned by an LSTM-based protein sequence model was sufficient to predict protein secondary structure, stability, and downstream function. Subsequent works focusing on bidirectional models, including TAPE-BERT, ESM-1b, ProtTrans and ProteinBERT (Rao et al., 2019; 2020; Rives et al., 2021; Elnaggar et al., 2021; Brandes et al., 2021) have improved upon this by employing more capable models based on Transformers (Vaswani et al., 2017). DeepSequence (Riesselman et al., 2017) and ESM-1v (Meier et al., 2021) have shown that these representations can also be successfully leveraged for variant effect prediction, whereas RGN2 (Chowdhury et al., 2021) recently demonstrated their utility for fast and accurate single-sequence tertiary structure prediction.

UniRep has been successfully employed for protein engineering (Biswas et al., 2020), and the more recent transformer-based ProGen (Madani et al., 2020) is capable of generating proteins with a number of desired characteristics by conditioning the model on a variety of sequence metadata.

Furthermore, several works have explored the use of generative protein sequence models as part of a fixed-backbone protein design pipeline, either by conditioning the sequence model on structural information through a cross-attention setup (Ingraham et al., 2019), by coupling it with a structure predictor enhanced decoding strategy (Moffat et al., 2021) or by iteratively finetuning it on sequences refined by AlphaFold 2 (Moffat et al., 2022).

3. Methods

3.1. Model Architecture

A range of techniques to control neural language generation have been developed recently (Weng, 2021; Zarrieß et al., 2021). However, to provide the scientific community with a model as generally applicable as possible, we chose to train our models as decoder-only transformer models without any conditioning information. We performed a small ablation study over positional embedding techniques, where we evaluated Rotary Positional Embeddings (RoPE) (Su et al., 2021) and AliBi (Press et al., 2021), and chose to use RoPE due to the resulting lower language modeling loss, shown in Table 7. We trained four different models in order to study the relationship between model size and downstream capability, and use the same model hyperparameters and naming scheme as GPT-3 (Brown et al., 2020).

While tokenization is widely used in natural language processing, there are important differences between natural languages and protein sequences: books consists of hun-

Table 3. **Enzyme function prediction:** We predict the functional properties of proteins in SwissProt and report the top-k accuracy. Performance scales smoothly with model size and RITA-XL gives the best results, highlighted in **bold**.

	RITA				BASELINE
TOP-K	SMALL	MEDIUM	LARGE	XLARGE	PROTGPT2
@1	87.8	89.5	90.8	**91.6**	85.7
@3	90.7	90.2	93.2	**93.7**	88.7
@10	92.4	93.9	94.5	**94.7**	90.6

dreds of thousands of characters while the average protein sequence in UniProtKB/TrEMBL is only 349 amino acids long. Protein sequences also lack a natural decomposition into something equivalent to words. Additionally, using tokenization schemes with a varied-length vocabulary may have undesirable side effects, such as producing tokenized sequences of different lengths for two proteins that would only differ by a single substitution. This would complicate the comparison of relative likelihoods, or the generation of sequences with a target output length (e.g., for fixed-backbone protein design).

3.2. Data

To preserve all information contained within the pre-training data we chose not to perform any clustering before training. We focus on three different pre-training corpora: UniRef-100 (The UniProt Consortium, 2020), MGnify (Mitchell et al., 2020) and Metaclust (Steinegger & Söding, 2018), each providing a sufficient amount of tokens for model pre-training without having to repeat the data. We then train three small models for a short amount of time to estimate transferability of each dataset to the others. The experiments showed that we would get the best results by utilizing UniRef-100 followed by Metaclust, and worst results with MGnify, as shown in Table 8. However, we note that using a combination of several datasets may be beneficial.

During pre-training we randomly map amino acids `B`, `Z` and `J` to `(D,N)`, `(E,Q)` and `(I,L)` respectively and remove any sequence containing `X`. We train both on the primary sequence and its reverse.

3.3. Training

We utilize the Megatron-Deepspeed framework to achieve high training throughput and train the models using a combination of data and pipeline parallelism.

All models were trained on a total of 150 billion amino acids, and the training runs were performed on the Jean Zay supercomputer of IDRIS. The models were trained for a total training time of over 25 thousand `Nvidia-V100` GPU hours. We utilize the Adam optimizer (Kingma & Ba, 2015), a batch size of 512, and a context size of 1024 for all experiments.

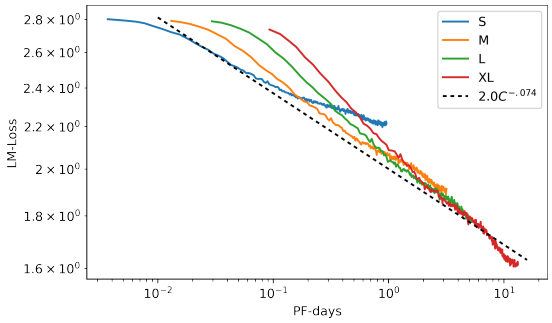

Figure 1. Protein modeling loss as a function of compute measured in PetaFLOPS-days (PF-days).

4. Evaluation

4.1. Perplexity evaluation

Autoregressive models are typically evaluated by their modeling loss. The model is trained on this task and it should broadly reflect its capabilities. We measure the perplexity on three different protein databases: UniRef-100, MGnify and Metaclust. Our models are trained on UniRef-100, a large collection of sequenced proteins, whereas Metaclust and MGnify consist of metagenomically transcribed proteins. We argue that this should provide a challenging distribution shift for the model. We additionally withheld a set of twenty protein families (the same held out by Madani et al. (2020)) to evaluate the generalization to unseen protein families.

For all datasets we compare our results with those of ProtGPT2 (Ferruz et al., 2022). Since ProtGPT2, is trained on tokenized sequences, we measure the perplexity per amino acid [2]. We present our results in Table 1 where we can see that there is a clear improvement with increasing model size.

4.2. Scaling Laws

By training four generative protein sequence models ranging over an order of magnitude in size, we are able to establish scaling laws similar to those established by Kaplan et al. (2020) for natural language processing, see Figure 1. We observe an exponent of 0.74, significantly steeper than the

[2] See appendix E for a discussion around perplexity per byte

Table 4. **Prompt tuning:** We perform prompt tuning to generate proteins from the family `PF03272`. The perplexity of the `Prompt-Tuned Model` is significantly lower than the one of the `Base Model`, indicating the the model has learned to better generate proteins from the target family.

	RITA			
PERPLEXITY	SMALL	MEDIUM	LARGE	XLARGE
BASE MODEL	15.69	13.60	10.43	7.35
PROMPT-TUNED MODEL	**10.37**	**9.19**	**6.99**	**4.96**

one observed in NLP of around 0.5. In the regime we have analyzed, up to 1.2 billion parameters, scaling up protein sequence models is thus significantly more beneficial than scaling up language models in NLP. However, in contrast to NLP, where the modeling loss typically follows the power law relationship remarkably closely, we observe some deviation. In particular, we observe a sharp decrease of loss for our largest model after a significant amount of training, and note that the loss does not go above 2.8 due to the small vocabulary size. Interestingly, even though our models are trained far beyond the point of optimality in NLP, all but our smallest model still appear to be undertrained. Our models are trained for 150 billion amino acids, whereas Kaplan et al. (2020) and Hoffmann et al. (2022) estimated optimality for our largest model at around 25 billion amino acids.

4.3. Mutation Effects Prediction

We assess the ability of our models to predict the effects of mutations by interpreting the likelihood that the model outputs for a given protein as its fitness value. We use the ProteinGym benchmarks (Notin et al., 2022) which provide experimentally-measured fitness values across 94 Deep Mutational Scanning (DMS) assays. On the substitution benchmark (Table 2), we compare against several baselines, including MSA Transformer (Rao et al., 2021), ESM-1v (Meier et al., 2021), Tranception (Notin et al., 2022) and EVE (Frazer et al., 2021). We observe that the performance of RITA models increases with model size, exceeding that of ESM-1v for the Large and XLarge variants. While alignment-based models (eg., EVE) and models relying on the marked-marginals heuristics for scoring (eg., ESM-1v, MSA Transformer) are unable to score indels, autoregressive transformers such as RITA models can quantify their fitness out-of-the-box, performing on par with the specialized models introduced in Shin et al. (2021) (Table 6).

4.4. Enzyme Function Prediction

To evaluate the capability of the models to predict enzyme function, we utilize the sequence representation obtained at the final token. Following ProteInfer (Sanderson et al., 2021), we extract enzyme commission metadata from SwissProt (Bairoch & Apweiler, 2000) and focus on the tags belonging to the finest classification level (randomly choosing one for those with multiple tags), obtaining a classification problem with 4793 classes. The processed dataset is available at `https://huggingface.co/datasets/lightonai/SwissProt-EC-leaf`.

We train a linear classifier for one epoch on top of the extracted representations, and present the results in Table 3. Similar to previous tasks, performance increases with scale.

4.5. Prompt Tuning

Language models can solve a wide variety of tasks by prefixing the generation with a manually created prompt. This practice has received the name of *prompt engineering*, in reference to the laborious process of finding a prompt that yields satisfactory generation. Inspired by this, Lester et al. (2021) developed *prompt tuning*, a method to automatically learn soft prompts in the embedding space. Prompt tuning has emerged as an important way to perform *parameter efficient fine-tuning*, learning only a fraction of the number of parameters typically needed for fine-tuning.

We investigate if it is possible to add controllable generation to pre-trained protein sequence models by leveraging *prompt tuning*. We arbitrarily chose one of the protein families that were held out during training, `PF03272`, and learned a prompt specializing in generating proteins from this family. In Table 4 we see a significant reduction in perplexity with prompt tuning, showing that the model is indeed able to learn to generate proteins from this protein family.

5. Conclusion and Future Work

In this work we have presented RITA, a family of generative protein sequence models, aiming to accelerate future work on protein design. We have systematically evaluated how model capabilities increase with size and taken a first step towards establishing scaling laws for protein sequence modeling. We believe that the release of our models will represent a building block for future endeavours into protein design. We also look forward to future work studying RITA-designed proteins *in-vitro*, further scaling up protein sequence models or augmenting them with target structure embeddings for fixed-backbone protein design.

References

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, 2019. doi: 10.1101/589333. URL https://www.biorxiv.org/content/early/2019/03/26/589333.

Arnold, F. H. Design by directed evolution. *Accounts of Chemical Research*, 31:125–131, 1998.

Baek, M. Twitter post: Adding a big enough number for residue index feature is enough to model hetero-complex using alphafold (green&cyan crystal structure magenta: predicted model w/ residue_index modification), July 2021. URL https://twitter.com/minkbaek/status/1417538291709071362.

Bairoch, A. and Apweiler, R. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, 28(1):45–48, 2000.

Biswas, S., Khimulya, G., Alley, E. C., Esvelt, K. M., and Church, G. M. Low-n protein engineering with data-efficient deep learning. *bioRxiv*, 2020. doi: 10.1101/2020.01.23.917682. URL https://www.biorxiv.org/content/early/2020/01/24/2020.01.23.917682.

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., and Linial, M. ProteinBERT: A universal deep-learning model of protein sequence and function, May 2021.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Chowdhury, R., Bouatta, N., Biswas, S., Rochereau, C., Church, G. M., Sorger, P. K., and AlQuraishi, M. Single-sequence protein structure prediction using language models from deep learning, August 2021.

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., and Rost, B. ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. ISSN 0162-8828, 2160-9292, 1939-3539. doi: 10.1109/TPAMI.2021.3095381.

Ferruz, N., Schmidt, S., and Höcker, B. A deep unsupervised language model for protein design. *bioRxiv*, 2022.

Frazer, J., Notin, P., Dias, M., Gomez, A. N., Min, J. K., Brock, K. P., Gal, Y., and Marks, D. S. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 2021.

Ganguli, D., Hernandez, D., Lovitt, L., DasSarma, N., Henighan, T. J., Jones, A., Joseph, N., Kernion, J., Mann, B., Askell, A., Bai, Y., Chen, A., Conerly, T., Drain, D., Elhage, N., Showk, S. E., Fort, S., Hatfield-Dodds, Z., Johnston, S., Kravec, S., Nanda, N., Ndousse, K., Olsson, C., Amodei, D., Amodei, D., Brown, T. B., Kaplan, J., McCandlish, S., Olah, C., and Clark, J. Predictability and surprise in large generative models. *ArXiv*, abs/2202.07785, 2022.

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., Rae, J. W., Vinyals, O., and Sifre, L. Training compute-optimal large language models. *ArXiv*, abs/2203.15556, 2022.

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

Ingraham, J., Garg, V., Barzilay, R., and Jaakkola, T. Generative models for graph-based protein design. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/f3a4ff4839c56a5f460c88cce3666a2b-Paper.pdf.

Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589, 2021.

Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2015.

Lester, B., Al-Rfou, R., and Constant, N. The power of scale for parameter-efficient prompt tuning. *arXiv preprint arXiv:2104.08691*, 2021.

Madani, A., McCann, B., Naik, N., Keskar, N. S., Anand, N., Eguchi, R. R., Huang, P.-S., and Socher, R. ProGen: Language Modeling for Protein Generation. *arXiv:2004.03497 [cs, q-bio, stat]*, March 2020.

Madani, A., Krause, B., Greene, E. R., Subramanian, S., Mohr, B. P., Holton, J. M., Olmos, J. L., Xiong, C., Sun, Z. Z., Socher, R., Fraser, J. S., and Naik, N. Deep neural language modeling enables functional protein generation across families. *bioRxiv*, 2021.

Meier, J., Rao, R., Verkuil, R., Liu, J., Sercu, T., and Rives, A. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021. doi: 10.1101/2021.07.09.450648. URL https://www.biorxiv.org/content/early/2021/11/17/2021.07.09.450648.

Mitchell, A. L., Almeida, A., Beracochea, M., Boland, M. A., Burgin, J., Cochrane, G., Crusoe, M. R., Kale, V., Potter, S. C., Richardson, L. J., Sakharova, E. A., Scheremetjew, M., Korobeynikov, A. I., Shlemov, A., Kunyavskaya, O., Lapidus, A. L., and Finn, R. D. Mgnify: the microbiome analysis resource in 2020. *Nucleic Acids Research*, 48:D570 – D578, 2020.

Moffat, L., Greener, J. G., and Jones, D. T. Using alphafold for rapid and accurate fixed backbone protein design. *bioRxiv*, 2021. doi: 10.1101/2021.08.24.457549. URL https://www.biorxiv.org/content/early/2021/08/26/2021.08.24.457549.

Moffat, L., Kandathil, S. M., and Jones, D. T. Design in the dark: Learning deep generative models for de novo protein design. *bioRxiv*, 2022. doi: 10.1101/2022.01.27.478087. URL https://www.biorxiv.org/content/early/2022/01/28/2022.01.27.478087.

Notin, P., Dias, M., Frazer, J., Marchena-Hurtado, J., Gomez, A. N., Marks, D. S., and Gal, Y. Tranception: protein fitness prediction with autoregressive transformers and inference-time retrieval. *ArXiv*, abs/2205.13760, 2022.

Press, O., Smith, N. A., and Lewis, M. Train short, test long: Attention with linear biases enables input length extrapolation. *ArXiv*, abs/2108.12409, 2021.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. Evaluating protein transfer learning with tape, 2019. URL https://arxiv.org/abs/1906.08230.

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., and Rives, A. Transformer protein language models are unsupervised structure learners, December 2020.

Rao, R., Liu, J., Verkuil, R., Meier, J., Canny, J. F., Abbeel, P., Sercu, T., and Rives, A. MSA Transformer, August 2021.

Riesselman, A. J., Ingraham, J. B., and Marks, D. S. Deep generative models of genetic variation capture mutation effects. *arXiv preprint arXiv:1712.06527*, 2017.

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, 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. *Proceedings of the National Academy of Sciences of the United States of America*, 118, 2021.

Sanderson, T., Bileschi, M. L., Belanger, D., and Colwell, L. Proteinfer: deep networks for protein functional inference. *bioRxiv*, 2021.

Shin, J.-E., Riesselman, A. J., Kollasch, A. W., McMahon, C., Simon, E., Sander, C., Manglik, A., Kruse, A. C., and Marks, D. S. Protein design and variant prediction using autoregressive generative models. *Nature Communications*, 12(1):2403, April 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-22732-w.

Steinegger, M. and Söding, J. Clustering huge protein sequence sets in linear time. *Nature Communications*, 9, 2018.

Su, J., Lu, Y., Pan, S., Wen, B., and Liu, Y. Roformer: Enhanced transformer with rotary position embedding. *ArXiv*, abs/2104.09864, 2021.

The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1): D480–D489, 11 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. URL https://doi.org/10.1093/nar/gkaa1100.

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*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Weng, L. Controllable neural text generation. *lilianweng.github.io*, 2021. URL https://lilianweng.github.io/posts/2021-01-02-controllable-text-generation/.

Yoshitaka, M. Twitter post: Alphafold2 can also predict heterocomplexes. all you have to do is input the two sequences you want to predict and connect them with a long linker., July 2021. URL hhttps://twitter.com/Ag_smith/status/1417063635000598528.

Zarrieß, S., Voigt, H., and Schüz, S. Decoding methods in neural language generation: A survey. *Inf.*, 12:355, 2021.

A. Full results from our fitness evaluation.

Table 5. **Fitness evaluation - ProteinGym substitution benchmark:** Spearman's rank correlation between experimentally measured fitness values for different proteins and the value predicted by the models. Baselines results are based on a single seed. Tranception NR and Tranception R are variants without and with retrieval respectively. We follow the same approach as in Notin et al. (2022) and aggregate results at the Uniprot ID level to avoid biasing results towards proteins for which several assays are available. To compare with models relying on evolutionary data in the form of multiple-sequence alignments (eg., EVE), we only evaluate on the subset of mutations where coverage is deemed high enough by these models to make a prediction.

Uniprot_ID	RITA				Baselines				
	S	M	L	XL	ESM-1v	MSA Transformer	Tranception NR	Tranception R	EVE
A0A140D2T1_ZIKV	0.350	0.308	0.317	0.304	-0.064	0.465	0.268	0.346	0.366
A0A192B1T2_9HIV1	0.492	0.504	0.504	0.501	0.488	0.510	0.510	0.509	0.510
A0A1I9GEU1_NEIME	-0.022	0.028	0.061	0.074	0.046	0.077	0.088	0.044	-0.004
A0A2Z5U3Z0_9INFA	0.456	0.518	0.502	0.525	0.485	0.326	0.528	0.545	0.529
A4D664_9INFA	0.329	0.386	0.404	0.398	0.026	0.333	0.404	0.393	0.409
A4GRB6_PSEAI	0.411	0.537	0.562	0.619	0.647	0.707	0.598	0.663	0.672
A4_HUMAN	0.322	0.276	0.312	0.300	0.309	0.394	0.364	0.452	0.301
AACC1_PSEAI	0.271	0.292	0.349	0.402	0.488	0.505	0.407	0.448	0.499
ADRB2_HUMAN	0.514	0.511	0.513	0.484	0.523	0.436	0.501	0.542	0.534
AMIE_PSEAE	0.488	0.517	0.532	0.563	0.606	0.611	0.501	0.585	0.558
B3VI55_LIPST	0.284	0.389	0.431	0.454	0.483	0.535	0.491	0.468	0.436
BLAT_ECOLX	0.564	0.556	0.546	0.524	0.646	0.681	0.489	0.627	0.682
BRCA1_HUMAN	0.389	0.397	0.495	0.499	0.442	0.402	0.538	0.574	0.322
C6KNH7_9INFA	0.394	0.369	0.371	0.373	0.420	0.408	0.401	0.445	0.435
CALM1_HUMAN	0.186	0.245	0.258	0.275	0.246	0.254	0.306	0.283	0.244
CAPSD_AAV2S	0.191	0.250	0.269	0.279	0.196	0.350	0.492	0.473	0.346
CCDB_ECOLI	0.142	0.076	0.029	0.210	0.426	0.489	0.309	0.456	0.490
CP2C9_HUMAN	0.596	0.593	0.603	0.581	0.623	0.596	0.595	0.652	0.625
DLG4_HUMAN	0.573	0.578	0.563	0.535	0.544	0.595	0.576	0.662	0.594
DLG4_RAT	0.381	0.390	0.377	0.371	0.565	0.507	0.304	0.446	0.523
DYR_ECOLI	0.198	0.361	0.267	0.313	0.420	0.488	0.348	0.424	0.468
ENV_HV1B9	0.380	0.358	0.408	0.419	0.415	0.380	0.404	0.407	0.388
ENV_HV1BR	0.352	0.36	0.372	0.364	0.322	0.345	0.358	0.363	0.341
ESTA_BACSU	0.122	0.199	0.283	0.301	0.304	0.428	0.263	0.325	0.375
F7YBW8_MESOW	-0.076	-0.123	-0.105	-0.006	0.382	0.375	0.434	0.425	0.411
GAL4_YEAST	0.287	0.345	0.356	0.353	0.441	0.583	0.326	0.526	0.511
GCN4_YEAST	0.385	0.417	0.411	0.407	0.288	0.288	0.384	0.356	0.252
GFP_AEQVI	0.080	0.108	0.182	0.096	0.099	0.652	0.631	0.677	0.679
GRB2_HUMAN	0.522	0.471	0.484	0.382	0.484	0.468	0.429	0.489	0.566
HIS7_YEAST	0.325	0.402	0.433	0.478	0.411	0.508	0.585	0.616	0.476
HSP82_YEAST	0.432	0.439	0.427	0.457	0.500	0.445	0.436	0.461	0.469
I6TAH8_I68A0	0.308	0.328	0.374	0.377	0.018	0.303	0.337	0.348	0.364
IF1_ECOLI	0.364	0.459	0.381	0.417	0.538	0.227	0.548	0.509	0.525
KCNH2_HUMAN	0.452	0.489	0.473	0.434	0.233	0.368	0.484	0.513	0.229
KKA2_KLEPN	0.296	0.425	0.538	0.556	0.614	0.576	0.584	0.584	0.597
MK01_HUMAN	0.220	0.129	0.099	0.053	0.183	0.153	0.034	0.139	0.251
MSH2_HUMAN	0.303	0.325	0.278	0.263	0.398	0.410	0.292	0.360	0.405
MTH3_HAEAE	0.358	0.491	0.625	0.677	0.701	0.687	0.673	0.655	0.710
NCAP_I34A1	0.352	0.382	0.408	0.413	0.019	0.338	0.415	0.424	0.363
NRAM_I33A0	0.583	0.633	0.584	0.571	0.162	0.519	0.551	0.621	0.584
NUD15_HUMAN	0.316	0.451	0.513	0.498	0.615	0.630	0.547	0.591	0.608
P53_HUMAN	0.364	0.484	0.478	0.448	0.487	0.396	0.388	0.461	0.495
P84126_THETH	0.415	0.506	0.477	0.552	0.546	0.631	0.533	0.541	0.567
PABP_YEAST	0.640	0.666	0.667	0.693	0.665	0.662	0.641	0.689	0.639
PA_I34A1	0.456	0.493	0.533	0.538	0.054	0.383	0.541	0.572	0.539
POLG_CXB3N	0.328	0.382	0.374	0.369	-0.057	0.476	0.347	0.405	0.465
POLG_HCVJF	0.390	0.434	0.443	0.487	0.605	0.600	0.525	0.577	0.614
PTEN_HUMAN	0.242	0.404	0.382	0.389	0.436	0.491	0.341	0.459	0.501
Q2N0S5_9HIV1	0.507	0.418	0.398	0.348	0.496	0.490	0.412	0.492	0.496
Q59976_STRSQ	0.579	0.638	0.644	0.654	0.506	0.674	0.645	0.659	0.647
R1AB_SARS2	0.214	0.259	0.274	0.289	-0.030	-0.037	0.216	0.401	0.600
RASH_HUMAN	0.439	0.419	0.423	0.399	0.359	0.415	0.377	0.447	0.454
REV_HV1H2	0.224	0.271	0.247	0.246	0.249	0.251	0.246	0.245	0.227
RL401_YEAST	0.384	0.522	0.488	0.436	0.288	0.392	0.355	0.397	0.395
SC6A4_HUMAN	0.420	0.410	0.400	0.411	0.472	0.509	0.400	0.465	0.489
SCN5A_HUMAN	0.121	0.151	0.167	0.131	0.176	0.157	0.073	0.093	0.199
SPG1_STRSG	0.252	0.216	0.214	0.208	0.237	0.142	0.279	0.289	0.247
SPIKE_SARS2	0.311	0.375	0.366	0.375	-0.043	0.471	0.369	0.342	0.347
SRC_HUMAN	0.439	0.413	0.421	0.373	0.561	0.258	0.348	0.493	0.505
SUMO1_HUMAN	0.223	0.369	0.407	0.391	0.430	0.423	0.424	0.488	0.531
SYUA_HUMAN	0.195	0.219	0.195	0.186	0.281	0.128	0.160	0.146	0.167
TADBP_HUMAN	0.174	0.070	-0.018	-0.011	0.060	0.050	0.123	0.125	0.071
TAT_HV1BR	0.378	0.363	0.396	0.399	0.342	0.310	0.206	0.244	0.337
TPK1_HUMAN	0.099	0.151	0.261	0.289	0.284	0.268	0.313	0.314	0.230
TPMT_HUMAN	0.373	0.462	0.509	0.515	0.54	0.508	0.445	0.522	0.548
TPOR_HUMAN	0.245	0.327	0.305	0.369	0.362	0.507	0.410	0.453	0.393
TRPC_SACS2	0.425	0.558	0.499	0.568	0.606	0.629	0.551	0.585	0.577
TRPC_THEMA	0.333	0.392	0.392	0.452	0.472	0.474	0.453	0.436	0.420
UBC9_HUMAN	0.237	0.438	0.473	0.452	0.479	0.503	0.433	0.485	0.538
UBE4B_MOUSE	0.117	0.325	0.334	0.293	0.462	0.347	0.256	0.388	0.476
VKOR1_HUMAN	0.166	0.173	0.343	0.375	0.447	0.472	0.466	0.502	0.462
YAP1_HUMAN	0.180	0.177	0.160	0.168	0.281	0.071	0.218	0.359	0.438
Average	0.330	0.370	0.381	0.387	0.371	0.422	0.406	0.451	0.448

Table 6. **Fitness evaluation - ProteinGym indel benchmark:** Spearman's rank correlation between experimentally measured fitness values for different proteins and the value predicted by the models. The Wavenet models are based on Shin et al. (2021). Tranception NR and Tranception R (Notin et al., 2022) are variants without and with retrieval respectively.

Uniprot_ID	RITA				Baselines		
	S	M	L	XL	Wavenet	Tranception NR	Tranception R
A0A1J4YT16_9PROT_Davidi_2020	-0.169	0.185	0.207	0.210	0.117	0.178	0.191
B1LPA6_ECOSM_Russ_2020	0.292	0.383	0.339	0.348	0.385	0.321	0.415
BLAT_ECOLX_Gonzalez_indels_2019	0.436	0.455	0.334	0.345	0.546	0.296	0.357
CAPSD_AAV2S_Sinai_indels_2021	0.253	0.319	0.453	0.463	0.699	0.563	0.598
HIS7_YEAST_Pokusaeva_indels_2019	0.638	0.656	0.677	0.684	0.457	0.549	0.586
P53_HUMAN_Kotler_deletions_2018	0.360	0.407	0.383	0.273	0.680	0.707	0.692
PTEN_HUMAN_Mighell_deletions_2018	0.612	0.575	0.504	0.523	0.001	0.395	0.401
Average	0.346	0.426	0.414	0.406	0.412	0.430	0.463

B. Positional embedding ablation

Table 7. **Ablating different positional embeddings:** We evaluate Rotary Positional Embeddings (RoPE) as well as ALiBi by training a small model for 3 billion amino acids. As shown, RoPE outperform ALiBi. However, we note that the training runs were stopped after a short amount of time to save computational resources, and that larger scale ablation is needed for reliable results. We also note that the training dataset differs from the one used for the main training runs, and that for this reason these results should not be directly compared to those presented in Table 1.

	SMALL-ROTARY	SMALL-ALIBI
PERPLEXITY	12.43	13.08

C. Dataset Selection

Table 8. **Ablating different datasets:** We train small models for $\sim$ 3 GT to evaluate the use of different pre-training datasets. All model are then evaluated on a combination of the datasets, and the results are shown below.

	UNIREF-100	METACLUST	MGNIFY
PERPLEXITY	14.28	14.62	15.34

D. Comparisons with ProtXLNet

A previous version of this paper contained faulty perplexity comparisons with ProtXLNet, after correspondence with the authors we have decided to entirely remove comparisons with ProtXLNet from the main part of the paper, and provide a fixed perplexity evaluation here, in the appendix . The goal of these comparisons was to contextualize our models compared to similar previous models. However, the XLNet architecture is rather different to that of a decoder-only autoregressive transformer. In table 9 we show the perplexity evaluation, with correctly computed values for ProtXLNet. Additionally in this evaluation we have also removed all proteins of length less than one hundred amino acids.

E. Comparing perplexity across different vocabularies

Evaluating the perplexity of a model has long been standard practice in natural language processing. However, the traditional way of computing the perplexity, per token, does not transfer across vocabularies. It is naturally much harder to predict the correct next token in a large vocabulary of tens of thousands of tokens compared to a small vocabulary with only a few dozens tokens to chose from.

In order to compare across vocabularies one must normalize the perplexity by the length of the untokenized sequence, instead of the tokenized sequence. This metric is typically called the *perplexity per byte*, although for proteins it may be more natural to call it the perplexity per amino-acid. If one merges tokens and assigns the probability of the merged tokens as the joint probability of its constituents, the perplexity per byte will remain constant. This property allows fair comparisons

Table 9. **Perplexity evaluation:** We evaluate generative protein models on the upstream modeling task by measuring the models perplexity-per-byte on four different datasets. In all cases performance is correlated with model size and RITA-XL provides the best results, highlighted in **bold**. We further note that both ProtGPT2 and ProtXLNet were trained on the Pfam families we held out of our training set.

	RITA				BASELINES	
DATASET	SMALL	MEDIUM	LARGE	XLARGE	PROTGPT2	PROTXLNET
UNIREF-100	9.93	7.32	6.05	**5.36**	17.96	18.32
METACLUST	12.93	11.26	9.93	**9.33**	20.16	22.33
MGNIFY	12.90	11.33	10.03	**9.21**	19.97	22.52
PFAM HELDOUT	11.76	10.66	9.21	**7.93**	15.02	16.39

across vocabulary sizes. However, perplexity per byte also has its flaws: in redundant vocabularies, where each sequence can be represented in multiple ways, the perplexity per byte will unjustly increase. In such a case the model needs to guess which of all possible tokenizations corresponds to the target sequence. See Table 10 for an example.

While all common tokenizers are deterministic, meaning that the model should be able to learn which of the possible tokenizations the tokenizer will chose, we hypothesize that this can be a difficult task when the length of each word grows longer. For proteins, where there are no word boundaries, this may cause problems when using standard tokenizers, such as BPE. In preliminary experiments using tokenization we saw improved results by breaking up the proteins into k-mers before tokenization.

While perplexity per byte is the standard way to compare upstream performance across vocabularies we would like to caution against reading too much into the exact values. Unfortunately, we are not aware of any better metric to compare across vocabularies.

Table 10. **Perplexity and vocabularies**: Example of how the perplexity per token and the perplexity per byte change with vocabularies given a uniform probabilities across the tokens in the vocabulary.

VOCAB	VOCAB TYPE	SEQUENCE	PERPLEXITY PER TOKEN	PERPLEXITY PER BYTE
A, B	UNTOKENIZED	A,B,B,A	$\exp(-\frac{4*ln(0.5)}{4}) = 2$	$\exp(-\frac{4*ln(0.5)}{4}) = 2$
AA, BB, AB, BA	TOKENIZED	AB, BA	$\exp(-\frac{2*ln(0.25)}{2}) = 4$	$\exp(-\frac{2*ln(0.25)}{4}) = 2$
A, B, AA, BB, AB, BA	REDUNDANTLY TOKENIZED	A, BB, A	$\exp(-\frac{3*ln(0.167)}{3}) = 6$	$\exp(-\frac{3*ln(0.167)}{4}) \approx 3.83$

F. Acknowledgments

The authors thank Julien Launay and Igor Carron for fruitful discussions.

This work was made possible through the use of the HPC/AI resources at IDRIS under GENCI allocation 2020-AD011012024 which provided access to the Jean Zay supercomputer. We thank Stéphane Réquena and the support team for their valuable help.

Daniel Hesslow is supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860360.
Pascal Notin is supported by GSK and the UK Engineering and Physical Sciences Research Council (EPSRC ICASE award no. 18000077).