Training, Evaluating, and Understanding Evolutionary Models for Protein Sequences

Roshan Rao

Electrical Engineering and Computer Sciences
University of California, Berkeley

Technical Report No. UCB/EECS-2022-1
http://www2.eecs.berkeley.edu/Pubs/TechRpts/2022/EECS-2022-1.html

January 8, 2022

Training, Evaluating, and Understanding Evolutionary Models for Protein Sequences

by

Roshan Maruthi Rao

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Computer Science

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor John Canny, Co-chair
Professor Pieter Abbeel, Co-chair
Professor Ian Holmes

Fall 2021

Training, Evaluating, and Understanding Evolutionary Models for Protein Sequences

Abstract

Training, Evaluating, and Understanding Evolutionary Models for Protein Sequences

by

Roshan Maruthi Rao

Doctor of Philosophy in Computer Science

University of California, Berkeley

Professor John Canny, Co-chair

Professor Pieter Abbeel, Co-chair

Novel protein sequences arise through mutation. These mutations may be deleterious, beneficial, or neutral; the effect of a mutation on an organism's evolutionary fitness is reflected in whether an organism survives long enough for its proteins to be sampled and deposited in a sequence database. Bioinformatics has long sought to use this evolutionary signal, commonly in the form of Multiple Sequence Alignments (MSAs), to make inferences as to the structure and function of novel proteins. With the advent of neural networks and self-supervised pretraining, a different approach emerged, where a large scale neural network could be pretrained using a language modeling objective to automatically produce informative features from an input protein sequence.

In this work, methods to train and evaluate protein language models on a common benchmark are introduced. Subsequently, the effects of increased model scaling, dataset preprocessing and training hyperparameters on the ability of transformers to learn protein contacts without supervision are explored. A novel method operating on MSAs instead of single sequences is then presented, and shown to achieve state-of-the-art performance on several downstream tasks. Finally, the utility of all of these methods in protein design is discussed.

To my family, Nagaraj, Suman, and Nisha Rao. Thank you for your unconditional support.

Contents

List of Figures

List of Tables

Acknowledgments

To start at the beginning, my first real research project was at the Space Telescope Science Institute, working with Antonella Nota and Perry Greenfield on converting images of galaxies into 3D-printable models. This was my first exposure to scientific computing. Thanks to both of you, and to the others on that project for inspiring the next 7-8 years of my research.

My first research project involving proteins actually began in undergrad, where I worked with Erik Sudderth and Jason Pacheco. The work introduced me to proteins, structure prediction, Bayesian modeling, and more. Thanks to Erik and Jason for being patient with me and guiding me along a wonderful project; it has clearly helped define this dissertation. I'm also sure Erik is thrilled to see the central role neural networks play in all my work.

I want to thank my advisors, John Canny and Pieter Abbeel, for their advice throughout my PhD. When I started working with John, I barely knew what a neural network was. It took me a year of fumbling from project to project before settling on a research direction pretty far outside of his interests. Pieter helped me settle on the vision of unsupervised learning and has always tried to be a sounding board for both research and career advice. As I wandered farther and farther away from their research areas, I wondered if I was going too far and if they would pull me back. They never did. Thanks for giving me the freedom to explore and build my own path, all while supporting the decisions I made.

When I decided I wanted to also work on proteins for my PhD, I struggled for several months to make any progress. It wasn't until I heard from Philippe Laban that there were two other PhD students interested in "BERT for proteins," and I met Nick Bhattacharya and Neil Thomas that things changed.

Neil brings this wonderful (and slightly chaotic) blend of energy and enthusiasm to all things. When I get lost in the weeds and forget how interesting this work is, Neil's joy can always remind me.

Nick is incredibly talented at understanding the interesting scientific questions in a dataset, and brings a very different perspective to our work. Our discussions have made the work in this dissertation better, and have made me a better scientist. During the pandemic, he's helped bring a more peaceful and meditative aspect to my life and is always available to talk.

Thank you both for helping me get started on this journey, and for making research so much fun. If only your names had slightly higher edit distance. Anyway, it wouldn't have been the same without you.

When the pandemic started, I began working with new people; Sergey Ovchinnikov, Joshua Meier, Tom Sercu, and Alexander Rives.

Sergey has been a wonderful researcher and mentor for the last year and a half. His deep knowledge of structural biology and his ability to quickly create experiments are inspiring. Simultaneously, he is continuously supportive and inclusive. I'll be very excited to see where he goes next.

Working with Alex, Josh, and Tom was synergistic. We cared about similar problems and thought about them in similar ways. When I pitched the MSA Transformer, they immediately bought into the vision. I was excited to see how the group changed as a result of the work

we did. I wish Josh the best of luck at Absci, and very much look forward to working with Tom and Alex in the future.

My friends at Berkeley have been very important, both to me and to this work. David Chan, Forrest Huang, Chandan Singh, and I met in the very first days of grad school. David is an incredibly technical person, with deep knowledge about computers, neural networks, and the best restaurants in the Bay. I've enjoyed learning from Forrest about sketching programs and sharing in his love of whiskey. And of course, I can't forget about Philippe Laban, whose love of pastries brought the lab together, and without whom my first project with Nick and Neil would never have come about.

I've known my roommates, Hayley Bounds, Ian Stewart, and Celia Ford since before coming to Berkeley; Hayley and Celia I've known since my first year of college. Living with them has added a degree of stability and comfort to an otherwise chaotic and isolating time. I can always count on them to celebrate or commiserate. I hope one day to read Celia's NPR biography as a pole-dancing, bass-playing science journalist while sipping on one of Hayley's cocktails and feeding their rats.

Finally, my family has been with me for my entire journey, always cheering my successes and listening to my frustrations. Anna, Amma, and Nisha, you've helped me so much throughout my life, and especially in these last few years. I am looking forward to being closer to home.

Chapter 1

Introduction

Evolutionary modeling of protein sequences has been a mainstay of computational biology for decades. This approach takes advantage of a particular filtering function - an organism's reproductive fitness - which influences the likelihood that particular protein sequences will survive to be sequenced and deposited in large protein databases.

The typical evolutionary modeling pipeline is as follows: take a query sequence, search for similar sequences in a protein database, align the resulting sequences, and build a model on top of the result. Even very simple models of aligned sequences can be powerful. Consider an independent-sites model (also known as a position-specific scoring matrix or PSSM). Residues that are highly conserved (i.e. have low entropy) under this model are likely to be critical to the protein's function. For example, these may belong to an active site of an enzyme where mutations to this site will cause the catalysis to fail.

Alternately, consider a pair of interacting residues. When a mutation occurs in one of these residues, the interaction between the residues must be preserved in order to maintain the protein's structure and function. Thus, only certain combinations of amino acids would be allowed at these positions. A pairwise-sites model (also known as a Potts model) can pick up on this interaction, thus providing evidence for the likely structure of the query protein.

While this pipeline can provide a great deal of information, there is room for improvement. The pipeline works well when the query sequence is similar to many existing sequences in the target database - Potts models, for example, generally require at least 1000 similar sequences to reach maximum accuracy. It can also be slow; each new query sequence requires a separate database search and alignment, a process which can take anywhere from several minutes to hours. Finally, it relies on a set of semi-independent components for each step instead of being end-to-end. Each of these components has been tuned extensively over the years, but lessons from computer vision, natural language processing (NLP), and other fields have shown that end-to-end learning can surpass the best designed features.

Into these gaps comes the field of unsupervised language modeling. The new approaches, popularized by works such as ELMo [11] and BERT [12] in natural language processing, train large scale neural networks on large databases of sequences without requiring manual supervision. Swapping out a database of English text for one of protein sequences requires

Title	Venue	Author List
Evaluating Protein Transfer Learning with TAPE	Neural Information Processing Systems, 2019	Roshan Rao[1], Nicholas Bhattacharya[1], Neil Thomas[1], Yan Duan, Xi Chen, John Canny, Pieter Abbeel, Yun S. Song
Transformer protein language models are unsupervised structure learners	International Conference on Learning Representations, 2021	Roshan Rao, Joshua Meier, Tom Sercu, Sergey Ovchinnikov, Alexander Rives
MSA Transformer	International Conference on Machine Learning, 2021	Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, Alexander Rives
Language models enable zero-shot prediction of the effects of mutations on protein function	Neural Information Processing Systems, 2021	Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, Alexander Rives

Table 1.1: List of prior work included in this dissertation, along with publication venues and coauthors.

essentially no changes to the model architecture or training procedure, making them a natural fit for developing neural models of proteins. Unsupervised language models promise to provide more informative features, with minimal supervision, all for the cost of a GPU forward pass.

This thesis focuses on the design and training of unsupervised language models for protein sequences, seeking to improve the learned features and to replace components or the entirety of the existing evolutionary modeling pipeline.

As an equally important contribution, it also draws connections between language models and existing independent- and pairwise-site models, and evaluates whether language models actually fulfill the promised goals.

1.1 Statement on Prior Publications

Much of the content presented in this dissertation has been previously published in different venues. See Table 1.1 for a list of prior works, including publication venues and coauthors. Computational biology is an inherently collaborative field. I was a key contributor in all included work and designed, wrote code for, ran, and analyzed experiments for each

[1]Denotes equal contribution.

publication. However, none of these publications would have been possible without the work from my coauthors.

Nicholas Bhattacharya and Neil Thomas were equal contributors to the work described in Chapter 2 and were involved in the selection of datasets, training and analysis of models, and paper writing. Yan Duan, Xi Chen, and Yun Song advised in the design of experiments and in framing the paper.

Chapters 3-5 describe work done in collaboration with the protein group at Facebook AI Research and with Sergey Ovchinnikov. The work in these chapters was greatly aided by the data collection, model scaling, and model evaluation done by Joshua Meier, Jason Liu, and Robert Verkuil. In addition, the work in Chapter 5 was spearheaded by Joshua, who designed the experimental protocols and ran experiments with the ESM-1v models. Sergey Ovchinnikov provided advice for work described in Chapter 3, and Tom Sercu and Alexander Rives provided advice throughout.

Finally, my advisors John Canny and Pieter Abbeel directly contributed to the framing and conceptualization of Chapters 2 and 4, and have generally shaped my work throughout my PhD.

As these Chapters were the result of joint effort, Chapters 2-5 use the pronoun 'we'. All coauthors have provided written consent for the inclusion of their work in this dissertation.

1.2 Overview of the Thesis

This dissertation attempts to answer three key questions regarding language modeling for protein sequences:

1. Do standard approaches to unsupervised learning in NLP learn biologically relevant features?
2. How can we tailor the data, model, and tasks used to train unsupervised models for proteins?
3. Can large scale unsupervised models for protein sequences be made useful for protein design?

Chapter 2 focuses on the first question. Introductions to multiple sequence alignments and typical protein modeling tasks are provided for readers more familiar with machine learning, while introductions to basic deep learning architectures (such as the Transformer [13]) are provided for those more familiar with biology or classical machine learning methods. Comparisons between language models and independent-sites models are introduced, and independent-sites models are shown to produce significantly better features for structure prediction based tasks.

Chapter 3 looks at both the first and second questions by examining *unsupervised* contact prediction with Transformer based language models. It explores representations extracted from Transformer attention maps, showing that these representations carry a great deal of information about protein structure, despite not being explicitly supervised to predict structure. Additionally, it explores the effects of model scale and hyperparameters on the resulting features.

Chapter 4 dives further into the second question by adapting a language model to take a multiple sequence alignment (MSA) as input. The chapter discusses necessary changes to model architecture, training, and inference when working with MSAs.

Chapter 5 begins to answer the third question, specifically in the context of predicting the effects of single mutations. The role of training data, model ensembling, and MSA selection are examined.

Chapter 6 concludes the dissertation, raises new questions, and discusses future work.

1.3 Research Contributions

The primary contributions in this thesis are as follows:

- **A benchmark dataset for evaluating different protein language models on the same tasks.** Chapter 2 introduces the Tasks Assessing Protein Embeddings (TAPE), suite of five biologically relevant downstream tasks that can be used to evaluate different protein language models on an even playing field. These datasets are made available for free in multiple easy-to-download formats.
- **Publicly available library implementing multiple protein language models.** As part of work in Chapter 2, multiple protein language models were implemented in PyTorch [14] using a HuggingFace-style API [15]. The resulting GitHub repository has been widely used in the field.
- **A method for predicting protein contacts from pretrained language models with little-to-no supervision.** Chapter 3 shows that it is possible to predict protein contacts from attention maps of transformers trained with masked language modeling, despite the fact there is no explicit structural supervision. A sparse subset of attention heads learn to predict protein contacts. A simple logistic regression, trained using between 1-20 protein structures, can select the attention heads which perform contact prediction. The trained logistic regressions are made available for download on GitHub.
- **A novel architecture for learning representations of MSAs.** Chapter 4 introduces the MSA Transformer, which extends protein language models to MSA-based language

models. This uses axial attention and other architectural modifications to achieve state-of-the-art performance for unsupervised contact prediction. The model is also analyzed and compared to pairwise-sites models and single sequence language models.

- **State-of-the-art variant effect prediction from single sequence or from MSAs.** Chapter 5 explores the use of single-sequence language models and MSA-based language models for variant effect prediction and shows that single-sequence language models perform as well as prior methods, while being orders of magnitude faster, and MSA-based language models outperform prior approaches. The question of what makes the optimal data for training these models is explored, and ensembles of the best performing models are provided on GitHub.

In addition to the concrete contributions above, the thesis seeks to help the reader place protein language models in the larger context of evolutionary modeling and to develop intuitions about why protein language models work and when they are likely to work well.

Chapter 2

Evaluating Protein Language Models with TAPE

2.1 Introduction

New sequencing technologies have led to an explosion in the size of protein databases over the past decades. These databases have seen exponential growth, with the total number of sequences doubling every two years [16]. Obtaining meaningful labels and annotations for these sequences requires significant investment of experimental resources, as well as scientific expertise, resulting in an exponentially growing gap between the size of protein sequence datasets and the size of annotated subsets. Billions of years of evolution have sampled the portions of protein sequence space that are relevant to life, so large unlabeled datasets of protein sequences are expected to contain significant biological information [17–19]. Advances in natural language processing (NLP) have shown that self-supervised learning is a powerful tool for extracting information from unlabeled sequences [11, 12, 20], which raises a tantalizing question: can we adapt NLP-based techniques to extract useful biological information from massive sequence datasets?

To help answer this question, we introduce the Tasks Assessing Protein Embeddings (TAPE), which to our knowledge is the first attempt at systematically evaluating semi-supervised learning on protein sequences. TAPE includes a set of five biologically relevant supervised tasks that evaluate the performance of learned protein embeddings across diverse aspects of protein understanding.

We choose our tasks to highlight three major areas of protein biology where self-supervision can facilitate scientific advances: structure prediction, detection of remote homologs, and protein engineering. We construct data splits to simulate biologically relevant generalization, such as a model's ability to generalize to entirely unseen portions of sequence space or to finely resolve small portions of sequence space. Improvement on these tasks range in application, including designing new antibodies [21], improving cancer diagnosis [22], and finding new antimicrobial genes hiding in the so-called "Dark Proteome": tens of millions of sequences

with no labels where existing techniques for determining protein similarity fail [23].

We assess the performance of three representative models (recurrent, convolutional, and attention-based) that have performed well for sequence modeling in other fields to determine their potential for protein learning. We also compare two previously proposed semi-supervised models (Bepler and Berger [24], Alley et al. [25]). With our benchmarking framework, these models can be compared directly to one another for the first time.

We show that self-supervised pretraining improves performance for almost all models on all downstream tasks. Interestingly, performance for each architecture varies significantly across tasks, highlighting the need for a multi-task benchmark such as ours. We also show that features from alignment-based independent sites models (also know as position-speciic scoring matrices or PSSMs) [26–29] outperform features learned via self-supervision on secondary structure and contact prediction, while learned features perform significantly better on remote homology detection.

Our results demonstrate that self-supervision for proteins is promising but considerable improvements need to be made before self-supervised models can achieve breakthrough performance. All code and data for TAPE are publicly available[1], and we encourage members of the machine learning community to participate in these exciting problems.

2.2 Background

Protein Terminology

Proteins are linear chains of amino acids connected by covalent bonds. We encode amino acids in the standard 25-character alphabet, with 20 characters for the standard amino acids, 2 for the non-standard amino acids selenocysteine and pyrrolysine, 2 for ambiguous amino acids, and 1 for when the amino acid is unknown [16, 30]. Throughout this paper, we represent a protein x of length L as a sequence of discrete amino acid characters $(x_1, x_2, \ldots, x_L)$ in this fixed alphabet.

Beyond its encoding as a sequence $(x_1, \ldots, x_L)$, a protein has a 3D molecular structure. The different levels of protein structure include *primary* (amino acid sequence), *secondary* (local features), and *tertiary* (global features). Understanding how primary sequence folds into tertiary structure is a fundamental goal of biochemistry [17]. Proteins are often made up of a few large *protein domains*, sequences that are evolutionarily conserved, and as such have a well-defined fold and function.

Evolutionary relationships between proteins arise because organisms must maintain certain functions, such as replicating DNA, as they evolve. Evolution has selected for proteins that are well-suited to these functions. Though structure is constrained by evolutionary pressures, sequence-level variation can be high, with very different sequences having similar structure [31]. Two proteins that share a common evolutionary ancestor are called *homologs*. Homologous proteins may have very different sequences if they diverged in the distant past.

[1] `https://github.com/songlab-cal/tape`

Quantifying these evolutionary relationships is very important for preventing undesired information leakage between data splits. We mainly rely on *sequence identity*, which measures the percentage of exact amino acid matches between aligned subsequences of proteins [32]. For example, filtering at a 25% sequence identity threshold means that no two proteins in the training and test set have greater than 25% exact amino acid matches. Other approaches besides sequence identity filtering also exist, depending on the generalization the task attempts to test [33].

Modeling Evolutionary Relationships with Sequence Alignments

The key technique for modeling sequence relationships in computational biology is alignment [26, 29, 34, 35]. Given a database of proteins and a query protein at test-time, an alignment-based method uses either carefully designed scoring systems [34] or Hidden Markov Models (HMMs) [29] to align the query protein against all proteins in the database. Good alignments give information about local perturbations to the protein sequence that may preserve, for example, function or structure. The distribution of aligned residues at each position is also an informative representation of each residue that can be fed into downstream models.

Semi-supervised Learning

The fields of computer vision and natural language processing have been dealing with the question of how to learn from unlabeled data for years [36]. Images and text found on the internet generally lack accompanying annotations, yet still contain significant structure. Semi-supervised learning tries to jointly leverage information in the unlabeled and labeled data, with the goal of maximizing performance on the supervised task. One successful approach to learning from unlabeled examples is *self-supervised learning*, which in NLP has taken the form of next token prediction [11], masked token prediction [12], and next sentence classification [12]. Analogously, there is good reason to believe that unlabelled protein sequences contain significant information about their structure and function [17, 19]. Since proteins can be modeled as sequences of discrete tokens, we test both next token and masked token prediction for self-supervised learning.

2.3 Related Work

The most well-known protein modeling benchmark is the Critical Assessment of Structure Prediction (CASP) [37], which focuses on structure modeling. Each time CASP is held, the test set consists of new experimentally validated structures which are held under embargo until the competition ends. This prevents information leakage and overfitting to the test set. The recently released ProteinNet [38] provides easy to use, curated train/validation/test splits for machine learning researchers where test sets are taken from the CASP competition and sequence identity filtering is already performed. We take the contact prediction task from

ProteinNet. However, we believe that structure prediction alone is not a sufficient benchmark for protein models, so we also use tasks not included in the CASP competition to give our benchmark a broader focus.

Semi-supervised learning for protein problems has been explored for decades, with lots of work on kernel-based pretraining [39, 40]. These methods demonstrated that semi-supervised learning improved performance on protein network prediction and homolog detection, but couldn't scale beyond hundreds of thousands of unlabeled examples. Recent work in protein representation learning has proposed a variety of methods that apply NLP-based techniques for transfer learning to biological sequences [24, 25, 41–43]. In a related line of work, Riesselman et al. [2] trained Variational Auto Encoders on aligned families of proteins to predict the functional impact of mutations. Alley et al. [25] also try to combine self-supervision with alignment in their work by using alignment-based querying to build task-specific pretraining sets.

Due to the relative infancy of protein representation learning as a field, the methods described above share few, if any, benchmarks. For example, both Rives et al. [43] and Bepler and Berger [24] report transfer learning results on secondary structure prediction and contact prediction, but they differ significantly in test set creation and data-splitting strategies. Other self-supervised work such as Alley et al. [25] and Yang et al. [44] report protein engineering results, but on different tasks and datasets. With such varied task evaluation, it is challenging to assess the relative merits of different self-supervised modeling approaches, hindering efficient progress.

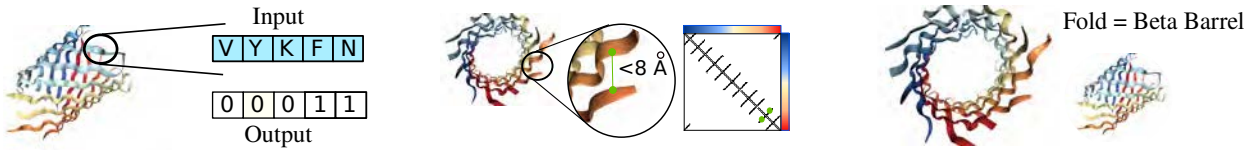
2.4 Datasets

Here we describe our unsupervised pretraining and supervised benchmark datasets. To create benchmarks that test generalization across large evolutionary distances and are useful in real-life scenarios, we curate specific training, validation, and test splits for each dataset. Producing the data for these tasks requires significant effort by experimentalists, database managers, and others. Following similar benchmarking efforts in NLP [45], we describe a set of citation guidelines in our repository[2] to ensure these efforts are properly acknowledged.

Unlabeled Sequence Dataset

We use Pfam [46], a database of thirty-one million protein domains used extensively in bioinformatics, as the pretraining corpus for TAPE. Sequences in Pfam are clustered into evolutionarily-related groups called *families*. We leverage this structure by constructing a test set of fully heldout families (see Appendix A.5 for details on the selected families), about 1% of the data. For the remaining data we construct training and test sets using a random 95/5% split. Perplexity on the uniform random split test set measures in-distribution generalization,

[2] `https://github.com/songlab-cal/tape#citation-guidelines`



(a) Secondary Structure (b) Contact Prediction (c) Remote Homology

Figure 2.1: Structure and Annotation Tasks on protein KgdM Porin (pdbid: 4FQE). (a) Viewing this Porin from the side, we show secondary structure, with the input amino acids for a segment (blue) and corresponding secondary structure labels (yellow and white). (b) Viewing this Porin from the front, we show a contact map, where entry i, j in the matrix indicates whether amino acids at positions i, j in the sequence are within 8 angstroms of each other. In green is a contact between two non-consecutive amino acids. (c) The fold-level remote homology class for this protein.

while perplexity on the heldout families test set measures out-of-distribution generalization to proteins that are less evolutionarily related to the training set.

Supervised Datasets

We provide five biologically relevant downstream prediction tasks to serve as benchmarks. We categorize these into structure prediction, evolutionary understanding, and protein engineering tasks. The datasets vary in size between 8,000-50,000 training examples (see Table A.1 for sizes of all training, validation and test sets). Further information on data processing, splits and experimental challenges is in Appendix A.1. For each task we provide:

(Definition) A formal definition of the prediction problem, as well as the source of the data.
(Impact) The impact of improving performance on this problem.
(Generalization) The type of understanding and generalization desired.
(Metrics) The metric reported in Table 2.2 to report results and additional metrics presented in Appendix A.8

Task 1: Secondary Structure (SS) Prediction (Structure Prediction Task)
(Definition) Secondary structure prediction is a sequence-to-sequence task where each input amino acid x_i is mapped to a label $y_i \in \{\text{Helix}(H), \text{Strand}(E), \text{Other}(C)\}$. See Fig. 2.1a for illustration. The data are from Klausen et al. [47].
(Impact) SS is an important feature for understanding the function of a protein, especially if the protein of interest is not evolutionarily related to proteins with known structure [47]. SS prediction tools are very commonly used to create richer input features for higher-level models [48].

(Generalization) SS prediction tests the degree to which models learn local structure. Data splits are filtered at 25% sequence identity to test for broad generalization.

(Metrics) We report accuracy on a per-amino acid basis on the CB513 [49] dataset. We further report three-way and eight-way classification accuracy for the test sets CB513, CASP12, and TS115.

Task 2: Contact Prediction (Structure Prediction Task)

(Definition) Contact prediction is a pairwise amino acid task, where each pair x_i, x_j of input amino acids from sequence x is mapped to a label $y_{ij} \in \{0, 1\}$, where the label denotes whether the amino acids are "in contact" (< 8Å apart) or not. See Fig. 2.1b for illustration. The data are from the ProteinNet dataset [38].

(Impact) Accurate contact maps provide powerful global information; e.g., they facilitate robust modeling of full 3D protein structure [50]. Of particular interest are medium- and long-range contacts, which may be as few as twelve sequence positions apart, or as many as hundreds apart.

(Generalization) The abundance of medium- and long-range contacts makes contact prediction an ideal task for measuring a model's understanding of global protein context. We select the data splits that was filtered at 30% sequence identity to test for broad generalization.

(Metrics) We report precision of the $L/5$ most likely contacts for medium- and long-range contacts on the ProteinNet CASP12 test set, which is a standard metric reported in CASP [37]. We further report Area under PR Curve and Precision at L, $L/2$, and $L/5$ for short-range, medium-range and long-range contacts in the supplement.

Task 3: Remote Homology Detection (Evolutionary Understanding Task)

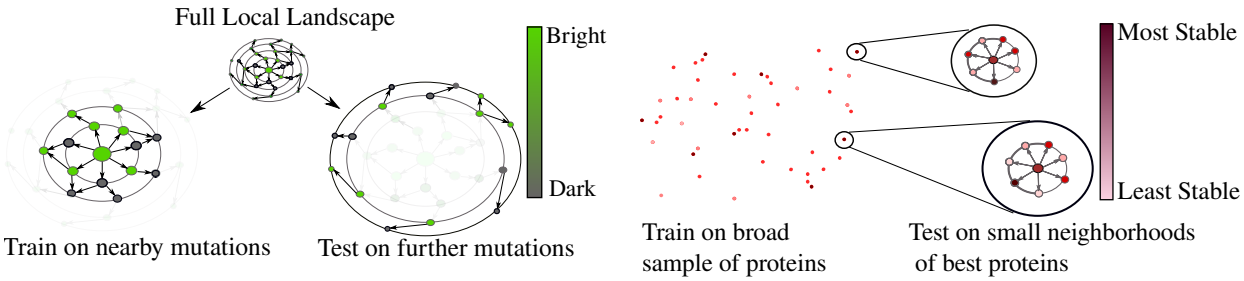
(Definition) This is a sequence classification task where each input protein x is mapped to a label $y \in \{1, \ldots, 1195\}$, representing different possible protein folds. See Fig. 2.1c for illustration. The data are from Hou et al. [51].

(Impact) Detection of remote homologs is of great interest in microbiology and medicine; e.g., for detection of emerging antibiotic resistant genes [52] and discovery of new CAS enzymes [53].

(Generalization) Remote homology detection measures a model's ability to detect structural similarity across distantly related inputs. We hold out entire evolutionary groups from the training set, forcing models to generalize across large evolutionary gaps.

(Metrics) We report overall classification accuracy on the fold-level heldout set from Liu et al. [53]. We further report top-one and top-five accuracy for fold-level, superfamily-level and family-level holdout sets in the supplement.

Task 4: Fluorescence Landscape Prediction (Protein Engineering Task)



(a) Fluorescence

(b) Stability

Figure 2.2: Protein Engineering Tasks. In both tasks, a parent protein p is mutated to explore the local landscape. As such, dots represent proteins and directed arrow $x \to y$ denotes that y has exactly one more mutation than x away from parent p. (a) The Fluorescence task consists of training on small neighborhood of the parent green fluorescent protein (GFP) and then testing on a more distant proteins. (b) The Stability task consists of training on a broad sample of proteins, followed by testing on one-mutation neighborhoods of the most promising sampled proteins.

(Definition) This is a regression task where each input protein x is mapped to a label $y \in \mathbb{R}$, corresponding to the log-fluorescence intensity of x. See Fig. 2.2a for illustration. The data are from Sarkisyan et al. [54].

(Impact) For a protein of length L, the number of possible sequences m mutations away is $O(L^m)$, a prohibitively large space for exhaustive search via experiment, even if m is modest. Moreover, due to epistasis (second- and higher-order interactions between mutations at different positions), greedy optimization approaches are unlikely to succeed. Accurate computational predictions could allow significantly more efficient exploration of the landscape, resulting in better optima. Machine learning methods have already seen some success in related protein engineering tasks [55].

(Generalization) The fluorescence prediction task tests the model's ability to distinguish between very similar inputs, as well as its ability to generalize to unseen combinations of mutations. The train set is a Hamming distance-3 neighborhood of the parent green fluorescent protein (GFP), while the test set has variants with four or more mutations. Hamming distance is measured at the amino acid level.

The choice of Hamming distance between amino acids does not always reflect evolution, since not all proteins at the same Hamming distance correspond to equal "evolutionary" distance in the sense of number of nucleotide substitutions. Since we are trying to highlight the protein engineering setting, we believe that this is an important feature of the Fluorescence task. Our goal is to test the models' ability to accurately predict phenotype as a function of an input molecule (e.g. one presented by a protein designer)

(Metrics) We report Spearman's ρ (rank correlation coefficient) on the test set. We further

Table 2.1: Language modeling metrics: Language Modeling Accuracy (Acc), Perplexity (Perp) and Exponentiated Cross-Entropy (ECE)

	Random Families			Heldout Families			Heldout Clans		
	Acc	Perp	ECE	Acc	Perp	ECE	Acc	Perp	ECE
Transformer	**0.45**	**8.89**	**6.01**	**0.35**	**11.77**	**8.87**	**0.28**	**13.54**	10.76
LSTM	0.40	**8.89**	6.94	0.24	13.03	12.73	0.13	15.36	16.94
ResNet	0.41	10.16	6.86	0.31	13.19	9.77	**0.28**	13.72	**10.62**
Bepler [24]	0.28	11.62	10.17	0.19	14.44	14.32	0.12	15.62	17.05
Alley [25]	0.32	11.29	9.08	0.16	15.53	15.49	0.11	16.69	17.68
Random	0.04	25	25	0.04	25	25	0.04	25	25

report MSE and Spearman's ρ for the full test set, only bright proteins, and only dark proteins in the supplement.

Task 5: Stability Landscape Prediction (Protein Engineering Task)

(Definition) This is a regression task where each input protein x is mapped to a label $y \in \mathbb{R}$ measuring the most extreme circumstances in which protein x maintains its fold above a concentration threshold (a proxy for intrinsic stability). See Fig. 2.2b for illustration. The data are from Rocklin et al. [56].

(Impact) Designing stable proteins is important to ensure, for example, that drugs are delivered before they are degraded. More generally, given a broad sample of protein measurements, finding better refinements of top candidates is useful for maximizing yield from expensive protein engineering experiments.

(Generalization) This task tests a model's ability to generalize from a broad sampling of relevant sequences and to localize this information in a neighborhood of a few sequences, inverting the test-case for fluorescence above. The train set consists of proteins from four rounds of experimental design, while the test set contains Hamming distance-1 neighbors of top candidate proteins.

(Metrics) We report Spearman's ρ on the test set. In the supplement we also assess classification of a mutation as stabilizing or non-stabilizing. We report Spearman's ρ and accuracy for this task broken down by protein topology in the supplement.

2.5 Models and Experimental Setup

Losses: We examine two self-supervised losses that have seen success in NLP. The first is *next-token prediction* [57], which models $p(x_i \mid x_1, \ldots, x_{i-1})$. Since many protein tasks are sequence-to-sequence and require bidirectional context, we apply a variant of next-token prediction which additionally trains the reverse model, $p(x_i \mid x_{i+1}, \ldots, x_L)$, providing full

context at each position (assuming a Markov sequence). The second is *masked-token prediction* [12], which models $p(x_{\text{masked}} \mid x_{\text{unmasked}})$ by replacing the value of tokens at multiple positions with alternate tokens.

Protein-specific loss: In addition to self-supervised algorithms, we explore another protein-specific training procedure proposed by Bepler and Berger [24]. They suggest that further *supervised* pretraining of models can provide significant benefits. In particular, they propose supervised pretraining on contact prediction and remote homology detection, and show it increases performance on secondary structure prediction. Similar work in computer vision has shown that supervised pretraining can transfer well to other tasks, making this a promising avenue of exploration [58].

Architectures and Training: We implement three architectures: an LSTM [59], a Transformer [13], and a dilated residual network (ResNet) [60]. We use a 12-layer Transformer with a hidden size of 512 units and 8 attention heads, leading to a 38M-parameter model. Hyperparameters for the other models were chosen to approximately match the number of parameters in the Transformer. Our LSTM consists of two three-layer LSTMs with 1024 hidden units corresponding to the forward and backward language models, whose outputs are concatenated in the final layer, similar to ELMo [11]. For the ResNet we use 35 residual blocks, each containing two convolutional layers with 256 filters, kernel size 9, and dilation rate 2. We chose these hyperparameters based on common choices from the literature. Our supervised tasks are of similar size to most of those in the SuperGLUE [45] benchmark, which has been instrumental in demonstrating the success of self-supervision in NLP. Since the models that were applied to GLUE have tens to hundreds of millions of parameters, we chose to make our models roughly the same size. See Appendix A.7 for model size ablation experiments. See Appendix A.2 for details of how these pretrained models are fed into downstream tasks.

In addition, we benchmark two previously proposed architectures that differ significantly from the three above. The first, proposed by Bepler and Berger [24], is a two-layer bidirectional language model, similar to the LSTM discussed above, followed by three 512 hidden unit bidirectional LSTMs. The second, proposed by Alley et al. [25], is a unidirectional mLSTM [61] with 1900 hidden units. Details on implementing and training these architectures can be found in the original papers.

The Transformer and ResNet are trained with masked-token prediction, while the LSTM is trained with next-token prediction. Both Alley et al. [25] and Bepler and Berger [24] are trained with next-token prediction. All self-supervised models are trained on four NVIDIA V100 GPUs for one week.

Baselines: We evaluate learned features against two baseline featurizations. The first is a one-hot encoding of the input amino acid sequence, which provides a simple baseline. Most current state-of-the-art algorithms for protein classification and regression take advantage

of alignment or HMM-based inputs (see Section 2.2). Alignments can be transformed into various features, such as mutation probabilities [51] or the HMM state-transition probabilities [47] for each amino acid position. These are concatenated to the one-hot encoding of the amino acid to form another baseline featurization. For our baselines we use alignment-based inputs that vary per task depending on the inputs used by the current state-of-the-art method. See Appendix A.3 for details on the alignment-based features used for each task. We do not use alignment-based inputs for protein engineering tasks. Proteins in the engineering datasets differ by only a single amino acid, while alignment-based methods search for proteins with high sequence identity, so alignment-based methods return the same set of features for all proteins we wish to distinguish between.

Experimental Setup: The goal of our experimental setup is to systematically compare all featurizations. For each task we select a particular supervised architecture, drawing from state-of-the-art where available, and make sure that fine-tuning on all language models is identical. See Appendix A.3 for details on supervised architectures and training.

2.6 Results

Table 2.1 contains accuracy, perplexity, and exponentiated cross entropy (ECE) on the language modeling task for the five architectures we trained with self-supervision as well as a random model baseline. We report metrics on both the random split and the fully heldout families. Supervised LSTM metrics are reported after language modeling pretraining, but before supervised pretraining. Heldout family accuracy is consistently lower than random-split accuracy, demonstrating a drop in the out-of-distribution generalization ability. Note that although some models have lower perplexity than others on both random-split and heldout sets, this lower perplexity does not necessarily correspond to better performance on downstream tasks. This replicates the finding in Rives et al. [43].

Table 2.2 contains results for all benchmarked architectures and training procedures on all downstream tasks in TAPE. We report accuracy, precision, or Spearman's ρ, depending on the task, so higher is always better and each metric has a maximum value of 1.0. See Section 2.4 for the metric reported in each task. Detailed results and metrics for each task are in Appendix A.8.

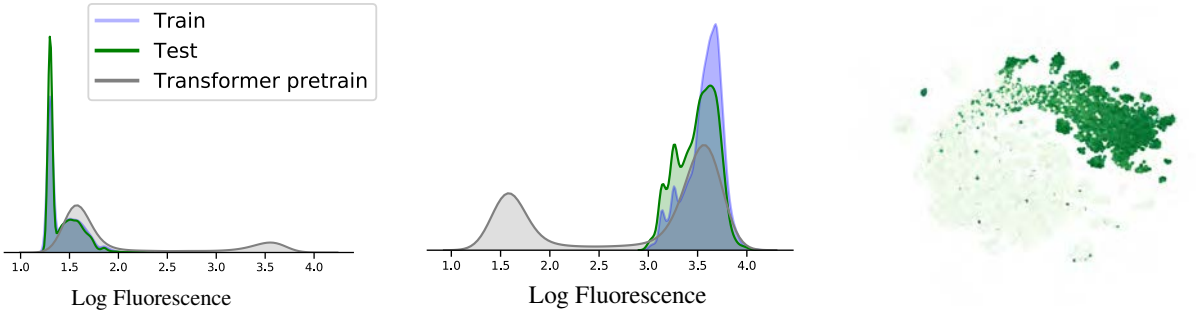
We see from Table 2.2 that self-supervised pretraining improves overall performance across almost all models and all tasks. Further analysis reveals aspects of these tasks with room for significant improvement. In the fluorescence task, the distribution is bimodal with a mode of bright proteins and a mode of dark proteins (see Fig. 2.3). Since one goal of using machine learning models in protein engineering is to screen potential variants, it is important for these methods to successfully distinguish between beneficial and deleterious mutations. Fig. 2.3 shows that the model does successfully perform some clustering of fluorescent proteins, but that many proteins are still misclassified.

Table 2.2: Results on downstream supervised tasks

Method		Structure		Evolutionary	Engineering	
		SS	Contact	Homology	Fluorescence	Stability
No Pretrain	Transformer	0.70	0.32	0.09	0.22	-0.06
	LSTM	0.71	0.19	0.12	0.21	0.28
	ResNet	0.70	0.20	0.10	-0.28	0.61
Pretrain	Transformer	0.73	0.36	0.21	**0.68**	**0.73**
	LSTM	0.75	0.39	**0.26**	0.67	0.69
	ResNet	0.75	0.29	0.17	0.21	**0.73**
	Bepler [24]	0.73	0.40	0.17	0.33	0.64
	Alley [25]	0.73	0.34	0.23	0.67	**0.73**
Baseline	One-hot	0.69	0.29	0.09	0.14	0.19
	Alignment	**0.80**	**0.64**	0.09	N/A	N/A

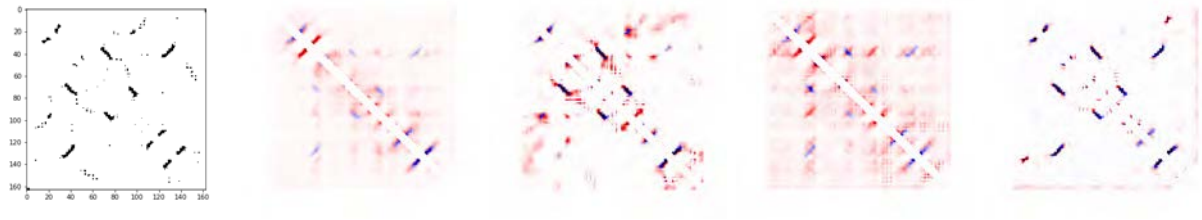
For the stability task, to identify which mutations a model believes are beneficial, we use the parent protein as a decision boundary and label a mutation as beneficial if its predicted stability is higher than the parent's predicted stability. We find that our best pretrained model achieves 70% accuracy in making this prediction while our best non-pretrained model achieves 68% accuracy (see Table A.9 for full results). Improving the ability to distinguish beneficial from deleterious mutations would make these models much more useful in real protein engineering experiments.

In the contact prediction task, long-range contacts are of particular interest and can be



(a) Dark Mode (b) Bright Mode (c) Embedding t-SNE

Figure 2.3: Distribution of training, test, and pretrained Transformer predictions on the dark and bright modes, along with t-SNE of pretrained Transformer protein embeddings colored by log-fluorescence.



(a) True Contacts (b) LSTM (c) LSTM Pretrain (d) One Hot (e) Alignment

Figure 2.4: Predicted contacts for chain 1A of a Bacterioferritin comigratory protein (pdbid: 3GKN). Blue indicates true positive contacts while red indicates false positive contacts. Darker colors represent more certainty from the model.

hundreds of positions apart. Fig. 2.4 shows the predictions of several models on a protein where the longest range contact occurs between the 8th and 136th amino acids. Pretraining helps the model capture more long-range information and improves the overall resolution of the predicted map. However, the hand-engineered alignment features result in a much sharper map, accurately resolving more long-range contacts. This increased specificity is highly relevant in structure prediction pipelines [50, 62] and highlights a clear challenge for pretraining.

2.7 Discussion

Comparison to state of the art. As shown in Table [47] achieves 85% accuracy on the CB513 [49] secondary structure dataset, compared to our best model's 75% accuracy, RaptorX [63] achieves 0.69 precision at $L/5$ on CASP12 contact prediction, compared to our best model's 0.49, and DeepSF [51] achieves 41% accuracy on remote homology detection compared to our best model's 26%.

Need for multiple benchmark tasks. Our results support our hypothesis that multiple tasks are required to appropriately benchmark performance of a given method. Our Transformer, which performs worst of the three models in secondary structure prediction, performs best on the fluorescence and stability tasks. The reverse is true of our ResNet, which ties the LSTM in secondary structure prediction but performs far worse for the fluorescence task, with a Spearman's ρ of 0.21 compared to the LSTM's 0.67. This shows that performance on a single task does not capture the full extent of a trained model's knowledge and biases, creating the need for multi-task benchmarks such as TAPE.

Chapter 3

Transformer Protein Language Models are Unsupervised Structure Learners

3.1 Introduction

Unsupervised modeling of protein contacts has an important role in computational protein design [64–66] and is a central element of all current state-of-the-art structure prediction methods [67–69]. The standard bioinformatics pipeline for unsupervised contact prediction includes multiple components with specialized tools and databases that have been developed and optimized over decades. In this Chapter we propose replacing the current multi-stage pipeline with a single forward pass of a pre-trained end-to-end protein language model.

Protein language modeling with an unsupervised training objective has been investigated by multiple groups [25, 42, 43, 70]. The longstanding practice in bioinformatics has been to fit linear models on focused sets of evolutionarily related and aligned sequences; by contrast, protein language modeling trains nonlinear deep neural networks on large databases of evolutionarily diverse and unaligned sequences. High capacity protein language models have been shown to learn underlying intrinsic properties of proteins such as structure and function from sequence data [43].

Both Chapter 2 and concurrent work by Rives et al. [43] propose the Transformer for protein language modeling. Originally developed in the NLP community to represent long range context, the main innovation of the Transformer model is its use of self-attention [13]. Self-attention has particular relevance for the modeling of protein sequences. Unlike convolutional or recurrent models, the Transformer constructs a pairwise interaction map between all positions in the sequence. In principle this mechanism has an ideal form to model protein contacts.

In theory, end-to-end learning with a language model has advantages over the bioinformatics pipeline: (i) it replaces the expensive query, alignment, and training steps with a single forward pass, greatly accelerating feature extraction; and (ii) it shares parameters for all protein families, enabling generalization by capturing commonality across millions of

evolutionarily diverse and unrelated sequences.

We demonstrate that Transformer protein language models learn contacts in the self-attention maps with state-of-the-art performance. We compare ESM-1b [43], a large-scale (650M parameters) Transformer model trained on UniRef50 [71] to the Gremlin [72] pipeline which implements a log linear model trained with pseudolikelihood [73, 74]. Contacts can be extracted from the attention maps of the Transformer model by a sparse linear combination of attention heads identified by logistic regression. ESM-1b model contacts have higher precision than Gremlin contacts. When ESM and Gremlin are compared with access to the same set of sequences the precision gain from the protein language model is significant; the advantage holds on average even when Gremlin is given access to an optimized set of multiple sequence alignments incorporating metagenomics data.

We find a linear relationship between language modeling perplexity and contact precision. We also find evidence for the value of parameter sharing: the ESM-1b model significantly outperforms Gremlin on proteins with low-depth MSAs. Finally we explore the Transformer language model's ability to generate sequences and show that generated sequences preserve contact information.

3.2 Background

Multiple Sequence Alignments (MSAs) A multiple sequence alignment consists of a set of evolutionarily related protein sequences. Since real protein sequences are likely to have insertions, deletions, and substitutions, the sequences are *aligned* by minimizing a Levenshtein distance-like metric over all the sequences. In practice heuristic alignment schemes are used. Tools like Jackhmmer and HHblits can increase the number and diversity of sequences returned by iteratively performing the search and alignment steps [10, 27].

Metrics For a protein of length L, we evaluate the precision of the top L, $L/2$, and $L/5$ contacts for short range ($|i-j| \in [6, 12)$), medium range ($|i-j| \in [12, 24)$), and long range ($|i=j| \in [24, \infty)$) contacts. We also separately evaluate local contacts ($|i-j| \in [3, 6)$) for secondary structure prediction in Appendix B.9. In general, all contacts provide information about protein structure and important interactions, with shorter-range contacts being useful for secondary and local structure, while longer range contacts are useful for determining global structure [75].

3.3 Related Work

There is a long history of protein contact prediction [76] both from MSAs, and more recently, with protein language models.

Supervised contact prediction Recently, supervised methods using deep learning have resulted in breakthrough results in *supervised* contact prediction [67–69, 77, 78]. State-of-the art methods use deep residual networks trained with supervision from many protein structures. Inputs are typically covariance statistics [77, 78], or inferred coevolutionary parameters [67–69, 79]. Other recent work with deep learning uses sequences or evolutionary features as inputs [80, 81]. Xu et al. [82] demonstrates the incorporation of coevolutionary features is critical to performance of current state-of-the-art methods.

Unsupervised contact prediction In contrast to supervised methods, unsupervised contact prediction models are trained on sequences *without information from protein structures*. In principle this allows them to take advantage of large sequence databases that include information from many sequences where no structural knowledge is available. The main approach has been to learn evolutionary constraints among a set of similar sequences by fitting a Markov Random Field (Potts model) to the underlying MSA, a technique known as Direct Coupling Analysis (DCA). This was proposed by Lapedes et al. [83] and reintroduced by Thomas et al. [84] and Weigt et al. [85].

Various methods have been developed to fit the underlying Markov Random Field, including mean-field DCA (mfDCA) [86], sparse inverse covariance (PSICOV) [87] and pseudolikelihood maximization [73, 74, 88]. Pseudolikelihood maximization is generally considered state-of-the-art for unsupervised contact prediction and the Gremlin [73] implementation is used as the baseline throughout. We also provide mfDCA and PSICOV baselines. Recently deep learning methods have also been applied to fitting MSAs, and Riesselman et al. [2] found evidence that factors learned by a VAE model may correlate with protein structure.

Structure prediction from contacts While we do not perform structure prediction in this work, many methods have been proposed to extend contact prediction to structure prediction. For example, EVFold [89] and DCAFold [90] predict co-evolving couplings using a Potts Model and then generate 3D conformations by directly folding an initial conformation with simulated annealing, using the predicted residue-residue contacts as constraints. Similarly, FragFold [91] and Rosetta [92] incorporate constraints from a Potts Model into a fragment assembly based pipeline. Senior et al. [93], use features from a Potts model fit with pseudolikelihood maximization to predict pairwise distances with a deep residual network and optimize the final structure using Rosetta. All of these works build directly upon the unsupervised contact prediction pipeline.

Contact prediction from protein language models Since the introduction of large scale language models for natural language processing [12, 13], there has been considerable interest in developing similar models for proteins [25, 42, 43, 70, 94, 95]. Rives et al. [43] were the first to study protein Transformer language models, demonstrating that information about residue-residue contacts could be recovered from the learned representations by linear projections supervised with protein structures. Recently Vig et al. [7] performed an extensive

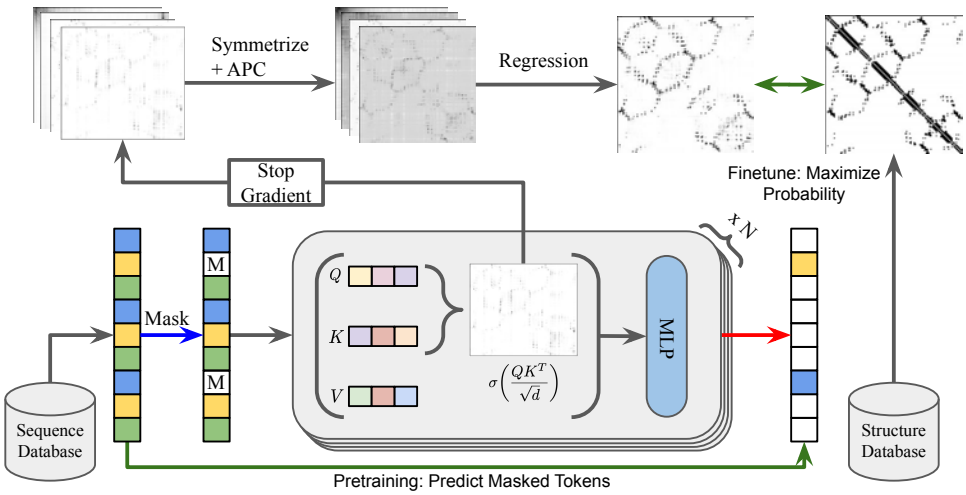


Figure 3.1: Contact prediction pipeline. The Transformer is first pretrained on sequences from a large database (Uniref50) via Masked Language Modeling. Once finished training, the attention maps are extracted, passed through symmetrization and average product correction, then into a regression. The regression is trained on a small number ($n \leq 20$) of proteins to determine which attention heads are informative. At test time, contact prediction from an input sequence can be done entirely on GPU in a single forward pass.

analysis of Transformer attention, identifying correspondences to biologically relevant features, and also found that different layers of the model are responsible for learning different features. In particular Vig et al. [7] discovered a correlation between self-attention maps and contact patterns, suggesting they could be used for contact prediction.

Prior work benchmarking contact prediction with protein language models has focused on the supervised problem. Bepler and Berger [24] were the first to fine-tune an LSTM pretrained on protein sequences to fit contacts. Chapter 2 as well as Rives et al. [43] perform benchmarking of multiple protein language models using a deep residual network fit with supervised learning on top of pretrained language modeling features.

In contrast to previous work on protein language models, we find that a state-of-the-art *unsupervised* contact predictor can be directly extracted from the Transformer self-attention maps. We perform a thorough analysis of the contact predictor, showing relationships between performance and MSA depth as well as language modeling perplexity. We also provide methods for improving performance using sequences from an MSA and for sampling sequences in a manner that preserves contacts.

3.4 Models

We compare Transformer models trained on large sequence databases to Potts Models trained on individual MSAs. While Transformers and Potts Models emerged in separate research communities, the two models share core similarities [96] which we exploit here.

Our main result is that just as Gremlin directly represents contacts via its pairwise component (the weights), the Transformer also directly represents contacts via its pairwise component (the self-attention).

Objectives

For a set of training sequences, X, Gremlin optimizes the following pseudolikelihood loss, where a single position is masked and predicted from its context. Inputs are aligned, so all have length L:

$$\mathcal{L}_{\text{PLL}}(X;\theta) = \mathop{\mathbb{E}}_{x\sim X} \sum_{i=1}^{L} \log p(x_i|x_{j\neq i};\theta) \tag{3.1}$$

The masked language modeling (MLM) loss used by the Transformer models can be seen as a generalization of the Potts Model objective when written as follows:

$$\mathcal{L}_{\text{MLM}}(X;\theta) = \mathop{\mathbb{E}}_{x\sim X} \mathop{\mathbb{E}}_{\text{mask}} \sum_{i\in\text{mask}} \log p(x_i|x_{j\notin\text{mask}};\theta) \tag{3.2}$$

In contrast to Gremlin, the MLM objective applied by protein language modeling is trained on unaligned sequences. The key distinction of MLM is to mask and predict multiple positions concurrently, instead of masking and predicting one at a time. This enables the model to scale beyond individual MSAs to massive sequence datasets. In practice, the expectation under the masking pattern is computed stochastically using a single sample at each epoch.

Gremlin

The log probability optimized by Gremlin is described in Appendix B.3.

Contacts are extracted from the pairwise Gremlin parameters by taking the Frobenius norm along the amino acid dimensions, resulting in an $L \times L$ coupling matrix. Average product correction (APC) is applied to this coupling matrix to determine the final predictions (Appendix B.2).

Gremlin takes an MSA as input. The quality of the output predictions are highly dependent on the construction of the MSA. We compare to Gremlin under two conditions. In the first condition, we present Gremlin with all MSAs from the trRosetta training set [69]. These MSAs were generated from all of Uniref100 and are also supplemented with metagenomic sequences when the depth from Uniref100 is too low. The trRosetta MSAs are a key ingredient in the state-of-the-art protein folding pipeline. See Yang et al. [69] for a discussion on the

significant impact of metagenomic sequences on the final result. In the second setting, we allow Gremlin access only to the same information as the ESM Transformers by generating MSAs via Jackhmmer on the ESM training set (a subset of Uniref50). See Appendix B.5 for Jackhmmer parameters.

Transformers

We evaluate several pre-trained Transformer models, including ESM-1 [43], ProtBert-BFD [94] and the TAPE Transformer from Chapter 2. The key differences between these models are the datasets, model sizes, and hyperparameters (major architecture differences described in Table B.1). Liu et al. [97] previously showed that these changes can have a significant impact on final model performance. In addition to ESM-1, we also evaluate an updated version, ESM-1b, which is the result of a hyperparameter sweep. The differences are described in Appendix B.4. The Transformer processes inputs through a series of blocks alternating multi-head self-attention and feed-forward layers. In each head of a self-attention layer, the Transformer views the encoded representation as a set of query-key-value triples. The output of the head is the result of scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{n}) \cdot V$$

Rather than only computing the attention once, the multi-head approach runs scaled dot-product attention multiple times in parallel and concatenates the output. Since self-attention explicitly constructs pairwise interactions (QK^T) between all positions in the sequence, the model can directly represent residue-residue interactions. In this work, we demonstrate that the QK^T pairwise "self attention maps" indeed capture accurate contacts.

Logistic Regression

To extract contacts from a Transformer, we first pass the input sequence through the model to obtain the attention maps (one map for each head in each layer). We then symmetrize and apply APC to each attention map independently. The resulting maps are passed through an L_1-regularized logistic regression, which is applied independently at each amino acid pair (i, j). At training time, we only train the weights of the logistic regression; we do not backpropagate through the entire model. At test time, the entire prediction pipeline can be run in a single forward pass, providing a single end-to-end pipeline for protein contact prediction that does not require any retrieval steps from a sequence database. See Appendix B.7 for a full description of the logistic regression setup.

3.5 Results

We evaluate models with the 15051 proteins in the trRosetta training dataset [69], removing 43 proteins with sequence length greater than 1024, since ESM-1b was trained with a context

Table 3.1: Average precision on 14842 test structures for Transformer models trained on 20 structures.

	$6 \leq \text{sep} < 12$			$12 \leq \text{sep} < 24$			$24 \leq \text{sep}$		
Model	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Gremlin (ESM Data)	15.2	23.0	37.8	18.1	27.9	44.3	31.3	43.1	55.5
mfDCA (Uniref100)	16.3	23.7	35.8	19.7	29.8	45.5	33.0	43.5	54.2
PSICOV[1](Uniref100)	15.4	23.6	39.2	18.3	28.4	45.7	32.6	45.2	58.1
Gremlin (Uniref100)	17.2	26.7	44.4	21.1	33.3	52.3	39.3	52.2	62.8
TAPE	9.9	12.3	16.4	10.0	12.6	16.6	11.2	14.0	17.9
ProtBERT-BFD	20.4	30.7	48.4	24.3	35.5	52.0	34.1	45.0	57.4
ESM-1 (6 layer)	11.0	13.2	15.9	11.5	14.6	19.0	13.2	16.7	21.5
ESM-1 (12 layer)	15.2	21.1	30.5	18.1	24.7	34.0	23.7	30.5	39.3
ESM-1 (34 layer)	20.3	30.2	46.0	23.8	34.3	49.2	34.7	44.6	56.0
ESM-1b	**21.6**	**33.2**	**52.7**	**26.2**	**38.6**	**56.4**	**41.1**	**53.3**	**66.1**

size of 1024. Of these sequences, Jackhmmer fails on 126 when we attempt to construct MSAs using the ESM training set (see Appendix B.5). This leaves us with 14882 total sequences. We reserve 20 sequences for training, 20 sequences for validation, and 14842 sequences for testing.

Table 3.1 shows evaluations of Gremlin, ESM-1, ESM-1b as well as the TAPE and ProtBERT-BFD models. Confidence intervals are within 0.5 percentage points for all statistics in Tables 3.1 and 3.2. In Table 3.1, all Transformer model contact predictors are trained with logistic regression on 20 proteins. We find that with only 20 training proteins ESM-1b has higher precision than Gremlin for short, medium, and long range contacts.

In addition to this set, we also evaluate performance on 15 CASP13 FM Domains in Appendix B.6. On average ESM-1b has higher short, medium, and long range precision than Gremlin on all metrics, and in particular can significantly outperform on MSAs with low effective number of sequences. We also provide a comparison to the bilinear model proposed by Rives et al. [43]. The logistic regression model achieves a long-range contact precision at L of 18.6, while the fully supervised bilinear model achieves a long range precision at L of 20.1, an increase of only 1.5 points despite being trained on 700x more structures.

Table 3.2: ESM-1b Ablations with limited supervision and with MSA information. n is the number of logistic regression training proteins. s is the number of sequences ensembled over.

		$6 \leq$ sep < 12			$12 \leq$ sep < 24			$24 \leq$ sep		
Model	Variant	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Gremlin	ESM Data	15.2	23.0	37.8	18.1	27.9	44.3	31.3	43.1	55.5
	Uniref100	17.2	26.7	44.4	21.1	33.3	52.3	39.3	52.2	62.8
ESM-1b (Ablations)	1 head	16.8	23.4	34.8	19.8	27.6	40.2	29.3	38.1	50.0
	5 heads	19.2	28.5	44.5	23.3	33.8	49.0	35.0	45.2	57.3
	10 heads	20.0	30.1	47.4	24.7	36.0	52.2	38.5	49.4	61.1
	n=1, s=1	19.4	29.7	47.1	25.1	37.1	54.0	39.2	50.6	63.0
	n=10, s=1	21.4	32.9	52.3	26.1	38.5	56.4	40.8	52.9	65.7
	n=20, s=1	21.6	33.2	52.7	26.2	38.6	56.4	41.1	53.3	66.1
	MSA, s=1	18.4	28.1	45.5	23.9	36.1	53.7	39.9	51.3	63.0
ESM-1b (s seqs)	n=20, s=16	21.9	33.8	53.6	26.7	39.4	57.5	41.9	54.3	67.3
	n=20, s=32	22.0	34.1	54.0	26.9	39.8	58.1	42.3	54.8	67.8
	n=20, s=64	**22.1**	**34.3**	**54.3**	**27.1**	**40.1**	**58.5**	**42.6**	**55.1**	**68.2**

Ablations: Limiting supervision

While the language modeling objective is fully unsupervised, the logistic regression is trained with a small number of supervised examples. In this section, we study the dependence of the results on this supervision, providing evidence that the contacts are indeed learned in the unsupervised phase, and the logistic regression is only necessary to extract the contacts.

Top Heads Here we use the logistic regression only to determine the most important heads. Once they are selected, we discard the weights from the logistic regression and simply average the attention heads corresponding to the top-k weight values. By taking the single best head from ESM-1b, we come close to Gremlin performance given the same data, and averaging the top-5 heads allows us to outperform Gremlin. Averaging the top-10 heads outperforms a full logistic regression on all other Transformer models and comes close to Gremlin given optimized MSAs.

Low-N The second variation we consider is to limit the number of supervised examples provided to the logistic regression. We find that with **only a single training example**, the

[1]PSICOV fails to converge on 24 sequences using default parameters. Following the suggestion in `github.com/psipred/psicov`, we increase ρ to 0.005, 0.01, and thereafter by increments of 0.01, to a maximum of 0.1. PSICOV fails to converge altogether on 6 / 14842 sequences. We assign a score of 0 for these sequences.

model achieves a long range top-L precision of 39.2, which is statistically indistinguishable from Gremlin ($p > 0.05$). Using only 10 training examples, the model outperforms Gremlin on all the metrics. Since these results depend on the sampled training proteins, we also show a bootstrapped performance distribution using 100 different logistic regression models in Appendix B.10. We find that with 1 protein, performance can vary significantly, with long range top-L precision mean of 35.6, a median of 38.4, and standard deviation 8.9. This variation greatly decreases when training on 20 proteins, with a long range top-L precision mean of 40.1, median of 41.1, and standard deviation of 0.3. See Fig. B.8 for the full distribution on all statistics.

MSA Only Finally, we consider supervising the logistic regression only with MSAs instead of real structures. This is the same training data used by the Gremlin baseline. To do this, we first train Gremlin on each MSA. We take the output couplings from Gremlin and mark the top L couplings with sequence separation ≥ 6 in each protein as true contacts, and everything else as false contacts, creating a binary decision problem. When trained on 20 MSAs, we find that this model achieves a long range P@L of 39.9, and generally achieves similar long range performance to Gremlin, while still having superior short and medium range contact precision.

Ensembling over MSA

Transformer models are fundamentally single-sequence models, but we can further boost performance by ensembling predictions from multiple sequences in the alignment. To do so, we unalign each sequence in the alignment (removing any gaps), pass the resulting sequence through the Transformer and regression, and realign the resulting contact maps to the original aligned indices. For these experiments, we use the logistic regression weights trained on single-sequence inputs, rather than re-training the logistic regression on multi-sequence inputs. We also simply take the first s sequences in the MSA. Table 3.2 shows performance improvements from averaging over 16, 32, and 64 sequences.

To better understand this result, we return to the single-sequence setting and study the change in prediction when switching between sequences in the alignment. We find that contact precision can vary significantly depending on the exact sequence input to the model, and that the initial query sequence of the MSA does not necessarily generate the highest contact precision (Fig. B.5).

Lastly, Alley et al. [25] presented a method of fine-tuning where a pretrained language model is further trained on the MSA of the sequence of interest ('evotuning'). Previously this has only been investigated for function prediction and for relatively low-capacity models. We fine-tune the full ESM-1b model (which has 50x more parameters than UniRep) on 380 protein sequence families. We find that after 30 epochs of fine-tuning, long range P@L increases only slightly, with an average of 1.6 percentage points (Fig. B.12).

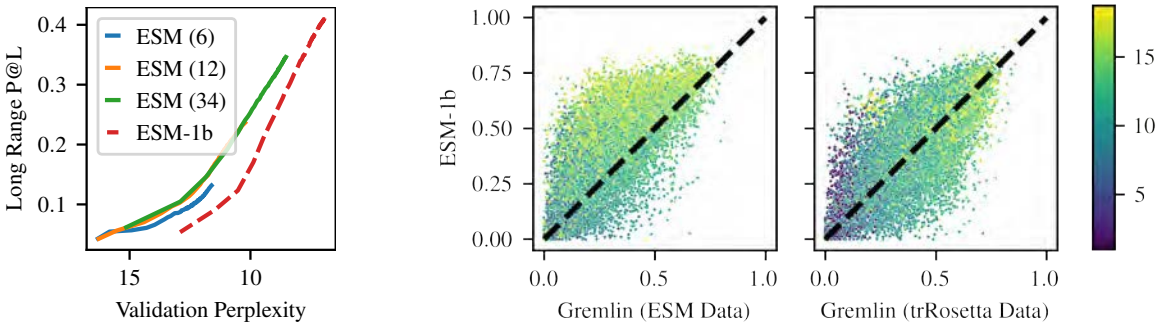


Figure 3.2: Left: Language modeling validation perplexity on holdout of Uniref50 vs. contact precision over the course of pre-training. ESM-1b was trained with different masking so perplexities between the versions are not comparable. Right: Long range P@L performance distribution of ESM-1b vs. Gremlin. Each point is colored by the log of the number of sequences in the MSA used to train Gremlin.

Performance Distribution

Although our model is, on average, better than Gremlin at detecting contacts, the performance distribution over all sequences in the dataset is still mixed. ESM-1b is consistently better at extracting short and medium range contacts (Fig. B.3), but only slightly outperforms Gremlin on long range contacts when Gremlin has access to Uniref100 and metagenomic sequences. Fig. 3.2 shows the distribution of long range P@L for ESM-1b vs. Gremlin. Overall, ESM-1b has higher long range P@L on 55% of sequences in the test set.

In addition, we examine the relationship between MSA depth and precision for short, medium, and long range contacts (Fig. 3.3). Although our contact prediction pipeline does not make explicit use of MSAs, there is still some correlation between MSA depth and performance, since MSA depth is a measure of how many related sequences are present in the ESM-1b training set. We again see that ESM-1b consistently outperforms Gremlin at all MSA depths for short and medium range sequences. We also confirm that ESM-1b outperforms Gremlin for long range contact extraction for sequences with small MSAs (depth < 1000). ESM-1b also outperforms Gremlin on sequences with the very largest MSAs (depth > 16000), which is consistent with prior work showing that Gremlin performance plateaus for very large MSAs and suggests that ESM-1b does not suffer from the same issues [98].

Logistic Regression Weights

In Section 3.5 we show that selecting only a sparse subset of the attention heads can yield good results for contact prediction. Overall, the L_1-regularized logistic regression identifies 102 / 660 heads as being predictive of contacts (Fig. B.2b). Additionally, we train separate logistic regressions to identify contacts at different ranges . These regressions identify an overlapping, but non-identical set of useful attention heads. Two attention heads have the

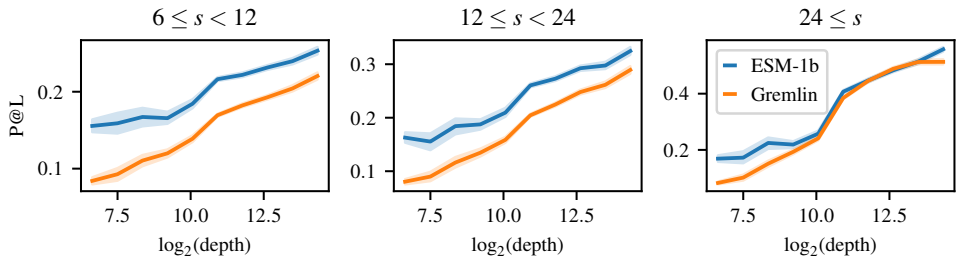


Figure 3.3: Gremlin (trRosetta) performance binned by MSA depth. For comparison, ESM-1b performance is also shown for the sequences in each bin.

top-10 highest weights for detecting contacts at all ranges. One attention head is highly positively correlated with local contacts, but highly negatively correlated with long range contacts. Lastly, we identify a total of 104 attention heads that are correlated (positively or negatively) with contacts at only one of the four ranges, suggesting that particular attention heads specialize in detecting certain types of contacts.

Perplexity vs. Contact Precision

Fig. 3.2 explores the relationship between performance on the masked language modeling task (validataion perplexity) and contact prediction (Long Range P@L). A linear relationship exists between validation perplexity and contact precision for each model. Furthermore, for the same perplexity, the 12-layer ESM-1 model achieves the same long range P@L as the 34 layer ESM-1 model, suggesting that perplexity is a good proxy task for contact prediction. ESM-1 and ESM-1b models are trained with different masking patterns, so their perplexities cannot be directly compared, although a linear relationship is clearly visible in both. ESM-1 and ESM-1b have a similar number of parameters; the key difference is in their hyperparameters and architecture. The models shown have converged in pre-training, with minimal decrease in perplexity (or increase in contact precision) in the later epochs. This provides clear evidence that both model scale and hyperparameters play a significant role in a model's ability to learn contacts.

Calibration, False Positives, and Robustness

One concern with neural networks is that, while they may be accurate on average, they can also produce spurious results with high confidence. We investigate this possibility from several perspectives. First, we find that logistic regression probabilities are close to true contact probability (mean-squared error = 0.014) and can be used directly as a measure of the model's confidence (Fig. B.7).

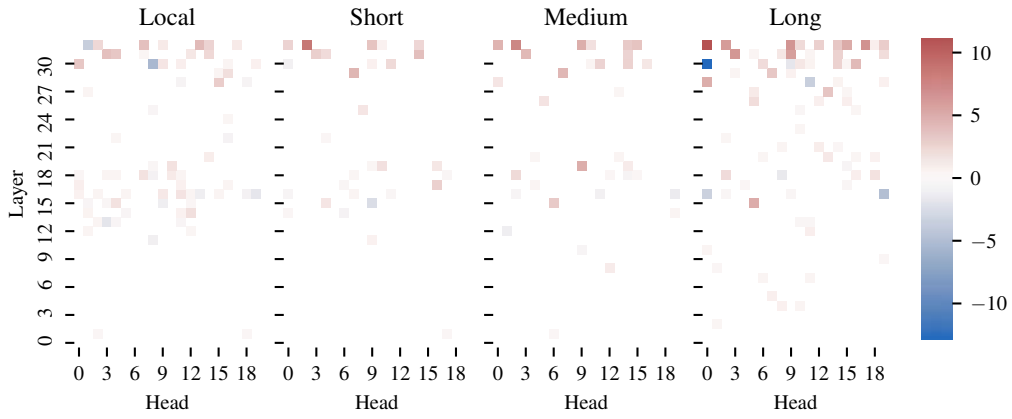


Figure 3.4: Logistic regression weights trained only on contacts in specific ranges: local [3, 6), short range [6, 12), medium range [12, 24), long range [24, ∞).

Second, we analyze the false positives that the model does predict. We find that these are very likely to be within a Manhattan distance of 1-4 of a true contact (Fig. B.9a). This suggests that false positives may arise due to the way a contact is defined (Cb-Cb distance within 8 angstroms), and could be marked as true contacts under a different definition [99]. Further, when we explore an example where the model's predictions are not near a true contact, we see that the example in question is a homodimer, and that the model is picking up on inter-chain interactions (Fig. B.10a). While these do not determine the structure of the monomer, they are important for its function [98].

Third, we test the robustness of the model to insertions by inserting consecutive alanines at the beginning, middle, or end of 1000 randomly chosen sequences. We find that ESM-1b can tolerate up to 256 insertions at the beginning or end of the sequence and up to 64 insertions in the middle of the sequence before performance starts to significantly degrade. This suggests that ESM-1b learns a robust implicit alignment of the protein sequence. See Appendix B.12 for more details.

MSA Generation

Wang and Cho [96] note that Transformers trained with the MLM objective can be used generatively. Here, we consider whether generations from ESM-1b preserve contact information. The ability to generate sequences that preserve this information is a necessary condition for generation of biologically active proteins [100]. We perform this evaluation by taking an input protein, masking out several positions, and re-predicting them. This process is repeated 10000 times to generate a pseudo-MSA for the input sequence (Algorithm 1). We feed the resulting MSA into Gremlin to predict contacts. Over all sequences from our test set, this procedure results in a long range contact P@L of 14.5. Fig. B.13 shows one example where the procedure works well, with Gremlin on the pseudo-MSA having long range P@L of 52.2.

3.6 Discussion

Transformer protein language models trained with an unsupervised objective learn the tertiary structure of a protein sequence in their attention maps. Residue-residue contacts can be extracted from the attention by sparse logistic regression. Attention heads are found that specialize in different types of contacts. An ablation analysis confirms that the contacts are learned without supervision, and that the logistic regression is only necessary to extract the part of the model that represents contacts.

These results have implications for protein structure determination and design. The initial studies proposing Transformers for protein language modeling in Chapter 2 and in Rives et al. [43], Elnaggar et al. [94] showed that representation learning could be used to derive state-of-the-art features across a variety of tasks, but were not able to show a benefit in the fully end-to-end setting. In this Chapter, we show that protein language models can outperform state-of-the-art unsupervised structure learning methods that have been intensively researched and optimized over decades.

Finally, we establish a link between language modeling perplexity and unsupervised structure learning. A similar scaling law has been observed previously for supervised secondary structure prediction [43], and parallels observations in the NLP community [101, 102]. Evidence of scaling laws for protein language modeling support future promise as models and data continue to grow.

Chapter 4

MSA Transformer

4.1 Introduction

Unsupervised models learn protein structure from patterns in sequences. Sequence variation within a protein family conveys information about the structure of the protein [18, 19, 103]. Since evolution is not free to choose the identity of amino acids independently at sites that are in contact in the folded three-dimensional structure, patterns are imprinted onto the sequences selected by evolution. Constraints on the structure of a protein can be inferred from patterns in related sequences. The predominant unsupervised approach is to fit a Markov Random Field in the form of a Potts Model to a family of aligned sequences to extract a coevolutionary signal [83–85].

This thesis explores unsupervised protein language models, an approach which fits large neural networks with shared parameters across millions of diverse sequences, rather than fitting a model separately to each family of sequences. At inference time, a single forward pass of an end-to-end model replaces the multi-stage pipeline, involving sequence search, alignment, and model fitting steps, standard in bioinformatics. Recently, promising results have shown that protein language models learn secondary structure, long-range contacts, and function via the unsupervised objective [43], making them an alternative to the classical pipeline. While Chapter 2 showed that small and recurrent models fall well short of state-of-the-art, Chapter 3 showed that the internal representations of very large transformer models are competitive with Potts models for unsupervised structure learning.

Potts models have an important advantage over protein language models during inference. The input to the Potts model is a set of sequences. Inference is performed by fitting a model that directly extracts the covariation signal from the input. Current protein language models take a single sequence as input for inference. Information about evolutionary variation must be stored in the parameters of the model during training. As a result, protein language models require many parameters to represent the data distribution well.

In this Chapter, we unify the two paradigms within a protein language model that takes sets of aligned sequences as input, but shares parameters across many diverse sequence

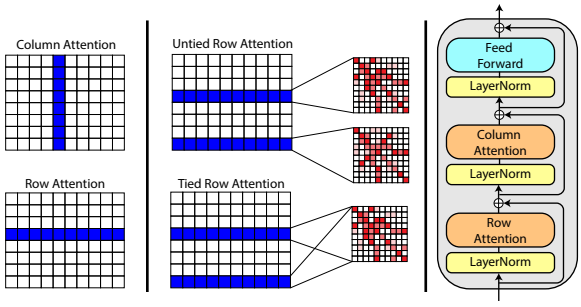


Figure 4.1: **Left:** Sparsity structure of the attention. By constraining attention to operate over rows and columns, computational cost is reduced from $O(M^2L^2)$ to $O(LM^2) + O(ML^2)$ where M is the number of rows and L the number of columns in the MSA. **Middle:** Untied row attention uses different attention maps for each sequence in the MSA. Tied row attention uses a single attention map for all sequences in the MSA, thereby constraining the contact structure. Ablation studies consider the use of both tied and untied attention. The final model uses tied attention. **Right:** A single MSA Transformer block. The depicted architecture is from the final model, some ablations alter the ordering of row and column attention.

families. Like prior protein language models operating on individual sequences, the approach benefits from learning from common patterns across protein families, allowing information to be generalized and transferred between them. By taking sets of sequences as input, the model gains the ability to extract information during inference, which improves the parameter efficiency.

We introduce the MSA Transformer, a model operating on sets of aligned sequences. The input to the model is a multiple sequence alignment. The architecture interleaves attention across the rows and columns of the alignment as in axial attention [104, 105]. We propose a variant of axial attention which shares a single attention map across the rows. The model is trained using the masked language modeling objective. Self supervision is performed by training the model to reconstruct a corrupted MSA.

We train an MSA Transformer model with 100M parameters on a large dataset (4.3 TB) of 26 million MSAs, with an average of 1192 sequences per MSA. The resulting model surpasses current state-of-the-art unsupervised structure learning methods by a wide margin, outperforming Potts models and protein language models with 650M parameters. The model improves over state-of-the-art unsupervised contact prediction methods across all multiple sequence alignment depths, with an especially significant advantage for MSAs with lower depth. Information about the contact pattern emerges directly in the tied row attention maps. Evaluated in a supervised contact prediction pipeline, features captured by the MSA Transformer outperform trRosetta [69] on the CASP13 and CAMEO test sets. We find that high precision contact predictions can be extracted from small sets of diverse sequences, with good results from as few as 8-16 sequences. We investigate how the model performs inference by independently destroying the covariation or sequence patterns in the input, finding that

the model uses both signals to make predictions.

4.2 Related Work

Unsupervised Contact Prediction The standard approach to unsupervised protein structure prediction is to identify pairwise statistical dependencies between the columns of an MSA, which are modeled as a Potts model Markov Random Field (MRF). Since exact inference is computationally intractable, a variety of methods have been proposed to efficiently fit the MRF, including mean-field inference [86], sparse-inverse covariance estimation [87], and the current state-of-the-art, pseudolikelihood maximization [73, 74, 88]. In this work we use Potts models fit with psuedolikelihood maximization as a baseline, and refer to features generated from Potts models as "co-evolutionary features." Making a connection with the attention mechanism we study here, Bhattacharya et al. [106] show that a single layer of self-attention can perform essentially the same computation as a Potts model.

Deep Models of MSAs Several groups have proposed to replace the shallow MRF with a deep neural network. Riesselman et al. [2] train deep variational autoencoders on MSAs to predict function. Riesselman et al. [107] train autoregressive models on MSAs, but discard the alignment, showing that function can be learned from unaligned sequences. In contrast to our approach which is trained on many MSAs, these existing models are trained on a single set of related sequences and do not provide a direct method of extracting protein contacts.

Supervised Structure Prediction Supervised structure prediction using deep neural networks has driven groundbreaking progress on the protein structure prediction problem [93, 108]. Initial models used coevolutionary features [67, 69, 78, 79, 93]. Recently MSAs have been proposed as input to supervised structure prediction methods. Mirabello and Wallner [109] and Kandathil et al. [110] study models that take MSAs as input directly, respectively using 2D convolutions or GRUs to process the input. More recently, AlphaFold2 [108] uses attention to process MSAs in an end-to-end model supervised with structures.

The central difference in our work is to model a collection of MSAs using *unsupervised learning*. This results in a model that contains features potentially useful for a range of downstream tasks. We use the emergence of structure in the internal representations of the model to measure the ability of the model to capture biology from sequences. This is a fundamentally distinct problem setting from supervised structure prediction. The MSA Transformer is trained in a purely unsupervised manner and learns contacts without being trained on protein structures.

Large protein sequence databases contain billions of sequences and are undergoing exponential growth. Unsupervised methods can directly use these datasets for learning, while supervised methods are limited to supervision from the hundreds of thousands of crystallized structures. Unsupervised methods can learn from regions of sequence space not covered by structural knowledge.

Protein Language Models Protein language modeling has emerged as a promising approach for unsupervised learning of protein sequences. Bepler and Berger [24] combined unsupervised sequence pre-training with structural supervision to produce sequence embeddings. Alley et al. [25] and Heinzinger et al. [42] showed that LSTM language models capture some biological properties. Simultaneously, Rives et al. [43] proposed to model protein sequences with self-attention, showing that transformer protein language models capture accurate information of structure and function in their representations. Chapter 2 evaluated a variety of protein language models across a panel of benchmarks concluding that small LSTMs and transformers fall well short of features from the bioinformatics pipeline.

A combination of model scale and architecture improvements has been instrumental to recent successes in protein language modeling. Elnaggar et al. [94] study a variety of transformer variants. Rives et al. [43] show that large transformer models produce state-of-the-art features across a variety of tasks. Notably, the internal representations of transformer protein language models are found to directly represent contacts. Vig et al. [7] find that specific attention heads of pre-trained transformers correlate directly with protein contacts. Chapter 3 uses these advances to combine multiple attention heads and predict contacts more accurately than Potts models, despite using just a single sequence for inference.

Alternatives to the masked language modeling objective have also been explored, such as conditional generation [70] and contrastive loss functions [95]. Most relevant to our work, Sturmfels et al. [111] and Sercu et al. [112] study alternative learning objectives using sets of sequences for supervision. Sturmfels et al. [111] extended the unsupervised language modeling to predict the position specific scoring matrix (PSSM) profile. Sercu et al. [112] used amortized optimization to simultaneously predict profiles and pairwise couplings. In natural language processing, recent work [113, 114] has explored models using multiple sequences. However, previous work on protein language models has not considered inference *directly* from sets of sequences.

4.3 Methods

Transformers are powerful sequence models capable of passing information from any position to any other position [13]. However, they are not trivially applied to a set of aligned sequences. Naively concatenating M sequences of length L in an MSA would allow attention across all sequences, but the $(ML)^2$ self-attention maps would be prohibitively memory-intensive. The main contribution of this paper is to extend transformer pre-training to operate on an MSA, while respecting its structure as an $M \times L$ character matrix.

We describe the input MSA as a matrix $\boldsymbol{x} \in \mathbb{R}^{M \times L}$, where rows correspond to sequences in the MSA, columns are positions in the aligned sequence, and entries $\boldsymbol{x}_{mi}$ take integer values[1] encoding the amino acid identity of sequence m at position i. After embedding the input, each layer has a $\mathbb{R}^{M \times L \times d}$ state as input and output. For the core of the transformer,

[1]The final vocab size is 29, consisting of 20 standard amino acids, 5 non-standard amino acids, the alignment character '.', gap character '-', the start token, and the mask token

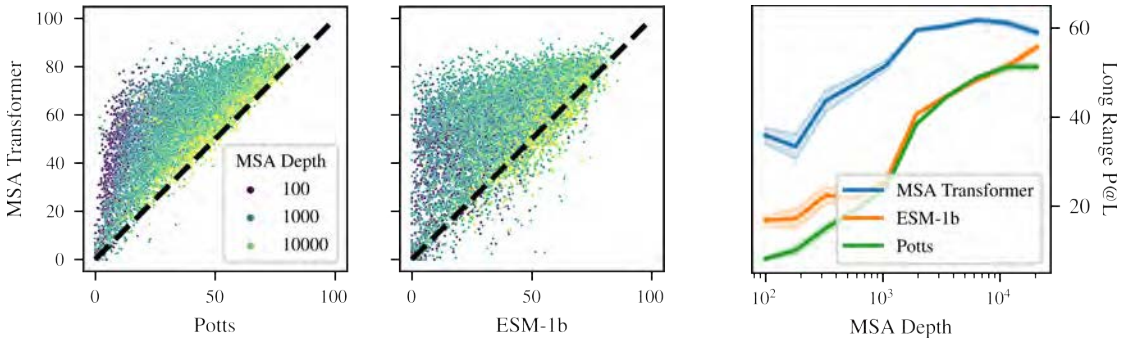


Figure 4.2: **Left:** Top-L long-range contact precision (higher is better). Comparison of MSA Transformer with Potts model (left scatter plot), and ESM-1b (right scatter plot). Each point represents a single protein (14,842 total) and is colored by the depth of the full MSA for the sequence. The Potts model is given the full MSA as input; ESM-1b is given only the reference sequence; and the MSA Transformer is given an MSA subsampled with hhfilter to a maximum of 256 sequences. The MSA Transformer outperforms both models for the vast majority of sequences. **Right:** Long-range contact precision performance as a function of MSA depth. Sequences are binned by MSA depth into 10 bins; average performance in each bin along with 95% confidence interval is shown. The minimum MSA depth in the trRosetta dataset is 100 sequences. While model performance generally increases with MSA depth, the MSA Transformer performs very well on sequences with low-depth MSAs, rivaling Potts model performance on MSAs 10x larger.

we adapt the axial attention approach from Ho et al. [104], Huang et al. [105], and Child et al. [115]. This approach alternates attention over rows and columns of the 2D state (see Fig. 4.1). This sparsity pattern in the attention over the MSA brings the attention cost to $O(LM^2)$ for the column attention, and $O(ML^2)$ for the row attention.

Feedforward Layers We deviate from Ho et al. [104] in the interleaving of the feedforward layers. Rather than applying a feedforward layer after each row or column attention, we apply row and column attention followed by a single feedforward layer (see Fig. 4.1). This choice follows more closely the transformer decoder architecture [13].

Position Embedding The standard transformer position embedding is a 1D signal added to each position in the sequence. Either fixed sinusoidal [13] or learned [12] position embeddings are most commonly used. Rives et al. [43] found that learned position embeddings generally resulted in better downstream performance for protein language models.

An MSA is a 2D input so we must consider two types of position embeddings. For all models trained, we provide a 1D *sequence* position embedding, which is added independently to each row of the MSA. This allows the model to distinguish different aligned positions. For one model, we also add a position embedding independently to each column of the MSA,

which allows the model to distinguish different sequences (without this, the model treats the input sequences as an unordered set). We also ensure that the first position in the sequence is always the reference so that it can always be uniquely identified through the position embedding. We find that incorporating the column position embedding increases performance slightly and so choose to use it in the final model (see Appendix C.3 for further discussion).

Tied Row Attention The standard implementation of axial attention allows for independent attention maps for each row and column of the input. However, in an MSA each sequence should have a similar structure; indeed, direct-coupling analysis exploits this fact to learn contact information. To leverage this shared structure we hypothesize it would be beneficial to tie the row attention maps between the sequences in the MSA. As an additional benefit, tied attention reduces the memory footprint of the row attentions from $O(ML^2)$ to $O(L^2)$.

Let M be the number of rows, d be the hidden dimension and Q_m, K_m be the matrix of queries and keys for the m-th row of input. We define tied row attention (before softmax is applied) to be:

$$\sum_{m=1}^{M} \frac{Q_m K_m^T}{\lambda(M, d)} \tag{4.1}$$

The denominator $\lambda(M, d)$ would be the normalization constant $\sqrt{d}$ in standard scaled-dot product attention. In tied row attention, we explore two normalization functions to prevent attention weights linearly scaling with the number of input sequences: $\lambda(M, d) = M\sqrt{d}$ (mean normalization) and $\lambda(M, d) = \sqrt{Md}$ (square-root normalization). Our final model uses square-root normalization.

Pre-training Objective We adapt the masked language modeling objective [12] to the MSA setting. The loss for an MSA $\boldsymbol{x}$, and masked MSA $\tilde{\boldsymbol{x}}$ is as follows:

$$\mathcal{L}_{\text{MLM}}(\boldsymbol{x}; \theta) = \sum_{(m,i) \in \text{mask}} \log p(x_{mi} | \tilde{\boldsymbol{x}}; \theta) \tag{4.2}$$

The probabilities are the output of the MSA transformer, softmax normalized over the amino acid vocabulary independently per position i in each sequence m. We consider masking tokens uniformly at random over the MSA or masking entire columns of the MSA. Masking tokens uniformly at random results in best performance (Table C.2). Note that the masked token can be predicted not only from context amino acids at different positions but also from related sequences at the same position.

Pre-training Dataset Models are trained on a dataset of 26 million MSAs. An MSA is generated for each UniRef50 [71] sequence by searching UniClust30 [116] with HHblits [117]. The average depth of the MSAs is 1192. See Fig. C.2 for MSA depth distribution.

Models and Training We train 100M parameters model with 12 layers, 768 embedding size, and 12 attention heads, using a batch size of 512 MSAs, learning rate 10^{-4}, no weight decay, and an inverse square root learning rate schedule with 16000 warmup steps. All models are trained on 32 V100 GPUs for 100k updates. The four models with best contact precision are then further trained to 150k updates. Finally, the best model at 150k updates is trained to 450k updates. Unless otherwise specified, all downstream experiments use this model.

Despite the use of axial attention and tied attention to lower the memory requirements, large MSAs still do not easily fit in memory at training time. The baseline model fits a maximum of $N = 2^{14}$ tokens on a 32 GB V100 GPU at training time. To work around this limitation we subsample the input MSAs to reduce the number of sequences and limit the maximum sequence length to 1024.

MSA Subsampling During Inference At inference time, memory is a much smaller concern. Nevertheless we do not provide the full MSA to the model as it would be computationally expensive and the model's performance can decrease when the input is much larger than that used during training. Instead, we explore four strategies for subsampling the sequences provided to the model.

- **Random:** This strategy parallels the one used at training time, and selects random sequences from the MSA (ensuring that the reference sequence is always included).
- **Diversity Maximizing:** This is a greedy strategy which starts from the reference and adds the sequence with highest average hamming distance to current set of sequences.
- **Diversity Minimizing:** This strategy is equivalent to the Diversity Maximizing strategy, but adds the sequence with lowest average hamming distance. It is used to explore the effects of diversity on model performance.
- **HHFilter:** This strategy applies hhfilter [117] with the `-diff M` parameter, which returns M or more sequences that maximize diversity (the result is usually close to M). If more than M sequences are returned we apply the Diversity Maximizing strategy on top of the output.

4.4 Results

We study the MSA Transformer in a panel of structure prediction tasks, evaluating unsupervised contact prediction from the attentions of the model, and performance of features in supervised contact and secondary structure prediction pipelines.

To calibrate the difficulty of the masked language modeling task for MSAs, we compare against two simple prediction strategies using the information in the MSA: (i) column frequency baseline, and (ii) nearest sequence baseline. These baselines implement the intuition that a simple model could use the column frequencies to make a prediction at

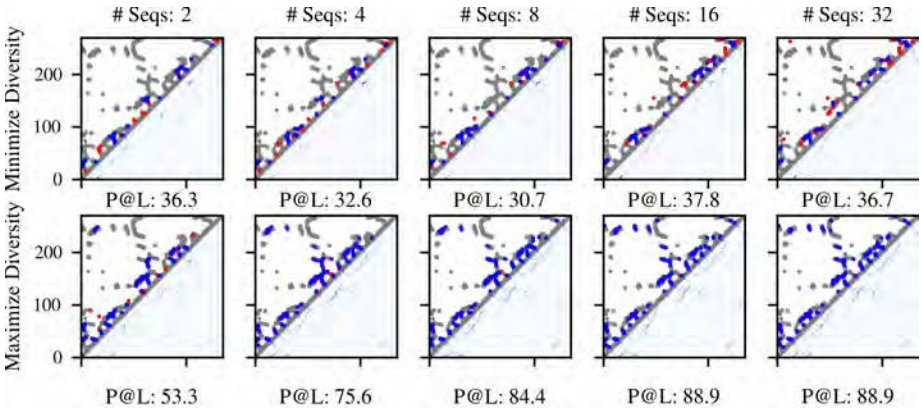


Figure 4.3: Contact prediction from a small set of input sequences. Predictions are compared under diversity minimizing and diversity maximizing sequence selection strategies. Visualized for 4zjp chain A. Raw contact probabilities are shown below the diagonal, top L contacts are shown above the diagonal. (blue: true positive, red: false positive, grey: ground-truth contacts). Top-L long-range contact precision below each plot. Contact precision improves with more sequences under both selection strategies. Maximizing the diversity enables identification of long-range contacts from a small set of sequences.

the masked positions, or copy the identity of the missing character from the most similar sequence in the input. Table C.1 reports masked language modeling performance. The MSA Transformer model (denoising accuracy of 64.0) significantly outperforms the PSSM (accuracy 41.4) and nearest-neighbor (accuracy 46.7) baselines.

Unsupervised Contact Prediction

Chapter 3 showed that transformer protein language models learned to capture information about protein structure in their attention maps using little to no supervision. This is done by training a small logistic regression (one parameter per attention head) on a limited number of protein structures to predict the probability of a contact between residues i and j based on the attentions between the residues for all attention heads. The logistic regression weights are shared for all pairs of positions (see Appendix C.1 for more details).

We use the same validation methodology. A logistic regression with 144 parameters is fit on 20 training structures from the trRosetta dataset [69]. This is then used to predict the probability of protein contacts on another 14842 structures from the trRosetta dataset (training structures are excluded). At inference time, we use hhfilter to subsample 256 sequences.

We compare to two state-of-the-art transformer protein language models: ESM-1b [43] with 650M parameters and ProTrans-T5 [94] with 3B parameters. For the single-sequence protein language models we use the sequence belonging to the structure as input. We also

Table 4.1: Average long-range precision for MSA and single-sequence models on the unsupervised contact prediction task.

Model	L	L/2	L/5
Potts	39.3	52.2	62.8
TAPE	11.2	14.0	17.9
ProTrans-T5	35.6	46.1	57.8
ESM-1b	41.1	53.3	66.1
MSA Transformer	**57.4**	**71.7**	**82.1**

compare against Potts models using the APC-corrected [118] Frobenius norm of the coupling matrix computed on the MSA [72].

Table 4.1 compares unsupervised contact prediction performance of the models. The MSA Transformer significantly outperforms all baselines, increasing top-L long-range contact precision by a full 15 points over the previous state-of-the-art. Table 4.2 shows results on harder test sets CAMEO hard targets [119] and CASP13-FM [120]. The CASP13-FM test set consists of 31 free modeling domains (from 25 targets); the CAMEO hard targets are a set of 131 domains (out of which we evaluate on the 129 that fit within the 1024 character maximum context length of the model). On the CASP13-FM test set, *unsupevised* contact prediction with the MSA Transformer (43.4 top-L long-range precision) is competitive with the trRosetta base model (45.7 top-L long-range precision), a fully *supervised* structure prediction model.

Fig. 4.2 shows the top-L long-range precision distribution across all structures, comparing the MSA Transformer with Potts models and ESM-1b. The MSA Transformer matches or exceeds Potts models on 98.5% of inputs and matches or exceeds ESM-1b on 91.0% of inputs. Fig. 4.2 also shows unsupervised contact performance as a function of MSA depth. The model outperforms ESM-1b and Potts models across all MSA depths and has a significant advantage for lower depth MSAs. We find no statistically significant correlation between sequence length and contact precision.

These experiments also help address performance increases resulting from overlap between the training set and test sets. The Potts models, ESM-1b, and MSA Transformer are trained using the same Uniref database version. The third panel in Fig. 4.2 shows that there is significant correlation between the performance of all three models and the depth of the MSA provided to the Potts model, despite the fact that ESM-1b receives only a single sequence as input, and the MSA Transformer receives a subsampled MSA. The depth of the MSA provided to the Potts model can be used as a proxy for the *density* of similar sequences in the training sets of ESM-1b and MSA Transformer. ESM-1b and MSA Transformer show a clear increase in performance with increased homology overlap, but are on par with or outperform Potts models for any given degree of homology. In addition, because the Uniref

Table 4.2: Unsupervised contact prediction on CASP13 and CAMEO (long-range precision). Note the large improvement of MSA Transformer over classical Potts models and ESM-1b.

	CASP13-FM		CAMEO	
Model	L	L/5	L	L/5
Potts	16.9	31.5	24.0	42.8
ProTrans-T5	16.5	27.0	25.9	43.4
ESM-1b	17.0	30.4	30.9	52.7
MSA Transformer	**44.8**	**72.5**	**43.5**	**66.8**

Table 4.3: Supervised contact prediction on CASP13 and CAMEO (long-range precision). *Uses outer-concatenation of the query sequence representation as features. †Additionally uses the row attention maps as features.

	CASP13-FM		CAMEO	
Model	L	L/5	L	L/5
$\text{trRosetta}_{\text{base}}$	45.7	69.6	50.9	75.5
$\text{trRosetta}_{\text{full}}$	51.8	80.1	53.2	77.5
Co-evolutionary	40.1	65.2	47.3	72.1
ProTrans-T5	25.0	41.4	40.8	63.3
ESM-1b	28.2	50.2	44.4	68.4
MSA Transformer*	54.5	**80.2**	53.6	78.0
MSA Transformer†	**54.6**	77.5	**55.8**	**79.1**

database version used was released prior to CASP13, there should be no train/test leakage for the results on CASP13 in Table 4.2.

Supervised Contact Prediction

Used independently, features from current state-of-the-art protein language models fall short of co-evolutionary features from Potts models on supervised contact prediction tasks [43].

We evaluate the MSA Transformer as a component of a supervised structure prediction pipeline. Following Rives et al. [43], we train a deep residual network with 32 pre-activation blocks, each with a filter size of 64, using learning rate 0.001. The network is supervised with binned pairwise distance distributions (distograms) using the trRosetta training set [69] of 15051 MSAs and structures.

We evaluate two different ways of extracting features from the model. In the first, we

Table 4.4: CB513 8-class secondary structure prediction accuracy.

Model	CB513
Netsurf	72.1
HMM Profile	71.2 ± 0.1
ProTrans-T5	71.4 ± 0.3
ESM-1b	71.6 ± 0.1
MSA Transformer	$\mathbf{73.4 \pm 0.3}$

use the outer concatenation of the output embeddings of the query sequence. In the second, we combine the outer concatenation with the symmetrized row self-attention maps. For comparison, we train the same residual network over co-evolutionary features from Potts models [88]. Additionally we compare to features from state-of-the-art protein language models ESM-1b and ProTrans-T5 using the outer concatenation of the sequence embeddings. Dropout of 0.1 is used for all language model-based contact predictors. We also compare to trRosetta [69], a state-of-the-art supervised structure prediction method prior to AlphaFold2 [108].

The MSA Transformer produces a substantial improvement over co-evolutionary features for supervised contact prediction. Table 4.3 shows a comparison between the models on the CASP13-FM and CAMEO test sets. The best MSA Transformer model, using the combination of attention maps with features from the final hidden layer, outperforms all other models including the trRosetta baseline model (which uses 36 residual blocks) and the trRosetta full model (which uses 61 residual blocks, data augmentation via MSA subsampling, and predicts inter-residue orientations). Model ensembling over all 5 released models is used in the evaluation of the trRosetta models. Table C.4 gives additional comparisons with LSTM and transformer protein language models available in the literature.

Secondary Structure Prediction

To further evaluate the quality of representations generated by the MSA Transformer, we train a state-of-the-art downstream head based on the Netsurf architecture [47]. The downstream model is trained to predict 8-class secondary structure from the pretrained representations. We evaluate models on the CB513 test set [49].

The models are trained on the Netsurf training dataset. Representations from the MSA Transformer (72.9%) surpass the performance of HMM profiles (71.2%) and ESM-1b embeddings (71.6%) (Table 4.4).

Ablation Study

We perform an ablation study over seven model hyperparameters, using unsupervised contact prediction on the validation set for evaluation. For each combination of hyperparameters, a model is pre-trained with the masked language modeling objective for 100k updates. Training curves for the models are shown in Fig. C.3 and Top-L long-range precision is reported in Table C.2.

The ablation studies show the use of tied attention plays a critical role in model performance. After 100k updates, a model trained with square-root normalized tied attention outperforms untied attention by more than 17 points and outperforms mean normalized tied-attention by more than 6 points on long-range contact precision.

Parameter count also affects contact precision. A model with half the embedding size (384) and only 30M parameters reaches a long-range precision of 52.8 after 100k updates, 3.5 points lower than the base model, yet 11.7 points higher than the state-of-the-art 650M parameter single-seequence model. See Appendix C.3 for further discussion.

4.5 Model Analysis

We examine how the model uses its input MSA in experiments to understand the role of sequence diversity, attention patterns, and covariation in the MSA.

Effect of MSA diversity

The diversity of the input sequences strongly influences inference of structure. We explore three inference time strategies to control the diversity of the input sequence sets: (i) diversity maximizing, (ii) diversity minimizing, and (iii) random selection (see Section 4.3). Fig. 4.4 shows average performance across the test set for each selection strategy as the number of sequences used for input increases. Two approaches to maximize diversity, greedy hamming distance maximization and hhfilter, perform equivalently. Both strategies surpass ESM-1b performance with just 16 input sequences. In comparison, the diversity minimizing strategy, hamming distance minimization, performs poorly, requiring 256 sequences to surpass ESM-1b. Random selection performs well, although it falls behind the diversity maximizing strategies. The qualitative effects of MSA diversity are illustrated in Fig. 4.3, where the addition of just one high-diversity sequence outperforms the addition of 31 low-diversity sequences.

In principle, the model's attention could allow it to identify and focus on the most informative parts of the input MSA. We find row attention heads that preferentially attend to highly variable columns. We also identify specific column attention heads that attend to more informative sequences. In this experiment random subsampling is used to select inputs for the model. Fig. 4.5 compares the distribution of attention weights with two measures of MSA diversity: (i) per-column entropy of the MSA; and (ii) computed sequence weights (Appendix C.5). Per column entropy gives a measure of how variable a position is in the MSA. Computed sequence weights measure how informative a sequence is in the context of

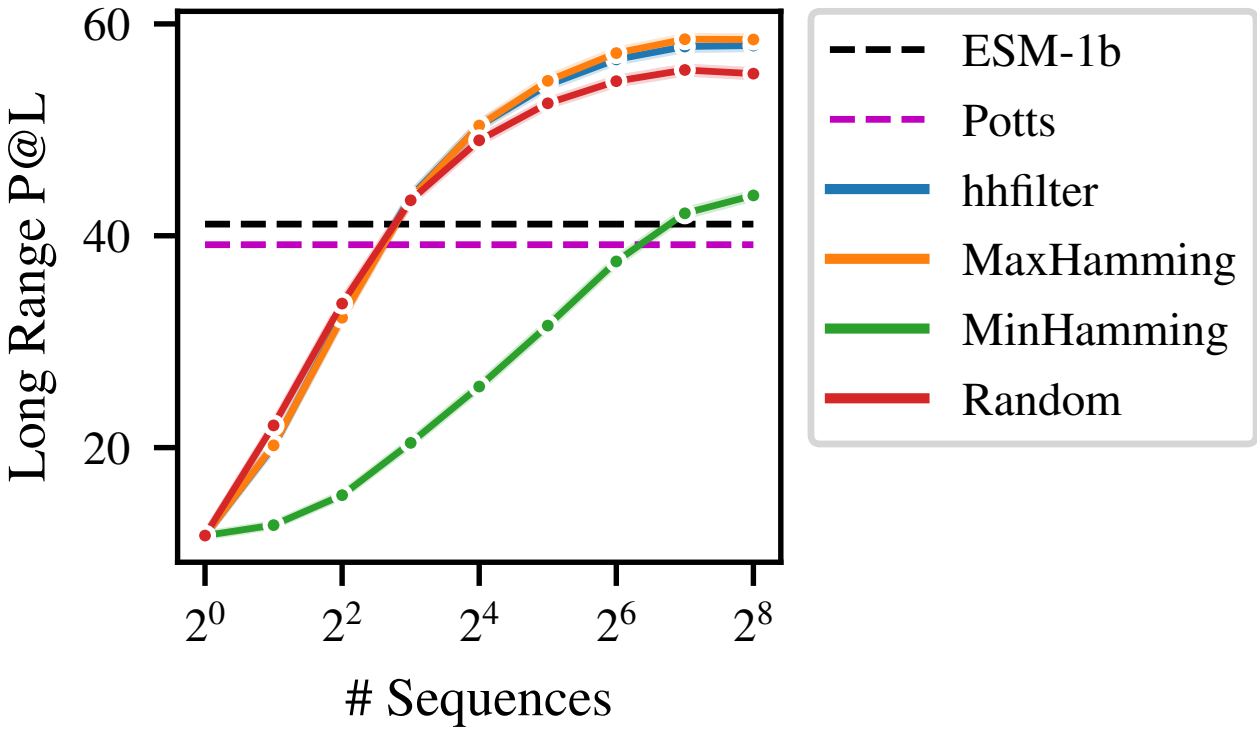


Figure 4.4: Comparison of MSA selection strategies. Model performance increases with more sequences. Selection strategies that maximize diversity of the input (MaxHamming and hhfilter) perform best. Random selection is nearly as good, suggesting the model has learned to compensate for the varying diversity during training time. Minimizing diversity performs worst. Using diversity maximizing approaches the MSA Transformer outperforms ESM-1b and Potts baselines using just 16 input sequences.

the other sequences in the MSA. Sequences with few similar sequences receive high weights. The maximum average Pearson correlation between a row attention head and column entropy is 0.59. The maximum average Pearson correlation between a column attention head and sequence weights is 0.58. These correlations between attention weights and measures of MSA diversity suggest the model is specifically looking for informative sequences when processing the input.

Attention Corresponds to Protein Contacts

In Section 4.4, we use the heads in the model's tied row attention directly to predict contacts in the protein's three-dimensional folded structure. Following methodology developed in Chapter 3, we fit a sparse logistic regression to the model's row attention maps to identify heads that correspond with contacts. Fig. C.1 shows the weight values in the learned sparse logistic regression fit using 20 structures. A sparse subset (45 / 144) of heads are predictive

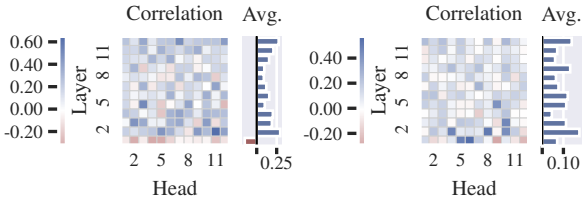


Figure 4.5: **Left:** Average correlation between row-attention and column entropy. This is computed by taking an average over the first dimension of each $L \times L$ row-attention map and computing correlation with per-column entropy of the MSA. **Right:** Average correlation between column-attention and sequence weights. This is computed by taking an average over the first two dimensions for each $L \times M \times M$ column-attention map and computing correlation with sequence weights (see Appendix C.5). Both quantities are measures of MSA diversity. The relatively high correlation (> 0.57) of some attention heads to these measures suggests the model explicitly looks at diverse sequences.

of protein contacts. The most predictive heads are concentrated in the final layers.

Inference: Covariance vs. Sequence Patterns

Potts models and single-sequence language models predict protein contacts in fundamentally different ways. Potts models are trained on a single MSA; they extract information directly from the covariance between mutations in columns of the MSA. Single-sequence language models do not have access to the MSA, and instead make predictions based on patterns seen during training. The MSA Transformer may use both covariance-based and pattern-based inference. To disentangle the two modes, we independently ablate the covariance and sequence patterns in the model's input via random shuffling. To ensure that there is enough information in the input for covariance-based extraction to succeed, we subsample each MSA to 1024 sequences using hhfilter, using only MSAs with at least 1024 sequences, and apply the model to unshuffled and shuffled inputs.

To remove covariance information, we randomly permute the values in each column of the MSA. This preserves per-column amino acid frequencies (PSSM information) while destroying pairwise correlations between columns. Under this condition, Potts model performance drops to the random guess baseline. Since ESM-1b takes a single sequence as input, the permutation trivially produces the same sequence, and the result is unaffected. Unlike the Potts model, the MSA Transformer retains some ability to predict contacts, which increases sharply as a function of MSA Depth. This indicates that the model can make predictions from patterns in the sequence profile in the absence of covariance.

To remove sequence patterns seen during training, we randomly permute the order of positions (columns) in the MSA. This preserves all covariance information between pairs of columns, but results in a scrambled input dissimilar to a real protein. Under this condition, Potts model performance is unaffected since its parameterization is invariant to sequence

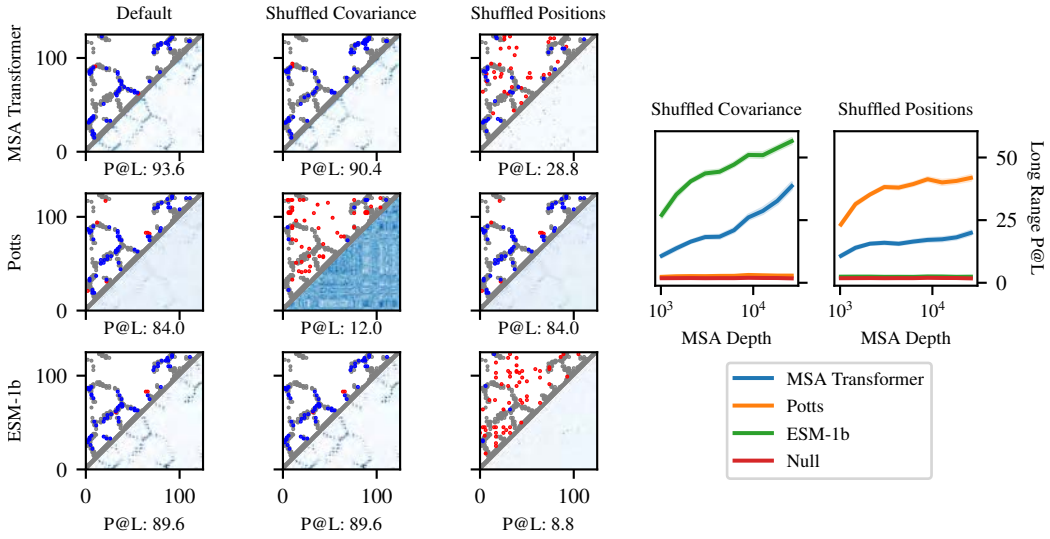


Figure 4.6: The MSA Transformer uses both covariance and similarity to training sequences to perform inference. **Left:** Examples (pdbid: 5ahw, chain: A) of model performance after independently shuffling each column of an MSA to destroy covariance information, and after independently permuting the order of positions to destroy sequence patterns. The MSA Transformer maintains reasonable performance under both conditions. A Potts model fails on the covariance-shuffled MSA, while a single-sequence language model (ESM-1b) fails on the position-shuffled sequence. **Right:** Model performance before and after shuffling, binned by depth of the original (non-subsampled) MSA. 1024 sequence selected with hhfilter are used as input to MSA Transformer and Potts models. MSAs with fewer than 1024 sequences are not considered in this analysis. Average Top-L long-range precision drops from 52.9 (no ablation) to 15.9 (shuffled covariance) and 27.9 (shuffled positions) respectively. A Null (random guessing) baseline is also considered. Potts model performance drops to the Null baseline under the first condition and ESM-1b performance drops to the Null baseline under the second condition. The MSA Transformer produces reasonable predictions under both scenarios, implying it uses both modes of inference.

order. ESM-1b performance drops to the random guess baseline. The MSA Transformer does depend on sequence order, and predicts spurious contacts along the diagonal of the reordered sequence. When predicted contacts with sequence separation < 6 are removed, the remaining predictions align with the correct contacts. This shows the model can predict directly from covariance when presented with sequence patterns unobserved in training.

Together these ablations independently destroy the information used by Potts models and single-sequence language models, respectively. Under both conditions, the MSA Transformer maintains some capability to predict contacts, demonstrating that it uses both modes of inference.

4.6 Discussion

Prior work in unsupervised protein language modeling has focused on inference from individual sequences. In this Chapter, we study an approach to perform inference from a set of aligned sequences in an MSA. We use axial attention to efficiently parameterize attention over the rows and columns of the MSA. This approach enables the model to extract information from dependencies in the input set and generalize patterns across MSAs. We find the internal representations of the model enable state-of-the-art unsupervised structure learning with an order of magnitude fewer parameters than current protein language models.

While supervised methods have produced breakthrough results for protein structure prediction [108], unsupervised learning provides a way to extract the information contained in massive datasets of sequences produced by low cost gene sequencing. Unsupervised methods can learn from billions of sequences, enabling generalization to regions of sequence space not covered by structural knowledge.

Models fit to MSAs are widely used in computational biology including in applications such as fitness landscape prediction [2], pathogenicity prediction [121, 122], remote homology detection [51], and protein design [64]. The improvements we observe for structure learning suggest the unsupervised language modeling approach here could also apply to these problems.

Improvement in unsupervised learning of structure and function with protein language models has been linked to scale of the models [43]. Further scaling the approach studied here in the number of parameters and input sequences is a potential direction for investigating the limits of unsupervised learning for protein sequences.

Chapter 5

Language models enable zero-shot prediction of the effects of mutations on protein function

5.1 Introduction

Proteins have a myriad of diverse functions that underlie the complexity of life. Protein sequences encode function via structure through the spontaneous folding of the sequence into the three dimensional structure of the protein [123]. The effects of sequence mutations on function form a landscape that reveals how function constrains sequence. Alterations at some sites in a protein sequence cannot be tolerated because they are essential to the protein's function. Other sites evolve together because the structure and function is determined by them collectively. Mutations can enhance the activity of a protein, attenuate it, or leave it unchanged.

The functional effect of sequence variations can be measured through deep mutational scanning experiments [124]. Consisting of thousands to hundreds of thousands of measurements of protein function, deep mutational scans give insight into the intrinsic constraints on a protein's structure and function. Due to the cost and difficulty of implementing such experiments, compilations of deep mutational scanning data include experiments on a few dozens of proteins at most, relative to the tens of thousands of proteins encoded in the human genome, and the millions more across the tree of life that we would like to understand.

A model that learns the landscape linking sequence to function can provide insight into function without having to do experiments. Unsupervised models of mutational effects can be learned from sequences [1, 125]. Statistical patterns in a family of evolutionarily related protein sequences contain information about structure and function [18, 19, 126]. This is because the properties of a protein act as constraints on the selection of sequences through evolution [103].

In the natural language modeling community, there has been interest in zero-shot transfer

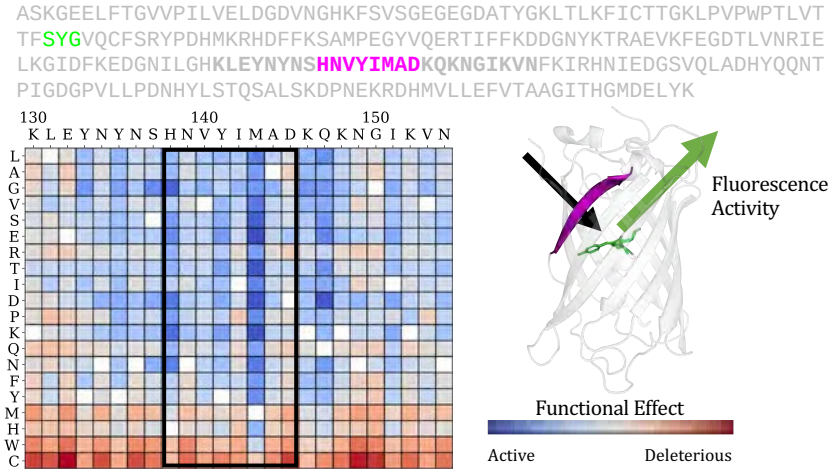


Figure 5.1: Depiction of a mutational effect prediction task. The objective is to score the effect of sequence mutations on the function of a protein. Deep mutational scanning experiments provide ground truth experimental measurements of the protein's function (fluorescence activity in the example here) for a large set of single mutations or combinations of mutations. For each protein, the prediction task is to score each possible mutation and rank its relative activity. Predictions for single substitutions can be described in a score matrix. The columns are the positions in the sequence. The rows are the possible variations at each position.

of models to new tasks. Massive language models can solve tasks they haven't been directly trained on [20, 102, 127]. Recent protein language models, introduced in Chapter 4 and elsewhere, have achieved state-of-the-art in various structure prediction tasks [43, 94]. Work to date has mainly focused on transfer in the classical representation learning setting, using pre-trained features with supervision on the downstream task.

In this Chapter we show that language models trained on large and diverse protein sequence databases can predict experimental measurements of protein function without further supervision. Prior work has focused on transferring the representations using supervision from experimental data [128, 129]. We find that language models can transfer to predict functional measurements without supervision. Language models perform *zero-shot* and *few-shot* prediction of mutational effects across a variety of proteins with widely differing functions. We perform experiments with state-of-the-art protein language models ESM-1b [43] and MSA Transformer from Chapter 4. We introduce a new protein language model, ESM-1v, with zero-shot performance comparable to state-of-the-art mutational effect predictors. Performance can be further improved by fine-tuning the model with sequences from the protein family. Predictions capture the functional landscape of the protein, correlate with amino acid conservation patterns in the core and surface, and identify residues responsible for binding and activity.

5.2 Zero-shot transfer

Zero-shot learning has classically described the extension of a classifier to a new set of classes that have not been seen in training [130]. In natural language processing this idea has been extended to describe the transfer of models to entirely new tasks without further training. Proposed as zero-data learning by Larochelle et al. [131], this perspective on transfer has been at the center of recent work understanding the generalization capabilities of large language models [20, 102, 127, 132]. The distinction from representation learning is that the models are used *directly* without additional supervision for the task. This means that the tasks must be learned purely from pre-training.

In this work we take a similar perspective on zero-shot transfer to that of GPT-3, described in Brown et al. [102]. We define zero-shot transfer to be transfer of a model to a new task without any further supervision to specialize the model to the task. We also consider the closely related idea of few-shot transfer. Here as in Brown et al. [102] we define the few-shot setting to be one in which a few positive examples are given to the model as inputs at inference time. As in the zero-shot setting, no gradient updates are performed to specialize the model. Similar to Brown et al. [102], the claim is not one of out-of-distribution generalization. The assumption is that in the pre-training stage, the model learns information relevant to the tasks to which it will later be transferred. In the case of protein language models, the pre-training dataset includes sequences from across evolution, which implies the model may see examples of sequences from protein families on which it will be evaluated. The essential departure from the standard approach in computational biology is that the model is *general purpose* and can be applied across a variety of tasks without specialization.

Measurements of function, a property of central importance to the understanding and design of proteins, are a practical ground for studying the generalization capability of protein language models. Deep mutational scanning experiments measure the effects of thousands to hundreds of thousands of mutations on a single protein, and have been performed on a variety of proteins having different functions and using various forms of experimental measurement. We study zero-shot and few-shot transfer of protein language models to function prediction using this data.

Supervised methods trained with data from experimental measurements [128, 129], and unsupervised methods trained only on sequences [1, 125] have been developed for prediction of mutational effects. Unsupervised mutational effect predictors are trained as task specific models on sequences from an individual protein family. In this view every protein is an independent prediction task where the objective is to score the effect of mutations on the protein's function. While mutational effect predictors trained on multiple sequence alignments (MSAs) are typically described as unsupervised, they can also be seen as weakly supervised. Hsu et al. [128] observe that such models have weak supervision on the task through the MSA, which describes the fitness landscape of the protein through positive examples.

If protein language models can learn the information necessary to solve a task from pre-training, then they can be applied *directly* to new instances of the task, without specialization. This would mean that in practice a single general purpose model can be trained once and

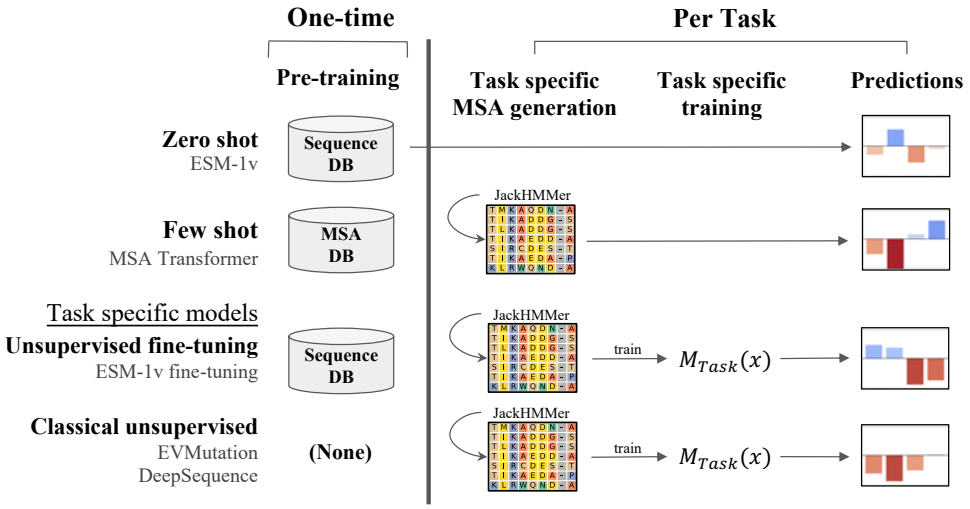


Figure 5.2: Steps involved in variant effect prediction methods. Compared with EVMutation [1] and DeepSequence [2], MSA Transformer and ESM-1v require no task-specific model training for inference. Moreover, ESM-1v does not require MSA generation.

then applied to a variety of possible tasks. Thus zero-shot and few-shot transfer represent fundamentally new unsupervised learning capabilities that protein language models can bring to the computational biology toolkit.

5.3 Method

Protein language models trained with the masked language modeling objective are supervised to output the probability that an amino acid occurs at a position in a protein given the surrounding context. We use this capability to score sequence variations. For a given mutation we can consider the amino acid in the wildtype protein as a reference state, comparing the probability assigned to the mutated amino acid with the probability assigned to the wildtype.

We score mutations using the log odds ratio at the mutated position, assuming an additive model when multiple mutations T exist in the same sequence:

$$\sum_{t \in T} \log p(x_t = x_t^{mt} | x_{\setminus T}) - \log p(x_t = x_t^{wt} | x_{\setminus T}) \tag{5.1}$$

Here the sum is over the mutated positions, and the sequence input to the model is masked at every mutated position.

Zero-shot and few-shot transfer

In the zero-shot setting, inference is performed directly on the sequence to be evaluated. Since the MSA Transformer can take multiple sequences as input at inference time, we use this model in the few-shot setting, where additional sequences from the protein family are provided along with the sequence to be evaluated. In both the zero-shot and few-shot settings, only forward passes of the models are performed during inference; no gradient updates are taken. Fig. 5.2 illustrates the approach in comparison to the current practice of fitting a new model for each task.

Inference efficiency

Inference with ESM-1v is more efficient than current state-of-the-art methods. This is a result of two important differences: (i) the effect of mutations can be inferred directly without training a task-specific model; (ii) fitness landscapes can be predicted with a single forward pass. Time requirements are summarized in Fig. D.1.

Scoring with MSA Transformer

We score mutations with MSA Transformer using the log odds ratio and additive model in Eq. (5.1). However, since MSA Transformer uses a set of sequences for inference, we input the sequence to be evaluated as the first sequence, and provide additional sequences from the MSA as context. Masking and scoring are performed on the first sequence only.

5.4 Results

Experimental setup

Prediction Models We compare to state-of-the-art unsupervised variant prediction methods, EVMutation [1] and DeepSequence [2]. We also examine performance of a variety of protein language models that have been recently introduced in the literature.

The position specific scoring matrix (PSSM), EVmutation [1], and DeepSequence [2] methods are all MSA based. The PSSM treats each position in the sequence independently, factorizing the likelihood into one term per sequence position. EVmutation is a Potts model, which adds pairwise terms modeling the interactions between positions. DeepSequence introduces a latent code, allowing potential higher-order interactions between positions.

UniRep [25], TAPE (Chapter 2), ProtBERT-BFD [94], ESM-1b [43], and ESM-1v (introduced here), are all single-sequence language models trained on large databases of unaligned and unrelated protein sequences (e.g. Pfam [46] or UniRef [71]). With the exception of UniRep, which is trained using next token prediction, all models are trained with masked language modeling [12].

Table 5.1: Comparison of protein language models to state-of-the-art methods. Average |Spearman ρ| on full and test sets. DeepSequence and ESM-1v models are each ensembles of 5 models. MSA Transformer is a single model, but is ensembled across 5 random samples of the MSA.

Models	**Full**	**Test**
PSSM	0.460	0.460
EVMutation (published)	0.508	0.495
EVMutation (replicated)	0.511	0.498
DeepSequence (published)	0.514	0.499
DeepSequence (replicated)	0.520	0.506
MSA Transformer	0.542	0.524
ESM-1v (zero shot)	0.509	0.482
ESM-1v (+further training)	0.538	0.519

Finally, the MSA Transformer introduced in Chapter 4 is a combination of both approaches; it is trained on a large database of MSAs using masked language modeling and takes an MSA as input during inference.

ESM-1v We train ESM-1v, a 650M parameter transformer language model for prediction of variant effects, on 98 million diverse protein sequences across evolution. The model is trained only on sequences, without any supervision from experimental measurements of function. We use Uniref90 `2020-03` [71], employing the ESM-1b architecture and masked language modeling approach of Rives et al. [43]. The model attains a perplexity of 7.29 on a set of held-out Uniref90 sequences (Table D.8). We train five models with different seeds to produce an ensemble.

Evaluation Models are evaluated on a set of 41 deep mutational scans collected by Riesselman et al. [2], which comprise a variety of tasks assessing a diverse set of proteins. Across tasks, the experiments differ in the functions tested and in the measurements performed. We treat each deep mutational scanning dataset as a separate prediction task, scoring each of the variants in the dataset with the model. The tasks are split into a validation set of ten mutational scanning datasets and a test set consisting of the remaining datasets. We evaluate performance by comparing the scores with the experimental measurements using Spearman rank correlation.

Comparisons Since the published versions of EVMutation and DeepSequence use MSAs generated from an earlier version of Uniref100, we generate new MSAs using EVMutation methodology and the version of Uniref100 concurrent with our pretraining dataset. We train

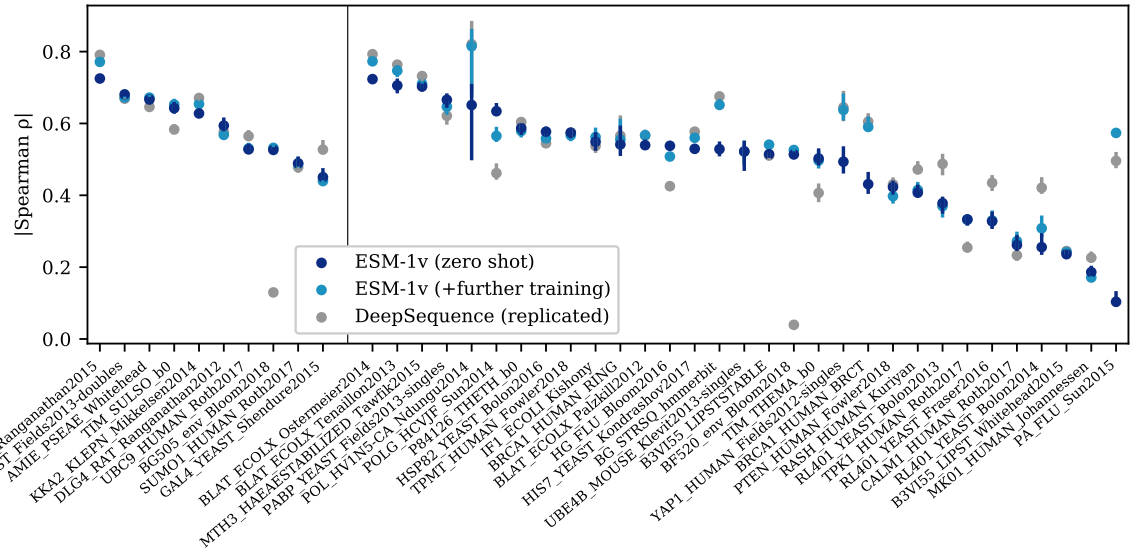


Figure 5.3: Per task performance. Comparison across 41 deep mutational scanning datasets. Points are |Spearman ρ| on each dataset, error bars show standard deviation of 20 bootstrapped samples. Validation proteins are shown to the left of the dividing line and test proteins to the right. In 17 out of the 41 tasks, ESM-1v zero-shot has a higher |Spearman ρ| than DeepSequence.

replications of EVMutation and DeepSequence using their open source code. The same MSAs are also used in few-shot experiments with MSA Transformer and unsupervised fine-tuning experiments with ESM-1v.

Language models enable zero-shot and few-shot prediction of the effects of mutations

ESM-1v and MSA Transformer models make state-of-the-art predictions. Table 5.1 compares overall performance of the models across the 41 mutational scanning datasets. Fig. 5.3 presents a comparison between ESM-1v and DeepSequence on each of the tasks. Zero-shot inference with ESM-1v has a better correlation with experimental measurements than DeepSequence on 17 of the 41 datasets. The two methods are not statistically distinguishable via a paired t-test.

Table 5.2 compares protein language models in the zero-shot setting. ESM-1v outperforms existing protein language models TAPE, UniRep [25], ProtBERT-BFD [94], and ESM-1b [43]. Fig. D.2 breaks down performance across each of the tasks.

Table 5.2: Zero-shot performance. Average |Spearman ρ| on full and test sets. †Average performance of five ESM-1v models. *Ensemble of the five ESM-1v models.

Models	**Full**	**Test**
UniRep	0.156	0.151
TAPE	0.171	0.175
ProtBERT-BFD	0.428	0.399
ESM-1b	0.459	0.424
ESM-1v†	0.484	0.457
ESM-1v*	0.509	0.482

Pre-training data We examine the effect of the clustering level of pre-training data. Fig. 5.4 compares models pre-trained on datasets clustered at increasing sequence identity thresholds. ESM-1b is trained on sequences clustered at a 50% identity threshold. Improvements are seen using a 70% threshold with greatest improvement at 90%. Uniref100 performance appears to deteriorate early in training despite being the largest of the datasets. These results establish a link between model performance and the data distribution, highlighting the importance of training data in the design of protein language models.

Scoring methods We compare four scoring methods on the validation set - masked marginals, wildtype marginals, mutant marginals, and psuedolikelihood. Table D.3 shows that the masked marginal approach described in Eq. (5.1) outperforms other scoring methods, including ones in which the likelihood changes at non-mutated positions are considered. The scoring methods are described in detail in Appendix D.1.

Parameter count Previous work in Chapter 3 and Rives et al. [43] has established a link between model scale and learning of protein structure. We examine zero-shot transfer performance as a function of parameter count. We train models using the same width, depth, and learning rate as described in Henighan et al. [6], observing improvements with scale (Fig. D.3). These findings suggest that continued scaling of the models will further improve results.

MSA Transformer

We examine how the sequences provided to MSA Transformer affect few-shot transfer. Table D.6 compares sequence selection methods that vary the diversity of the sequences. Providing a more diverse set of sequences improves few-shot performance. Selecting a set of sequences to maximize diversity outperforms selecting a diversity minimizing set of sequences. Random sampling performs even better, and sampling sequences according to sequence weights [86] performs best.

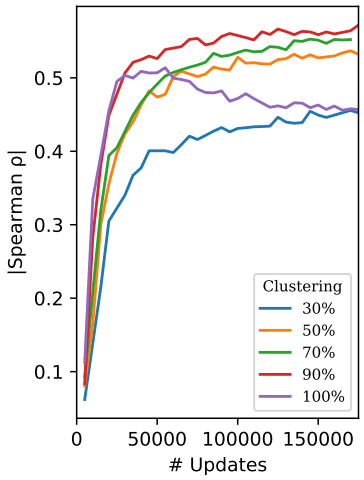


Figure 5.4: Comparison of pre-training datasets. Average |Spearman ρ| on the single-mutation validation set. While a 50% clustering threshold was used for ESM-1b, training with 90% clustering results in a significant improvement on variant prediction tasks. Notably, models trained on Uniref100, the largest dataset in this figure, appear to deteriorate early in training. These results establish a link between model performance and the data distribution, and highlight the importance of training data in the design of protein language models.

We also vary the number of sequences used for inference. Fig. D.5 shows few-shot performance as a function of the number of sequences given as input. The model performs well using only a few sequences, but performs best with 384 total sequences. In the main tables we report results sampling 384 sequences using sequence reweighting and ensembling predictions over five different subsamples from the MSA.

Unsupervised fine-tuning on MSAs

While ESM-1v performs well when evaluated in the zero-shot setting, we explore whether results can be improved by fine-tuning on the MSA. Fine-tuning on MSAs has been used in previous work [25, 129] as a stage in transfer learning to specialize a pre-trained model to a protein family, before applying supervision with labeled data. Here we consider using the fine-tuned model to make unsupervised predictions directly, without adding supervision from experimental data.

We observe that naively fine-tuning the model on the MSA results in rapid overfitting and poor performance on the prediction tasks (Fig. D.6). While we experiment with a variety of approaches to freezing parameters during fine-tuning, detailed in Appendix D.2, none produce significant improvements. We find that an approach using pre-training sequences to regularize the fine-tuning performs well and enables training of all parameters without overfitting (Fig. D.7). Spiked fine-tuning improves average absolute Spearman rho on the full

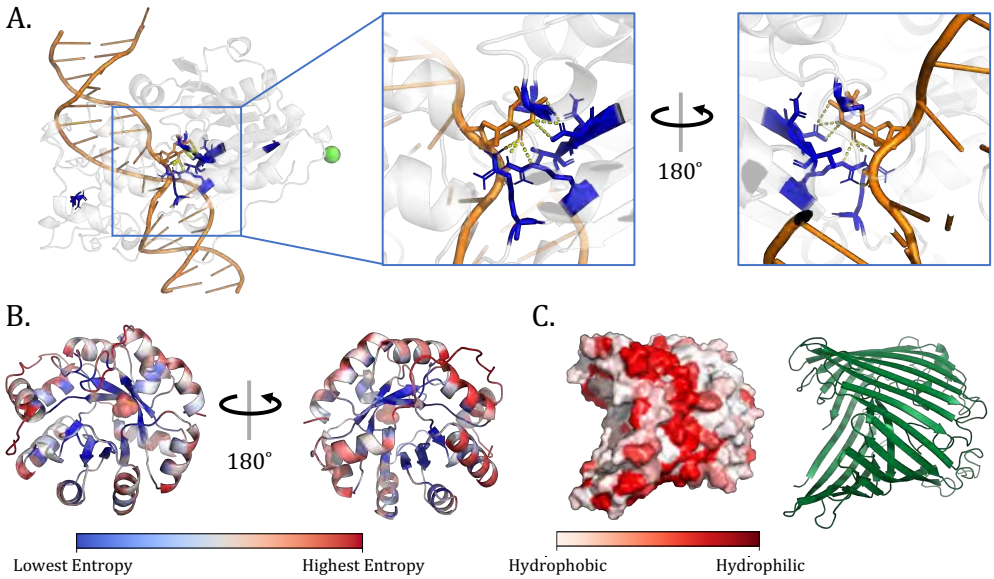


Figure 5.5: ESM-1v reflects the molecular basis of function in proteins. **(A)** DNA methylase HaeIII (pdbid: 1DCT [3]). Side chains for the top 10 positions with lowest prediction entropy shown in blue. Low-entropy positions cluster in the active site. **(B)** TIM Barrel (pdbid: 1IGS [4]) with residues colored by entropy. The model's predictions for residues on the surface have highest entropy (red) while those in the core have lower entropy (blue). Notably, residues on the alpha helices show a clear gradient from high to low entropy as residues transition from surface-facing to core-facing. **(C)** Sucrose-specific Porin (pdbid: 1A0T [5]), a transmembrane protein. The model predicts a hydrophobic band where the protein is embedded in the membrane.

dataset from 0.510 for zero-shot evaluation to 0.537 with fine-tuning.

5.5 Analysis of models

Protein structure and function ESM-1v probabilities reflect the functional properties of sites within the protein. We use the entropy of the model's predictions for a position as a measure of its estimation of conservation. The lowest entropy predictions cluster at binding sites. Fig. D.8 compares the distribution of the model's entropy between binding sites and non-binding sites. A significant difference is observed between the entropy assignment to binding and non-binding site residues. Fig. 5.5 visualizes the side chains of the 10 lowest entropy residues as predicted by the model on the crystal structure of DNA methyltransferase M.HaeIII interacting with its DNA substrate. In the crystal structure a cytosine of the

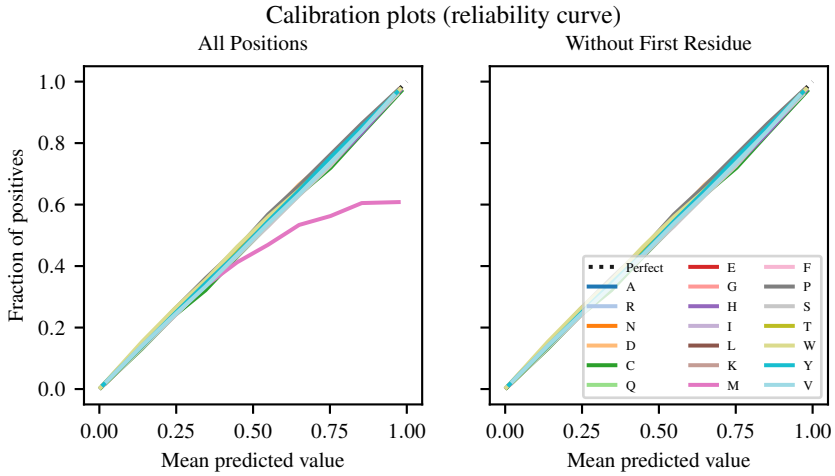


Figure 5.6: Calibration plot for ESM-1v predictions on each of the 20 naturally occurring amino acids on the trRosetta dataset. The multi-class classification is converted into a set of 20 one-versus-all classifications for the purpose of this analysis. Left and right plots show calibration of all positions and positions excluding the first residue, respectively. Since full sequences always start with Methionine, the model overwhelmingly predicts it in the first position. When evaluating the model on subsequences, such as those in the trRosetta dataset, this causes a miscalibration at the first residue. Including the first residue, the model has an average calibration error (ACE) of 0.011 in the first case and 0.006 in the second.

substrate is inserted into the active site of the enzyme. The low entropy residues cluster in the active site and interact with the cytosine. Additional examples are visualized in Fig. D.12.

The model probabilities also correspond to structure. Fig. D.9 compares the entropy assigned to sites that are buried in the core of the protein vs. exposed on the surface. The model assigns significantly lower entropy to sites that are in the core of the protein, consistent with the idea that tight packing in the core places greater constraints on the selection of residues. Fig. 5.5B visualizes the entropy assigned by the model to each position overlayed on the structure of Indole-3-glycerolphosphate Synthase, a TIM barrel protein. Higher entropy is assigned to residues having outward facing side chains on the alpha helices, while lower entropy is assigned to the inward facing positions. Fig. D.11 compares the probability assigned to hydrophobic, polar, and charged amino acids for buried sites vs. non-buried sites. The model prefers hydrophobic residues in the core and hydrophilic residues on the surface. The model probabilities closely match the empirical probabilities and those from the PSSM. Fig. 5.5C visualizes probability assigned to hydrophobic amino acids on the structure of Sucrose-specific Porin, a transmembrane protein. The model predicts a hydrophobic band in the center where the protein embeds in the membrane.

Calibration We evaluate model calibration using 15008 sequences with length < 1024 from the trRosetta [133] dataset. ESM-1v probabilities for each amino acid at each position are calculated with the masked marginal probability in Eq. (5.1). Fig. 5.6 shows that the model is generally well calibrated for all amino acids except Methionine. ESM-1v always predicts Methionine as the first position in the sequence since full protein sequences always start with it, so care must be used when applying the model to subsequences. When excepting the first residue, the model achieves an average calibration error (defined for the multi-class setting in Appendix D.4) of 0.006.

We also explore the relationship between conservation (entropy of the PSSM) and the model's predicted entropy. Fig. D.10 shows that these are well correlated (Pearson's $r = 0.44$), suggesting the model is able to identify conserved positions.

5.6 Related Work

Protein language models

In addition to Chapters 2 to 4, a number of groups have developed language models for protein sequences [24, 25, 43, 70, 94, 95]. These models have been used for many tasks, including supervised low-N function prediction [43, 129], remote homology detection [43], and protein generation [70]. The approach to the tasks typically involves transfer learning, where a pretrained language model is fine-tuned for a particular problem. Vig et al. [7] found that transformer attention corresponds to known biological properties such as structure and binding sites.

Mutation effect prediction

Supervised and unsupervised methods have been developed for prediction of mutational effects. Supervised methods train models using experimental measurements or labels from databases of clinical variants. Standard machine learning tools including linear regression, random forests, and support-vector machines can be used [133]. Models have been designed specifically for proteins, using feature engineering such as Envision [134] and PolyPhen-2 [135], ensemble methods such as Revel [136], MPC [137], CADD [138], and M-CAP [139], language models such as UniRep [25, 129] and ESM [43], and other representation learning approaches [121, 140].

Unsupervised mutation effect predictors work by inferring the likelihood of a mutation from the evolutionary landscape of the original protein. A density model fit to related sequences is used for scoring. SIFT [141] is a first order approach using a position-specific-scoring-matrix. EVMutation [1] extends this to a second-order approach by training a Potts model on the MSA. DeepSequence [2] includes higher-order interactions by training a VAE on the MSA instead, using the ELBO to score mutations. Riesselman et al. [107] proposes using an autoregressive model that does not require the sequences to be aligned.

Hsu et al. [128] show that unsupervised mutational effect predictors can be extended to perform supervised predictions, with better unsupervised predictors generally resulting in better supervised predictors. This suggests improving unsupervised prediction can drive progress in both settings. Concurrent with our work, Hie et al. [142] use open-source protein language models ESM-1b and TAPE to predict the direction of evolution in protein fitness landscapes.

5.7 Discussion

Advances in language modeling at scale are bringing the goal of a general purpose model for proteins closer to realization. This line of work aspires to a model that learns to read and write biology in its native language, that can be directly applied across a range of protein understanding and design tasks. For scalability, learning from sequences is important: while there are no central databases of high-throughput functional measurements, and few compilations exist, billions of sequences are available to learn from in sequence databases [143, 144]. Sequences give an unparalleled view into the vast diversity and complexity of molecular parts invented by nature through billions of years of evolution.

Unsupervised structure [73, 83–87] and function [1, 125] learning methods first effectively realized the idea that biological properties could be read directly from sequences without supervision from experimental measurements. However these methods are not general purpose in the sense that a specialized model must be trained for every protein for which a prediction is to be made. We show that the same performance can be realized by a general purpose model that has been trained across many diverse protein families. Similar to observations on the learning of tertiary protein structure in large language models (Chapter 3, [43]), we find that increasing the scale of models leads to improvements in function learning. The understanding of mutational landscapes in the models correlates with the molecular basis of function in proteins, capturing binding sites and amino acid preferences that are determined by the folded structure.

Zero-shot transfer is an interesting capability of large scale language models, and represents a major point of departure from the unsupervised learning methods that are the basis for current state-of-the-art inference of protein structure and function. The capability for zero-shot transfer implies that a model can be trained once and then applied to perform inference for many tasks. It is also a window into deeper questions about the forms of generalization that are possible in learning from sequences. Reading structural and functional design principles from sequences is a necessary capability for writing new biologically active sequences. Generalization in the zero-shot setting suggests the potential for large language models to capture knowledge that can be transferred to generating new functional proteins.

Chapter 6

Conclusions

6.1 Future Work

The developments proposed in this dissertation, along with advances such as AlphaFold [108] (which combined an MSA Transformer-like architecture and loss with additional equivariant modeling and supervised losses), shows that computational modeling (and specifically deep neural network-based modeling) of proteins can produce strong results.

Variant Prediction

Chapter 5 looks at the performance of unsupervised language model for variant effect prediction, specifically in the context of protein design. However, variant prediction can also be of enormous clinical value [175]. As sequencing costs decrease, genome sequencing will become a more and more common diagonstic tool. Improving the accuracy with which we are able to predict the effects of mutations would enable the early detection and treatment of rare genetic mutations for a large swath of the population. Expanding current models from proteins to nucleotides could also have an impact, allowing the detection of mutational effects of synonymous substitutions or in non-coding regions.

Representation Learning

Current neural models of proteins are largely based on evolutionary relationships between sequences. Evolutionary information is clearly powerful, as shown both in this work and elsewhere, and is able to capture significant information about protein structure and function.

However, as we move further from the realm of natural proteins and towards designed proteins, evolutionary priors begin to lose some of their utility. Will this protein bind to a novel ligand, one which may never have been seen in nature? Will this protein be stable at 100C? Will this protein be soluble at very high concentrations?

Answering these questions will require new model architectures, training schemes, and better representations. These advances may come in the form of joint representations of

proteins and ligands, physical priors and attempts to encode kinematics, or multitask learning on other metadata and computational predictions.

The ultimate goal of protein representation learning should not be to predict structure, stability, or binding affinity. Rather the goal is to create an invertible mapping between sequence and any arbitrary function, while also being sensitive to changes in the environment. Even the best neural models today fail to account for temperature, concentration, pH, or the presence of other molecules. There's still a long way to go.

6.2 Reflections

This dissertation started with three questions:

1. Do standard approaches to unsupervised learning in NLP learn biologically relevant features?
2. How can we tailor the data, model, and tasks used to train unsupervised models for proteins?
3. Can large scale unsupervised models for protein sequences be made useful for protein design?

Over the course of this work, methods for training and evaluating language models on protein sequences have been developed. Language models have been shown to learn important information regarding the structure and function of protein sequences with no explicit supervisory signal. Effects of model scale, data deduplication, and the use of MSAs have been investigated.

Additionally, multiple publicly available artifacts have been produced, including the TAPE benchmark suite, the TAPE model repository, the MSA Transformer, ESM-1v, and scripts for contact and variant effect prediction. These projects have already directly enabled additional research, from further analysis of protein language models [7, 176], to improving machine learning directed evolution [177].

I would therefore answer the first question with a resounding *yes*. It is clear that language models learn biologically relevant information. To the second question we have some answers. Deduplicating data and incorporating MSAs play key roles in improving the features learned by large scale models, but there are multiple avenues of optimization left to explore. And to the third question, I would say we are only just getting started. With the release of AlphaFold [108], deep neural network-based modeling of proteins is poised to become a major component of protein design. More work on representation learning, transfer learning, and evolutionary modeling will be required to realize the ultimate goal of fully *de novo* design.

Bibliography

[1] Thomas A. Hopf, John B. Ingraham, Frank J. Poelwijk, Charlotta P.I. Schärfe, Michael Springer, Chris Sander, and Debora S. Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35(2):128–135, February 2017. ISSN 15461696. doi: 10.1038/nbt.3769. URL `http://www.nature.com/articles/nbt.3769`. Publisher: Nature Publishing Group.

[2] Adam J. Riesselman, John B. Ingraham, and Debora S. Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature Methods*, 15(10): 816–822, October 2018. ISSN 15487105. doi: 10.1038/s41592-018-0138-4. URL `http://www.nature.com/articles/s41592-018-0138-4`. Publisher: Nature Publishing Group.

[3] KM Reinish, L Chen, GL Verdine, and WN Lipscomb. The crystal structure of HaeIII methyltransferase covalently complexed to DNA: an extrahelical cytosine and rearranged base pairing. *Cell*, 82(1):143–153, 1995.

[4] Michael Hennig, Beatrice Darimont, Reinhard Sterner, Kasper Kirschner, and Johan N Jansonius. 2.0 Å structure of indole-3-glycerol phosphate synthase from the hyperthermophile Sulfolobus solfataricus: possible determinants of protein stability. *Structure*, 3 (12):1295–1306, 1995. Publisher: Elsevier.

[5] Doris Forst, Wolfram Welte, Thomas Wacker, and Kay Diederichs. Structure of the sucrose-specific porin ScrY from Salmonella typhimurium and its complex with sucrose. *Nature structural biology*, 5(1):37–46, 1998. Publisher: Nature Publishing Group.

[6] Tom Henighan, Jared Kaplan, Mor Katz, Mark Chen, Christopher Hesse, Jacob Jackson, Heewoo Jun, Tom B. Brown, Prafulla Dhariwal, Scott Gray, Chris Hallacy, Benjamin Mann, Alec Radford, Aditya Ramesh, Nick Ryder, Daniel M. Ziegler, John Schulman, Dario Amodei, and Sam McCandlish. Scaling Laws for Autoregressive Generative Modeling. *CoRR*, abs/2010.14701, 2020. URL `https://arxiv.org/abs/2010.14701`. _eprint: 2010.14701.

[7] Jesse Vig, Ali Madani, Lav R. Varshney, Caiming Xiong, Richard Socher, and Nazneen Rajani. BERTology Meets Biology: Interpreting Attention in Protein Language Models. September 2020. URL `https://openreview.net/forum?id=YWtLZvLmud7`.

[8] Andrew Leaver-Fay, Matthew J O'meara, Mike Tyka, Ron Jacak, Yifan Song, Elizabeth H Kellogg, James Thompson, Ian W Davis, Roland A Pache, Sergey Lyskov, and others. Scientific benchmarks for guiding macromolecular energy function improvement. *Methods in enzymology*, 523:109–143, 2013. Publisher: Elsevier.

[9] Didier Nurizzo, Steven C Shewry, Michael H Perlin, Scott A Brown, Jaydev N Dholakia, Roy L Fuchs, Taru Deva, Edward N Baker, and Clyde A Smith. The crystal structure of aminoglycoside-3ı-phosphotransferase-IIa, an enzyme responsible for antibiotic resistance. *Journal of molecular biology*, 327(2):491–506, 2003. Publisher: Elsevier.

[10] L. Steven Johnson, Sean R. Eddy, and Elon Portugaly. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics*, 11(1): 431, August 2010. ISSN 14712105. doi: 10.1186/1471-2105-11-431. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-431`. Publisher: BioMed Central.

[11] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep Contextualized Word Representations. In *Proceedings of the 2018 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://www.aclweb.org/anthology/N18-1202`.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North {A}merican Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL `http://arxiv.org/abs/1810.04805`.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. URL `https://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf`.

[14] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/`

9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[15] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*, July 2020. URL http://arxiv.org/abs/1910.03771. arXiv: 1910.03771.

[16] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge. *Nucleic Acids Research*, 47(D1):D506–D515, 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1049. URL https://doi.org/10.1093/nar/gky1049. _eprint: http://oup.prod.sis.lan/nar/article-pdf/47/D1/D506/27437297/gky1049.pdf.

[17] C B ANFINSEN, E HABER, M SELA, F H WHITE, and Jr. The kinetics of formation of native ribonuclease during oxidation of the reduced polypeptide chain. *Proceedings of the National Academy of Sciences of the United States of America*, 47(9):1309–14, September 1961. ISSN 0027-8424. doi: 10.1073/pnas.47.9.1309. URL http://www.ncbi.nlm.nih.gov/pubmed/13683522http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC223141. Publisher: National Academy of Sciences.

[18] C Yanofsky, V Horn, and D Thorpe. Protein Structure Relationships Revealed By Mutational Analysis. *Science (New York, N.Y.)*, 146(3651):1593–4, December 1964. ISSN 0036-8075. URL http://www.ncbi.nlm.nih.gov/pubmed/14224506.

[19] D Altschuh, T Vernet, P Berti, D Moras, and K Nagai. Coordinated amino acid changes in homologous protein families. *Protein Engineering, Design and Selection*, 2(3):193–199, 1988. ISSN 0269-2139. URL http://www.ncbi.nlm.nih.gov/pubmed/3237684. Publisher: Oxford University Press.

[20] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language Models are Unsupervised Multitask Learners. *OpenAI Blog*, 1:8, 2019.

[21] Timothy A Whitehead, Aaron Chevalier, Yifan Song, Cyrille Dreyfus, Sarel J Fleishman, Cecilia De Mattos, Chris A Myers, Hetunandan Kamisetty, Patrick Blair, Ian A Wilson, and others. Optimization of affinity, specificity and function of designed influenza inhibitors using deep sequencing. *Nature biotechnology*, 30(6):543, 2012. Publisher: Nature Publishing Group.

[22] Esther Vazquez, Neus Ferrer-Miralles, Ramon Mangues, Jose L Corchero, Jr Schwartz, Antonio Villaverde, and others. Modular protein engineering in emerging cancer

therapies. *Current pharmaceutical design*, 15(8):893–916, 2009. Publisher: Bentham Science Publishers.

[23] Nelson Perdigão, Julian Heinrich, Christian Stolte, Kenneth S. Sabir, Michael J. Buckley, Bruce Tabor, Beth Signal, Brian S. Gloss, Christopher J. Hammang, Burkhard Rost, Andrea Schafferhans, and Seán I. O'Donoghue. Unexpected features of the dark proteome. *Proceedings of the National Academy of Sciences*, 112(52):15898–15903, 2015. ISSN 0027-8424. doi: 10.1073/pnas.1508380112. URL `https://www.pnas.org/content/112/52/15898`. Publisher: National Academy of Sciences _eprint: https://www.pnas.org/content/112/52/15898.full.pdf.

[24] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=SygLehCqtm`.

[25] Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-only deep representation learning. *Nature Methods*, 12:1315–1322, March 2019. ISSN 15487105. doi: 10.1101/589333. URL `https://www.biorxiv.org/content/10.1101/589333v1`. Publisher: Cold Spring Harbor Laboratory.

[26] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research*, 25(17): 3389–3402, 1997. Publisher: Oxford University Press.

[27] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature Methods*, 9(2):173–175, February 2012. ISSN 1548-7091. doi: 10.1038/nmeth.1818. URL `http://www.nature.com/articles/nmeth.1818`. Publisher: Nature Publishing Group.

[28] J. Soding, A. Biegert, and A. N. Lupas. The HHpred interactive server for protein homology detection and structure prediction. *Nucleic Acids Research*, 33(Web Server):W244–W248, July 2005. ISSN 0305-1048. doi: 10.1093/nar/gki408. URL `https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gki408`. Publisher: Narnia.

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

[30] IUPAC-IUB. Recommendations on Nomenclature and Symbolism for Amino Acids and Peptides. *Pure Appl. Chem*, 56:595–623, 1984.

[31] Thomas E Creighton. *Proteins: structures and molecular properties*. Macmillan, 1993.

[32] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein Engineering, Design and Selection*, 12(2):85–94, February 1999. ISSN 1741-0134. doi: 10.1093/protein/12.2.85. URL `https://academic.oup.com/peds/article-lookup/doi/10.1093/protein/12.2.85`. Publisher: Narnia.

[33] Steven E Brenner, Cyrus Chothia, and Tim JP Hubbard. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. *Proceedings of the National Academy of Sciences*, 95(11):6073–6078, 1998. Publisher: National Acad Sciences.

[34] Stephen F. Altschul, Warren Gish, Webb Miller, Eugene W. Myers, and David J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. ISSN 00222836. doi: 10.1016/S0022-2836(05)80360-2. URL `https://www.sciencedirect.com/science/article/pii/S0022283605803602?via{%}3Dihub`.

[35] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[36] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien. Semi-supervised learning. *IEEE Transactions on Neural Networks*, 20(3):542–542, 2009. Publisher: IEEE.

[37] John Moult, Krzysztof Fidelis, Andriy Kryshtafovych, Torsten Schwede, and Anna Tramontano. Critical assessment of methods of protein structure prediction (CASP)-Round XII. *Proteins: Structure, Function, and Bioinformatics*, 86:7–15, March 2018. ISSN 08873585. doi: 10.1002/prot.25415. URL `http://doi.wiley.com/10.1002/prot.25415`. Publisher: John Wiley & Sons, Ltd.

[38] Mohammed AlQuraishi. ProteinNet: a standardized data set for machine learning of protein structure. *bioRxiv*, February 2019. URL `http://arxiv.org/abs/1902.00249`.

[39] Jason Weston, Dengyong Zhou, André Elisseeff, William S Noble, and Christina S Leslie. Semi-supervised protein classification using cluster kernels. In *Advances in neural information processing systems*, pages 595–602, 2004.

[40] Hyunjung Shin, Koji Tsuda, B Schölkopf, A Zien, and others. Prediction of protein function from networks. In *Semi-supervised learning*, pages 361–376. MIT press, 2006.

[41] Ehsaneddin Asgari and Mohammad R. K. Mofrad. Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLOS ONE*, 10(11):e0141287, November 2015. ISSN 1932-6203. doi: 10.1371/journal.pone.0141287. URL `https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0141287`. Publisher: Public Library of Science.

[42] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling the language of life – Deep Learning Protein Sequences. *bioRxiv*, page 614313, 2019. doi: 10.1101/614313. URL `https://www.biorxiv.org/content/10.1101/614313v3`. Publisher: Cold Spring Harbor Laboratory.

[43] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), April 2021. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/content/118/15/e2016239118`. Publisher: National Academy of Sciences Section: Biological Sciences.

[44] Kevin K. Yang, Zachary Wu, Claire N. Bedbrook, and Frances H. Arnold. Learned protein embeddings for machine learning. *Bioinformatics*, 34(15):2642–2648, August 2018. ISSN 14602059. doi: 10.1093/bioinformatics/bty178. Publisher: Oxford University Press.

[45] Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems. *arXiv preprint arXiv:1905.00537*, 2019.

[46] Robert D. Finn, Alex Bateman, Jody Clements, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, Andreas Heger, Kirstie Hetherington, Liisa Holm, Jaina Mistry, Erik L.L. Sonnhammer, John Tate, and Marco Punta. Pfam: The protein families database, January 2014. URL `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3965110/`. ISSN: 03051048 Issue: D1 Pages: D222 Publication Title: Nucleic Acids Research Volume: 42.

[47] 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, and Paolo Marcatili. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 87(6):520–527, June 2019. ISSN 0887-3585. doi: 10.1002/prot.25674. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25674`.

[48] Alexey Drozdetskiy, Christian Cole, James Procter, and Geoffrey J. Barton. JPred4: a protein secondary structure prediction server. *Nucleic Acids Research*, 43(W1):W389–W394, July 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv332. URL `https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkv332`. Publisher: Narnia.

[49] James A. Cuff and Geoffrey J. Barton. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. *Proteins: Structure, Function and Genetics*, 34(4):508–519, March 1999. ISSN 08873585. doi: 10.1002/(SICI)1097-0134(19990301)34:4<508::AID-PROT10>3.0.CO;2-4. URL `https://pubmed.ncbi.nlm.nih.gov/10081963/`. Publisher: Proteins.

[50] David E. Kim, Frank DiMaio, Ray Yu-Ruei Wang, Yifan Song, David Baker, Ray Yu-Ruei Wang, Yifan Song, and David Baker. One contact for every twelve residues allowsrobust and accurate topology-level proteinstructure modeling. *Proteins: Structure, Function, and Bioinformatics*, 82:208–218, February 2014. URL `http://doi.wiley.com/10.1002/prot.24374http://www.ncbi.nlm.nih.gov/pubmed/23900763http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4128384`.

[51] Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303, April 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/btx780. URL `https://academic.oup.com/bioinformatics/article/34/8/1295/4708302`. Publisher: Oxford University Press.

[52] Leticia Stephan Tavares, Carolina dos Santos Fernandes da Silva, Vinicius Carius Souza, Vânia Lúcia da Silva, Cláudio Galuppo Diniz, and Marcelo De Oliveira Santos. Strategies and molecular tools to fight antimicrobial resistance: resistome, transcriptome, and antimicrobial peptides. *Frontiers in microbiology*, 4:412, 2013. Publisher: Frontiers.

[53] Jun-Jie Liu, Natalia Orlova, Benjamin L Oakes, Enbo Ma, Hannah B Spinner, Katherine LM Baney, Jonathan Chuck, Dan Tan, Gavin J Knott, Lucas B Harrington, and others. CasX enzymes comprise a distinct family of RNA-guided genome editors. *Nature*, 566(7743):218, 2019. Publisher: Nature Publishing Group.

[54] 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, Natalya S. Bogatyreva, Peter K. Vlasov, Evgeny S. Egorov, Maria D. Logacheva, Alexey S. Kondrashov, Dmitry M. Chudakov, Ekaterina V. Putintseva, Ilgar Z. Mamedov, Dan S. Tawfik, Konstantin A. Lukyanov, and Fyodor A. Kondrashov. Local fitness landscape of the green fluorescent protein. *Nature*, 533(7603):397–401, May 2016. ISSN 14764687. doi: 10.1038/nature17995. URL `https://www.nature.com/articles/nature17995`. Publisher: Nature Publishing Group.

[55] Kevin K Yang, Zachary Wu, and Frances H Arnold. Machine learning in protein engineering. *arXiv preprint arXiv:1811.10775*, 2018.

[56] Gabriel J. Rocklin, Tamuka M. Chidyausiku, Inna Goreshnik, Alex Ford, Scott Houliston, Alexander Lemak, Lauren Carter, Rashmi Ravichandran, Vikram K. Mulligan, Aaron

Chevalier, Cheryl H. Arrowsmith, and David Baker. Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, 357(6347):168–175, July 2017. ISSN 10959203. doi: 10.1126/science.aan0693. URL `http://science.sciencemag.org/`. Publisher: American Association for the Advancement of Science.

[57] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Eleventh annual conference of the international speech communication association*, 2010.

[58] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Journal of Machine Learning Research*, October 2013. URL `http://arxiv.org/abs/1310.1531`.

[59] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. Publisher: MIT Press.

[60] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. Dilated Residual Networks. *Computer Vision and Pattern Recognition*, pages 472–480, 2017. URL `http://openaccess.thecvf.com/content_cvpr_2017/html/Yu_Dilated_Residual_Networks_CVPR_2017_paper.html`.

[61] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. Multiplicative LSTM for sequence modelling. *arXiv preprint arXiv:1609.07959*, 2016.

[62] Joerg Schaarschmidt, Bohdan Monastyrskyy, Andriy Kryshtafovych, and Alexandre MJJ Bonvin. Assessment of contact predictions in CASP12: Co-evolution and deep learning coming of age. *Proteins: Structure, Function, and Bioinformatics*, 86:51–66, 2018. Publisher: Wiley Online Library.

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

[64] William P. Russ, Matteo Figliuzzi, Christian Stocker, Pierre Barrat-Charlaix, Michael Socolich, Peter Kast, Donald Hilvert, Remi Monasson, Simona Cocco, Martin Weigt, and Rama Ranganathan. An evolution-based model for designing chorismate mutase enzymes. *Science*, 369(6502):440–445, July 2020. ISSN 10959203. doi: 10.1126/science.aba3304. Publisher: American Association for the Advancement of Science.

[65] Pengfei Tian, John M. Louis, James L. Baber, Annie Aniana, and Robert B. Best. Co-Evolutionary Fitness Landscapes for Sequence Design. *Angewandte Chemie - International Edition*, 57(20):5674–5678, May 2018. ISSN 15213773. doi: 10.1002/anie.201713220. Publisher: Wiley-VCH Verlag.

[66] Tomasz Blazejewski, Hsing I. Ho, and Harris H. Wang. Synthetic sequence entanglement augments stability and containment of genetic information in cells. *Science*, 365(6453): 595–598, August 2019. ISSN 10959203. doi: 10.1126/science.aav5477. Publisher: American Association for the Advancement of Science.

[67] Sheng Wang, Siqi Sun, Zhen Li, Renyu Zhang, and Jinbo Xu. Accurate De Novo Prediction of Protein Contact Map by Ultra-Deep Learning Model. *PLOS Computational Biology*, 13(1):1–34, January 2017. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1005324. URL `https://dx.plos.org/10.1371/journal.pcbi.1005324`. Publisher: Public Library of Science.

[68] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W.R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Improved protein structure prediction using potentials from deep learning. *Nature*, 577(7792): 706–710, January 2020. ISSN 14764687. doi: 10.1038/s41586-019-1923-7. URL `https://doi.org/10.1038/s41586-019-1923-7`. Publisher: Nature Research.

[69] Jianyi Yang, Ivan Anishchenko, Hahnbeom Park, Zhenling Peng, Sergey Ovchinnikov, and David Baker. Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences*, 117(3):1496–1503, January 2020. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.1914677117. URL `https://www.pnas.org/content/117/3/1496`. Publisher: National Academy of Sciences Section: Biological Sciences.

[70] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R. Eguchi, Po-Ssu Huang, and Richard Socher. ProGen: Language Modeling for Protein Generation. *bioRxiv*, March 2020. URL `http://arxiv.org/abs/2004.03497`.

[71] Baris E. Suzek, Hongzhan Huang, Peter McGarvey, Raja Mazumder, and Cathy H. Wu. UniRef: Comprehensive and non-redundant UniProt reference clusters. *Bioinformatics*, 23(10):1282–1288, May 2007. ISSN 13674803. doi: 10.1093/bioinformatics/btm098. URL `http://www.uniprot.org`. Publisher: Oxford Academic.

[72] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence- and structure-rich era. *Proceedings of the National Academy of Sciences of the United States of America*, 110(39):15674–15679, September 2013. ISSN 00278424. doi: 10.1073/pnas.1314045110. URL `https://www.pnas.org/content/110/39/15674`. Publisher: National Academy of Sciences.

[73] Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G. Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families.

Proteins: Structure, Function, and Bioinformatics, 79(4):1061–1078, April 2011. ISSN 08873585. doi: 10.1002/prot.22934. URL http://doi.wiley.com/10.1002/prot.22934. Publisher: John Wiley & Sons, Ltd.

[74] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics*, 87(1):012707, January 2013. ISSN 15393755. doi: 10.1103/PhysRevE.87.012707. URL https://link.aps.org/doi/10.1103/PhysRevE.87.012707. Publisher: Phys Rev E Stat Nonlin Soft Matter Phys.

[75] Todd J. Taylor, Hongjun Bai, Chin Hsien Tai, and Byungkook Lee. Assessment of CASP10 contact-assisted predictions. *Proteins: Structure, Function and Bioinformatics*, 82(SUPPL.2):84–97, February 2014. ISSN 08873585. doi: 10.1002/prot.24367. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6961783/. Publisher: NIH Public Access.

[76] Badri Adhikari and Jianlin Cheng. Protein residue contacts and prediction methods. In *Methods in Molecular Biology*, volume 1415, pages 463–476. Humana Press Inc., August 2016. doi: 10.1007/978-1-4939-3572-7_24. URL https://pubmed.ncbi.nlm.nih.gov/27115648/. ISSN: 10643745.

[77] David T. Jones and Shaun M. Kandathil. High precision in protein contact prediction using fully convolutional neural networks and minimal sequence features. *Bioinformatics*, 34(19):3308–3315, October 2018. ISSN 14602059. doi: 10.1093/bioinformatics/bty341. URL https://academic.oup.com/bioinformatics/article/34/19/3308/4987145. Publisher: Oxford University Press.

[78] Badri Adhikari and Arne Elofsson. DEEPCON: Protein contact prediction using dilated convolutional neural networks with dropout. *Bioinformatics*, 36(2):470–477, January 2020. ISSN 14602059. doi: 10.1093/bioinformatics/btz593. URL https://academic.oup.com/bioinformatics/article/36/2/470/5540673. Publisher: Oxford University Press.

[79] Yang Liu, Perry Palmedo, Qing Ye, Bonnie Berger, and Jian Peng. Enhancing Evolutionary Couplings with Deep Convolutional Neural Networks. *Cell Systems*, 6(1):65–74, January 2018. ISSN 24054720. doi: 10.1016/j.cels.2017.11.014. URL https://pubmed.ncbi.nlm.nih.gov/29275173/. Publisher: Cell Press.

[80] Mohammed AlQuraishi. End-to-End Differentiable Learning of Protein Structure. *Cell Systems*, 8(4):292–301.e3, April 2019. ISSN 2405-4712. doi: 10.1016/j.cels.2019.03.006. URL https://www.sciencedirect.com/science/article/pii/S2405471219300766.

[81] John Ingraham, Adam Riesselman, Chris Sander, and Debora Marks. Learning protein structure with a differentiable simulator. *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[82] Jinbo Xu, Matthew McPartlon, and Jin Li. Improved protein structure prediction by deep learning irrespective of co-evolution information. *Nature Machine Intelligence*, 3 (7):601–609, July 2021. ISSN 2522-5839. doi: 10.1038/s42256-021-00348-5. URL `https://www.nature.com/articles/s42256-021-00348-5`. Bandiera_abtest: a Cg_type: Nature Research Journals Number: 7 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computer science;Machine learning;Protein structure predictions Subject_term_id: computer-science;machine-learning;protein-structure-predictions.

[83] Alan S. Lapedes, Bertrand G. Giraud, LonChang Liu, and Gary D. Stormo. Correlated Mutations in Models of Protein Sequences: Phylogenetic and Structural Effects. *Lecture Notes-Monograph Series*, 33:236–256, 1999. ISSN 07492170. doi: 10.2307/4356049. URL `http://www.jstor.org/stable/4356049`. Publisher: Institute of Mathematical Statistics.

[84] John Thomas, Naren Ramakrishnan, and Chris Bailey-Kellogg. Graphical models of residue coupling in protein families, April 2008. URL `https://pubmed.ncbi.nlm.nih.gov/18451428/`. ISSN: 15455963 Issue: 2 Pages: 183–197 Publication Title: IEEE/ACM Transactions on Computational Biology and Bioinformatics Volume: 5.

[85] Martin Weigt, Robert A. White, Hendrik Szurmant, James A. Hoch, and Terence Hwa. Identification of direct residue contacts in protein-protein interaction by message passing. *Proceedings of the National Academy of Sciences of the United States of America*, 106(1):67–72, January 2009. ISSN 00278424. doi: 10.1073/pnas.0805923106. URL `https://www.pnas.org/content/106/1/67https://www.pnas.org/content/106/1/67.abstract`. Publisher: National Academy of Sciences.

[86] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S. Marks, Chris Sander, Riccardo Zecchina, José N. Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences of the United States of America*, 108(49):E1293–E1301, December 2011. ISSN 00278424. doi: 10.1073/pnas.1111471108. URL `https://www.pnas.org/content/108/49/E1293`. Publisher: National Academy of Sciences.

[87] David T. Jones, Daniel W. A. Buchan, Domenico Cozzetto, and Massimiliano Pontil. PSICOV: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28 (2):184–190, January 2012. ISSN 1460-2059. doi: 10.1093/bioinformatics/

btr638. URL https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btr638. Publisher: Oxford Academic.

[88] Stefan Seemayer, Markus Gruber, and Johannes Söding. CCMpred - Fast and precise prediction of protein residue-residue contacts from correlated mutations. *Bioinformatics*, 30(21):3128–3130, May 2014. ISSN 14602059. doi: 10.1093/bioinformatics/btu500. URL https://pubmed.ncbi.nlm.nih.gov/25064567/. Publisher: Oxford University Press.

[89] Debora S. Marks, Lucy J. Colwell, Robert Sheridan, Thomas A. Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D Structure Computed from Evolutionary Sequence Variation. *PLOS ONE*, 6(12):1–20, December 2011. doi: 10.1371/journal.pone.0028766. URL https://doi.org/10.1371/journal.pone.0028766. Publisher: Public Library of Science.

[90] J. I. Sulkowska, F. Morcos, M. Weigt, T. Hwa, and J. N. Onuchic. Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, 109(26):10340–10345, 2012. doi: 10.1073/PNAS.1207864109. URL http://www.pnas.org/cgi/doi/10.1073/pnas.1207864109.

[91] Tomasz Kosciolek and David T. Jones. De Novo Structure Prediction of Globular Proteins Aided by Sequence Variation-Derived Contacts. *PLOS ONE*, 9(3):1–15, March 2014. doi: 10.1371/journal.pone.0092197. URL https://doi.org/10.1371/journal.pone.0092197. Publisher: Public Library of Science.

[92] Sergey Ovchinnikov, David E. Kim, Ray Yu-Ruei Wang, Yuan Liu, Frank DiMaio, and David Baker. Improved de novo structure prediction in CASP11 by incorporating coevolution information into Rosetta. *Proteins: Structure, Function, and Bioinformatics*, 84(S1):67–75, 2016. doi: 10.1002/prot.24974. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.24974. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1002/prot.24974.

[93] Andrew W. Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Žídek, Alexander W. R. Nelson, Alex Bridgland, Hugo Penedones, Stig Petersen, Karen Simonyan, Steve Crossan, Pushmeet Kohli, David T. Jones, David Silver, Koray Kavukcuoglu, and Demis Hassabis. Protein structure prediction using multiple deep neural networks in the 13th Critical Assessment of Protein Structure Prediction (CASP13). *Proteins: Structure, Function, and Bioinformatics*, 87(12):1141–1148, December 2019. ISSN 0887-3585. doi: 10.1002/prot.25834. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25834.

[94] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, Debsindhu Bhowmik, and Burkhard Rost. ProtTrans: Towards Cracking the Language of Life's

Code Through Self-Supervised Deep Learning and High Performance Computing. *bioRxiv*, July 2020. URL http://arxiv.org/abs/2007.06225.

[95] Amy X Lu, Haoran Zhang, Marzyeh Ghassemi, and Alan Moses. Self-Supervised Contrastive Learning of Protein Representations By Mutual Information Maximization. *bioRxiv*, page 2020.09.04.283929, September 2020. doi: 10.1101/2020.09.04.283929. URL https://doi.org/10.1101/2020.09.04.283929. Publisher: Cold Spring Harbor Laboratory.

[96] Alex Wang and Kyunghyun Cho. BERT has a Mouth, and It Must Speak: BERT as a Markov Random Field Language Model. February 2019. URL http://arxiv.org/abs/1902.04094.

[97] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa: A Robustly Optimized BERT Pretraining Approach. July 2019. URL http://arxiv.org/abs/1907.11692.

[98] Ivan Anishchenko, Sergey Ovchinnikov, Hetunandan Kamisetty, and David Baker. Origins of coevolution between residues distant in protein 3D structures. *Proceedings of the National Academy of Sciences of the United States of America*, 114(34):9122–9127, August 2017. ISSN 10916490. doi: 10.1073/pnas.1702664114. URL https://www.pnas.org/content/early/2017/08/03/1702664114https://www.pnas.org/content/early/2017/08/03/1702664114.abstract. Publisher: National Academy of Sciences.

[99] Fan Zheng and Gevorg Grigoryan. Sequence statistics of tertiary structural motifs reflect protein stability. *PLoS ONE*, 12(5):e0178272, May 2017. ISSN 19326203. doi: 10.1371/journal.pone.0178272. URL https://doi.org/10.1371/journal.pone.0178272. Publisher: Public Library of Science.

[100] Alex Hawkins-Hooker, Florence Depardieu, Sebastien Baur, Guillaume Couairon, Arthur Chen, and David Bikard. Generating functional protein variants with variational autoencoders. *PLOS Computational Biology*, 17(2):1–23, February 2021. doi: 10.1371/journal.pcbi.1008736. URL https://doi.org/10.1371/journal.pcbi.1008736. Publisher: Public Library of Science.

[101] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[102] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark

Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language Models are Few-Shot Learners. *CoRR*, abs/2005.14165, 2020. URL `https://arxiv.org/abs/2005.14165`. _eprint: 2005.14165.

[103] Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317, 1994. ISSN 10970134. doi: 10.1002/prot.340180402. URL `https://pubmed.ncbi.nlm.nih.gov/8208723/`. Publisher: Proteins.

[104] Jonathan Ho, Nal Kalchbrenner, Dirk Weissenborn, and Tim Salimans. Axial Attention in Multidimensional Transformers. *arXiv*, December 2019. URL `http://arxiv.org/abs/1912.12180`. Publisher: arXiv.

[105] Zilong Huang, Xinggang Wang, Lichao Huang, Chang Huang, Yunchao Wei, and Wenyu Liu. CCNet: Criss-Cross Attention for Semantic Segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 603–612, 2019. URL `https://github.com/speedinghzl/`.

[106] Nicholas Bhattacharya, Neil Thomas, Roshan Rao, Justas Dauparas, Peter K. Koo, David Baker, Yun S. Song, and Sergey Ovchinnikov. Single Layers of Attention Suffice to Predict Protein Contacts. *bioRxiv*, page 2020.12.21.423882, December 2020. doi: 10.1101/2020.12.21.423882. Publisher: Cold Spring Harbor Laboratory.

[107] Adam Riesselman, Jung-Eun Shin, Aaron Kollasch, Conor McMahon, Elana Simon, Chris Sander, Aashish Manglik, Andrew Kruse, and Debora Marks. Accelerating Protein Design Using Autoregressive Generative Models. *bioRxiv*, page 757252, 2019. doi: 10.1101/757252. URL `https://www.biorxiv.org/content/10.1101/757252v1`. Publisher: Cold Spring Harbor Laboratory.

[108] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596 (7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL `https://www.nature.com/articles/s41586-021-03819-2`. Bandiera_abtest: a Cc_license_type: cc_by Cg_type: Nature Research Journals Number: 7873 Primary_atype: Research Publisher: Nature Publishing Group Subject_term: Computational biophysics;Machine learning;Protein structure predictions;Structural biol-

ogy Subject_term_id: computational-biophysics;machine-learning;protein-structure-predictions;structural-biology.

[109] Claudio Mirabello and Björn Wallner. rawMSA: End-to-end Deep Learning using raw Multiple Sequence Alignments. *PLOS ONE*, 14(8):e0220182, August 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0220182. URL `https://dx.plos.org/10.1371/journal.pone.0220182`. Publisher: Public Library of Science.

[110] Shaun M Kandathil, Joe G Greener, Andy M Lau, and David T Jones. Deep learning-based prediction of protein structure using learned representations of multiple sequence alignments. *bioRxiv*, page 2020.11.27.401232, November 2020. doi: 10.1101/2020.11.27.401232. Publisher: Cold Spring Harbor Laboratory.

[111] Pascal Sturmfels, Jesse Vig, Ali Madani, and Nazneen Fatema Rajani. Profile Prediction: An Alignment-Based Pre-Training Task for Protein Sequence Models. *bioRxiv*, November 2020. URL `http://arxiv.org/abs/2012.00195`.

[112] Tom Sercu, Robert Verkuil, Joshua Meier, Brandon Amos, Zeming Lin, Caroline Chen, Jason Liu, Yann LeCun, and Alexander Rives. Neural Potts Models. *MLCB*, pages 1–13, 2020. URL `https://openreview.net/forum?id=U6Xpa5R-E1`.

[113] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and others. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*, 2020.

[114] Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. Issue: 1.

[115] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating Long Sequences with Sparse Transformers. *CoRR*, abs/1904.10509, 2019. URL `http://arxiv.org/abs/1904.10509`. _eprint: 1904.10509.

[116] Milot Mirdita, Lars Von Den Driesch, Clovis Galiez, Maria J. Martin, Johannes Soding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Research*, 45(D1):D170–D176, January 2017. ISSN 13624962. doi: 10.1093/nar/gkw1081. Publisher: Oxford University Press.

[117] Martin Steinegger, Markus Meier, Milot Mirdita, Harald Vöhringer, Stephan J. Haunsberger, and Johannes Söding. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics*, 20(1):473, September 2019. ISSN 14712105. doi: 10.1186/s12859-019-3019-7. URL `https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-019-3019-7`. Publisher: BioMed Central Ltd.

[118] S. D. Dunn, L. M. Wahl, and G. B. Gloor. Mutual information without the influence of phylogeny or entropy dramatically improves residue contact prediction. *Bioinformatics*, 24(3):333–340, February 2008. ISSN 13674803. doi: 10.1093/bioinformatics/btm604. Publisher: Oxford Academic.

[119] Jürgen Haas, Alessandro Barbato, Dario Behringer, Gabriel Studer, Steven Roth, Martino Bertoni, Khaled Mostaguir, Rafal Gumienny, and Torsten Schwede. Continuous Automated Model EvaluatiOn (CAMEO) complementing the critical assessment of structure prediction in CASP12. *Proteins: Structure, Function and Bioinformatics*, 86 (Suppl 1):387–398, March 2018. ISSN 10970134. doi: 10.1002/prot.25431. Publisher: John Wiley and Sons Inc.

[120] Rojan Shrestha, Eduardo Fajardo, Nelson Gil, Krzysztof Fidelis, Andriy Kryshtafovych, Bohdan Monastyrskyy, and Andras Fiser. Assessing the accuracy of contact predictions in CASP13. *Proteins: Structure, Function, and Bioinformatics*, 87(12): 1058–1068, December 2019. ISSN 0887-3585. doi: 10.1002/prot.25819. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.25819`. Publisher: John Wiley and Sons Inc.

[121] Laksshman Sundaram, Hong Gao, Samskruthi Reddy Padigepati, Jeremy F McRae, Yanjun Li, Jack A Kosmicki, Nondas Fritzilas, Jörg Hakenberg, Anindita Dutta, John Shon, and others. Predicting the clinical impact of human mutation with deep neural networks. *Nature Genetics*, 50(8):1161–1170, 2018. ISSN 15461718. doi: 10.1038/s41588-018-0167-z. Publisher: Nature Publishing Group.

[122] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Kelly Brock, Yarin Gal, and Debora S. Marks. Large-scale clinical interpretation of genetic variants using evolutionary data and deep learning. *bioRxiv*, page 2020.12.21.423785, December 2020. doi: 10.1101/2020.12.21.423785. Publisher: Cold Spring Harbor Laboratory.

[123] Charles J Epstein, Robert F Goldberger, and Christian B Anfinsen. The genetic control of tertiary protein structure: studies with model systems. In *Cold Spring Harbor symposia on quantitative biology*, volume 28, pages 439–449. Cold Spring Harbor Laboratory Press, 1963.

[124] Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature Methods*, 11(8):801–807, 2014. ISSN 1548-7091. doi: 10.1038/nmeth.3027. URL `http://www.nature.com/articles/nmeth.3027`. Publisher: Nature Publishing Group.

[125] Matteo Figliuzzi, Hervé Jacquier, Alexander Schug, Oliver Tenaillon, and Martin Weigt. Coevolutionary landscape inference and the context-dependence of mutations in beta-lactamase TEM-1. *Molecular biology and evolution*, 33(1):268–280, 2016. Publisher: Oxford University Press.

[126] D. Altschuh, A. M. Lesk, A. C. Bloomer, and A. Klug. Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of Molecular Biology*, 193(4):693–707, 1987. ISSN 0022-2836. doi: 10.1016/0022-2836(87)90352-4.

[127] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning Transferable Visual Models From Natural Language Supervision, 2021. _eprint: 2103.00020.

[128] Chloe Hsu, Hunter Nisonoff, Clara Fannjiang, and Jennifer Listgarten. Combining evolutionary and assay-labelled data for protein fitness prediction. *bioRxiv*, page 2021.03.28.437402, March 2021. doi: 10.1101/2021.03.28.437402. Publisher: Cold Spring Harbor Laboratory.

[129] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. Low-N protein engineering with data-efficient deep learning. *bioRxiv*, page 2020.01.23.917682, 2020. doi: 10.1101/2020.01.23.917682. Publisher: Cold Spring Harbor Laboratory.

[130] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958. IEEE, 2009.

[131] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. Zero-data learning of new tasks. In *AAAI*, volume 1, page 3, 2008. Issue: 2.

[132] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-Shot Text-to-Image Generation. In *Proceedings of the 38th International Conference on Machine Learning*, pages 8821–8831. PMLR, July 2021. URL `https://proceedings.mlr.press/v139/ramesh21a.html`. ISSN: 2640-3498.

[133] Kevin K. Yang, Zachary Wu, and Frances H. Arnold. Machine-learning-guided directed evolution for protein engineering, August 2019. ISSN: 15487105 Issue: 8 Pages: 687–694 Publication Title: Nature Methods Volume: 16 _eprint: 1811.10775.

[134] Vanessa E. Gray, Ronald J. Hause, Jens Luebeck, Jay Shendure, and Douglas M. Fowler. Quantitative Missense Variant Effect Prediction Using Large-Scale Mutagenesis Data. *Cell Systems*, 6(1):116–124.e3, January 2018. ISSN 24054720. doi: 10.1016/j.cels.2017.11.003. Publisher: Cell Press.

[135] Ivan A. Adzhubei, Steffen Schmidt, Leonid Peshkin, Vasily E. Ramensky, Anna Gerasimova, Peer Bork, Alexey S. Kondrashov, and Shamil R. Sunyaev. A method and server for predicting damaging missense mutations, April 2010. ISSN: 15487091 Issue: 4 Pages: 248–249 Publication Title: Nature Methods Volume: 7.

[136] Nilah M Ioannidis, Joseph H Rothstein, Vikas Pejaver, Sumit Middha, Shannon K McDonnell, Saurabh Baheti, Anthony Musolf, Qing Li, Emily Holzinger, Danielle Karyadi, and others. REVEL: an ensemble method for predicting the pathogenicity of rare missense variants. *The American Journal of Human Genetics*, 99(4):877–885, 2016. Publisher: Elsevier.

[137] Kaitlin E Samocha, Jack A Kosmicki, Konrad J Karczewski, Anne H O'Donnell-Luria, Emma Pierce-Hoffman, Daniel G MacArthur, Benjamin M Neale, and Mark J Daly. Regional missense constraint improves variant deleteriousness prediction. *BioRxiv*, page 148353, 2017. Publisher: Cold Spring Harbor Laboratory.

[138] Martin Kircher, Daniela M Witten, Preti Jain, Brian J O'Roak, Gregory M Cooper, and Jay Shendure. A general framework for estimating the relative pathogenicity of human genetic variants. *Nature genetics*, 46(3):310–315, 2014. Publisher: Nature Publishing Group.

[139] Karthik A Jagadeesh, Aaron M Wenger, Mark J Berger, Harendra Guturu, Peter D Stenson, David N Cooper, Jonathan A Bernstein, and Gill Bejerano. M-CAP eliminates a majority of variants of uncertain significance in clinical exomes at high sensitivity. *Nature genetics*, 48(12):1581, 2016. Publisher: Nature Publishing Group.

[140] Haicang Zhang, Michelle S Xu, Wendy K Chung, and Yufeng Shen. Predicting functional effect of missense variants using graph attention neural networks. *bioRxiv*, 2021. Publisher: Cold Spring Harbor Laboratory.

[141] Ngak Leng Sim, Prateek Kumar, Jing Hu, Steven Henikoff, Georg Schneider, and Pauline C. Ng. SIFT web server: Predicting effects of amino acid substitutions on proteins. *Nucleic Acids Research*, 40(W1):W452, July 2012. ISSN 03051048. doi: 10.1093/nar/gks539. Publisher: Oxford University Press.

[142] Brian L Hie, Kevin K Yang, and Peter S Kim. Evolutionary velocity with protein language models. *bioRxiv*, 2021. Publisher: Cold Spring Harbor Laboratory.

[143] Henrik Nordberg, Michael Cantor, Serge Dusheyko, Susan Hua, Alexander Poliakov, Igor Shabalov, Tatyana Smirnova, Igor V Grigoriev, and Inna Dubchak. The genome portal of the Department of Energy Joint Genome Institute: 2014 updates. *Nucleic acids research*, 42(D1):D26–D31, 2014. Publisher: Oxford University Press.

[144] Martin Steinegger, Milot Mirdita, and Johannes Söding. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nature Methods*, 16 (7):603–606, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0437-4. URL `https://doi.org/10.1038/s41592-019-0437-4`.

[145] Emily E Wrenbeck, Matthew S Faber, and Timothy A Whitehead. Deep sequencing methods for protein engineering and design. *Current opinion in structural biology*, 45: 36–44, 2017. Publisher: Elsevier.

[146] Justin R Klesmith, John-Paul Bacik, Ryszard Michalczyk, and Timothy A Whitehead. Comprehensive sequence-flux mapping of a levoglucosan utilization pathway in E. coli. *ACS synthetic biology*, 4(11):1235–1243, 2015. Publisher: ACS Publications.

[147] Hugh K Haddox, Adam S Dingens, Sarah K Hilton, Julie Overbaugh, and Jesse D Bloom. Mapping mutational effects along the evolutionary landscape of HIV envelope. *Elife*, 7:e34420, 2018. Publisher: eLife Sciences Publications Limited.

[148] Philip A Romero, Tuan M Tran, and Adam R Abate. Dissecting enzyme function with microfluidic-based deep mutational scanning. *Proceedings of the National Academy of Sciences*, 112(23):7159–7164, 2015. Publisher: National Acad Sciences.

[149] Elad Firnberg, Jason W Labonte, Jeffrey J Gray, and Marc Ostermeier. A comprehensive, high-resolution map of a gene's fitness landscape. *Molecular biology and evolution*, 31 (6):1581–1592, 2014. Publisher: Oxford University Press.

[150] Zhifeng Deng, Wanzhi Huang, Erol Bakkalbasi, Nicholas G Brown, Carolyn J Adamski, Kacie Rice, Donna Muzny, Richard A Gibbs, and Timothy Palzkill. Deep sequencing of systematic combinatorial libraries reveals β-lactamase sequence constraints at high resolution. *Journal of molecular biology*, 424(3-4):150–167, 2012. Publisher: Elsevier.

[151] Michael A Stiffler, Doeke R Hekstra, and Rama Ranganathan. Evolvability as a function of purifying selection in TEM-1 β-lactamase. *Cell*, 160(5):882–892, 2015. Publisher: Elsevier.

[152] Hervé Jacquier, André Birgy, Hervé Le Nagard, Yves Mechulam, Emmanuelle Schmitt, Jérémy Glodt, Beatrice Bercot, Emmanuelle Petit, Julie Poulain, Guilène Barnaud, and others. Capturing the mutational landscape of the beta-lactamase TEM-1. *Proceedings of the National Academy of Sciences*, 110(32):13067–13072, 2013. Publisher: National Acad Sciences.

[153] Scott D Findlay and Lynne-Marie Postovit. Comprehensive characterization of transcript diversity at the human NODAL locus. *BioRxiv*, page 254409, 2018. Publisher: Cold Spring Harbor Laboratory.

[154] Richard N McLaughlin Jr, Frank J Poelwijk, Arjun Raman, Walraj S Gosal, and Rama Ranganathan. The spatial architecture of protein function and adaptation. *Nature*, 491 (7422):138–142, 2012. Publisher: Nature Publishing Group.

[155] Jacob O Kitzman, Lea M Starita, Russell S Lo, Stanley Fields, and Jay Shendure. Massively parallel single-amino-acid mutagenesis. *Nature methods*, 12(3):203–206, 2015. Publisher: Nature Publishing Group.

[156] Michael B Doud and Jesse D Bloom. Accurate measurement of the effects of all amino-acid mutations on influenza hemagglutinin. *Viruses*, 8(6):155, 2016. Publisher: Multidisciplinary Digital Publishing Institute.

[157] K Pokusaeva, C Johnson, B Luk, G7 Uribe, Y Fu, N Oezguen, RK Matsunami, M Lugo, A Major, Y Mori-Akiyama, and others. GABA-producing Bifidobacterium dentium modulates visceral sensitivity in the intestine. *Neurogastroenterology & Motility*, 29(1): e12904, 2017. Publisher: Wiley Online Library.

[158] Parul Mishra, Julia M Flynn, Tyler N Starr, and Daniel NA Bolon. Systematic mutant analyses elucidate general and client-specific aspects of Hsp90 function. *Cell reports*, 15 (3):588–598, 2016. Publisher: Elsevier.

[159] Eric D Kelsic, Hattie Chung, Niv Cohen, Jimin Park, Harris H Wang, and Roy Kishony. RNA structural determinants of optimal codons revealed by MAGE-Seq. *Cell systems*, 3(6):563–571, 2016. Publisher: Elsevier.

[160] Alexandre Melnikov, Peter Rogov, Li Wang, Andreas Gnirke, and Tarjei S Mikkelsen. Comprehensive mutational scanning of a kinase in vivo reveals substrate-dependent fitness landscapes. *Nucleic acids research*, 42(14):e112–e112, 2014. Publisher: Oxford University Press.

[161] Lisa Brenan, Aleksandr Andreev, Ofir Cohen, Sasha Pantel, Atanas Kamburov, Davide Cacchiarelli, Nicole S Persky, Cong Zhu, Mukta Bagul, Eva M Goetz, and others. Phenotypic characterization of a comprehensive set of MAPK1/ERK2 missense mutants. *Cell reports*, 17(4):1171–1183, 2016. Publisher: Elsevier.

[162] Liat Rockah-Shmuel, Ágnes Tóth-Petróczy, and Dan S Tawfik. Systematic mapping of protein mutational space by prolonged drift reveals the deleterious effects of seemingly neutral mutations. *PLoS computational biology*, 11(8):e1004421, 2015. Publisher: Public Library of Science San Francisco, CA USA.

[163] Nicholas C Wu, C Anders Olson, Yushen Du, Shuai Le, Kevin Tran, Roland Remenyi, Danyang Gong, Laith Q Al-Mawsawi, Hangfei Qi, Ting-Ting Wu, and others. Functional constraint profiling of a viral protein reveals discordance of evolutionary conservation and functionality. *PLoS genetics*, 11(7):e1005310, 2015. Publisher: Public Library of Science San Francisco, CA USA.

[164] Christopher D Aakre, Julien Herrou, Tuyen N Phung, Barrett S Perchuk, Sean Crosson, and Michael T Laub. Evolving new protein-protein interaction specificity through promiscuous intermediates. *Cell*, 163(3):594–606, 2015. Publisher: Elsevier.

[165] Hangfei Qi, C Anders Olson, Nicholas C Wu, Ruian Ke, Claude Loverdo, Virginia Chu, Shawna Truong, Roland Remenyi, Zugen Chen, Yushen Du, and others. A quantitative high-resolution genetic profile rapidly identifies sequence determinants of hepatitis C viral fitness and drug sensitivity. *PLoS pathogens*, 10(4):e1004064, 2014. Publisher: Public Library of Science San Francisco, USA.

[166] Kenneth A Matreyek, Lea M Starita, Jason J Stephany, Beth Martin, Melissa A Chiasson, Vanessa E Gray, Martin Kircher, Arineh Khechaduri, Jennifer N Dines, Ronald J Hause, and others. Multiplex assessment of protein variant abundance by massively parallel sequencing. *Nature genetics*, 50(6):874–882, 2018. Publisher: Nature Publishing Group.

[167] Pradeep Bandaru, Neel H Shah, Moitrayee Bhattacharyya, John P Barton, Yasushi Kondo, Joshua C Cofsky, Christine L Gee, Arup K Chakraborty, Tanja Kortemme, Rama Ranganathan, and others. Deconstruction of the Ras switching cycle through saturation mutagenesis. *Elife*, 6:e27810, 2017. Publisher: eLife Sciences Publications Limited.

[168] Benjamin P Roscoe, Kelly M Thayer, Konstantin B Zeldovich, David Fushman, and Daniel NA Bolon. Analyses of the effects of all ubiquitin point mutants on yeast growth rate. *Journal of molecular biology*, 425(8):1363–1377, 2013. Publisher: Elsevier.

[169] Benjamin P Roscoe and Daniel NA Bolon. Systematic exploration of ubiquitin sequence, E1 activation efficiency, and experimental fitness in yeast. *Journal of molecular biology*, 426(15):2854–2870, 2014. Publisher: Elsevier.

[170] David Mavor, Kyle Barlow, Samuel Thompson, Benjamin A Barad, Alain R Bonny, Clinton L Cario, Garrett Gaskins, Zairan Liu, Laura Deming, Seth D Axen, and others. Determination of ubiquitin fitness landscapes under different chemical stresses in a classroom setting. *Elife*, 5:e15802, 2016. Publisher: eLife Sciences Publications Limited.

[171] Yvonne H Chan, Sergey V Venev, Konstantin B Zeldovich, and C Robert Matthews. Correlation of fitness landscapes from three orthologous TIM barrels originates from sequence and structure constraints. *Nature communications*, 8(1):1–12, 2017. Publisher: Nature Publishing Group.

[172] Daniel Melamed, David L Young, Caitlin E Gamble, Christina R Miller, and Stanley Fields. Deep mutational scanning of an RRM domain of the Saccharomyces cerevisiae poly (A)-binding protein. *Rna*, 19(11):1537–1551, 2013. Publisher: Cold Spring Harbor Lab.

[173] Lea M Starita, Jonathan N Pruneda, Russell S Lo, Douglas M Fowler, Helen J Kim, Joseph B Hiatt, Jay Shendure, Peter S Brzovic, Stanley Fields, and Rachel E Klevit. Activity-enhancing mutations in an E3 ubiquitin ligase identified by high-throughput mutagenesis. *Proceedings of the National Academy of Sciences*, 110(14):E1263–E1272, 2013. Publisher: National Acad Sciences.

[174] Carlos L Araya, Douglas M Fowler, Wentao Chen, Ike Muniez, Jeffery W Kelly, and Stanley Fields. A fundamental protein property, thermodynamic stability, revealed solely from large-scale measurements of protein function. *Proceedings of the National Academy of Sciences*, 109(42):16858–16863, 2012. Publisher: National Acad Sciences.

[175] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K. Min, Kelly Brock, Yarin Gal, and Debora S. Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, November 2021. ISSN 0028-0836, 1476-4687. doi: 10.1038/s41586-021-04043-8. URL `https://www.nature.com/articles/s41586-021-04043-8`.

[176] Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. What is a meaningful representation of protein sequences? *arXiv:2012.02679 [cs, q-bio]*, October 2021. URL `http://arxiv.org/abs/2012.02679`. arXiv: 2012.02679.

[177] Bruce J. Wittmann, Yisong Yue, and Frances H. Arnold. Informed training set design enables efficient machine learning-assisted directed protein evolution. *Cell Systems*, 12(11): 1026–1045.e7, November 2021. ISSN 2405-4712. doi: 10.1016/j.cels.2021.07.008. URL `https://www.cell.com/cell-systems/abstract/S2405-4712(21)00286-6`. Publisher: Elsevier.

[178] Yuedong Yang, Jianzhao Gao, Jihua Wang, Rhys Heffernan, Jack Hanson, Kuldip Paliwal, and Yaoqi Zhou. Sixty-five years of the long march in protein secondary structure prediction: the final stretch? *Briefings in bioinformatics*, 19(3):482–494, 2016. Publisher: Oxford University Press.

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

[180] Naomi K Fox, Steven E Brenner, and John-Marc Chandonia. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic acids research*, 42(D1):D304–D309, 2013. Publisher: Oxford University Press.

[181] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000. Publisher: Oxford University Press.

[182] Cindy J Castelle and Jillian F Banfield. Major new microbial groups expand diversity and alter our understanding of the tree of life. *Cell*, 172(6):1181–1197, 2018. Publisher: Elsevier.

[183] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training Deep Nets with Sublinear Memory Cost. *arXiv*, April 2016. URL `http://arxiv.org/abs/1604.06174`.

[184] 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, Erik L L Sonnhammer, Layla Hirsh, Lisanna Paladin, Damiano Piovesan, Silvio C E Tosatto, and Robert D Finn. The Pfam protein families database in 2019. *Nucleic Acids Research*, 47(D1):D427–D432, 2019. ISSN 0305-1048. doi: 10.1093/nar/gky995. URL `https://academic.oup.com/nar/article/47/D1/D427/5144153`. Publisher: Narnia.

[185] Robbie P Joosten, Tim AH Te Beek, Elmar Krieger, Maarten L Hekkelman, Rob WW Hooft, Reinhard Schneider, Chris Sander, and Gert Vriend. A series of PDB related databases for everyday needs. *Nucleic acids research*, 39(suppl_1):D411–D419, 2010. Publisher: Oxford University Press.

[186] Chengxin Zhang, Wei Zheng, S M Mortuza, Yang Li, and Yang Zhang. DeepMSA: constructing deep multiple sequence alignment to improve contact prediction and fold-recognition for distant-homology proteins. *Bioinformatics*, 36(7):2105–2112, April 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz863. URL `https://doi.org/10.1093/bioinformatics/btz863`.

[187] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. ISSN 1533-7928. URL `http://scikit-learn.sourceforge.net.`

[188] Rocío Espada, R. Gonzalo Parra, Thierry Mora, Aleksandra M. Walczak, and Diego U. Ferreiro. Capturing coevolutionary signals inrepeat proteins. *BMC Bioinformatics*, 16(1):1, December 2015. ISSN 14712105. doi: 10.1186/s12859-015-0648-3. URL `http://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-015-0648-3`. Publisher: BioMed Central Ltd.

[189] Julian Salazar, Davis Liang, Toan Q. Nguyen, and Katrin Kirchhoff. Pseudolikelihood Reranking with Masked Language Models. *CoRR*, abs/1910.14659, 2019. URL `http://arxiv.org/abs/1910.14659`. _eprint: 1910.14659.

[190] Lukas Neumann, Andrew Zisserman, and Andrea Vedaldi. Relaxed Softmax: Efficient Confidence Auto-Calibration for Safe Pedestrian Detection. page 8, 2018.

[191] Jeremy Nixon, Mike Dusenberry, Ghassen Jerfel, Timothy Nguyen, Jeremiah Liu, Linchuan Zhang, and Dustin Tran. Measuring Calibration in Deep Learning. April 2019. URL `http://arxiv.org/abs/1904.01685`. _eprint: 1904.01685.

Appendix A

Evaluating Protein Transfer Learning with TAPE

A.1 Dataset Details

In Table A.1 we show the size of all train, validation, and test sets.

We provide further details about dataset sources, preprocessing decisions, data splitting, and experimental challenges in obtaining labels for each of our supervised tasks below. For ease of reading, each section starts with the following items:
(Dataset) The source of the dataset and creation of train/test splits.
(Labeling) The current approach to acquiring supervised labels for this task.

Secondary Structure Details

(Dataset) We use a training and validation set from Klausen et al. [47], which is filtered such that no two proteins have greater than 25% sequence identity. We use three test sets, CB513 [49], CASP12 [37], and TS115 [178]. The training set is also filtered at the 25%

Table A.1: Dataset sizes

Task	Train	Valid	Test
Language Modeling	32,207,059	N/A	2,147,130 (Random split) / 44,314 (Heldout)
Secondary Structure	8,678	2,170	513 (CB513) / 115 (TS115) / 21 (CASP12)
Contact Prediction	25,299	224	40 (CASP12)
Remote Homology	12,312	736	718 (Fold) / 1,254 (Superfamily) / 1,272 (Family)
Fluorescence	21,446	5,362	27,217
Stability	53,679	2,447	12,839

identity threshold with these test sets. This filtering tests the model's ability to generalize in the interesting case where test proteins are not closely related to train proteins.

(Labeling) Determining the secondary structure of a protein experimentally requires high-resolution imaging of the structure, a particularly labor intensive task for structural biologists. Imaging often uses Cryo Electron-Microscopy or X-Ray Crystallography, which can take between weeks and years and can cost over $\$200,000$ [179].

Contact Prediction Details

(Dataset) We use training, validation, and test sets from ProteinNet [38], which uses a test set based on the CASP12 [37] competition, with training and validation sets filtered at the 30% sequence identity threshold. This tests the ability of the model to generalize to proteins that are not closely related to any train proteins.

(Labeling) Determining the contacts of a protein requires knowing its full 3D structure. As with secondary structure, determining the 3D structure requires imaging a protein.

Remote Homology Details

(Dataset) We use a training, validation, and test set from [51], derived from the SCOP 1.75 database [180] of hierarchically classified protein domains. All proteins of a given fold are further categorized into related *superfamilies*. Entire superfamilies are held out from the training set, allowing us to evaluate how the model generalizes across evolutionary distance when structure is preserved.

(Labeling) Each fold is annotated from the structure of the sequence, which SCOP pulls from the Protein DataBank [180, 181]. Finding new superfamilies with the same fold is a challenging task, requiring sequencing in extreme environments as is often done in metagenomics [182].

Fluorescence Details

(Dataset) We use data generated by an experimental technique called Deep Mutational Scanning (DMS) [54]. This technique allows for extensive characterizations of small neighborhoods of a parent protein through mutagenesis. We create training, validation, and test splits ourselves, partitioning the data so that train and validation are in a Hamming distance 3 neighborhood of the original protein, while test data is a sample from the Hamming distance 4-15 neighborhood.

(Labeling) DMS is efficient for local characterization near a single protein, but its samples become vanishingly small once neighborhoods start to expand outside of Hamming distance 2.

Stability Details

(Dataset) We use data generated by a novel combination of parallel DNA synthesis and protein stability measurements [56]. We create training, validation, and test splits ourselves, partitioning the data so that training and validation sets come from four rounds of experimental data measuring stability for many candidate proteins, while our test set consists of seventeen 1-Hamming distance neighborhoods around promising proteins observed in the four rounds of experimentation.

(Labeling) This approach for observing stability is powerful because of its throughput, allowing the authors to find the most stable proteins ever observed for certain classes [56]. The authors observe that the computational methods used to guide their selection at each stage could be improved, meaning that in this case better models could actually lead to better labeled data in a virtuous cycle.

A.2 Featurization of Pretrained Models

We followed standard practice for feeding large pretrained models into downstream supervised architectures. For the Transformer, ResNet, and UniRep we extracted a vector of dimension 512, 256 and 1900, respectively, at each position. For the LSTM and Bepler, we obtained a forward and backward vector at each position, which we concatenated. This resulted in vectors of dimension 2048 and 1024, respectively. For details on how these vectors were used for downstream tasks, see the next section.

A.3 Supervised Architectures

For each task, we fixed one supervised architecture and tried one-hot, alignment-based, and neural net based features. We did not perform hyperparameter tuning or significant architecture optimization, as the main goal was to compare feature extraction techniques.

For each task we define the supervised architecture below. If this is a state of the art architecture from other work, we highlight any novel training procedure or inputs they take.

Secondary Structure Architecture

We used the NetSurfP2.0 model from Klausen et al. [47]. The model consists of two convolutional layers followed by two bidirection LSTM layers and a linear output layer. The convolutional layers have filter size 32 and kernel size 129 and 257, respectively. The bidirectional LSTM layers have 1024 hidden units each.

Our evolutionary features for secondary structure prediction are transition probabilities and state information from HHblits [27], an HMM-HMM alignment method. In the original model, the authors take HHblits outputs in addition to a one-hot encoding of the sequence, giving 50-dimensional inputs at each position. They train the model on multiple tasks

including secondary structure prediction (3 and 8 class), bond-angle prediction, and solvent accessibility prediction. For clarity, we only compared to the model trained without the multitask training, which in our experiments contributed an extra one to two percent in test accuracy. In addition to multitask training, they balance the losses between different tasks to achieve maximum accuracy on secondary structure prediction. All features and code to do the full multitask training is available in our repository.

Contact Prediction Architecture

We used a supervised network inspired by the RaptorX-Contact model from Ma et al. [63]. Since a contact map is a 2D pairwise prediction, we form a 2D input from our 1D features by concatenating the features at position i and j for all i, j. This 2D input is then passed through a convolutional residual network with. The 2D network contains 30 residual blocks with two convolutional layers each. Each convolution in the residual block has filter size 64 and a kernel size of 3.

Currently our evolutionary features for contact prediction are Position Specific Scoring Matrices (PSSMs) available in ProteinNet [38]. The original RaptorX method has a more complex evolutionary featurization: they construct a Multiple Sequence Alignment for each protein, then pass it through CCMpred [88] - a Markov Random Field based contact prediction method. This outputs a 2D featurization including mutual information and pairwise potential. This, along with 1D HMM alignment features and the one-hot encoding of each amino acid are fed to their network. We are currently recreating this pipeline to use these features instead of PSSMs, as the results reported by RaptorX are better than those with the PSSMs.

Remote Homology Architecture

Remote homology requires a single prediction for each protein. To obtain a sequence-length invariant protein embedding we compute an attention-weighted mean of the amino acid embeddings. More precisely, we predict an attention value for each position in the sequence using a trainable dense layer, then use those attention values to compute an attention-weighted mean protein embedding. This protein embedding is then passed through a 512 hidden unit dense layer, a relu nonlinearity, and a final linear output layer to predict logits for all 1195 classes. We note that Hou et al. [51] propose a deep architecture for this task and report state of the art performance. When we compared the performance of this supervised architecture to that of the attention-weighted mean above, the attention-based embedding performed better for all featurizations. As such, we choose to report results using the simpler attention-based downstream architecture.

Our evolutionary features for remote homology detection are Position Specific Scoring Matrices (PSSMs), following the recent work DeepSF [51]. The current state of the art method in this problem, DeepSF [51], takes in a one-hot encoding of the amino acids, predicted secondary structure labels, predicted solvent accessibility labels, and a 1D alignment-based features. In an ablation study, the authors show that the secondary structure labels are most

useful for performance of their model. We report only one-hot and alignment-based results in the main paper to maintain consistency with alignment-based featurizations for other tasks. All input features used by DeepSF are available in our repository.

Protein Engineering Architectures

Protein engineering also requires a single prediction for each protein. Therefore we use the same architecture as we do for remote homology, computing an attention-weighted mean protein embedding, a dense layer with 512 hidden units, a relu nonlinearity and a final linear output layer to predict the quantity of interest (either stability or fluorescence).

Since we create these training, validation, and test splits ourselves, no clear previous state of the art exists. Related work on protein engineering has used a similar architecture by computing a single protein embedding followed by some form of projection (linear or with a small feed forward network) [25, 43]. These methods also do not take in alignment-based features and only use one-hot amino acids as inputs.

A.4 Training Details

Self-supervised models were all trained on four NVIDIA V100 GPUs on AWS for 1 week. Training used a learning rate of 10^{-3} with a linear warm up schedule, the Adam optimizer, and a 10% dropout rate. Since proteins vary in length significantly, we use variable batch sizes depending on the length of the protein. These sizes also differ based on model architecture, as some models (e.g. the Transformer) have significantly higher memory requirements. Specific batch sizes for each model at each protein length are available in our repository.

Supervised models were trained on two NVIDIA Titan Xp GPUs until convergence (no increase in validation accuracy for 10 epochs) with the exception of the memory-intensive Contact Prediction task, which was trained on two NVIDIA Titan RTX GPUs until convergence. Training used a learning rate of 10^{-4} with a linear warm up schedule, the Adam optimizer, and a 10% dropout rate. We backpropagated fully through all models during supervised fine-tuning.

In addition, due to high memory requirements of some downstream tasks (especially contact prediction) we use memory saving gradients [183] to fit more examples per batch on the GPU.

A.5 Pfam Heldout Families

The following Pfam clans were held out during self-supervised training: CL0635, CL0624, CL0355, CL0100, CL0417, CL0630. The following Pfam families were held out during self-supervised training: PF18346, PF14604, PF18697, PF03577, PF01112, PF03417. First, a "clan" is a cluster of families grouped by the maintainers of Pfam based on shared function or evolutionary origin (see [184] for details). We chose holdout clans and families in pairs,

Table A.2: Results for small pretrained models on downstream supervised tasks

Method	Structure		Evolutionary	Engineering	
	SS	Contact	Homology	Fluorescence	Stability
Transformer (small)	0.70	0.31	0.13	0.68	0.68
LSTM (small)	0.73	0.26	0.18	0.66	0.67
ResNet (small)	0.73	0.37	0.11	0.43	0.68

where a clan of novel function is held out together with a family that is similar in sequence but different evolutionarily or functionally. This serves to simultaneously test generalization across large distances (entirely held out families) and between similar looking unseen groups (e.g. the paired holdout clan and holdout family).

A.6 Bepler Supervised Training

We perform supervised pretraining using the same architecture described in Bepler and Berger [24]. We train on the same tasks, a paired remote homology task and contact map prediction task. However, in order to accurately report results on downstream secondary structure, contact map, and remote homology datasets, which were filtered by sequence identity, we perform this same sequence identity filtering on the supervised pretraining set. This reduced the supervised pretraining dataset size by 75% which likely reduced the effectiveness of the supervised pretraining. Both filtered and unfiltered supervised pretraining datasets are made available in our repository.

A.7 Model Size Ablation

In this benchmark we made the choice to train relatively large, 40 million parameter models as larger models have been found to improve performance in other applications of deep learning. To determine whether this trend holds for our benchmark, as well as to quantify the performance difference, we evaluate smaller versions of our three models (the Transformer, LSTM, and ResNet).

Our Transformer model has 6 layers, a hidden dimension of 256, a filter dimension of 512, and 8 attention heads, for a total of 3,315,200 parameters. Our LSTM model has 3 layers with 128 hidden units each, for a total of 796288 parameters. Our ResNet has 8 layers, a filter size of 256, a kernel size of 3, and a dilation rate of 2, for a total of 3,268,992 parameters. Each model was trained for 1,000,000 gradient updates on Pfam, in the same manner that the corresponding large models were trained. Results are reported in Table A.2.

Table A.3: Detailed secondary structure results

		Three-Way Accuracy (Q3)			Eight-Way Accuracy (Q8)		
		CB513	CASP12	TS115	CB513	CASP12	TS115
No Pretrain	Transformer	0.70	0.68	0.73	0.51	0.52	0.58
	LSTM	0.71	0.69	0.74	0.47	0.48	0.52
	ResNet	0.70	0.68	0.73	0.55	0.56	0.61
Pretrain	Transformer	0.73	0.71	0.77	0.59	0.59	0.64
	LSTM	0.75	0.70	0.78	0.59	0.57	0.66
	ResNet	0.75	0.72	0.78	0.58	0.58	0.64
Supervised [24]	LSTM	0.73	0.70	0.76	0.58	0.57	0.65
UniRep [25]	mLSTM	0.73	0.72	0.77	0.57	0.59	0.63
Baseline	One-hot	0.69	0.68	0.72	0.52	0.53	0.58
	Alignment	**0.8**	**0.76**	**0.81**	**0.63**	**0.61**	**0.68**

We note several interesting phenomena from this table. First, we see a drop in performance across all models and tasks, with the exception of the ResNet on the Contact Prediction task and the Transformer on the Fluorescence task. Second, with the exception of the Contact Prediction task, the relative ordering of the models is preserved, even while overall performance decreases. As the Contact Prediction task has the most complicated downstream architecture, it suggests that the downstream architecture has a large effect on performance.

A.8 Detailed Results on Supervised Tasks

Here we provide detailed results on each task, examining multiple metrics and test-conditions to further determine what the models are learning.

Secondary Structure Results

We perform both three-class and eight-class secondary structure classification following the DSSP labeling system [185]. Three way classification tags each position as either Helix, Strand or Other. Eight-way classification breaks these three labels into more specialized classes, for example Helix is broken into 3-turn, 4-turn or 5-turn helix. Table A.3 shows results on these tasks. We note that test-set performance is comparable for all three test sets, in particular alignment does better at both eight-way and three-way classification by a large margin.

We follow the standard notation, where Q3 refers to three-way classification accuracy and Q8 refers to eight-way classification accuracy.

Table A.4: Detailed short-range contact prediction results. Short range contacts are contacts between positions separated by 6 to 11 positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.29	0.25	0.32	0.4
	LSTM	0.23	0.22	0.26	0.33
	ResNet	0.2	0.18	0.24	0.31
Pretrain	Transformer	0.35	0.28	0.35	0.46
	LSTM	0.35	0.26	0.36	0.49
	ResNet	0.32	0.25	0.34	0.46
Supervised [24]	LSTM	0.33	0.27	0.35	0.44
UniRep [25]	mLSTM	0.27	0.23	0.3	0.39
Baseline	One-hot	0.3	0.26	0.34	0.42
	Alignment	**0.51**	**0.35**	**0.5**	**0.66**

Table A.5: Detailed medium-range contact prediction results. Medium range contacts are contacts between positions separated by 12 to 23 positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.2	0.18	0.24	0.31
	LSTM	0.13	0.13	0.15	0.19
	ResNet	0.15	0.14	0.18	0.23
Pretrain	Transformer	0.23	0.19	0.25	0.33
	LSTM	0.23	0.2	0.26	0.34
	ResNet	0.23	0.18	0.25	0.35
Supervised [24]	LSTM	0.26	0.22	0.29	0.37
UniRep [25]	mLSTM	0.2	0.17	0.23	0.32
Baseline	One-hot	0.2	0.17	0.23	0.3
	Alignment	**0.45**	**0.32**	**0.45**	**0.59**

Contact Prediction Results

We report all metrics commonly used to capture contact prediction results [63] in Tables A.5 and A.6. The metrics "P@K" are precision for the top K contacts, where all contacts are sorted from highest confidence to lowest confidence. Note that L is the length of the protein, so "P@L/2", for example, denotes the precision for the $L/2$ most likely predicted contacts in

Table A.6: Detailed long-range contact prediction results. Long range contacts are contacts between positions separated by 24 or more positions, inclusive.

		AUPRC	P@L	P@L/2	P@L/5
No Pretrain	Transformer	0.09	0.15	0.17	0.19
	LSTM	0.05	0.1	0.12	0.15
	ResNet	0.06	0.11	0.13	0.15
Pretrain	Transformer	0.1	0.17	0.2	0.24
	LSTM	0.11	0.2	0.23	0.27
	ResNet	0.06	0.1	0.13	0.17
Supervised [24]	LSTM	0.11	0.18	0.22	0.26
UniRep [25]	mLSTM	0.09	0.17	0.2	0.22
Baseline	One-hot	0.07	0.13	0.16	0.22
	Alignment	**0.2**	**0.33**	**0.42**	**0.51**

a protein of length L. In Table A.5 we report all metrics for medium range contacts, which are contacts for positions between five and twelve amino acids apart. In Table A.6 we report all metrics for long range contacts, which are contacts for positions greater than 12 amino acids apart.

All results decay as we transition from short range to long range contacts, which we note is *not* the case for many state of the art methods from recent CASP competitions [62, 63].

Remote Homology Results

In Table A.7, we report results on three remote homology test datasets constructed in Hou et al. [51]. Recall that "Fold" has the most distantly related proteins from train, while "Superfamily" and "Family" are increasingly related (see Appendix A.1 for more details). This is reflected in the accuracies in Table A.7, which increase drastically as the test sets get easier.

Fluorescence Results

Fluorescence distribution in the train, validation, and test sets is bimodal, with one mode corresponding to bright proteins and one mode corresponding to dark proteins. The dark mode is significantly more diverse in the test set than the train and validation sets, which makes sense as most random mutations will destroy the refined structure necessary for fluorescence. With this in mind, we report Spearman's ρ and mean-squared-error (MSE) on the whole test-set, on only dark mode, and on only the bright mode in Table A.8. The drop in MSE for both modes shows that pretraining helps our best models distinguish between

Table A.7: Detailed remote homology prediction results

		Fold		Superfamily		Family	
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5
No Pretrain	Transformer	0.09	0.21	0.07	0.2	0.31	0.58
	LSTM	0.12	0.28	0.13	0.29	0.68	0.85
	ResNet	0.1	0.24	0.07	0.19	0.39	0.6
Pretrain	Transformer	0.21	0.37	0.34	0.51	0.88	0.94
	LSTM	**0.26**	**0.43**	**0.43**	**0.59**	**0.92**	**0.97**
	ResNet	0.17	0.29	0.31	0.44	0.77	0.87
Supervised [24]	LSTM	0.17	0.30	0.20	0.36	0.79	0.91
UniRep [25]	mLSTM	0.23	0.39	0.38	0.54	0.87	0.94
Baseline	One-hot	0.09	0.21	0.08	0.21	0.39	0.66
	Alignment	0.09	0.21	0.09	0.24	0.53	0.77

Table A.8: Detailed fluorescence prediction results. ρ denotes Spearman ρ.

		Full Test Set		Bright Mode Only		Dark Mode Only	
		MSE	ρ	MSE	ρ	MSE	ρ
No Pretrain	Transformer	2.59	0.22	0.08	0.08	3.79	0
	LSTM	2.35	0.21	0.11	0.05	3.43	-0.01
	ResNet	2.79	-0.28	**0.07**	-0.07	4.1	-0.01
Pretrain	Transformer	0.22	**0.68**	0.09	0.60	0.29	**0.05**
	LSTM	**0.19**	0.67	0.12	0.62	**0.22**	0.04
	ResNet	3.04	0.21	0.12	0.05	4.45	0.02
Supervised [24]	LSTM	2.17	0.33	0.08	0.06	3.17	0.02
UniRep [25]	mLSTM	0.20	0.67	0.13	**0.63**	0.24	0.04
Baseline	One-hot	2.69	0.14	0.08	0.03	3.95	0.0

Table A.9: Overall stability prediction results

		Spearman's ρ	Accuracy
No Pretrain	Transformer	-0.06	0.5
	LSTM	0.28	0.6
	ResNet	0.61	0.68
Pretrain	Transformer	**0.73**	**0.70**
	LSTM	0.69	0.69
	ResNet	**0.73**	0.66
Supervised [24]	LSTM	0.64	0.67
UniRep [25]	mLSTM	**0.73**	0.69
Baseline	One-hot	0.19	0.58

dark and bright proteins. However low Spearman's ρ for the dark mode suggests that models are not able to rank proteins within this mode.

Stability Results

Table A.10: Stability prediction results broken down by protein topology

		$\alpha\alpha\alpha$		$\alpha\beta\beta\alpha$		$\beta\alpha\beta\beta$		$\beta\beta\alpha\beta\beta$	
		ρ	Acc	ρ	Acc	ρ	Acc	ρ	Acc
No Pretrain	Transformer	-0.39	0.49	-0.41	0.47	0.52	0.5	0.25	0.52
	LSTM	-0.07	0.57	0.39	0.7	-0.43	0.56	-0.34	0.56
	ResNet	0.64	0.69	0.16	0.69	0.63	0.67	0.65	0.67
Pretrain	Transformer	0.66	0.68	**0.48**	0.73	0.65	**0.71**	0.65	0.67
	LSTM	0.71	**0.7**	0.17	0.73	**0.68**	0.67	**0.67**	**0.7**
	ResNet	0.68	0.68	0.15	0.63	0.61	0.68	0.6	0.68
Supervised [24]	LSTM	0.33	0.66	0.24	**0.79**	0.54	0.7	0.58	0.53
UniRep [25]	mLSTM	**0.72**	0.66	0.11	0.76	**0.68**	0.66	0.65	0.67
Baseline	One-hot	0.58	0.59	0.04	0.58	-0.05	0.58	0.54	0.58

The goal of the Rocklin et al. [56] experiment was to find highly stable proteins. In the last stage of this experiment they examine variants of the the most promising candidate proteins.

Therefore we wish to measure both whether our model was able to learn the landscape around these candidate proteins, as well as whether it successfully identified those variants with greater stability than the original parent proteins. In Table A.9 we report Spearman's ρ to measure the degree to which the landscape was learned. In addition, we report classification accuracy of whether a mutation is beneficial or harmful using the predicted stability of the parent protein as a decision boundary.

In Table A.10 report all metrics separately for each of the four protein topologies tested in Rocklin et al. [56], where α denotes a helix and β denotes a strand (or β-sheet). We do this because success rates varied significantly by topology in their experiments, so some topologies (such as $\alpha\alpha\alpha$ were much easier to optimize than others (such as $\alpha\beta\beta\alpha$). We find that our prediction success also varies significantly by topology.

Appendix B

Transformer protein language models are unsupervised structure learners

B.1 Notation

In the figures, we report contact precision in the range of 0.0 to 1.0. In the text and in the tables, we report contact precision in terms of percentages, in the range of 0 to 100.

B.2 Average Product Correction (APC)

In protein contact prediction, APC is commonly used to correct for background effects of entropy and phylogeny [118]. Given an $L \times L$ coupling matrix F, APC is defined as

$$F_{ij}^{\text{APC}} = F_{ij} - \frac{F_i F_j}{F} \tag{B.1}$$

Where F_i, F_j, and F are the sum over the i-th row, j-th column, and the full matrix respectively. We apply APC independently to the symmetrized attention maps of each head in the Transformer. These corrected attention maps are passed in as input to a logistic regression.

B.3 Gremlin Implementation Details

Gremlin is trained by optimizing the pseudolikelihood of W and V, which correspond to pairwise and individual amino acid propensities. The pseudolikelihood approximation models the conditional distributions of the original joint distribution and can be written:

$$\log p(x_i^d = a | x_{j\neq i}^d; W_i, V_i) = \log \frac{\exp\left(V_{ia} + \sum_{j=1,j\neq i}^{N} \sum_{b=1}^{20} \mathbb{1}(x_j^d = b) W_{ijab}\right)}{\sum_{c=1}^{20} \exp\left(V_{ic} + \sum_{j=1,j\neq i}^{N} \sum_{b=1}^{20} \mathbb{1}(x_j^d = b) W_{ijcb}\right)} \tag{B.2}$$

Table B.1: Major Architecture Differences in Protein Transformer Language Models

Name	Layers	Hidden Size	Attn Heads	Parameters	Dataset
TAPE	12	768	12	92M	Pfam
ProtBERT-BFD	30	1024	16	420M	BFD100
ESM-1 (6 layer)	6	768	12	43M	Uniref50
ESM-1 (12 layer)	12	768	12	85M	Uniref50
ESM-1 (34 layer)	34	1280	20	670M	Uniref50
ESM-1b	33	1280	20	650M	Uniref50

subject to the constraint that $W_{ii} = 0$ for all i, and that W_{ijab} is symmetric in both sequence (i, j) and amino acid (a, b). Additionally, Gremlin uses a regularization parameter that is adjusted based on the depth of the MSA.

B.4 ESM-1 Implementation Details

The original ESM-1 models were described in [43]. ESM-1 is trained on Uniref50 in contrast to the TAPE model, which is trained on Pfam [46] and the ProtBERT-BFD model, which is trained on Uniref100 and BFD100 [144]. ESM-1b is a new model, which is the result of an extensive hyperparameter sweep that was performed on smaller 12 layer models. ESM-1b is the result of scaling up that model to 33 layers.

Compared to ESM-1, the main changes in ESM-1b are: higher learning rate; dropout after word embedding; learned positional embeddings; final layer norm before the output; and tied input/output word embeddings. Weights for all ESM-1 and ESM-1b models can be found at `https://github.com/facebookresearch/esm`.

B.5 Jackhmmer Details

We use Jackhmmer version 3.3.1 with a bitscore threshold of 27 and 8 iterations to construct MSAs from the ESM training set. The failures on 126 sequences noted in Section 3.4 result from a segmentation fault in hmmbuild after several iterations (the number of successful iterations before the segmentation fault varies depending on the input sequence). Since we see this failure for less than 1% of the dataset we choose to ignore these sequences during evaluation.

Additionally, we evaluated alternate MSAs by running Jackhmmer until a Neff of 128 was achieved (with a maximum of 8 iterations), a procedure described by Zhang et al. [186]. This resulted in very similar, but slightly worse results (average long range P@L 29.3, versus 31.3 when always using the output of the eighth iteration). We therefore chose to report results using the 8 iteration maximum.

Table B.2: Average metrics on 15 CASP13 FM Targets. All baselines use MSAs generated via the trRosetta MSA generation approach.

		long			medium			short		
sep										
Model	Variant	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Baselines	mfDCA	11.0	13.6	19.7	12.8	17.9	26.2	14.4	19.4	26.6
	PSICOV[1]	10.6	14.0	18.3	12.2	17.1	25.9	14.1	19.8	27.9
	Gremlin	12.1	16.1	23.6	14.5	20.8	32.5	16.8	23.4	28.5
ESM-1b (attention)	top-1 heads	11.8	15.8	23.8	17.0	20.8	29.6	13.6	17.9	22.7
	top-5 heads	15.3	20.9	29.6	18.7	27.0	33.0	14.6	20.6	26.8
	top-10 heads	16.6	22.7	32.1	21.8	29.5	39.8	17.9	23.2	30.4
	n=1, s=1	16.4	23.5	34.7	23.0	30.8	41.6	18.1	23.3	29.9
	n=10, s=1	18.6	25.3	39.3	24.1	31.9	41.4	18.7	25.2	33.2
	n=20, s=1	19.3	26.6	37.0	24.0	31.5	40.2	18.6	25.0	33.8
	MSA, s=1	14.2	20.3	30.5	21.0	29.1	42.3	18.4	23.7	31.5
ESM-1b (bilinear)	n=20	14.1	17.7	19.8	17.7	20.9	27.9	11.2	13.9	17.0
	n=14257	**20.8**	**31.1**	**45.9**	**25.7**	**33.7**	**43.3**	**20.1**	**26.2**	**34.2**

B.6 Results on CASP13

In Table B.2 we report results on the 15 CASP13 Free Modeling targets for which PDBs were publicly released. The specific domains evaluated are: T0950-D1, T0957s2-D1, T0960-D2, T0963-D2, T0968s1-D1, T0968s2-D1, T0969-D1, T0980s1-D1, T0986s2-D1, T0990-D1, T0990-D3, T1000-D2, T1021s3-D1, T1021s3-D2, T1022s1-D1. ESM-1b is able to outperform Gremlin, and simply averaging the top-10 heads of ESM-1b has comparable performance to Gremlin.

In addition, we compare our logistic regression model to the bilinear contact prediction model proposed by Rives et al. [43]. This model trains two separate linear projections of the final representation layer and computes contact probabilities via the outer product of the two projections plus a bias term, which generates the following unnormalized log probability:

$$\log p(\text{contact}) \propto (xW_1)(xW_2)^T + b \tag{B.3}$$

Here x is a sequence-length vector of features in $\mathbb{R}^{L\times d}$. Each W_i is a matrix in $\mathbb{R}^{d\times k}$, where k is a hyperparameter controlling the projection size.

We train this model in both the limited supervision ($n = 20$) and full supervision ($n = 14257$) setting. For the limited supervision setting, we use the same 20 proteins used

[1]PSICOV fails to converge on 3 / 15 targets with default parameters. We follow the procedure suggested in `https://github.com/psipred/psicov` to increase `rho` to 0.005 for those domains.

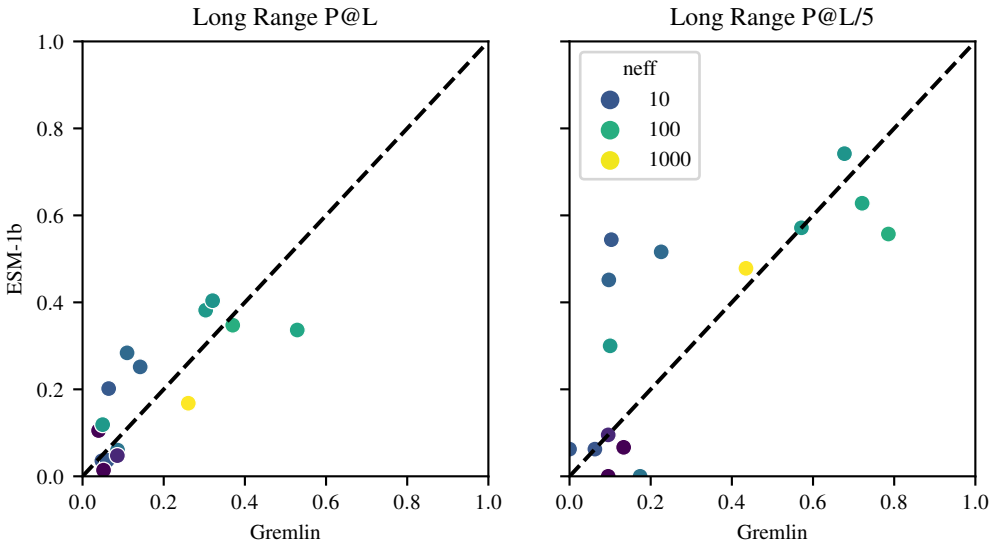


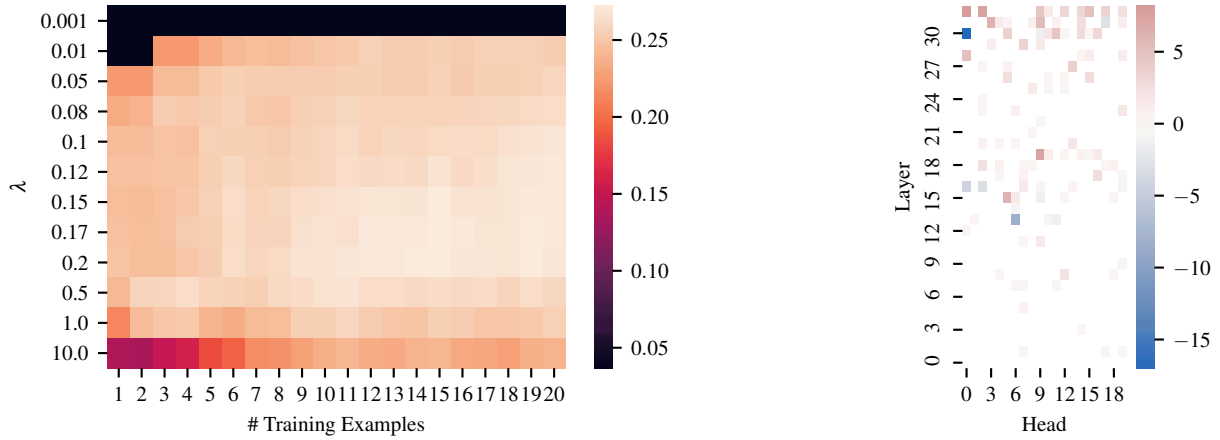
Figure B.1: Results on 15 CASP13 FM Domains colored by Neff.

to train the sparse logistic regression model. For the full supervision setting we generate a 95/5% random training/validation split of the 15008 trRosetta proteins with sequence length ≤ 1024.

We performed independent grid searches over learning rate, weight decay, and hidden size for the two settings. For the $n = 20$ setting, we found a learning rate of 0.001, weight decay of 10.0, and projection size of 512 had best performance on the validation set. For the $n = 14257$ setting we found a learning rate of 0.001, weight decay of 0.01, and projection size of 512 had best performance on the validation set. All models were trained to convergence to maximize validation long range P@L with a patience of 10. The $n = 20$ models were trained with a batch size of 20 (i.e. 1 batch = 1 epoch) and the $n = 14257$ models were trained with a batch size of 128.

The bilinear model performs very poorly in the limited supervision setting, worse than simply taking the top-1 attention head. With full supervision, it moderately outperforms the logistic regression for an increase in long range P@L of 1.5 while using 700x more data.

In Fig. B.1 we display results on the 15 FM targets colored by effective number of sequences. ESM-1b shows higher precision at L and L/5 on average, and is sometimes significantly higher for sequences with low Neff. Since ESM-1b training data was generated prior to CASP13, this suggests ESM-1b is able to generalize well to new sequences.



(a) Gridsearch (b) Weights

Figure B.2: (a) Gridsearch on logistic regression over number of training examples and number regularization penalty. Values shown are long range P@L over a validation set of 20 proteins. (b) Per-head and layer weights of the logistic regression on the best ESM-1b model.

B.7 Logistic Regression Details

Given a model with M layers, H heads, and an input sequence x of length L, let A_{mh} be the $L \times L$ contact map from the h-th head in the m-th layer. We first symmetrize this map and apply APC and let a_{mhij} be the coupling weight between sequence position i and j in the resulting map. Then we define the probability of a contact between positions i and j according to a logistic regression with parameters β:

$$p(c_{ij}^d; \beta) = \frac{1}{1 + \exp\left(-\beta_0 - \sum_{m=1}^{M} \sum_{h=1}^{H} \beta_{mh} a_{mhij}^d \right)} \tag{B.4}$$

To fit β, let $\mathcal{D}$ be a set of training proteins, k be a minimum sequence separation, and λ be a regularization weight. The objective can then be defined as follows:

$$\mathcal{L}(\mathcal{D}; \beta) = \prod_{d \in \mathcal{D}} \prod_{i=1}^{L_d - k} \prod_{j=i+k}^{L_d} p(c_{ij}^d; \beta) \tag{B.5}$$

$$\hat{\beta} = \max_{\beta} \mathcal{L}(\mathcal{D}; \beta) + \frac{1}{\lambda} \sum_{m=1}^{M} \sum_{h=1}^{H} |\beta_{mh}| \tag{B.6}$$

We fit the parameters β via scikit-learn [187] and do not backpropagate the gradients through the attention weights. In total, our model learns $MH + 1$ parameters, many of which are zero thanks to the L_1 regularization.

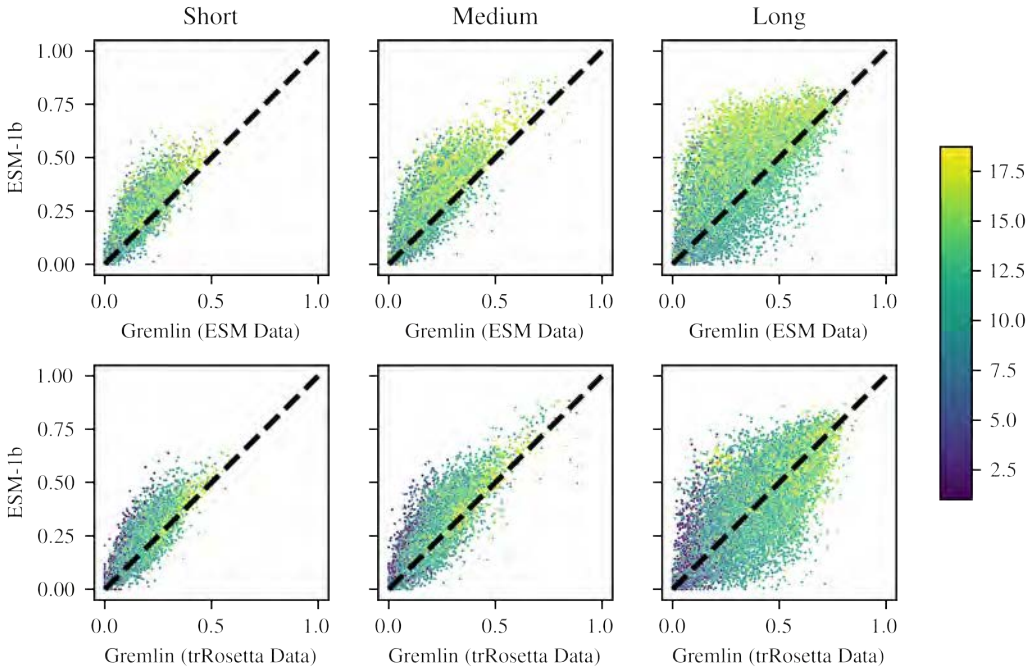


Figure B.3: Short, medium, and long range P@L performance distribution of ESM-1b vs. Gremlin. Each point is colored by the $\log_2$ of the number of sequences in the MSA.

There are three hyperparameters in our training setup: the number of proteins in our training set $\mathcal{D}$, the regularization parameter λ, and the minimum sequence separation of training contacts k. We find that performance improves significantly when increasing the $\mathcal{D}$ from 1 protein to 10 proteins, but that the performance gains drop off when $\mathcal{D}$ increases from 10 to 20 (Fig. 3.1). Through a hyperparameter sweep, we determined that the optimal λ is 0.15. We find that ignoring local contacts ($|i-j|<6$) is also helpful. Therefore, unless otherwise specified, all logistic regressions are trained with $|\mathcal{D}|=20, \lambda=0.15, k=6$. See Fig. B.2a for a gridsearch over the number of training proteins and regression penalty. We used 20 training proteins and 20 validation proteins for this gridsearch. Fig. B.2b shows the weights of the final logistic regression used for ESM-1b.

B.8 Performance Distribution

Fig. B.3 shows the full distribution of performance of ESM-1b compared with Gremlin. When we provide Gremlin access to Uniref100, along with metagenomic sequences, ESM-1b still consistenly outperforms Gremlin when extracting short and medium range contacts. For long range contacts, Gremlin is much more comparable, and has higher contact precision on 47% of sequences. With access to the same set of sequences, ESM-1b consistently outperforms Gremlin in detecting short, medium, and long range contacts. This suggests that ESM-1b

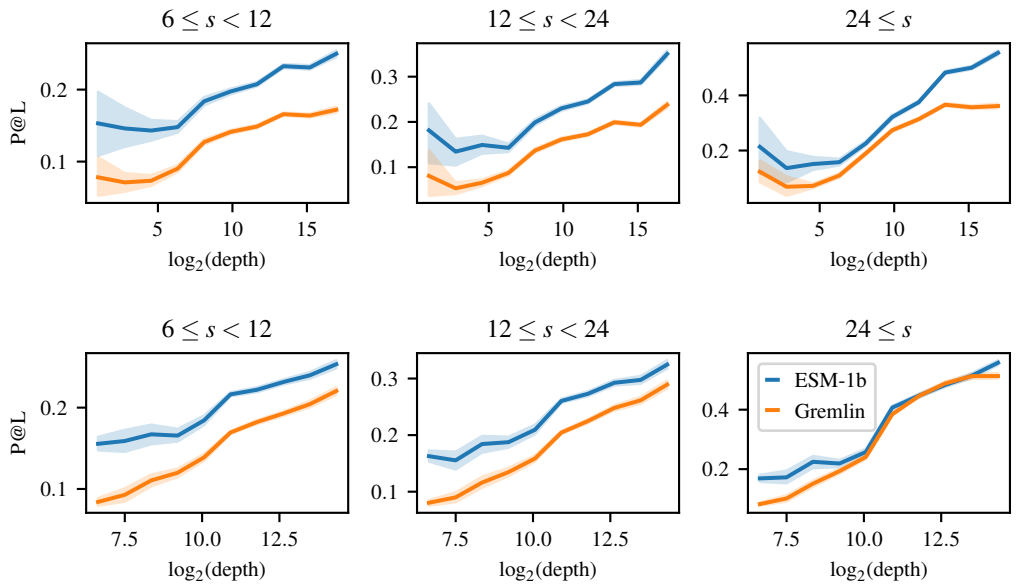


Figure B.4: Gremlin performance binned by MSA depth using both ESM (top) and trRosetta (bottom) MSAs. For comparison, ESM-1b performance is also shown for the sequences in each bin.

can much better extract information from the same set of sequences and suggests that further scaling of training data may improve ESM-1b even further.

This analysis is further borne out in Fig. B.4. Given the same set of sequences, ESM-1b outperforms Gremlin on average for short, medium, and long-range contacts regardless of the depth of the MSA generated from the ESM-1b training set.

Additionally, we find that ESM-1b can provide varying contact maps for different sequences in the MSA (Fig. B.5). This is not possible for Gremlin, which is a family-level model. We leverage this in a fairly simple way to provide a modest boost to the contact precision of ESM-1b (Section 3.5).

B.9 Secondary Structure

In Section 3.5 we show that some heads that detect local contacts (which often correspond to secondary structure) are actually negatively correlated with long range contacts. We test ESM-1b's ability to detect secondary structure via attention by training a separate logistic regression on the Netsurf dataset [47]. As with the logistic regression on contacts, we compute attentions and perform APC + symmetrization. To predict the secondary structure of amino acid i, we feed as input the couplings a_{mhij} for each layer m, for each head h, and for $j \in [i-5, i+5]$, for a total of 7260 input features. Using just 100 of the 8678 training

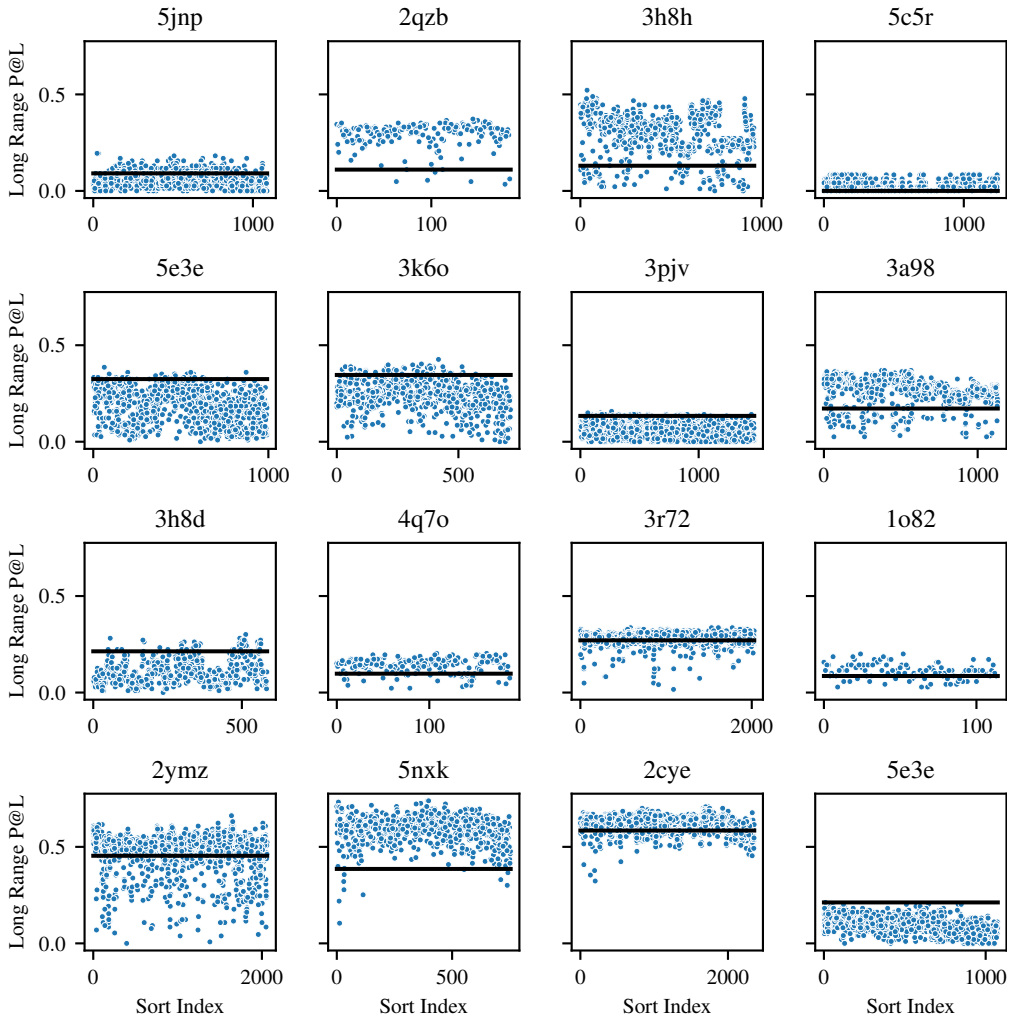


Figure B.5: Distribution of contact perplexity when evaluating different sequences from the same MSA. The x-axis shows the index of each sequence, sorted in ascending order by hamming distance from the query sequence (query sequence is always index 0). The y-axis shows long range P@L. The black line indicates Gremlin performance on that MSA.

.

proteins, we achieve 79.3% accuracy on 3-class secondary structure prediction on the CB513 test set [49]. Fig. B.6 shows the importance of each layer to predicting the three secondary structure classes. There are spikes in different layers for all three classes, indicating that particular heads within those layers are specializing in detecting specific classes of secondary structure.

Fig. B.6 shows importance of each Transformer layer to predicting each of the three secondary structure classes. We see that, as with contact prediction, the most important layers are in the middle layers (14-20) and the final layers (29-33). Some layers spike more

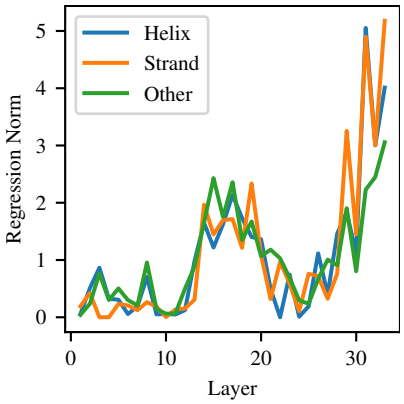


Figure B.6: L_2 norm of weights for 3-class secondary structure prediction by Transformer layer.

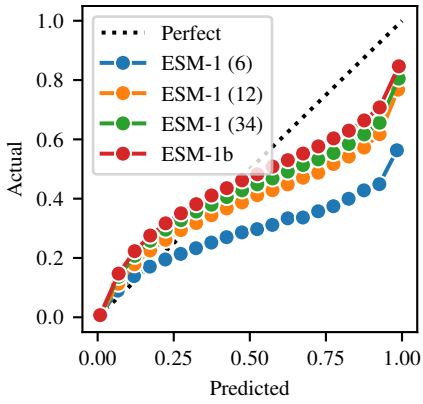


Figure B.7: Calibrated probability of a real contact given predicted probability of contact over all test proteins.

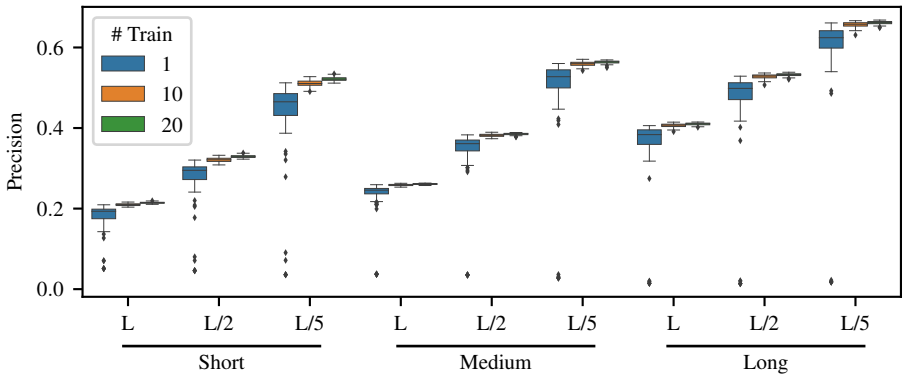
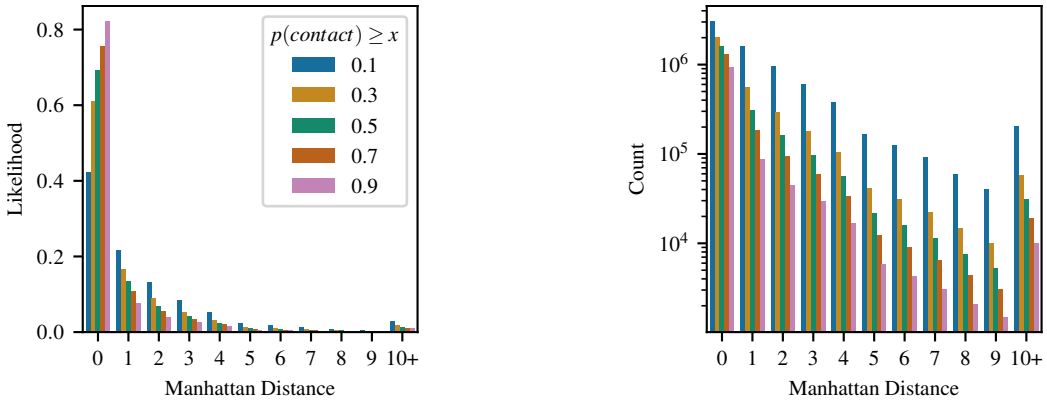


Figure B.8: Distribution of precision for all reported statistics using 100 different logistic regression models. Each regression model is trained on a random sample of $N = 1, 10, 20$ proteins.

heavily on particular contact classes (e.g. layer 33 is important for all classes, but particularly important for β-strand prediction). This suggests that particular heads within these layers activate specifically for certain types of secondary structure.

B.10 Bootstrapped Low-N Confidence Interval

Section 3.5 shows results from Low-N supervision on 1, 10, and 20 proteins. Since performance in this case depends on the particular proteins sampled we use bootstrapping to determine a confidence interval for each of these estimates. Using the full training, validation, and test



(a) Manhattan distance to nearest contact. (b) Count of predictions by Manhattan distance.

Figure B.9: (a) Distribution of Manhattan distance between the coordinates of predicted contacts and the nearest true contact at various thresholds of minimum $p(contact)$. A distance of zero corresponds to a true contact. (b) Actual counts of predictions by Manhattan distance across the full dataset (note y-axis is in log scale).

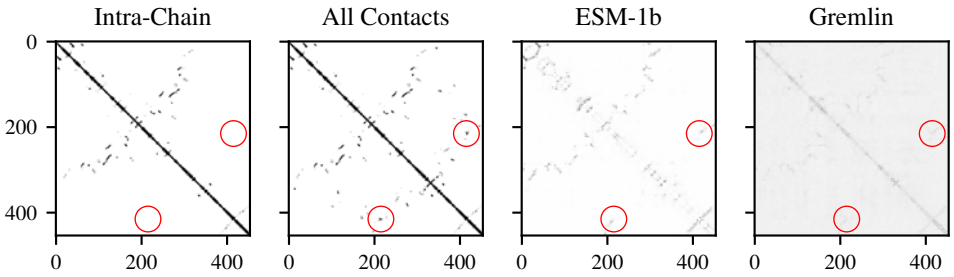
set of 14882 proteins, we train 100 logistic regression models using a random sample of N proteins, for $N = 1$, 10, and 20. Each model is then evaluated on the remaining $14882 - N$ proteins. The full distribution of samples can be seen in Fig. B.8. The confidence interval estimates for long range precision at L with 1, 10, and 20 training proteins are: 35.6 ± 1.8, 40.6 ± 0.1, and 41.0 ± 0.1 respectively.

B.11 Model Calibration and False Positives

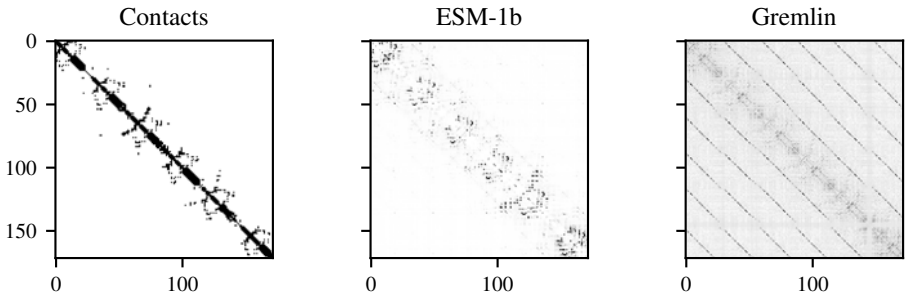
Vig et al. [7] suggested that the attention probability from the TAPE Transformer was a well-calibrated estimator for the probability of a contact. In Fig. B.7 we examine the same with the logistic regression trained on the ESM-1 and ESM-1b models. We note that ESM-1b, in addition to being more accurate overall than Gremlin, also provides actual probabilities.

We find that as with model accuracy, model calibration increases with larger scale and better hyperparameters. The 6, 12, and 34 layer ESM-1 models have mean-squared error of 0.074, 0.028, and 0.020 between predicted and actual contact probabilities, respectively. ESM-1b has a mean squared error of 0.014. Mean squared error is computed between contact probabilites split into 20 bins according to the scikit-learn calibration_curve function. It is therefore reasonable to use the logistic regression probability as a measure of the model's confidence.

In the case of false positive contacts we attempt to measure the Manhattan distance between the coordinates of predicted contacts and the nearest true contact (Fig. B.9a). We observe that the Manhattan distance between the coordinates of false positive contacts are often very close (Manhattan distance between 1-4) to real contacts, and that very few false



(a) Intra-chain, inter-chain, and predicted contacts for 5mlt, which is a homodimer.



(b) Real and predicted contacts for the CTCF protein (pdbid: 5yel).

Figure B.10: Illustration of two modes for ESM-1b where significant numbers of spurious contacts are predicted. (a) Predicted contacts which do occur in the full homodimer complex, but are not present as intra-chain contacts. (b) CTCF protein contacts. A small band of contacts near the 30-residue off-diagonal is predicted by ESM-1b. This band, along with additional similar bands are also predicted by Gremlin.

positives have a Manhattan distance ≥ 10 from a true contact. With a threshold contact probability of 0.5, 83.8% of proteins have at least one predict contact with Manhattan distance > 4 to the nearest contact. This drops to 71.7% with a threshold probability of 0.7, and to 52.5% with a threshold probability of 0.9.

Fig. B.10 highlights two modes for ESM-1b where signficant numbers of spurious contacts are predicted. Fig. B.10a shows one example where the model does appear to hallucinate contacts around residues 215 and 415, which do not appear in the contact map for this protein. However, this protein is a homodimer and these contacts are present in the inter-chain contact map. This suggests that some 'highly incorrect' false positives may instead be picking up on inter-chain contacts. Fig. B.10b shows an example of a repeat protein, for which evolutionary coupling methods are known to pick up on additional 'bands' of contacts [98, 188]. Multiple bands are visible in the Gremlin contact map, while only the first band, closest to the diagonal, is visible in the ESM-1b contact map. More analysis would be necessary to determine the frequency of these modes, along with additional potential modes.

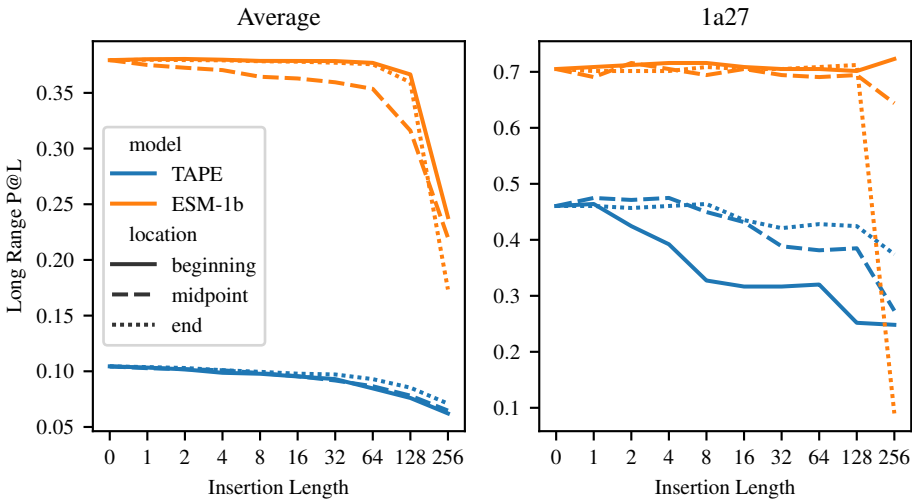


Figure B.11: Robustness of ESM-1b and TAPE models to insertions of Alanine at the beginning, middle, and end of sequence

.

B.12 Alignment

One hypothesis as to the benefit of large language models as opposed to simpler Potts models is that they may be able to learn an implicit alignment due to their learned positional embedding. For a Potts Model, an alignment enables a model to relate positions in the sequence given evolutionary context despite the presence of insertions or deletions. We test the robustness of the model to insertions by inserting consecutive alanines at the beginning, middle, or end of 1000 randomly chosen sequences with initial sequence length < 512 (we limit initial sequence length in order to avoid out-of-memory issues after insertion). We find that ESM-1b can tolerate up to 256 insertions at the beginning or end of the sequence and up to 64 insertions in the middle of the sequence before performance starts to significantly degrade. This suggests that ESM-1b learns a robust implicit alignment of the protein sequence.

On the other hand, we find that the TAPE Transformer is less robust to insertions. On one sequence (pdbid: 1a27), we find the TAPE Transformer drops in precision by 12 percentage points after adding just 8 alanines to the beginning of the sequence, while ESM-1b sees minimal degradation until 256 alanines are inserted. We hypothesize that, because TAPE was trained on protein domains, it did not learn to deal with mis-alignments in the input sequence.

B.13 Evolutionary Finetuning Details

We finetuned each model using a learning rate of 1e-4, 16k warmup updates, an inverse square root learning rate schedule, and a maximum of 30 epochs. This resulted in a varying

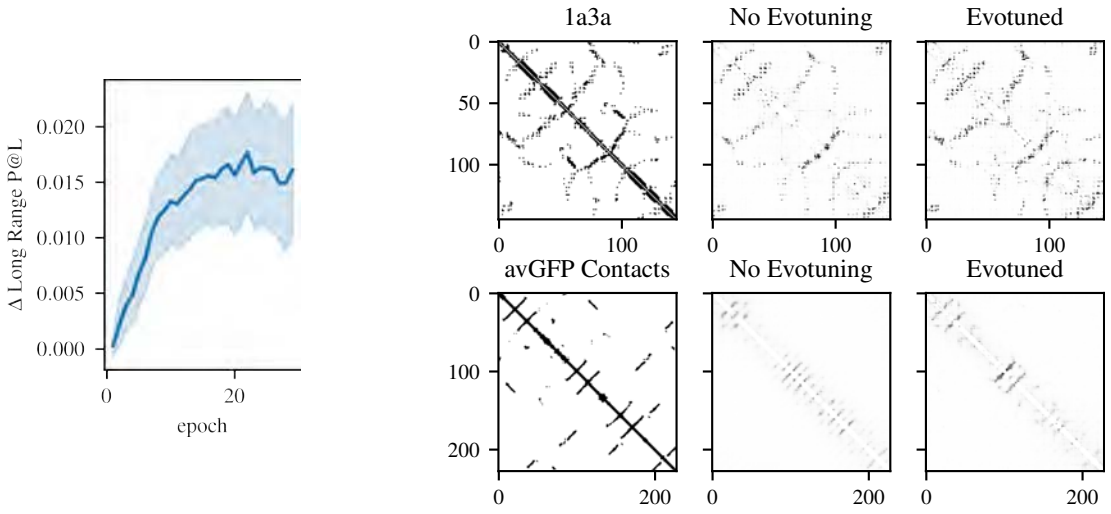


Figure B.12: Left: Average change in contact precision vs. number of finetuning epochs over 380 proteins. Right: Real and predicted contacts before and after evolutionary finetuning for 1a3a and avGFP. For 1a3a, long range P@L improves from 54.5 to 61.4. For avGFP, long range P@L improves from 7.9 to 11.4.

number of total updates depending on the size of the MSA, with larger MSAs being allowed to train for more updates. This should ideally help prevent the model from overfitting too quickly on very small MSAs. We use a variable batch size based on the length of the input proteins, fixing a maximum of 16384 tokens per batch (so for a length 300 protein this would correspond to a batch size of 54). We use MSAs from trRosetta for finetuning all proteins with the exception of avGFP, where we use the same set of sequences from Alley et al. [25].

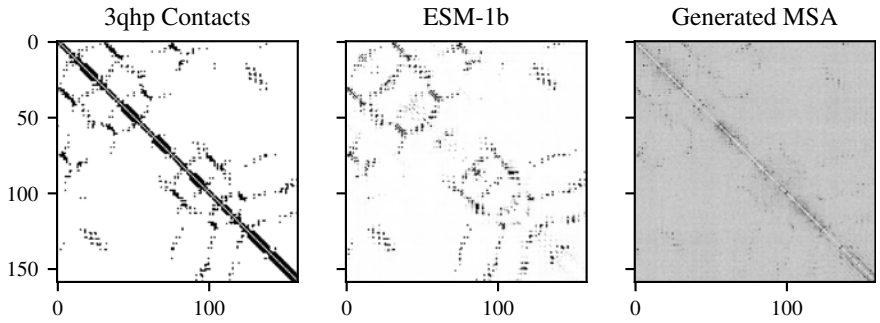


Figure B.13: Contacts for 3qhp from Gremlin trained on pseudo-MSA generated by ESM-1b, compared to real and ESM-1b predicted contacts. The generated MSA achieves a long-range P@L of 52.2 while the attention maps achieve a precision of 76.7.

B.14 MSA Generation

Result: Generated MSA
input `// protein sequence`
curr = input `// optionally, the input can be repeated for batching`
for $0 \leq i < 10000$ **do**
    masked = mask 20% of positions in curr;
    pred = model(masked);
    curr[masked positions] = pred[masked positions];
    MSA.append(curr);
    if $random() < 0.1$ **then**
        curr = input;
    end
end

Algorithm 1: Quickly generate a pseudo-MSA from an input sequence.

Algorithm 1 presents the algorithm used to generate pseudo-MSAs from ESM-1b. Each pseudo-MSA is passed to GREMLIN in order to evaluate the preservation of contact information (Fig. B.13).

Appendix C

MSA Transformer

C.1 Unsupervised Contact Prediction

For unsupervised contact prediction, we adopt the methodology from Chapter 3, which shows that sparse logistic regression trained on the attention maps of a single-sequence transformer is sufficient to predict protein contacts using a small number (between $1 - 20$) of training structures. To predict the probability of contact between amino acids at position i and j, the attention maps from each layer and head are independently symmetrized and corrected with APC [118]. The input features are then the values $\bar{a}_{lhij}$ for each layer l and head h. The models have 12 layers and 12 heads for a total of 144 attention heads.

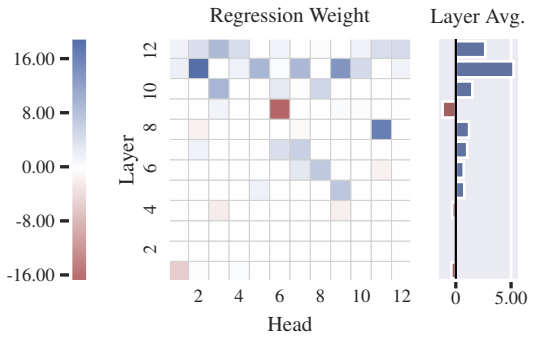


Figure C.1: Weight values of learned sparse logistic regression trained on 20 structures. A sparse subset (55 / 144) of contact heads, largely in the final layers, are predictive of protein contacts.

An L1-regularization coefficient of 0.15 is applied. The regression is trained on all contacts with sequence separation ≥ 6. 20 structures are used for training. Trained regression weights are shown in Fig. C.1.

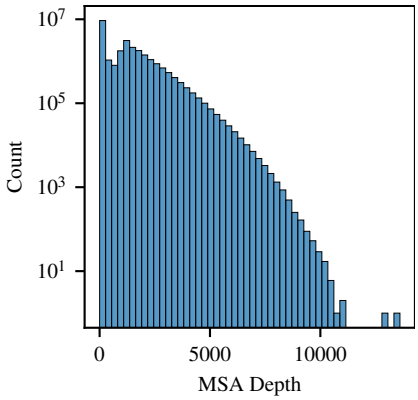


Figure C.2: Distribution of MSA depths in the MSA Transformer training set. Average MSA depth is 1192 and median MSA depth is 1101.

C.2 Dataset Generation

For the unsupervised training set we retrieve the UniRef-50 [71] database dated 2018-03. The UniRef50 clusters are partitioned randomly in 90% train and 10% test sets. For each sequence, we construct an MSA using HHblits, version 3.1.0. [117] against the $\text{UniClust30}_{2017-10}$ database [116]. Default settings are used for HHblits except for the the number of search iterations (`-n`), which we set to 3.

C.3 Ablation Studies

Ablation studies are conducted over a set of seven hyperparameters listed in Table C.2. Since the cost of an exhaustive search over all combinations of hyperparameters is prohibitive, we instead train an exhaustive search over four of the hyperparameters (embedding size, block order, tied attention, and masking pattern) for 10k updates. The best run is then selected as the base hyperparameter setting for the full ablation study, in which only one parameter is changed at a time.

For the full ablation study, each model is trained for 100k updates using a batch size of 512. The four best performing models are then further trained to 150k updates. Contact prediction on the trRosetta dataset [133] is used as a validation task. Precision after 100k updates (and 150k for the best models) is reported in Table C.2 and the full training curves are shown in Fig. C.3. The model with best hyperparameters is then further trained to 450k updates. The performance of this model is reported in Table C.3. Validation perplexity is also reported in Table C.2. In general we find limited correspondence between perplexity and contact prediction performance across models.

Potts [73], TAPE transformer (see Chapter 2), ESM-1b [43], ProtBERT-BFD, and ProTrans-T5 [94] are used as unsupervised contact prediction comparisons. The best MSA

Table C.1: Validation perplexity and denoising accuracy on UniRef50 validation MSAs. PSSM probabilities and nearest-neighbor matching are used as baselines. To compute perplexity under the PSSM, we construct PSSMs using the input MSA, taking the cross-entropy between the PSSM and a one-hot encoding of the masked amino acid. When calculating PSSM probabilities, we search over pseudocounts in the range $[10^{-10}, 10)$, and select 10^{-2}, which minimizes perplexity. For denoising accuracy, the argmax for each column is used. For nearest-neighbor matching, masked tokens are predicted using the values from the sequence with minimum hamming distance to the masked sequence. This does not provide a probability distribution, so perplexity cannot be calculated. MSAs with depth 1 are ignored, since the baselines fail in this condition. Perplexity ranges from 1 for a perfect model to 21 for a uniform model selecting over the common amino acids and gap token.

Model	Perplexity	Denoising Accuracy
PSSM	14.1	41.4
Nearest-Neighbor	-	46.7
MSA Transformer	**2.44**	**64.0**

Table C.2: Hyperparameter search on MSA Transformer. P@L is long-range ($s \geq 24$) precision on unsupervised contact prediction following. Perplexity is reported after 100k updates and precision is reported after 100k and 150k updates.

D	Block	Tied	Masking	Mask p	MSA Pos Emb	Subsample	P@L (100k)	P@L (150k)	Ppl (100k)
768	Row-Column	Sqrt	Uniform	0.15	No	Log-uniform	**56.3**	56.3	3.01
384							52.8	-	3.10
	Column-Row						55.7	-	3.01
		None					42.1	-	3.03
		Mean					50.1	-	3.00
			Column				38.8	-	3.54
				0.2			**56.6**	56.3	3.04
					Yes		**56.5**	**57.1**	3.00
						Full	**56.5**	56.1	**2.91**

Transformer outperforms all other methods by a wide margin, increasing long-range precision at L by a full 16 points. See below for a discussion of all seven hyperparameters.

Embedding Size (D)

Since the MSA Transformer is provided with more information than single sequence protein language models, it is possible that many fewer parameters are needed to learn the data

Table C.3: Average precision on 14842 test structures for MSA and single-sequence models trained on 20 structures.

	$6 \leq \text{sep} < 12$			$12 \leq \text{sep} < 24$			$24 \leq \text{sep}$		
Model	L	L/2	L/5	L	L/2	L/5	L	L/2	L/5
Potts	17.2	26.7	44.4	21.1	33.3	52.3	39.3	52.2	62.8
TAPE	9.9	12.3	16.4	10.0	12.6	16.6	11.2	14.0	17.9
ProtBERT-BFD	20.4	30.7	48.4	24.3	35.5	52.0	34.1	45.0	57.4
ProTrans-T5	20.1	30.6	48.5	24.6	36.1	52.4	35.6	46.1	57.8
ESM-1b	21.6	33.2	52.7	26.2	38.6	56.4	41.1	53.3	66.1
MSA Transformer	**25.6**	**41.0**	**64.6**	**31.9**	**48.9**	**71.1**	**57.4**	**71.7**	**82.1**

Table C.4: Supervised Contact Prediction performance on CASP13-FM and CAMEO-hard targets. Reported numbers are long-range ($s \geq 24$) contact precision. Three variants of the MSA Transformer are included for comparison: *unsupervised model, †supervised model using final hidden representations of the reference sequence as input, ‡supervised model using final hidden representations of reference sequence and all attention maps as input. Baseline and final trRosetta models are also included for comparison. L is defined as the number of valid residues.

	CASP13-FM			CAMEO		
Model	L	L/2	L/5	L	L/2	L/5
Co-evolutionary	40.1	52.5	65.2	47.3	60.9	72.1
Unirep	11.2	14.5	16.6	17.8	23.0	30.8
SeqVec	13.8	18.3	21.9	22.5	30.3	39.8
TAPE	12.3	14.4	17.8	15.9	20.6	26
ProtBERT-BFD	24.7	32.1	40.6	37.0	48.1	60.0
ProTrans-T5	25.0	32.9	41.4	40.8	52.5	63.3
ESM-1b	28.2	37.4	50.2	44.4	57.2	68.4
$\text{trRosetta}_{\text{base}}$	45.7	58.4	69.6	50.9	64.6	75.5
$\text{trRosetta}_{\text{full}}$	51.8	66.6	80.1	53.2	67.1	77.5
MSA Transformer*	44.8	59.7	72.5	43.5	55.9	66.8
MSA Transformer†	54.5	**70.0**	**80.2**	53.6	68.4	78.0
MSA Transformer‡	**54.6**	68.4	77.5	**55.8**	**69.8**	**79.1**

distribution. To test this hypothesis we train a model with half the embedding size (384 instead of 768) resulting in 30M total parameters. The resulting model achieves a Top-L long-range precision of 52.8 after 100k updates, 3.5 points lower than the baseline model. This suggests that model size is still an important factor in contact precision, although also shows that a model with fewer than 30M parameters can still outperform 650M and 3B parameter single-sequence models.

Masking Pattern

We consider two strategies for applying BERT masking to the MSA: uniform and column. Uniform masking applies masking uniformly at random across the MSA. Column masking always masks full columns of the MSA. This makes the training objective substantially more difficult since the model cannot look within a column of an MSA for information about masked tokens. We find that column masking is significantly worse (by almost 20 points) than uniform masking.

Block Ordering

Row attention followed by column attention slightly outperforms column attention followed by row attention.

Tied Attention

We consider three strategies for row attention: untied, mean normalization, and square root normalization (see Section 4.3). We find that tied attention substantially outperforms untied attention and that square root normalization outperforms mean normalization.

Masking Percentage

As the MSA Transformer has more context than single sequence models, its training objective is substantially easier than that of single sequence models. Therefore, we explore whether increasing the masking percentage (and thereby increasing task difficulty) would improve the model. However, we do not find a statistically significant difference between masking 15% or 20% of the positions. Therefore, we use a masking percentage of 15% in all other studies for consistency with ESM-1b and previous masked language models.

MSA Positional Embedding

An MSA is an unordered set of sequences. However, due to the tools used to construct MSAs, there may be some pattern to the ordering of sequences in the MSA. We therefore examine the use of a learned MSA positional embedding in addition to the existing learned sequence positional embedding. The positional embedding for a sequence is then a learned function of

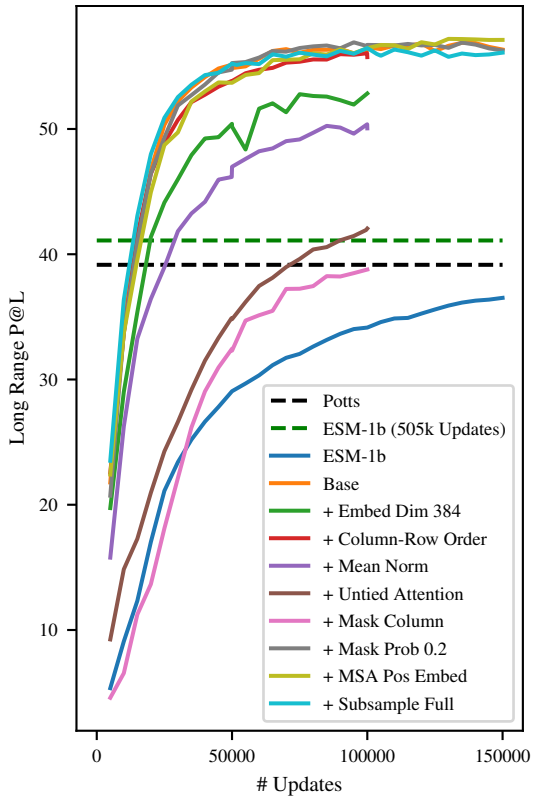


Figure C.3: Training curves for MSA Transformer with different hyperparameters. See Section 4.4 for a description of each hyperparameter searched over. ESM-1b training curve, ESM-1b final performance (after 505k updates), and average Potts performance are included as dashed lines for comparison.

its position in the input MSA (not in the full MSA). Subsampled sequences in the input MSA are sorted according to their relative ordering in the full MSA. We find that the inclusion of an MSA positional embedding does modestly increase model performance, and therefore include it in our final model.

Subsample Strategy

At training time we explore two subsampling strategies. The first strategy is adapted from Yang et al. [133]: we sample the number of output sequences from a log-uniform distribution, with a maximum of N/L sequences to avoid exceeding the maximum tokens we are able to fit in GPU memory. Then, we sample that number of sequences uniformly from the MSA, ensuring that the reference sequence is always chosen. In the second strategy, we always sample the full N/L sequences from the MSA. In our hyperparameter search, most models use the first strategy, while our final model uses the second. We find no statistically

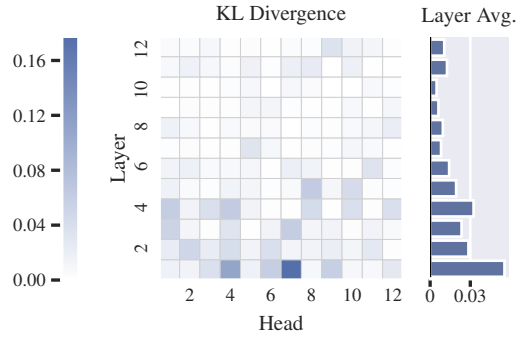


Figure C.4: KL Divergence between distribution of row attention across amino acids and background distribution of amino acids. The fraction of attention on an amino acid k is defined as the average over the dataset of $a_i^{lh}\mathbb{1}\{x_i == k\}$, where x_i is a particular token in the input MSA and a^{lh} is the attention in a particular layer and head. KL Divergence is large for early layers but decreases in later layers.

significant difference in performance between the two strategies. However, it is possible that the log-uniform strategy would help prevent overfitting and ultimately perform better after more training.

The CCMpred implementation of Potts [73, 74], UniRep [25], SeqVec [42], TAPE transformer (see Chapter 2), ESM-1b [43], ProtBERT-BFD, and ProTrans-T5 [94] are used as supervised contact prediction comparisons. In Table C.4 we show the complete results for long-range precision over the CASP-13 FM targets and CAMEO-hard domains referenced in [133]. All baseline models are trained for 200 epochs with a batch size of 4.

C.4 Attention to Amino Acids

Vig et al. [7] examine the distribution of amino acids attended to by single-sequence models. The attention in single-sequence models is roughly equivalent to the row-attention in our model, but there is no column-attention analogue. We therefore examine the distribution of amino acids attended to by the column attention heads. In Fig. C.4 we show the KL-divergence between the distribution of attention across amino acids and the background distribution of amino acids. The divergence is large for earlier layers in the model but decreases in later layers, suggesting the model stops focusing on the amino acid identities in favor of focusing on other properties.

C.5 Sequence Weights

Sequence reweighting is a common technique used for fitting Potts models which helps to compensate for data bias in MSAs [86]. Informally, sequence reweighting downweights groups

of highly similar sequences to prevent them from having as large of an effect on the model. The sequence weight w_i is defined as,

$$w_i = \left(1 + \sum_{j \neq i} \mathbb{1}\left\{d_{\text{hamming}}(x_i, x_j) < 0.2\right\}\right)^{-1} \tag{C.1}$$

where x_i, x_j are the i-th and j-th sequences in the MSA, d_{hamming} is the hamming distance between two sequences normalized by sequence length, and w_i is the sequence weight of the i-th sequence.

Appendix D

Language models enable zero-shot prediction of the effects of mutations on protein function

D.1 Extraction methods

ESM-1v is pre-trained to output the probability for each possible amino acid at a masked position. We explore four methods of scoring the effects of mutations using the model:

- **Masked marginal**: Probabilities are extracted according to the mask noise during pre-training. At each position, we introduce a mask token and record the model's predicted probabilities of the tokens at that position.
- **Mutant marginal**: Probabilities are extracted according to the random token noise during pre-training. Among the 15% predicted positions in the sequence during pre-training, 10% of those are randomly mutated and 10% retain their original identities. The model is tasked to predict the correct token at those positions. Therefore, in this extraction method, we follow the pre-training methodology by passing in mutated tokens and recording the model's probability that they are correct.
- **Wildtype marginal**: We perform a single forward pass using the wildtype sequence. This method enables fast scoring as just a single forward pass is used.
- **Pseudolikelihood**: This method is proposed in the literature for scoring with masked language models [189].

In all cases, we assume an additive model when multiple mutations are present in a sequence. Results are summarized in Tables D.3 and D.5.

Let x^{mt} and x^{wt} represent the mutant and wildtype sequences. We refer to x_{-i} as the sequence x with a mask introduced at position i. We refer to the set of mutations that are

Table D.1: Zero-shot learning is a natural extension of the various approaches that have been used for mutational effect prediction to date. Rather than training a new model for every task, a single general purpose model is trained and can be directly applied across multiple tasks. The approach is fully unsupervised, no information from experimental measurements of function is used.

Approach	**Formal setting**	**Task level supervision**	**Representative Methods**
Supervised mutation prediction	Supervised	Direct supervision from experimental measurements	Reviewed in [44]
Model trained on sequences from individual family	Unsupervised	Weak positive supervision from MSA	[1, 2]
Fine-tuning on experimental data	Semi-supervised transfer	Supervision from experimental measurements	[25, 43, 129]
Fine-tuning on MSA	Transfer learning with weak-positive supervision	Weak positive supervision from MSA	Introduced for transfer learning in [25]
Direct forward pass	Zero-shot learning	None	This work

introduced as the set M. For example, if mutations are introduced at positions 3 and 6, then $M = \{3, 6\}$.

Masked marginal probability (L forward passes) This method performs best among the four. We introduce masks at the mutated positions and compute the score for a mutation by considering its probability relative to the wildtype amino acid (Strategy a):

$$\sum_{i \in M} \log p(x_i = x_i^{mt} | x_{-M}) - \log p(x_i = x_i^{wt} | x_{-M})$$

This formulation assumes an additive model, consistent with the training objective. We show that this assumption is justified empirically by evaluating the model with different choices at the non-mutated positions. First, the wildtype sequence (Strategy b):

$$\sum_{i \in M} \log p(x_i = x_i^{mt} | x_{-i}^{mt}) - \log p(x_i = x_i^{wt} | x_{-i}^{wt})$$

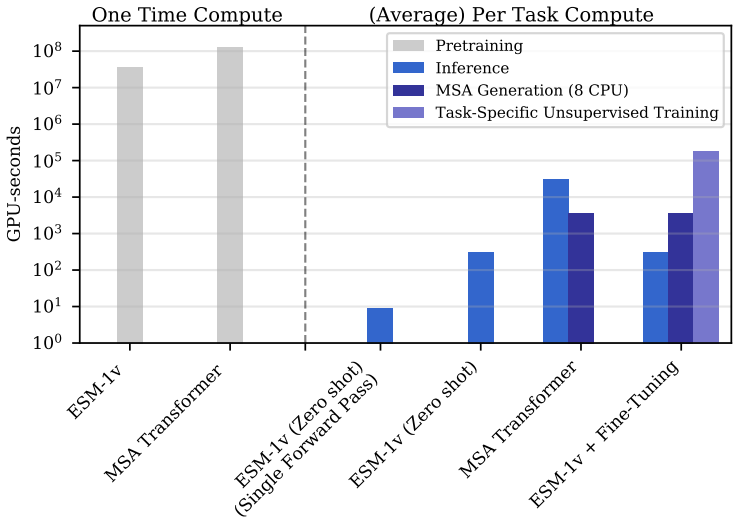


Figure D.1: Compute requirements in GPU-seconds for (left) pre-training and (right) average task. With open-sourced pre-trained models, end users bypass the pre-training phase and only incur inference costs. ESM-1v and MSA Transformer amortize compute cost into a single expensive pre-training run. After pre-training, inference is fast. On average, it takes < 10 seconds to label a deep mutational scan from Riesselman et al. [2] with ESM-1v (Zero-shot, Single Forward Pass). Performance improves marginally with the more expensive scoring scheme (Table D.3).

Table D.2: Average |Spearman ρ| on the single-mutation validation set after training a 650M parameter Transformer model for 170,000 updates on various sequence identity clusterings of Uniref.

Clustering	**| Spearman ρ |**
30%	0.456
50%	0.537
70%	0.552
90%	**0.564**
100%	0.458

and the mutant sequence (Strategy c):

$$\sum_{i \in M} \log p(x_i = x_i^{mt} | x_{-i}^{mt}) - \log p(x_i = x_i^{wt} | x_{-i}^{mt})$$

Strategy (a), where we mask all positions at the same time, performs best on the PABP Yeast Doubles validation dataset (Table D.5).

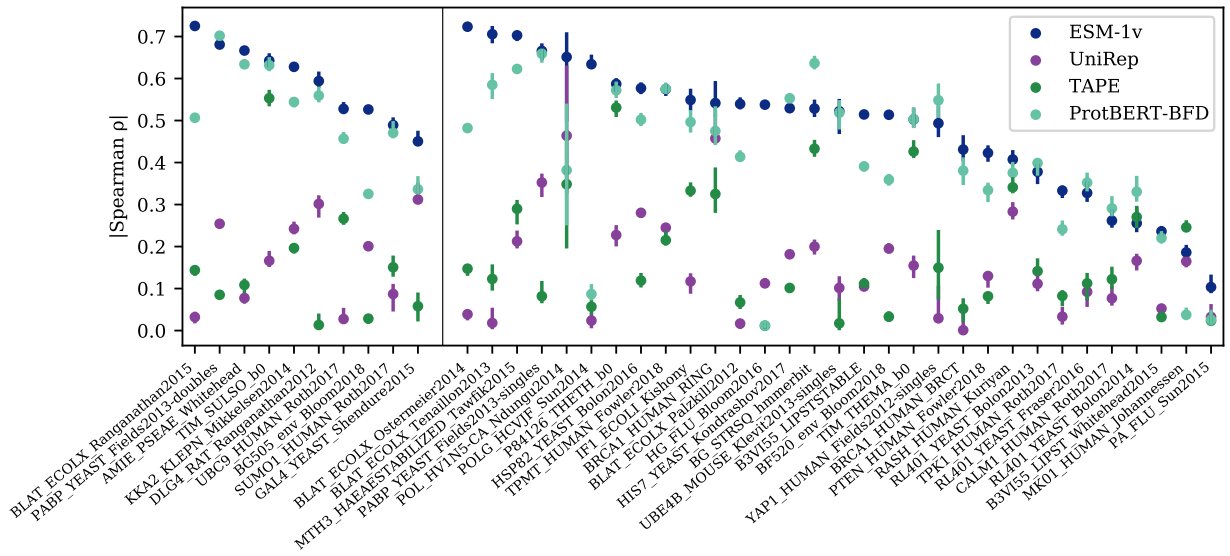


Figure D.2: Zero-shot performance of ESM-1v compared to earlier protein language models on all 41 deep mutational scans. Points are |Spearman ρ| on each dataset, error bars show standard deviation of 20 bootstrapped samples. Validation proteins are shown to the left of the dividing line and test proteins to the right. ESM-1v is the best performing method on 30 of the 41 deep mutational scans.

Method	**| Spearman ρ |**
Masked marginal	**0.582**
Mutant marginal	0.578
Wildtype marginal	0.572
Pseudo-likelihood	0.552

Table D.3: Benchmarking scoring schemes on the single-mutation validation set. The means across the validation set are listed. The masked marginal scheme performs best.

Mutant marginal probability This method is analogous to the wildtype marginal probability, except we use the mutant sequence instead.

$$\sum_{i \in M} [\log p(x_i = x_i^{mt} | x^{mt}) - \log p(x_i = x_i^{wt} | x^{mt})]$$

This method requires a single forward pass for every mutation.

Wildtype marginal probability (1 forward pass) In the fastest scheme, we perform a single forward pass using the wildtype sequence as input. For a set of mutations at positions

Input	Consensus columns only	| Spearman ρ |
MSA seed	Yes	0.573
MSA seed	No	0.567
Uniprot	N/A	**0.582**

Table D.4: ESM-1v performs better when including the full protein sequence as listed in Uniprot, compared to using the seed sequence of the MSA corresponding to the deep mutational scan. Results on single-mutation validation set. The means across the validation set are listed. We experiment with a number of strategies for inference: (i) the consensus columns only; (ii) the aligned part of the query sequence; and (iii) the complete Uniprot sequence. The complete Uniprot sequence performs best, possibly because the model was pre-trained on complete Uniprot sequences. We use the MSA seed sequence from the MSAs released by [2] corresponding to the deep mutational scans.

Method	| Spearman ρ |
Masked marginal (a)	**0.692**
Masked marginal (b)	0.482
Masked marginal (c)	0.483
Mutant marginal	0.694
Wildtype marginal	0.672
Pseudo-likelihood	0.608

Table D.5: Ablating scoring schemes on the PABP Yeast Doubles dataset. The masked marginal scheme performs best when masking all mutated sites together. Mean absolute Spearman ρ across the single-mutation validation tasks is reported.

M, the score is:

$$\sum_{i \in M} [\log p(x_i = x_i^{mt} | x^{wt}) - \log p(x_i = x_i^{wt} | x^{wt})]$$

We find that the method performs well with a minor 1% decrease in absolute performance, while requiring very limited computational resources. The strong performance indicates that the masked language modeling objective causes the model to capture the fitness landscape of the protein in its outputs.

Pseudolikelihood Psuedolikelihood has been proposed in the literature as a method to score sequences using masked language models [189]. We compute the score as follows:

$$\sum_i \log p(x_i = x_i^{mt} | x_{-i}^{mt}) - \log p(x_i = x_i^{wt} | x_{-i}^{wt})$$

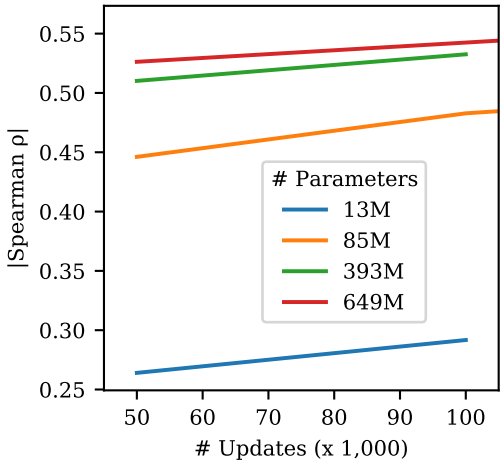


Figure D.3: Larger models perform better on variant prediction. We trained four models of various scales, following the hyperparameters listed in Henighan et al. [6]. Results on single-mutation validation set.

MSA Subsample Strategy	**Context size**	**| Spearman ρ |**
Diversity minimizing	256 sequences	0.255
Random	256 sequences	0.535 $\pm$ 0.024
HHFilter	256 sequences	0.550 $\pm$ 0.015
Sequence reweighting	256 sequences	0.578 $\pm$ 0.005

Table D.6: Subsampling strategies for MSA Transformer evaluated on the single-mutation validation set. Sequence reweighting performs best. When sampling methods are stochastic, 5 seeds are run and the mean and standard deviation is reported. With HHFilter, we run with the `-diff M` parameter and randomly subsample the output if more than `M` sequences are returned. We use a coverage parameter of 75 and a sequence identity parameter of 99. Mean absolute Spearman ρ across the single-mutation validation tasks is reported.

As mutation prediction is a ranking task and as the contribution from the second term is constant throughout the deep mutation scan (i.e. the wildtype sequence is always the same), we can safely drop it from the computation.

Evaluation

We compare the methods described above on the validation set, finding that the masked marginal scheme performs best. To determine the specific mode of inference when multiple mutations are present, we examine each method on the "doubles" component of the PABP

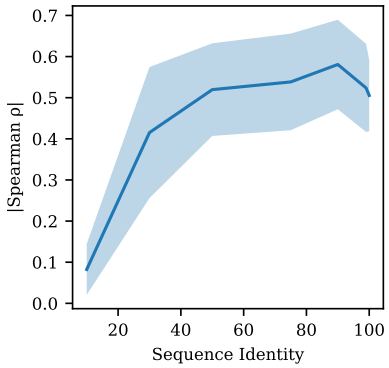


Figure D.4: Filtering sequences with high sequence identity to the query improves performance. The curve illustrates mean ± standard deviation across the 9 validation proteins. HHFilter is used to filter the MSAs with coverage of 75 and various sequence identity values as shown on x-axis. After filtering, 384 sequences are sampled for inference. Each sequence identity value s refers to using sequences with no more than s% sequence identity to the seed sequence. The MSA Transformer appears to primarily use sequences that are close to the seed sequence, yet performance drops if sequences that are *too similar* remain in the MSA. Results are broken down across the single-mutation validation set in Table D.7.

Yeast dataset finding the masked marginal (a) strategy performs best. This scoring method is used across the results.

Evaluating ESM-1v on subsequences

DeepSequence, EVMutation, and the MSA Transformer use the consensus columns of a MSA as input. We construct MSAs using the seed sequences from the DeepSequence paper, which usually correspond to a subsequence of the protein capturing the domain where the deep mutational scan was performed.

Table D.4 explores using the MSA seed sequence vs. the full Uniprot sequence for inference on the validation set. We find that the full Uniprot sequence performs best, possibly because the model was pre-trained on Uniprot sequences. We note in Figure Fig. 5.6 that the model captures some bias in the Uniprot dataset, for example that most proteins begin with a methionine (corresponding to the start codon).

D.2 Unsupervised fine-tuning ESM-1v

Experimental setup We assess a number of approaches for fine-tuning ESM-1v on task-specific MSAs. We evaluate modeling decisions by fine-tuning on tasks from the validation set and examining the mean change in Spearman ρ over the course of training. For efficiency, we compute Spearman ρ using the wildtype marginal strategy, as this requires just a single

Table D.7: Filtering MSAs from the single-mutation validation set with HHFilter coverage 75 and various sequence identity values. Filtering sequences to an identity threshold of 75% or 90% consistently performs best. The Spearman rank correlation between MSA Transformer predictions and experimental data is shown for each deep mutational scan.

Sequence Identity (%)	**10**	**30**	**50**	**75**	**90**	**99**	**100**
AMIE_PSEAE_Whitehead	0.025	0.461	0.467	0.365	**0.665**	0.654	0.622
BG505_env_Bloom2018	0.055	0.055	0.417	**0.482**	0.452	0.450	0.457
BLAT_ECOLX_Ranganathan2015	0.060	0.630	0.745	0.776	**0.795**	0.662	0.478
DLG4_RAT_Ranganathan2012	0.008	0.431	0.416	0.431	**0.457**	0.418	0.400
GAL4_YEAST_Shendure2015	0.080	0.287	0.366	0.441	**0.576**	0.542	0.388
SUMO1_HUMAN_Roth2017	0.131	0.430	0.516	**0.541**	0.500	0.492	0.495
TIM_SULSO_b0	0.026	0.581	0.633	0.625	**0.649**	0.324	0.632
UBC9_HUMAN_Roth2017	0.165	0.376	0.557	**0.583**	0.494	0.555	0.475
KKA2_KLEPN_Mikkelsen2014	0.192	0.484	0.560	0.601	**0.637**	0.616	0.602

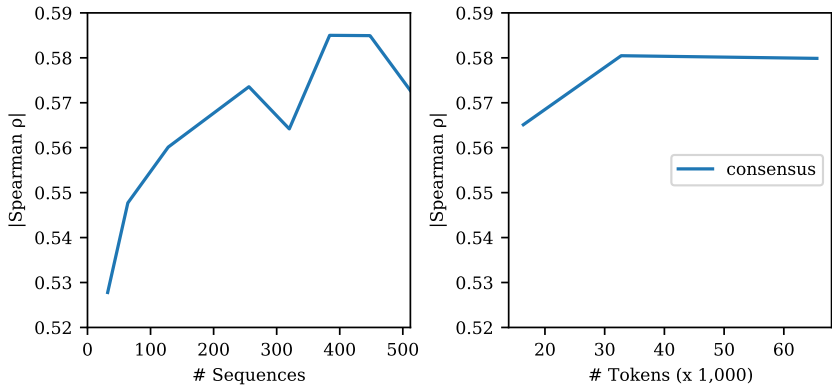


Figure D.5: Few-shot performance of the MSA Transformer is robust to the number of sequences used for inference. **Left:** Varying the number of sequences used in inference. **Right:** Varying the number of tokens used for inference. Since the number of sequences in each MSA varies, we assess the effect of fixing the total number of tokens sampled from each MSA and drawing the corresponding number of sequences to fill the context. Results on single-mutation validation set.

forward pass. After the final modeling decisions are selected, we train all models for 7500 updates and evaluate on all proteins using the masked marginals strategy. All models in this section were trained with a constant learning rate of 10^{-5} using the masked language modeling objective. For reference, the ESM-1v pre-training was performed with a target batch size of 1M tokens.

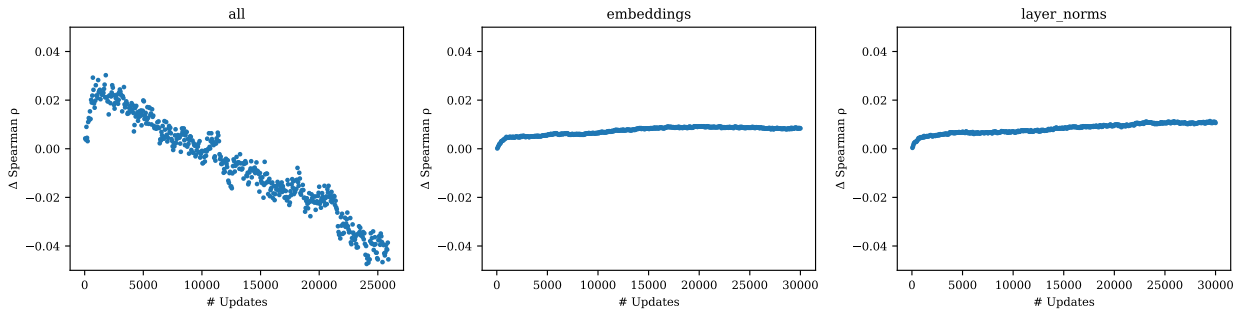


Figure D.6: Unsupervised fine-tuning baselines. Mean change in Spearman ρ across 9 models trained on the single-mutation validation set tasks. The title of each plot denotes the parameters that are trained. We find that fine-tuning the entire model results in overfitting, but limiting the training to just the embeddings or just the layer norms does not improve performance with respect to the pre-trained initialization. The choice of gap token and label smoothing has limited effect.

Unsupervised fine-tuning baselines The concept of unsupervised fine-tuning of an MSA has been previously proposed [25, 129]. Fig. D.6 studies a basic fine-tuning setup on the consensus columns of the MSA. Each model is fine-tuned on a single MSA with a target batch size of 8192 tokens. We first observe that that models overfit quickly if the entire model is trained. This results in a decrease in Spearman ρ compared to initialization. As the fine-tuning is performed on the consensus columns of the MSA, we sought to regularize the model by fine-tuning only the embeddings. As a PSSM already captures information relevant to the task, we hypothesize that tuning the embeddings could capture similar information and boost performance. Similarly, we experiment with tuning only the layer normalizations, as these have also been recently shown to enable transfer to new tasks. In both cases, we found no improvement to the average Spearman ρ. We also assessed label smoothing and replacing the gap token with a mask token or a pad token finding no significant impact; for simplicity, we omit label smoothing and use the mask token for future experiments.

Minimal models We also examine a set of minimal models, in which we freeze all parameters in the Transformer and learn a projection from the ESM-1v outputs onto a PSSM, taking the sum of the projection and the PSSM. We experiment with freezing the PSSM or allowing it to train. We did not see a change in Spearman ρ of more than 0.01.

Spiked unsupervised fine-tuning Next, we examine a new strategy, which we call `spiked fine-tuning`. In spiked fine-tuning, we regularize the fine-tuning by continuing to spike pre-training sequences into the fine-tuning batch. In this setting, we train on the entire MSA, including non-consensus positions. We find that spiked fine-tuning with a small ratio (0.01)

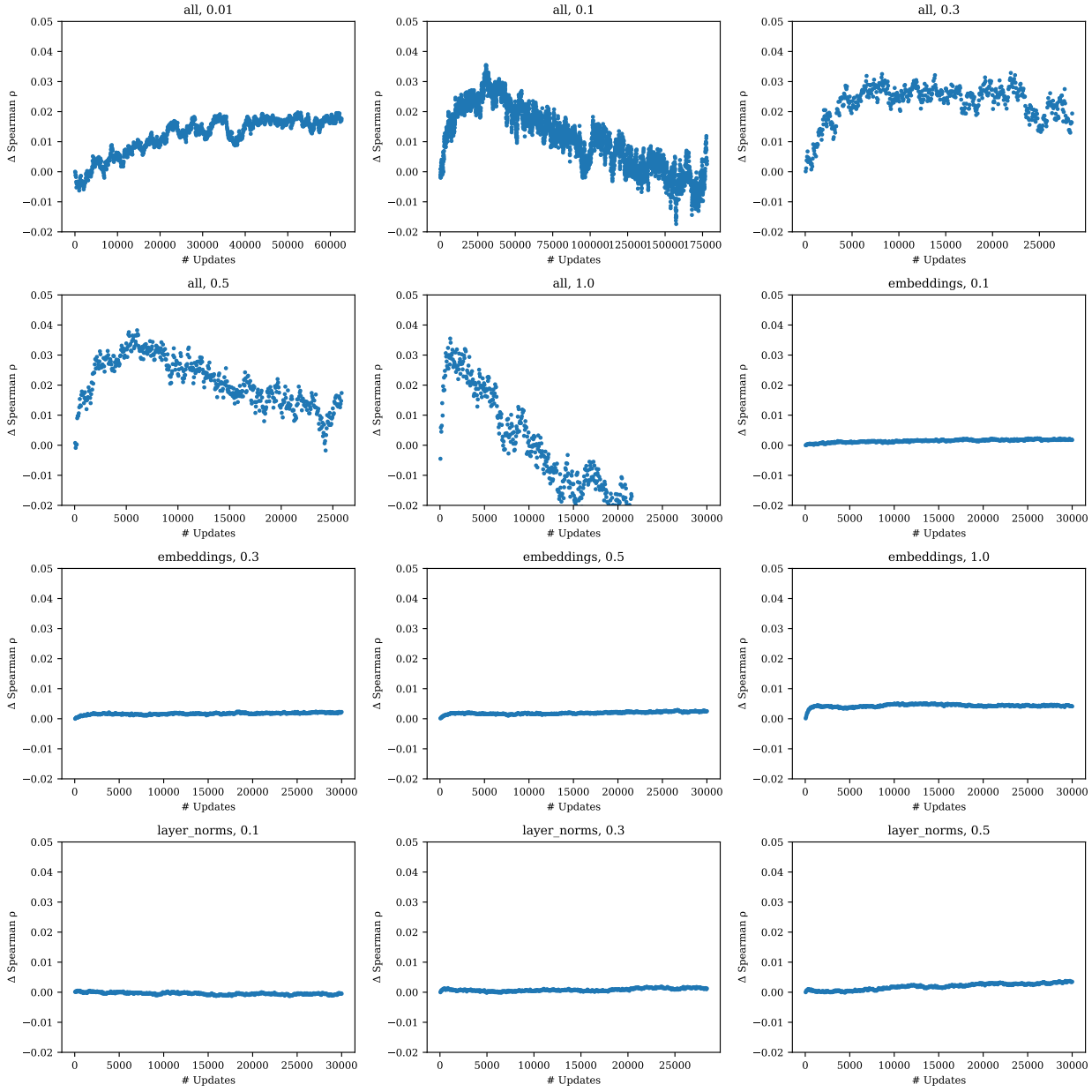


Figure D.7: Spiked unsupervised fine-tuning. Mean change in Spearman ρ across 9 models trained on the single-mutation validation tasks. The title of each plot denotes the parameters that are trained; and the ratio of MSA tokens to pre-training tokens. We find that a small ratio performs well and reduces the tendency for the model to overfit, while preserving strong performance. Performance is not improved if the fine-tuning is limited to just the embeddings or just the layer norms.

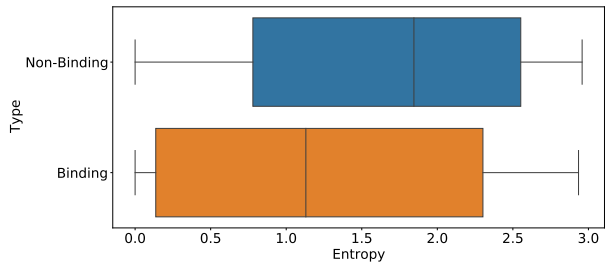


Figure D.8: Box plot comparing entropy scores for binding vs non-binding positions in structures labeled in the Provis validation dataset (as described in Appendix B.4 of [7]). A Welch's t-test determines that the difference between the two means is statistically significant ($p < 0.01$).

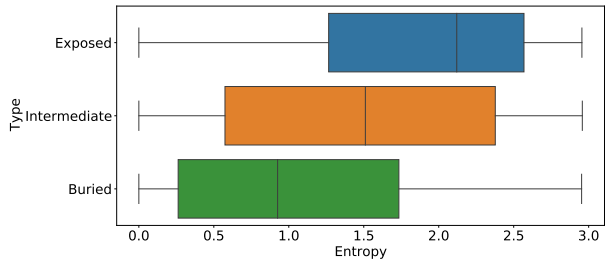


Figure D.9: Box plot comparing entropy scores across residue depths in structures from the trRosetta dataset. Residue depths are categorized based on the number of neighboring residues with C-beta distance<10 angstroms. (exposed≤16, buried≥24 [8]). A one way Anova test determines that the differences between all three means are statistically significant ($p < 0.01$).

of MSA tokens to pre-training tokens performs best and enables training of all parameters without overfitting.

The final models were trained for 7500 updates using spiked fine-tuning with a batch size of 500k tokens. To produce an ensemble, we perform the fine-tuning scheme on five models that were pre-trained with different seeds. Each model was also fine-tuned with a unique seed.

D.3 Datasets

Evaluation Tasks

We evaluate models on a set of 41 deep mutational scans collected by Riesselman et al. [2], which comprise a variety of tasks assessing a diverse set of proteins. Across tasks, the

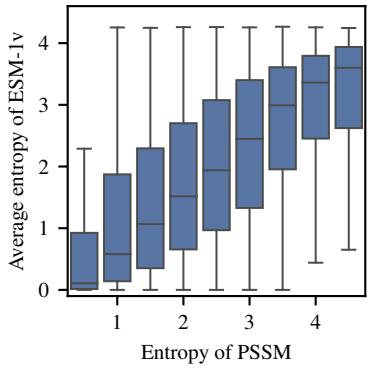


Figure D.10: Entropy of PSSM versus ESM-1v predicted entropy on the trRosetta dataset. PSSM entropy determines the level of conservation at a given position in a protein family. ESM-1v entropy is well correlated with PSSM entropy (Pearson's $r = 0.44$), suggesting the model is able to identify conserved positions.

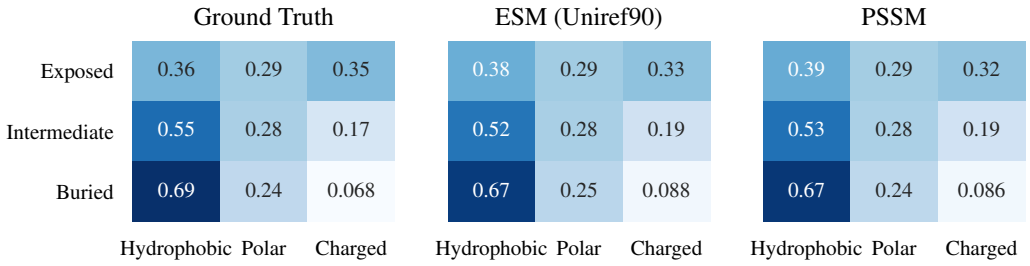


Figure D.11: Predicted distribution of hydrophobic, polar and charged amino acids at the surface and core of proteins in the trRosetta dataset. We compare to the actual proportion in the protein structure. We classify residues into buried, intermediate or exposed by residue depths based on the number of neighboring residues with C-beta distance < 10 angstroms (exposed ≤ 16, buried ≥ 24) [8]. ESM-1v and PSSM both see increased hydrophobicity predictions for buried residues, in correspondence with the ground truth data. Predicted probabilities are produced by introducing a mask token at each position.

experimental data differ widely in the functions tested and in the experimental measurements performed. Of the 41 datasets, 37 are single-mutation only, 1 is double-mutation only, and the rest contain a variable number of mutations per sequence between 1 and 28. The median number of mutations is 2979, and the average is 16822; the smallest dataset has 37 mutations, and the largest has 496137. We randomly select 9 single-mutation experiments as a validation set. We also ablate the multiple mutation scoring approach on the double mutations from the PABP Yeast deep mutational scan. We exclude the 10 tasks used for validation and

ablations from the test set. These datasets are reported in results for the full set. While the original compilation has 43 datasets, we exclude the tRNA (which is not a protein) and the toxin-antitoxin complex (which comprises multiple proteins).

We treat each deep mutational scanning dataset as a separate prediction task, scoring each of the variants in the dataset with the model. We evaluate performance by comparing the scores with the experimental measurements using Spearman rank correlation. Results are broken out between the test set, which excludes the validation set, as well as the full set of 41 datasets. All ablations are performed on the single mutant validation set or the PABP Yeast doubles experiment. Only the final models are evaluated on the test set.

Pre-training datasets

For the clustering sweep in Fig. 5.4, we use the Uniref50 and Uniref90 databases from the `2020_03` release of Uniref [71], a publicly available database of proteins, clustered respectively to 50% and 90% sequence identity. For the 30% sequence identity dataset, we use Uniclust30 `2020_03` [116]. For the 70% sequence identity dataset, we Uniref100 is hierarchically clustered to the 90%, then 70% sequence identity level. MMseqs settings are those used by Uniref: 80% overlap with longest sequnece in the cluster, which translates to `mmseqs-cluster -min-seq-id 90,70 -cov-mode 0 -alignment-mode 3 -c 0.8`. In order to compute pretraining perplexities on a heldout validation set, we randomly select 1% of sequences each from Uniref30, Uniref50, and Uniref90. We then exclude sequences that are similar to the validation sequences by removing all sequences found with MMSeqs search (`-min-seq-id 0.xx`) for validation set xx. We use the most sensitive settings in MMSeqs `-alignment-mode 3 -max-seqs 300 -s 7`, taking the train set as the query database and the validation set as the target database. We use settings `-c 0.8 -cov-mode 0` to match the settings of Uniref. Pretraining perplexities on the validation sets are reported in Table D.8.

Baselines

The MSAs used for training DeepSequence and EVMutation are generated from the 2017-10 version of Uniref100, whereas the models we study are trained on sequences from Uniref90 2020-03. In the case of MSA Transformer, the model is pre-trained on the 2018-03 Uniref, but we use 2020-03 MSAs for inference. In order to provide a fair comparison, we regenerate MSAs against the 2020-03 Uniref according to the methodology in Hopf et al. [1] and retrain EVMutation (replication) and DeepSequence (replication) on these datasets using their open-source codebases. For the viral proteins BF520_env_Bloom2018, BG505_env_Bloom2018, HG_FLU_Bloom2016, PA_FLU_Sun2015, POLG_HCVJF_Sun2014, POL_HV1N5-CA_Ndungu2014, we compute the sequence weights with $\theta = 0.01$ (versus default $\theta = 0.2$) following Riesselman et al. [2]. In the replication of the DeepSequence ensemble, for BF520_env_Bloom2018, BG505_env_Bloom2018, one of the five runs failed so we reran with a different random seed.

Validation and test set

The single-mutation validation set consists of the following deep mutational scans:

- AMIE_PSEAE_Whitehead
- BG505_env_Bloom2018
- BLAT_ECOLX_Ranganathan2015
- BRCA1_HUMAN_RING
- DLG4_RAT_Ranganathan2012
- GAL4_YEAST_Shendure2015
- POLG_HCVJF_Sun2014
- SUMO1_HUMAN_Roth2017
- TIM_SULSO_b0
- UBC9_HUMAN_Roth2017
- KKA2_KLEPN_Mikkelsen2014

For ablations studies with multiple mutations the following dataset is used:

- PABP_YEAST_Fields2013-doubles

The test set consists of the following deep mutational scans:

- B3VI55_LIPSTSTABLE
- B3VI55_LIPST_Whitehead2015
- BF520_env_Bloom2018
- BG_STRSQ_hmmerbit
- BLAT_ECOLX_Ostermeier2014
- BLAT_ECOLX_Palzkill2012
- BLAT_ECOLX_Tenaillon2013
- BRCA1_HUMAN_BRCT
- CALM1_HUMAN_Roth2017
- HG_FLU_Bloom2016
- HIS7_YEAST_Kondrashov2017
- HSP82_YEAST_Bolon2016
- IF1_ECOLI_Kishony
- MK01_HUMAN_Johannessen
- MTH3_HAEAESTABILIZED_Tawfik2015
- P84126_THETH_b0
- PABP_YEAST_Fields2013-singles
- PA_FLU_Sun2015
- POL_HV1N5-CA_Ndungu2014
- PTEN_HUMAN_Fowler2018
- RASH_HUMAN_Kuriyan
- RL401_YEAST_Bolon2013
- RL401_YEAST_Bolon2014
- RL401_YEAST_Fraser2016
- TIM_THEMA_b0
- TPK1_HUMAN_Roth2017
- TPMT_HUMAN_Fowler2018
- UBE4B_MOUSE_Klevit2013-singles
- YAP1_HUMAN_Fields2012-singles

D.4 Methodology

Model selection

ESM-1b and MSA Transformer model checkpoints are selected based on performance on the single mutation validation set. Open sourced checkpoints are used for ESM-1b and other protein language model baselines.

Treatment of synonymous mutations

Synonymous mutations are mutations in DNA that do not change the protein sequence that is expressed. The deep mutational scanning datasets that we evaluate here can therefore include DNA mutations that do not change the protein sequence itself. Synonymous mutations are excluded from results.

Bootstraps

To compute bootstraps for the pointplots, we randomly resample each deep mutational scan (with replacement) and compute the Spearman ρ between the experimental data and model predictions.

Average calibration error

The standard expected calibration error (ECE) performs poorly for highly imbalanced data [190]. Following Neumann et al. [190] and Nixon et al. [191] we adapt average calibration error for the multi-class setting as follows:

$$\frac{1}{K}\sum_{k=1}^{K}\frac{1}{B_k^+}\sum_{b=1}^{B_k^+}|\texttt{acc}(b,k)-\texttt{conf}(b,k)|$$

where K is the number of classes, B_k^+ is the number of non-empty bins for class k, and `acc` and `conf` are the accuracy and confidence for bin b and class k.

D.5 Performance by MSA depth

We examine the relationship between the number of related sequences in the pre-training set and performance on the task. We use Jackhmmer [10] version 3.3.1 with a bitscore threshold of 27 and 8 iterations to construct MSAs from the ESM-1v training set. We do not observe a strong correlation between MSA depth and the observed absolute value of Spearman ρ (Figure Fig. D.13).

D.6 Compute costs

ESM-1v models are pre-trained for 6 days on 64 V100 GPUs. Weights for the MSA Transformer were retrieved from the open-source repository released by the authors; the model was pre-trained for 13 days on 128 V100 GPUs. Once trained, the models can be used directly for function prediction tasks. Forward inference is efficient, meaning that for applications of the models, the additional compute is minimal. In total, five ESM-1v models were trained on various Uniref clustering thresholds to five different levels: 30%, 50%, 70%, 90%, and

100%. For the 90% sequence identity level, five total models with different random seeds were trained, for use in an ensemble. As illustrated in Fig. D.1, inference is inexpensive by comparison. Batch inference was performed with preemptible, short (shorter than one hour), single V100 GPU jobs on a shared compute cluster.

D.7 Licenses

We note the following licenses for datasets used in our work:

- 41 deep mutational scans from Riesselman et al. [2]: License unclear
- Sequences and binding-site information from Vig et al. [7]: BSD3 License
- PDB structures from Protein Data Bank [181]: free of all copyright restrictions and is made fully and freely available for both non-commercial and commercial use.
- 15K sequences and structural information from Yang et al. [133]: License unclear, as the code uses MIT license but dataset has no license listed. The dataset is a derivative of the PDB [181].

Table D.8: Perplexities on heldout pre-training validation sequences after training a 650M parameter Transformer model for 170,000 updates on various sequence identity clusterings of Uniref.

Clustering	**Valid (30%)**	**Valid (50%)**	**Valid (90%)**
30%	8.93	8.33	7.29
50%	8.90	7.77	6.27
70%	9.05	7.80	5.85
90%	9.37	8.10	5.56
100%	9.89	8.65	6.05

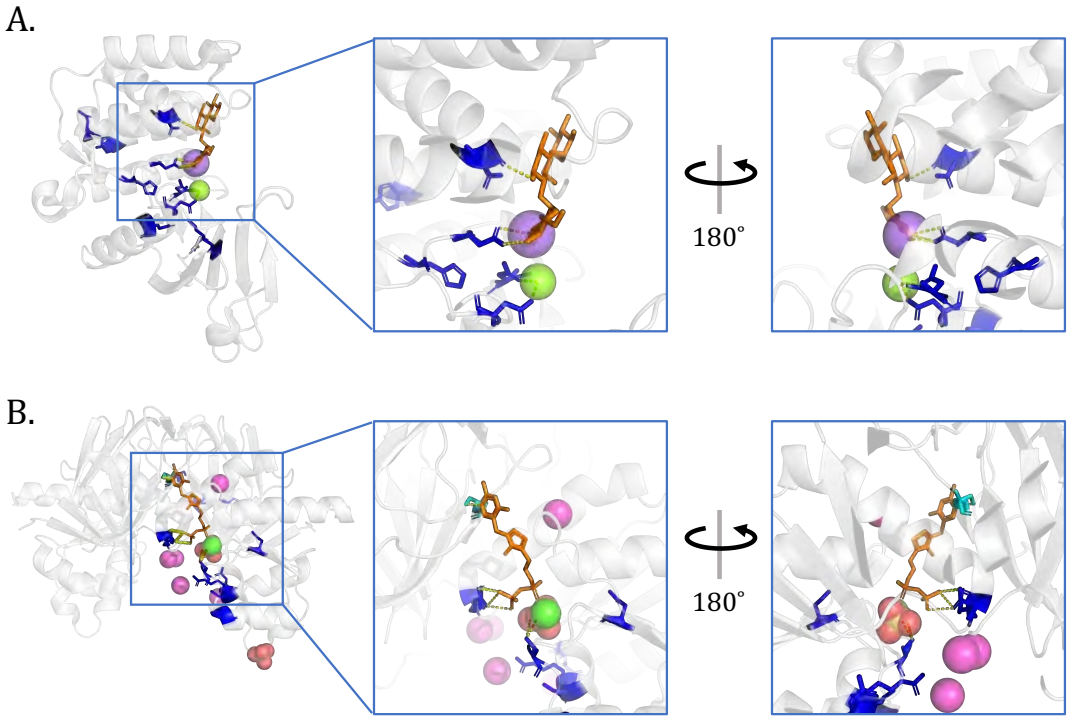


Figure D.12: ESM-1v accurately captures functional properties. Further examples. The ten positions with lowest predicted entropy highlighted in blue. **(A)** Kanamycin kinase APH(3')-II (pdbid: 1ND4 [9]). The highlighted residues interact with the kanamycin aminoglycoside, as well as the magnesium and sodium ions. **(B)** Thiamin pyrophosphokinase 1 (pdbid: 3S4Y). Residue 216 is one of the 10 lowest entropy residues, and we highlight it on the other chain (in cyan) to show both chains of the dimer interacting with the thiamine diphosphate.

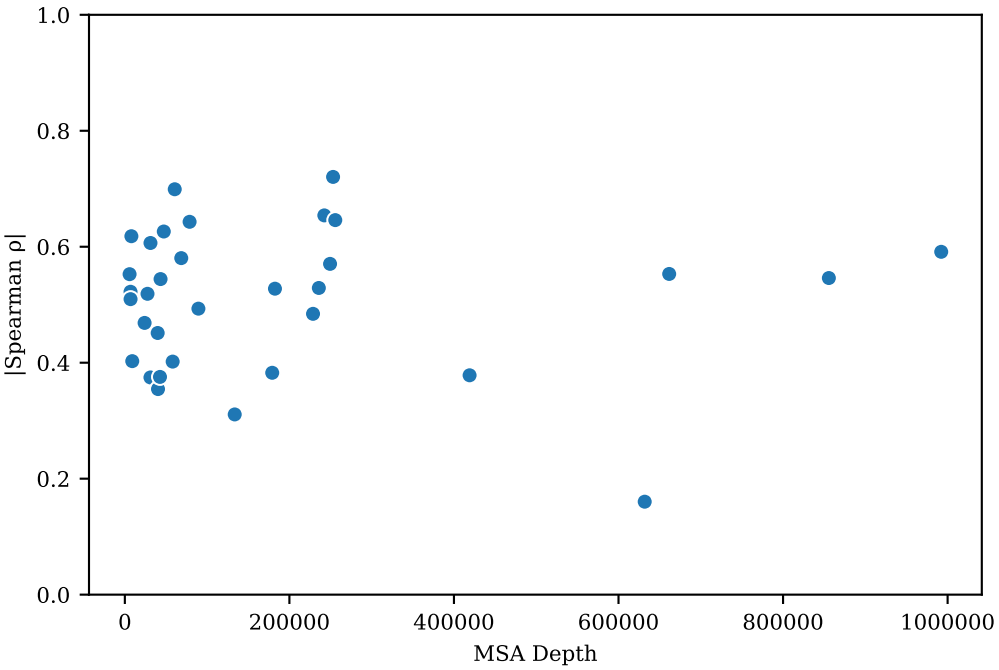


Figure D.13: Relation between MSA depth and zero-shot performance of ESM-1v. We use JackHMMer [10] version 3.3.1 with a bitscore threshold of 27 and 8 iterations to construct MSAs from the ESM-1v training set. We do not observe a strong correlation between MSA depth and the observed |Spearman ρ|.