Improving Protein Function Annotation via Unsupervised Pre-training: Robustness, Efficiency, and Insights

David Dohan
Andreea Gane
Maxwell L. Bileschi
David Belanger
Lucy Colwell

(ddohan,gandreea,mlbileschi,dbelanger,lcolwell)@google.com
Google Research

ABSTRACT

Recent work demonstrated a large ensemble of convolutional neural networks (CNNs) outperforms industry-standard approaches at annotating protein sequences that are far from the training data. These results highlight the potential of deep learning to significantly advance protein sequence annotation, but this particular system is not a practical tool for many biologists because of the computational burden of making predictions using a large ensemble. In this work, we fine-tune a transformer model that is pre-trained on millions of unlabeled natural protein sequences in order to reduce the system's compute burden at prediction time and improve accuracy. By switching from a CNN to the pre-trained transformer, we lift performance from 73.6% to 90.5% using a single model on a challenging clustering-based train-test split, where the ensemble of 59 CNNs achieved 89.0%. Through extensive stratified analysis of model performance, we provide evidence that the new model's predictions are trustworthy, even in cases known to be challenging for prior methods. Finally, we provide a case study of the biological insight enabled by this approach.

CCS CONCEPTS

• **Computing methodologies** → *Neural networks*; • **Applied computing** → *Bioinformatics*.

KEYWORDS

deep learning, bert, pfam; protein families, protein function annotation, bioinformatics, neural networks, unsupervised learning

ACM Reference Format:

David Dohan, Andreea Gane, Maxwell L. Bileschi, David Belanger, and Lucy Colwell. 2021. Improving Protein Function Annotation via Unsupervised Pre-training: Robustness, Efficiency, and Insights. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore*. ACM, New York, NY, USA, 10 pages. <https://doi.org/10.1145/3447548.3467163>

This work is licensed under a Creative Commons Attribution International 4.0 License.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

© 2021 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-8332-5/21/08.

<https://doi.org/10.1145/3447548.3467163>

1 INTRODUCTION

Modern biology relies on an array of computational tools to annotate the rapidly growing corpora of biological sequence data made accessible by modern sequencing technologies [12]. At least one third of all known protein sequences cannot be reliably annotated, therefore systems to automatically and accurately infer their function are critical to our understanding of life [40]. Accurately predicting the function of newly discovered protein sequences can reveal, for example, enzymes that catalyze previously unknown reactions, natural product molecules with novel antimicrobial activity, or new molecular reagents for use in biotechnology [32, 63]. However, existing protein annotation pipelines are limited by the capacity and expressivity of the available tools.

To address these limitations, researchers have introduced deep learning models that are capable of capturing higher-order interactions between positions in an amino acid sequence. Such interactions are not accessible to models used in bioinformatics such as profile hidden Markov models (pHMMs) [14, 24, 44, 49, 55, 58]. Deep models have demonstrated outstanding performance on tasks such as protein function prediction and protein structure prediction [7, 8, 39, 48, 64]. In particular, we recently demonstrated that an ensemble of convolutional neural networks outperforms the state-of-the-art HMM-based tool HMMER at assigning Pfam labels to protein domains, while not requiring access to sequence alignments [7].

Despite their empirical success, deep learning methods have not achieved widespread use by practitioners. To address concerns expressed by potential users we investigate: (1) how to optimize performance using deep learning, particularly for hard examples distant from the training set (2) how to deploy a deep learning system that is practical for biologists, and (3) how to convince users that our model will behave reliably on sequences that evade existing approaches.

This paper describes our efforts to address these challenges by building better, more efficient models and by scrutinizing them more carefully. Transformer language models [59], such as BERT [13], provided groundbreaking accuracy improvements in natural language processing. These models are first fit on large unlabeled corpora and then fine-tuned for particular downstream tasks using small labeled datasets. Further work has demonstrated that such models trained on protein sequences capture biologically relevant properties [34, 45, 47].

We trained a BERT model across 27 million protein sequences and fine-tuned using the same labeled train set as Bileschi et al. [7]. This two-step approach increased accuracy of a single model from 73.59% for the best CNN to 90.5% for the fine-tuned transformer, which also outperforms our original best-performing model, an ensemble of 59 CNNs. Notably, in the absence of pre-training the transformer provides comparable performance to the single CNN model. By transitioning from a large ensemble of CNNs to a single transformer, we considerably reduce the computational burden of our tool. This is an important step towards releasing a web-based tool, along the lines of InterProScan or BLASTp [4, 27], key components of researcher workflows.

Finally, we carefully analyzed our new model to ensure the model meaningfully generalizes far from the training data. We present stratified analyses that isolate performance on examples far from the train data and examples that belong to rare classes, and we highlight examples where baseline algorithms tend to fail. Finally, we provide a case study describing a laborious, but important, procedure where we manually verify disagreements between our model and the ground truth by leveraging and combining alternative data sources and protein analysis tools.

With this new model and our systematic analyses, we hope to build additional practitioner confidence in the promise of practical deep learning methods to revolutionize the field of protein function annotation.

2 BACKGROUND AND RELATED WORK

2.1 Terminology

A protein is composed of a sequence of *amino acid residues*; there are 20 commonly occurring natural amino acids. In this work, we develop models that map directly from the amino acid sequence of a protein domain to a class label describing its functional role, and rely on the Pfam database to provide these labels. These labels describe the function of a protein domain, such as the type of reaction that an enzyme catalyzes.

Remote homology detection is the task of predicting whether two protein sequences have shared evolutionary ancestry, even if this common ancestry is ancient.

2.2 Deployment Requirements

When given a new protein sequence, a biologist has an array of tools available to provide information about its functional activity. The usability and feasibility of deploying a new tool relies on three aspects:

- Prediction accuracy: How reliable are the predictions?
- Ease of sharing: Does using the software require large up-front downloads?
- Throughput: How long does it take to annotate an individual sequence, and how does the software scale to large batches of sequences?

Most tools allow users to submit one or more query sequences to a web portal that returns either similarities to annotated sequences, or computationally-predicted annotations. These tools have been widely adopted due to their ease of use: computation runs on a remote server and thus the user has little setup. However, since most

methods depend on computing many sequence-sequence alignments or performing prediction across large sets of HMMs, these remote annotation tools do not scale to rapid annotation of thousands of sequences.

The most longstanding solution, BLASTp [4], is highly popular with molecular biologists and allows one or more protein sequences to be searched interactively against a number of large databases. Anecdotally, web-based BLASTp searches take > 10s to return results for each query, though for sequences with high similarity to the training set, Quick BLASTp works very quickly [1]. Other acceleration strategies have also been published [10, 16, 29, 30, 53]. InterProScan [27] is a related tool that integrates many different sets of pHMM-based annotations. InterProScan is typically used in a cluster setting, without GPU acceleration. Locally installing tools such as InterProScan presents two challenges: the size of the model/reference dataset, and the speed at which protein sequences can be annotated.

2.3 Transformer Sequence Models

In natural language processing, transformer models [59] have become the standard deep learning architecture for working with sequences. They maintain a hidden state for each position in the sequence and update these states by alternating two types of neural network layers: (1) an *attention mechanism* aggregates information across the sequence and (2) a multi-layer perceptron locally updates each hidden state.

Figure 1: Standard pre-training approach. Pretrained weights (BERT, left) are used to warm-start the transformer classifier’s embedding network (right). On the left, the goal is to predict the masked residues, while on the right, the objective is to predict the class label given the unaltered sequence.

A key advantage of transformers vs. convolutional neural networks (CNNs) or recurrent neural networks (RNNs) is that the direct pairwise interactions between positions provided by the attention mechanism are better at capturing long-range interactions. Additionally, the attention mechanism is interpretable, as one can use the activations of the attention matrix to visualize the learned dependencies, or their gradients in the backward pass for credit assignment on a prediction.

While transformer classifiers can be trained *from scratch*, starting with randomly initialized weights, unsupervised pre-training has become a core technique for improving their performance in

natural language processing systems. Standard transformers [59], when trained in an unsupervised fashion as in BERT [13], can help warm-start supervised learning methods and substantially improve performance on downstream tasks such as question-answering and summarization [13, 25, 41] by effectively increasing the fine-tuning dataset size [23]. When using a pre-trained model for classification, only the parameters of the final classification layer are initialized randomly, and all other parameters are initialized to the pre-trained model weights (Figure 1).

2.4 Deep Learning for Proteins

Alley et al. [2] pretrained an mLSTM model [31] on the UniProt database [12] and demonstrated that the learned representations have characteristics that reflect known biological properties. For protein engineering tasks, the authors fine-tuned the representation on sequences that are evolutionarily related to the target, and trained a supervised model on the resulting representations. The proposed approach demonstrated exciting performance across multiple protein engineering tasks. Furthermore, Biswas et al. [8] showed that the fine-tuning procedure of Alley et al. [2] can be successful in engineering tasks with as few as 96 labeled examples.

For evaluating the effect of pretraining on downstream tasks, Rao et al. [44] introduced TAPE, which assesses pretrained models using tasks including structure prediction, remote homology detection, and prediction of phenotypes that have been used in protein engineering. On these tasks, pretraining improved performance across multiple model architectures, though follow-up work suggests that similar results could be achieved without pretraining [50]. An additional line of work [33, 61, 62] benchmarks various protein sequence embeddings on the CAFA benchmark [64], though BERT and related architectures are not considered.

In addition, Rives et al. [47] combined BERT with task-specific fine-tuning to accurately predict contact maps, secondary structure, and mutation effects, while [60] focuses on how attention aligns with structural features. Remarkably, the attention mechanism in a pre-trained transformer model was shown to accurately capture residue-residue interactions [45] in the absence of significant supervision, a result that holds exciting promise for further advances in tertiary and quaternary protein structure prediction. Finally, transformers have also been shown to generate novel protein sequences with desirable characteristics [34], while Sturmfels et al. [56] analyzed a new pre-training task – profile prediction – and showed promising results. Taken together, these results highlight that deep learning methods have considerable potential to revolutionise biological sequence analysis.

3 EXPERIMENTAL METHODS

3.1 Evaluation Datasets

Pfam, the Protein Family Database, is a set of trusted protein function annotations curated by a team at EBI, the European Bioinformatics Institute, through a combination of human expertise coupled with predictions made by Hidden Markov Models (HMMs). The machine learning task for this dataset is to predict the appropriate Pfam family given the amino acid sequence of a protein.

The database consists of a large collection of protein families, where each family is described by a single HMM. A carefully-curated representative set of seed sequences for each family is

used to build a seed alignment, which is then used by the HMMER3 package to fit an HMM for each family [15]. These HMM models are then used to annotate protein domain sequences from a large sequence database from the UniProtKB reference proteomes [12, 37]. Version 32 of Pfam contains 17,929 families and provides ~54 million functional domain annotations. We use the well-curated and representative seed sequences (used to fit the HMMs) to benchmark the performance of different machine learning models. Section 3.1.2 describes two train-evaluation splits of this data: an i.i.d. (random) split, and a clustered split.

3.1.1 Pfam Clans. In Pfam, a clan is a set of evolutionarily-related protein families. A protein domain may belong to zero or one clan. For families that belong to clans, the ground-truth annotation of which specific family within the clan a sequence belongs to is less clear cut. In some cases, multiple families exist because a single HMM model does not have enough capacity to capture the full diversity of a single protein family [20, 21]. Moreover, sequence membership of closely related families is sometimes resolved using a special clan competition workflow [37]. Reporting accuracy at the lifted clan level avoids emphasizing these known limitations of HMMs. Around 55% of Pfam sequence domains do not belong to a clan because their family is not homologous to other Pfam families [17]. For these families, the *lifted clan* annotation simply retains the family label.

3.1.2 Splits of Pfam Seed. Protein sequences are related through their shared evolutionary history. This means that while some sequences are very different, others are extremely similar. *Remote homology detection* refers to the ability of the trained model to accurately annotate test examples for which the nearest training example is distant in sequence space. Implicit in this definition is a distance function: different distance functions will give rise to slightly different remote homologs, though most top-performing methods rely on a measure tied to sequence identity [3] or profile HMM alignment [14, 46].

For applications like remote homology detection, the ability of a model to accurately annotate sequences that are similar to the training set is not relevant. As we focus on this task, we focus on a clustered split in which each family is split following Eddy [15]: sequences in the held-out test set are each at most 25% identical to any sequence from the training set. In addition, to measure model performance across test sequences that cover a broad range of distances from the training set, we report performance using a random (i.i.d.) split of each family. The exact splits used are the random and clustered Pfam seed split provided by Bileschi et al. [7].

3.2 Baselines

3.2.1 Top pick HMM. Pfam uses HMMER to fit a pHMM to the seed sequences for each protein family [37]. For this baseline, we use HMMER to refit these pHMMs to the train subset of sequences from each family seed. This means that we work with pHMMs trained on a subset of the data used by Pfam. To avoid artifacts from realignment, and enable optimal baseline performance, we retain the alignment of the original Pfam seed. In contrast, no alignment information is present in the data used to train the deep learning

models and other baselines. Pfam uses hand-tuned *gathering thresholds* to determine whether a sequence belongs to a given family. To avoid re-tuning the gathering thresholds and undermining HMMER performance we predict family membership by determining which pHMM assigns the highest bit score, a normalized version of HMM likelihood, to a query sequence. Following Bileschi et al. [7], we name this baseline *Top pick HMM* in our experiments below, and used `hmmbuild` from HMMER 3.1b to construct a pHMM from the aligned training sequences for each family. Each unaligned text sequence was then searched against all 17,929 profiles using `hmmsearch --max`. Note that we do not expect to achieve 100% accuracy because the training data is a subset of Pfam seed.

3.2.2 BLASTp. BLASTp [3] uses a nearest-neighbors approach to functional characterization, relying on pairwise local alignment to compare a protein query sequence to each sequence in a database. Sequences are ranked according to their similarity to the query sequence, allowing functional annotations for the query sequence to be inferred from pre-existing annotations of the top-ranked sequences. We first use `makeblastdb` (version 2.7.1+) to build a BLAST database from all sequences contained in the training set. We then query each test sequence against this database, using `blastp -query`, retaining only the top-scoring hit. This implementation returns no hit for 259 (0.21%) of the 126,171 sequences in the i.i.d.-split Pfam-seed test set, and 93 (0.44%) of the 21,293 sequences in the clustered test set.

3.2.3 ProtCNN and ProtENN. These deep learning approaches to protein sequence annotation use 1D convolutional neural networks with residual connections (ResNets) [7]. Convolutional networks rely on translational invariance to handle the lack of alignment between sequences in the same Pfam class. The ProtCNN model is a single 5-layer ResNet with 51M parameters. Bileschi et al. [7] showed that ProtCNN outperforms existing approaches, including alignment-based methods, on multiple Pfam classification sub-tasks, making ProtCNN our highest-performing single-model baseline. Moreover, this work demonstrated that an ensemble of identical CNNs trained with different random initializations improves results even further, at the expense of computational performance. This is our strongest baseline, with 3 billion parameters (59 ensemble elements).

3.3 Masked Language Modeling Pre-training

3.3.1 Pre-training Data. For unsupervised pre-training, we use the UniRef50 dataset [57]. Several other datasets, such as TrEMBL [5], UniRef100, and BFD [52, 54], have been used for pre-training deep learning models. Some prior work has demonstrated that smaller datasets with reduced redundancy, such as UniRef50, result in improved performance on downstream tasks [47]. Relatedly, Elnaggar et al. [19] compared models trained on UniRef100 and BFD, and found that training on the larger dataset (2.12B sequences in BFD vs. 216M sequences in UniRef100) improved downstream task performance, but not significantly.

The UniRef50 dataset is obtained by clustering sequences from UniRef100 that have at least 50% sequence identity to the longest sequence in the cluster¹. UniRef50 retains a single representative

for each cluster and we split these representatives randomly into train/validation/test. The resulting dataset has high diversity among training examples and also between train and evaluation sets, but similar sequence length statistics (Table 5).

3.3.2 Pre-training Approach. When pre-training for downstream tasks, sequence models such as the transformer are typically trained to either predict the token at position t given the tokens at positions $1, \dots, (t-1)$ (causal language models) as in [42] or trained to fill in a percentage of tokens masked (masked language models) as in [13]. Following Rives et al. [47], we focus on the latter. This masking-based pre-training is commonly known as BERT in natural language processing.

During pre-training, the model’s objective is to predict the original sequence after observing a noisy version. Following standard BERT training guidelines, we apply a cross-entropy loss between the model predictions and ground truth token for 15% of the tokens in a batch. Within the selected sequence positions, inputs are modified to add *noise* as follows: 80% are replaced with a random non-padding token, 10% are set to the special [MASK] token, and 10% are left unchanged.

Method	#Params
Top pick HMM	194.0M
BLAST	207.8M
ProtCNN	51.0M
ProtENN	3000.0M
ProtTNN-12L	99.5M

Table 1: Number of parameters of various models trained on our clustered split of the data. The proposed model has a significantly smaller number of parameters than the baseline ProtENN. The ProtTNN-12L line reflects both from scratch and fine-tuned transformer variants.

Our main model, 12L BERT, uses the medium transformer model architecture from Rives et al. [47], with embedding dimension 768, 12 heads, 12 layers, query-key-value dimension 768, and MLP dimension 3072. The model contains 85.0M parameters during pre-training, and 99.5M with the addition of learnable Pfam family embeddings in the final classification layer during fine-tuning. Table 1 compares this model with the different baselines in terms of the number of free parameters. We also compared the performance of the 12 layer model with a much smaller transformer model, 1L BERT, that has single layer and 7.1M parameters.

We pre-train the model to perform masked language modeling across 27M training sequences from the UniRef50 sequence database. Training proceeds synchronously across 64 TPUv3 chips [28] with a batch size of 24 per chip, yielding a total batch size of 1536. Each batch consists of sequences padded or sampled to a length of 1024 tokens. We stop training after a million steps with no dropout (about 50 epochs), and observe that the per-masked-amino acid validation accuracy continues to improve.

We follow the standard approach to utilizing BERT models and split pre-training and fine-tuning into a separate stages. We note that it is possible to combine the pre-training and fine-tuning steps so that both sequence denoising and classification are learned simultaneously. However, this approach typically does not perform

¹<https://www.uniprot.org/help/uniref>

as well as separate stages [43], so we only present results with separate pre-training and fine-tuning here.

3.3.3 Pre-training Performance. Our decision to use pre-trained models was motivated by our initial experiments on Pfam classification in which the transformer classifiers trained from scratch achieved comparable accuracy to the CNN models. We report the pre-training performance in Table 2, which describes both the perplexity and the next token prediction accuracy. The perplexity obtained by the 12-layer model approaches the performance reported in Rives et al. [47] (i.e. 10.45, see Table 1 in the cited work). While we use the same model architecture, we attribute the performance difference to the hyper-parameters used for training, such as the number of epochs and the learning rate schedule. We note that additional hyper-parameter tuning will likely improve pre-training performance, which may also further improve the classification accuracy for downstream tasks.

Transformer	ECE	Accuracy	#Params
Empirical Baseline	17.88	9.64%	27
1L BERT	14.47	19.33%	7.1M
12L BERT	11.32	26.46%	85M

Table 2: Pre-training performance on the UniRef50 dataset. We include the perplexity (exponentiated cross-entropy: ECE), masked-token prediction accuracy and the number of trainable parameters.

The table also includes a simple empirical baseline, obtained by computing the frequency at which each amino acid is observed in the UniRef50 database and drawing amino acids according to this frequency distribution. In computing this baseline, we used a random subset of 1,024,000 sequences from the UniRef50 training set that we cropped to maximum length 1024 to match the data processing performed for the Transformer models. The results are similar to those reported in Madani et al. [34].

3.3.4 Finetuning. To use transformer models for classification, we replace the final per-residue classification layer, which maps per-residue hidden states to amino acid predictions, with a pooling layer across all residues (e.g. max-pooling as shown in Fig. 1). The pooled output is followed by a parameterized projection layer to compute the predicted probability that the sequence belongs to each output class. Any padding tokens are ignored within the model and when applying the pooling layer. The model is trained to minimize the cross-entropy loss between the model predictions and the ground truth class label.

We refer to the transformer models as ProtTNN-scratch, indicating that is trained to do classification from newly initialized weights, or ProtTNN-fine-tuned if it is fine-tuned to do classification from pre-trained protein BERT model weights.

4 RESULTS

We first present a performance comparison across transformer models of different sizes, both with and without pre-training, before comparing the performance of our best single model with the set of established baselines. Throughout, we focus on the clustered split of the data because this reflects robustness to the real-world use

case of hard-to-annotate sequences that are highly distinct from the existing corpus of well-studied proteins. Moreover, this split avoids the performance over-estimation that may result from near-duplicates between randomly split train and test sets due to the evolutionary process that generated the underlying data.

4.1 Model Performance

Table 3 reports the performance achieved by various transformer models at classifying held-out test sequences from the challenging clustered split. We note that performance varies substantially across these models. We demonstrate further a performance improvement via a majority voting ensemble² of three models fine-tuned from the same initial pre-trained weights, but with different hyperparameters. The hyperparameters are chosen as the top 3 performing settings based on a random sweep of the 1-layer model, evaluated on the validation split. We vary learning rate, label smoothing, weight decay, and dropout rate.

Model	Accuracy
ProtTNN-1L-Scratch	79.30
ProtTNN-1L-Fine-tuned	83.12
ProtTNN-12L-Scratch	74.27
ProtTNN-12L-Fine-tuned	90.50
ProtTNN-12L-Fine-tuned ensemble (3)	91.66

Table 3: Classification performance on clustered train-test split across transformer model variants, showing that pre-training provides the necessary boost to match the ProtENN performance. Ensembling 3 fine-tuned models further improves performance. Table 4 compares the top models to existing solutions on both the i.i.d. and clustered splits.

Overall, the small single layer model reports better performance in the absence of pre-training compared to the 12-layer model at 1 million steps. In contrast, we find that pre-training significantly improves downstream accuracy for the 12 layer model in particular. Notably the gap between the from-scratch and fine-tuned models increases with model size: 79.3 $\rightarrow$ 83.1 for a 3.8% boost in the 1-layer model, while 74.27 $\rightarrow$ 90.5 for a 16.23% boost in the 12-layer model.

Method	Accuracy	
	I.I.D. split	Clustered split
Top pick HMM	99.78	88.08
BLAST	99.13	70.14
ProtCNN	99.58	73.59
ProtENN	99.89	89.00
ProtTNN-12L-Fine-tuned	99.87	90.50

Table 4: Performance of the best pre-trained single transformer model compared with established baselines. The proposed model reports comparable performance to ProtENN on the i.i.d. split and significantly outperforms all baselines on the challenging clustered split ($p < 0.01$, 2-sided McNemar test). The clustered split is a better measure of robustness, as an i.i.d. split may contain evolutionarily related near-duplicates across splits.

²Ties are broken arbitrarily.

Table 4 compares the classification performance of our best-performing single transformer model with all baseline models across both the i.i.d. and clustered splits of the data. We find that the single 12 layer pre-trained transformer model is competitive with the strongest baseline model, ProtENN, on the i.i.d. split, and significantly outperforms this model on the challenging clustered split that reflects the central challenge of remote homology detection. We assess significance of the performance difference between the pre-trained model and baselines using a 2-sided McNemar test, and find that the single pre-trained transformer significantly outperforms all baselines with $p < 0.005$ on the clustered split. To further build confidence in this single model, we study the performance of the model along several axes.

4.2 Analysis of Model Performance

The field of protein function annotation is chiefly concerned with improving the accuracy of annotation tools on sequences that are difficult to annotate using existing tools. There are a number of reasons an example may be difficult to annotate; here we focus on three archetypes. First, examples that are distant from the training set – this is remote homology detection. Secondly, examples from small families, for which the model has limited supervision. Finally, examples that involve repeats – these are particularly difficult for HMM models to accurately annotate. Throughout, we study the accuracy of model predictions at the *lifted clan* level, which combines closely related families into higher order groupings or clans as described in Sec. 3.

Figure 2: Clustered split accuracy as a function of distance to nearest training example. ProtTNN matches or exceeds the performance of ProtENN.

4.2.1 Distance from nearest training example. We use the clustered seed split (see Sec. 3.1.2) to measure the ability of different models to generalize beyond the training set. Moreover, within this difficult test set we further stratify accuracy analysis by distance to the training data, so as to quantify the difficulty of each test example. For instance, a test example with 20% sequence identity to the closest member of the training set shares at most 20% amino acid identity with any training sequence within the same family (function), where percent identity is calculated using the Pfam seed alignment. Sequences below 25% sequence identity are considered

to be very difficult to annotate correctly [11]; our clustered split limits within-family sequence identity to a maximum of 25% following Eddy [15].

Fig. 2 shows that the transformer model matches or exceeds the accuracy of ProtENN across every tranche of our clustered split dataset. Notably, performance of all methods degrades substantially as maximum sequence identity decreases, in accordance with expectation. This helps assuage any concern that the choice of distance function confounds experimental observations. Bileschi et al. [7] explicitly compare BLAST sequence identity with Pfam seed multiple sequence alignment identity on the Pfam seed test set and find high correlation.

4.2.2 Amount of training data. A second determinant of test example difficulty is the amount of training data that is available for a given class. Across Pfam seed, the number of training examples per class varies dramatically. This issue is further exacerbated by the number of target classes: there are almost 18000 training families. The exacting requirements of the clustered split mean that just 3000 of these families appear in the test set, of which most have < 50 training examples per family, as shown in Fig. 3.

Figure 3: The scarcity of training data for many output classes makes this a fewshot learning problem. The x axis is truncated, with all larger classes falling into the rightmost bin.

For a remote homology search where there are few training examples, we find that ProtTNN outperforms the large CNN ensemble across essentially all training-set-size tranches of the test data. In particular, we see that in the most-difficult fewshot setting, where there are fewer than 17 training examples available per test example, ProtTNN achieves a large improvement over the CNN ensemble. Notably, ProtTNN is able to match or exceed the Top pick HMM performance for all but the smallest decile of family sizes. For this decile, while the Top pick HMM still outperforms both ProtENN and the transformer, the single pre-trained transformer model closes the performance gap by more than 50%.

Figure 4: Model performance as a function of the number of training sequences for each of the ~3000 families that have members in the clustered split test set. Bins are deciles where each contains the same number of families.

4.2.3 Protein family type. Each Pfam family is also classified as one of the following types: family, domain, repeat, motif, coiled-coil, and disordered. Of most interest to this study is the repeat type.

Repeats are typically shorter sequence segments that tend to appear more than once in tandem. They are functionally diverse building blocks that are short, are unstable in isolation, and have relatively low conservation between tandem copies [18].

These structures appear in 14% of all annotated proteins [35], and are important in many disease mechanisms [38], including for example the mechanisms involved in brain cancer [18]. Because of their importance, many profile HMMs have been specifically built to detect repeats, though their creation is largely a manual process that requires expert hand-tuning of boundaries and model parameters, as discussed below.

Figure 5: The transformer model addresses a known weakness of the currently-deployed HMMER system: short sequences, and in particular *repeated* short sequences with low conservation are difficult for HMMER to classify correctly. We show this on the i.i.d. split to highlight the difficulty HMMER has on the *easier* task of the i.i.d. split.

Part and parcel to using pHMMs for remote homology detection is creating a multiple sequence alignment (MSA). This requires the

alignment of homologous pairs of residues among all sequences in the family. For repeats, there is relatively little *signal* (repeats are typically short sequences) and a large amount of noise, caused by substantial variation in terms of pairwise sequence identity between different copies of the repeat element [18]. These factors together mean that creating an MSA is a very difficult. Manual oversight and adjustment is often required to achieve an acceptable MSA. Indeed, even determining the start and end of a repeat region is a difficult process that dramatically affects the ability of the pHMM fit using the alignment to detect remote homologs.

As a result, defining one profile per instance of a repeat is difficult in practice. This has led to both many profiles being generated per function (e.g. Pfam TPR clan CL0020 has entry TPR_2, which is defined as TPR repeats not matched by TPR_1), and profiles containing more than one repeat instance so as to maximize recall of these difficult regions (e.g. PF13306 includes 6 copies of the Leucine rich repeat region) [17].

Neural network methods have great promise in repeat-sequence annotation, as shown in Fig. 5.

4.2.4 Sequence length. Finally, in Fig. 6 we stratify model performance on the clustered split by sequence length, where the length of each held-out test sequence is simply the number of amino acids contained in the region annotated by Pfam - this means that different test sequences within the same family can have quite different lengths. It is known that the HMM model struggles to accurately classify sequences with short lengths, and this result is recapitulated in Fig. 6.

Figure 6: Accuracy on the clustered split as a function of sequence length. The transformer matches the ensemble, and both substantially outperform HMMER on short sequences.

In this challenging region, the ProtTNN is able to improve upon the performance of ProtENN, significantly outperforming the single CNN model. To some extent, these results also reflect family type, because, as mentioned above, repeat families often contain shorter sequences. The results shown in Fig. 6 provide another view into the promise that neural networks hold to be able to address biases in the performance of current production-methods at protein sequence annotation.

5 BIOLOGICAL INSIGHTS

The trustworthiness and prevalence of adoption of the annotations in Pfam-seed provide a strong motivation to use this dataset for

machine learning tasks. To build additional understanding in the predictions made by the single pre-trained transformer model, it is therefore instructive to look more closely at places where this model confidently disagrees with current annotations. To this end, we examine protein Q88J38, an uncharacterized protein from the saprotrophic soil bacterium *Pseudomonas putida*, whose ground truth Pfam label is DUF1302. A domain of unknown function (DUF) such as this is Pfam family that contains evolutionarily related sequences that currently have unknown function. More than 20% of Pfam families are annotated as DUFs [17]. Indeed, more specific annotation of DUFs is an important research direction [40] where we anticipate neural networks will be able to make significant contributions.

Pfam seed models classify Q88J38 (residues 24-547) as DUF1302 with an extremely high confidence score of 473 bits (a measure of the log odds of the sequence under the DUF1302 pHMM over the null hypothesis), whereas the transformer classifies this sequence as a TonB dependent receptor (PF00593) with a probability of 0.99. The TonB system is known to control transport of a biphenyl compound derived from lignin across the outer membrane [22]. While a BLAST search for sequence homologs with > 90% sequence identity to Q88J38 turns up only similarly *unannotated* sequences, the hierarchical EggNOG database v5.0 predicts that this sequence domain belongs to cluster of orthologous groups COG3203, which has the functional annotation “formation of passive diffusion pores that allow small molecular weight hydrophilic materials across the outer membrane”, commensurate with the known function of TonB, i.e. agreeing with the novel ProfTNN prediction [26].

Figure 7: PSIPRED predicts a secondary structure that agrees with the predicted beta barrel of Q88J38.

The Pfam TonB family PF00593 includes only the conserved beta-barrel structure of the larger TonB complex [9], so it’s reasonable to ask whether Q88J38 has a beta barrel structure. While the structure of Q88J38 has not yet been experimentally determined, Fig. 7 shows the secondary structure prediction made from sequence by PSIPRED [36] revealing a high confidence prediction of mostly beta sheets, as would be expected for a beta-barrel tertiary structure.

We note that Pfam reports no profile-profile similarity between the DUF and the TonB families via HHsearch [51] and low SCOOP [6] score (0.017). HHsearch provides model-model comparisons which are used to identify family homology with very high sensitivity, whereas SCOOP compares the outputs of two models to provide a measure of family-family similarity. The lack of profile-profile similarity strongly suggests that ProtTNN does not *misclassify* Q88J38 as TonB simply because of high similarity or well-known homology between DUF1302 and TonB, but rather ProfTNN may have

identified distant homology that is beyond the detection threshold of profile HMMs. This suggests the transformer is learning information that these state-of-the-art methods are unable to glean.

6 CONCLUSION

Improvements in computational sequence annotation and remote homology detection are required to enable us to leverage the ever-growing corpora of biological sequence data. Deep learning models offer an exciting path forward; however, the painstaking experimental verification required to truly validate the predictions made by new models can only be tackled by a large and distributed network of molecular biologists, many of whom are not equipped to serve large deep models locally. This means that providing widespread access to inference using these models via a web-based API is a significant priority. In this paper, we address a key requirement for such deployment by leveraging pre-training across millions of unlabeled sequences to improve the state-of-the-art reported performance, while significantly reducing the compute burden by avoiding the need to train and serve a large ensemble of networks. The performance of these pre-trained models will thus facilitate user access via a deep learning system that is practical to run inter-actively, enabling them to be subject to rigorous assessment and potential adoption by a broad community.

REFERENCES

- [1] 2018. Database resources of the national center for biotechnology information. *Nucleic acids research* 46, D1 (2018), D8–D13.
- [2] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. 2019. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods* 16, 12 (2019), 1315–1322.
- [3] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. 1990. Basic local alignment search tool. *Journal of molecular biology* 215, 3 (1990), 403–410.
- [4] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. 1997. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic acids research* 25, 17 (1997), 3389–3402.
- [5] Amos Bairoch and Rolf Apweiler. 2000. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic acids research* 28, 1 (2000), 45–48.
- [6] Alex Bateman and Robert D Finn. 2007. SCOOP: a simple method for identification of novel protein superfamily relationships. *Bioinformatics* 23, 7 (2007), 809–814.
- [7] Maxwell L Bileschi, David Belanger, Drew Bryant, Theo Sanderson, Brandon Carter, D Sculley, Mark A DePristo, and Lucy J Colwell. 2019. Using deep learning to annotate the protein universe. *bioRxiv* (2019), 626507.
- [8] Surojit Biswas, Grigory Khimulya, Ethan C Alley, Kevin M Esvelt, and George M Church. 2020. Low-N protein engineering with data-efficient deep learning. *BioRxiv* (2020).
- [9] Susan K Buchanan, Barbara S Smith, Lalitha Venkatramani, Di Xia, Lothar Esser, Maya Palnitkar, Ranjan Chakraborty, Dick Van Der Helm, and Johann Deisenhofer. 1999. Crystal structure of the outer membrane active transporter FepA from *Escherichia coli*. *Nature structural biology* 6, 1 (1999), 56–63.
- [10] Benjamin Buchfink, Chao Xie, and Daniel H Huson. 2015. Fast and sensitive protein alignment using DIAMOND. *Nature methods* 12, 1 (2015), 59–60.
- [11] Su Yun Chung and S Subbiah. 1996. A structural explanation for the twilight zone of protein sequence homology. *Structure* 4, 10 (1996), 1123 – 1127. [https://doi.org/10.1016/S0969-2126\(96\)00119-0](https://doi.org/10.1016/S0969-2126(96)00119-0)
- [12] UniProt Consortium. 2019. UniProt: a worldwide hub of protein knowledge. *Nucleic acids research* 47, D1 (2019), D506–D515.
- [13] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [14] Sean R. Eddy. 1998. Profile hidden Markov models. *Bioinformatics (Oxford, England)* 14, 9 (1998), 755–763.
- [15] Sean R Eddy. 2011. Accelerated profile HMM searches. *PLoS Comput Biol* 7, 10 (2011), e1002195.
- [16] Robert C Edgar. 2010. Search and clustering orders of magnitude faster than BLAST. *Bioinformatics* 26, 19 (2010), 2460–2461.

- [17] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al. 2019. The Pfam protein families database in 2019. *Nucleic acids research* 47, D1 (2019), D427–D432.
- [18] S El-Gebali, L Richardson, and R Finn. 2018. Repeats in Pfam. https://doi.org/10.6019/tol.pfam_repeats-t.2018.00001.1
- [19] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, et al. 2020. ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Deep Learning and High Performance Computing. *arXiv preprint arXiv:2007.06225* (2020).
- [20] Robert D Finn, Penelope Coghill, Ruth Y Eberhardt, Sean R Eddy, Jaina Mistry, Alex L Mitchell, Simon C Potter, Marco Punta, Matloob Qureshi, Amaia Sangrador-Vegas, et al. 2015. The Pfam protein families database: towards a more sustainable future. *Nucleic acids research* 44, D1 (2015), D279–D285.
- [21] Robert D Finn, Jaina Mistry, Benjamin Schuster-Böckler, Sam Griffiths-Jones, Volker Hollich, Timo Lassmann, Simon Moxon, Mhairi Marshall, Ajay Khanna, Richard Durbin, et al. 2006. Pfam: clans, web tools and services. *Nucleic acids research* 34, suppl_1 (2006), D247–D251.
- [22] Masaya Fujita, Kosuke Mori, Hirofumi Hara, Shojiro Hishiyama, Naofumi Kamimura, and Eiji Masai. 2019. A TonB-dependent receptor constitutes the outer membrane transport system for a lignin-derived aromatic compound. *Communications biology* 2, 1 (2019), 1–10.
- [23] Danny Hernandez, Jared Kaplan, Tom Henighan, and Sam McCandlish. 2021. Scaling Laws for Transfer. *arXiv:cs.LG/2102.01293*
- [24] Jie Hou, Badri Adhikari, and Jianlin Cheng. 2017. DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics* 34, 8 (2017), 1295–1303.
- [25] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).
- [26] Jaime Huerta-Cepas, Damian Szklarczyk, Davide Heller, Ana Hernández-Plaza, Sofia K Forslund, Helen Cook, Daniel R Mende, Ivica Letunic, Thomas Rattei, Lars J Jensen, et al. 2019. eggNOG 5.0: a hierarchical, functional and phylogenetically annotated orthology resource based on 5090 organisms and 2502 viruses. *Nucleic acids research* 47, D1 (2019), D309–D314.
- [27] Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, et al. 2014. InterProScan 5: genome-scale protein function classification. *Bioinformatics* 30, 9 (2014), 1236–1240.
- [28] Norman P Jouppi, Doe Hyun Yoon, George Kurian, Sheng Li, Nishant Patil, James Laudon, Cliff Young, and David Patterson. 2020. A domain-specific supercomputer for training deep neural networks. *Commun. ACM* 63, 7 (2020), 67–78.
- [29] W James Kent. 2002. BLAT—the BLAST-like alignment tool. *Genome research* 12, 4 (2002), 656–664.
- [30] Szymon M Kielbasa, Raymond Wan, Kengo Sato, Paul Horton, and Martin C Frith. 2011. Adaptive seeds tame genomic sequence comparison. *Genome research* 21, 3 (2011), 487–493.
- [31] Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative LSTM for sequence modelling. *arXiv preprint arXiv:1609.07959* (2016).
- [32] Takuto Kumano, Etsuko Fujiki, Yoshiteru Hashimoto, and Michihiko Kobayashi. 2016. Discovery of a sesamin-metabolizing microorganism and a new enzyme. *Proceedings of the National Academy of Sciences* 113, 32 (2016), 9087–9092.
- [33] Maria Littmann, Michael Heinzinger, Christian Dallago, Tobias Olenyi, and Burkhard Rost. 2021. Embeddings from deep learning transfer GO annotations beyond homology. *Scientific reports* 11, 1 (2021), 1–14.
- [34] Ali Madani, Bryan McCann, Nikhil Naik, Nitish Shirish Keskar, Namrata Anand, Raphael R Eguchi, Po-Ssu Huang, and Richard Socher. 2020. Progen: Language modeling for protein generation. *arXiv preprint arXiv:2004.03497* (2020).
- [35] EM Marcotte, M Pellegrini, TO Yeates, and D Eisenberg. 1999. A census of protein repeats. *Journal of molecular biology* 293, 1 (October 1999), 151–160. <https://doi.org/10.1006/jmbi.1999.3136>
- [36] Liam J McGuffin, Kevin Bryson, and David T Jones. 2000. The PSIPRED protein structure prediction server. *Bioinformatics* 16, 4 (2000), 404–405.
- [37] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. 2021. Pfam: The protein families database in 2021. *Nucleic Acids Research* 49, D1 (2021), D412–D419.
- [38] Leila K Mosavi, Tobin J Cammett, Daniel C Desrosiers, and Zheng-yu Peng. 2004. The ankyrin repeat as molecular architecture for protein recognition. *Protein science* 13, 6 (2004), 1435–1448.
- [39] Ananthan Nambiar, Maeve Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. 2020. Transforming the language of life: Transformer neural networks for protein prediction tasks. In *Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics*. 1–8.
- [40] Morgan N Price, Kelly M Wetmore, R Jordan Waters, Mark Callaghan, Jayashree Ray, Hualan Liu, Jennifer V Kuehl, Ryan A Melnyk, Jacob S Lamson, Yumi Suh, et al. 2018. Mutant phenotypes for thousands of bacterial genes of unknown function. *Nature* (2018), 1.
- [41] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. (2018).
- [42] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language Models are Unsupervised Multitask Learners. (2019).
- [43] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2019. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *CoRR* abs/1910.10683 (2019). [arXiv:1910.10683](https://arxiv.org/abs/1910.10683) <http://arxiv.org/abs/1910.10683>
- [44] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S Song. 2019. Evaluating protein transfer learning with tape. *Advances in Neural Information Processing Systems* 32 (2019), 9689.
- [45] Roshan Rao, Sergey Ovchinnikov, Joshua Meier, Alexander Rives, and Tom Sercu. 2020. Transformer protein language models are unsupervised structure learners. *bioRxiv* (2020).
- [46] Michael Remmert, Andreas Biegert, Andreas Hauser, and Johannes Söding. 2012. HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nature methods* 9, 2 (2012), 173–175.
- [47] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Demi Guo, Mylène Ott, C Lawrence Zitnick, Jerry Ma, and Rob Fergus. 2020. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. (Aug. 2020), 622803 pages.
- [48] Andrew W Senior, Richard Evans, John Jumper, James Kirkpatrick, Laurent Sifre, Tim Green, Chongli Qin, Augustin Zidek, Alexander WR Nelson, Alex Bridgland, et al. 2020. Improved protein structure prediction using potentials from deep learning. *Nature* 577, 7792 (2020), 706–710.
- [49] Seokjun Seo, Minsik Oh, Youngjune Park, and Sun Kim. 2018. DeepFam: deep learning based alignment-free method for protein family modeling and prediction. *Bioinformatics* 34, 13 (2018), i254–i262.
- [50] Amir Shalehshadeh, David Belanger, and David Dohan. 2020. Is Transfer Learning Necessary for Protein Landscape Prediction? *arXiv preprint arXiv:2011.03443* (2020).
- [51] Johannes Söding. 2005. Protein homology detection by HMM–HMM comparison. *Bioinformatics* 21, 7 (2005), 951–960.
- [52] Martin Steinegger, Milot Mirdita, and Johannes Söding. 2019. Protein-level assembly increases protein sequence recovery from metagenomic samples manifold. *Nature methods* 16, 7 (2019), 603–606.
- [53] Martin Steinegger and Johannes Söding. 2017. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* 35, 11 (2017), 1026–1028.
- [54] Martin Steinegger and Johannes Söding. 2018. Clustering huge protein sequence sets in linear time. *Nature communications* 9, 1 (2018), 1–8.
- [55] Nils Strothhoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. 2020. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics* 36, 8 (2020), 2401–2409.
- [56] Pascal Sturmfels, Jesse Vig, Ali Madani, and Nazneen Fatema Rajani. 2020. Profile Prediction: An Alignment-Based Pre-Training Task for Protein Sequence Models. *arXiv preprint arXiv:2012.00195* (2020).
- [57] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McFarvey, Cathy H Wu, and UniProt Consortium. 2015. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* 31, 6 (2015), 926–932.
- [58] Balázs Szalkai and Vince Grolmusz. 2018. Near perfect protein multi-label classification with deep neural networks. *Methods* 132 (2018), 50–56.
- [59] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762* (2017).
- [60] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. 2020. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222* (2020).
- [61] A Villegas-Morcillo, S Makrodimitris, RCHJ van Ham, AM Gomez, V Sanchez, and MJT Reinders. 2020. Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics (Oxford, England)* (2020).
- [62] Cen Wan and David T Jones. 2019. Improving protein function prediction with synthetic feature samples created by generative adversarial networks. *bioRxiv* (2019), 730143.
- [63] Birgit Wiltschi, Tomislav Cernava, Alexander Dennig, Meritxell Galindo Casas, Martina Geier, Steffen Gruber, Marianne Haberbauer, Petra Heinding, Enrique Herrero Acero, Regina Kratzer, et al. 2020. Enzymes revolutionize the bioproduction of value-added compounds: From enzyme discovery to special applications. *Biotechnology advances* 40 (2020), 107520.
- [64] Naihui Zhou, Yuxiang Jiang, Timothy R Bergquist, Alexandra J Lee, Balint Z Kacsó, Alex W Crocker, Kimberley A Lewis, George Georgiou, Huy N Nguyen, Md Nafiz Hamid, et al. 2019. The CAFA challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology* 20, 1 (2019), 1–23.

7 APPENDIX

Set name	Count	Sequence length			
		Min	Max	Mean	Median
test	3,019,006	11	29,202	311.52	217
train	27,052,653	11	36,991	311.54	217
valid	90,452	11	21,165	311.19	217

Table 5: Pre-training dataset statistics (UniRef50): the number of sequences (count) and sequence length statistics.

Sequence Identity Interval	Number of Sequences
10-12	62
12-14	426
14-16	1058
16-18	2516
18-20	4419
20-22	6013
22-24	4892
24-25	1902

Table 6: Number of sequences per sequence identity bucket for clustered Pfam-seed split.