Profile Prediction: An Alignment-Based Pre-Training Task for Protein Sequence Models

Pascal Sturmfels[*]
Paul G. Allen School of Computer Science
University of Washington
`psturm@cs.washington.edu`

Jesse Vig
Salesforce Research
`jvig@salesforce.com`

Ali Madani
Salesforce Research
`amadani@salesforce.com`

Nazneen Fatema Rajani
Salesforce Research
`nazneen.rajani@salesforce.com`

Abstract

For protein sequence datasets, unlabeled data has greatly outpaced labeled data due to the high cost of wet-lab characterization. Recent deep-learning approaches to protein prediction have shown that pre-training on unlabeled data can yield useful representations for downstream tasks. However, the optimal pre-training strategy remains an open question. Instead of strictly borrowing from natural language processing (NLP) in the form of masked or autoregressive language modeling, we introduce a new pre-training task: directly predicting protein profiles derived from multiple sequence alignments. Using a set of five, standardized downstream tasks for protein models, we demonstrate that our pre-training task along with a multi-task objective outperforms masked language modeling alone on all five tasks. Our results suggest that protein sequence models may benefit from leveraging biologically-inspired inductive biases that go beyond existing language modeling techniques in NLP.

1 Introduction

Experimental measurements of protein properties, ranging from structural to functional characterization, are usually far more expensive than gathering unlabeled protein sequence data (Rocklin et al., 2017; Stevens, 2003). Predicting protein properties directly from sequence data using machine learning models could potentially enable a new era of biological discovery. However, the vast majority of existing protein sequence data is unlabeled, lacking both structural (e.g. x-ray crystallography) or functional (e.g. enzymatic activity) labels. In other domains, such as computer vision or natural language processing (NLP), existing approaches leverage large, unlabeled datasets via self-supervised pre-training: the practice of first training a model using a loss function derived solely from unlabeled data. In the protein domain, self-supervision has recently been successful at improving downstream performance on canonical tasks in protein science and engineering (Devlin et al., 2019; Rao et al., 2019).

There are several similarities between protein sequence modeling and NLP—namely, sequences comprised of a discrete set of characters as input, and far more unlabeled data than labeled. This has inspired many recent papers adapting NLP models to protein sequence tasks (Alley et al., 2019; Elnaggar et al., 2020; Consortium, 2019; Madani et al., 2020; Rives et al., 2019; Strodthoff et al., 2020; Nambiar et al., 2020). These papers employ not only architectures inspired by NLP but also

---

[*]This research was conducted during the author's internship at Salesforce Research
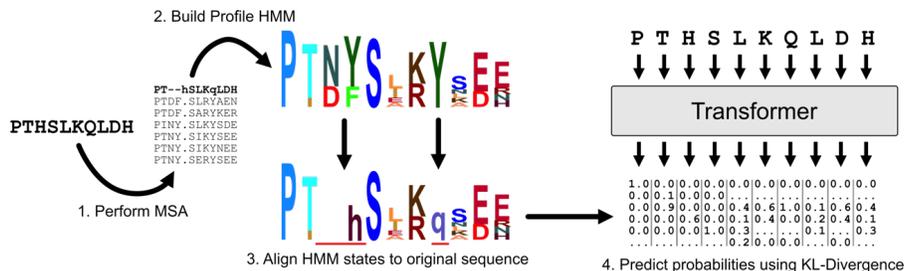
Figure 1: An overview of our proposed task. We start with an initial sequence "PTHSLKQLDH". We generate a multiple sequence alignment for that sequence by searching it against a reference database, and we then generate a profile HMM for the alignment. The first H and the Q in our sequence correspond to inserted amino acids that didn't match any columns in the alignment. Therefore, for those amino acids we use insertion state emissions as labels rather than match state emissions. The rest of the amino acids in our sequence were in match states, so we use the match state emission probabilities as labels. Our protein has deletions in two of the match states in the MSA (columns 2 and 3). We omit these from the label since they have no corresponding amino acids as inputs. Finally, we predict the corresponding label using KL divergence, averaged over the length of the sequence.

pre-training objectives: masked language modeling and auto regressive generation. Unfortunately, on some tasks, such as secondary structure and contact prediction, purely using masked language modeling objectives for pretraining has only marginally improved performance as compared to standard alignment-based techniques (Rao et al., 2019). This presents an opportunity to design new pre-training tasks and inductive biases that are better tailored toward the underlying protein biology.

Some existing work has already begun to investigate protein-specific pre-training tasks. Bepler and Berger (2018) propose a new task called soft symmetric alignment (SSA), which defines a custom similarity function that compares the pairwise distances between protein embeddings. Lu et al. (2020) introduce a contrastive loss that trains an RNN to discriminate fragments coming from a source sequence versus randomly sampled fragments from other sequences. Our work differs from both existing works in that we make direct use of multiple sequence alignments (MSAs).

In this paper, we introduce a new pre-training task for protein sequence models. Our task is inspired by alignment-based protein structure predictors: the very models that outperform existing deep NLP models at protein structure prediction (Klausen et al., 2019; Elnaggar et al., 2020; Ma et al., 2015). Such models take as input features derived from multiple sequence alignments (MSAs), which cluster proteins with related sequences. Features derived from MSAs, such as position-specific scoring matrices and hidden Markov model profiles, have long known to be useful features for predicting the structure of a protein (Cuff and Barton, 1999; Jones, 1999; Cuff and Barton, 2000; Rost, 2001). Our task posits the reverse: that, in order to predict profiles derived from MSAs from a single protein in the alignment, a network must learn information about that protein's structure. Specifically, we propose to use HMM profiles derived from MSAs as labels during pre-training, rather than as input features in a downstream task. An overview of the task is depicted in Figure 1.

The remainder of this paper is organized as follows. In Section 2, we describe our pre-training task in detail as well as the architecture and hyperparameters we use for further experiments. In Section 3, we compare our pre-training task against masked language modeling and a multitask loss that combines the two of them, while keeping architecture, hyperparameters and pre-training data constant. We also compare our model to existing models in the literature. Section 4 concludes with possibilities for future work.

2 Methods

In this section, we describe in detail our pre-training task as well as specifics regarding training and architecture details. We begin with an overview on multiple sequence alignment.

2.1 Multiple Sequence Alignments

Multiple sequence alignment (MSA) is a method for grouping a series of related proteins. MSA arranges proteins in a matrix whose rows are individual protein sequences and whose columns contain amino acids that either come from the same position in some ancestral sequence (homologous), or play a common structural or functional role. Building an MSA is a common first step in many downstream tasks including predicting structure and finding homologous proteins (Edgar and Batzoglou, 2006). For our models, we use the pre-training data introduced in Rao et al. (2019), which comprises 32 million sequences from Pfam (El-Gebali et al., 2019). Pfam further contains pre-built MSAs for each of its entries, grouped into a set of families. Although it would be possible to build a set of multiple sequence alignments for any protein sequence dataset using standard alignment tools (Higgins et al., 1992; Corpet, 1988; Sievers et al., 2011), we use the existing multiple sequence alignments from the 32.0 release of Pfam.

2.2 Predicting Alignment Profiles

Once an MSA is built, it is common to fit a profile HMM, which model the probabilities of amino acids appearing in the columns of an MSA, as well as the probability of inserting additional amino acids between columns or deleting existing columns. It is common to use features derived from profile HMMs as input to structure prediction models, as these profiles contain information about the evolutionary history of a protein (Klausen et al., 2019). In particular, the emission probabilities give insight into which positions in the proteins are likely to mutate or remain constant over the course of evolution. This in turn illuminates which portions of the protein are critical for the protein's structure or function. For a review on profile HMMs, see Eddy (1998). We build profile HMMs from multiple sequence alignments using HMMER (Finn et al., 2011) with the default arguments.

Our proposed task is to predict a protein's profile HMM directly from it sequence. Formally, we represent a protein sequence x of length n as $x_1 x_2 \ldots x_n$ and the MSA it belongs to as a matrix:

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{km} \end{pmatrix}$$

where k is the number of sequences in the alignment and $m \geq n$ is the length of the alignment. Without loss of generality, we assume that x is the first sequence in the alignment; that is, there exists an injective map $g : [n] \mapsto [m]$ such that $i \leq g(i)$ and $x_i = a_{1g(i)}$ for all $i \in [n]$. We let $h : \{a_{ij} \in A\} \mapsto \{M, I, D\}$ be the MSA state function which maps amino acids to the three possible states in an MSA:

1. Match: a_{ij} is an amino acid that is related, evolutionarily or structurally, to other amino acids in column j.

2. Insertion: a_{ij} is an amino acid that is not related to other amino acids in its column, but is more likely the result of a mutation that inserted additional amino acids.

3. Deletion: a_{ij} is not an amino acid, but rather a column in which protein i is missing an amino acid where other proteins in the MSA have amino acids that are either matched or inserted.

A profile HMM built from this MSA is represented by the match state emissions $p_1^M p_2^M \ldots p_\ell^M$ and the insertion state emissions $p_1^I p_2^I \ldots p_\ell^I$, as well as an injective function $f : [\ell] \mapsto [m]$ which maps the indices of the profile back to the columns of the MSA. p_j^M and p_j^I are probability vectors of size S containing the probability of seeing each amino acid in column $f(j)$ in match or insertions states respectively:

$$\sum_{s=1}^{S} (p_j^M)_s = 1, (p_j^M)_s \geq 0 \text{ and } \sum_{s=1}^{S} (p_j^I)_s = 1, (p_j^I)_s \geq 0$$

for an amino acid alphabet of size S. In our experiments, we use the standard 20 amino acids during profile creation. Note also that f has a well-defined inverse $f^{-1} : [m] \mapsto [\ell]$.

We define a sequence of vector labels, $l_1 l_2 \ldots l_n$ associated with the input sequence x, defined as:

$$l_i(x) = \begin{cases} p^M_{f^{-1}(g(i))} & \text{if } h(a_{1g(i)}) = M \\ p^I_{f^{-1}(g(i))} & \text{if } h(a_{1g(i)}) = I \end{cases}$$

The $l_i(x)$ are well-defined: $h(a_{1g(i)}) \neq D, \forall i$ since $g(i)$ only maps to columns in the alignment where x contains amino acids. Given a network function F, our proposal task's loss function can be written as follows:

$$L_{\text{PP}}(x, \theta) = \frac{1}{n} \sum_{i=1}^{n} \sum_{s=1}^{S} l_{i,s}(x)(\log(l_{i,s}(x)) - \log(F_{i,s}(x; \theta))) = \frac{1}{n} \sum_{i=1}^{n} \text{KLDiv}(l_i(x), F_i(x; \theta))$$

where for $F_{i,s}(x; \theta)$ and $l_{i,s}$ the i indexes into the sequence position and the s indexes into the amino acid output probability. The process is described graphically in Figure 1. We contrast this loss with the standard masked language modeling objective:

$$L_{\text{MLM}}(x, \theta) = \frac{1}{n} \sum_{i \in \text{mask}} \sum_{s=1}^{S} L_{i,s}(x) \log(F_{i,s}(x; \theta)) = \frac{1}{n} \sum_{i \in \text{mask}} \text{CrossEntropy}(L_i(x), F_i(x; \theta))$$

for one-hot labels $L_{i,s}(x)$ that are equal to 1 if x_i is the sth amino-acid in the vocabulary, and 0 otherwise. We additionally introduce the joint loss:

$$L_{\text{JOINT}}(x, \theta, \lambda) = \lambda L_{\text{MLM}}(x, \theta) + (1 - \lambda)L_{\text{PP}}(x, \theta)$$

for a scaling parameter λ, which in practice we set such that $L_{\text{MLM}}(x, \theta) \approx L_{\text{PP}}(x, \theta)$ over the course of training.

We note here that our task L_{PP} is non-trivial. From an NLP perspective, it is akin to predicting a distribution over possible ways to rephrase a sentence while preserving its meaning from only the original sentence itself. This requires not only knowing which words carry the meaning of the sentence but also knowing the synonyms of these words in the context of that sentence. Doing so would require a significant understanding of language. As such, we believe our task encourages the transformer to learn about underlying protein biology more than simply predicting masked-out amino acids.

2.3 Data

To evaluate our pre-training task we make use of the TAPE benchmark: a set of five standardized protein sequence prediction tasks with associated datasets plus a large unlabeled pre-training dataset derived from Pfam (Rao et al., 2019). We build labels for the pre-training data set using the procedure described above. We then evaluate our pre-trained models on the five downstream TAPE tasks: secondary structure prediction (Klausen et al., 2019), contact prediction (AlQuraishi, 2019), remote homology detection (Hou et al., 2018), fluorescence prediction (Sarkisyan et al., 2016), and stability prediction (Rocklin et al., 2017), using the metrics specified by TAPE.

2.4 Hyperparameters and Training Details

For all experiments we train the default transformer architecture used by Rao et al. (2019), but we note that our pre-training task is not architecture-specific. We train three models with the three different objectives above. The profile prediction model used a learning rate of 0.00025, while the multi-task and masked language modeling models use a learning rate of 0.0001. These learning rates represented the largest learning rates that did not cause the model to diverge during the course of training, searching from 0.00001 in increments of 0.00005. All models were pre-trained for 34 epochs.[2] The learning rate uses a warm-up schedule and dynamic batch sizing, both of which are

---

[2]We cut off pre-training at 34 epochs due to time constraints. It is possible that further gains may result from continuing to pre-train for longer.

Pretraining Task	Secondary Structure			Contact Prediction
	CASP12	CB513	TS115	CASP12
Profile Prediction	**0.71**	**0.74**	**0.77**	**0.33**
Masked Language Modeling	0.67	0.72	0.75	0.24
Multi-Task	**0.71**	0.72	0.76	0.28
No Pretraining	0.67	0.64	0.68	0.05

Table 1: Comparing Downstream Performance on Structure Prediction

Pretraining Task	Fold	Superfamily	Family
Profile Prediction	**0.23**	**0.45**	**0.86**
Masked Language Modeling	0.14	0.16	0.45
Multi-Task	0.16	0.28	0.68
No Pretraining	0.05	0.05	0.21

Table 2: Comparing Downstream Performance on Homology Detection

described in Rao et al. (2019). Pre-training a single model took approximately two weeks with 8 NVIDIA Tesla V100 GPUs.

Training details for all downstream tasks follow the procedure laid out by Rao et al. (2019): namely, a learning rate of 0.0001 with linear warm-up schedule, the Adam optimizer and backpropagation through the entire pre-trained model. The downstream prediction heads all follow those in Rao et al. (2019), except for contact prediction which uses a single linear layer rather than a 30-layer convolutional architecture.[3]

3 Results

3.1 Comparing Pre-training Tasks

We first compare our pre-training task against masked language modeling and the multitask model which combines both tasks, keeping hyperparameters and architecture fixed. The results are shown in Tables 1, 2, and 3. For both structure prediction tasks—secondary structure and contact prediction—profile prediction pre-training outperforms multitasking, which in turn outperforms masked language modeling. All three tasks outperform the same model that was not pre-trained. Although it is not surprising that profile pre-training outperforms mask language modeling on structure prediction—namely because HMM profiles are known to contain information relevant to a protein's structure—the differences between the evaluated models are not large. This may mean that potentially more than just a new pre-training task is needed to continue to improve structure predictors, such as different architectures, or larger pre-training datasets (Elnaggar et al., 2020).

The remote homology detection task demonstrates the largest gap between profile prediction and mask language modeling. The model pre-trained with profile prediction is about 2 to 3 times more accurate than the model pre-trained using masked language modeling. The performance of the multitask model lies between that of the other two models and all three again outperform a randomly initialized model. This may be because HMM profiles also contain significant amounts of information about evolutionarily related proteins, which is closely related to the structural or functional groupings that a protein falls into.

We find the same pattern on the fluorescence task: profile prediction leads to the best test set performance, followed by multitasking, masked language modeling and no pre-training in that order. Finally, on the stability task, the masked language modeling model and the multitask model both outperform profile prediction. This may be because this task tests models' ability to generalize to proteins with a single amino acid difference from proteins in the training set—a task that masked language modeling is particularly suited for. Taken as a whole, these results indicate that there may not be a one-size-fits-all pre-training task for all downstream prediction tasks. Rather, it may be beneficial to tailor the pre-training task to the downstream task: for structure or evolutionary tasks,

---

[3]This new linear prediction head is the default in the current version of the `tape-proteins` package.

Pretraining Task	Downstream Task	
	Fluorescence	Stability
Profile Prediction	**0.38**	0.55
Masked Language Modeling	0.25	**0.63**
Multi-Task	0.28	**0.63**
No Pretraining	0.04	0.45

Table 3: Comparing Downstream Performance on Engineering Tasks

	Method	Structure	Evolutionary	Engineering	
		SS	Homology	Fluorescence	Stability
Pre-trained models from Rao et al. (2019)	Transformer	0.73	0.21	**0.68**	**0.73**
	LSTM	0.75	**0.26**	0.67	0.69
	ResNet	0.75	0.17	0.21	**0.73**
Baselines from Rao et al. (2019)	One-hot	0.69	0.09	0.14	0.19
	Alignment	**0.80**	0.09	N/A	N/A
	Bepler and Berger (2018)	0.73	0.17	0.33	0.64
	Alley et al. (2019)	0.73	0.23	0.67	**0.73**
	Lu et al. (2020)	0.70	0.13	**0.68**	0.68
Our models	Profile Prediction	0.74	0.23	0.38	0.55
	Masked Language Modeling	0.72	0.14	0.25	0.63
	Multi-Task	0.74	0.16	0.28	0.63

Table 4: Results on downstream supervised tasks

incorporating profile information may be beneficial, but for fine-grained engineering tasks, masked language modeling may be a better choice.

3.2 Comparing Against TAPE Benchmarks

In this section, we compare against the models presented in the original TAPE benchmark, as well as some previous literature that makes use of the TAPE benchmark. We omit the contact prediction task because the `tape-proteins` package significantly changed the downstream contact prediction architecture after the paper's original publication. We note that this was also done by previous work (Lu et al., 2020). For secondary structure, we present results from the CB513 test set. For remote homology detection we present results from the fold level prediction task. The results are presented in Table 4.

On the secondary structure task, we fall short of the NetsurfP2.0 model presented by Klausen et al. (2019), which is the alignment baseline from Rao et al. (2019). We also fall short of the LSTM and ResNet models from Rao et al. (2019), but outperform both the Transformer model as well as all previous work proposing protein-specific pre-training tasks (Bepler and Berger, 2018; Lu et al., 2020), and a recent paper employing an auto-regressive LSTM (Alley et al., 2019). On the remote homology task we outperformed all existing models except the TAPE benchmark's LSTM model and the LSTM presented by (Alley et al., 2019). We again note that we outperform the protein-specific pre-training tasks in Bepler and Berger (2018) and Lu et al. (2020). In the engineering tasks, all of our models are outperformed by existing work, including models from the original TAPE benchmark and the related works mentioned above.

We also note that we do not view most existing work as mutually exclusive with ours. Instead, we believe that combining our task with the architectures and pre-training tasks present in existing work may lead to further performance gains.

4 Conclusion

In this work, we produced a new pre-training task—profile prediction—which encourages a transformer to learn a distribution over evolutionarily-related protein sequences from a single input se-

quence. Our task alone outperforms masked language modeling on downstream tasks related to structure and homology detection. However, we do find that mask language modeling is still a useful pre-training task for engineering-related downstream tasks, namely, predicting intrinsic stability. We believe that our work shows that pre-training tasks borrowed from natural language processing models may not always be ideal for protein sequence models. Instead, we hope that these results inspire additional interest in developing biologically-informed protein sequence pre-training tasks. The combination of novel pre-training tasks, new architectures, and massively unlabeled datasets may be the key to eventually outperforming alignment-based predictors.

References

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. 2019. Unified rational protein engineering with sequence-only deep representation learning. *bioRxiv*, page 589333.

Mohammed AlQuraishi. 2019. Proteinnet: a standardized data set for machine learning of protein structure. *BMC bioinformatics*, 20(1):1–10.

Tristan Bepler and Bonnie Berger. 2018. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*.

UniProt Consortium. 2019. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515.

Florence Corpet. 1988. Multiple sequence alignment with hierarchical clustering. *Nucleic acids research*, 16(22):10881–10890.

James A Cuff and Geoffrey J Barton. 1999. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 34(4):508–519.

James A Cuff and Geoffrey J Barton. 2000. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins: Structure, Function, and Bioinformatics*, 40(3):502–511.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sean R. Eddy. 1998. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763.

Robert C Edgar and Serafim Batzoglou. 2006. Multiple sequence alignment. *Current opinion in structural biology*, 16(3):368–373.

Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al. 2019. The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, et al. 2020. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*.

Robert D Finn, Jody Clements, and Sean R Eddy. 2011. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37.

Desmond G Higgins, Alan J Bleasby, and Rainer Fuchs. 1992. Clustal v: improved software for multiple sequence alignment. *Bioinformatics*, 8(2):189–191.

Jie Hou, Badri Adhikari, and Jianlin Cheng. 2018. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303.

David T Jones. 1999. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202.

Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjærgaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Sønderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. 2019. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527.

Amy X. Lu, Haoran Zhang, Marzyeh Ghassemi, and Alan Moses. 2020. Self-supervised contrastive learning of protein representations by mutual information maximization. *bioRxiv*.

Jianzhu Ma, Sheng Wang, Zhiyong Wang, and Jinbo Xu. 2015. Protein contact prediction by integrating joint evolutionary coupling analysis and supervised learning. *Bioinformatics*, 31(21):3506–3513.

Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. 2020. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497*.

Ananthan Nambiar, Maeve Elizabeth Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. 2020. Transforming the language of life: Transformer neural networks for protein prediction tasks. *BioRxiv*.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. 2019. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9689–9701.

Alexander Rives, Siddharth Goyal, Joshua Meier, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2019. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, page 622803.

Gabriel J Rocklin, Tamuka M Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K Mulligan, Aaron Chevalier, et al. 2017. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175.

Burkhard Rost. 2001. Protein secondary structure prediction continues to rise. *Journal of structural biology*, 134(2-3):204–218.

Karen S Sarkisyan, Dmitry A Bolotin, Margarita V Meer, Dinara R Usmanova, Alexander S Mishin, George V Sharonov, Dmitry N Ivankov, Nina G Bozhanova, Mikhail S Baranov, Onuralp Soylemez, et al. 2016. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401.

Fabian Sievers, Andreas Wilm, David Dineen, Toby J Gibson, Kevin Karplus, Weizhong Li, Rodrigo Lopez, Hamish McWilliam, Michael Remmert, Johannes Söding, et al. 2011. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1):539.

Raymond C Stevens. 2003. The cost and value of three-dimensional protein structure. *Drug Discovery World*, 4(3):35–48.

Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. 2020. Udsmprot: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409.