Final Project: CSE 587

Archisman Ghosh
CSE Department
Penn State University
State College, PA, USA
apg6127@psu.edu

Debarshi Kundu
CSE Department
Penn State University
State College, PA, USA
dqk5620@psu.edu

***Abstract*—Functional annotation of hypothetical proteins remains a critical bottleneck in genome interpretation, particularly as sequencing technologies continue to outpace functional characterization methods. Traditional homology-based tools like BLAST or InterProScan often fail to predict functions for novel or divergent proteins, leaving a substantial portion of proteomes annotated only as "hypothetical." To address this challenge, we explore the use of large language models (LLMs) for generating functional annotations in natural language, leveraging both protein sequence and contextual information such as Pfam domains and neighboring gene data. We construct a dataset from the SwissProt database by extracting reviewed protein entries with curated functional descriptions, converting them into instruction-style prompt-response pairs suitable for fine-tuning. Using GPT-2 as the base model, we fine-tune it on 100 examples of protein function descriptions formatted with domain and gene context as input. We evaluate the fine-tuned model using BLEU, ROUGE-L, and BERTScore to measure lexical and semantic similarity between predicted and reference annotations. While BLEU (0.0508) and ROUGE-L (0.1368) scores reflect low n-gram overlap, the model achieves a strong BERTScore F1 of 0.7732, indicating high semantic relevance of the generated outputs. Our results demonstrate that LLMs can generalize beyond sequence similarity, inferring biologically meaningful functions based on contextual signals. This approach offers a promising direction for enriching genome annotations, particularly in cases where traditional sequence alignment tools fall short.**

***Index Terms*—Large Language Models, GPT-2, Protein Annotation**

I. Introduction

Proteins are central to virtually all biological processes, and understanding their functions is essential for interpreting genomes, designing therapeutics, and exploring the molecular basis of disease. Despite the rapid growth of genomic data, a large proportion of predicted proteins remain functionally uncharacterized and are broadly labeled as "hypothetical." This presents a critical gap in biological knowledge, especially in newly sequenced or poorly studied organisms, where traditional annotation tools such as BLAST or InterProScan [1] often fail due to lack of close homologs or sequence divergence. Accurate functional annotation of proteins is therefore a cornerstone of bioinformatics and a key enabler for downstream applications in biotechnology and medicine. Machine learning, and more recently, large language models (LLMs), offer a new paradigm for approaching this challenge. Trained on vast corpora of structured and unstructured data, LLMs can learn complex patterns and associations, making them well-suited for tasks requiring contextual understanding and natural language generation [2]. In the context of protein annotation, LLMs can synthesize information from protein sequences, domain architectures, and genomic context to generate plausible functional descriptions—even when sequence homology is weak or absent. This project explores the fine-tuning of an open-source LLM to generate natural language annotations of proteins, offering a scalable and interpretable alternative to conventional methods.

A. Motivation and Problem Statement

With the exponential growth of genome sequencing, the need for scalable protein annotation has intensified. Despite advances in homology-based tools, a substantial fraction of predicted proteins remain labeled as "hypothetical" due to insufficient sequence conservation or lack of annotated homologs. These methods rely heavily on evolutionary similarity and are often ineffective for divergent gene families, novel organisms, or metagenomic assemblies. This limitation necessitates alternative approaches capable of inferring function beyond direct alignment. Large language models (LLMs), pre-trained on vast textual corpora, offer a complementary solution by learning statistical associations across biological and natural language features. Their ability to condition on contextual cues such as sequence motifs, domain annotations, and gene neighborhoods allows them to generate biologically plausible functional hypotheses. As such, LLMs present a viable strategy for context-aware, zero- or few-shot functional annotation in data-sparse or alignment-intractable settings.

The goal of this work is to develop a large language model-based approach for the automated functional annotation of hypothetical proteins, using sequence-derived features and contextual metadata as input. Unlike traditional methods that rely exclusively on alignment-based similarity or curated domain databases, our model is trained to infer functional descriptions by learning from natural language descriptions and protein context. Specifically, we frame the problem as a sequence-to-text generation task, where the input includes the amino acid sequence, annotated Pfam domains, and neighboring gene information, and the output is a natural language description of the protein's biological role. We construct a training dataset from manually curated SwissProt entries, fine-tune GPT-2 [3] using instruction-style prompt-response pairs, and evaluate the model using both lexical (BLEU [4], ROUGE [5]) and

semantic (BERTScore [6]) similarity metrics. This formulation enables generalized function prediction in scenarios where classical annotation fails, especially for low-homology or poorly characterized proteins.

B. Contributions

The major contributions of this project are as follows:

1) We demonstrate the feasibility of fine-tuning a language model to generate accurate natural language functional annotations for proteins using contextual biological features.
2) We introduce a curated dataset and evaluation framework combining sequence, domain, and gene context for LLM-based protein function prediction.

C. Report Structure

Section II provides a background and related works. Section III covers the dataset. Section IV describes the proposed idea and the hyperparameters. Section V presents the results. We discuss the results in Section VI and conclude in Section VII.

II. Background

A. Functional Annotation of Proteins

Functional annotation refers to the process of assigning biological meaning to protein sequences, typically describing their molecular roles, involvement in cellular processes, or localization within the cell. It is a foundational task in computational biology, essential for interpreting genomic content, understanding organismal biology, and informing downstream applications such as drug discovery, metabolic modeling, and disease gene identification. Functional annotations are traditionally derived through sequence alignment to well-characterized homologs, domain architecture analysis (e.g., Pfam [7], SMART), and ontology-driven frameworks such as Gene Ontology (GO) [8]. However, these approaches are inherently limited when applied to novel or poorly conserved sequences that lack close homologs or contain uncharacterized motifs. As a result, a large fraction of predicted proteins in newly sequenced genomes are labeled as "hypothetical proteins," with no known function. This bottleneck is particularly acute in metagenomics and non-model organisms, where classical methods often fail to resolve functional roles. In this context, alternative strategies that can infer function from contextual, structural, or sequence-derived features are gaining attention. Emerging machine learning approaches, particularly those leveraging natural language representations, offer a powerful avenue for capturing implicit functional patterns in protein data and generating descriptive, interpretable annotations in biologically relevant terms.

B. LLMs in bioinformatics

Large Language Models (LLMs) are autoregressive generative models trained to estimate the probability distribution of token sequences. GPT-2, a prominent LLM developed by OpenAI, is based on the Transformer decoder architecture and models the joint probability of a sequence $x = (x_1, x_2, \ldots, x_n)$ as:

$$P(x) = \prod_{t=1}^{n} P(x_t \mid x_1, \ldots, x_{t-1}; \theta)$$

where θ denotes the model parameters. The architecture comprises multiple layers of masked multi-head self-attention and position-wise feedforward networks, enabling the model to capture long-range dependencies and contextual semantics. GPT-2 is trained using maximum likelihood estimation (MLE), minimizing the negative log-likelihood loss:

$$\mathcal{L}_{\text{MLE}} = -\sum_{t=1}^{n} \log P(x_t \mid x_{<t})$$

For biological sequence tasks, this formulation allows GPT-2 to learn from structured prompts containing protein sequences, domains, and gene context, and to generate coherent, biologically relevant descriptions as responses. In this work, we adapt GPT-2 to the task of protein function annotation by fine-tuning it on curated sequence-function pairs. Its autoregressive nature enables fluent, interpretable output, while the self-attention mechanism facilitates reasoning over compositional biological features such as motifs and domains, making it well-suited for functional prediction in low-homology contexts.

III. Dataset

The dataset used in this study was constructed from the UniProtKB/Swiss-Prot [9] database, a manually curated repository of protein sequences and functional annotations. We utilized the flat-file format `uniprot_sprot.dat`, which encodes biological data for each protein entry in a structured text format. A custom parser was implemented to extract three primary fields: (1) the amino acid sequence from the `SQ` section, (2) the set of annotated functional domains from lines beginning with `FT DOMAIN`, and (3) the free-text functional description from `CC -!- FUNCTION` comment lines.

Entries lacking a sequence or functional annotation were excluded. The dataset was capped at 1000 curated examples to maintain a balanced size for fine-tuning. Each sample was formatted into an instruction-response pair compatible with language model training, and serialized into JSON Lines (JSONL) format (Fig. 1).


```
{
  "prompt": "Protein Sequence
    :\nMAFSAEDVLKEYDRRRRMEALLLSLYYPNDRKLLDYKEWSPPRVQVECPKAPVEWNNPPSEKGLIVGHFSGIKY
    KGEKAQASEVDVNKMCCWVSKFKDAMRRYQGIQTCKIPGKVLSDLDAKIKAYNLTVEGVEGFVRYSRVTKQHVAAFL
    KELRHSKQYENVNLIHYILTDKRVDIQHLEKDLVKDFKALVESAHRMRQGHMINVKYILYQLLKKHGHGPDGPDILT
    VKTGSKGVLYDDSFRKIYTDLGWKFTPL\nDomains: None",
  "response": "Transcription activation. {ECO:0000305}"
}
```


Fig. 1. Example of a training instance in the JSONL dataset used for fine-tuning. Each entry includes a prompt consisting of a protein sequence and associated domain information (if available), and a response containing the curated functional annotation derived from UniProtKB/Swiss-Prot.

IV. Procedure

A. Dataset Preprocessing

The dataset was preprocessed to align with the instruction-based fine-tuning paradigm commonly used for large language models. Each training instance was structured as a prompt-response pair, where the prompt encodes the protein's amino acid sequence and any associated domain annotations in a templated natural language format:

```
### Instruction:
Protein Sequence:
<SEQUENCE>
Domains: <DOMAIN_1>, <DOMAIN_2>, ...

### Response:
<FUNCTIONAL_DESCRIPTION>
```

This format guides the language model to treat function prediction as a conditional generation task. Entries lacking functional annotation or protein sequence were excluded to ensure data quality and consistency. Tokenization was performed using Byte Pair Encoding (BPE) [10], the subword segmentation algorithm used in GPT-2. BPE iteratively merges the most frequent pair of adjacent tokens in the vocabulary to form new tokens, balancing vocabulary size and the ability to represent rare sequences. Formally, given an input sequence $x = (x_1, x_2, \ldots, x_n)$ over a character alphabet, BPE applies the following transformation:

$$\begin{aligned} x^{(t+1)} &= \text{Merge}(x^{(t)}, (a, b)) \\ \text{where} \quad (a, b) &= \arg\max_{(a,b)} \text{freq}(a, b) \end{aligned} \tag{1}$$

where $\text{freq}(a, b)$ is the frequency of the pair (a, b) in the current token sequence. This process continues until a predefined vocabulary size is reached. BPE allows the tokenization of protein sequences and domain descriptors with high fidelity while retaining efficiency in training. All input sequences were padded or truncated to a fixed maximum length of 512 tokens to ensure consistent input dimensions during training and to accommodate the model's maximum context window.

B. LSTM Architecture

We employed the GPT-2 architecture, a transformer-based causal language model consisting of 12 decoder-only layers, each containing multi-head self-attention, feedforward sublayers, and residual connections. The model comprises approximately 124 million parameters and is trained to optimize the autoregressive objective, modeling the conditional probability of the next token given the previous context. The masked attention mechanism ensures unidirectional information flow, allowing the model to generate biologically relevant annotations by conditioning on structured input features such as sequences and domain context.

C. Hyperparameters

The model was fine-tuned using supervised learning on the task-specific dataset. Training was conducted over 3 epochs using a batch size of 2, with gradient accumulation to simulate larger batch behavior. The AdamW optimizer was used with a learning rate of $5e-05$, and weight decay was applied with a coefficient of 0.01 to prevent overfitting. Training utilized mixed precision (FP16) when supported by the hardware to reduce memory usage and increase throughput. Model checkpoints and logs were saved at the end of each epoch, and evaluation was performed using exact match accuracy between predicted and reference responses. A detailed desription is given in Table I

TABLE I
Training Hyperparameters for GPT-2 Fine-tuning

Hyperparameter	Value
Model	GPT-2 (117M parameters)
Epochs	3
Batch Size (per device)	2
Sequence Length	512 tokens
Learning Rate	5×10^{-5}
Weight Decay	0.01
Optimizer	AdamW
Gradient Accumulation	1
Mixed Precision (FP16)	Enabled (if supported)
Logging Steps	10
Evaluation Strategy	Epoch
Save Strategy	Epoch
Loss Function	Cross-Entropy (NLL)

V. Results

A. Simulation Setup

The classifier was implemented using PyTorch libraries and the pre-trained GPT-2 model was used from the `transformers` library from HuggingFace, and run on a machine with 16GB RAM on an Intel Core i7-6700 CPU at a clock frequency of 3.40 GHz.

B. Classification

The fine-tuning process for GPT-2 in the task of protein function annotation employs the standard *causal language modeling* (CLM) objective. Given a tokenized input sequence $x = (x_1, x_2, \ldots, x_n)$, the model is trained to predict each token x_t conditioned on all previous tokens $x_{<t}$, using the autoregressive formulation. The objective is to minimize the *negative log-likelihood* (NLL) loss, defined as:

$$\mathcal{L}_{\text{NLL}} = -\sum_{t=1}^{n} \log P(x_t \mid x_1, x_2, \ldots, x_{t-1}; \theta)$$

where θ denotes the parameters of the model. The GPT-2 model learns to assign high probability to the ground truth tokens by updating its parameters through backpropagation. During training, the loss is computed over all non-padding tokens, and gradients are accumulated across steps to compensate for small batch sizes. The trend of loss over the epochs is shown in Fig. 2. We can observe a decreasing trend

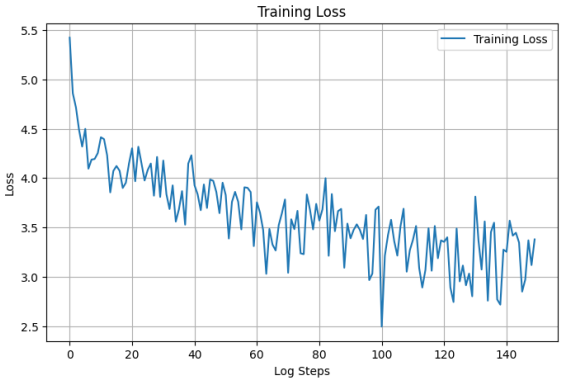

Fig. 2. Training loss curve for the fine-tuned GPT-2 model on the protein function annotation dataset. The loss decreases over successive logging steps, indicating progressive optimization of the model's parameters during supervised fine-tuning using the causal language modeling objective.

highlighting the successful training of the fine-tuned GPT-2 model. This setup enables the model to learn mappings from structured biological prompts—including protein sequences and domain annotations—to coherent and biologically plausible functional descriptions. Fine-tuning effectively aligns the pretrained distribution of GPT-2 with the target domain, enhancing its applicability to protein annotation tasks without sacrificing language fluency or contextual coherence.

C. Metrics

To evaluate the performance of the fine-tuned GPT-2 model in generating biologically accurate and semantically coherent protein function annotations, we employed three metrics: BLEU, ROUGE-L, and BERTScore. These metrics capture lexical overlap and semantic similarity between the generated responses and the ground truth annotations.

1) BLEU-score: The Bilingual Evaluation Understudy (BLEU) score measures n-gram precision between the predicted output and the reference text. For a candidate sentence c and a reference sentence r, the BLEU score with brevity penalty BP is computed as:

$$\text{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right)$$

where p_n is the modified n-gram precision, and w_n are weights (typically uniform). The brevity penalty compensates for overly short predictions:

$$BP = \begin{cases} 1 & \text{if } |c| > |r| \\ \exp\left(1 - \frac{|r|}{|c|}\right) & \text{otherwise} \end{cases}$$

The average BLEU score obtained was **0.0508**, indicating low exact lexical overlap, as expected in free-form biomedical descriptions.

2) ROUGE-L metric: The ROUGE-L metric computes the F1 score based on the longest common subsequence (LCS) between the predicted and reference texts. Given candidate c and reference r, the ROUGE-L F1 score is:

$$\text{ROUGE-L}_{F1} = \frac{(1+\beta^2) \cdot \text{LCS}(c,r)}{\text{len}(c) + \beta^2 \cdot \text{len}(r)}$$

where β controls the balance between precision and recall. This metric captures structural similarity and is more tolerant to rephrasing. The model achieved a ROUGE-L F1 score of **0.1368**.

3) BERT-score: BERTScore evaluates semantic similarity by computing token-level cosine similarity using contextual embeddings. For a candidate sentence $c = \{c_1, \ldots, c_n\}$ and a reference $r = \{r_1, \ldots, r_m\}$, the F1 component is:

$$\text{BERTScore}_{F1} = \frac{1}{n}\sum_{i=1}^{n} \max_{j \in [1,m]} \text{cosine}(\mathbf{e}(c_i), \mathbf{e}(r_j))$$

where $\mathbf{e}(x)$ denotes the contextualized embedding of token x. This metric is robust to paraphrasing and better captures meaning in biomedical contexts. The model attained a high BERTScore F1 of **0.7732**, indicating strong semantic alignment between predicted and true annotations.

VI. Discussion

In this section, we discuss the approach to the problem, potential improvements, and future research.

A. Dataset

While the dataset derived from UniProtKB/Swiss-Prot [9] provides high-quality, manually curated annotations, its limited size (1,000 entries) and taxonomic bias toward well-studied model organisms restrict the generalizability of the fine-tuned model. The lack of representation from divergent taxa and under-characterized protein families reduces the model's applicability to real-world scenarios, such as metagenomic annotation or orphan protein function prediction. Moreover, the functional descriptions are often brief and structurally templated, constraining linguistic diversity and negatively impacting lexical overlap metrics such as BLEU and ROUGE. The current representation of domain information as unstructured strings omits the hierarchical relationships inherent in ontologies like Pfam [7], InterPro [1], and Gene Ontology (GO) [8], limiting the model's capacity to capture domain-function dependencies. To scale this framework, future iterations should incorporate larger, heterogeneous datasets from sources such as UniProtKB/TrEMBL [11], UniRef clusters [12], and MGnify [13]. Enriching input representations with structured ontological metadata and leveraging semi-supervised or few-shot learning strategies may substantially improve generalization to low-homology or data-sparse settings.

B. GPT-2

The current architecture employs GPT-2, a general-purpose autoregressive language model originally trained on natural language text. While it demonstrates the feasibility of generating functional annotations from structured biological prompts, it lacks inductive biases and domain-specific representations tailored for protein sequences. GPT-2 treats amino acid sequences as generic text tokens, and therefore fails to capture structural or evolutionary features such as residue-level conservation, functional motifs, or 3D proximity. Moreover, without biologically meaningful embeddings, the model may struggle

to disambiguate functionally similar proteins with divergent sequences or interpret rare domain configurations, especially in low-data regimes. This limits its generalizability and biological fidelity. To overcome these limitations, pretrained protein language models such as ESM-2 [14] or ProtBERT [15] can be integrated to provide rich, biologically grounded embeddings.

C. Future Work

To improve on this project idea, we could extend the study in a plethora of ways:

1) Embedding Techniques: Our proposed idea currently uses GPT-2 as the LLM for the annotation of proteins. However, it has several limitations as discussed above. In this part we extend the study by exploring a pre-trained embedding model alongside the GPT-2 architecture. These models are trained on large-scale protein sequence datasets using masked language modeling objectives and implicitly capture structural and functional relationships. By using fixed, high-dimensional embeddings from these models (e.g., ESM-2) as additional input features or conditioning context, the GPT-2 decoder can benefit from prior knowledge of protein biophysics and evolution. This can be achieved via concatenation with token embeddings, prefix tuning, or cross-attention mechanisms, enabling the model to reason over both symbolic and biological representations. Such hybrid architectures can significantly enhance annotation accuracy, especially for proteins lacking close homologs or exhibiting non-canonical domain arrangements.

We observe a layout of our idea (not implemented due to lack of time and resources) in Algorithm 1. It computes a mean-pooled embedding of the protein sequence using a pretrained ESM model. This embedding is projected to GPT-2's hidden space and prepended to the tokenized input prompt. The modified input is passed through GPT-2, which generates a functional annotation conditioned on both sequence features and biological context. ESM embeddings [14] capture structural and evolutionary features from raw sequences, enabling biological understanding beyond lexical similarity. By conditioning GPT-2 on these embeddings, the model gains inductive bias about protein behavior. This fusion helps GPT-2 generate functionally relevant annotations, especially for sequences lacking explicit homologs or seen during pretraining.

2) Lessons Learnt: This project highlights several technical insights into leveraging language models for biological sequence understanding. First, general-purpose models like GPT-2 can be effectively fine-tuned for domain-specific tasks such as protein function annotation when provided with well-structured biological prompts. Second, the quality, diversity, and scale of training data are critical, as limited dataset size and templated language restrict model generalization. Third, tokenization schemes originally designed for natural language may not optimally capture the semantics of protein sequences, motivating the integration of biologically grounded embeddings. Fourth, incorporating pretrained representations from protein language models like ESM can enhance performance by injecting structural and evolutionary priors into the generation process. Lastly, metrics such as BLEU and ROUGE are insufficient alone for evaluating biological relevance, underscoring the value of semantic metrics like BERTScore and the need for domain-specific evaluation criteria in bioinformatics applications.

Algorithm 1: Protein Function Annotation using ESM-augmented GPT-2

Input: Protein sequence S, domain metadata D, pretrained ESM model $\mathcal{E}$, pretrained GPT-2 model $\mathcal{G}$

Output: Generated functional annotation $\hat{y}$

Step 1: Preprocessing and Embedding Extraction

Extract amino acid sequence S from input;

Extract contextual domain features D (e.g., Pfam domains);

Construct prompt $P \leftarrow$ `"Protein Sequence:` S`\nDomains:` D`"`;

Compute tokenized input $x \leftarrow \text{Tokenizer}(P)$;

Compute ESM embedding $z \leftarrow \mathcal{E}(S)$ `// Mean-pooled sequence embedding`

Step 2: Input Fusion

Project z into GPT-2 embedding space: $\tilde{z} \leftarrow Wz + b$, where $W \in \mathbb{R}^{d \times d_z}$;

Form extended embedding sequence: $x' \leftarrow [\tilde{z}; \text{Embed}(x)]$;

Step 3: Generation via GPT-2

Pass extended input through GPT-2: $\hat{y} \leftarrow \mathcal{G}(x')$;

Decode $\hat{y}$ to natural language output using the GPT-2 tokenizer;

return $\hat{y}$

VII. Conclusion

This project investigated the use of large language models—specifically GPT-2—for protein functional annotation via natural language generation. Traditional annotation tools rely heavily on sequence homology, limiting their effectiveness for novel or uncharacterized proteins. To address this, we explored whether a language model could infer function based on amino acid sequences and domain context alone. A curated dataset from UniProt was formatted as instruction-response pairs to fine-tune GPT-2 for this task. Evaluation using lexical and semantic metrics showed promising results: a BLEU score of 0.0508, ROUGE-L F1 of 0.1368, and a high BERTScore F1 of 0.7732, indicating strong semantic alignment with expert annotations. These results demonstrate that LLMs can generate biologically relevant descriptions even with limited lexical overlap. Nonetheless, dataset scale and linguistic diversity constrained generalizability. Future work should incorporate protein-specific embeddings from models like ESM-2 to inject structural priors, and expand datasets using sources such as UniRef and MGnify. This hybrid modeling approach offers

promising improvements in biological interpretability and the scalability of automated protein annotation.

REFERENCES

[1] Philip Jones, David Binns, Hsin-Yu Chang, Matthew Fraser, Weizhong Li, Craig McAnulla, Hamish McWilliam, John Maslen, Alex Mitchell, Gift Nuka, Sebastien Pesseat, Antony F. Quinn, Amaia Sangrador-Vegas, Maxim Scheremetjew, Siew-Yit Yong, Rodrigo Lopez, and Sarah Hunter. Interproscan 5: genome-scale protein function classification. *Bioinformatics*, 30(9):1236–1240, 2014.

[2] Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2025.

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

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA, 2002. Association for Computational Linguistics.

[5] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[6] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[7] T. Paysan-Lafosse, A. Andreeva, M. Blum, S. Chuguransky, T. Grego, B. Lazaro Pinto, G. A. Salazar, M. L. Bileschi, F. Llinares-López, L. Meng-Papaxanthos, L. J. Colwell, N. V. Grishin, R. D. Schaeffer, D. Clementel, S. C. E. Tosatto, E. Sonnhammer, V. Wood, and A. Bateman. The pfam protein families database: embracing ai/ml. *Nucleic Acids Research*, 2024.

[8] Paul D. Thomas, David Ebert, Anushya Muruganujan, Takako Mushayahama, Laurent P. Albou, and Huaiyu Mi. Panther: Making genome-scale phylogenetics accessible to all. *Protein Science*, 31(1):8–22, 2022.

[9] The UniProt Consortium. UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49(D1):D480–D489, 2021.

[10] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining, 2020.

[11] Amos Bairoch and Rolf Apweiler. The swiss-prot protein sequence data bank and its supplement trembl in 2000. *Nucleic Acids Research*, 28:45–48, 2000.

[12] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[13] L. J. Richardson, B. Allen, G. Baldi, M. Beracochea, M. Bileschi, T. Burdett, J. Burgin, J. Caballero-Pérez, G. Cochrane, L. Colwell, T. Curtis, A. Escobar-Zepeda, T. Gurbich, V. Kale, A. Korobeynikov, S. Raj, A. B. Rogers, E. Sakharova, S. Sanchez, D. Wilkinson, and R. D. Finn. MGnify: the microbiome sequence data analysis resource in 2023. *Nucleic Acids Research*, 2023.

[14] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, 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):e2016239118, 2021. bioRxiv 10.1101/622803.

[15] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, DEBSINDHU BHOWMIK, and Burkhard Rost. Prottrans: Towards cracking the language of life's code through self-supervised deep learning and high performance computing. *bioRxiv*, 2020.