ProtNLM: Model-based Natural Language Protein Annotation

Andreea Gane[1, 2], Maxwell L. Bileschi[1, 2, *], David Dohan[1], Elena Speretta[3], Amélie Héliou[1], Laetitia Meng-Papaxanthos[1],Hermann Zellner[3], Eugene Brevdo[1], Ankur Parikh[1], Maria J. Martin[3], Sandra Orchard[3], UniProt Collaborators[3], Lucy J. Colwell[1, 4, *]

Note to readers: this is a preprint - **it's a work in progress**! If there's something you want to know that's not adequately described, please reach out to: lcolwell@google.com and mlbileschi@google.com.

[1] Google Research, Cambridge, MA, USA.
[2] These authors contributed equally
[3] European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Hinxton, UK
[4] Department of Chemistry, University of Cambridge, Cambridge, UK

Abstract

A central challenge across the biological sciences is to predict the functional properties of a protein directly from its sequence, and thus discover new proteins with specific functionality. A particularly valuable advance would be a model that identifies sequences that satisfy functional requirements that are specified using natural language. This requires the model to build a mapping between amino acid sequence and natural language. Here, as a first step in this direction, we train models to predict free text natural language captions that describe the functional properties of amino acid sequences. We developed a set of metrics to measure model performance, and expert curators from the Pfam and UniProt databases manually evaluated model predictions in cases where the predictions did not match existing functional annotations. This feedback was used to improve performance, and the resulting model was used to predict protein names for ~49 million previously uncharacterized proteins that have now been released as part of the UniProt database.

Introduction

The ability to predict functional attributes of proteins directly from sequence is key to our ability to exploit the vast amounts of data unleashed by advances in high-throughput sequencing for a wide range of applications. The set of known, naturally occurring protein sequences is growing rapidly, although fewer than 1% have experimental annotations (Das and Orengo 2016). Great strides have been made towards assembling information into structured annotations and databases such as those provided by the Gene Ontology consortium and UniProt (UniProt Consortium 2021, Gene Ontology Consortium 2021). However, at least 30% of proteins cannot

be computationally annotated because their sequences are too distant from those with known function (Das and Orengo 2016, Price et al. 2018). Given the high costs and long timescales required for experimental protein annotation, new approaches to computational protein function prediction are urgently required.

Recent breakthroughs in natural language processing (Mikolov et al. 2013, Raffel et al. 2020, Brown et al. 2020, Devlin et al. 2018) suggest a potential path forward. Language models have been used to train models that accurately classify protein sequences (Nambiar et al. 2020, Dohan et al. 2021, Nallapareddy et al. 2022). Such *categorical* models are fundamentally limited, since they can predict only annotations seen during model training (Larochelle et al. 2008, Thomas et al. 2019). Discovering new protein functions requires the ability to *describe* the function of a protein beyond assigning it to a preexisting class. In principle, this could be realized by using the full expressivity of natural language, by analogy to tasks such as image captioning (Fig 1a), (Alayrac, et al. 2022, Stefanini, et al. 2022). Models that describe the function of a protein could intelligently interpolate among existing annotations and extrapolate outside this set, editing their component terms (Fig 1b) to generate descriptions that were not seen during training. The available structured ontologies provide numerous metadata fields describing proteins that are well understood, with free text annotations of varying length and complexity, often compiled by expert human annotators (Fig 1c, d). These resources provide a robust data source with which to train ML models that map protein sequence to natural language descriptions of protein function.

Here, we take a first step towards addressing this challenge, and report models that generate natural language annotations for ~49 million UniProt protein sequences that were previously called “Uncharacterized protein”. These model-predicted annotations are now available in UniProt. We show that models trained using data from the Pfam and UniProt databases (Mistry et al. 2021, UniProt Consortium 2021) perform robustly across protein annotation tasks with different database sizes that involve different types of metadata–from Pfam family accessions to UniProt protein names. Strikingly, we demonstrate that models can not only synthesize the natural language protein sequence annotations provided by structured ontologies, they can also annotate held out sequences when trained using free text annotations with remarkable fidelity. We work closely with expert professional curators from EMBL-EBI to evaluate accuracy, and demonstrate that in 88% of cases, curators either prefer the ProtNLM-predicted names, or have no strong preference versus the current name. When applied to unseen and unannotated protein sequences, our models can generate accurate novel descriptions. As an example, we identify >1500 previously uncharacterized proteins that are predicted to be homologous to CRISPR-Cas9, and we use a structure-prediction based pipeline to support many of these predictions. Moreover, we demonstrate that ProtNLM understands the query 'Small Cas9 homolog', which is not found in the training data, and returns appropriate results.

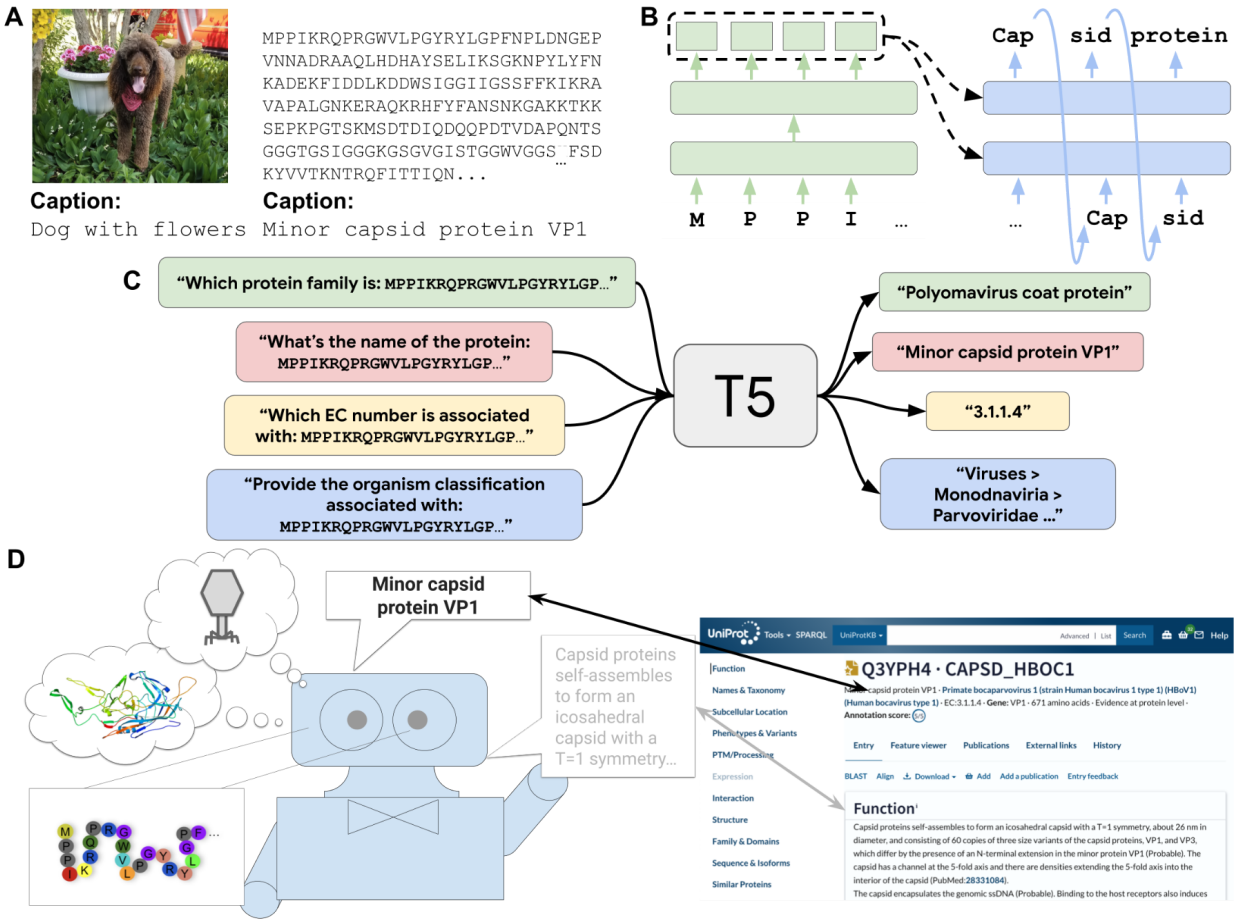


Fig 1 (a) Analogy between (left) the image captioning task and (right) the task of predicting protein function directly from the amino acid sequence. (b) We use an encoder-decoder framework where the encoder takes in the amino acid sequence and the decoder produces the output, one token (group of characters) at a time. (c) We use the T5 methodology (figure adapted from Raffel et al. 2020) to train a model on protein sequence annotation tasks. (d) The goal of the resulting model is to accurately predict descriptions of protein function directly from amino acid sequence.

Results

To demonstrate that language models can learn the relationship between amino acid sequences and English language descriptions of their functions, we use the Pfam database to define an initial task. Pfam v32.0 contains 17,841 unique single line descriptions, roughly one for each of the 17,929 families (El-Gebali et al. 2019). Given an unseen protein domain sequence, we use the flexible T5 framework to train a single model that predicts the corresponding single-line free text Pfam family description, the Pfam family id and the Pfam family accession (Raffel et al. 2020). The model is pre-trained across the UniRef50 2018_03 dataset (Suzek et al. 2015) of ~30 million diverse unlabelled protein sequences, then fine-tuned using labeled protein domain sequences from Pfam (see methods for details). To evaluate performance, we use a previously

published clustered split with 21293 held out test sequences, each of which is no more than 25% sequence-identical to the closest training sequence in the same Pfam seed family (Eddy 2011, Bileschi et al. 2019). Table 1 and Figs 2a, b show that ProtNLM is as accurate as the best existing categorical model, a pre-trained transformer (Dohan et al. 2021).

Model	Clan accuracy
Top pick HMM	88.08 ± 0.45
ProtNLM ★ (family id)	88.23 ± 0.45
ProtNLM ★ (description outputs)	88.28 ± 0.44
ProTNN (current SOTA)	90.50 ± 0.40
ProtNLM ★ (family accession)	90.73 ± 0.40

Table 1. ***Remote homology detection by ProtNLM matches current SOTA performance.*** When asked for the family accession of a given domain sequence that is no more than 25% sequence-identical to the training data, ProtNLM ★ (family accession) is highly accurate.

To measure performance at family description prediction, we adopt a strict exact-match criterion: we consider the ProtNLM-predicted description for each sequence to be correct only if it is *identical* to a Pfam description from the same clan. Descriptions deemed incorrect by this criterion may still be accurate, so this metric provides a useful lower bound on performance. We find that the natural language descriptions predicted by ProtNLM have an accuracy of 88.28%, comparable to the baseline Top pick HMM approach to family classification, which achieves 88.08%. Our exact-match evaluation does not address the possibility that ProtNLM may generate accurate *novel* descriptions. Indeed, >250 model-generated annotations were not seen in the training data. Examples include the ProtNLM description “phage endonuclease h2” for a domain from Q6NHN8, annotated in Pfam as Domain of Unknown Function (DUF1524). Indeed, Q6NHN8 has as a subsequence the protein A0A2T1BTI4, which is an HNH endonuclease, suggesting that Q6NHN8 is also an endonuclease. Upon training from scratch, we identified additional such cases. For example, the Pfam “glycosyl transferase family 11” domain from Q56870 is described by the model as “glycosyltransferase family 19 (fucosyltransferase)”. This family is similar to O-FucT, a family of O-fucosyltransferases. Based on our findings, Pfam have renamed “glycosyl transferase family 11” and merged it into a clan of fucosyltransferases.

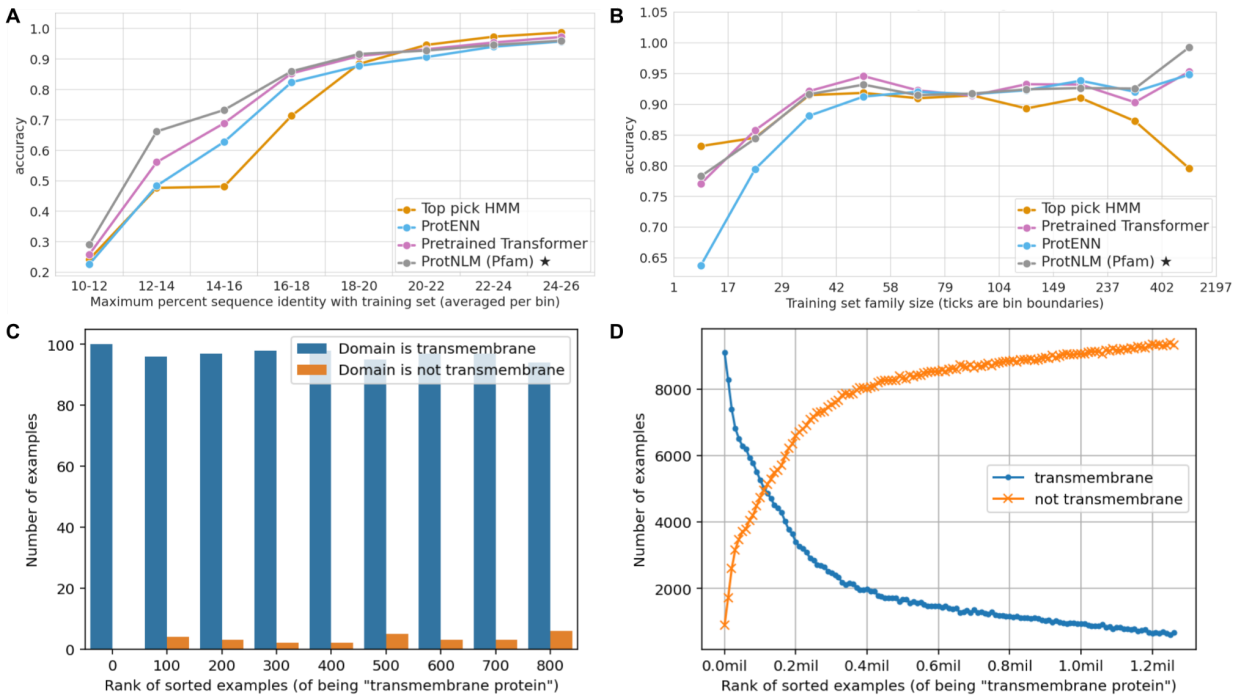


Fig 2. (a) ProtNLM ★ performance at very remote homology detection. ProtNLM statistically outperforms Top pick HMM at sequence identities between 12% and 20%, and also outperforms a pretrained transformer (Dohan et al. 2021) for 12-16%. Top pick HMM outperforms ProtNLM at sequence identities between 20% and 25%. Other differences are not statistically significant ($p < 0.05$, 2-sided McNemar test). (b) Performance of ProtNLM and other models at remote homology detection stratified by training set family size. Top pick HMM outperforms other models for families with <17 training sequences, while differences for 17-104 training sequences are not significant. ProtNLM outperforms Top pick HMM for families with >104 training sequences ($p < 0.05$, 2-sided McNemar test). ProtNLM accurately identifies transmembrane protein domains for which the Pfam description does not contain 'transmembrane' or 'tm', as judged by Phobius, over (c) 10,000 domains with highest likelihood and (d) >1.2 million Pfam seed protein domains.

We next examine whether the model understands whether a specific phrase applies to a sequence when this phrase is not contained in its Pfam description. We used the trained model to rank ~1.2 million Pfam-seed domain sequences by the likelihood that they match the description “transmembrane domain” removing all examples that mention “transmembrane” or “tm” in their training descriptions. To assess accuracy, we use Phobius transmembrane predictions (Käll et al, 2005). Strikingly, Phobius agrees with all of our top 100 predictions, while high model accuracy is maintained over our top 10,000 predictions (Fig. 2c). Moreover, the monotonicity of Fig. 2d suggests that the model accurately ranks sequences across the entire set.

Overall, these results suggest that despite being trained using just 17,841 one-line protein family descriptions, ProtNLM performs strongly. Encouraged, we turn to the UniProt database,

which provides multiple free text and categorical metadata fields for ~230 million protein sequences. We target the UniProt 'Protein Name' field (UP), which provides a succinct description of the known or predicted function of a whole protein sequence. The key use cases for UniProt are to annotate the millions of sequences added between one release and the next, and those sequences lacking annotation. To evaluate ProtNLM performance, we built a temporal split: models were trained on sequences/labels from UniProtKB 2021_02, sequences introduced in UniProtKB 2021_03 are used as for validation, while sequences introduced after UniProtKB 2021_03 are used as test data (Zhou, et al., 2019). We use UniRef50 2022_01 to build a clustered subset of the test set to evaluate performance across sequences that are at least 50% distinct from the closest training sequence (Strodthoff et al. 2020).

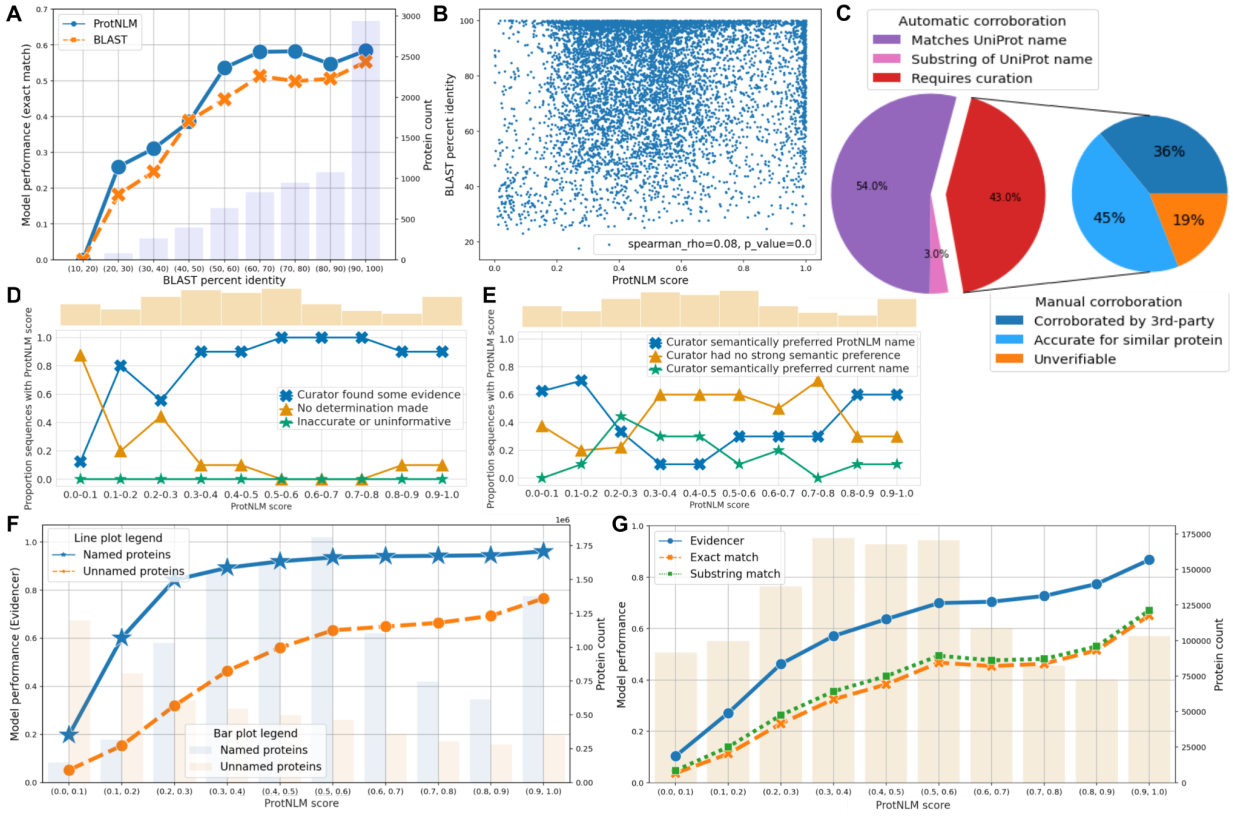


Fig 3. (a) Exact match accuracy of the ProtNLM-predicted protein name for held out proteins as a function of the % identity to the top BLAST hit from the training set (orange line). For comparison, we also plot the accuracy with which the protein name of the top BLAST hit from the training data matches the UniProt assigned protein name (blue line). (b) We observe low correlation between the ProtNLM score and the % identity to the top BLAST hit from the training set, suggesting that ProtNLM learns complementary information. (c) Across the dev split, 57% of ProtNLM-predicted names match a UniProt protein name exactly, or are an exact substring. A subset of the remaining 43% of predictions were sent for expert manual curation, of which 79/97 were deemed accurate, while 18/97 could not be verified or invalidated by the curators. (d) Manual curation results (accuracy) stratified by ProtNLM score. (e) Manual curation results (preference) stratified by ProtNLM score. (f) Evidencer corroboration of ProtNLM-predicted names for held-out sequences as a function of ProtNLM score. (g) ProtNLM performance for challenging proteins for which no sequence from the same UniRef50 cluster is present in the

training set. We report exact match (orange), substring match (green) and 'Evidencer' tool results.

We provide the amino acid sequence as input, and short, free-text UniProt protein name(s) as output and evaluate whether the ProtNLM-predicted name exactly matches the UniProt protein name. We compare performance with BLAST, a natural baseline that transfers the name of the most similar training sequence and hence is limited to protein names seen during training (Altschul et al 1997). Our goal is for ProtNLM to retain the performance of BLAST, while adding the ability to generate novel protein names. Figure 3a shows comparable exact match accuracy for ProtNLM and BLAST, stratified by the distance (BLAST % identity) of each held-out test sequence from the training data. Perhaps surprisingly, Figure 3b shows that the ProtNLM score (normalized to [0, 1], higher is better) for each sequence correlates weakly with distance (BLAST % identity) from the training data. This suggests that ProtNLM learns complementary information, a phenomena previously reported for deep learning approaches to protein sequence annotation (Bileschi et al. 2019).

How can we evaluate model performance? Across the temporal dev split, 57% of ProtNLM-predicted names match a UniProt protein name exactly, or are an exact substring (Fig. 3c). Evaluating accuracy for the remaining 43% can be challenging, so we asked the expert UniProt curators at EMBL-EBI to *manually* evaluate a subset of 100 predictions chosen at random with stratified ProtNLM scores. In three cases ProtNLM predicted EC numbers, which are not recommended as protein names by UniProt. The curators reached consensus on each remaining example and 79 predicted names were verified as correct, while the remaining 18 predicted names could neither be verified or invalidated. Evidence to support the ProtNLM-predicted name could more often be found when the model was more confident (Fig. 3d). The curators reported no semantic difference between model-generated and UniProt names for 47 of the remaining 97 examples, while they semantically preferred ProtNLM-predicted names in 28 cases and UniProt names in 22 cases. These preferences were independent of ProtNLM scores (Fig. 3e). Curators preferred predicted names with ProtNLM scores as low as 0.05 or with BLAST % identity as low as 33% to the closest training sequence over the existing UniProt names. Overall, these results suggest that curators found the ProtNLM-predicted names to be valuable.

To further evaluate model performance, inspired by the curators, we developed an automated verification tool (the 'Evidencer') that checks whether a predicted name occurs in the full uniprot entry for any protein in the same UniRef50 2022_01 cluster as the query sequence. The Evidencer faithfully reproduces curator decisions for 81 out of 97 examples. In 10 cases it could not find the curator's corroboration, and in six cases it found evidence that the curator did not find/trust. Across the entire test set, the Evidencer corroborates ProtNLM-predicted protein names for ~87% of characterized proteins (Fig. 3f). To probe performance on sequences that are further from the training set, we turn to the clustered test split, where performance as measured by the exact match and substring metrics and by the Evidencer increases smoothly with ProtNLM score (Fig. 3g).

Can ProtNLM accurately predict names for uncharacterized proteins? Of the 97 examples examined by the UniProt curators, 25 sequences corresponded to uncharacterized proteins. For 17 of these examples, the experts found evidence to support the ProtNLM-predicted names, while the remaining 8 could not be verified or invalidated. These results motivated us to consider the challenge of predicting names for the ~30% (~56 million) of UniProt proteins that are uncharacterized. To assess model performance for this set, we predicted names for 214 randomly chosen uncharacterized proteins, and sent them to UniProt for curation. Results, stratified by (left) ProtNLM score and (right) % identity to the top BLAST hit from the training set, are shown in Fig. 4a. Overall, only six (< 3%) ProtNLM-predicted names were judged to contain errors or be uninformative. Half of these were uninformative names such as 'expressed protein' that we have now filtered out from model predictions. The curators found evidence supporting the ProNLM names for 82 examples, while 126/214 could not be verified or invalidated.

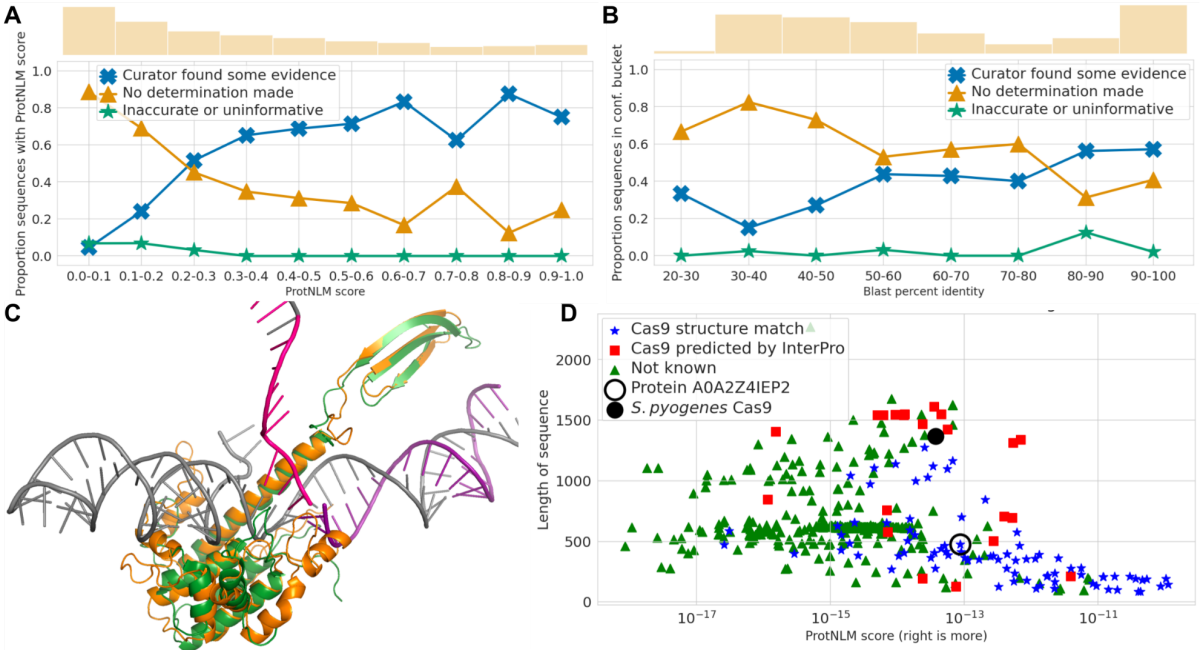


Fig 4. Manual curation results for a subset of uncharacterized proteins from the temporal test set stratified by (a) ProtNLM score and (b) BLAST % identity to the top hit from the training set. Curators more often found evidence for predicted names with higher ProtNLM score. (c) Structural alignment of the Rec1 domain from Nme1Cas9 (green, from pdb structure 6KC8 (Sun et al. 2019), sgRNA in grey, DNA in pink/purple) with the AlphaFold2 predicted structure of the ProtNLM-predicted Cas9 Rec1 domain from the uncharacterized protein A0A2Z4IEP2. (d) We queried ProtNLM with the phrase 'Small Cas9 homolog' and identified the 500 currently uncharacterized proteins with the highest model likelihood. Of these, 52 were predicted to contain Cas9 domains by InterPro (red squares), while those sequences whose AlphaFold2 predicted structure returns a FoldSeek match with e-value <0.01 to a Cas9 PDB structure are shown as blue stars.

Motivated by these results we used ProtNLM to predict protein names for every one of the ~49 million uncharacterized proteins in the UniProt database, using the Evidencer to evaluate every prediction (see supplementary data). Fig. 3e demonstrates the Evidencer could better

corroborate ProtNLM predictions with higher ProtNLM score, mirroring the behavior of expert curators in Fig. 4a, b. Among the set of uncharacterized proteins we observe numerous examples with low ProtNLM score where predictions can be validated either by manual curators or using the Evidencer, indicating that even low-score ProtNLM-predicted names can be highly accurate.

To further probe the extent to which we can trust ProtNLM-predicted protein names, we used recent breakthroughs in protein structure prediction to probe the 1579 currently uncharacterized proteins whose ProtNLM-predicted names contain 'Cas9'. For each sequence, we used Foldseek (van Kempen et al. 2022) to search the AlphaFold2 predicted structure (Jumper et al. 2021, Varadi et al 2022) against the PDB. Sequences with ProtNLM score as low as 0.07 had Cas9 as their top Foldseek hit (e-value 10^{-39}). Out of the 100 most confident predictions, we found strong evidence that 68 are homologs of Cas9, while 16 are homologs of variants such as Cas12 or Cas13. For the remaining cases 13 had no available predicted structure, or no significant hit to the PDB, while 3 had significant similarity to non-cas proteins, suggesting that the ProtNLM-predicted names are not accurate. Figure 4c shows a structural alignment between the AlphaFold2-predicted structure of the uncharacterized protein A0A2Z4IEP2 and its top Foldseek hit, the Rec1 domain of Nme1Cas9 (PDB 6KC8, Sun et al. 2019). While InterPro finds no significant homology between A0A2Z4IEP2 and Cas9, ProtNLM predicts the name 'HNH Cas9-type domain-containing protein' with a score of 0.64.

Finally, we asked whether a user can successfully ask ProtNLM (using language) for a protein with specific functional properties. To increase the challenge, we queried ProtNLM for a a 'Small Cas9 homolog', a phrase that is not found within UniProt. For ProtNLM to succeed, it has to have a general sense of what “small” means–independent of context–and to apply that knowledge to putative Cas9 homologs. The 500 proteins with highest model likelihood contain Cas9 in their ProtNLM-predicted name. Fig. 4d shows sequence length (size) as a function of the ProtNLM score for the phrase 'Small Cas9 homolog'. We observe significant spearman (-0.79, p-value 10^{-19}) and pearson (-0.35, p-value 0.001) correlations between model likelihood and protein size, suggesting the model understands the concept 'small'. We used both interpro (red squares) and AlphaFold2+Foldseek (blue stars) to corroborate a number of these predictions. The *S. pyogene*s Cas9 homolog and the uncharacterized protein A0A2Z4IEP2 shown in Figure 4c are also indicated in Figure 4d. Taken together, these results suggest that ProtNLM
predicts protein names for currently uncharacterized proteins with high accuracy.

After rigorous expert evaluation of the ProtNLM predictions, the team at UniProt decided to adopt ProtNLM-predicted names for 49 million of 56 million uncharacterized proteins. These predictions are available and visible by default on each protein’s UniProt page, e.g. at https://www.uniprot.org/uniprotkb/A0A2Z4IEP2/entry. A file containing predictions for 218 million proteins from TrEMBL, along with information from the Evidencer describing whether and how ProtNLM’s prediction is supported via UniRef is available in Google Cloud Storage. An interactive colab notebook is provided on github for easy access. Finally, a colab allowing users

to put in their own protein sequences, and get ProtNLM's prediction of the protein's name, is also available on github.

References

1. Das, Sayoni, and Christine A. Orengo. "Protein function annotation using protein domain family resources." Methods 93 (2016): 24-34.
2. UniProt: the universal protein knowledgebase in 2021. Nucleic acids research 49, no. D1 (2021): D480-D489.
3. Gene Ontology Consortium. "The Gene Ontology resource: enriching a GOld mine." Nucleic acids research 49.D1 (2021): D325-D334.
4. Price, Morgan N., et al. "Mutant phenotypes for thousands of bacterial genes of unknown function." Nature 557.7706 (2018): 503-509.
5. Mikolov, Tomas, et al. "Efficient estimation of word representations in vector space." arXiv preprint arXiv:1301.3781 (2013).
6. Devlin, Jacob, et al. "Bert: Pre-training of deep bidirectional transformers for language understanding." arXiv preprint arXiv:1810.04805 (2018).
7. Brown, Tom, et al. "Language models are few-shot learners." Advances in neural information processing systems 33 (2020): 1877-1901.
8. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." J. Mach. Learn. Res. 21.140 (2020): 1-67.
9. Nambiar, Ananthan, et al. "Transforming the language of life: transformer neural networks for protein prediction tasks." Proceedings of the 11th ACM International Conference on Bioinformatics, Computational Biology and Health Informatics. 2020.
10. Dohan, David, et al. "Improving protein function annotation via unsupervised pre-training: Robustness, efficiency, and insights." Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021.
11. Nallapareddy, Vamsi, et al. "CATHe: Detection of remote homologues for CATH superfamilies using embeddings from protein language models." bioRxiv (2022).
12. Larochelle, Hugo, Dumitru Erhan, and Yoshua Bengio. "Zero-data learning of new tasks." AAAI. Vol. 1. No. 2. 2008.
13. Thomas, Paul D., et al. "Gene Ontology Causal Activity Modeling (GO-CAM) moves beyond GO annotations to structured descriptions of biological functions and systems." Nature genetics 51.10 (2019): 1429-1433.
14. Alayrac, Jean-Baptiste, et al. "Flamingo: a visual language model for few-shot learning." arXiv preprint arXiv:2204.14198 (2022).
15. Stefanini, Matteo, et al. "From show to tell: a survey on deep learning-based image captioning." IEEE Transactions on Pattern Analysis and Machine Intelligence (2022).
16. Mistry, Jaina, et al. "Pfam: The protein families database in 2021." Nucleic acids research 49.D1 (2021): D412-D419.
17. El-Gebali, Sara, et al. "The Pfam protein families database in 2019." Nucleic acids research 47.D1 (2019): D427-D432.

18. Suzek, Baris E., et al. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches." Bioinformatics 31.6 (2015): 926-932.
19. Eddy, Sean R. "Accelerated profile HMM searches." PLoS computational biology 7.10 (2011): e1002195.
20. Bileschi, Maxwell L., et al. "Using deep learning to annotate the protein universe." Nature Biotechnology (2022): 1-6.
21. Berman, Helen M., et al. "The protein data bank." Nucleic acids research 28.1 (2000): 235-242.
22. Käll, Lukas, Anders Krogh, and Erik LL Sonnhammer. "An HMM posterior decoder for sequence feature prediction that includes homology information." Bioinformatics 21.suppl_1 (2005): i251-i257.
23. UP https://www.uniprot.org/help/protein_names.
24. Zhou, Naihui, et al. "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.
25. Strodthoff, N., Wagner, P., Wenzel, M. and Samek, W., 2020. UDSMProt: universal deep sequence models for protein classification. Bioinformatics, 36(8), pp.2401-2409.
26. Altschul, Stephen F., et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic acids research 25.17 (1997): 3389-3402.
27. Sun, Wei, et al. "Structures of Neisseria meningitidis Cas9 complexes in catalytically poised and anti-CRISPR-inhibited states." Molecular cell 76.6 (2019): 938-952.
28. van Kempen, Michel, et al. "Foldseek: fast and accurate protein structure search." bioRxiv (2022).
29. Jumper, John, et al. "Highly accurate protein structure prediction with AlphaFold." Nature 596.7873 (2021): 583-589.
30. Varadi, Mihaly, et al. "AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models." Nucleic acids research 50.D1 (2022): D439-D444.

Methods

Datasets

We use two datasets, Pfam and Uniprot, to build tasks that we use to evaluate model performance.

Pfam is a public database of protein families and domains [1]). Each family is associated with several pieces of metadata, including: a unique family accession (PF followed by 5 digit ID, such as "PF11916"), a short natural language description of the family (such as "Vacuolar protein 14 C-terminal Fig4p binding"), and a unique alphanumeric id which is generally derived from the description (such as “Vac14_Fig4_bd”). The Pfam family descriptions are short, free text

descriptions such as "Helix-loop-helix DNA-binding domain" or "ATPase family associated with various cellular activities (AAA)" that often provide information about protein function. Pfam domain descriptions are typically written by curators and are recorded in the "DE" line of the Pfam-A Stockholm files. We use the 17,929 different Pfam family labels from Pfam v32.0, each with an associated id and description. We train and evaluate models on a clustered split of the Pfam seed dataset introduced in previous work [2]. In this setting, the validation and test sets are constructed, via clustering, such that no sequence has more than 25% sequence identity with any element of the training set in the Pfam-seed alignment. Pfam-seed alignment creation is overseen by professional curators.

UniProt is a public database of protein sequences and known information about them [3]. The UniProt web page corresponding to any protein identifier (accession), found at https://www.uniprot.org/uniprotkb/`accession`/entry (e.g. https://www.uniprot.org/uniprotkb/P03135/entry), contains the protein name, the associated gene label, the associated functional domains and functional components, the known or predicted protein structure, and many other pieces of information.

UniProt names are short, free-text descriptions of proteins. A UniProt name typically consists of a summary of the protein's important properties. For instance, the name "Capsid Protein VP1", associated with the protein P03135, tells us that we are dealing with a viral protein because a capsid is the protein shell of a virus, and capsid proteins assemble to form the shell. Additionally, the name tells us that this is a type-1 viral protein (VP). Among other use cases, UniProt can help a user quickly scan the outputs of a protein similarity search to identify common functions. Therefore, a good name must provide comprehensive, specific information about the protein, while being accurate, concise, and consistent. The International Protein Nomenclature Guidelines (https://www.ncbi.nlm.nih.gov/genome/doc/internatprot_nomenguide/) provide additional information and guidelines regarding protein names.

Existing UniProt names come from a range of different sources including 1) Manual observations made–or carefully reviewed by–human curators, and 2) Automatic annotations extracted from other databases or obtained using learned or rule-based systems that have not yet been manually reviewed. Specific annotation source labels and their descriptions from UniProt are shown in the table below, containing information reproduced from https://www.uniprot.org/help/evidences.

Whether the assertion was made by humans / computers		Description (details from UniProt)
Manual assertions	Automatic assertions	
ECO:0000269	N/A	**Experimental evidence**: manually curated information for which there is published experimental evidence.

ECO:0000303	N/A	**Non-traceable author statement evidence**: manually curated information that is based on statements in scientific articles for which there is no experimental support.
ECO:0000305	N/A	**Curator inference evidence**: manually curated information which has been inferred by a curator based on their scientific knowledge or on the scientific content of an article.
ECO:0000250	N/A	**Sequence similarity evidence**: manually curated information which has been propagated from a related experimentally characterized protein.
ECO:0000255	ECO:0000256[1] ECO:0000259[2]	**Sequence model evidence**: information which has been generated by the UniProtKB automatic annotation system (e.g. based on UniRule and ARBA); ECO:0000255 is also used for information which has been generated by various sequence analysis programs that are used during the manual curation process and which has been verified by a curator.
ECO:0000312	ECO:0000313	**Imported information evidence**: information which has been imported from another database.
ECO:0007744	ECO:0007829	**Combinatorial evidence**: information inferred from a combination of experimental and computational evidence.

The UniProt database already employs various automatic annotation systems (e.g. UniRule, SAAS/ARBA 17, UniProt Consortium 2021). The existing annotation systems use (manually and automatically) rules to give new proteins predefined labels or descriptions and rely on annotations such as protein families, domains, or important sites, obtained with simple sequence models such as HMMs. The rules are typically selected to have good specificity at the cost of low coverage, so more than 30% of proteins are still uncharacterized, even using automated systems. Our goal is to increase the number of annotated sequences, and we also compare our predictions with the existing predictions for proteins that are already annotated as a measure of the specificity of our predictions.

UniProt Data splits

While a standard approach in machine learning is to build a *random split* where random subsets of the data are held out for evaluation at validation and test time, protein datasets require alternative approaches. In particular, due to evolution, there are dense clusters of proteins so a random split can yield a high proportion of test sequences that are very close to the training set.

1

2

Temporal split. The key use case for UniProt is to train a model that can annotate sequences added between one release and the next, especially those sequences lacking other annotation. This use case is best reflected via a “temporal” (time-based) split, where we use one data release for training, and a future data release for evaluation [18]. This task is important because UniProt adds millions of sequences per year, requiring significant curatorial resources.

To this end, we train models on the 2021_02 UniProt release, validating on protein sequences introduced into UniProt between 2021_02 and 2021_03 (dev set), and testing on sequences introduced after 2021_03 that are included in the 2022_01 UniProt. For the dev set, we exclude any protein whose UniProt protein names include "Uncharacterized protein" (ignoring case). For the dev and test sets, we filter the relevant UniProt release to exclude any proteins for which either the accession or the amino acid sequence appears in the previous UniProt release.

UniProt Name processing: We use the UniProt 2021_02 release to build the training data. The release contains 214,971,037 proteins. Each protein is associated with a unique accession, but note that there are many distinct proteins that have the exact same protein sequence. We parse the UniProt flat file data (.dat files) to get the following name types (as each protein can have multiple names):

- **Overall**: names that apply to the entire protein. E.g. A0A099UB22: “Arginine biosynthesis bifunctional protein ArgJ”
- **Includes**: names that apply to a functional domain; we can use these names as overall protein names. E.g. A0A099UB22: “Glutamate N-acetyltransferase” or “Amino-acid acetyltransferase”.
- **Contains**: names that apply to a functional component; we cannot use these names as overall protein names. E.g. A0A099UB22: “Arginine biosynthesis bifunctional protein ArgJ alpha chain” or “Arginine biosynthesis bifunctional protein ArgJ beta chain”.

The majority of names are Overall names. Here, for reference, we reproduce the relevant lines from an example record (.dat file):

```
ID   A0A099UB22_9HELI        Unreviewed;       404 AA.
AC   A0A099UB22;
DT   07-JAN-2015, integrated into UniProtKB/TrEMBL.
DT   07-JAN-2015, sequence version 1.
DT   03-AUG-2022, entry version 45.
DE   RecName: Full=Arginine biosynthesis bifunctional protein ArgJ
{ECO:0000256|HAMAP-Rule:MF_01106};
DE   Includes:
DE     RecName: Full=Glutamate N-acetyltransferase
{ECO:0000256|HAMAP-Rule:MF_01106};
DE              EC=2.3.1.35 {ECO:0000256|HAMAP-Rule:MF_01106};
DE     AltName: Full=Ornithine acetyltransferase
{ECO:0000256|HAMAP-Rule:MF_01106};
DE              Short=OATase {ECO:0000256|HAMAP-Rule:MF_01106};
```

```
DE     AltName: Full=Ornithine transacetylase
{ECO:0000256|HAMAP-Rule:MF_01106};
DE   Includes:
DE     RecName: Full=Amino-acid acetyltransferase
{ECO:0000256|HAMAP-Rule:MF_01106};
DE              EC=2.3.1.1 {ECO:0000256|HAMAP-Rule:MF_01106};
DE     AltName: Full=N-acetylglutamate synthase
{ECO:0000256|HAMAP-Rule:MF_01106};
DE              Short=AGSase {ECO:0000256|HAMAP-Rule:MF_01106};
DE   Contains:
DE     RecName: Full=Arginine biosynthesis bifunctional protein ArgJ
alpha chain {ECO:0000256|HAMAP-Rule:MF_01106};
DE   Contains:
DE     RecName: Full=Arginine biosynthesis bifunctional protein ArgJ
beta chain {ECO:0000256|HAMAP-Rule:MF_01106};
```

Our ground truth data includes both expert-curated names and names imported from many other databases or uploaded by users, and as a result the ground truth names can be noisy. For example, some protein names are uninformative, contain extraneous information, contain information our models wouldn't be able to learn or otherwise don't conform to the International Protein Nomenclature Guidelines. To ameliorate this issue, we perform an initial data cleaning step to remove or process the input names

- We start by removing proteins which don't have any name that is different from "uncharacterized protein" (insensitive to case) - we do not want to train our models to predict this name as "uncharacterized protein" is a placeholder text to suggest that no name has been assigned to the protein. The resulting data size is 156,495,706 accessions and 235,025,652 protein-name pairs.

- We next extract the set of unique names from the data, keeping only those names that appear in the "Overall" or "Includes" name fields of UniProt entries. The resulting set of 4,165,734 unique names corresponds to 233,768,763 protein-name pairs and 156,495,706 unique protein accessions. Note that keeping only "Overall" and "Includes" names did not remove any proteins from our training set as all proteins that have a "Contains" name also have at least one "Overall" name.

- We remove uninformative names such as "N/A", "NA" and other names which are treated as missing values by the pandas package [5]. We note however that, while the name "N/A" is typically a submitted name indicating an empty name, the term "NA" can also appear as a shortcut from "Neuraminidase". Nevertheless, when the term "NA" appears as a short name, it typically also appears in its long form (the protein would have at least two names), so we can remove the short name while still maintaining the protein in the training data.

- We compute a standardized format of each remaining unique name as follows:
    - **Remove terms that indicate some level of uncertainty**, either in the quality of the sequence or the quality of the annotation, such as “putative”, “probable”, “low quality protein”. It is unclear what criteria different sources use to determine both (i) the level of certainty, and (ii) whether to indicate some level of uncertainty in the name, so to be consistent we remove these terms.
    - **Strip isoform ending.** We don’t expect a model to learn this information. Example: “Dystrobrevin binding protein 1 isoform 1”.
    - **Strip the last token unless it’s a common name**. Oftentimes, gene loci are included as the final term in the protein, and we don’t expect a model to learn this numbering system. Example: “B3 domain-containing protein LOC_Os12g40090”
    - **Uppercase first letter.** Many sources that are imported into UniProt don’t have a standard on capitalization, so, we standardize the capitalization.
- We remove the resulting name if:
    - **If it contains the term isoform**, after stripping the isoform ending. Some proteins have isoform information, but this information is not at the end. Further, there are multiple ways this information appears, and as such, instead of attempting to clean this data, we discard it. Example: “CD34 antigen (Predicted), CD34 antigen isoform 1, CD34 molecule”.
    - **Has disallowed terms**, such as 'uncharacterized', 'genomic scaffold', 'genomic, scaffold', 'whole genome shotgun sequence', '|'. Example: “Unplaced genomic scaffold scaffold_4”
    - **Is an uninformative name**, such as those consisting only of the terms 'uncharacterized', 'genomic scaffold', 'genomic, scaffold', 'whole genome shotgun sequence', '|'.
    - **Is a single token** that is unique in the data. Oftentimes, gene loci are included as the final term in the protein, and we don’t expect a model to learn this numbering system. Example: “LOC_Os12g40090”

This results in a set of unique processed names. In practice, many names are not changed at all by this processing. Finally, we replace each name in the training data by its corresponding unique processed name. Specifically, for each protein in the UniProt 2021_02 release, we first create a sequence-name pair for every protein name found in the overall, includes or contains fields of the UniProt entry. For each sequence-name pair, we then check whether the name is present in the set of unique names, and if necessary replace it with the corresponding unique processed name. If the name is not present in the set of unique names, then this sequence name pair is discarded.

The size of the resulting data set is: 153,502,756 accessions and 231,697,973 sequence-name pairs. Note that despite these processing steps, the training data does still contain some undesirable names, such as:

- **Names that are too vague**, such as 'alternative protein', 'expressed protein', 'prediction', 'cdna', 'unspecified product', 'similar to', 'similarity', 'bgt-', 'Zmp:0000', 'conserved protein domain' (ignoring case).
- **Names that contain terms that are typically used for uncharacterized proteins**, such as 'hypothetical', 'uncharacterized', 'unassigned function', 'unnamed product', 'str fm', 'str. fm', 'wgs project ccbq000000000', 'genome assembly', '---na', 'predicted: ', ‘na’, ‘null’ (ignoring case).

Future work could improve the processing pipeline to better handle these examples.

The resulting data is split on accession into 215 random shards, which can have different sizes (because some accessions have more sequence-name pairs than others). As a result, each accession appears in a single shard, possibly in multiple rows, one row per protein_accession-sequence-name pair.

Test set: We start with the parsed UniProtKB 2022_01 data. We have 230,895,644 proteins (see the release notes for release 2022_01). Of these, there are 28,452,009 proteins whose sequence does not appear in the 2021_02 release (used to construct the training data), and 107,928 entries whose primary accession appears in 2021_02 (despite that their sequence does not - these are entries which changed their sequence slightly between releases 2021_02 and 2022_01).

In addition, we check each primary accession from the test set against all accessions (primary or secondary) from the previous release (2021_03), resulting in 28,344,081 proteins. Finally, there are 10,033,492 proteins whose sequence appears in the 2021_03 release (used to construct the validation set). Excluding these as well, we are left with 18,310,589 proteins. Of these, there are 62 proteins whose primary accession appears in 2021_03 (despite that their sequence does not) - these are entries which changed their sequence slightly between releases. Excluding these as well, we obtain 18,310,527 proteins.

We note that a number of newly-introduced proteins are deleted from the UniProt database between releases either as requested by the original submitters or because there is enough evidence that the corresponding amino acid sequence does not actually code for proteins (see e.g. the description on the UniProt page). We exclude these proteins from our reported results on the test set. Specifically, we exclude from the test set the proteins whose accessions have become obsolete between release 2022_01 (used to construct the test set) and 2022_04 (the current release, for which we provide predictions). There are 1,964,227 proteins removed at this stage.

Challenging subset of test set. Nevertheless, to address the potential concern that the sequences being evaluated are too close to the training set, such that one cannot assess the model’s ability to generalize to further sequences, we also evaluate our models on a challenging subset of evaluation sequences designed to be far from the closest sequence in the training set. In particular, we use UniRef50 2022_01 clustering and identify the subset of evaluation

sequences whose UniRef cluster is not seen in the training set, i.e. that have at most 50% similarity to another protein sequence in the training set [4]. In particular, we extract the test set accessions whose UniRef50 cluster in the 2022_01 release does not contain any members whose accession (primary or secondary) appears among the training set primary accessions.

The resulting protein counts by dataset are outlined here:

Fold	**Count proteins**
Train set	153,502,756
Test set	18,310,527
Challenging subset of the test set	5,561,028

Standard Challenges in Captioning

The protein name prediction problem closely resembles the image captioning problem in computer vision, where in this analogy, the image corresponds to the amino acid sequence and the caption corresponds to the protein name.

One protein, multiple captions. One important challenge in image captioning is that each image can have multiple, very different captions, either because of synonymy or because the different captions focus on different information in the image. The same is true for proteins. For instance, the protein Q58842 has multiple names in our dataset, including "`Bifunctional enzyme Fae/Hps`", "`Formaldehyde-activating enzyme`", and "`3-hexulose-6-phosphate synthase`". The second name highlights that the protein contains a "Formaldehyde-activating enzyme" region, which is found at residues 1-150, while the third name highlights that the protein contains a "3-hexulose-6-phosphate synthase1" region, which is found at residues 151-381. This is equivalent to describing a dog playing with a frisbee in one section of the image and a child building a sandcastle in another. The first, recommended name highlights this bifunctional nature of the protein, where `Fae` and `Hps` are abbreviations. In this case the first name actually encodes all (or most) information provided in the other names, but this does not have to be the case: an example of a less informative name might be "Multifunctional fusion protein" with no additional detail, equivalent to the caption "Image with multiple activities going on" that makes no mention of the dog or the sandcastle.

Therefore, because of this multiplicity of potential correct solutions, it can be challenging to evaluate descriptions that differ from the ground truth ones [19].

One caption, multiple proteins. Of course the same caption can also be assigned to multiple images. The same is true for proteins. For instance, two different proteins can be named "Capsid protein", even though they correspond to different types of viruses, they have different amino acid sequences, the virus shells they belong to are slightly different in shape, etc.

Signature Challenges in Protein Captioning

Protein captions are particularly challenging to assess. An important goal is to build one model that can label uncharacterized proteins with existing names when possible, and invent new names when the existing names cannot accommodate the new protein. Among other use cases, a generation approach could enable more comprehensive naming for proteins that contain previously unseen combinations of functional regions. However, unlike related tasks such as image captioning and title prediction, for protein captioning, manual annotators need to have significant biology expertise and a good knowledge of available resources for corroborating the protein information. Furthermore, captions may include protein information that cannot be corroborated to be either True or False without carrying out experiments to test the resulting hypotheses.

Baselines

Pfam Domain Description Prediction

We quantitatively evaluate our models' classification accuracy against an HMM baseline and CNN models introduced in [2]. We additionally compare to the protein BERT model presented in [6]. These baselines provide robust competition for family classification performance, but are unable to generate descriptions outside of the categorical class prediction.

Top pick HMM The first baseline is the standard approach for predicting protein families. The Pfam database is constructed by fitting a profile HMM using HMMER to curate seed sequences for each protein family [1, 7]. For comparison to our models, we construct a pHMM using the aligned training sequences for each family. To classify a new protein, we use `hmmsearch` to search its unaligned amino acid sequence against the 17,929 trained pHMMs. Note that neither the ProtCNN nor our model have access to the alignment information used to train the pHMMs. We used `hmmbuild` from HMMER 3.1b2 to construct a pHMM from the aligned train sequences for each family in Pfam 32.0. We implement a simple top pick strategy to avoid any handicap from the filters built into HMMER 3.1b2. We first use 'hmmsearch' to search all 17,929 profiles against each unaligned test sequence and report at least one hit (using '--max' if necessary). We then call the profile with the highest score as the prediction.

ProtCNN and ProtENN: ProtCNN uses a residual network convolutional neural network architecture with dilated convolutions, with a final classification layer to predict which family a protein domain sequence belongs to [2]. Sequences are represented using one-hots and presented to the model in batches, where each sequence is padded to the length of the longest sequence in the batch. At train time, the weights and biases of the model are updated using standard forward and back propagation. Each ProtCNN model consists of 5 residual blocks and 51M parameters. To further improve performance, 59 separate models are ensembled together to build ProtENN.

ProtTNN: A protein BERT model that is pretrained across the Uniref50 dataset then fine tuned on Pfam to predict family membership [6]. This approach outperforms the ProtENN ensemble.

UniProt Name Prediction

BLAST. BLASTp is arguably a practitioner's default choice for functional annotation, moreover BLAST-based protein names are by construction interpretable, because the annotation derives from a similar protein sequence [8]. We implemented an alignment-based baseline in which BLASTp is used to identify the closest sequence in the training set to a query sequence. Labels are then imputed for the query sequence by transferring those labels that apply to the annotated match from the train set. We use BLASTp as a one-nearest neighbor algorithm by first using 'makeblastdb' (version 2.7.1+) with the training data. We then query sequences from that database using 'blastp -query', taking only the top hit by bit score.

Evaluation Metrics

Pfam descriptions

We train on a clustered split of Pfam seed v32.0 [2] using the task of "given the amino acid sequence, predict the DE line of the Pfam-A.seed Stockholm file". For example, for the GPCR family PF00001, the correct label is "7 transmembrane receptor (rhodopsin family)".

When we evaluate, we consider whether a label is correct at the "lifted clan" level, as described in [2]. In the case of predicting family descriptions, this means that

1. If a prediction is not the description for any family, it's wrong.
2. If the prediction is the description of a family *F* that's not in a clan, the description is only correct if the input sequence is in family *F*.
3. If we predict a label that is the description of a family *F* that is in a clan *C*, the description is only correct if the input sequence is in clan *C*.

UniProt protein names

Evaluation using UniProt names: The trivial evaluation approach is to select proteins that have at least one ground truth name and compute the degree of overlap between the prediction and ground truth name(s). We consider the following metrics:

- *exact match*: we check whether a predicted name matches any ground truth name exactly, including capitalization,

- *substring match*: we check whether a predicted name can be found as a substring of any ground truth name. For substring match we consider both a case sensitive and a case insensitive search. On the dev set, we find that 57% of predictions are corroborated by substring match when including capitalization. This number increases to 60% when ignoring capitalization. Given the small difference, we report only the more stringent (case sensitive) metric in the main paper.

When computing these metrics, we use as ground truth all the UniProt names for the protein. Additionally, whenever the name can be found in the training set, we consider augmenting the

list of ground truth names by also including the preprocessed version of each ground truth name, to ensure that propagating the name found in training is counted as correct. We show results both with raw ground truth names and results with augmented ground truth names. For instance, on the dev set, we observe that using the augmented set of candidate ground truth names results in only a slightly higher percentage of corroborated predictions when using case sensitive matches, with the exact match corroboration percentage increasing from 54 to 56% and the case sensitive substring match percentage increasing from 57 to 59%. When using the case insensitive substring match the percentage is 60% both with and without augmentation. These differences are small, likely due to the fact that the fraction of protein names updated via preprocessing is small. Furthermore the processing typically removes tokens at the beginning or end of the name, a variation often accounted for when using the substring match approaches, especially when ignoring case. Given the small difference, the exact match and substring match results in the main paper use the more stringent version which only evaluates overlap with respect to the raw ground truth names.

Finally, we observe that when multiple ground truth protein names are available, they often have little to no overlap to each other because either (1) they encode similar information with very different terms, or (2) they refer to different known properties of the protein. This means that even alternative ground truth protein names may score poorly under these metrics. This issue also arises in image captioning and predicted captions are typically evaluated against multiple ground truth candidates [9].

Manual Evaluation: To address the automatic evaluation challenges mentioned above, we also performed manual evaluation. Similar to above, given a protein accession and a predicted protein name, the goal was to determine whether the predicted name contains accurate information about the protein. Resources that a curator might turn to include computational tools such as Interpro [10] and BLAST [8].

A methodology outlined in the image captioning literature [e.g.11, 9] prescribes that each predicted caption is annotated using a score from 1 to 4, where the scores have the following interpretations: [1] the image is described with an unrelated description, [2] the image is described with a somewhat related description, [3] the image is described with minor errors, [4] the image is described without any errors. While we found this scoring scheme to be somewhat informative about the performance of our models in early experiments on the validation set, we also identified several limitations relevant to our specific use case:

Mistakes vs. information that cannot be corroborated. A predicted caption may contain information that cannot be corroborated due to incomplete annotations in the existing databases, and this type of predicted caption should not necessarily be discouraged as it can potentially lead to model-assisted scientific discoveries. Information that is known to be wrong according to the literature should be strongly discouraged, so we updated the scoring scheme to differentiate between mistakes and information that cannot be corroborated.

Related descriptions vs. accurate descriptions for related proteins. In protein captioning, it is difficult to consistently assess which descriptions are closely related because this requires understanding the relationships between different protein functions. However, the UniProt database identifies proteins that are similar in sequence to the query protein, making it simple to assess whether a description is accurate for a related protein.

Finally, we asked the curation to rate their subjective preference for the existing vs. the predicted UniProt name. These considerations helped us to prepare the following evaluation instructions:

1. Provide your subjective preference for the UniProt vs. predicted name

[A] I prefer the UniProt name because I think it is more useful.

For instance, one may wish to choose this option whenever the predicted name contains inaccurate information, or when the predicted name is accurate but too coarse grained or vague. The UniProt curators can exercise judgment in determining usefulness. E.g. if a protein is correctly described in more than one way, but one of the ways is more useful to a "general user of UniProt", this judgment may be used. Basically, you can choose this label if you would prefer to use this name for the protein in UniProt vs the predicted label.

[B] I prefer the UniProt name because of formatting.

For instance, the predicted and UniProt names are very similar, but there are slight syntactic differences, e.g. Homer scaffold protein 1 vs Homer scaffolding protein 1.

[C] I have no preference between the two names.

For instance, one may wish to choose this option if the two names are (semantically) the same, and there is no preference for different word orderings or formatting.

[D] I prefer the predicted name because of formatting.

For instance, the predicted and UniProt names are very similar, but there are slight syntactic differences, e.g. Homer scaffold protein 1 vs Homer scaffold**ing** protein 1.

[E] I prefer the predicted name because I think it is more useful.

For instance, one may wish to choose this option if the predicted name is accurate and more specific than the UniProt name. Basically, you can choose this label if you would prefer to use this name for the protein in UniProt vs the current UniProt label.

[F] I cannot pick a preference because I don't have enough information.

For instance, one may wish to choose this option if both the predicted name and UniProt name are related, and the predicted name provides extra information that is not corroborated. E.g. if the UniProt name is NTPase and the predicted name is ATPase, but it's not clear from other sources whether the protein prefers ATP or GTP.

2. Score the predicted name's accuracy

If there is no preference between the names, you do not need to score the accuracy of the predicted name.

We would like you to select one of the following options:

The predicted name …

[4] is known/consistently predicted by a trusted 3rd-party or 3rd-party model

The name contains no errors or pieces of information that are not known to be associated with the protein. Further, a current SwissProt name counts as reliable evidence. A BLAST search where all members are indicating the same function also counts.

[3] is accurate for a very related protein

The predicted name is not fully accurate but rather contains a mix of accurate information and information that is associated with a very related protein, e.g. a protein in the same UniRef cluster. E.g. an InterPro match, or an InterPro match on a confident BLAST hit.

[2] describes the protein with unverifiable information

The predicted name contains pieces of information that we cannot corroborate either positive or negative. For instance, one may select this option if the UniProt name has the general enzyme class and we predict the same enzyme class but with greater specificity, and we can't corroborate the greater specificity, even by looking at closely-related sequences.

[1] contains errors or is uninformative

The predicted name contains errors or is a name that mainly indicates uncertainty about the protein function and/or it can be applied to any protein. For instance "uncharacterized protein", "predicted protein", "putative protein", are uninformative names. However, "multifunctional protein" is an informative name as it indicates that the protein has multiple functional regions and this is the name that UniProt recommends as a top name whenever the protein has at least 3

functional regions. Obvious structural nonhomology is another case where this label is warranted.

3. Provide notes about your assessment as you feel appropriate

Manual Evaluation - proteins drawn from the temporal dev set: We took one shard of the dev set (8625 proteins, drawn at random). We then removed any accessions that have since been deprecated or whose name contained "hypothetical protein" (ignoring case) to leave 7563 proteins. We found that 57.01% of ProtNLM-predicted protein names can be automatically corroborated as an exact match or substring of the ground truth UniProt protein names.

To manually curate ProtNLM-predicted names for the remaining sequences, we selected 10 proteins from each ProtNLM confidence bucket from (0, 0.1] to (0.9, 1] at random.

Manual Evaluation - uncharacterized proteins: We drew 1000 proteins at random from UniProt, and excluded all proteins whose UniProt protein names did not include "uncharacterized protein" or "hypothetical protein" to leave a set of 247 proteins. We further excluded all proteins that had subsequently been deprecated from UniProt, or whose ProtNLM-predicted protein name was "hypothetical protein" (ignoring case), or an EC number, as we would not recommend these predictions, to leave 214 proteins. While some of these sequences have been introduced before our training data cutoff, they either:

- Had the name "uncharacterized protein", which means that they were not included in the training set,
- Used to have a name but that name was changed to "Uncharacterized protein" recently, this means the sequence could be included in the training data with the wrong name.
- Had the name "hypothetical protein" and we predicted a non-hypothetical name.

To perform protein captioning for these 214 uncharacterized proteins with BLAST, we ran a BLAST search against the model's training set, and propagated the captions of the highest scoring BLAST hit, using the BLAST scores as confidence scores. We found that among the **214** proteins, there were **59** proteins where the prediction was found exactly among the UniProt names of the BLAST top hit protein. From the remaining **155** proteins, there were **6** proteins without a BLAST hit:

Accession	Prediction	Confidence
E6VMT4	TetR family transcriptional regulator	0.39
A0A2D7ZYA8	Secreted protein	0.19
A0A2D7KW47	Transcriptional regulator	0.32
A0A2H6H598	AraC family transcriptional regulator	0.05

A0A150GX88	Expressed protein	0.14
A0A1L5R398	Diguanylate cyclase	0.01

Evaluation using the full UniProt entry: As described in the main text, we were inspired by our interaction with the professional curators to develop an automated verification tool that uses the full UniProt entry information in evaluation. This includes alternative names, descriptions, protein function annotations, and so on. At its core, this metric looks at whether a predicted name is a substring of the full UniProt entry information. However, there are a few modifications that we make when searching for a name. For instance, a model may predict “AAA domain-containing protein.” If the protein has a call from an InterPro signature called “AAA ATPase domain” (IPR041664), we’d like to recover this information, and count this prediction as correct [10]. So, we strip off the suffix `domain-containing protein` before checking whether the prediction is a substring. We do a few additional cleanups to canonicalize a protein name prediction, described in python here:

```python
def _strip_end(prediction: str, suffix: str) -> str:
  """Strips the suffix from the end of `prediction`."""
  return re.sub(f'{suffix}$', '', prediction)

def sanitize_prediction_for_substring_match(prediction: str) -> str:
  """Removes extra words like 'domain-containing protein' from prediction.

  The goal is to get a prediction string that has the same meaning as
  `prediction`, but is easier to find as a substring match.

  Args:
    prediction: str.

  Returns:
    string with many of the lower information-content words (like 'protein')
    removed.
  """
  prediction = prediction.rstrip('.,')

  prediction = prediction.replace('family protein', '')
  prediction = _strip_end(prediction, 'domain-containing protein')
  prediction = _strip_end(prediction, 'repeat-containing protein')
  prediction = _strip_end(prediction, 'protein')

  prediction = prediction.rstrip(' ')

  return prediction.lower()
```

This approach of substring-detection has some drawbacks:

- It still has false negatives - as a community, we still do not know what many proteins do. Further, this problem is amplified because oftentimes annotations are already propagated to neighbors.
- It has false positives. For example, we do a case-insensitive search to try to alleviate false-negatives, which trades off against false positives, because case information often carries semantics. As a concrete case of such false-positives, our model predicted "VOC-domain containing protein" for protein A0A1X0ZS48, and our metric detected the string "VOC" in the UniProt page of protein A0A6D1WG03, found in the same UniRef50 cluster, despite the substring being part of one of the authors' names.

Evaluation using UniRef50 members: We can also look for any occurrence of our ProtNLM-predicted name in any related protein, and the precomputed UniRef clusters (from UniRef50 2022_01) provide a very convenient way to do so for every protein in UniProtKB [4]. In this case, we consider a protein's predicted name to have "evidence" that is correct when any members of the same UniRef50 cluster mention the predicted name in their full UniProt entry.

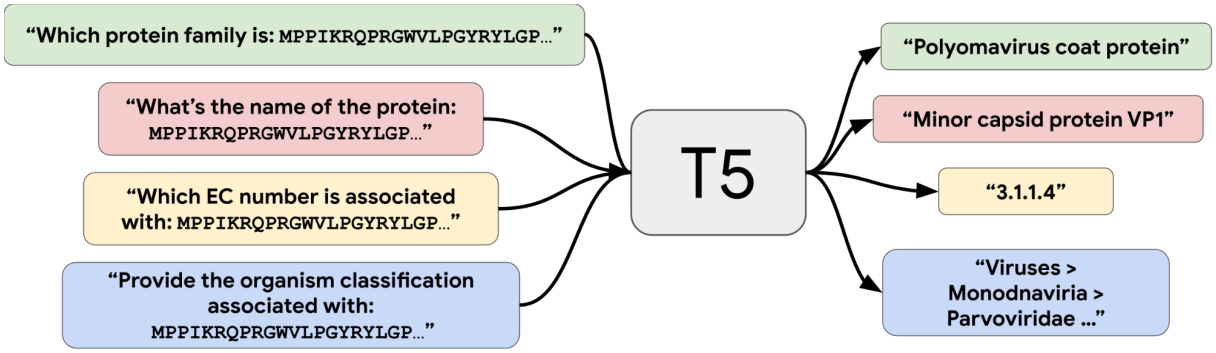
Modeling approach

Models

In natural language processing, a successful modeling paradigm has been to formulate problems as text-to-text tasks that can be solved jointly by training a single sequence-to-sequence model. In particular, Raffel et al. (2020) outlined how sequence-to-sequence modeling can be applied not only to tasks that can be expressed naturally as text (such as document summarization and language translation), but also to tasks that involve categorical or numerical data. Furthermore, the work demonstrates that training a single model on multiple sequence-to-sequence tasks specified via natural language prompts can improve performance compared to learning each task in isolation.

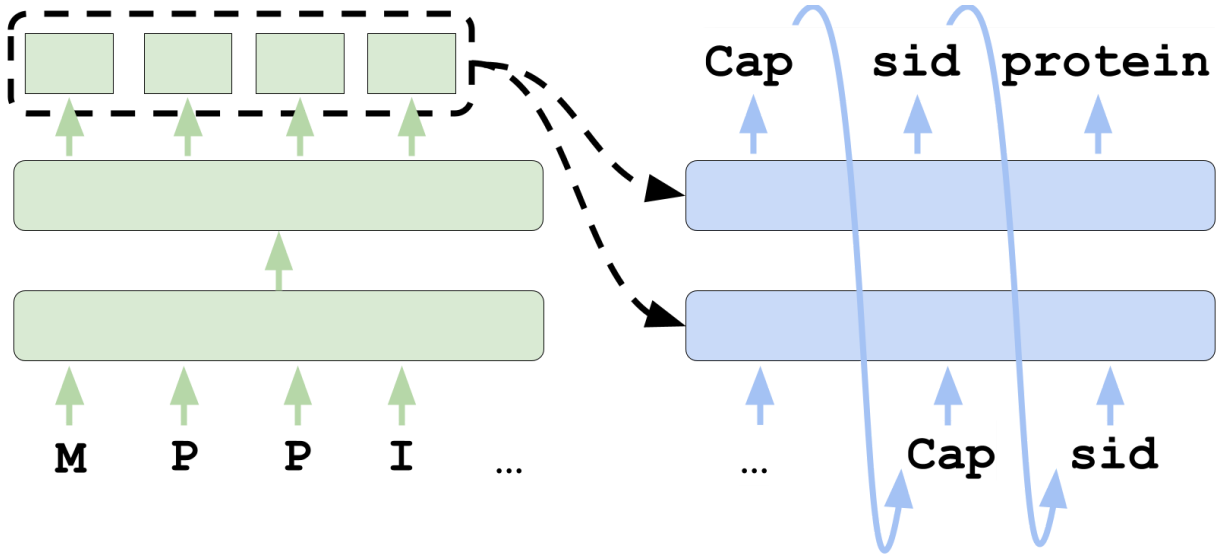
We use the same methodology to train models which take amino acid domain sequences as inputs and return various protein properties. For instance, in the Pfam domain description case, we train a model which takes in domain amino acid sequences and produces the one-line description of the corresponding Pfam family, in addition to the family accession label and a unique alphanumeric family ID. In the UniProt name prediction case, we train a model which takes in full protein amino acid sequences and produces the UniProt name.

We note that as an extension of our work, we could train a single model across both sets of tasks, and include additional protein-level protein annotation tasks, such EC number and GO label prediction.



[Figure adapted from [12].]

One could use a variety of architectures for modeling the resulting sequence-to-sequence problem, including encoder-decoder models, decoder-only models, or prompting language models. Following Raffel et al. (2020), we use a standard transformer model-based encoder-decoder framework [13]. The encoder takes in the prompt and amino acid sequence encoded as a sequence of tokens and produces a variable length embedding of the same size as the input. The decoder takes as input the encoder representation and produces the model outputs, one per input prompt, via auto-regressive decoding.



Training Details

We treat the two tasks - Pfam domain description prediction and UniProt name prediction - separately and make problem specific modeling decisions as outlined below.

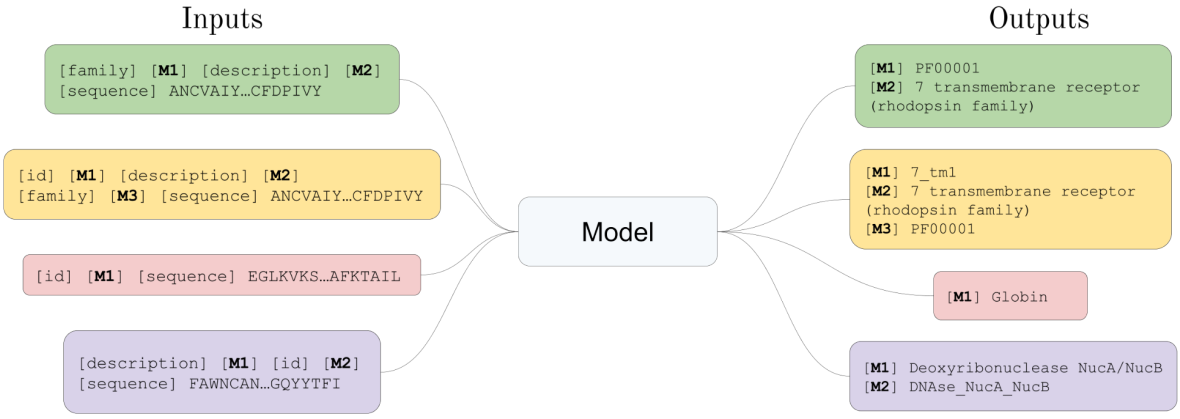
For instance, the dataset used for the Pfam domain description prediction task has approx. 1.3 million training examples and previous work demonstrated that pretraining on the UniRef50 dataset can help improve performance [6]. As a result, we use a pre-training plus fine-tuning training strategy for this task.

The dataset available for training the UniProt name prediction task is significantly larger and we train models from scratch on this task.

Pfam Domain Description Prediction

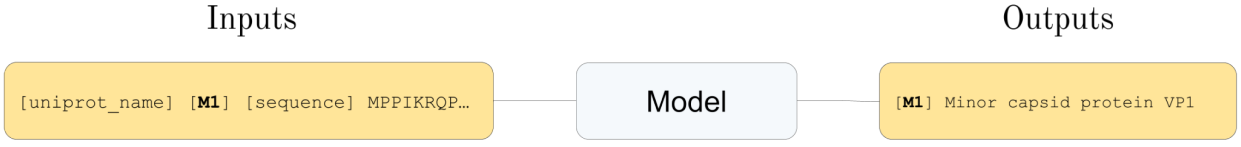
Pretraining. Following prior work on learning representations for proteins [14, 15], we pre-train the model on the clustered Uniref50 database. Specifically, we use a dataset consisting of the center of each cluster to encourage diversity among sampled sequences. The dataset contains 30 million sequences. We use the T5) span denoising task as a pretraining task: given a sequence of amino acids, we mask out spans of tokens and train the model to predict the contents of these spans [12].

Fine-tuning. After pretraining, the model is trained to predict protein family accessions, ids, and their natural language descriptions. Each time a sequence is selected for a batch of data, the training algorithm chooses a random set of fields from accession, id, and description. These fields are presented in a randomly permuted order before the protein sequence, and the model is trained to predict the output for each prompt in the given order. In this way, the model learns to predict any subset of tags about a sequence: it can predict just the class, generate a description of the protein, or predict all of the description, id, and sequence fields at once. The prediction is done by *decoding* a sequence of subword tokens from the model auto-regressively, conditioned on the protein sequence and the fields to predict.



UniProt Name prediction

Fine-tuning. As mentioned above, due to the large set of labeled sequences available, we train the UniProt name prediction models from scratch. For this task we use a single field, the uniprot name. We train our models on sequence-name pairs extracted from the UniProt database:



In both the Pfam domain description and UniProt name prediction problems, having expressed our tasks as sequence to sequence tasks, we train the models using the maximum likelihood objective [12]. For each protein, the model received the protein amino acid sequence, truncated to 512 total tokens.

Vocabulary

Constructing a suitable vocabulary is essential to effectively representing the protein sequences and generating natural language descriptions.

Pfam Domain Description Prediction

For the Pfam domain description task, to leverage pretraining, we train a model to both denoise amino acid sequences and to generate natural language descriptions and other domain annotations using the same encoder-decoder weights. As a result, the model requires a shared vocabulary capable of encoding protein sequences, plain text descriptions, and categorical labels.

We use the sentencepiece library, specifically the unigram language model [16], to generate a vocabulary suitable to encode both the amino acid sequences and the natural language descriptions. We additionally introduce a single token per Pfam family accession to enable each Pfam family to have its own embedding. While it is possible to encode Pfam accessions as a sequence of tokens (e.g. PF11916 → PF _11 _91 _6), in our initial experiments, we obtained better performance with models which had a single token embedding per class.

To construct the vocabulary, we first trained two separate vocabularies, each with 8192 tokens: one on amino acid sequences and one on the set of Pfam family text descriptions. This prevents one data modality from dominating the vocabulary. We then combine these vocabularies, adding in an additional token per family accession (e.g. PF11916) to allow each family accession to be encoded via a single token and therefore to have its own embedding. The vocabulary contains additional control tokens for the different fields to predict, [description], [accession], and [id], as well as a [sequence] control token that always precedes the amino acid sequence. We lowercase all natural language text, leaving amino acids uppercase. The same vocabulary is used for the T5 encoder and decoder models.

UniProt Name Prediction

For this task we use an amino-acid specific vocabulary for the encoder and a text-specific vocabulary for the decoder.

Encoder vocabulary. We assume that we don't want any groups of amino-acids in the vocabulary, so we construct a vocabulary consisting of single amino-acid tokens. We use the set of 20 standard amino acids and assume that any alternative amino acids would be mapped to the same “unknown” token. Despite training on a single task, for consistency across models, we include prompt specific tokens: [sequence] and [protein_name_in_english].

Decoder vocabulary. Similar to the Pfam task, we use the unigram language model to construct a vocabulary with 32K tokens using the set of all unique names found in the temporal split training set [16]. In the resulting vocabulary, common words such as “protein”, “domain”, “binding”, “transporter”, “transferase” are mapped to a token, while less common words such as “Tetratricopeptide” are split into multiple tokens, e.g. “T” + “etratricopeptide”.

Finally, using separate encoder and decoder vocabularies enables the capitalization of UniProt names to be kept intact, while avoiding any confusion between uppercase letters and amino acid symbols. In particular, similar to natural language processing, lower-casing protein captions is desirable because it enables the model to recognize the same word, whether it appears in the beginning of a protein caption with an initial capital letter, or later in the caption without the initial capital. E.g. "exoribonuclease" appears in both "Exoribonuclease, phosphorolytic domain 1" and "5'-3' exoribonuclease" with the same meaning in both contexts. On the other hand, protein captions include terms that contain a mix of lowercase and uppercase characters and the capitalization needs to be recovered before providing the captions to the end user. To this end, in this version of the paper, we chose to keep-case all protein captions and rely on the data-specific tokenization to model the case variation. For instance “Tetratricopeptide” and “tetratricopeptide” are tokenized into “T” + “etratricopeptide” and “t” + “etratricopeptide” respectively.

Inference

Generating predicted names for a new protein sequence involves passing the sequence, together with tags indicating which fields we wish to predict, into the model. As is standard, we then sample one token at a time in an autoregressive manner from the decoder, conditioned on the encoded input. This can be done either using standard temperature sampling or beam search, ranking by the likelihood under the model.

Beam search is a greedy algorithm for estimating the maximum likelihood prediction, here used to generate descriptions of proteins, based on the model's estimation of how likely each description is conditioned on its amino acid sequence. This algorithm is commonly used in generative modeling in machine learning. Our models generate text left-to-right, and as such, when sampling the next token as part of a protein's description, some search algorithm over potential “next tokens” is a useful component of description generation. The result of a beam search is a set of K beams sampled from the model, ranked by how much probability the model assigns to the sequence. For the reported results we use beam search with beam size 10. The model returns 10 outputs, and we report evaluation results corresponding to top prediction.

Finally, we perform the processing steps outlined below. Unless specified otherwise, the reported results use the post-processed predictions.

Post processing

Short proteins. We do not provide name predictions for proteins who have less than 20 amino acids as we assumed there would not be enough information for the model to make predictions.

EC number translation. Because we trained our model on all names that appear as overall or “Includes” names in UniProt, including short names and EC numbers, our model sometimes predicts such names. However, UniProt does not recommend EC numbers as protein names, so we aim to translate EC numbers to the corresponding description.

We use the enzyme.dat file from https://ftp.expasy.org/databases/enzyme. Note however that this mapping contains only full EC numbers (no dash-containing number). We use this mapping when the EC number is found.

Note that when a protein has an EC number as a name in the .dat file, it also has a corresponding full name description. We expect (and observed in a small sample of examples) that the alternative predictions produced by the model often contain a full name description that can be used instead. Nevertheless, we only translated the names found in the enzyme.dat file and flag the remaining predictions for removal (see below).

Filtered-out predictions. We flag for removal the following predictions:

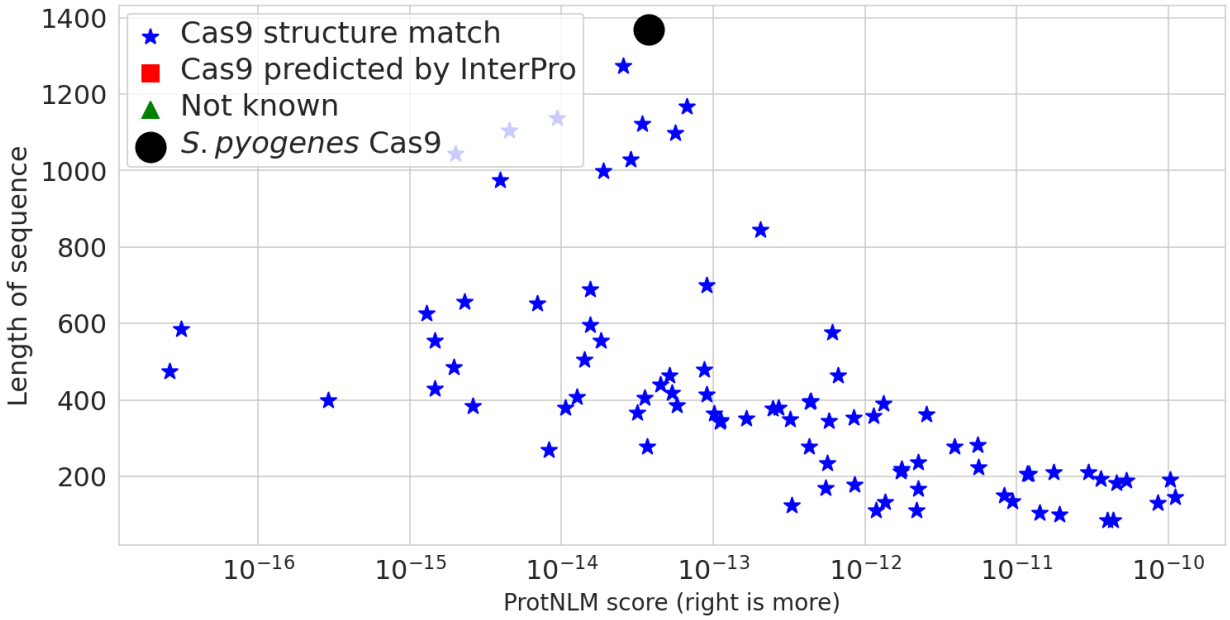
- Predictions containing the terms “hypothetical”, “uncharacterized”, “expressed protein”, “unnamed product”, “str fm”, “str. fm” (ignoring case).
- Predictions which did not contain any letters. We assumed these predictions are EC numbers or otherwise IDs that should not be recommended as UniProt names.

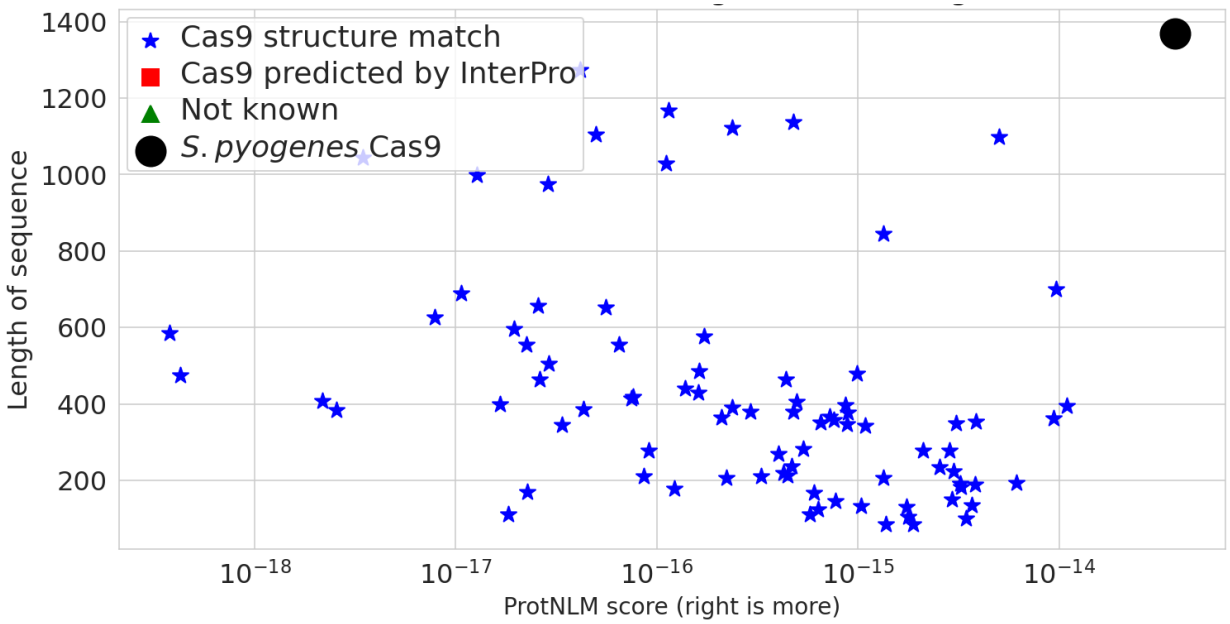
CRISPR/Cas9 annotation

In order to disentangle whether the model has learned if “Small” means anything in the context of ranking proteins by their likelihood under the label “Small Cas9 Homolog”, we perform an ablation experiment where we leave out various words from the prompt, and consider only the Foldseek+AlphaFold corroborated Cas9 homologs from our set of 500 putative Cas9 homologs. On this set, we compute the correlation between sequence length and ProtNLM score.

We find that including the word “Small” increases the negative rank (Spearman) correlation between sequence length and ProtNLM score. We further find that, although including “Large” instead of “Small” doesn’t switch the correlation to a positive correlation between sequence length and negative sequence length, we see that the rank correlation has been substantially diminished, and the Pearson correlations have no statistical significance.

prompt	pearsonr	pearsonr_pvalue	spearmanr	spearmanr_pvalue
Small Cas9	-0.306325	0.004355	-0.778012	1.958016e-18
Small Cas9 homolog	-0.348189	0.001093	-0.790353	2.393076e-19
Cas9	-0.313470	0.003486	-0.634554	6.991804e-11
Cas9 homolog	-0.351754	0.000963	-0.620219	2.447220e-10
Large Cas9	0.173144	0.113042	-0.170545	1.186360e-01
Large Cas9 homolog	-0.143788	0.189228	-0.486073	2.405806e-06





References

1. Mistry, Jaina, et al. "Pfam: The protein families database in 2021." Nucleic acids research 49.D1 (2021): D412-D419.
2. Bileschi, Maxwell L., et al. "Using deep learning to annotate the protein universe." *Nature Biotechnology* (2022): 1-6.
3. "UniProt: the universal protein knowledgebase in 2021." *Nucleic acids research* 49, no. D1 (2021): D480-D489.
4. Suzek, Baris E., et al. "UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches." Bioinformatics 31.6 (2015): 926-932.
5. McKinney, Wes, and P. D. Team. "Pandas-Powerful python data analysis toolkit." Pandas—Powerful Python Data Analysis Toolkit 1625 (2015).
6. Dohan, David, et al. "Improving protein function annotation via unsupervised pre-training: Robustness, efficiency, and insights." *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 2021.
7. Eddy, Sean R. "Accelerated profile HMM searches." PLoS computational biology 7.10 (2011): e1002195.
8. Altschul, Stephen F., et al. "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." Nucleic acids research 25.17 (1997): 3389-3402.
9. Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
10. Blum, Matthias, et al. "The InterPro protein families and domains database: 20 years on." Nucleic acids research 49.D1 (2021): D344-D354.
11. Hodosh, Micah, Peter Young, and Julia Hockenmaier. "Framing image description as a ranking task: Data, models and evaluation metrics." *Journal of Artificial Intelligence Research* 47 (2013): 853-899.

12. Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer." *J. Mach. Learn. Res.* 21.140 (2020): 1-67.
13. Vaswani, Ashish, et al. "Attention is all you need." Advances in neural information processing systems 30 (2017).
14. Elnaggar, Ahmed, et al. "ProtTrans: towards cracking the language of lifes code through self-supervised deep learning and high performance computing." IEEE transactions on pattern analysis and machine intelligence (2021).
15. Rives, Alexander, et al. "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences." Proceedings of the National Academy of Sciences 118.15 (2021): e2016239118.
16. Kudo, Taku. "Subword regularization: Improving neural network translation models with multiple subword candidates." arXiv preprint arXiv:1804.10959 (2018).
17. MacDougall, Alistair, et al. "UniRule: a unified rule resource for automatic annotation in the UniProt Knowledgebase." Bioinformatics 36.17 (2020): 4643-4648.
18. Zhou, Naihui, et al. "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.
19. Papineni, Kishore, et al. "Bleu: a method for automatic evaluation of machine translation." Proceedings of the 40th annual meeting of the Association for Computational Linguistics. 2002.

Oct 12 2022