BIOLOGICAL STRUCTURE AND FUNCTION EMERGE FROM SCALING UNSUPERVISED LEARNING TO 250 MILLION PROTEIN SEQUENCES

Alexander Rives [*] [†] [‡] **Siddharth Goyal** [*] [§] **Joshua Meier** [*] [§] **Demi Guo** [*] [§]
Myle Ott [§] **C. Lawrence Zitnick** [§] **Jerry Ma** [†] [§] **Rob Fergus** [†] [‡] [§]

ABSTRACT

In the field of artificial intelligence, a combination of scale in data and model capacity enabled by unsupervised learning has led to major advances in representation learning and statistical generation. In biology, the anticipated growth of sequencing promises unprecedented data on natural sequence diversity. Learning the natural distribution of evolutionary protein sequence variation is a logical step toward predictive and generative modeling for biology. To this end we use unsupervised learning to train a deep contextual language model on 86 billion amino acids across 250 million sequences spanning evolutionary diversity. The resulting model maps raw sequences to representations of biological properties without labels or prior domain knowledge. The learned representation space organizes sequences at multiple levels of biological granularity from the biochemical to proteomic levels. Unsupervised learning recovers information about protein structure: secondary structure and residue-residue contacts can be identified by linear projections from the learned representations. Training language models on full sequence diversity rather than individual protein families increases recoverable information about secondary structure. The unsupervised models can be adapted with supervision from quantitative mutagenesis data to predict variant activity. Predictions from sequences alone are comparable to results from a state-of-the-art model of mutational effects that uses evolutionary and structurally derived features.

1 INTRODUCTION

The size of biological sequence datasets is experiencing approximately exponential growth resulting from reductions in the cost of sequencing technology. This data, which is sampled across diverse natural sequences, provides a promising enabler for studying predictive and generative techniques for biology using artificial intelligence. We investigate scaling high-capacity neural networks to this data to extract general and transferable information about proteins from raw sequences.

The idea that biological function and structure are recorded in the statistics of protein sequences selected through evolution has a long history (Yanofsky et al., 1964; Altschuh et al., 1987; 1988). Out of the possible random perturbations to a sequence, evolution is biased toward selecting those that are consistent with fitness (Göbel et al., 1994). The unobserved variables that determine a protein's fitness, such as structure, function, and stability, leave a record in the distribution of observed natural sequences (Göbel et al., 1994).

Unlocking the information encoded in evolutionary protein sequence variation is a longstanding problem in biology. An analogous problem in the field of artificial intelligence is natural language understanding, where the distributional hypothesis posits that a word's semantics can be derived from the contexts in which it appears (Harris, 1954).

Recently, techniques based on self-supervision, a form of unsupervised learning in which context within the text is used to predict missing words, have been shown to materialize representations of word meaning that can generalize across problems (Collobert & Weston, 2008; Dai & Le, 2015; Peters et al., 2018; Devlin et al., 2018). The ability to learn such representations improves significantly with larger training datasets (Baevski et al., 2019; Radford et al., 2019). Can similar self-supervision techniques leveraging the growth of sequence datasets learn useful and general protein sequence representations?

In this paper, we apply self-supervision in representation learning to the problem of understanding protein sequences and explore what information can be learned. Similar to approaches used to model language, we train a neural network representation by predicting masked amino acids. For data, we use the sequences contained within the Uniparc database (The UniProt Consortium, 2007), the largest sampling of protein sequences available, spanning a wide range of evolutionary diversity. The dataset contains 250M protein sequences with 86 billion amino acids. We provide preliminary investigations

[*] Equal contribution

[†] Correspondence to <arives@cs.nyu.edu>, <maj@fb.com>, and <robfergus@fb.com>

[‡] Dept. of Computer Science, New York University, USA

[§] Facebook AI Research, USA

into the organization of the internal representations, and look for the presence of information about biological structure and activity. We also study generalizability of the representations to new problems. [1]

2 Background

Efficient sequence search methods have been the computational foundation for extracting biological information from sequences (Altschul et al., 1990; Altschul & Koonin, 1998; Eddy, 1998; Remmert et al., 2011). Search across large databases of evolutionary diversity assembles related sequences into multiple sequence alignments (MSAs). Within families, mutational patterns convey information about functional sites, stability, tertiary contacts, binding, and other properties (Altschuh et al., 1987; 1988; Göbel et al., 1994). Conserved sites correlate with functional and structural importance (Altschuh et al., 1987). Local biochemical and structural contexts are reflected in preferences for distinct classes of amino acids (Levitt, 1978). Covarying mutations have been associated with function, tertiary contacts, and binding (Göbel et al., 1994).

The prospect of inferring biological structure and function from evolutionary statistics has motivated development of machine learning on individual sequence families. Raw covariance, correlation, and mutual information have confounding effects from indirect couplings (Weigt et al., 2009). Maximum entropy methods disentangle direct interactions from indirect interactions by inferring parameters of a posited generating distribution for the sequence family (Weigt et al., 2009; Marks et al., 2011; Morcos et al., 2011; Jones et al., 2011; Balakrishnan et al., 2011; Ekeberg et al., 2013b). The generative picture can be extended to include latent variables parameterized by neural networks that capture higher order interactions than pairwise (Riesselman et al., 2018).

Recently, self-supervision has emerged as a core direction in artificial intelligence research. Unlike supervised learning which requires manual annotation of each datapoint, self-supervised methods use unlabeled datasets and thus can exploit far larger amounts of data, such as unlabeled sequence data. Self-supervised learning uses proxy tasks for training, such as predicting the next word in a sentence given all previous words (Bengio et al., 2003; Dai & Le, 2015; Peters et al., 2018; Radford et al., 2018; 2019) or predicting words that have been masked from their context (Devlin et al., 2018; Mikolov et al., 2013a).

Increasing the dataset size and the model capacity has shown improvements in the learned representations. In recent work, self-supervision methods used in conjunction with large data and high-capacity models produced new state-of-the-art results approaching human performance on various question answering and semantic reasoning benchmarks (Devlin et al., 2018), and coherent natural text generation (Radford et al., 2019).

This work explores self-supervised language models that have demonstrated state-of-the-art performance on a range of text processing tasks, applying them to protein data in the form of raw amino acid sequences. Significant differences exist between the two domains, so it is unclear whether the approach will be effective in this new domain. Compared to natural language, amino acid sequences are long and use a small vocabulary of twenty elements (25 accounting for rarely occurring tokens in large datasets); accordingly the modeling problem is more similar to character-level language models (Mikolov et al., 2012; Kim et al., 2016) than traditional word-level models. In this work we explore whether self-supervised language models are able to discover knowledge about biology in evolutionary data.

3 Scaling language models to 250 Million diverse protein sequences

Large protein sequence databases contain a rich sampling of evolutionary sequence diversity. We explore scaling high-capacity language models to protein sequences using the largest available sequence database. In our experiments we train on 250 million sequences of the Uniparc database (The UniProt Consortium, 2007) which has 86 billion amino acids, excluding a held-out validation set of 1 million sequences. This data is comparable in size to the large text datasets that are being used to train high-capacity neural network architectures on natural langauge (Devlin et al., 2018; Radford et al., 2019).

The amount of available evolutionary protein sequence data and its projected growth make it important to find approaches that can scale to this data. As in language modeling, self-supervision is a natural choice for learning on the large unlabeled data of protein sequences. To fully leverage this data, neural network architectures must be found that have sufficient capacity and the right inductive biases to represent the immense diversity of sequence variation in the data.

We investigate the Transformer (Vaswani et al., 2017), which has emerged as a powerful general-purpose model architecture for representation learning and generative modeling of textual data, outperforming more traditionally employed recurrent and convolutional architectures, such as LSTM and GConv networks (Hochreiter & Schmidhuber, 1997; Dauphin et al., 2017). We use a deep bidirectional Transformer (Devlin et al., 2018) with the raw character sequences of amino acids from the proteins as input. The Transformer model processes inputs through a series of blocks that alternate self-attention with feedforward connections. Self-attention allows the network to build up complex representations that incorporate context from across the sequence. The inductive bias of self-attention has interesting parallels to parametric distributions that have

[1] A formal discussion of the experimental methodology used to study generalization can be found in Appendix A.2.

Model	**Parameterization**	**# Params**	**ECE**
Baselines	Random	0	25
	0-gram	25	17.92
n-gram	Unidirectional ($n = 9$)	758.3M	10.12
	Bidirectional ($n = 10$)	865.1M	10.17
Transformer (full data)	12 layer	85.1M	6.02
	24 layer	170.2M	5.63
	36 layer	708.6M	**4.31**
Transformer (limited data)	36 layer (10% data)	708.6M	4.54
	36 layer (1% data)	708.6M	5.34
	36 layer (0.1% data)	708.6M	10.99

Table 1: Exponentiated cross-entropy (ECE) values for various amino acid language models trained on the unfiltered Uniparc dataset. Lower values correspond to better modeling, with a value of 1 implying perfect modeling of the target data. Dataset sizes correspond to the Uniparc pre-training corpus. For baselines and n-gram models, the ECE metric corresponds to perplexity. All models are validated on a held-out Uniparc validation set. [4]

been developed to detect amino acid covariation in multiple sequence alignments (Ekeberg et al., 2013a). These methods operate by modeling long-range pairwise dependencies between amino acids. Self-attention explicitly constructs interactions between all positions in the sequence, which could allow it to directly model residue-residue dependencies.

We use a variant of a self-supervision objective that is effective in language modeling (Devlin et al., 2018). We train by noising a fraction of amino acids in the input sequence and have the network predict the true amino acid at the noised positions from the complete sequence context. Performance is reported as the average exponentiated cross entropy (ECE) of the model's predicted probabilities with the true amino acid identities. ECE describes the mean uncertainty of the model among its set of options for every prediction: an ECE of 1 implies the model predicts perfectly, while an ECE of 25 indicates a completely random prediction. [2]

To understand how well the Transformer models fit the data, we establish a baseline using n-gram frequency models. These models estimate the probability of a sequence via an auto-regressive factorization, where the probability of each element is estimated based on the frequency that the element appears in the context of the preceding (or following) $n - 1$ elements. We fit n-gram models with unidirectional (left-to-right) and bidirectional context across a wide range of context lengths (up to $n = 10^4$) and different levels of Laplace smoothing on a random subset of 3M sequences sampled from the full Uniparc dataset. We find the best such unidirectional model attains a validation ECE of 10.1 at context size $n = 9$ with Laplace smoothing $\alpha = 0.01$. ECE values of the best unidirectional and bidirectional n-gram models are nearly the same.

We train Transformer models of various sizes on the full training dataset of 249M sequences and smaller subsets of 10% and 1% of the training data, as well as on the MSA data of individual sequence families. [3] We find an expected relationship between the capacity of the network (measured in number of parameters) and its ability to accurately predict the noised inputs. The largest models we study are able to achieve an ECE of 4.31 in predicting noised tokens (Table 1). We also observed an inverse relation between the amount of training data and best ECE. However, even the largest models we trained containing over 700M parameters are not able to overfit the full training dataset.

4 Multi-scale organization in sequence representations

The network is trained to predict amino acids masked from sequences across evolutionary diversity. To do well at this prediction task, the learned representations must encode the underlying factors that influence sequence variation in the data. Variation in large sequence databases is influenced by processes at many scales, including properties that affect fitness directly, such as activity, stability, structure, binding, and other properties under selection (Hormoz, 2013; Hopf et al., 2017) as well as by contributions from phylogenetic bias (Gabaldon, 2007), experimental and selection biases (Wang et al., 2019; Overbaugh & Bangham, 2001), and sources of noise such as random genetic drift (Kondrashov et al., 2003).

[2] In traditional autogressive language modeling, ECE corresponds to the commonly-used perplexity metric.

[3] Details on the model architecture, data, and training hyperparameters are presented in Appendix A.4.

[4] Methodology for train/validation/test splits for all experiments are summarized in Appendix A.2, and additional details can be found throughout the Appendix A in the sections describing specific experiments.

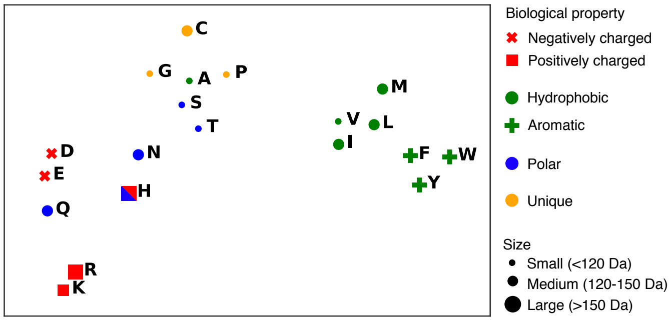


Figure 1: The trained Transformer model encodes amino acid properties in the model's output embeddings, visualized here with t-SNE. Amino acids cluster in representation space according to biochemistry and molecular weight. The polar amino acid Glutamine (Q) which is structurally similar to both the positively charged Arginine (R) and negatively charged Glutamic Acid (E) lies approximately halfway between the two in the representation space. Histidine (H), polar or positively charged depending on context, can be clustered with either category.

We investigate the learned representation space of the network at multiple scales including at the level of individual amino acids, protein families, and proteomes to look for signatures of biological organization.

The contextual language models we study contain inductive biases that already impart structure to representations prior to learning. Furthermore, a basic level of intrinsic organization is expected in the sequence data itself as a result of biases in amino acid composition. To disentangle the contribution of learning from the inductive bias of the models and frequency biases in the data, we compare against (1) untrained contextual language models: an untrained LSTM, and the Transformer at its initial state prior to training; and (2) a frequency baseline that maps a sequence to a vector of normalized amino acid counts.

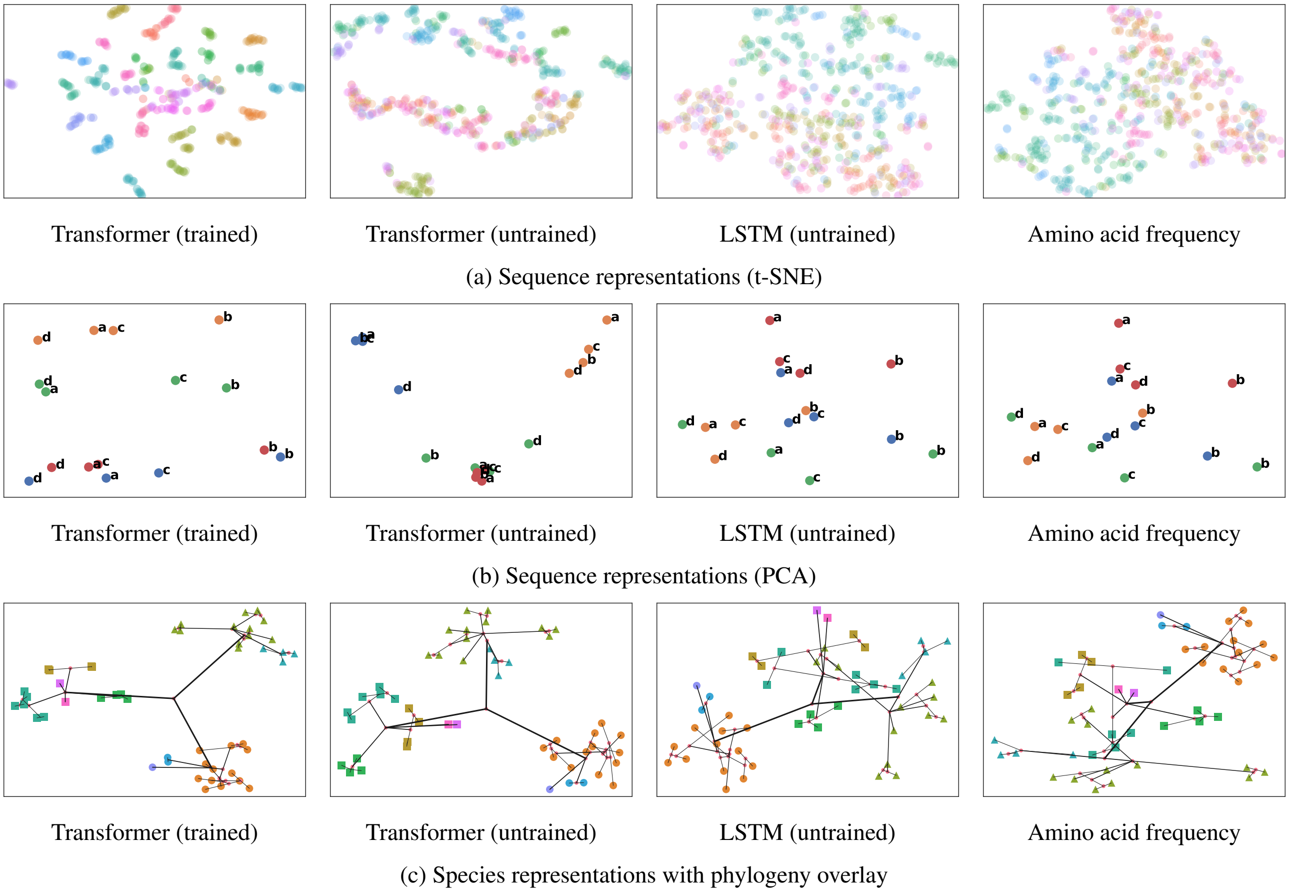
4.1 Learning encodes biochemical properties

The neural network represents the identity of each amino acid in its input and output embeddings. The input embeddings project the input amino acid tokens into the first Transformer block. The output embeddings project the final hidden representations back to logarithmic probabilities. Within given structural and functional contexts, amino acids are more or less interchangeable depending on their biochemical properties (Hormoz, 2013). Self-supervision might capture these dependencies to build a representation space that reflects biochemical knowledge.

To investigate if the network has learned to encode biochemistry in representations, we project the weight matrix of the final embedding layer of the network into two dimensions with t-SNE (Maaten & Hinton, 2008) and present visualizations in Figure 1. Visual inspection reveals that the representation of amino acids in the network's embeddings is organized by biochemical properties. Hydrophobic and polar residues are clustered separately, including a tight grouping of the aromatic amino acids Tryptophan (W), Tyrosine (Y), and Phenylalanine (F). The negatively charged amino acids Aspartic Acid (D) and Glutamic Acid (E), as well as the positively charged Arginine (R) and Lysine (K) are respectively paired. Histidine (H), charged or polar depending on its environment, is grouped with the polar residues. The polar amino acid Glutamine (Q) which is structurally similar to both the positively charged Arginine and negatively charged Glutamic Acid lies approximately halfway between the two in the representation space. The three smallest molecular weight amino acids Glycine (G), Alanine (A), and Serine (S) are grouped, as are Hydroxyl-containing Serine (S) and Threonine (T). Cysteine is a relative outlier which might relate to its ability to form disulfide bonds. In this analysis, we omit the infrequently appearing tokens: B (Asparagine), U (Selenocysteine), Z (Glutamine), O (Pyrrolysine) as well as X (the unknown token). We plot representations of the remaining 20 amino acids.

4.2 Embeddings of orthologous proteins

The final hidden representation output by the network is a sequence of vectors, one for each position in the sequence. An embedding of the complete sequence, a vector summary that is invariant to sequence length, can be produced by averaging features across the full length of the sequence. These embeddings represent sequences as points in high dimensional space. Each sequence is represented as a single point and similar sequences are mapped to nearby points.


Transformer (trained)
Transformer (untrained)
LSTM (untrained)
Amino acid frequency

(a) Sequence representations (t-SNE)

Transformer (trained)
Transformer (untrained)
LSTM (untrained)
Amino acid frequency

(b) Sequence representations (PCA)

Transformer (trained)
Transformer (untrained)
LSTM (untrained)
Amino acid frequency

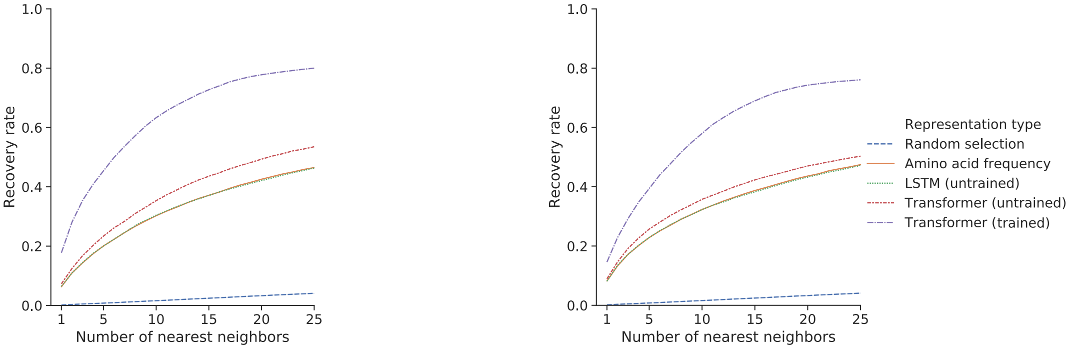
(c) Species representations with phylogeny overlay

Figure 2: Protein sequence representations encode and organize orthological, species, and phylogentic information of sequences. (a) visualizes representations of several orthologous groups from different species via t-SNE. Sequences from the same orthologous group have the same color. The trained Transformer representations cluster by orthology more densely than do baseline representations. (b) visualizes ortholog sequence representations via PCA (a linear projection technique). Character label denotes species and genes are colored by orthologous group. The trained Transformer representations self-organize along a species axis (horizontal) and orthology axis (vertical), in contrast to baseline representations. (c) visualizes per-species averaged sequence representations via a learned hierarchical distance projection; a phylogenetic hierarchy (colors represent distinct phyla) is overlaid. The trained Transformer provides more phylogenetically structured representations than do baselines.

We investigate organization of the space of protein embeddings using sets of proteins related by orthology. Orthologous genes are corresponding genes between species that have evolved from a common ancestor by speciation. Generally, these genes will have different sequences but retain the same function (Huerta-Cepas et al., 2018).

Similarity-based spatial organization of orthologous genes Proteins that are related by orthology share a high level of sequence similarity by virtue of their common ancestry. Intrinsic organization in orthology data might be expected from bias in the amino acid compositions of proteins from different species. This similarity is likely to be captured by inductive biases of untrained language models even prior to learning, as well as by the frequency baseline. We find in Figure 2a that the baseline representations show some organization by orthology. Clustering in the LSTM representations and frequency baseline are similar. The untrained Transformer is more clustered than the other baselines. The trained Transformer representations show tight clustering of orthologous genes implying that learning has identified and represented their similarity.

Principal components and directional encoding We hypothesize that trained models encode similar representations of orthologous genes by encoding information about both orthology and species by *direction* in the representation vector space. We apply principal component analysis (PCA), to recover principal directions of variation in the representations. If information about species and orthology specify the two main directions of variation in the high dimensional data then they will be recovered in the first two principal components. We select 4 orthologous genes across 4 species to look for directions of variation. Figure 2b indicates that linear dimensionality reduction on trained-model sequence representations



(a) Recovery rates under group translation

(b) Recovery rate under species translation

Figure 3: Learned sequence representations can be translated between orthologous groups and species. Depicted is the recovery rate of nearest-neighbor search under (a) orthologous group translation, and (b) species translation; in both settings, the trained Transformer representation space has a higher recovery rate.

recovers species and orthology as the major axes of variation, with the primary (horizontal) axis corresponding to species and the secondary (vertical) axis corresponding to ortholog family. While the baseline representations are often able to cluster sequences by ortholog family, they fail to encode orthology or species information in their orientation.

We note that the principal components of the untrained LSTM representations are oriented almost identically to those of the amino acid frequency representations, suggesting that the LSTM's inductive bias primarily encodes frequency information. In contrast, the untrained Transformer representation yields a tighter clustering by ortholog family, indicating that the inductive biases of the Transformer model capture higher-order information about sequences even without any learning.

Phylogenetic organization Just as averaging per-residue representations produces protein embeddings, averaging per-protein embeddings for a subset of the proteins of a species produces species representations. We qualitatively examine the extent to which these representations correspond to phylogeny.

Reusing the set of orthologous groups from above, we generate species representations for 2609 species and subsequently learn a phylogenetic distance projection on the representations along the lines of Hewitt & Manning (2019) [5]. Figure 2c, which depicts the projected representations of 48 randomly-selected species evenly split across the three taxonomic domains, suggests that the Transformer species representations cohere strongly with natural phylogenetic organization, in contrast to the LSTM and frequency baselines.

4.3 Translating between proteins in representation space

To quantitatively investigate the learned structural similarities explored above, we assess nearest neighbor recovery under vector similarity queries in the representation space. If biological properties are encoded along independent directions in representation space, then corresponding proteins with a unique biological variation are related by linear vector arithmetic.

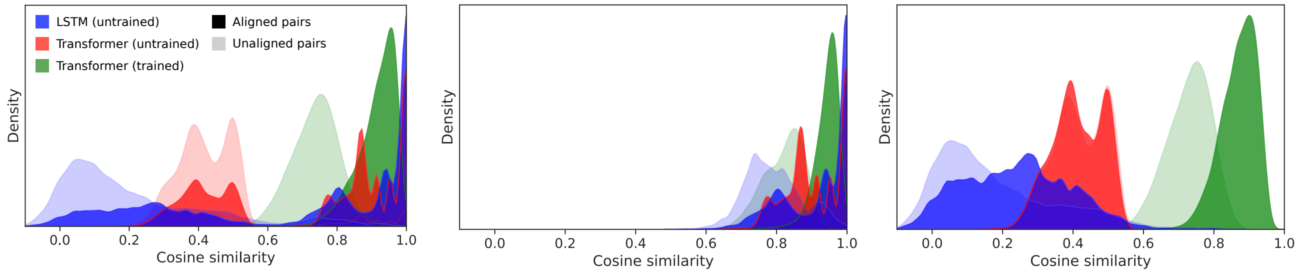
We cast the task of measuring the similarity structure of the representation space as a search problem. To perform a translation in representation space, we compute the average representations of proteins belonging to the source and target sets. Then, the difference between these averages defines a direction in representation space, which can be used for vector arithmetic. [6] After applying a vector translation to the source, a fast nearest neighbor lookup in the vector space can identify the nearest proteins to the query point. The rate of recovering the target of the query in the nearest neighbor set quantifies directional regularity. We report the *top-k recovery rate*, the probability that the query target recovers the target protein among its top-k nearest neighbors.

We study two kinds of protein translations in vector space: (1) translations between orthologous groups within the same species and (2) translations between different species within the same orthologous group. Intuitively, if the representation space linearly encodes orthology and species information, then we expect to recover the corresponding proteins with high probability.

Translation between different orthologs of the same species To translate between protein a and protein b of the same species, we define the source and target sets as the average of protein a or protein b across all 24 diverse species. If

[5] See Appendix A.7 for full details of the learned projection.

[6] See Appendix A.8 for full details of the sequence representation translations.



(a) Overall

(b) Identical amino acid pairs

(c) Distinct amino acid pairs

Figure 4: Per-residue representations from trained models implicitly align sequences. Depicted are distributions of cosine similarity of representations of sequences from within PFAM family PF01010. Note that (a) is an additive composition of (b) and (c). The differences between dark green and light green distributions imply that the trained Transformer representations are a powerful discriminator between aligned and unaligned positions, especially when compared to the baseline representations.

Representation type	Overall	Identical amino acid pairs	Distinct amino acid pairs
Transformer (trained)	**0.841**	**0.870**	**0.792**
Transformer (untrained)	0.656	0.588	0.468
LSTM (untrained)	0.710	0.729	0.581

Table 2: Area under the ROC curve (AUC) of per-residue representational cosine similarities in distinguishing between aligned and unaligned pairs of residues within a Pfam family. Results displayed are averaged across 128 families.

representation space linearly encodes orthology, then adding the difference in these averages to protein a of some species will recover protein b in the same species. Figure 3a shows recovery rates for the trained model and baselines for randomly selected proteins and pairs of orthologous groups.

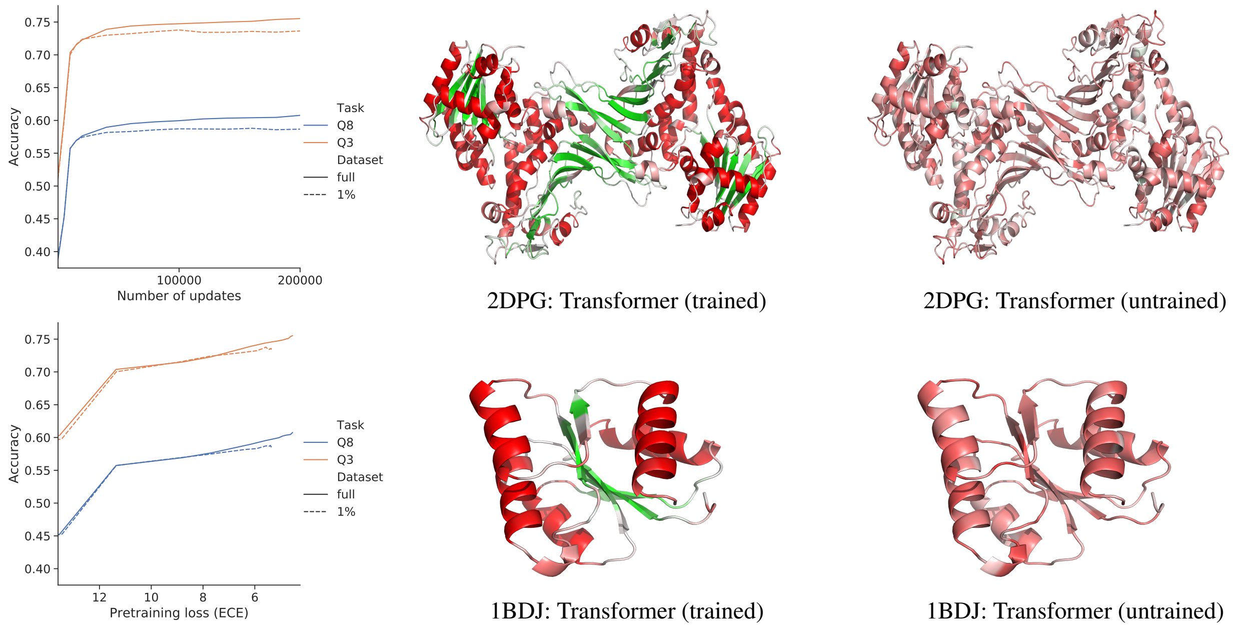
Translation between corresponding orthologous genes in different species We use an analogous approach to translate a protein of a source species s to its ortholog in the target species t. Here, we consider the average representation of the proteins in s and in t. If representation space is organized linearly by species, then adding the difference in average representations to a protein in species s will recover the corresponding protein in species t. Figure 3b shows recovery for species translations, finding comparable results to orthologous group translation.

In both translation experiments, representations from the trained Transformer have the highest recovery rates across all values of k, followed by the untrained Transformer baseline. The recovery rate approaches 80% for a reasonable $k = 20$ (Figure 3b), indicating that information about orthology and species is loosely encoded directionally in the structure of the representation space. The recovery rates for the untrained LSTM and amino acid frequency baselines are nearly identical, supporting the observation that the inductive bias of the LSTM encodes frequency information. Figure 2b visualizes the projected representations of four orthologous groups across four species.

4.4 Learning encodes sequence alignment features

Multiple sequence alignments (MSAs) identify corresponding sites across a family of related sequences (Ekeberg et al., 2013a). These correspondences give a picture of the variation at different sites within the sequence family. The neural network receives as input *single* sequences and is given no access to family information except via learning. We investigate the possibility that the contextual per-residue representations learned by the network encode information about the family.

One way family information could appear in the network is through assignment of similar representations to positions in different sequences that are aligned in the family's MSA. Using the collection of MSAs of structurally related sequences in Pfam (Bateman et al., 2013), we compare the distribution of cosine similarities of representations between pairs of residues that are aligned in the family's MSA to a background distribution of cosine similarities between unaligned pairs of residues. A large difference between the aligned and unaligned distributions implies that self-supervision has developed contextual representations that use shared features for related sites within all the sequences of the family.



2DPG: Transformer (trained)

2DPG: Transformer (untrained)

1BDJ: Transformer (trained)

1BDJ: Transformer (untrained)

(a)

(b)

Figure 5: Information about secondary structure can be extracted by supervised linear projections from the pre-trained representation. (a) displays top-1 secondary structure prediction test accuracy for projections of the final hidden representation of the Transformer over the course of pre-training, with the top graph depicting accuracy as a function of model pre-training update steps, and the bottom graph depicting accuracy as a function of pre-training loss. (b) illustrates the 3-class projections on two example proteins (PDB IDs 2DPG and 1BDJ; Cosgrove et al., 1998; Kato et al., 1999), having maximum sequence identity of 0.274 and 0.314, respectively, with training sequences. Red denotes a prediction of helix, green denotes a prediction of strand, and white denotes a prediction of coil. Color intensities indicate prediction confidence.

Figure 4a depicts the distribution of cosine similarity values between aligned and unaligned positions within a representative family for the trained model and baselines. A marked shift between aligned and unaligned pairs results from learning. Visually, trained Transformer representations effectively discriminate between aligned and unaligned positions, while the untrained Transformer model is unable to separate the distributions. The untrained LSTM representations show a slight separation of the distributions. We observe in Figure 4b and Figure 4c that these trends hold even under the constraints that the residue pairs (1) share the same amino acid identity or (2) have different amino acid identities.

We estimate differences between the aligned and unaligned distributions across 128 Pfam families using the area under the ROC curve (AUC) as a metric of discriminative power between aligned and unaligned pairs. Average AUC is shown for each representation type in (Table 2). The aggregate results demonstrate a shift in the discrimination power across the families from low for the untrained Transformer to high for the trained Transformer after learning. These results support the hypothesis that self-supervision is capable of encoding contextual features found by MSA.

5 Emergence of secondary structure and tertiary contacts

5.1 Secondary structure

Structural information likely constitutes part of a minimal code for predicting contextual dependencies in sequence variation. Information about compatibility of a sequence with secondary structure is present in sequence variation, implying that self-supervision may be able to compress variation in its training data through secondary structure features. We use secondary structure prediction accuracy of a linear projection of per-residue representations as a proxy to measure secondary structure information discovered in the course of learning.

We examine final hidden representations for the presence of information about secondary structure at different stages of training. To establish a baseline, we compare to (1) one-hot amino acid representations, which results in the prior that predicts the most likely label for each position's amino acid, and (2) representations from family-level frequencies summarized in position-specific scoring matrices ("PSSM"s; Jones, 1999) for each protein. The untrained model contains some information

Representation type	PF00005	PF00069	PF00072	CB513
Amino acid identity	0.516	0.506	0.536	0.488
LSTM (untrained)	0.728	0.671	0.712	0.544
12-layer Transformer (untrained)	0.818	0.719	0.835	0.519
12-layer Transformer (PF00005)	0.864	0.725	0.842	0.565
12-layer Transformer (PF00069)	0.816	0.842	0.850	0.602
12-layer Transformer (PF00072)	0.789	0.688	0.888	0.544
12-layer Transformer (full dataset)	0.900	0.872	**0.906**	0.731
36-layer Transformer (full dataset)	**0.902**	**0.884**	0.902	**0.755**

Table 3: Top-1 test accuracy on three class secondary structure prediction by linear projection of per-amino-acid vector representations. Evaluations are reported for three PFAM families (PF00005: ATP-binding domain of the ABC transporters; PF00069: Protein kinase domain; PF00072: Response regulator receiver domain), as well as on the CB513 evaluation dataset. For Transformer models, the pre-training dataset is indicated in parentheses. Underline indicates the representations trained on the same Pfam family used for evaluation. Comparisons are relative to the family (columnwise), since each of the families differ in difficulty. Representations learned on single families perform well for predictions within the family, but do not generalize as well to sequences outside the family. Representations trained on the full data of Uniparc sequences outperform the single-family representations in all cases.

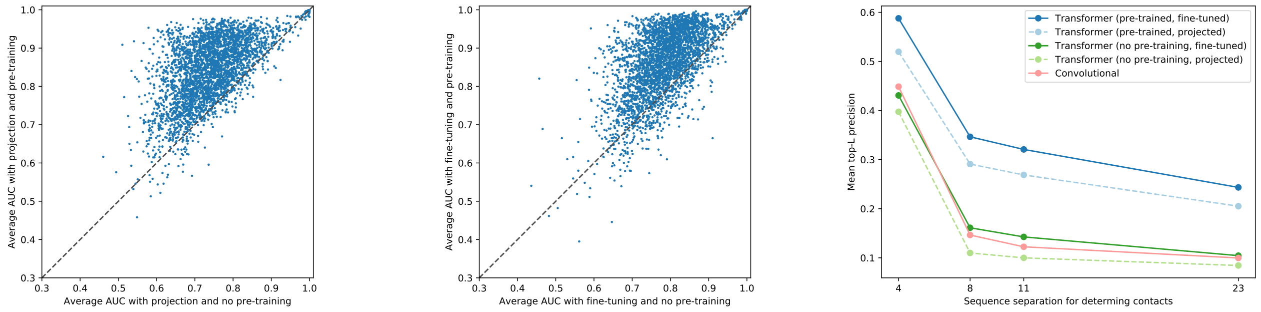
about secondary structure above the prior but performs worse than the PSSM features. In the initial stages of training a rapid increase in the information about secondary structure is observed (Figure 5a). A 36-layer Transformer model trained on the full pre-training dataset yields representations that project to 75.5% Q3 (3-class) or 60.8% Q8 (8-class) test accuracy. For calibration, current state-of-the-art on the same dataset is 84% on the Q3 (3-class) task (Wang et al., 2016) or 70.5% on Q8 (8-class) task (Drori et al., 2018) using a technique considerably more sophisticated than linear projection. Full results for both tasks are presented in Table S1.

Pre-training dataset scale The amount of information encoded in the learned representation is related to the scale of the data used for training. We examine the effect of pre-training dataset size on the model by training the same model with a randomly chosen subset of 1% of the full pre-training dataset. A gap is observed both in the relationship between accuracy and number of updates and also in accuracy for the same pre-training loss (Figure 5a). We conclude that two models with the same predictive performance on the self-supervision objective differ in their information content. Specifically, increasing the amount of training data also increases the amount of secondary structure information in the representation, even after controlling for training progress (*i.e.* for models with the same pre-training validation loss).

Pre-training dataset diversity To clarify the role that learning across diverse sets of sequences plays in generating representations that contain generalizable structural information, we compare the above setting (training across evolutionary statistics) to training on single protein families. We train separate models on the multiple sequence alignments of the three most common domains in nature longer than 100 amino acids, the ATP-binding domain of the ABC transporters, the protein kinase domain, and the response regulator receiver domain. We test the ability of models trained on one protein family to generalize information within-family and out-of-family by evaluating on sequences with ground truth labels from the family the model was trained on or from the alternate families. In all cases, the model trained on within-family sequences outperforms the models trained on out-of-family sequences (Table 3), indicating poor generalization of training on single MSA families. More significantly, however, the model trained across the full sequence diversity has a higher accuracy than the single-family model accuracies, even on the same-family evaluation dataset. This suggests that the representations learned from the full dataset are capturing general purpose information about secondary structure learned outside the sequence family.

5.2 Residue-residue contacts

Spatial contacts between amino acids describe a rotation- and translation-invariant representation of 3-dimensional protein structure. Contacts can be inferred through evolutionary correlations between residues in a sequence family (Marks et al., 2011; Anishchenko et al., 2017). Contacts predicted from evolutionary data have enabled large-scale computational prediction of protein structures (Ovchinnikov et al., 2017). Recently, deep neural networks have shown strong performance on predicting protein contacts and tertiary structure using evolutionary data. For inputs, these methods use raw frequency and covariance statistics, or parameters of maximum-entropy sequence models fitted on MSAs (Jones & Kandathil, 2018; Xu, 2018; Senior et al., 2018). However, these approaches often model structure at the family level and require access to many related sequences (*e.g.* to construct MSAs). Here we study whether networks trained on full evolutionary diversity have information that can be used to predict tertiary contacts directly from *single* sequences.

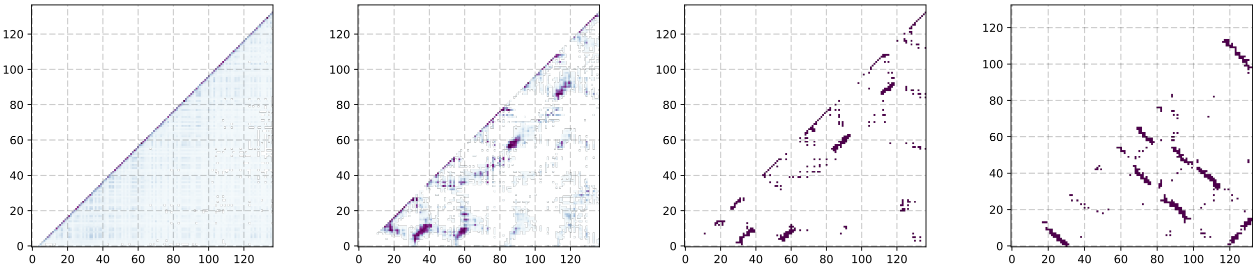


(a)

(b)

(c)

Figure 6: Information about residue-residue contacts can be recovered from the representations developed by unsupervised pre-training. In (a) and (b) each point is an AUC score for a single protein in the test set. The y and x coordinates show the performance with and without pre-training, respectively. In (a) the Transformer model parameters are frozen and linear projections are fit from empirical contacts; in (b) supervision is also backpropagated to all parameters of the Transformer model. The shift above the diagonal indicates the contribution of pre-training. Additionally, information about long range residue-residue contacts is recoverable from the representations; (c) depicts mean top-L precision values for contacts at different minimum sequence separations on test data. Fine-tuning and projections of the pre-trained Transformer both outperform direct end-to-end training of the same Transformer architecture with no pre-training, as well as a 1d convolutional baseline.



(a)

(b)

(c)

(d)

Figure 7: Contacts recovered by linear projections of the final hidden representations on a test domain, B12-binding subunit of glutamate mutase from Clostridium cochlearium (PDB: 1B1A; Hoffmann et al., 1999). Heatmaps indicate the probability assigned by the models: (a) the projected Transformer without pre-training, and (b) the projected pre-trained Transformer. True contacts are shown for: (c) the 1B1A sequence, and (d) the nearest neighbor of 1B1A (2OZN chain A, residues 774 - 907) by sequence identity in the training data. Additional examples are shown in Figure S2.

To adapt the Transformer network to predict contacts, we represent interactions between amino acids at positions i and j by a quadratic form in the final hidden representations, h, with learned projections, P and Q, having the network output:

$$w_{ij} = h_i^T [P^T Q] h_j$$

Training and test contact maps are constructed from empirical structures of protein domains, after filtering out all sequences in the train set that have an identity greater than 20% to any sequence in the test set. The network parameters are optimized by minimizing the binary cross entropy between w_{ij} and the data. To examine if tertiary contact information is linearly encoded in the final hidden representations, we freeze the network parameters and train only the linear projections P and Q of the learned final hidden representations. To confirm that the information is encoded in the representations and not simply a result of good projections, we control the experiment by comparing to a Transformer without pre-training. In this experiment, the pre-trained Transformer performs well (Table 4 and Figure S1), while the control performs worse than a convolutional baseline. Fine-tuning the network end-to-end gives significant additional gains, suggesting that additional information about tertiary structure is distributed throughout the network weights.

Model type	Full data	10% data
Transformer (pre-trained, fine-tuned)	0.863	0.806
Transformer (pre-trained, projected)	0.842	0.810
Transformer (no pre-training, fine-tuned)	0.783	0.727
Transformer (no pre-training, projected)	0.739	0.732
Convolutional	0.764	0.724

Table 4: Unsupervised pre-training enables generalization of residue-residue contact information from a small number of examples. Average AUC values on contact prediction test set. Results are shown comparing supervision from the full residue-residue contact training data with 10% of the training data. Corresponding ROC curves are shown in Figure S1. Pre-trained models combined with 10% of the contact data outperform models without pre-training that use 100% of the data.

Long range contacts We explore performance on tertiary contacts of increasing range of sequence separation. We evaluate mean top-L precision scores (where L is length of a protein sequence) on test data for sequence separations greater than 4, 8, 11, and 23 positions. Results are shown in Figure 6c. As expected, precision declines for all models as sequence separation is increased. The contacts predicted by fine-tuning the pre-trained model are consistently more precise than other baselines across contact separation distances. The benefit of pre-training is also highlighted in Figure 6a and Figure 6b, where we see that pre-trained Transformers consistently result in higher AUC scores for test proteins for both fine-tuning and projected variants, as compared to untrained Transformers. Notably, the projected pre-trained model (which can only make use of information directly present in the final representation) performs better than complete end-to-end training of the Transformer baselines.

Figure 7 shows an example of contacts recovered by linear projections of the final hidden representations for a domain in the test set, B12-binding subunit of glutamate mutase from Clostridium cochlearium (PDB: 1B1A; Hoffmann et al., 1999). Projections without pre-training appear to output a prior with probabilities that decreases with increasing sequence separation. Projections of the pre-trained Transformer recover interfaces between secondary structural elements of the fold including long range contacts. The nearest neighbor by sequence identity in the training data shows a completely different contact pattern. Additional examples for domains in the test set having diverse folds are shown in Figure S2.

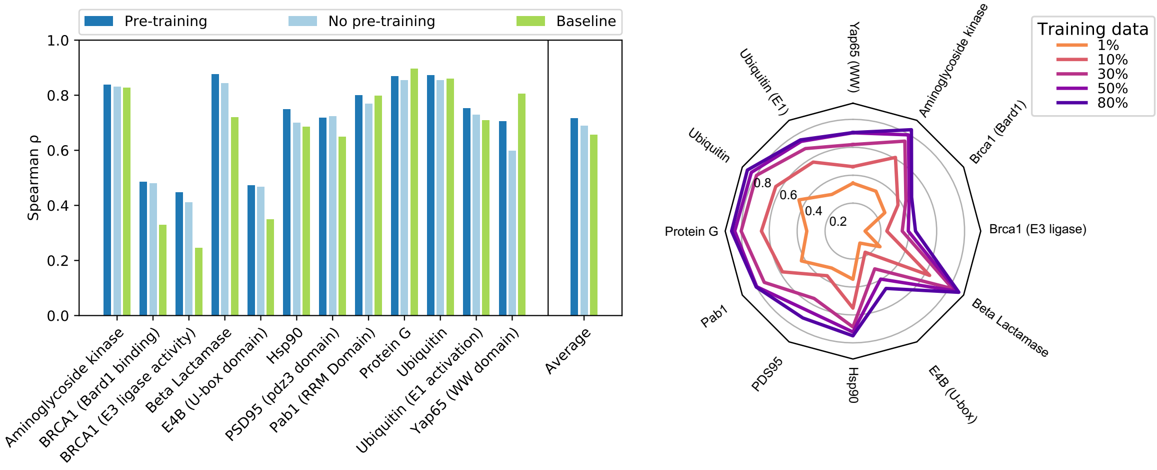
Generalization with pre-trained models We examine the benefit of pre-training when the fine-tuning dataset is small. We train all models on a randomly chosen subset of 10% of the full fine-tuning dataset. In this limited data regime, the pre-trained models perform significantly better than the non-pretrained or convolutional baseline (Table 4), highlighting the advantage of pre-training when the task data is limited.

6 Learned representations can be adapted to predict mutational effects on protein function

The mutational fitness landscape provides deep insight into biology: determining protein activity (Fowler & Fields, 2014), linking genetic variation to human health (Lek et al., 2016; Bycroft et al., 2018; Karczewski et al., 2019), and enabling for rational protein engineering (Slaymaker et al., 2016). For synthetic biology, iteratively querying a model of the mutational fitness landscape could help efficiently guide the introduction of mutations to enhance protein function (Romero & Arnold, 2009), inform protein design using a combination of activating mutants (Hu et al., 2018), and make rational substitutions to optimize protein properties such as substrate specificity (Packer et al., 2017), stability (Tan et al., 2014), and binding (Ricatti et al., 2019).

Computational variant effect predictors are useful for assessing the effect of point mutations (Gray et al., 2018; Adzhubei et al., 2013; Kumar et al., 2009; Hecht et al., 2015; Rentzsch et al., 2018). Often their predictions are based on basic evolutionary, physiochemical, and protein structural features that have been selected for their relevance to protein activity. However, features derived from evolutionary homology are not available for all proteins; and protein structural features are only available for a small fraction of proteins.

Coupling next generation sequencing with a mutagenesis screen allows parallel readout of tens of thousands of variants of a single protein (Fowler & Fields, 2014). The detail and coverage of these experiments provides a view into the mutational fitness landscape of individual proteins, giving quantitative relationships between sequence and protein function suitable for machine learning. Building on the finding that unsupervised learning develops knowledge of intrinsic protein properties in the internal representations of the Transformer – including biochemical, structural, and homology information relevant



(a)

(b)

Figure 8: After pre-training, the Transformer can be adapted to predict mutational effects on protein function. A 12-layer pre-trained Transformer model is fine-tuned on mutagenesis data: (a) compares the best Transformer model with and without pre-training, along with the baseline state-of-the-art variant effect model that uses structural and evolutionary features (Gray et al., 2018); where for each model 80% of the mutagenesis data is provided for supervision and the remainder is held out for evaluation; (b) shows performance on each protein when supervised with smaller fractions of the data.

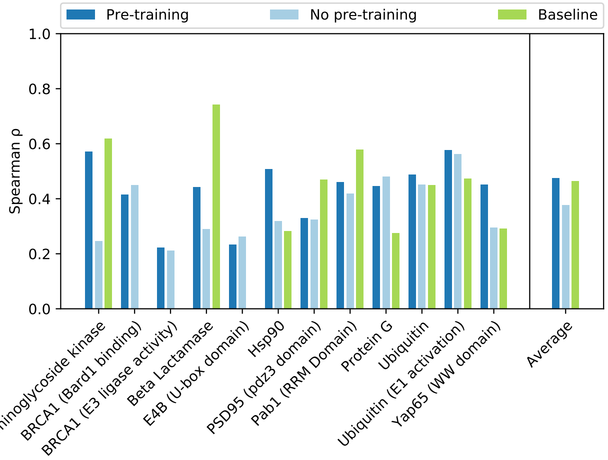


Figure 9: Pre-training improves the ability of the Transformer to generalize to the mutational fitness landscape of held-out proteins. All mutagenesis data from the protein selected for evaluation are held out, and the model is supervised with data from the remaining proteins. For each evaluation protein, a comparison is shown for the 12-layer Transformer with and without pre-training, along with the baseline. Spearman ρ is reported for the three models. Average performance across held-out proteins is also shown. [7]

to protein molecular function – we explore the possibility of adapting the Transformer to predict the quantitative effect of mutations.

We explore two settings for generalization of the information from large-scale mutagenesis data. The first is intra-protein variant effect prediction, where data from a limited sampling of mutations is used to predict the effect of unobserved mutations. This form of prediction is enabling for protein design and optimization applications, where repeated rounds of supervised

[7] Averages were computed across all proteins shown excluding E4B, BRCA1 (E3 ligase activity), and BRCA1 (Bard1 binding) which were not evaluated in Gray et al. (2018).

training and prediction can guide an iterative exploration of the fitness landscape to design a biological property of a protein. The second setting is generalization to the mutational fitness landscape of a completely new protein for which the model has not received any prior quantitative mutation information.

For a baseline, we compare with a recent state-of-the-art study (Gray et al., 2018) that combined over 20,000 variant effect measurements from nine large-scale experimental mutagenesis datasets to train a supervised model of the effect of missense mutations on protein activity. The baseline relies on protein structural and evolutionary features to generalize. We explore whether the Transformer can achieve similar generalization results, without direct access to structural features, using information learned through self-supervision on evolutionary sequence data. The same methodology for partitioning data for training and evaluation is used as in the baseline to allow a comparison of the results. Fine-tuning the Transformer (after unsupervised learning on evolutionary data) with labeled variant activity data yields a variant effect predictor with performance matching the baseline while making predictions directly from protein sequences.

Intra-protein variant effect prediction The Transformer exceeds the performance of the baseline on 10 of the 12 proteins. For each protein a fraction $p = 0.8$ of the data is used for training and the remaining data is used for testing (Figure 8a). For this experiment, we report the result of 5-fold cross-validation. There are minimal deviations across each run. When using 80% of the data for training, standard deviations in Spearman ρ are < 0.06 for each protein task (Table S2).

Fine-tuning on minimal data We also evaluate the performance of the fine-tuned Transformer in a limited data regime. For each protein, we vary the fraction of data that is used for training. We find that 86% and 40% of final performance can be achieved by training on just 30% or 1% of the data respectively (Figure 8b). As the average number of mutants tested per protein is 2379, this corresponds to training on data from < 25 mutants on average.

Generalization to held-out proteins We analyze the Transformer's ability to generalize to the fitness landscape of a new protein. To evaluate performance on a given protein, we train on data from the remaining $n - 1$ proteins and test on the held-out protein. Figure 9 shows that the Transformer's predictions from raw sequences perform better than the baseline on 5 of the 9 tasks.

These results suggest that the intrinsic knowledge of proteins discovered by self-supervision on large-scale protein data generalizes to predicting protein functional activity; and that the features developed through this learning – which are available for all proteins because they are from raw sequences – can be used to match performance of a state-of-the-art variant effect predictor that makes use of powerful structural and evolutionary feature inputs.

7 Discussion

One of the goals for artificial intelligence in biology could be the creation of controllable predictive and generative models that can read and generate biology in its native language. Accordingly, research will be necessary into methods that can learn intrinsic biological properties directly from protein sequences, which can be transferred to prediction and generation.

We investigated deep learning across evolution at the scale of the largest available protein sequence databases, training contextual language models across 86 billion amino acids from 250 million sequences. The space of representations learned from sequences by high-capacity networks reflects biological structure at many levels, including that of amino acids, proteins, groups of orthologous genes, and species. Information about secondary and tertiary structure is internalized and represented within the network in a generalizable form.

This information emerges without supervision – no learning signal other than sequences is given during training. We find that networks that have been trained across evolutionary data generalize: information can be extracted from representations by linear projections or by adapting the model using supervision. It is possible to adapt networks that have learned on evolutionary data to give results matching state-of-the-art on variant activity prediction directly from sequence – using only features that have been learned from sequences and without evolutionary and structural prior knowledge.

While the contextual language models we use are comparable with respect to data and capacity to large models studied in the literature for text data, our experiments have not yet reached the limit of scale. We observed that even the highest capacity models we trained (with approximately 700M parameters) underfit the 250M sequences, due to insufficient model capacity. The relationship we find between the information in the learned representations and predictive ability in the self-supervision task suggests that higher capacity models will yield better representations.

Combining high-capacity generative models with gene synthesis and high throughput characterization can enable generative biology. The network architectures we have trained can be used to generate new sequences (Wang & Cho, 2019). If neural networks can transfer knowledge learned from protein sequences to design functional proteins, this could be coupled with predictive models to jointly generate and optimize sequences for desired functions. The size of current sequence data and its projected growth point toward the possibility of a general purpose generative model that can condense the totality of sequence statistics, internalizing and integrating fundamental chemical and biological concepts including structure, function, activity, localization, binding, and dynamics, to generate new sequences that have not been seen before in nature but that are biologically active.

Acknowledgments

We thank Tristan Bepler, Richard Bonneau, Yilun Du, Vladimir Gligorijevic, Anika Gupta, Omer Levy, Ian Peikon, Hetunandan Kamisetty, Ethan Perez, Oded Regev, Neville Sanjana, and Emily Wang for feedback on the manuscript and insightful conversations.

References

Ivan Adzhubei, Daniel M Jordan, and Shamil R Sunyaev. Predicting functional effect of human missense mutations using polyphen-2. *Current protocols in human genetics*, 76(1):7–20, 2013.

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.

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.

Stephen F. Altschul and Eugene V. Koonin. Iterated profile searches with psi-blast – a tool for discovery in protein databases. *Trends in Biochemical Sciences*, 23(11):444–447, 11 1998. ISSN 0968-0004. doi: 10.1016/S0968-0004(98)01298-5.

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 0022-2836. doi: 10.1016/S0022-2836(05)80360-2.

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*, 114(34):9122–9127, 2017.

Alexei Baevski, Sergey Edunov, Yinhan Liu, Luke Zettlemoyer, and Michael Auli. Cloze-driven pretraining of self-attention networks. *CoRR*, abs/1903.07785, 2019. URL `http://arxiv.org/abs/1903.07785`.

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, 2011.

Alex Bateman, Andreas Heger, Erik L. L. Sonnhammer, Jaina Mistry, Jody Clements, John Tate, Kirstie Hetherington, Liisa Holm, Marco Punta, Penelope Coggill, Ruth Y. Eberhardt, Sean R. Eddy, and Robert D. Finn. Pfam: the protein families database. *Nucleic Acids Research*, 42(D1):D222–D230, 11 2013. ISSN 0305-1048. doi: 10.1093/nar/gkt1223.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. *Journal of machine learning research*, 3(Feb):1137–1155, 2003.

Clare Bycroft, Colin Freeman, Desislava Petkova, Gavin Band, Lloyd T Elliott, Kevin Sharp, Allan Motyer, Damjan Vukcevic, Olivier Delaneau, Jared O'Connell, et al. The uk biobank resource with deep phenotyping and genomic data. *Nature*, 562 (7726):203, 2018.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. The MIT Press, 2006.

Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pp. 160–167. ACM, 2008.

Michael S. Cosgrove, Claire Naylor, Søren Paludan, Margaret J. Adams, and H. Richard Levy. On the mechanism of the reaction catalyzed by glucose 6-phosphate dehydrogenase,. *Biochemistry*, 37(9):2759–2767, 03 1998. ISSN 0006-2960. doi: 10.1021/bi972069y.

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

Andrew M Dai and Quoc V Le. Semi-supervised sequence learning. In *Advances in Neural Information Processing Systems*, pp. 3079–3087, 2015.

Yann N. Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pp. 933–941, 2017. URL `http://proceedings.mlr.press/v70/dauphin17a.html`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

Iddo Drori, Isht Dwivedi, Pranav Shrestha, Jeffrey Wan, Yueqi Wang, Yunchu He, Anthony Mazza, Hugh Krogh-Freeman, Dimitri Leggas, Kendal Sandridge, Linyong Nan, Kaveri A. Thakoor, Chinmay Joshi, Sonam Goenka, Chen Keasar, and Itsik Pe'er. High quality prediction of protein Q8 secondary structure by diverse neural network architectures. In *32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada, Workshop on Machine Learning for Molecules and Materials*, 2018. URL `https://arxiv.org/abs/1811.07143`.

Sean R. Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 10 1998. ISSN 1367-4803. doi: 10.1093/bioinformatics/14.9.755.

Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. *Phys. Rev. E*, 87:012707, Jan 2013a. doi: 10.1103/PhysRevE.87.012707. URL `https://link.aps.org/doi/10.1103/PhysRevE.87.012707`.

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*, 87(1):012707, 2013b.

Douglas M Fowler and Stanley Fields. Deep mutational scanning: a new style of protein science. *Nature methods*, 11(8):801, 2014.

Toni Gabaldon. Evolution of proteins and proteomes: a phylogenetics approach. *Evol Bioinform Online*, 1:51–61, 2007.

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.

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, 2018.

Zellig S. Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.

Maximilian Hecht, Yana Bromberg, and Burkhard Rost. Better prediction of functional effects for sequence variants. *BMC genomics*, 16(8):S1, 2015.

John Hewitt and Christopher D. Manning. A structural probe for finding syntax in word representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, Minneapolis, Minnesota, USA, June 2-7, 2019, Volume 2 (Short Papers)*, 2019.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313 (5786):504–507, 2006.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735.

Sepp Hochreiter, Yoshua Bengio, and Paolo Frasconi. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies. 2001.

Bernd Hoffmann, Robert Konrat, Harald Bothe, Wolfgang Buckel, and Bernhard Kräutler. Structure and dynamics of the b12-binding subunit of glutamate mutase from clostridium cochlearium. *European journal of biochemistry*, 263(1): 178–188, 1999.

Thomas Hopf, John Ingraham, Frank Poelwijk, Charlotta Scharfe, Michael Springer, Chris Sander, and Debora Marks. Mutation effects predicted from sequence co-variation. *Nature Biotechnology*, 35:128–135, 1 2017.

Sahand Hormoz. Amino acid composition of proteins reduces deleterious impact of mutations. *Scientific reports*, 3:2919, 2013.

Johnny H Hu, Shannon M Miller, Maarten H Geurts, Weixin Tang, Liwei Chen, Ning Sun, Christina M Zeina, Xue Gao, Holly A Rees, Zhi Lin, et al. Evolved cas9 variants with broad pam compatibility and high dna specificity. *Nature*, 556 (7699):57, 2018.

Jaime Huerta-Cepas, Sofia K Forslund, Peer Bork, Ana Hernández-Plaza, Christian von Mering, Damian Szklarczyk, Davide Heller, Helen Cook, Lars J Jensen, Daniel R Mende, Ivica Letunic, and Thomas Rattei. eggNOG 5.0: a hierarchical, functionally and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic Acids Research*, 47(D1):D309–D314, 11 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1085.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with gpus. *CoRR*, abs/1702.08734, 2017. URL `http://arxiv.org/abs/1702.08734`.

David Jones. Protein secondary structure prediction based on position-specific scoring matrices. *Journal of Molecular Biology*, 292:195–202, 10 1999. doi: 10.1006/jmbi.1999.3091.

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, 2018.

David T Jones, Daniel WA 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, 2011.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410*, 2016.

Wolfgang Kabsch and Christian Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637, 1983.

Konrad J Karczewski, Laurent C Francioli, Grace Tiao, Beryl B Cummings, Jessica Alföldi, Qingbo Wang, Ryan L Collins, Kristen M Laricchia, Andrea Ganna, Daniel P Birnbaum, et al. Variation across 141,456 human exomes and genomes reveals the spectrum of loss-of-function intolerance across human protein-coding genes. *BioRxiv*, pp. 531210, 2019.

Masato Kato, Toshiyuki Shimizu, Takeshi Mizuno, and Toshio Hakoshima. Structure of the histidine-containing phosphotransfer (hpt) domain of the anaerobic sensor protein arcb complexed with the chemotaxis response regulator chey. *Acta Crystallographica Section D*, 55(7):1257–1263, 07 1999. doi: 10.1107/S0907444999005053.

Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. Sharp nearby, fuzzy far away: How neural language models use context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 284–294, 2018.

Daisuke Kihara. The effect of long-range interactions on the secondary structure formation of proteins. *Protein Sci*, 2005. doi: 10.1110/ps.051479505.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, pp. 2741–2749, 2016. URL `http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12489`.

Fyodor A. Kondrashov, Peer Bork, Shamil Sunyaev, and Vasily Ramensky. Impact of selection, mutation rate and genetic drift on human genetic variation. *Human Molecular Genetics*, 12(24):3325–3330, 12 2003. ISSN 0964-6906. doi: 10.1093/hmg/ddg359. URL `https://doi.org/10.1093/hmg/ddg359`.

Prateek Kumar, Steven Henikoff, and Pauline C Ng. Predicting the effects of coding non-synonymous variants on protein function using the sift algorithm. *Nature protocols*, 4(7):1073, 2009.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. Unsupervised machine translation using monolingual corpora only. *CoRR*, abs/1711.00043, 2017. URL `http://arxiv.org/abs/1711.00043`.

Monkol Lek, Konrad J Karczewski, Eric V Minikel, Kaitlin E Samocha, Eric Banks, Timothy Fennell, Anne H O'Donnell-Luria, James S Ware, Andrew J Hill, Beryl B Cummings, et al. Analysis of protein-coding genetic variation in 60,706 humans. *Nature*, 536(7616):285, 2016.

Michael Levitt. Conformational preferences of amino acids in globular proteins. *Biochemistry*, 17(20):4277–4285, 1978.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov): 2579–2605, 2008.

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):e28766, 2011.

Tomáš Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocky. Subword language modeling with neural networks. *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 8, 2012.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a. URL `http://arxiv.org/abs/1301.3781`.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pp. 3111–3119, 2013b.

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*, 108(49):E1293–E1301, 2011.

Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, pp. 807–814, USA, 2010. Omnipress. ISBN 978-1-60558-907-7. URL `http://dl.acm.org/citation.cfm?id=3104322.3104425`.

Christine A Orengo, AD Michie, S Jones, David T Jones, MB Swindells, and Janet M Thornton. Cath–a hierarchic classification of protein domain structures. *Structure*, 5(8):1093–1109, 1997.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*, 2019.

Sergey Ovchinnikov, Hahnbeom Park, Neha Varghese, Po-Ssu Huang, Georgios A Pavlopoulos, David E Kim, Hetunandan Kamisetty, Nikos C Kyrpides, and David Baker. Protein structure determination using metagenome sequence data. *Science*, 355(6322):294–298, 2017.

Julie Overbaugh and Charles Bangham. Selection forces and constraints on retroviral sequence variation. *Science*, 292: 1106–1109, 5 2001. doi: 10.1126/science.1059128.

Michael S. Packer, Holly A. Rees, and David R. Liu. Phage-assisted continuous evolution of proteases with altered substrate specificity. *Nature Communications*, 8(1):956, 2017. ISSN 2041-1723. doi: 10.1038/s41467-017-01055-9. URL `https://doi.org/10.1038/s41467-017-01055-9`.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1532–1543, 2014. URL `http://aclweb.org/anthology/D/D14/D14-1162.pdf`.

Matthew E. 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 American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pp. 2227–2237, 2018. URL `https://aclanthology.info/papers/N18-1202/n18-1202`.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

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

Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature Methods*, 9:173 EP, 12 2011. doi: 10.1038/nmeth.1818.

Philipp Rentzsch, Daniela Witten, Gregory M Cooper, Jay Shendure, and Martin Kircher. Cadd: predicting the deleteriousness of variants throughout the human genome. *Nucleic acids research*, 47(D1):D886–D894, 2018.

Jimena Ricatti, Laura Acquasaliente, Giovanni Ribaudo, Vincenzo De Filippis, Marino Bellini, Ramiro Esteban Llovera, Susi Barollo, Raffaele Pezzani, Giuseppe Zagotto, Krishna C Persaud, et al. Effects of point mutations in the binding pocket of the mouse major urinary protein mup20 on ligand affinity and specificity. *Scientific reports*, 9(1):300, 2019.

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, 2018. ISSN 1548-7105. doi: 10.1038/s41592-018-0138-4.

Philip A Romero and Frances H Arnold. Exploring protein fitness landscapes by directed evolution. *Nature reviews Molecular cell biology*, 10(12):866, 2009.

Andrew Senior, John Jumper, and Demis Hassabis. AlphaFold: Using AI for scientific discovery, 12 2018. URL `https://deepmind.com/blog/alphafold/`.

Ian M. Slaymaker, Linyi Gao, Bernd Zetsche, David A. Scott, Winston X. Yan, and Feng Zhang. Rationally engineered cas9 nucleases with improved specificity. *Science*, 351(6268):84–88, 2016. ISSN 0036-8075. doi: 10.1126/science.aad5227. URL `https://science.sciencemag.org/content/351/6268/84`.

Johannes Söding. Protein homology detection by hmm–hmm comparison. *Bioinformatics*, 21(7):951–960, 2004.

Chen Sun, Austin Myers, Carl Vondrick, Kevin Murphy, and Cordelia Schmid. Videobert: A joint model for video and language representation learning. *CoRR*, abs/1904.01766, 2019. URL `http://arxiv.org/abs/1904.01766`.

Nikki Y. Tan, Ulla-Maja Bailey, M. Fairuz Jamaluddin, S. Halimah Binte Mahmud, Suresh C. Raman, and Benjamin L. Schulz. Sequence-based protein stabilization in the absence of glycosylation. *Nature Communications*, 5:3099 EP –, Jan 2014. URL `https://doi.org/10.1038/ncomms4099`. Article.

The UniProt Consortium. The universal protein resource (uniprot). *Nucleic Acids Research*, 36(suppl_1):D190–D195, 11 2007. ISSN 0305-1048. doi: 10.1093/nar/gkm895.

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*, pp. 5998–6008, 2017.

Alex Wang and Kyunghyun Cho. BERT has a mouth, and it must speak: BERT as a markov random field language model. *CoRR*, abs/1902.04094, 2019. URL `http://arxiv.org/abs/1902.04094`.

Sheng Wang, Jian Peng, Jianzhu Ma, and Jinbo Xu. Protein secondary structure prediction using deep convolutional neural fields. *Scientific Reports*, 6, Jan 2016. URL `https://doi.org/10.1038/srep18962`. Article.

Shou-Wen Wang, Anne-Fllorence Bitbol, and Ned Wingreen. Revealing evolutionary constraints on proteins through sequence analysis. *PLoS Comput Biol*, 15(4), 2019. doi: 10.1371/journal.pcbi.1007010.

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*, 106(1):67–72, 2009.

Jinbo Xu. Distance-based protein folding powered by deep learning. *arXiv preprint arXiv:1811.03481*, 2018.

Charles Yanofsky, Virginia Horn, and Deanna Thorpe. Protein structure relationships revealed by mutational analysis. *Science*, 146(3651):1593–1594, 1964.

Jian Zhou and Olga G. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, pp. 745–753, 2014. URL `http://jmlr.org/proceedings/papers/v32/zhou14.html`.

A Approach & Data

A.1 Background on language models and embeddings

Deep language models are parametric functions that map sequences into distributed word representations in a learned vector space (Bengio et al., 2003; Mikolov et al., 2013a;b; Pennington et al., 2014; Lample et al., 2017). These vectors, called embeddings, distribute the meaning of the source sequence across multiple components of the vector, such that semantically similar objects are mapped to nearby vectors in representation space.

Recently, results of large-scale pre-trained language models have advanced the field of natural language processing and impacted the broader deep learning community. Peters et al. (2018) used a coupled language model objective to train bidirectional LSTMs in order to extract context dependent word embeddings. These embeddings showed improvement for a range of NLP tasks. Radford et al. (2018) explored a semi-supervised method using left-to-right Transformer models to learn universal representations using unsupervised pre-training, followed by fine-tuning on specific downstream tasks. Recently, Devlin et al. (2018) proposed BERT to pre-train deep representations using bidirectional Transformer models. They proposed a pre-training objective based on masked language modeling, which allowed capturing both left and right contexts to obtain deep bidirectional token representations, and obtained state-of-the-art results on 11 downstream language understanding tasks, such as question answering and natural language inference, without substantial task-specific architecture modifications. Radford et al. (2019) extended their earlier work and proposed GPT-2, highlighting the importance of scale along dimensions of number of model parameters and size of pre-training data, and demonstrated their learned language model is able to achieve surprisingly good results on various NLP tasks without task-specific training. Analogous follow-up work on other domains have also shown promising results (Sun et al., 2019).

A.2 Inductive Semi-supervised Learning

Semi-supervised learning (Chapelle et al., 2006) is a hybrid of two classical learning problems: unsupervised learning and supervised learning. In unsupervised learning, the assumption is that samples, x, are drawn, typically i.i.d., from a density $p(x)$. The objective is to learn about the structure of the data, $\mathbf{X}$, without access to labels. In supervised learning, the assumption is that data, x, and labels, y, are jointly sampled from $p(x, y)$. Discriminative methods seek to estimate $p(y|x)$ from labeled data.

Semi-supervised learning combines unsupervised and supervised learning under the hypothesis that information about the structure of $p(x)$ will be useful for modeling $p(y|x)$. In the inductive learning setting, the objective is to learn $p(y|x)$ for the whole of the data, $\mathbf{X}$, encompassing its labeled and unlabeled parts. When the labels are sparse or restricted to small regions of the overall data, the problem is one of generalization: the information in the labels must be extended across the whole data using structure that has been learned from the unlabeled data.

Unsupervised representation learning using deep neural networks (Hinton & Salakhutdinov, 2006) has been effective for generalization in this setting. An initial unsupervised pre-training stage exploits the samples from $p(x)$ to learn about the structure of the data and identify useful features. In the second stage the representations are used to generalize from the labeled examples. The representations and parameters can be used either directly as input features to a classifier, or through adaptation by fine-tuning. Unsupervised learning improves generalization by bringing information into the representation and parameters of the network that is useful for modeling the data (Hinton & Salakhutdinov, 2006).

The promise of inductive semi-supervised learning lies in its ability to exploit unlabeled information. In many applications (including the biological problems considered in this paper) unlabeled data is available in large quantities far exceeding what is available with labels. Exploiting unlabeled information to make more accurate predictions and also to extend predictions into regions of the data that are not well-represented (or represented at all) with labels is a current direction in artificial intelligence research undergoing rapid progress (Radford et al., 2018; Devlin et al., 2018).

Formally, the experiments of this paper follow the classic inductive semi-supervised learning setup. Unlabeled data of protein sequences, $\mathbf{X}$, is given as i.i.d. samples of individual sequences, x, from $p(x)$, the empirical distribution of protein sequences. For a small fraction of the sequences, additional labeled information, y, is available. This labeled information is about the biological properties of the sequences, such as their structural features or biological activity. The goal is to learn a predictor that generalizes from the labeled examples, formally sequence label pairs (x, y), to all the sequences in $\mathbf{X}$.

Unsupervised pre-training In the unsupervised pre-training stage we train a contextual language model using self-supervision on only sequence data, excluding a held-out validation set used to track performance and control overfitting during training. No information about labels is provided to the network during the unsupervised stage of the training; only raw sequences are used. To effectively model this data, information about the underlying patterns and structure of protein sequence data will need to be discovered and compressed into the network's representations and parameters.

Generalization, projections and fine-tuning In the second stage we investigate the representations and parameters learned during the pre-training stage for structure and organization that reflects biological properties. For prediction problems we look for the emergence of information developed through unsupervised learning that generalizes in: 1) the final hidden representations; and 2) the parameters of the network. To look for information in the representations, we use linear projections to identify linearly encoded information. To look for information in the parameters, we fine-tune the network by adapting all parameters with supervision from labeled data. In both cases to test for generalization we evaluate on training and test partitions that have been filtered to exclude sequences from the training set that are similar to test sequences (details below). This form of evaluation explores the ability of the learned representations to generalize the labeled information.

Connections to classical methods in computational biology In the inductive semi-supervised learning formalism adopted by this paper, the way that the labeled and unlabeled data is treated is the same as in classical computational biology methods using large sequence databases. Sequence search underlies many classical and state of the art approaches to biological prediction. Structural and functional properties can be imputed by sequence homology (Eddy, 1998; Söding, 2004). Single-family generative models fit parametric distributions to MSA data (Weigt et al., 2009; Marks et al., 2011; Morcos et al., 2011; Jones et al., 2011; Balakrishnan et al., 2011; Ekeberg et al., 2013b). These methods seek to exploit the information in the unlabeled data of sequences to infer useful structure for their prediction problems. In this paper, unsupervised pre-training on raw sequences has an analogous function, aiming to learn from the dependencies and variations among highly related, as well as very different, sequences to develop a common representation through a generative model that explains their variations. The distinction in this paper is that instead of training a parametric model on a single family of related sequences, a single parametric model is trained across all sequences. The model thereby proposes to capture and represent dependencies that extend beyond the single sequence family level to the full variation in the sequence database.

A note on generalization There are multiple ways the network might generalize information across the pre-training domain. One is sequence similarity: by assigning similar representations to similar sequences, unsupervised pre-training could enable transfer of labeled information by sequence similarity. At a deeper level, the network might discover and represent information about biological structure and function in the form of commonalities across all sequences. In this paper we explore the representations developed by the network from both perspectives, looking for organization according to phylogenetic relations, as well as for the presence of information about more fundamental biological properties that generalize beyond basic sequence similarity.

A question we explore is whether through unsupervised learning across all protein sequences, the model discovers information about the intrinsic properties of proteins. If the network's generalization capability reduces to making similar predictions for similar sequences, then we would only expect information such as secondary structure, residue-residue contacts, and variant activity, to transfer between proteins that have related sequences. The presence of information about structure and function that generalizes beyond sequence similarity argues that information has been learned at a more basic level.

Data splitting methodology Experimental methodology for the evaluations is summarized below and further details are provided in later sections of the Appendix.

- In the experiments on secondary structure, the training dataset described in Zhou & Troyanskaya (2014), was used with evaluation on the CB513 benchmark (Cuff & Barton, 1999). In brief, Zhou & Troyanskaya (2014) obtained a set of structures with better than 2.5Å resolution, with no two proteins having sequence similarity above 30%. The training set was further filtered to remove sequences with greater than 25% identity with the CB513 dataset.
- In the experiments on residue-residue contacts, the S40 subset of CATH (Orengo et al., 1997) is used to construct the dataset. S40 is defined so that no two domains have sequence similarity greater than 40%. Only domains present in the S40 set, are used (rather than entire PDB chains). Non-contiguous domains are split into their constituent parts. Sequences less than 50 residues in length are excluded. A random subset of 10% of the domains is chosen for the test set. Then the train set is searched using blastp to exclude any sequences in the train set with sequence identity greater than 20% to the test set domains. For validation during training, a further 10% of the remaining train proteins are held out.
- In the experiments on variant activity prediction, the same data and methodology for generating train and test splits is used following Gray et al. (2018), the method compared against. Generalization to the fitness landscape of proteins unrelated to those in the training data is examined by leaving one protein out of the training data and training the predictor on mutagenesis data from the remaining proteins. Evaluations are reported on the held-out protein.

A.3 Model Architecture

We use a deep bidirectional Transformer encoder model (Devlin et al., 2018; Vaswani et al., 2017) and process data at the character-level, corresponding to individual amino-acids in our application. In contrast to models that are based on recurrent

# Layers	# Heads	Embedding Dim	MLP Dim	# Params	Steps (A)	Steps (B)
12	12	768	3072	85.1M	1.5M	1.6M
24	12	768	3072	170.2M	220k	300k
36	20	1280	5120	708.6M	200k	300k

Table 5: Hyperparameters for Transformer models trained in this paper. Embedding dim refers to the amino acid representation dimensionality. MLP dim refers to the width of hidden layers in the Transformer's MLPs. (A) refers to the number of pre-training steps before analyzing secondary structure. (B) gives the number of pre-training steps before visualizations were performed and fine-tuning on tertiary structure and protein mutagenesis data was conducted.

or convolutional neural networks, the Transformer makes no assumptions on the ordering of the input and instead uses position embeddings. Particularly relevant to protein sequences is the Transformer's natural ability to model long range dependencies, which are not effectively captured by RNNs or LSTMs (Khandelwal et al., 2018). One key factor affecting the performance of LSTMs on these tasks is the path lengths that must be traversed by forward activation and backward gradient signals in the network (Hochreiter et al., 2001). It is well known that structural properties of protein sequences are reflected in long-range dependencies (Kihara, 2005). Classical methods that aim to detect pairwise dependencies in multiple sequence alignments are able to model entire sequences. Similarly, the Transformer builds up a representation of a sequence by alternating self attention with non-linear projections. Self attention structures computation so that each position is represented by a weighted sum of the other positions in the sequence. The attention weights are computed dynamically and allow each position to choose what information from the rest of the sequence to integrate at every computation step. Developed to model large contexts and long range dependencies in language data, self-attention architectures currently give state-of-the-art performance on various natural language tasks, mostly due to the Transformer's scalability in parameters and the amount of context it can integrate (Devlin et al., 2018). The tasks include token-level tasks like part-of-speech tagging, sentence-level tasks such as textual entailment, and paragraph-level tasks like question-answering.

Each symbol in the raw input sequence is represented as a 1-hot vector of dimension 25, which is then passed through a learned embedding layer before being presented to the first Transformer layer. The Transformer consists of a series of layers, each comprised of (i) multi-head scaled dot product attention and (ii) a multilayer feed-forward network. In (i) a layer normalization operation (Ba et al., 2016) is applied to the embedding, before it undergoes three separate linear projections via learned weights (distinct for each head) W_q, W_k, W_v to produce query q, key k and valve v vectors. These vectors from all positions in the sequence are stacked into corresponding matrices Q, K and V. The output of a single head is $\frac{\text{softmax}(QK^T)}{\sqrt{d_k}} \cdot V$, where d_k is the dimensionality of the projection. This operation ensures that each position in the sequence attends to all other positions. The outputs from multiple heads are then concatenated. This output is summed with a residual tensor from before (i) and is passed to (ii). In (ii), another layer normalization operation is applied and the output is passed through a multi-layer feed-forward network with ReLU activation (Nair & Hinton, 2010) and summed with a residual tensor from before (ii). This result is an intermediate embedding for each position of the input sequence and is then passed to the next layer. Thus, the overall network takes a sequence as input and outputs a sequence of vectors forming a distributed representation of the input sequence which we refer to as the final representation for the sequence. The parameters of the model are the weight matrices W_q, W_k, W_v for each layer and attention head; the weights of the feedforward network in each layer/head and the batch norm scale/shift parameters. The training procedure for these weights is detailed in Appendix A.4.

In our work, we experimented with Transformer models of various depths, including a 36 layer Transformer with 708.6 million parameters (Table 5). All models were trained using the fairseq toolkit (Ott et al., 2019) on 128 NVIDIA V100 GPUs for $\tilde{4}$ days with sinusoidal positional embeddings (Vaswani et al., 2017), without layer norm in the final layer, and using half precision. The 36 and 24 layer models were trained with initializations from Radford et al. (2019), while the 12 layer model was trained with initializations from Vaswani et al. (2017). Batch size was approximately 2500 sequences for each model.

A.4 Pre-training the models

During unsupervised pre-training, the final high dimensional sequence representations are projected, via a linear layer, to vectors of unnormalized log probabilities. These probabilities correspond to the model's posterior estimate that a given amino acid is at that position, given all other amino acids in the sequence. These estimates are optimized using a masked language model objective. We follow the masking procedure from Devlin et al. (2018) in which the network is trained by noising (either by masking out or randomly perturbing) a fraction of its inputs and optimizing the network to predict the amino acid at the noised position from the complete noised sequence. In contrast to Devlin et al. (2018), we do not use any additional auxiliary prediction losses.

Protein sequence pre-training data We train on the full Uniparc dataset (The UniProt Consortium, 2007) without any form of preprocessing or data cleaning. This dataset contains 250 million sequences comprising 86 billion amino acids. For context, other recently published language modeling work includes Radford et al. (2019), who used 40GB of internet text or roughly 40 billion characters; Baevski et al. (2019), who used 18 billion words; and Jozefowicz et al. (2016), who used 100 billion words.

We constructed a “full data” split by partitioning the full dataset into 249M training and 1M validation examples. We subsampled the 249M training split to construct three non-overlapping “limited data” training sets with 10%, 1%, and 0.1% data, which consist of 24M, 2M and 240k sequences respectively. The validation set used for these “limited data” experiments was the same as in the “full data” split.

The 24 layer and 12 layer Transformer models were trained on a different split of the full dataset, with 225M sequences randomly selected for training and the remaining 25M sequences used for validation.

Pre-training task The pre-training task follows Task #1 in Section 3.3.1 of Devlin et al. (2018). Specifically, we select as supervision 15% of tokens randomly sampled from the sequence. For those 15% of tokens, we change the input token to a special “masking” token with 80% probability, a randomly-chosen alternate amino acid token with 10% probability, and the original input token (i.e. no change) with 10% probability. We take the loss to be the whole batch average cross entropy loss between the model’s predictions and the true token for these 15% of amino acid tokens. We then report the mean exponentiated cross-entropy (ECE) for the various models trained, as shown in Table 1.

Pre-training details Our model was pre-trained using a context size of 1024 tokens. As most Uniparc sequences (96.7%) contain fewer than 1024 amino acids, the Transformer is able to model the entire context in a single model pass. For those sequences that are longer than 1024 tokens, we sampled a random crop of 1024 tokens during each training epoch. The model was optimized using Adam ($\beta_1 = 0.9, \beta_2 = 0.999$) with learning rate 10^{-4}. The models follow a warm-up period of 16000 updates, during which the learning rate increases linearly. Afterwards, the learning rate follows an inverse square root decay schedule. The number of pre-training steps is listed in Table 5.

A.5 Baselines

In addition to comparing to past work, we also implemented a variety of deep learning baselines for our experiments.

Frequency (n-gram) models To establish a meaningful performance baseline on the sequence modeling task (Section 3), we construct n-gram frequency-based models for context sizes $1 \leq n \leq 10^4$, applying optimal Laplace smoothing for each context size. The Laplace smoothing hyperparameter in each case was tuned on the validation set. Both unidirectional (left conditioning) and bidirectional (left and right conditioning) variants are used.

Recurrent neural networks For this baseline (applied throughout Section 4), we embedded the amino acids to a representation of dimension 512 and applied a single layer recurrent neural network with long-short term memory (LSTM) with hidden dimension 1280.

Convolutional neural networks For residue-residue contact prediction (Section 5.2), we trained convolution baselines where we embedded the amino acids to a representation of dimension k, resulting in a k x N image (where N is the sequence length). We then applied a convolutional layer with C filters and kernel size M followed by a multilayer perceptron with hidden size D. We tried different values of k, C, M and D, to get best performing models. The model was trained using Adam with $\beta_1 = 0.9, \beta_2 = 0.999$ and fixed learning rate $= 10^{-4}$.

Ablations on Transformer model To investigate the importance of pre-training, we compared to a Transformer with random weights (i.e. using the same random initializations as the pre-trained models, but without performing any training) for the analyses in Section 4 and Section 5. In Section 5, we refer to these baselines as “no pre-training”, and elsewhere, we refer to these baselines as “untrained”. Separately, to assess the effect of full network fine-tuning, we tested Transformer models where all weights outside task-specific layers are frozen during fine-tuning for the analysis in Section 5.2. In these “projected” baselines, none of the base Transformer parameters are modified during fine-tuning.

A.6 Representational similarity-based alignment of sequences within MSA families

Family selection For the analysis in Section 4.4, we selected structural families from the Pfam database (Bateman et al., 2013). We first filtered out any families whose longest sequence is less than 32 residues or greater than 1024 residues in length. We then ranked the families by the number of sequences contained in each family and selected the 128 largest families and associated MSAs. Finally, we reduced the size of each family to 128 sequences by uniform random sampling.

Aligned pair distribution For each family, we construct an empirical distribution of aligned residue pairs by enumerating all pairs of positions and indices that are aligned within the MSA and uniformly sampling 50000 pairs.

Unaligned pair distribution We also construct for each family a background empirical distribution of unaligned residue pairs. This background distribution needs to control for within-sequence position, since the residues of two sequences that have been aligned in an MSA are likely to occupy similar positions within their respective unaligned source sequences. Without controlling for this bias, a difference in the distributions of aligned and unaligned pairs could arise from representations encoding positional information rather than actual context. We control for this effect by sampling from the unaligned-pair distribution in proportion to the observed positional differences from the aligned-pair distribution. Specifically, the following process is repeated for each pair in the empirical aligned distribution:

1. Calculate the absolute value of the difference of each residue's within-sequence positions in the aligned pair.
2. Select a pair of sequences at random.
3. For that pair of sequences, select a pair of residues at random whose absolute value of positional difference equals the one calculated above.
4. Verify that the residues are unaligned in the MSA; if so, add the pair to the empirical background distribution.
5. Otherwise, return to step 2.

This procedure suffices to compute a empirical background distribution of 50000 unaligned residue pairs.

Similarity distributions Finally, for each family and each distribution, we apply the cosine similarity operator to each pair of residues to obtain the per-family aligned and unaligned distribution of representational cosine similarities.

Area under the ROC curve Area under the ROC curve is commonly used to measure the performance of classification problem at various thresholds settings. ROC is a probability curve illustrating the relationship between true positive rate and false positive rate for a binary classification task, and AUC represents degree or measure of separability. It quantifies the model's capability of distinguishing between classes. Here, we use cosine similarity scores as outputs of a classifier aimed at differentiate between aligned and unaligned pairs.

A.7 Orthology visualations

Full orthology dataset For the analyses in Section 4, an orthologous group dataset was constructed from eggNOG 5.0 (Huerta-Cepas et al., 2018) by selecting 25 COG orthologous groups toward maximizing the size of the intersected set of species within each orthologous group. Through a greedy algorithm, we selected 25 COG groups with an intersecting set of 2609 species. This dataset was used directly in Section 4.2's phylogenetic organization analysis.

Diverse orthology dataset For all analyses in Section 4 besides the aforementioned phylogenetic organization analysis, we shrank the dataset above by selecting only one species from each of 24 phyla in order to ensure species-level diversity.

Phylogenetic distance projection To learn the distance projection used in the phylogenetic organization analysis, we applied an adapted version of the technique of Hewitt & Manning (2019), originally employed toward learning a natural language parse-tree encoding; we modify the technique to instead learn a phylogenetic tree encoding. We optimize a fixed-size projection matrix $\boldsymbol{B}$ toward the following minimization objective:

$$\boldsymbol{B}^* \triangleq \arg\min_{\boldsymbol{B}} \sum_{s_i, s_j} \left(d(s_i, s_j) - \|\boldsymbol{B}(r(s_i) - r(s_j))\|_2^2 \right)^2$$

where s_i and s_j denote species within the dataset, $d(s_i, s_j)$ denotes a pairwise distance kernel between two species, and $r(s)$ denotes the representation of species s.

In this analysis, we define the species representation $r(s)$ to be an average of the representations of the species' sequences within the dataset. We also define the distance kernel as follows:

$$d(s_i, s_j) \triangleq 2^{-1/3 \cdot \mathrm{LCA}(s_i, s_j)}$$

where LCA denotes the height of the least common ancestor of species s_i and s_j in the phylogenetic tree. We found the optimal projection via mini-batch stochastic gradient descent, which quickly converges on this objective and dataset.

A.8 Vector-based protein search

For the analysis in Section 4.3, we define various translation operations on the representation space of the diverse orthology dataset described above. Notationally, we use s to denote a single sequence, $r(s)$ to denote a representation of a sequence, $\mathbb{G}$ to denote an orthologous group, and $\mathbb{S}$ to denote a species. Both $\mathbb{G}$ and $\mathbb{S}$ are considered to be collections containing individual sequences.

Orthologous group translation For a given sequence representation function r, we define a *group mean translation* $\mathcal{T}^{G}_{r,\mathbb{G}\to\mathbb{G}'}(s)$ on a sequence s from orthologous group $\mathbb{G}$ to group $\mathbb{G}'$ as follows:

$$\mathcal{T}^{G}_{r,\mathbb{G}\to\mathbb{G}'}(s) \triangleq r(s) - \left[\frac{1}{|\mathbb{G}|}\sum_{s'\in\mathbb{G}} r(s')\right] + \left[\frac{1}{|\mathbb{G}'|}\sum_{s'\in\mathbb{G}'} r(s')\right]$$

This translation subtracts from $r(s)$ the averaged sequence representation across all sequences in orthologous group $\mathbb{G}$ and subsequently adds the averaged sequence representation across all sequences in group $\mathbb{G}'$. Intuitively, when $\mathbb{G}$ is the orthologous group containing sequence s, one might expect a representation that linearly encodes orthology and species information to better "recover" the sequence belonging to the same species and group $\mathbb{G}'$. We formally quantify this as *top-k recovery rate*, which is the probability with which a sequence undergoing group mean translation from its orthologous group $\mathbb{G}$ to a randomly selected group $\mathbb{G}' \neq \mathbb{G}$ recovers amongst its top k nearest neighbors (under L^2 norm in representation space) the sequence corresponding to the same species and to orthologous group $\mathbb{G}'$.

Orthologous species translation For completeness, the analogous operation of *species mean translation* $\mathcal{T}^{S}_{r,\mathbb{S}\to\mathbb{S}'}(s)$ on a sequence s from species $\mathbb{S}$ to species $\mathbb{S}'$ is defined as follows:

$$\mathcal{T}^{S}_{r,\mathbb{S}\to\mathbb{S}'}(s) \triangleq r(s) - \left[\frac{1}{|\mathbb{S}|}\sum_{s'\in\mathbb{S}} r(s')\right] + \left[\frac{1}{|\mathbb{S}'|}\sum_{s'\in\mathbb{S}'} r(s')\right]$$

Here, the top-k recovery rate is defined as the probability with which a sequence undergoing species mean translation from its species $\mathbb{S}$ to a randomly selected species $\mathbb{S}' \neq \mathbb{S}$ recovers amongst its top k nearest neighbors the sequence corresponding to the same orthologous group and to different species $\mathbb{S}'$.

Implementation We used the Faiss similarity search engine (Johnson et al., 2017) to calculate the top-k recovery rates for both types of translations, for $1 \leq k \leq 25$.

A.9 Secondary structure projections

For the analysis in Section 5.1, we derived and evaluated optimal linear projections of per-residue representations as follows.

Secondary structure data For secondary structure experiments, we used established secondary structure training and evaluation datasets from Zhou & Troyanskaya (2014). The authors obtained a set of structures with better than 2.5Å resolution, with no two proteins having sequence similarity above 30%. The training set was further filtered to remove sequences with greater than 25% identity with the CB513 dataset (Cuff & Barton, 1999). This training dataset, labeled `5926_filtered` by the authors, contains the amino acid identities, Q8 (8-class) secondary structure labels, family-level frequency profiles, and miscellaneous other attributes for 5365 protein sequences.

For the test dataset, we used the CB513 dataset (Cuff & Barton, 1999), also as preprocessed by Zhou & Troyanskaya (2014). The only inputs that were provided to the network were raw sequences, and the outputs regressed were Q3 and Q8 secondary structure labels.

Linear projection technique For each representation type, the optimal linear projection for secondary structure prediction was derived via multiclass logistic regression from the per-residue representation vectors to the secondary structure labels, with residues aggregated across all sequences in the training dataset.

Single-family data and analysis For each of the three domains used, we extracted all domain sequences from the Pfam dataset (Bateman et al., 2013) and located the subset of PDB files containing the domain, using the latter to derive ground truth secondary structure labels (Kabsch & Sander, 1983).

Pre-training follows the methodology of Appendix A.4, except that the datasets were from the domain sequences rather than Uniparc sequences. The domain sequences were randomly partitioned into training, validation, and testing datasets. For each family, the training dataset comprises 65536 sequences, the validation dataset comprises either 16384 sequences (PF00005 and PF00069) or 8192 sequences (PF00072), and the test dataset comprises the remainder.

Each Pfam family also forms an evaluation (projection) dataset; from the sequences with corresponding crystal structures, the training dataset comprises 80 sequences and the test dataset comprises the remainder.

A.10 Fine-tuning procedure

After pre-training the model with unsupervised learning, we can adapt the parameters to supervised tasks. By passing the inputs through our pre-trained model, we obtain a final vector representation of the input sequence h. During pre-training, this representation is projected to a predicted distribution l. Recall that a softmax over l represents the model's posterior for the amino acid at that position. These final representations are fine-tuned in a task-dependent way.

A.10.1 Residue-residue contacts

Residue-residue contact data For the analysis in Section 5.2, the dataset was derived from the S40 non-redundant set of domains in CATH (Orengo et al., 1997). The S40 subset has only those domain sequences where any pair of domains shares $< 40\%$ sequence identity. The contacts and sequences for each domain were extracted using the PDB files provided by CATH. The total number of files in this subset is $30,744$. Contacts were extracted following a standard criterion used in the literature (Jones et al., 2011), where a contact is defined as any pair of residues where the C$-\beta$ to C$-\beta$ distance (C$-\alpha$ to C$-\alpha$ distance in the case of glycine) is < 8 Å, and the residues are separated by at least 4 amino-acids in the original sequence.

We split non-contiguous protein chains into contiguous subsequences, and then filtered out sequences with less than 50 residues or more than 768 residues. This resulted in $32,994$ total data points (pairs of sequences and contact maps). We constructed train, validation, and test sets using these data points. We ensured that for each CATH domain which resulted in multiple contiguous subsequences, all the subsequences were confined to a single set (either train, valid or test). First, we randomly sampled a 10% subset of all sequences to obtain a test set. This resulted in 3339 test examples. The remaining sequences not assigned to test were filtered for similarity to the test sequences to produce the train and validation examples. All sequences that are similar to any sequence assigned to the test set were removed from the train and validation partition. To identify similar sequences, we used BLAST with an e-value of 0.1 and maximum sequence identity threshold of 20%. Using this filtering criteria, we ended up with $18,696$ remaining data points, which were randomly split into train and validation sets in the ratio $9:1$. This resulted in $16,796$ train $1,900$ validation examples.

To construct the 10% subset of the full fine-tuning dataset, we randomly sampled 10% of the train sequences to derive a smaller training set which had $1,557$ examples. The validation and test sets for the 10% training dataset are the same as used for the complete contact training dataset.

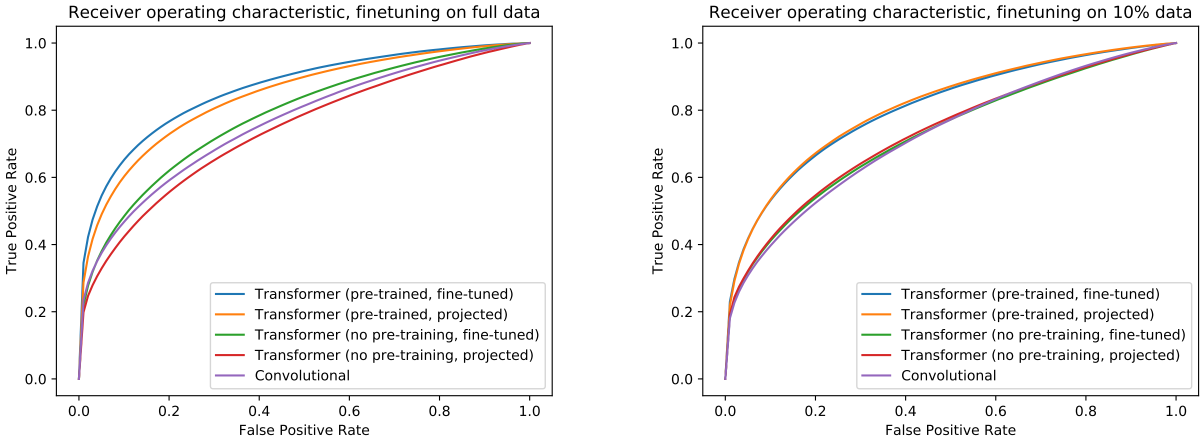
Fine-tuning procedure To fine-tune the model to predict residue-residue contacts, the final representation h was projected to a 512 dimensional vector. Since a $N \times N$ contact map has to be predicted, the $N \times 512$ dimensional tensor is first split into two tensors, each with shapes $N \times 256$. This is followed by a matrix outer product of the split tensors to obtain a $N \times N$ tensor. We found rescaling the predicted tensor by dividing by $\frac{1}{\sqrt{256}}$ to be helpful in achieving higher accuracy.

A.10.2 Mutagenesis

Mutagenesis data For the analysis in Section 6, we used a collection of 21,026 variant effect measurements from nine experimental mutagenesis datasets. We used the data originally collected, collated and normalized by Gray et al. (2018). They used this data to train a machine learning model based on features that incorporate expert knowledge of relevant biochemical, structural, and evolutionary features. We use this model as a baseline.

Fine-tuning procedure To fine-tune the model to predict the effect of changing a single amino acid (e.g. for mutagenesis prediction), we regress $\frac{l_{\pi(\text{mt AA})}}{l_{\pi(\text{wt AA})}}$ to the scaled mutational effect where π maps an amino acid to its corresponding index; mt AA is the mutant amino acid; and wt AA is the wildtype amino acid. As an evaluation metric, we report the Spearman ρ between the model's predictions and known values.

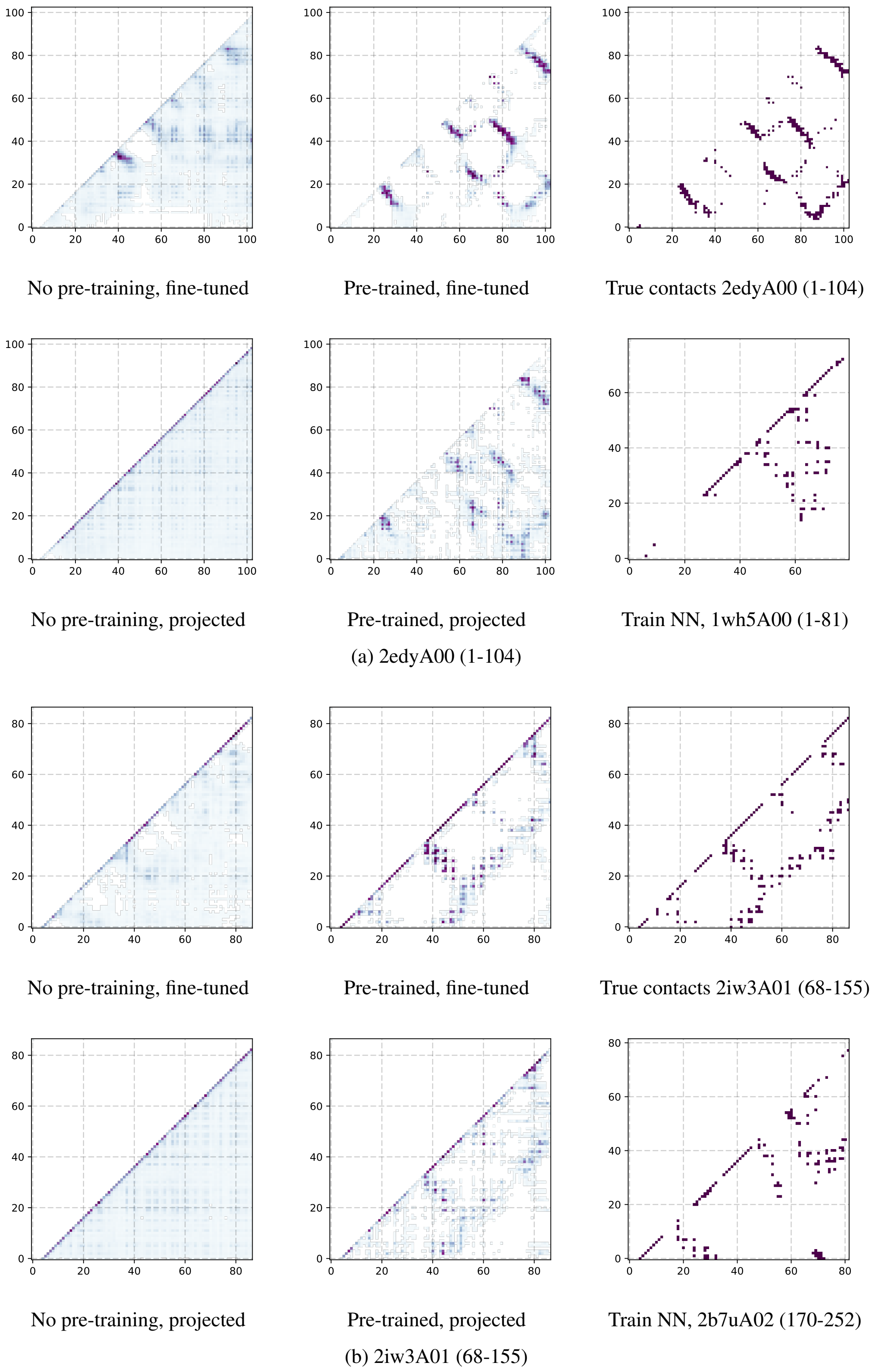
B Supplementary results



(a)

(b)

Figure S1: Fine-tuned and pre-trained language models predict residue-residue contacts consistently better than other baselines. Average ROC curves are depicted for the fine-tuned Transformer model and other baselines which are trained using (a) full dataset, and (b) 10% of the dataset. The ROC curves for each model are averaged over all test dataset sequences.



No pre-training, fine-tuned

Pre-trained, fine-tuned

True contacts 2edyA00 (1-104)

No pre-training, projected

Pre-trained, projected

Train NN, 1whA00 (1-81)

(a) 2edyA00 (1-104)

No pre-training, fine-tuned

Pre-trained, fine-tuned

True contacts 2iw3A01 (68-155)

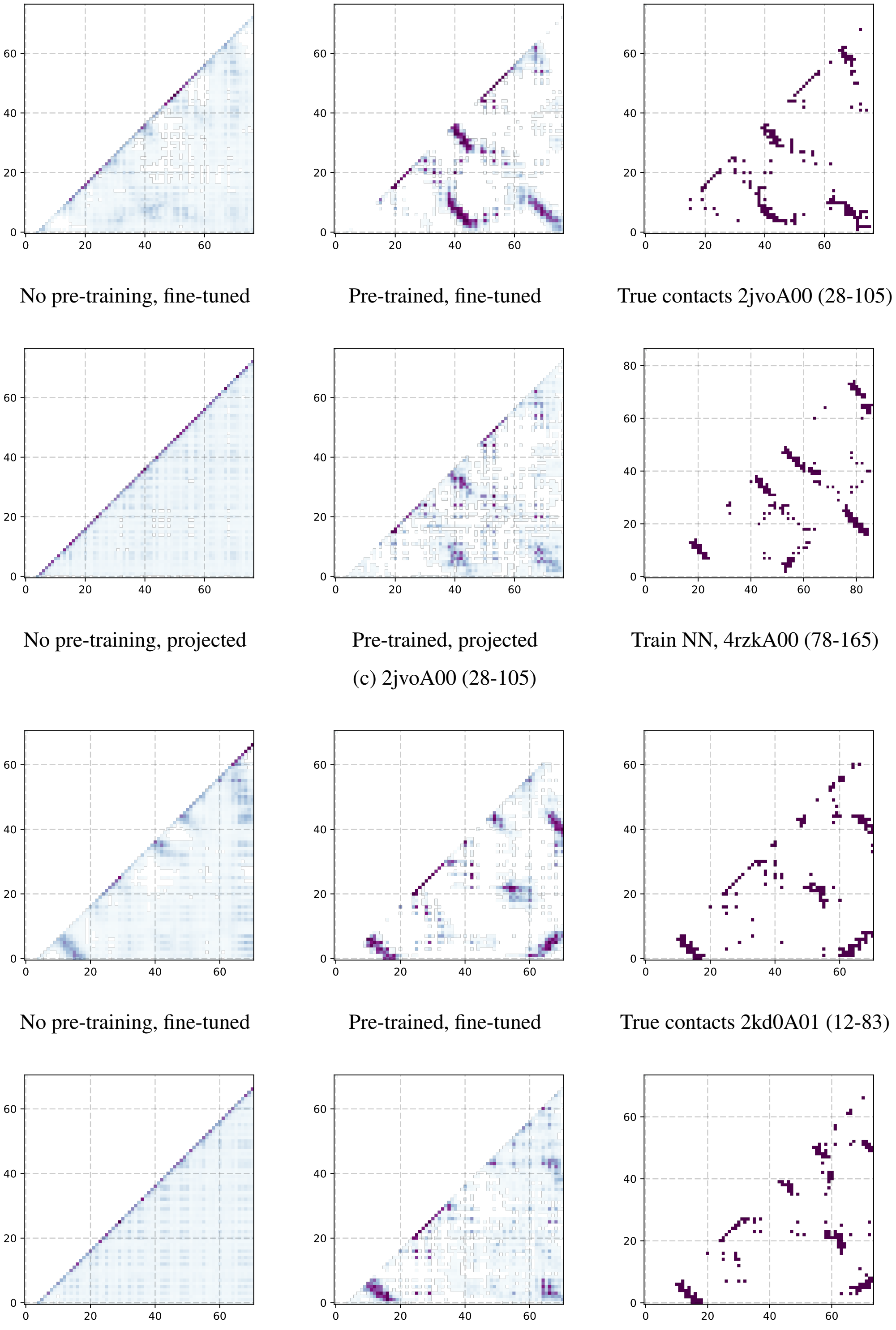
No pre-training, projected

Pre-trained, projected

Train NN, 2b7uA02 (170-252)

(b) 2iw3A01 (68-155)

Figure S2: Predictions for a selection of CATH domains in the test set with diverse folds. Predictions of the fine-tuned models are compared with and without pre-training, along with projections of the final hidden representations with and without pre-training. For each test sequence, the nearest neighbor by sequence identity in the training dataset is shown. Transformer models are able to generalize supervision from contacts and do not trivially copy the contact pattern of the nearest neighbor sequence.



No pre-training, fine-tuned

Pre-trained, fine-tuned

True contacts 2jvoA00 (28-105)

No pre-training, projected

Pre-trained, projected

Train NN, 4rzkA00 (78-165)

(c) 2jvoA00 (28-105)

No pre-training, fine-tuned

Pre-trained, fine-tuned

True contacts 2kd0A01 (12-83)

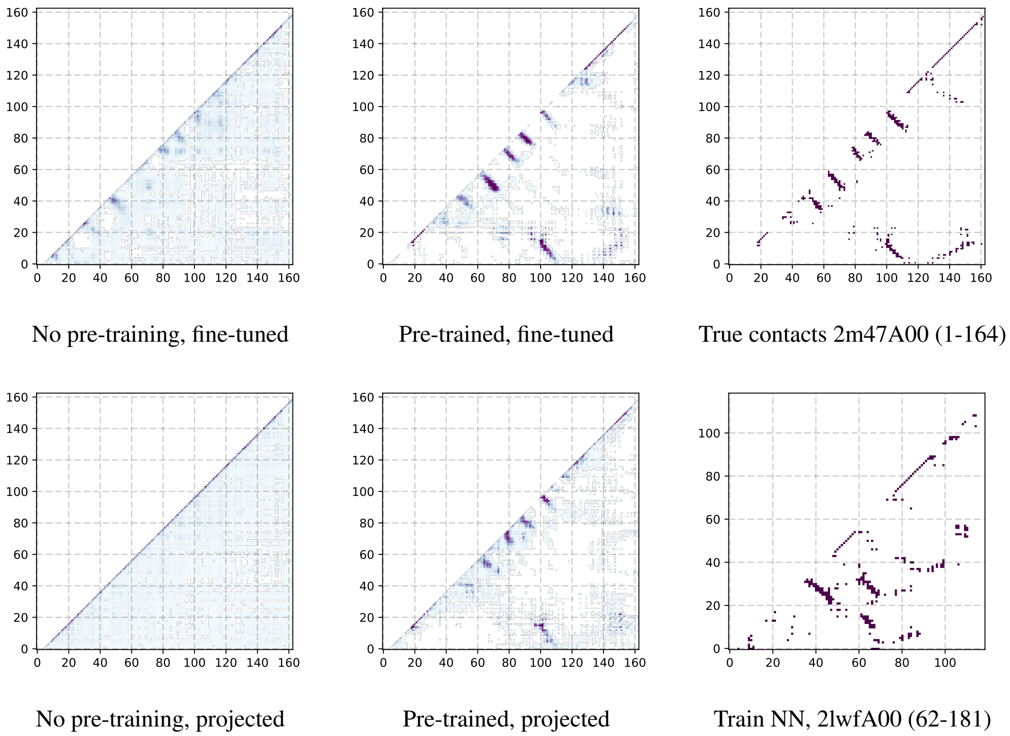
No pre-training, projected

Pre-trained, projected

Train NN, 1dgjA01 (1-75)

(d) 2kd0A01 (12-83)

Figure S2: continued from above.



No pre-training, fine-tuned

Pre-trained, fine-tuned

True contacts 2m47A00 (1-164)

No pre-training, projected

Pre-trained, projected

Train NN, 2lwfA00 (62-181)

(e) 2m47A00

Figure S2: continued from above.

	Q8		**Q3**	
Representation type	**Train (cullpdb)**	**Test (CB513)**	**Train (cullpdb)**	**Test (CB513)**
Amino acid identity (prior)	0.375	0.348	0.490	0.488
PSSM	0.453	0.421	0.577	0.566
PSSM + amino acid identity	0.454	0.422	0.577	0.566
12-layer Transformer (untrained)	0.420	0.390	0.523	0.519
24-layer Transformer (untrained)	0.418	0.393	0.521	0.520
36-layer Transformer (untrained)	0.417	0.388	0.519	0.515
12-layer Transformer (full dataset)	0.624	0.581	0.752	0.731
24-layer Transformer (full dataset)	0.636	0.592	0.765	0.740
36-layer Transformer (1% dataset)	0.632	0.587	0.760	0.737
36-layer Transformer (full dataset)	**0.655**	**0.608**	**0.782**	**0.755**

Table S1: Top-1 secondary structure prediction accuracy by an optimal linear projection of per-residue representations on the dataset from Zhou & Troyanskaya (2014). For Transformer models, the pre-training dataset is stated in parentheses. Q8 denotes 8-class prediction task, Q3 denotes 3-class prediction task.

	1% data		**10% data**		**30% data**		**50% data**		**80% data**	
Protein	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
Yap65 (WW domain)	0.343	0.049	0.461	0.104	0.619	0.056	0.704	0.050	0.706	0.059
Ubiquitin (E1 activation)	0.303	0.085	0.570	0.021	0.682	0.035	0.742	0.013	0.754	0.033
Ubiquitin	0.447	0.033	0.639	0.011	0.798	0.014	0.841	0.011	0.874	0.027
Protein G	0.331	0.110	0.656	0.070	0.801	0.035	0.847	0.028	0.870	0.035
Pab1 (RRM Domain)	0.427	0.035	0.584	0.036	0.734	0.023	0.796	0.012	0.801	0.030
PSD95 (pdz3 domain)	0.305	0.020	0.369	0.043	0.557	0.033	0.656	0.034	0.718	0.057
Hsp90	0.345	0.051	0.552	0.016	0.686	0.010	0.722	0.011	0.750	0.016
E4B (U-box domain)	0.099	0.052	0.175	0.043	0.312	0.019	0.397	0.039	0.474	0.043
Beta Lactamase	0.224	0.035	0.635	0.025	0.825	0.009	0.860	0.008	0.876	0.010
BRCA1 (E3 ligase activity)	0.086	0.041	0.243	0.024	0.353	0.010	0.397	0.019	0.448	0.047
BRCA1 (Bard1 binding)	0.264	0.099	0.372	0.034	0.438	0.013	0.466	0.045	0.485	0.032
Aminoglycoside kinase	0.330	0.039	0.609	0.011	0.744	0.003	0.795	0.004	0.838	0.004

Table S2: Mean and standard deviations of spearman ρ performance for the fine-tuned Transformer on intraprotein tasks. Performance was assessed by 5-fold cross-validation.