Protein2Text: Providing Rich Descriptions from Protein Sequences

Edo Dotan, Iris Lyubman, Marcelo Ehrlich, Eran Bacharach, Tal Pupko, & Yonatan Belinkov

The Henry and Marilyn Taub Faculty of Computer Science
Technion – Israel Institute of Technology
Haifa 3200003, Israel
{belinkov}@technion.ac.il

The Shmunis School of Biomedicine and Cancer Research
George S. Wise Faculty of Life Sciences
Tel Aviv University
Tel Aviv 69978, Israel
{talp}@tauex.tau.ac.il

Abstract

Understanding the functionality of proteins has been a focal point of biological research due to their critical roles in various biological processes. However, this endeavor is challenging due to the complex nature of proteins, requiring sophisticated experimental designs and extended timelines to uncover their specific functions. In this work, we introduce BetaDescribe, a collection of models designed to generate detailed and rich textual descriptions of proteins, encompassing properties such as function, catalytic activity, involvement in specific metabolic pathways, subcellular localizations, and the presence of specific domains. The trained BetaDescribe model receives protein sequences as input and outputs a textual description of these properties. The model was trained on datasets containing both biological and English text, which allowed the incorporation of biological knowledge. We demonstrate the utility of BetaDescribe by providing descriptions for proteins that share little to no sequence similarity to proteins with functional descriptions in public datasets. Using *in-silico* mutagenesis, we show that BetaDescribe relies on functionally important regions, as part of its prediction, suggesting that the model identifies regions of importance for the protein functionality without needing homologous sequence. BetaDescribe offers a powerful tool to explore protein functionality, augmenting existing approaches such as annotation transfer based on sequence or structure similarity.

1 Introduction

Since the discovery of the first protein sequence (Sanger & Thompson, 1953), researchers have been fascinated by understanding the intricate functionality of proteins. Proteins play a vital role in almost every biological process, serving as catalysts for chemical reactions, transmitting signals within cells, providing structural support, and much more. Unraveling protein functions is crucial for the advancement of fields such as medicine, agriculture, and biotechnology. However, discovering protein functions can be complex and requires meticulous planning, innovative techniques, and sophisticated instrumentation. Experimental determination of a new protein functionality may take years. As a result, the functions of most proteins across all domains of life are computationally predicted.

In the last decade, artificial neural networks have emerged as a powerful paradigm for solving complex problems in different fields (LeCun et al., 2015) such as computer vision

(Voulodimos et al., 2018), natural language processing (NLP; Young et al. (2018)), speech (Nassif et al., 2019), and structural biology (Jumper et al., 2021). Biological sequences, like natural languages, are composed of discrete characters: letters in human languages, nucleotides in DNA sequences, and amino acids in proteins. These characters form the foundation for more complex structures, such as sentences and genes, which ultimately create documents and genomes (Simon et al., 2024; Dotan et al., 2024). However, there are many differences between the two. While human languages follow known grammatical rules, specific morphological structures, and contextual cues, biological sequences are arranged in highly specific and intricate patterns that convey complex biochemical information. Furthermore, evaluating texts written in natural languages, is relatively straightforward, a simple read-through by a native speaker can reveal errors and convey meaning. Nevertheless, similar properties of protein sequences and English text allow adapting NLP-based techniques for protein analyses (Hayes et al., 2025; Lin et al., 2023; Nijkamp et al., 2023; Zhang et al., 2024).

Early work in the intersection of protein science and deep learning, was primarily focused on classification and regression tasks. Regression tasks include predicting continuous-valued biological properties, such as protein fluorescence (Wang et al., 2022) or stability (Alley et al., 2019; Gong et al., 2023). On the classification side, deep learning models have been developed to predict protein subcellular localization (Elnaggar et al., 2022; Jiang et al., 2023b; 2021), infer structural characterization (Hou et al., 2017), identify type III effectors (Wagner et al., 2022), classify antibiotic resistance genes properties (Li et al., 2021) or recognize antimicrobial activity (Veltri et al., 2018). A major application area has been function prediction, formulated as the task of assigning proteins to Gene Ontology (GO) terms. Many GO predictors have been trained (Cao & Shen, 2021; Gligorijević et al., 2021; Kulmanov et al., 2018; Littmann et al., 2021; Sanderson et al., 2023; Strodthoff et al., 2020; Sureyya Rifaioglu et al., 2019; Yuan et al., 2024).

More recently, protein language models were successfully trained to predict structures from sequences, as exemplified by landmark models such as AlphaFold3 (Abramson et al., 2024) and ESM3 (Hayes et al., 2025). These models demonstrate that transformer-based architectures can internalize physical principles and accurately predict three-dimensional structures from sequence alone, thereby enabling transformative applications in structural biology and drug discovery.

Building on these developments, researchers have begun to explore generative models that go beyond prediction and classification to produce new sequences or descriptions conditioned on biological context. For example, ProGen2 (Nijkamp et al., 2023) demonstrated the feasibility of controlled protein sequence generation. Of particular relevance to our work are models that generate natural language descriptions of proteins, an emerging direction that aims to provide a detailed output tailored specifically to each protein. While recent studies (Abdine et al., 2024; Zhuo et al., 2024) have proposed early approaches to protein captioning, there remains a lack of systematic evaluation regarding the consistency and biological relevance of the generated text.

In this work, we developed BetaDescribe, a collection of models, trained to accurately generate rich textual descriptions of proteins. Given a protein sequence as input, our trained model provides a description that may include several properties, such as the protein's function, catalytic activity, its subcellular localization, and the PTMs it can undergo. The starting model of BetaDescribe is a LLAMA2 model (Touvron et al., 2023), trained on trillions of English tokens. We further trained this model on more than 120 billion tokens containing protein knowledge extracted from UniProt. We tested the performance of BetaDescribe by evaluating the differences between its generated descriptions for protein sequences with a known function and the functions reported in UniProt. We also demonstrate its applicability for predicting the function of proteins with insignificant sequence similarity to any of the proteins used for training.

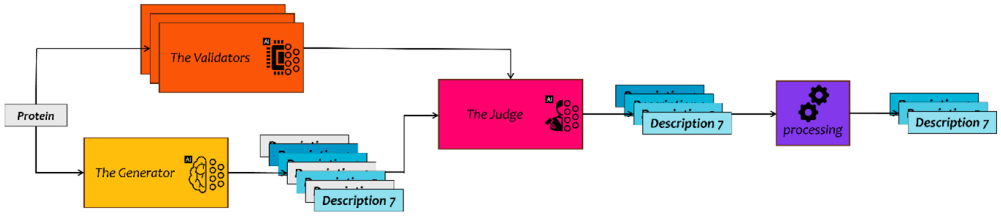


Figure 1: BetaDescribe workflow. The generator processes the protein sequences and creates multiple candidate descriptions. The validators, independently, provide simple textual properties of the protein. The judge receives the candidate descriptions (from the generator) and the predicted properties (from the validators) and rejects or accepts each description. Finally, BetaDescribe provides up to three alternative descriptions for each protein.

2 New Approaches

2.1 Outline

BetaDescribe is a collection of deep-learning models designed to generate and validate optimal descriptions of proteins. This collection comprises three components that we termed *generator*, *validators*, and *judge* (Figure 1). The generator creates rich and detailed candidate textual descriptions for each protein. The validators predict simple properties of proteins (e.g., the subcellular localization of the protein). The judge receives a candidate description (from the generator), and the predicted properties (from the validators) and rejects or accepts the candidate. The generator and the validators process the protein sequence and generate textual (English) descriptions and properties, respectively. The judge processes English text only. Specifically, we trained a large model (7 billion parameters) as the generator, and smaller models (150 million parameters) as validators (since text generation is more complex than its validation). In addition, we harnessed GPT4 (OpenAI et al., 2024) to serve as the judge. Similar techniques to generate and validate solutions have been proposed in the context of code generation (Haluptzok et al., 2023). As the final output, BetaDescribe provides a set of possible descriptions, ranked by their likelihood.

2.2 The Generator

The generator is the central model in BetaDescribe. It is a decoder-only model (Radford et al., 2018) trained to generate textual descriptions of proteins in English, serving as a bridge between the protein domain and the English domain. The model was pretrained on two trillion English tokens by Meta (Touvron et al., 2023). Next, we continued training the model on 120 billion additional tokens, which include protein sequences and their descriptions, thus incorporating biological knowledge. The model was trained to predict the next token, i.e., given a sequence prefix, the model was trained to complete it. The model was trained in several phases, designed to gradually move from general English language modeling to generating textual descriptions of proteins.

2.2.1 Training

Starting from the original LLAMA2 model pretrained on general English text (Touvron et al., 2023), BetaDescribe was trained in three stages to incorporate biological knowledge. In the Stage 1, we introduced Dataset 1 ($\sim 29B$ tokens), a mix of 70% protein sequences from UniRef-90 and 30% general English text from the RedPajama dataset, allowing the model to learn protein data without losing general language abilities. Next, the model was trained on Dataset 2 ($\sim 13B$ tokens) paired protein sequences with their corresponding English descriptions from the UniProt dataset (The UniProt Consortium, 2016), providing biological context and expert vocabulary. This dataset was composed of 45% protein-then-description pairs, 45% description-then-protein pairs, and 10% unrelated English text. In

the final stage, Dataset 3 ($\sim 83B$ tokens) was used to train the model specifically to predict protein descriptions from sequences. Each stage built on the previous one using transfer learning. For specific models, training hyperparameters, and dataset configurations, see Supplementary Information S1.

2.2.2 *Memorization and Generalization*

Multiple proteins in the dataset have the same descriptions, which led the generator to memorize some of them. With a temperature of 0, most of the generated descriptions appear in the training set. This resembles a "search" operation within the description domain. The generation of novel descriptions could provide additional insights. By increasing the temperature hyperparameter, the generator can predict descriptions that are not present in our training set. In a preliminary analysis, a temperature of 1.0 provided a good balance between description diversity and lack of hallucinations (not shown).

2.2.3 *Architecture and Inference*

The model architecture is based on the LLAMA2 model with its default tokenization (Touvron et al., 2023). It is a decoder-only model with seven billion parameters, comprising 32 layers and 32 attention heads. The hidden state size is 4,096. See Touvron et al. (2023) for additional details on the architecture. During inference, we employed two strategies to generate multiple candidate descriptions: varying the prompt, and generating multiple outputs by sampling with the temperature parameter (for more details, see Supplementary Information S2).

2.3 The Validators

We trained three different validators, each predicting a specific protein property given the input protein sequence. We selected properties that can be accurately predicted and are relevant for many proteins. The three properties were: (1) higher-level taxonomic classification, "viruses", "bacteria", "archaea" and "eukaryota", based on the UniProt lineage property; (2) subcellular localization consisting of 388 categories including "acrosome", "chlorosome envelope", "Golgi apparatus lumen", and "plasmodesma". Since proteins may function in more than one location, this classification is multi-labeled; and (3) presence of enzymatic activity (a binary-classification task). The starting point for each validator is the ESM2 base model (Lin et al., 2023), which is associated with 150 million parameters. For training and evaluating the validators, we extracted the proteins and their corresponding labels from the UniProt dataset (The UniProt Consortium, 2016). For the models, tokenizers, training hyperparameters, and datasets, see Supplementary Information S3.

2.4 The Judge

The congruence between the validators' predictions and a candidate description is determined by the judge. For example, the judge is expected to reject a bacterial protein (property predicted by a validator) with a description that includes activity related to the eukaryotic spliceosome (a description generated by the generator). The judge is an external LLM trained on English text, some of which relates to biological knowledge. The generator and the validators convert the protein sequences to English descriptions, and English properties, respectively, and the resulting text is used as input for the judge (Figure 1). Specifically, we used a combination of rule-based decision and prompt-based queries to GPT4 (OpenAI et al., 2024) to reject unlikely descriptions (see Supplementary Information S4).

2.5 Selecting a Subset of Diverse Descriptions

From the resulting descriptions, we aim to select a subset of representative descriptions reflecting the diversity of the suggested protein functions. Specifically, we selected three descriptions from the descriptions that passed the judge (up to 45 descriptions, 15 for each of the three prompts). To this end, we created a graph, in which nodes are descriptions,

and edges are the string-based distances between two descriptions; specifically, we used the Character n-gram F score (ChrF; Popović (2015); Supplementary Information S5). Next, we computed communities (clusters), i.e., groups of nodes (descriptions) that are more densely connected to each other than to the rest of the nodes (Blondel et al., 2008; Radicchi et al., 2004). We focused on the three largest communities and for each such community, we selected the representative description with the highest average ChrF value. The yielded descriptions are ranked by the community size. We used the Networkx library to implement the graph and the search for communities (Hagberg et al., 2008).

2.6 Evaluation

To evaluate the performance of BetaDescribe on the test set, we compared each inferred description to the true one, the latter provided by UniProt (The UniProt Consortium, 2016). We sampled dozens of proteins and conducted an extensive manual assessment of their descriptions generated by BetaDescribe and BlastP. Furthermore, four automatic metrics were used to evaluate performance: exact match, ChrF (Popović, 2015), SacreBLEU (Post, 2018) and cosine similarity (see Supplementary Information S5).

3 Results

3.1 BetaDescribe Performance

A total of 2.5 million proteins, comprising the test set, were divided into three categories according to their similarity to the training set, evaluated by BlastP E-values (Altschul et al., 1990). Category 1 sequences lack BlastP hits ($E - value > 10$, 151 proteins), Category 2 sequences had statistically insignificant hits against the training data ($1 < E - value <= 10$, 151 proteins), and Category 3 sequences had nearly significant or significant hits (E-value ¡= 1, $\sim 2.5 \times 10^6$ proteins). In our analysis, we compared BetaDescribe predictions to the best BlastP hit (with the lowest E-value) when searching the training set.

3.1.1 Providing Descriptions when BlastP is Unavailable

Within Category 1, BlastP failed to find proteins that share the same sequence due to some peptides being too short to yield any E-value (see Supplementary Information S6). Accordingly, we excluded all proteins with an identical sequence in the training and test data from all further analyses. In general, across all measures of accuracy, the differences between the first, second, and third predictions were relatively small (Table 1). For 151 Category 1 proteins, 69 had the exact same description in the training set but for a protein with a different sequence. For example, protein A0A8C0XGC0 in the test set is 50 amino-acid long and has the following description in UniProt: "Protamines substitute for histones in the chromatin of sperm during the haploid phase of spermatogenesis. They compact sperm DNA into a highly condensed, stable and inactive complex". This function was perfectly predicted by our model, probably because the function exists in the training set. However, the same functional description appears for a related protein, that differs from A0A8C0XGC0 by three indels of length one, and nine substitutions, i.e., 77% identity. The presence of low-complexity regions and repetitive elements probably failed BlastP to detect significant sequence similarity (see Supplementary Information S7).

Table 1: Performance of BetaDescribe on Category 1 proteins, i.e., test proteins without BlastP hits when searched against the training data. The number of descriptions for each column is stated in parentheses. Predictions 1,2, and 3 are the first, second, and third descriptions, respectively provided by BetaDescribe.

Metric	Prediction 1 (133)	Prediction 2 (114)	Prediction 3 (93)
Exact match (count)	4	6	2
ChrF	0.34 ± 0.2	0.33 ± 0.22	0.3 ± 0.19
SacreBLEU	0.15 ± 0.22	0.15 ± 0.24	0.12 ± 0.19
Cosine similarity	0.6 ± 0.17	0.59 ± 0.17	0.58 ± 0.15

Notably, even when the performance scores are substantially less than 1, components of the predicted descriptions may be accurate. Consider the case of protein A0A1E3Q8Q4, for which the accuracy was: 0.35, 0.081, and 0.61 for The ChrF, SacreBLEU, and cosine similarity scores, respectively. The description of this 84 amino-acid long protein in UniProt is: ‘‘**FUNCTION$** Binds tightly to hydroxyapatite. Appears to form an integral part of the mineralized matrix. Probably important to cell-matrix interaction. Promotes Arg-Gly-Asp-dependent cell attachment, **SUBCELLULAR LOCATION$** Secreted.’’

In comparison, our model prediction is: ‘‘**FUNCTION$** Plays a role in cell adhesion and tissue remodeling. May be a cell-cell adhesion protein with cell-adhesive properties, **SUBCELLULAR LOCATION$** Secreted, extracellular space, extracellular matrix, **PTM$** May be proteolytically cleaved, **SUBUNIT$** Interacts with SPRED1, **SIMILARITY$** Belongs to the invertebrate Chitin-binding protein family.’’ The predicted description correctly captures that this protein is secreted and is involved in cell-cell and cell-extracellular matrix interactions. Here too, the reason for BlastP's failure to find a similar sequence in the training set is the presence of low-complexity regions (see Supplementary Information S7), while a similar description appears in the training data. To conclude, for many of the proteins in this category, BlastP technically fails to find a hit as the proteins are short, include low-complexity regions, or both (see Supplementary Information S6).

3.1.2 Providing Descriptions when BlastP E-value is High

Category 2 includes 151 proteins with no significant hits in the training data (E-value higher than 1). For four of the proteins, BetaDescribe failed to provide a valid description (2.6%). Of the remaining 147 cases, BetaDescribe provided an exact match for two and three proteins for predictions 1, and 2, respectively. Although a Blast E-value above 1 is considered insignificant (usually the threshold is much lower, e.g., Moreno-Hagelsieb & Latimer, 2008), retrieving the description from the best hit (lowest E-value) returned five proteins with the exact match. When comparing the cosine similarity score, BetaDescribe performance (prediction 1) was superior to BlastP: 0.58 and 0.48, respectively (paired t-test; p¡0.0001; see Supplementary Information S8). The ChrF and the SacreBLEU scores were not significantly different. We additionally tested PSI-Blast (Altschul et al., 1997) and HMMER (Eddy, 2011), both of which underperformed compared to BetaDescribe (see Supplementary Information S8). We note, that E-values may not be reliable for short sequences, and thus, we tested the performance of BetaDescribe and BlastP on additional stricter criteria (Supplementary Information S8). While there is no clear value of a cosine score above which a prediction is considered reliable, from our experience, prediction scores above 0.6 were relatively accurate. With this cutoff, out of the 147 predictions, Prediction 1 of BetaDescribe was accurate in 55 predictions and BlastP was accurate in 27 cases (Supplementary Information S9). In addition, Supplementary Information S9 presents an example that highlights the advantages of having multiple candidate descriptions.

3.1.3 Providing Descriptions when BlastP Hits are Significant

We sampled 1,000 proteins from the test set, for which similar proteins are present in the training set, i.e., E-values lower (or equal) to 1.0 when running BlastP against the training set. Taking the description from the closest hit as the predicted function, yielded accurate predictions (Supplementary Information S8). Across all metrics, BlastP-based descriptions were significantly better than the ones provided by BetaDescribe (paired t-test; $p < 0.0001$), suggesting that functionality transfer based on BlastP between closely related proteins is highly accurate. However, we found a strong correlation between the accuracy of the BlastP prediction if it “agrees” with BetaDescribe prediction (Supplementary Information S10), suggesting that BetaDescribe could be applied to boost confidence for BlastP predictions. Furthermore, we evaluated whether public LLMs: GPT4 (OpenAI et al., 2024), Gemini (Gemini Team et al.) and Claude (https://www.anthropic.com/news/claude-3-5-sonnet) could generate accuarte descriptions. The descriptions produced by GPT-4 and Gemini did not differ significantly from random scores (see Supplementary Information S13).

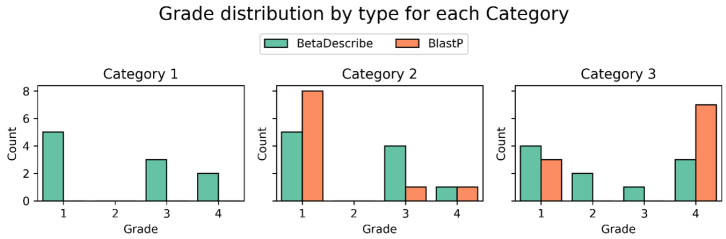


Figure 2: Histogram reporting the manual evaluation results. The subplots correspond to the different categories. Each bar represents the count of items assigned a specific grade (1 - 4), grouped by the description type (BetaDescribe or BlastP), in which a grade of 4 represents highly accurate descriptions and a grade of 1, highly inaccurate ones.

3.1.4 Analyzing the Source of Errors

In some cases, the judge rejects a description generated by the generator, based on input from the validators (Figure 1). Such cases can result from three scenarios: either the generator's description is false, and the validators correctly disagree with it, or the generator's description is correct, but the validator's predictions are wrong, which is the source of the disagreement. Another option is that both the validator and the generated description are correct, but the judge erroneously determines that they are incompatible. Supplementary Information S11 provides detailed ablation tests, including validators and judge performance, the effect of pretraining on the generator, performance of accepted and rejected descriptions and performance of descriptions part of the top predictions and not.

3.1.5 Manual Evaluation

We manually evaluated a subset of the predicted descriptions to test the agreement between cosine similarity scores and expert assessments. We randomly sampled ten proteins from each of the three categories and compared the predictions obtained using either BlastP or BetaDescribe. As BlastP fails to produce predictions for Category 1, the evaluation set included 50 descriptions in total. Each expert assigned a grade to each of these 50 descriptions. Grades were between 1 and 4, where 4 indicates an excellent match, and 1 an inadequate match (see Supplementary Information S5).

As shown in Figure 2, the distributions differ notably between methods and categories. In Category 1, BetaDescribe shows a reasonable spread of scores, with half of the predictions receiving a grade of 3 or 4, suggesting moderate quality even in low-similarity settings. In Category 2, BetaDescribe continues to show a similar distribution of five predictions receiving a score of 3 or 4, while BlastP's predictions cluster heavily at grade 1. By contrast, in Category 3, where sequence similarity is high, BlastP performs better, as most of its predictions receive a perfect match, while BetaDescribe shows a more balanced distribution of scores, including some high-scoring predictions. These distributions reinforce the notion that BetaDescribe is less dependent on sequence similarity than BlastP, maintaining stable performance even when similarity is low. In addition, the results indicate a highly significant positive correlation ($p < 0.0001$; $R^2 = 0.827$) between our primary automatic metric, cosine similarity, and the grades assigned by the experts (see Supplementary Information S12).

3.2 Providing Descriptions for Proteins with Unknown Functions

We demonstrate the usage of BetaDescribe by analyzing six examples of proteins with no experimentally proven functionality, but whose function can be predicted by other attributes, e.g., location in the genome and structural features. Specifically, we selected five proteins from three different viruses, encompassing two proteins with putative envelope glycoproteins and three RNA-depended RNA polymerase subunits. We also analyzed a newly discovered bacterial protein involved in the immune system. Supplementary Information S14 provides a detailed discussion of three likely-to-be RNA-dependent RNA polymerase

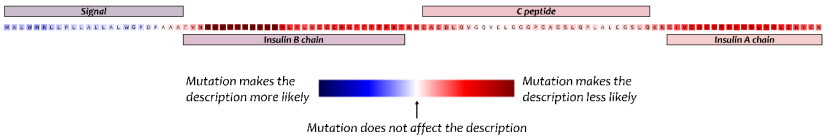


Figure 3: Identifying functionally important regions for the preproinsulin protein. The four regions of the insulin are marked: Signal, Insulin B chain, C peptide, and Insulin A chain.

subunits, SnRV-Env protein, CRISPR related protein and the descriptions generated by BetaDescribe and those retrieved via BlastP.

3.2.1 TGV-S (UniProt ID: UPI0027A96E0A)

The first example consists of the putative spike (S) protein of the recently identified nidovirus, the Trout Granulomatous Virus (TGV). The inferred function of this protein is based on the genomic localization of its open reading frame and the protein's domain structure, which are typical for nidoviruses (Karniely et al., 2023). BlastP search with TGV-S sequence against the training set yielded a significant hit (Q28042; $E - value < 10^{-5}$). This known protein has an unrelated function in fertilization. In contrast, BetaDescribe provided two valid predictions, describing a membrane virion protein (see Supplmenetary Infomration S14).

3.3 Association of the Prediction with Functionally Important Protein Regions

Functionally important protein regions can be identified through mutagenesis experiments (Hutchison et al., 1978). We next hypothesized that BetaDescribe relies on functionally important regions, as part of its prediction. This suggests that *in-silico* alanine-scanning mutagenesis experiments will affect descriptions within functionally important regions substantially more than in the remaining protein regions. Specifically, we quantified the fit (negative log-likelihood) of the description of the wild-type protein to that of the altered protein sequence, expecting that disturbing functionally important regions would substitutionally reduce this fit (see Supplementary Information S15).

We report preliminary results for this approach on the extensively studied human insulin protein (P01308; an additional example, the protein RecA, is provided in Supplementary Information S15). The preproinsulin precursor is comprised of four domains: Signal peptide, Insulin B chain, C peptide, and Insulin A chain. The Signal peptide and the C peptide (which are less evolutionary conserved) are excised during insulin protein maturation, leaving insulin B and A chains (which are highly evolutionary conserved) to form functional insulin (Steiner et al., 2009). Figure 3 reports the importance of each of the amino acids in the insulin sequence. As expected, BetaDescribe descriptions were mostly affected by mutating amino acids within the Insulin A and B chains, and substantially less by mutations in the Signal peptide and C peptide. This analysis suggests that the BetaDescribe model learned to capture biological meaningful domains. Furthermore, we have included a rigorous *in-silico* mutagenesis analysis across 165 well-characterized proteins from ProteinGym (Notin et al., 2023). This analysis statistically evaluates the association of BetaDescribe's predicted important regions within annotated functional domains, providing a quantitative measure of biological relevance (see Supplementary Information S15).

4 Discussion

BetaDescribe harnesses generative capabilities of LLMs to provide rich and accurate textual descriptions for any protein of interest. To this end, the generator was trained on rich datasets of millions of biological pairs of sequences and their descriptions. In this work, we provide evidence that BetaDescribe is mostly useful in cases where BlastP fails (Category 1) or yields insignificant hits (Category 2). BetaDescribe provides up to three alternative possible predictions for each protein. In addition, as exemplified by Category 3 proteins,

when the predictions of BetaDescribe and BlastP are congruent, the confidence in each of the provided predictions increases. Notably, disagreements among BetaDescribe alternative predictions suggest that each prediction is uncertain. Alternative descriptions could be viewed as hypotheses that need to be experimentally tested.

Explaining model predictions is common in the NLP domain (Atanasova et al., 2020). As exemplified by the preproinsulin and the RecA analysis (Supplementary Information S15), explanation techniques can be utilized in order to gain a better understanding of the functional importance of different protein regions. In our analysis, we applied a simple sliding window alanine scanning approach for this task. Future work could investigate exposing interacting regions via *in-silico* complex mutation. We envision a system that not only predicts protein functions but also highlights functionally important regions, providing context-specific insights into their roles.

5 Acknowledgments

T.P. was supported by the Tel Aviv University Center for AI and Data Science (TAD).

References

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. Prot2text: Multimodal protein's function generation with GNNs and transformers. *AAAI*, 38(10):10757–10765, 2024. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v38i10.28948. URL http://arxiv.org/abs/2307.14367.

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature*, 630(8016):493–500, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07487-w. URL https://www.nature.com/articles/s41586-024-07487-w. Publisher: Nature Publishing Group.

Reem Abu Rass, Talia Kustin, Rachel Zamostiano, Nechama Smorodinsky, Daniella Ben Meir, Daniel Feder, Nischay Mishra, W. Ian Lipkin, Avi Eldar, Marcelo Ehrlich, Adi Stern, and Eran Bacharach. Inferring protein function in an emerging virus: Detection of the nucleoprotein in tilapia lake virus. *J Virol*, 96(6):e0175721, 2022. ISSN 1098-5514. doi: 10.1128/JVI.01757-21.

David H. Ackley, Geoffrey E. Hinton, and Terrence J. Sejnowski. A learning algorithm for boltzmann machines. *Cognitive Science*, 9(1):147–169, 1985. ISSN 0364-0213. doi: 10.1016/S0364-0213(85)80012-4. URL https://www.sciencedirect.com/science/article/pii/S0364021385800124.

M. Mar Albà, Roman A. Laskowski, and John M. Hancock. Detecting cryptically simple protein sequences using the SIMPLE algorithm. *Bioinformatics*, 18(5):672–678, 2002. ISSN 1367-4803. doi: 10.1093/bioinformatics/18.5.672.

Ethan C. Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M. Church. Unified rational protein engineering with sequence-based deep representation learning. *Nat Methods*, 16(12):1315–1322, 2019. ISSN 1548-7105. doi: 10.1038/s41592-019-0598-1. URL https://www.nature.com/articles/s41592-019-0598-1. Publisher: Nature Publishing Group.

S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J Mol Biol*, 215(3):403–410, 1990. ISSN 0022-2836. doi: 10.1016/S0022-2836(05)80360-2.

Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402, 1997. ISSN 0305-1048. doi: 10.1093/nar/25.17.3389. URL https://doi.org/10.1093/nar/25.17.3389.

Benoit Arragain, Martin Pelosse, Albert Thompson, and Stephen Cusack. Structural and functional analysis of the minimal orthomyxovirus-like polymerase of tilapia lake virus from the highly diverged amnoonviridae family. *Nat Commun*, 14(1):8145, 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-44044-x.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. A diagnostic study of explainability techniques for text classification. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 3256–3274. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-main.263. URL https://aclanthology.org/2020.emnlp-main.263/.

Eran Bacharach, Nischay Mishra, Thomas Briese, Michael C. Zody, Japhette Esther Kembou Tsofack, Rachel Zamostiano, Asaf Berkowitz, James Ng, Adam Nitido, André Corvelo, Nora C. Toussaint, Sandra Cathrine Abel Nielsen, Mady Hornig, Jorge Del Pozo, Toby Bloom, Hugh Ferguson, Avi Eldar, and W. Ian Lipkin. Characterization of a novel orthomyxo-like virus causing mass die-offs of tilapia. *mBio*, 7(2):e00431–00416, 2016. ISSN 2150-7511. doi: 10.1128/mBio.00431-16.

Greta Bigelyte, Brigita Duchovska, Rimante Zedaveinyte, Giedrius Sasnauskas, Tomas Sinkunas, Indre Dalgediene, Giedre Tamulaitiene, Arunas Silanskas, Darius Kazlauskas, Lukas Valančauskas, Julene Madariaga-Marcos, Ralf Seidel, Virginijus Siksnys, and Tautvydas Karvelis. Innate programmable DNA binding by CRISPR-cas12m effectors enable efficient base editing. *Nucleic Acids Res*, 52(6):3234–3248, 2024. ISSN 1362-4962. doi: 10.1093/nar/gkae016.

Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *J. Stat. Mech.*, 2008(10):P10008, 2008. ISSN 1742-5468. doi: 10.1088/1742-5468/2008/10/P10008. URL https://dx.doi.org/10.1088/1742-5468/2008/10/P10008.

Yue Cao and Yang Shen. TALE: Transformer-based protein function annotation with joint sequence–label embedding. *Bioinformatics*, 37(18):2825–2833, 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab198. URL https://doi.org/10.1093/bioinformatics/btab198.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast and memory-efficient exact attention with IO-awareness, 2022. URL http://arxiv.org/abs/2205.14135.

Edo Dotan, Gal Jaschek, Tal Pupko, and Yonatan Belinkov. Effect of tokenization on transformers for biological sequences. *Bioinformatics*, 40(4):btae196, 2024. ISSN 1367-4811. doi: 10.1093/bioinformatics/btae196.

Sean R. Eddy. Accelerated profile HMM searches. *PLoS Comput Biol*, 7(10):e1002195, 2011. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002195.

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: Toward understanding the language of life through self-supervised learning. *IEEE Trans Pattern Anal Mach Intell*, 44(10):7112–7127, 2022. ISSN 1939-3539. doi: 10.1109/TPAMI.2021.3095381.

Marina Eyngor, Rachel Zamostiano, Japhette Esther Kembou Tsofack, Asaf Berkowitz, Hillel Bercovier, Simon Tinman, Menachem Lev, Avshalom Hurvitz, Marco Galeotti, Eran Bacharach, and Avi Eldar. Identification of a novel RNA virus lethal to tilapia. *J Clin Microbiol*, 52(12):4137–4146, 2014. ISSN 1098-660X. doi: 10.1128/JCM.00827-14.

Vladimir Gligorijević, P. Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C. Taylor, Ian M. Fisk, Hera Vlamakis, Ramnik J. Xavier, Rob Knight, Kyunghyun Cho, and Richard Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nat Biotechnol*, 12(1): 3168, 2021. ISSN 2041-1723. doi: 10.1038/s41467-021-23303-9. URL https://www.nature.com/articles/s41467-021-23303-9. Publisher: Nature Publishing Group.

Jianting Gong, Lili Jiang, Yongbing Chen, Yixiang Zhang, Xue Li, Zhiqiang Ma, Zhiguo Fu, Fei He, Pingping Sun, Zilin Ren, and Mingyao Tian. THPLM: a sequence-based deep learning framework for protein stability changes prediction upon point variations using pretrained protein language model. *Bioinformatics*, 39(11):btad646, 2023. ISSN 1367-4811. doi: 10.1093/bioinformatics/btad646. URL https://doi.org/10.1093/bioinformatics/btad646.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. Don't stop pretraining: Adapt language models to domains and tasks, 2020. URL http://arxiv.org/abs/2004.10964.

Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using NetworkX. pp. 11–15, 2008. doi: 10.25080/TCWV9851. URL https://doi.curvenote.com/10.25080/TCWV9851.

Patrick Haluptzok, Matthew Bowers, and Adam Tauman Kalai. Language models can teach themselves to program better, 2023. URL http://arxiv.org/abs/2207.14502.

Paul M. Harrison. fLPS: Fast discovery of compositional biases for the protein universe. *BMC Bioinformatics*, 18(1):476, 2017. ISSN 1471-2105. doi: 10.1186/s12859-017-1906-3.

D. Hart, G. N. Frerichs, A. Rambaut, and D. E. Onions. Complete nucleotide sequence and transcriptional analysis of snakehead fish retrovirus. *J Virol*, 70(6):3606–3616, 1996. ISSN 0022-538X. doi: 10.1128/JVI.70.6.3606-3616.1996.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J. Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q. Tran, Jonathan Deaton, Marius Wiggert, Rohil Badkundri, Irhum Shafkat, Jun Gong, Alexander Derry, Raul S. Molina, Neil Thomas, Yousuf A. Khan, Chetan Mishra, Carolyn Kim, Liam J. Bartie, Matthew Nemeth, Patrick D. Hsu, Tom Sercu, Salvatore Candido, and Alexander Rives. Simulating 500 million years of evolution with a language model. *Science*, 387(6736):850–858, 2025. doi: 10.1126/science.ads0018. URL https://www.science.org/doi/10.1126/science.ads0018. Publisher: American Association for the Advancement of Science.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL http://arxiv.org/abs/1503.02531.

Jie Hou, Badri Adhikari, and Jianlin Cheng. DeepSF: deep convolutional neural network for mapping protein sequences to folds, 2017. URL http://arxiv.org/abs/1706.01010.

C. A. Hutchison, S. Phillips, M. H. Edgell, S. Gillam, P. Jahnke, and M. Smith. Mutagenesis at a specific position in a DNA sequence. *J Biol Chem*, 253(18):6551–6560, 1978. ISSN 0021-9258.

Y. Ishino, H. Shinagawa, K. Makino, M. Amemura, and A. Nakata. Nucleotide sequence of the iap gene, responsible for alkaline phosphatase isozyme conversion in escherichia coli, and identification of the gene product. *J Bacteriol*, 169(12):5429–5433, 1987. ISSN 0021-9193. doi: 10.1128/jb.169.12.5429-5433.1987.

Mona Dverdal Jansen, Ha Thanh Dong, and Chadag Vishnumurthy Mohan. Tilapia lake virus: a threat to the global tilapia industry? *Reviews in Aquaculture*, 11(3):725–739, 2009. ISSN 1753-5131. doi: 10.1111/raq.12254. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/raq.12254. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/raq.12254.

Patryk Jarnot, Joanna Ziemska-Legiecka, Laszlo Dobson, Matthew Merski, Pablo Mier, Miguel A Andrade-Navarro, John M Hancock, Zsuzsanna Dosztányi, Lisanna Paladin, Marco Necci, Damiano Piovesan, Silvio C E Tosatto, Vasilis J Promponas, Marcin Grynberg, and Aleksandra Gruca. PlaToLoCo: the first web meta-server for visualization and annotation of low complexity regions in proteins. *Nucleic Acids Research*, 48:W77–W84, 2020. ISSN 0305-1048. doi: 10.1093/nar/gkaa339. URL https://doi.org/10.1093/nar/gkaa339.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023a. URL http://arxiv.org/abs/2310.06825.

Yuexu Jiang, Duolin Wang, Yifu Yao, Holger Eubel, Patrick Künzler, Ian Max Møller, and Dong Xu. MULocDeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation. *Computational and Structural Biotechnology Journal*, 19:4825–4839, 2021. ISSN 2001-0370. doi: 10.1016/j.csbj.2021.08.027. URL https://www.sciencedirect.com/science/article/pii/S2001037021003585.

Yuexu Jiang, Lei Jiang, Chopparapu Sai Akhil, Duolin Wang, Ziyang Zhang, Weinan Zhang, and Dong Xu. MULocDeep web service for protein localization prediction and visualization at subcellular and suborganellar levels. *Nucleic Acids Research*, 51:W343–W349, 2023b. ISSN 0305-1048. doi: 10.1093/nar/gkad374. URL https://doi.org/10.1093/nar/gkad374.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, Alex Bridgland, Clemens Meyer, Simon A. A. Kohl, Andrew J. Ballard, Andrew Cowie, Bernardino Romera-Paredes, Stanislav Nikolov, Rishub Jain, Jonas Adler, Trevor Back, Stig Petersen, David Reiman, Ellen Clancy, Michal Zielinski, Martin Steinegger, Michalina Pacholska, Tamas Berghammer, Sebastian Bodenstein, David Silver, Oriol Vinyals, Andrew W. Senior, Koray Kavukcuoglu, Pushmeet Kohli, and Demis Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. A study of BFLOAT16 for deep learning training, 2019. URL http://arxiv.org/abs/1905.12322.

Sharon Karniely, Adi Faigenboim, Salsabeel Watted, Katia Lapin, Eduard Berenshtein, Avshalom Hurvitz, Arieli Bouznach, Ezra Rozenblut, Massimo Orioles, Marco Galeotti, Irene Salinas, Asaf Berkowitz, Eran Bacharach, and Avi Eldar. Discovery of an unrecognized nidovirus associated with granulomatous hepatitis in rainbow trout. *iScience*, 26(4):106370, 2023. ISSN 2589-0042. doi: 10.1016/j.isci.2023.106370.

Kazutaka Katoh and Daron M. Standley. MAFFT multiple sequence alignment software version 7: Improvements in performance and usability. *Molecular Biology and Evolution*, 30(4):772–780, 2013. ISSN 0737-4038. doi: 10.1093/molbev/mst010. URL https://doi.org/10.1093/molbev/mst010.

Maxat Kulmanov, Mohammed Asif Khan, Robert Hoehndorf, and Jonathan Wren. DeepGO: predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics*, 34(4):660–668, 2018. ISSN 1367-4811. doi: 10.1093/bioinformatics/btx624.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, 2015. ISSN 1476-4687. doi: 10.1038/nature14539.

Chien-Der Lee and Ting-Fang Wang. The n-terminal domain of escherichia coli RecA have multiple functions in promoting homologous recombination. *Journal of Biomedical Science*, 16(1):37, 2009. ISSN 1423-0127. doi: 10.1186/1423-0127-16-37. URL https://doi.org/10.1186/1423-0127-16-37.

Simon A. Lee, Anthony Wu, and Jeffrey N. Chiang. Clinical ModernBERT: An efficient and long context encoder for biomedical text, 2025. URL http://arxiv.org/abs/2504.03964.

Wellington C. Leite, Renato F. Penteado, Fernando Gomes, Jorge Iulek, Rafael M. Etto, Sérgio C. Saab, Maria B. R. Steffens, and Carolina W. Galvão. MAW point mutation impairs h. seropedicae RecA ATP hydrolysis and DNA repair without inducing large conformational changes in its structure. *PLOS ONE*, 14(4):e0214601, 2019. ISSN 1932-6203. doi: 10.1371/journal.pone.0214601. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0214601. Publisher: Public Library of Science.

Yu Li, Zeling Xu, Wenkai Han, Huiluo Cao, Ramzan Umarov, Aixin Yan, Ming Fan, Huan Chen, Carlos M. Duarte, Lihua Li, Pak-Leung Ho, and Xin Gao. HMD-ARG: hierarchical multi-task deep learning for annotating antibiotic resistance genes. *Microbiome*, 9(1):40, 2021. ISSN 2049-2618. doi: 10.1186/s40168-021-01002-3. URL https://doi.org/10.1186/s40168-021-01002-3.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan Dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637): 1123–1130, 2023. ISSN 1095-9203. doi: 10.1126/science.ade2574.

Maria Littmann, Michael Heinzinger, Christian Dallago, Tobias Olenyi, and Burkhard Rost. Embeddings from deep learning transfer GO annotations beyond homology. *Sci Rep*, 11(1):1160, 2021. ISSN 2045-2322. doi: 10.1038/s41598-020-80786-0. URL https://www.nature.com/articles/s41598-020-80786-0. Publisher: Nature Publishing Group.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2018. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Dharia A. McGrew and Kendall L. Knight. Molecular design and functional organization of the RecA protein. *Crit Rev Biochem Mol Biol*, 38(5):385–432, 2003. ISSN 1040-9238. doi: 10.1080/10409230390242489.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. Mixed precision training. 2018. URL https://openreview.net/forum?id=r1gs9JgRZ.

Pablo Mier, Lisanna Paladin, Stella Tamana, Sophia Petrosian, Borbála Hajdu-Soltész, Annika Urbanek, Aleksandra Gruca, Dariusz Plewczynski, Marcin Grynberg, Pau Bernadó, Zoltán Gáspári, Christos A. Ouzounis, Vasilis J. Promponas, Andrey V. Kajava, John M. Hancock, Silvio C. E. Tosatto, Zsuzsanna Dosztanyi, and Miguel A. Andrade-Navarro. Disentangling the complexity of low complexity proteins. *Brief Bioinform*, 21(2):458–472, 2020. ISSN 1477-4054. doi: 10.1093/bib/bbz007.

Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan. Speech recognition using deep neural networks: A systematic review. *IEEE Access*, 7: 19143–19165, 2019. ISSN 2169-3536. doi: 10.1109/ACCESS.2019.2896880. URL https://ieeexplore.ieee.org/document/8632885/.

Erik Nijkamp, Jeffrey A. Ruffolo, Eli N. Weinstein, Nikhil Naik, and Ali Madani. ProGen2: Exploring the boundaries of protein language models. *Cell Systems*, 14(11):968–978.e3, 2023. ISSN 2405-4712. doi: 10.1016/j.cels.2023.10.002. URL https://www.sciencedirect.com/science/article/pii/S2405471223002727.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. ProteinGym: Large-scale benchmarks for protein fitness prediction and design. *Advances in Neural Information Processing Systems*, 36:64331–64379, 2023. URL https://papers.nips.cc/paper_files/paper/2023/hash/cac723e5ff29f65e3fcbb0739ae91bee-Abstract-Datasets_and_Benchmarks.html.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, C. J. Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech

Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. GPT-4 technical report, 2024. URL http://arxiv.org/abs/2303.08774.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Pierre Isabelle, Eugene Charniak, and Dekang Lin (eds.), *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pp. 311–318. Association for Computational Linguistics, 2002. doi: 10.3115/1073083.1073135. URL https://aclanthology.org/P02-1040/.

Maja Popović. chrF: character n-gram f-score for automatic MT evaluation. In Ondřej Bojar, Rajan Chatterjee, Christian Federmann, Barry Haddow, Chris Hokamp, Matthias Huck, Varvara Logacheva, and Pavel Pecina (eds.), *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pp. 392–395. Association for Computational Linguistics, 2015. doi: 10.18653/v1/W15-3049. URL https://aclanthology.org/W15-3049/.

Matt Post. A call for clarity in reporting BLEU scores, 2018. URL http://arxiv.org/abs/1804.08771.

V. J. Promponas, A. J. Enright, S. Tsoka, D. P. Kreil, C. Leroy, S. Hamodrakas, C. Sander, and C. A. Ouzounis. CAST: an iterative algorithm for the complexity analysis of sequence tracts. complexity analysis of sequence tracts. *Bioinformatics*, 16(10):915–922, 2000. ISSN 1367-4803. doi: 10.1093/bioinformatics/16.10.915.

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

Filippo Radicchi, Claudio Castellano, Federico Cecconi, Vittorio Loreto, and Domenico Parisi. Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663, 2004. doi: 10.1073/pnas.0400054101. URL https://www.pnas.org/doi/10.1073/pnas.0400054101. Publisher: Proceedings of the National Academy of Sciences.

Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. A systematic survey of prompt engineering in large language models: Techniques and applications, 2025. URL http://arxiv.org/abs/2402.07927.

Theo Sanderson, Maxwell L Bileschi, David Belanger, and Lucy J Colwell. ProteInfer, deep neural networks for protein functional inference. *eLife*, 12:e80942, 2023. ISSN 2050-084X. doi: 10.7554/eLife.80942. URL https://doi.org/10.7554/eLife.80942. Publisher: eLife Sciences Publications, Ltd.

F. Sanger and E. O. P. Thompson. The amino-acid sequence in the glycyl chain of insulin. 2. the investigation of peptides from enzymic hydrolysates. *Biochem J*, 53(3):366–374, 1953. ISSN 0264-6021. doi: 10.1042/bj0530366. URL https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1198158/.

Elana Simon, Kyle Swanson, and James Zou. Language models for biological research: a primer. *Nat Methods*, 21(8):1422–1429, 2024. ISSN 1548-7105. doi: 10.1038/s41592-024-02354-y.

D. F. Steiner, S.-Y. Park, J. Støy, L. H. Philipson, and G. I. Bell. A brief perspective on insulin production. *Diabetes Obes Metab*, 11 Suppl 4:189–196, 2009. ISSN 1463-1326. doi: 10.1111/j.1463-1326.2009.01106.x.

Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. UDSMProt: universal deep sequence models for protein classification. *Bioinformatics*, 36(8):2401–2409, 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa003. URL https://doi.org/10.1093/bioinformatics/btaa003.

Ahmet Sureyya Rifaioglu, Tunca Doğan, Maria Jesus Martin, Rengul Cetin-Atalay, and Volkan Atalay. DEEPred: Automated protein function prediction with multi-task feed-forward deep neural networks. *Sci Rep*, 9(1):7344, 2019. ISSN 2045-2322. doi: 10.1038/

s41598-019-43708-3. URL https://www.nature.com/articles/s41598-019-43708-3. Publisher: Nature Publishing Group.

Baris E. Suzek, Yuqi Wang, Hongzhan Huang, Peter B. McGarvey, Cathy H. Wu, and UniProt Consortium. UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015. ISSN 1367-4811. doi: 10.1093/bioinformatics/btu739.

Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning, 2018. URL http://arxiv.org/abs/1808.01974.

The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Research*, 45:D158–D169, 2016. ISSN 0305-1048. doi: 10.1093/nar/gkw1099. URL https://doi.org/10.1093/nar/gkw1099. _eprint: https://academic.oup.com/nar/article-pdf/45/D1/D158/23819877/gkw1099.pdf.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL http://arxiv.org/abs/2307.09288.

Daniel Veltri, Uday Kamath, and Amarda Shehu. Deep learning improves antimicrobial peptide recognition. *Bioinformatics*, 34(16):2740–2747, 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty179. URL https://doi.org/10.1093/bioinformatics/bty179.

Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep learning for computer vision: A brief review. *Computational Intelligence and Neuroscience*, 2018(1):7068349, 2018. ISSN 1687-5273. doi: 10.1155/2018/7068349. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/7068349. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2018/7068349.

Naama Wagner, Michael Alburquerque, Noa Ecker, Edo Dotan, Ben Zerah, Michelle Mendonca Pena, Neha Potnis, and Tal Pupko. Natural language processing approach to model the secretion signal of type III effectors. *Front. Plant Sci.*, 13, 2022. ISSN 1664-462X. doi: 10.3389/fpls.2022.1024405. URL https://www.frontiersin.org/journals/plant-science/articles/10.3389/fpls.2022.1024405/full. Publisher: Frontiers.

Zichen Wang, Steven A. Combs, Ryan Brand, Miguel Romero Calvo, Panpan Xu, George Price, Nataliya Golovach, Emmanuel O. Salawu, Colby J. Wise, Sri Priya Ponnapalli, and Peter M. Clark. LM-GVP: an extensible sequence and structure informed deep learning framework for protein property prediction. *Sci Rep*, 12(1):6832, 2022. ISSN 2045-2322. doi: 10.1038/s41598-022-10775-y. URL https://www.nature.com/articles/s41598-022-10775-y. Publisher: Nature Publishing Group.

Tanner Wiegand, Florian T. Hoffmann, Matt W. G. Walker, Stephen Tang, Egill Richard, Hoang C. Le, Chance Meers, and Samuel H. Sternberg. TnpB homologues exapted from transposons are RNA-guided transcription factors. *Nature*, 631(8020):439–448, 2024. ISSN 1476-4687. doi: 10.1038/s41586-024-07598-4. URL https://www.nature.com/articles/s41586-024-07598-4. Publisher: Nature Publishing Group.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer,

Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. Transformers: State-of-the-art natural language processing. In Qun Liu and David Schlangen (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.emnlp-demos.6. URL https://aclanthology.org/2020.emnlp-demos.6/.

John C. Wootton and Scott Federhen. Statistics of local complexity in amino acid sequences and sequence databases. *Computers & Chemistry*, 17(2):149–163, 1993. ISSN 0097-8485. doi: 10.1016/0097-8485(93)85006-X. URL https://www.sciencedirect.com/science/article/pii/009784859385006X.

In-Kwon Yeo and Richard A. Johnson. A new family of power transformations to improve normality or symmetry. *Biometrika*, 87(4):954–959, 2000. ISSN 0006-3444. URL https://www.jstor.org/stable/2673623. Publisher: [Oxford University Press, Biometrika Trust].

Tom Young, Devamanyu Hazarika, Soujanya Poria, and Erik Cambria. Recent trends in deep learning based natural language processing, 2018. URL http://arxiv.org/abs/1708.02709.

Qianmu Yuan, Chong Tian, Yidong Song, Peihua Ou, Mingming Zhu, Huiying Zhao, and Yuedong Yang. GPSFun: geometry-aware protein sequence function predictions with language models. *Nucleic Acids Research*, 52:W248–W255, 2024. ISSN 0305-1048. doi: 10.1093/nar/gkae381. URL https://doi.org/10.1093/nar/gkae381.

Chengxin Zhang, Quancheng Liu, and Lydia Freddolino. StarFunc: fusing template-based and deep learning approaches for accurate protein function prediction, 2024. URL https://www.biorxiv.org/content/10.1101/2024.05.15.594113v1. Pages: 2024.05.15.594113 Section: New Results.

Le Zhuo, Zewen Chi, Minghao Xu, Heyan Huang, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. ProtLLM: An interleaved protein-language LLM with protein-as-word pre-training, 2024. URL http://arxiv.org/abs/2403.07920.

A Appendix

A.1 Supplementary Information S1: Generator – Training Hyperparameters, and Training Datasets

A.1.1 Training Outline

Starting from the original Llama2 model (Touvron et al., 2023), pretrained on general English text only (from multiple public open sources), we trained the generator in three consecutive stages. We first added protein sequences to the training data. These training data (Dataset 1, $\sim 2.9 \times 10^{10}$ tokens) contained a mixture of English sentences from the RedPajama (https://www.together.ai/blog/redpajama) dataset (30%) and a large set of diverse proteins (70%), randomly sampled from the Uniref-90 dataset (Suzek et al., 2015). Training on both English and protein sequences together allowed the model to incorporate biological knowledge without losing the ability to generate English text.

While Dataset 1 contained both English and protein sequences, the English text was not directly connected to the protein sequences. In Dataset 2, we incorporated protein sequences and their cognate textual description (in English), thus adding expert vocabulary and knowledge regarding protein functionality. Specifically, this dataset contained protein sequences followed by their English descriptions (45%), English descriptions followed by their cognate proteins (45%), and English sentences not related to biology (10%). In total, this dataset contained $\sim 1.3 \times 10^{10}$ tokens, in which the biological proteins and their cognate-rich textual description were derived from the UniProt dataset (The UniProt Consortium, 2016).

For example, protein "A0A0V0V610" with the following protein sequence, "MGREDKT-TWKSNYFLKLV[...]", is trained to predict the following: ''**FUNCTION$** Ribosomal protein P0 is the functional equivalent of E.coli protein L10, **SIMILARITY$** Belongs to the universal ribosomal protein uL10 family[...]'', and vice versa, i.e., the model is trained to predict the protein sequence given the rich description.

Finally, we trained the model to predict the protein description given the protein sequences (Dataset 3). This dataset contained $\sim 8.3 \times 10^{10}$ tokens. Between the different stages, we used transfer learning, i.e., the optimal model trained in the previous stage was the starting point for the training of the next stage (Tan et al., 2018). We tested two pretraining models and different values for the learning rate for the processing of biological data.

A.1.2 Preliminary Stage

As a starting point for training the generator, we used the LAMMA2 model developed by Meta (Touvron et al., 2023). This model was trained on 2 trillion English tokens. The model was downloaded from Hugginface (Wolf et al., 2020). The training dataset is English only, from multiple public open sources.

A.1.3 Stage 1

We downloaded the Uniref-90, which contains about 175 million protein sequences (Suzek et al., 2015), and the RedPajama dataset, which includes general English text (https://www.together.ai/blog/redpajama). From these datasets, we established the training data which includes 70% protein sequences ($\sim 4,900,000$ proteins) and 30% English text ($\sim 2,100,000$ sentences). The model architecture is the same as the LLAMA2 model, with the initial weights equal to the weights provided by the LLAMA2 model. We did some experiments to find the best learning rate and initial models (see below) and chose a learning rate of 3×10^{-5} with the LLAMA2 model (Touvron et al., 2023).

Regarding the specific hyperparameters, we kept the same batch size ($1,024$) and the maximum sequence length ($4,096$) as was done in the training of the LLAMA2 model. Due to memory limitations, we applied the following techniques to reduce the memory footprint: numbers are kept in a Brain Floating Point (bfloat16) format (Kalamkar et al., 2019), gradient accumulation of 128 (with a micro-batch size of 1), and flash-attention 2 (Dao et al., 2022). We used a constant scheduler, i.e., a fixed learning rate for the entire training, and the AdamW optimizer (Loshchilov & Hutter, 2018), with betas of 0.9 and 0.999. The training was done with the Accelerator library (Wolf et al., 2020), on a single node with eight cores of NVIDIA A100-SXM4-80GB.

A.1.4 Choosing Architecture and Learning Rate for Stage 1

For optimal results, we started by finding the best model and learning rates for the biological data. We did experiments to evaluate the optimal learning rates as well as the initial model. We considered two models of similar sizes (about 7 billion parameters): Mistral (Jiang et al., 2023a) and LLAMA2 (Touvron et al., 2023). Each of these models was trained for 500 training steps with four learning rates: 3×10^{-5}, 1×10^{-5}, 1×10^{-6}, and 1×10^{-7}. The models were evaluated on three different validation datasets: English (RedPajama), protein sequences (Uniref-90), and protein sequences and their descriptions (from UniProt). Table 2 reports the validation loss of the models trained with different learning rates. As expected, the loss of the biological data, and the protein sequences with their corresponding descriptions, starts high and decreases. However, the loss on the English dataset starts low and slightly increases. We anticipated this behavior as the model was trained for many steps on a similar English dataset. The performance of the LLAMA2 with a learning rate of 3×10^{-5} had the lowest loss on the biological data, and thus, we decided to continue training with this specific learning rate.

Table 2: We report the validation perplexity of LLAMA2 (a) and Mistral (b). For each model, we tested four different learning rates for 500 steps (corresponding to 2.1 billion tokens). Validation was conducted on three datasets: (1) English sentences (RedPajama); (2) protein sequences (Uniref-90); and (3) proteins and their corresponding descriptions (from UniProt). Training data contain 70% of proteins and 30% of English (Dataset 1).

(a)

Learning rate	Validation set	0	100	200	300	400	500
3×10^{-5}	English	1.7686	1.7788	1.7831	1.786	1.7883	1.7895
3×10^{-5}	Proteins	4.6591	4.3469	4.3245	4.3101	4.2963	4.2835
3×10^{-5}	Descriptions	2.827	2.8223	2.8074	2.8214	2.8005	2.7961
1×10^{-5}	English	1.7686	1.7618	1.7608	1.7605	1.7598	1.7597
1×10^{-5}	Proteins	4.6591	4.3465	4.3306	4.3216	4.3149	4.3026
1×10^{-5}	Descriptions	2.827	2.7885	2.789	2.7886	2.7834	2.7805
1×10^{-6}	English	1.7686	1.7665	1.7621	1.7612	1.76	1.7588
1×10^{-6}	Proteins	4.6591	4.3805	4.3643	4.3529	4.3451	4.3409
1×10^{-6}	Descriptions	2.827	2.7858	2.7799	2.7783	2.7762	2.7743
1×10^{-7}	English	1.7686	1.7682	1.7682	1.7709	1.7706	1.7688
1×10^{-7}	Proteins	4.6591	4.5394	4.4688	4.4228	4.4017	4.3937
1×10^{-7}	Descriptions	2.827	2.8155	2.8074	2.8051	2.7975	2.7936

(b)

Learning rate	Validation set	0	100	200	300	400	500
3×10^{-5}	English	1.8559	2.0395	2.1036	2.1301	2.1304	2.1513
3×10^{-5}	Proteins	4.8126	4.5226	4.4713	4.4426	4.418	4.3975
3×10^{-5}	Descriptions	2.8405	3.0433	3.0032	3.0125	2.9999	2.9918
1×10^{-5}	English	1.8559	1.8659	1.878	1.8895	1.8936	1.8983
1×10^{-5}	Proteins	4.8126	4.4834	4.4507	4.4299	4.4115	4.3928
1×10^{-5}	Descriptions	2.8405	2.848	2.8296	2.8273	2.828	2.814
1×10^{-6}	English	1.8559	1.821	1.8189	1.817	1.8153	1.8143
1×10^{-6}	Proteins	4.8126	4.4982	4.4744	4.4671	4.4573	4.4448
1×10^{-6}	Descriptions	2.8405	2.7822	2.7812	2.7833	2.7793	2.7769
1×10^{-7}	English	1.8559	1.8354	1.8298	1.8268	1.8247	1.8232
1×10^{-7}	Proteins	4.8126	4.5595	4.5403	4.5277	4.5167	4.5072
1×10^{-7}	Descriptions	2.8405	2.7994	2.7902	2.7863	2.7839	2.7829

A.1.5 Stage 2

We trained the model on a mixture of the UniProt (The UniProt Consortium, 2016) and the RedPajama datasets. Unlike the training in Stage 1, which contained sequences and English text, here the UniProt dataset contains both protein sequences and their English descriptions (32,409,736 pairs of proteins and English descriptions). We converted the UniProt data from a Json format to a text format, extracting only specific fields, specifically: function, catalytic activity, pathway, subcellular localization, domains, cofactors, PTMs, subunits, assignment to protein families, activity regulations, keywords, and features. An example of the resulting text for protein entry A0A6I7XUQ0 is:

protein sequence: MTRIILPGKTIGIIGGGQLGRMMALAAKEMGYKIAVLDPTKHSPCAQVADI-
EIVASYDDLKAIQHLAEISDVVTYEFENIDYRCLQWLEKHAYLPQGSQLLS-
KTQNRFTEKNAIENAGLPVATYRLVQTQEQLTEAITELSYPSVLKTTTGGY-
DGKGQVVLRSEADVDKARKLANAAECILEKWVPFEKEVSVIVIRSVSGETK-
VFPVAENIHVNNILHESIVPARITEELSQKAIAYARVLADELELVGTLAVE-
MFATADGEIYINELAPRPHNSGHYTQDACETSQFGQHIRAICNLPLGETNL-
LKPVVMVNILGEHIEGVLRQVNRLTGCYLHLYGKEEAKAQRKMGHVNILND-
NIEVALEKAKSLHIWDHQEQLLEGKR description: **FUNCTION$** Catalyzes the ATP-dependent conversion of 5-aminoimidazole ribonucleotide (AIR) and HCO(3)(-)

to N5-carboxyaminoimidazole ribonucleotide (N5-CAIR), **FUNCTION$** Catalyzes the ATP-dependent conversion of 5-aminoimidazole ribonucleotide (AIR) and HCO(3)- to N5-carboxyaminoimidazole ribonucleotide (N5-CAIR), **CATALYTIC ACTIVITY$** 5-amino-1-(5-phospho-beta-D-ribosyl)imidazole + ATP + hydrogencarbonate = 5-carboxyamino-1-(5-phospho-D-ribosyl)imidazole + ADP + 2 H(+) + phosphate, **PATHWAY$** Purine metabolism; IMP biosynthesis via de novo pathway; 5-amino-1-(5-phospho-D-ribosyl)imidazole-4-carboxylate from 5-amino-1-(5-phospho-D-ribosyl)imidazole (N5-CAIR route): step 1/2, **SUBUNIT$** Homodimer, **SIMILARITY$** Belongs to the PurK/PurT family. keywords: ATP-binding & Ligand, Ligase & Molecular function, Nucleotide-binding & Ligand, Purine biosynthesis & Biological process. features: Domain* 1, Binding site* 7.

We removed all proteins that do not have a specific functional property in their description. The training data at this stage were divided in a way that 45% (corresponding to $2,100,000$ samples) of the data were assigned to generate the protein amino-acid sequence given the description, 45% of the data were assigned to generate the description given the protein sequence, and 10% (corresponding to $460,000$ sentences) were assigned for English sentences. The same hyperparameters were used as described in Stage 1.

A.1.6 Stage 3

We trained the model to generate the descriptions given the amino-acid sequence from the UniProt dataset (The UniProt Consortium, 2016), containing $20,480,000$ samples of proteins and their descriptions. The optimal model from Stage 2 was the starting position for this stage. We used the same training hyperparameters and only reduced the learning rate to 1×10^{-5}.

A.2 Supplementary Information S2: Effect of Temperature and Multiple Prompts on Diversity

A.2.1 Multiple Candidate Descriptions

We used two techniques to generate multiple candidate descriptions. The first relies on changing the prompts, which are used as the input to the LLMs. For English LLMs, the prompt includes the instructions (the task, e.g., "shorten the following text"), and the context (additional information needed to perform the task, e.g., the text to be shortened). Different prompts lead to different responses (Sahoo et al., 2025). For our generator, we used three alternative prompts (see below). Note that detailed instructions are not needed in our case, as the model was trained for the specific task, which is to describe the function of the query protein.

For each prompt, we generated 15 alternative descriptions using a temperature hyperparameter (Ackley et al., 1985; Hinton et al., 2015). Generating text is done by choosing the next token until reaching the special token marking the end of the text or reaching a predefined maximum length. By default, the next token chosen is the one with the highest probability. However, the model can choose the next token based on the distribution of token probabilities. A high temperature flattens this distribution and thus, introduces variability in the generated outputs, i.e., the descriptions (see below).

A.2.2 The Temperature Hyperparameter

Decoder-only models are trained to predict the next token, i.e., given the start of a sentence, the models are trained to predict the next word. Typically, the next predicted token is the one with the highest probability. However, it is possible to sample the next token from a distribution based on token probabilities. Low temperature results in a sharper distribution of tokens, i.e., the next token is likely to be the one with the highest probability. In contrast, high temperature flattens the distribution, increases diversity, and thus produces different alternative descriptions. We used the value of 1.0 for the temperature.

Table 3: Detailed prompts used for generating the descriptions.

Prompt 1	protein sequence: MTRIILPGKTIGIIGGGQLGRMMALAAKEMGYKIAVLD-PTKHSPCAQVADIEIVASYDDLKAIQHLAEISDVVTYEFENIDYRCLQWLE-KHAYLPQGSQLLSKTQNRFTEKNAIENAGLPVATYRLVQTQEQLTEAITEL-SYPSVLKTTTGGYDGKGQVVLRSEADVDKARKLANAAECILEKWVPFEKEV-SVIVIRSVSGETKVFPVAENIHVNNILHESIVPARITEELSQKAIAYARVL-ADELELVGTLAVEMFATADGEIYINELAPRPHNSGHYTQDACETSQFGQHI-RAICNLPLGETNLLKPVVMVNILGEHIEGVLRQVNRLTGCYLHLYGKEEAK-AQRKMGHVNILNDNIEVALEKAKSLHIWDHQEQLLEGKR description:
Prompt 2	protein sequence: MTRIILPGKTIGIIGGGQLGRMMALAAKEMGYKIAVLD-PTKHSPCAQVADIEIVASYDDLKAIQHLAEISDVVTYEFENIDYRCLQWLE-KHAYLPQGSQLLSKTQNRFTEKNAIENAGLPVATYRLVQTQEQLTEAITEL-SYPSVLKTTTGGYDGKGQVVLRSEADVDKARKLANAAECILEKWVPFEKEV-SVIVIRSVSGETKVFPVAENIHVNNILHESIVPARITEELSQKAIAYARVL-ADELELVGTLAVEMFATADGEIYINELAPRPHNSGHYTQDACETSQFGQHI-RAICNLPLGETNLLKPVVMVNILGEHIEGVLRQVNRLTGCYLHLYGKEEAK-AQRKMGHVNILNDNIEVALEKAKSLHIWDHQEQLLEGKR description: **FUNCTION$**
Prompt 3	protein sequence: MTRIILPGKTIGIIGGGQLGRMMALAAKEMGYKIAVLD-PTKHSPCAQVADIEIVASYDDLKAIQHLAEISDVVTYEFENIDYRCLQWLE-KHAYLPQGSQLLSKTQNRFTEKNAIENAGLPVATYRLVQTQEQLTEAITEL-SYPSVLKTTTGGYDGKGQVVLRSEADVDKARKLANAAECILEKWVPFEKEV-SVIVIRSVSGETKVFPVAENIHVNNILHESIVPARITEELSQKAIAYARVL-ADELELVGTLAVEMFATADGEIYINELAPRPHNSGHYTQDACETSQFGQHI-RAICNLPLGETNLLKPVVMVNILGEHIEGVLRQVNRLTGCYLHLYGKEEAK-AQRKMGHVNILNDNIEVALEKAKSLHIWDHQEQLLEGKR description: **FUNCTION$**

Formally, consider t_i, l_i, $P(.)$ to be the i'th token from a vocabulary with size n, the logits of the t_i and the probability function, respectively. With the softmax approach, we select the token with the highest probability:

$$P(t_i) = \frac{e^{l_i}}{\sum_{k=1}^{n} e^{l_k}}$$

This expression is modified to account for the temperature, t, but here, we sample the next token according to the resulting probability:

$$P(t_i) = \frac{e^{l_i/t}}{\sum_{k=1}^{n} e^{l_k/t}}$$

A.2.3 Alternative Prompts

We considered three different prompts (Table 3): (1) protein sequence alone; (2) protein sequence with the ''**FUNCTION$**'' property; and (3) protein sequence with double space and the ''**FUNCTION$**'' property. For the above example, the three prompts are:

A.2.4 Combining the Temperature and the Alternative Prompts Provides Diversity in the Generated Descriptions

The goal of accounting for the temperature and alternative prompts is to introduce stochasticity in the model, thus providing several descriptions for each input protein sequence. Specifically, we selected a temperature $t = 1.0$ and generated 15 alternative descriptions for each alternative prompt (totaling 45 alternative descriptions). After removing invalid

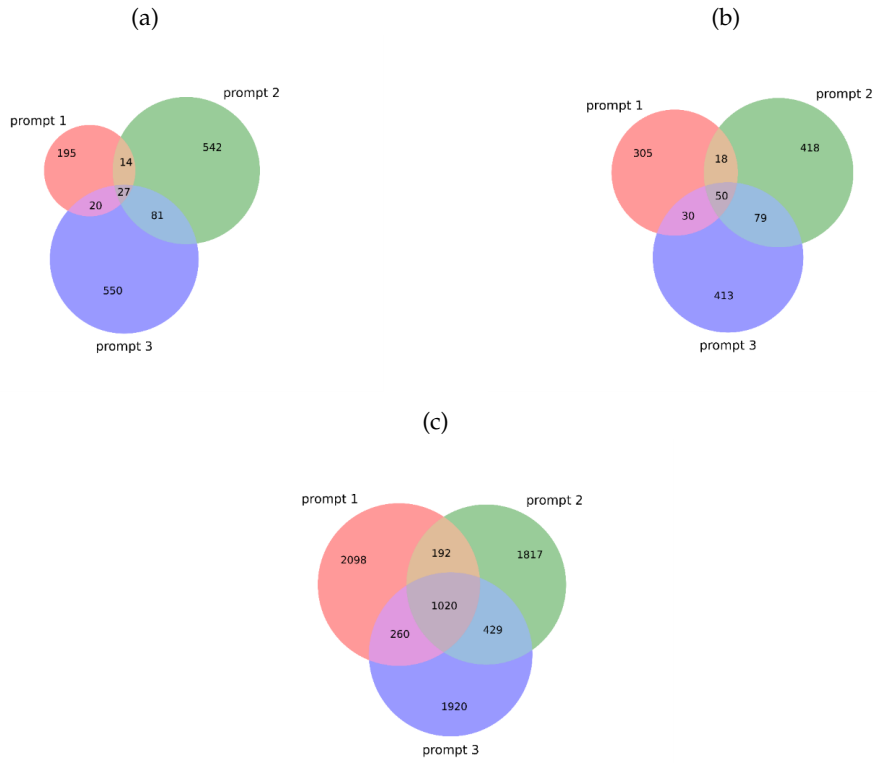


Figure 4: We report the number of unique and shared descriptions using the different prompts on Categories 1, 2 and, 3, for figures (a), (b), and (c), respectively. For example, in Category 1, we analyzed 151 proteins and obtained 6,795 descriptions (151 proteins times 45 alternative descriptions). After processing (verifying the function property), removing cases rejected by the judge and removing duplications, we received the following 1,429 descriptions. For example, of these 1,429 descriptions, 14 generated by prompts 1 and 2 were identical, but different from those generated in prompt 3.

predictions and those rejected by the judge, we plotted the shared and unique descriptions of each of the prompts. As can be seen in Figure 4, most alternative descriptions are unique, i.e., it is relatively rare that the different prompts have the same exact prediction.

A.3 Supplementary Information S3: Validators Architecture, Training, Datasets and Tokenization

We fine-tuned the ESM2 (Lin et al., 2023), an encoder-only model with 150 million parameters for predicting each of the following properties: subcellular localization, higher taxonomy, and enzyme activity. The base model has 30 hidden layers, 20 attention heads, a hidden size of 640, an intermediate size of 2,560, and a max sequence length of 1,026. The tokenizer for the model encodes each of the amino acids separately, and thus the vocabulary size is 33 (including special tokens).

We used the following training parameters for each of the validators: maximum sequence length of 1,026 (same as the pretraining), batch size of 8, learning rate of 2×10^{-5}, 2,000 steps for warmup, weight decay of 0.01, and maximum training steps of 1,200,000. We trained with a mixed-precision approach (Micikevicius et al., 2018), i.e., parts of the model are saved in a reduced format using only 16 bits compared to the traditional training that uses 32 bits. We used the Huggingface library for this training (Wolf et al., 2020), and a single core of NVIDIA RTX A6000 48GB.

A.3.1 Validators Dataset

We trained a classifier for each property, e.g., one classifier is trained to classify protein sequences into different categories of subcellular localization (multiclass). To train these validator classifiers, we extracted sequences and their corresponding properties from UniProt. Specifically, we sampled the training data for the generator and the validator from the same pool. Because these are trained in parallel, the testing data for the entire pipeline were not used for training the generator or the validators.

Subcellular localization: We extracted $19,225,898$ and $40,000$ proteins that included subcellular localization attributes in their description, for the training set and the test set, respectively.

Higher taxonomic level: Each protein in the UniProt dataset, has a taxonomy lineage attribute. The first level divides the proteins into four categories: "viruses", "bacteria", "archaea" and "eukaryota". We extracted $35,632,741$ and $40,000$ proteins with taxonomic level attributes, for the training and test datasets, respectively.

Enzyme: We extracted proteins that belong to an enzyme family, as well as proteins that have catalytic activities. We also extracted the same number of proteins, which do not show any enzymatic activity. The training data contained $51,681,180$ proteins with half tagged as enzymes and half tagged as non-enzymes. The test data included $20,000$ proteins with enzymatic/catalytic activity and $20,000$ proteins without.

A.4 Supplementary Information S4: Judge Prompts

Rejecting unlikely descriptions comprises two parts. First, we wrote a Python script to verify that those proteins that the validator determined are enzymes, have enzymatic activity in their description and vice versa: non-enzyme proteins do not include the term enzymatic activity in their description. Specifically, if either the string "catalytic activity" (this string is one of the fields in the UniProt database) or the string "belongs to" followed by "enzyme family" are in the description, we determine the description to be of an enzyme. Otherwise, we determined the description to be of non-enzyme. If the validator and the description are congruent, we accept that description and continue to the next checks by the judge. While this enzymatic activity test was rule-based, the following checks are prompt-based.

We utilize GPT4 to implement the judge. It receives as input an English description from the generator and an English text describing the results from the validators. It uses this information to reject unlikely descriptions. The input to the judge is provided as a prompt. In fact, for each description, the judge is executed up to three times. We first provide a description and the validator prediction regarding the subcellular localization. Specifically, we provide the following prompt:

You're a biology expert, and you can answer only yes or no. No explanation is needed.

The protein subcellular localization is probably in one or more of the following locations: (predicted_cell_locations)

Do you think the following function is possible? please answer yes or no only. (protein_function_prediction)

If GPT4 returns a "yes", the description will be tested against additional validator properties. Otherwise, it is rejected. Similarly to the above test, we next ask the judge about the congruence between the description and the validator's higher taxonomy level prediction:

You're a biology expert, and you can answer only yes or no. No explanation is needed.

We think that the following protein belongs to the (predicted_higher_taxonomy_level).

Do you think the following function is possible? please answer yes or no only.

(protein_function_prediction)

Lastly, if the description is valid given the previous prompt, the judge determines whether the description is possible by both properties, i.e., the higher taxonomy level and the

subcellular localization combined (we combine these two properties because the subcellular localization highly depends on the taxonomy level classification):

You're a biology expert, and you can answer only yes or no. No explanation is needed.

We think that the following protein belongs to the (predicted_higher_taxonomy_level).

In addition, the protein subcellular localization is probably in one or more of the following locations: (predicted_cell_locations)

Do you think the following function is possible? please answer yes or no only. (protein_function_prediction)

A.5 Supplementary Information S5: Evaluation Metrics

A.5.1 ChrF

Character n-gram F-score (ChrF) is an evaluation metric for strings, particularly useful in the context of text translation (Popović, 2015). It quantifies the level of substring sharing between two strings. ChrF values range from 0 (low similarity) to 1 (identical strings). ChrF captures finer nuances of the text by considering substrings (unlike word-based metrics), making it more sensitive. Formally, ChrF extracts n-grams (substrings of length n) from the reference and the hypothesis strings. Then, it calculates the precision (number of matching n-grams divided by the total number of n-grams in the hypothesis string) and the recall (number of matching n-grams divided by the total number of n-grams in the reference). Next, the F-score is computed, which in this case is called ChrF:

$$ChrF = \frac{(1+\beta^2)(Precision \times Recall)}{(\beta^2 \times Precision + Recall)}$$

We used the default values of n and β from the Huggingface library (Wolf et al., 2020): $n = 6$ and $\beta = 2$.

A.5.2 SacreBLEU

SacreBLEU (Post, 2018) is a metric for evaluating the quality of machine-translated text by comparing it to a reference translation. It is a more standardized and reproducible version of the Bilingual Evaluation Understudy (BLEU) score, which measures the overlap of n-grams (contiguous sequences of words) between two strings (Papineni et al., 2002). The metric range is between 0 to 1, with higher values indicating higher similarity. In machine translation, e.g., English to Spanish, a high-quality translation tends to have scores between 0.2 to 0.5. We compute this metric using the Huggingface library (Wolf et al., 2020).

A.5.3 Cosine Similarity Score

The cosine similarity score measures the similarity between two strings by computing the cosine of the angle between their vector representations (embeddings) in a multidimensional space. To compute this score, we embed the strings with our generator (embeddings from the last hidden layer). The cosine similarity score is then calculated as the dot product of these vectors divided by the product of their magnitudes. The range is from -1 to 1, where a value of -1 indicates strings that are diametrically opposed, 0 indicates orthogonal strings with no common terms, and 1 indicates identical strings. Cosine similarity is particularly useful in text analysis and information retrieval since the metric does not focus on the strings themselves but on their numeric representation. Thus, this metric quantifies the distance between the meanings of strings and not the characters or substrings that make up these strings.

A.5.4 Manual Evaluation

We randomly sampled 30 proteins, 10 from each of the three categories. For each protein, we gathered BetaDescribe (first option) and BlastP predictions. Since BlastP does not

Table 4: Average scores of ChrF, SacreBLEU, and cosine similarity, and the count for exact matches, for Category 1. Here, we include the 38 proteins that had identical sequences in the training set.

	Prediction 1 (170)	**Prediction 2 (126)**	**Prediction 3 (102)**
Exact match (count)	27	10	2
ChrF	0.43 ± 0.29	0.36 ± 0.25	0.3 ± 0.19
SacreBLEU	0.27 ± 0.35	0.18 ± 0.27	0.12 ± 0.19
Cosine similarity	0.66 ± 0.21	0.61 ± 0.18	0.59 ± 0.14

return results for Category 1, the final evaluation dataset consisted of 50 descriptions. The evaluation was conducted in a blind setting, i.e., the experts were unaware of the source of the prediction (whether it was BlastP or BetaDescribe). The experts' results are provided as part of the Supplementary Data 1. Experts were provided with the reference description and a link to the corresponding UniProt page. They graded the predicted description using the following scale: 1 - inadequate match, 2 - poor match, 3 - acceptable match, or 4 - perfect match. The evaluation was conducted by three senior biologists, each with over 30 years of experience. Grades were determined by majority vote. In most cases, all experts agreed on the assigned grade.

A.6 Supplementary Information S6: Categories 1, 2 and 3 Protein Data

A.6.1 Identical Sequences in Training and Test Sets

The first prediction of BetaDescribe had an exact match to the true description for 27 out of the 189 Category 1 proteins (14.3%; Table 4). These exact descriptions were surprising because we demanded that no protein be shared between the training and the test datasets. To gain further insights into such cases, we inspected specific cases. An example of such a case is protein B0M8U4 in the test set, which corresponds to the short peptide of "TDRN-FLRL". We found that a different protein in the training set, B0M3D0, shares the exact same sequence as well as the same description: "FMRFamides and FMRFamide-like peptides are neuropeptides". The reason that BlastP failed in this case is that BlastP searches for significant hits, and these peptides are too short to yield any E-values. Altogether, we found 38 cases of Category 1 proteins with identical sequences (and identical or nearly identical descriptions) for a protein in the training set and another one in the test set. For Category 2, we found 21 test proteins that have identical sequences in the train. The proteins that have an exact match in the training set, have been removed.

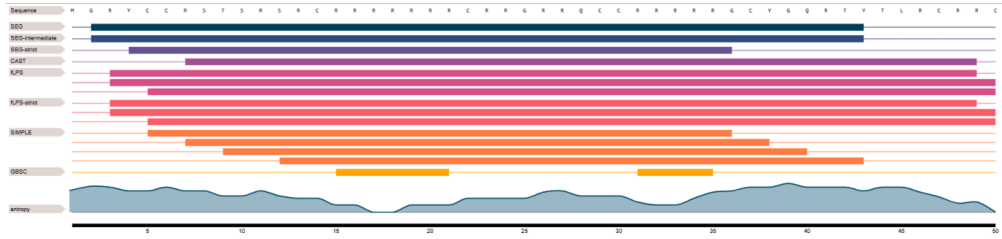
A.7 Supplementary Information S7: Low-Complexity Regions

In low-complexity regions, the variability in the number of different amino acids is reduced compared to typical protein regions. In most of these regions, there is a repeated motif, such as, a single amino acid, or a specific pattern (Mier et al., 2020). To identify such regions, we used the PlaToLoCo webserver (Jarnot et al., 2020), which computes such regions based on five programs: SEG (Wootton & Federhen, 1993), CAST (Promponas et al., 2000), fLPS (Harrison, 2017), SIMPLE (Albà et al., 2002), and GBSC (Jarnot et al., 2020). Figure 5 provides the low complexity regions computed by PlaToLoCo for proteins A0A8C0XGC0 and A0A1E3Q8Q4. As can be seen, all programs identified low-complexity regions in these proteins.

A.8 Supplementary Information S8: Detailed Performance on Categories 2 and 3

Table 5 provides detailed scores of ChrF, SacreBLEU, and cosine similarity and the number of exact matches (cases in which the output description perfectly matches the true one) for Category 2 (proteins with insignificant hits) and Category 3 (proteins with significant hits).

(a)



(b)

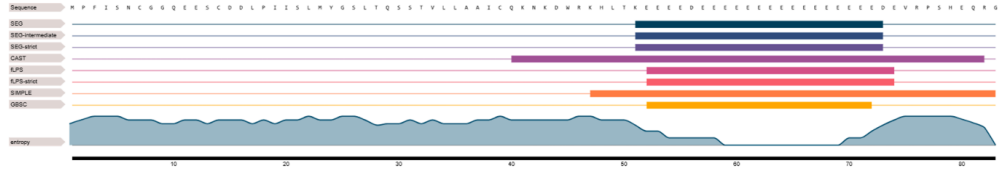


Figure 5: Low complexity regions of proteins A0A8C0XGC0 (a) and A0A1E3Q8Q4 (b) as computed by the PlaToLoCo webserver. The first row displays the amino acids, the last row reports the entropy, and other rows correspond to the program results: a thin line corresponds to a normal region, and a bold line corresponds to a low complexity region.

For Category 2 proteins, we compared BetaDescribe's top predictions (prediction 1), and the yielded by the BlastP hit. This comparison shows a higher cosine similarity score for BetaDescribe (0.58 vs. 0.48) but lower ChrF (0.27 vs. 0.28) and SacreBLEU (0.09 vs. 0.1) scores. When testing the differences using a paired t-test, only the cosine similarity score was statistically significant (p-values: cosine similarity ¡ 0.0001; ChrF = 0.62; SacreBLEU = 0.84). The hit provided by BlastP performed significantly better than BetaDescribe on all metrics when comparing the performance of Category 3 proteins. In addition, we compared our results to PSI-Blast (Altschul et al., 1997) and (Eddy, 2011) for retrieving the nearest descriptions. Since PSI-Blast and HHMER rely on initial hits from BlastP, they did not produce any results for Category 1. PSI-Blast improved upon BlastP's performance in Category 2 (cosine similarity of 0.52 vs. 0.48) but not in Category 3 (cosine similarity of 0.89 vs. 0.92). HMMER showed no improvement over BlastP in Category 2 or Category 3. Finally, we compared two additional GO predictors, DeepFRI (Gligorijević et al., 2021) and GPSFun (Yuan et al., 2024). The results show that BetaDescribe is superior to GO-based predictors (see below).

Table 5: Average scores of ChrF, SacreBLEU, and cosine similarity, and the count for exact matches, for Categories 2 (a) and 3 (b). Predictions 1, 2, and 3 are top predictions generated by BetaDescribe.

(a)

	Prediction 1 (147)	**Prediction 2 (108)**	**Prediction 3 (87)**	**BlastP (147)**	**PSI-Blast (147)**	**HMMER (140 / 147)**
Exact Match (count)	2	3	0	5	6	2
ChrF	0.27 ± 0.17	0.30 ± 0.18	0.24 ± 0.15	0.28 ± 0.19	0.3 ± 0.2	0.26 ± 0.16
SacreBLEU	0.09 ± 0.16	0.11 ± 0.19	0.07 ± 0.11	0.1 ± 0.20	0.11 ± 0.22	0.08 ± 0.15
Cosine similarity	0.58 ± 0.15	0.57 ± 0.16	0.55 ± 0.14	0.48 ± 0.18	0.52 ± 0.19	0.46 ± 0.16

(b)

	Prediction 1 (910)	**Prediction 2 (443)**	**Prediction 3 (285)**	**BlastP (910)**	**PSI-Blast (910)**	**HMMER (910)**
Exact Match (count)	383	104	12	627	560	449
ChrF	0.65 ± 0.35	0.52 ± 0.33	0.4 ± 0.26	0.85 ± 0.27	0.81 ± 0.3	0.73 ± 0.34
SacreBLEU	0.54 ± 0.44	0.37 ± 0.4	0.21 ± 0.29	0.79 ± 0.36	0.74 ± 0.38	0.64 ± 0.42
Cosine similarity	0.78 ± 0.23	0.7 ± 0.23	0.63 ± 0.18	0.92 ± 0.17	0.89 ± 0.2	0.83 ± 0.24

A.8.1 Evaluation under Stricter Sequence Similarity Filtering

Since BlastP E-values may not be reliable for short sequences, we introduced an additional filtering criterion based on sequence identity and alignment coverage. Specifically, we excluded test proteins that shared more than 30% sequence identity or more than 80% alignment coverage with any protein in the training set. This filtering was applied to Categories 2 and 3 (as Category 1 lacks BlastP hits), resulting in a subset of 30 proteins, of which 24 from Category 2 and 6 from Category 3. We then compared BetaDescribe's predictions (option 1) with those of BlastP on this filtered subset. BetaDescribe achieved significantly higher cosine similarity scores than BlastP (0.56 vs. 0.46; paired t-test, $p < 0.003$), although differences in ChrF (0.27 vs. 0.23; $p = 0.23$) and SacreBLEU (0.06 vs. 0.05; $p = 0.44$) were not statistically significant. These findings underscore BetaDescribe's ability to generate meaningful and functionally relevant descriptions, even in the absence of close sequence homologs.

A.8.2 PSI-Blast and HMMER implementation

We used PSI-Blast (version 2.14.0+; Altschul et al. (1997)), and HHMER (version 3.3.2; Eddy (2011)). We kept all parameters at their default values except for the maximum number of iterations which we set to ten in PSI-Blast configuration (the default maximum number of iterations is one). Running an HHMER search requires a profile built from a multiple-sequence alignment. To generate this profile, we first identified homologous sequences using BlastP, and then aligned them with MAFFT (Katoh & Standley, 2013). We selected HMMER hits that were not part of the initial BlastP results to ensure they were truly novel. Including the best HMMER hit, even if it was originally identified by BlastP, slightly improved performance in Category 2 (cosine similarity of: 0.48 vs 0.46; ChrF of: 0.29 vs. 0.26; SacreBLEU of: 0.09 vs 0.08).

A.8.3 Comparing BetaDescribe to GO predictors

To further contextualize BetaDescribe's performance, we conducted an approximate comparison with two recent GO-based protein function predictors: DeepFRI (Gligorijević et al., 2021), a deep-learning model that processes the protein sequence, and predicts GO annotations and GPSFun (Yuan et al., 2024), a deep-learning model that classifies the GO annotation based on the structure of the protein. Since these models output sets of GO terms rather than rich English descriptions, we converted their predictions into free-text by concatenating the predicted GO terms ordered by their scores (we used the suggested score of 0.5 as the cutoff for the GO annotations predicted by DeepFRI). We then evaluated the resulting text using the same metrics applied to BetaDescribe: cosine similarity, ChrF and SacreBLEU. To perform the comparison, we used the GPSFun webserver, and downloaded the code and model of DeepFRI.

For this comparison, we evaluated Categories 1, 2 and randomly sampled 100 proteins from Category 3. The results, summarized below, show that GO-based methods underperform relative to BetaDescribe, particularly in Categories 1 and 3. In Category 1, the cosine similarity scores for DeepFRI, GPSFun, and BetaDescribe were 0.53, 0.52, and 0.58, respectively. In Category 2, the scores were 0.61 for DeepFRI, 0.56 for GPSFun, and 0.58 for BetaDescribe. Notably, DeepFRI generated GO predictions (with scores > 0.5), for 91 out of 151 proteins in Category 1 (60.2%, and for only 23 out of 151 proteins in Category 2 (15.2%, highlighting a coverage limitation. On the 100 proteins from Category 3, DeepFRI and GPSFun achieved cosine similarities of 0.47 and 0.51, respectively, while BetaDescribe reached a significantly higher score of 0.75. Additionally, on ChrF and SacreBLEU metrics, GO-based predictions performed poorly due to their limited string-level similarity with the natural language descriptions.

A.9 Supplementary Information S9: Evaluating Category 2 Proteins

A.9.1 Evaluation Using Cosine Similarity Cutoff

Based on our empirical assessment, a threshold of 0.6 generally corresponds to meaningful and biologically relevant descriptions. Using this cutoff, we analyzed the overlap in accurate predictions across the three BetaDescribe predictions and BlastP (Figure 6). Among the 147 proteins evaluated, Prediction 1 from BetaDescribe was the most frequently accurate, with 55 descriptions surpassing the similarity threshold. Prediction 2 and Prediction 3 each contributed additional unique accurate predictions not captured by Prediction 1, suggesting that the ensemble of predictions offers complementary perspectives. In contrast, BlastP produced 27 descriptions with similarity scores above 0.6. Interestingly, nine of these BlastP descriptions were judged to be more accurate and biologically appropriate than those generated by BetaDescribe (see Supplementary Information S16), highlighting cases where sequence similarity methods retain an edge. Figure 6 reveals several notable trends: first, while there is some overlap between BetaDescribe predictions and BlastP (e.g., 5 cases shared by all methods), a substantial portion of the accurate descriptions are uniquely identified by individual predictions, especially Prediction 1 (20). This suggests that BetaDescribe is not simply recapitulating what traditional sequence alignment tools like BlastP provide, but is capturing distinct and potentially novel semantic relationships. Furthermore, the diversity of accurate predictions across the three BetaDescribe outputs underscores the value of generating multiple hypotheses. The non-overlapping accurate predictions (e.g., 20 unique to Prediction 1, 9 to Prediction 2) imply that different generations from the model can offer unique insights, possibly due to different pathways of semantic generalization. Overall, these results support the utility of BetaDescribe in producing high-quality functional descriptions that both complement and, in some cases, exceed those derived from traditional sequence-based methods.

A.9.2 Benefits of Multiple Candidate Descriptions

The following example highlights the benefits of providing several alternative descriptions. In the test set, the description of protein C6Y4A7 is:

‘‘**FUNCTION$** The EMC seems to be required for efficient folding of proteins in the endoplasmic reticulum (ER), **SUBCELLULAR LOCATION$** Endoplasmic reticulum membrane, **SUBUNIT$** Component of the ER membrane protein complex (EMC), **SIMILARITY$** Belongs to the membrane magnesium transporter (TC 1.A.67) family.’’

In this specific case, the third prediction of BetaDescribe was the most accurate (cosine similarity of 0.79):

‘‘**FUNCTION$** Part of the endoplasmic reticulum membrane protein complex (EMC) that enables the energy-independent insertion into endoplasmic reticulum membranes of newly synthesized membrane proteins. May be involved in Mg(2+) transport, **SUBCELLULAR LOCATION$** Endosome membrane, **SIMILARITY$** Belongs to the membrane magnesium transporter (TC 1.A.67) family.’’

The first and second predictions captured some aspects of the “true” protein description (ER-related function). The first (cosine score of 0.72) was:

‘‘**FUNCTION$** Involved in the early part of the secretory pathway, **SUBCELLULAR LOCATION$** Golgi apparatus membrane, **SIMILARITY$** Belongs to the KISH family.’’,

and the second (cosine score of 0.78):

‘‘**FUNCTION$** Required for the assembly of the V0 complex of the vacuolar ATPase (V-ATPase) in the endoplasmic reticulum, **SUBCELLULAR LOCATION$** Endoplasmic reticulum membrane, **SIMILARITY$** Belongs to the VMA21 family.’’

The BlastP-based prediction (best hit had an E-value of 6.5) had a lower cosine score (0.45) and substantially deviated from the provided descriptions:

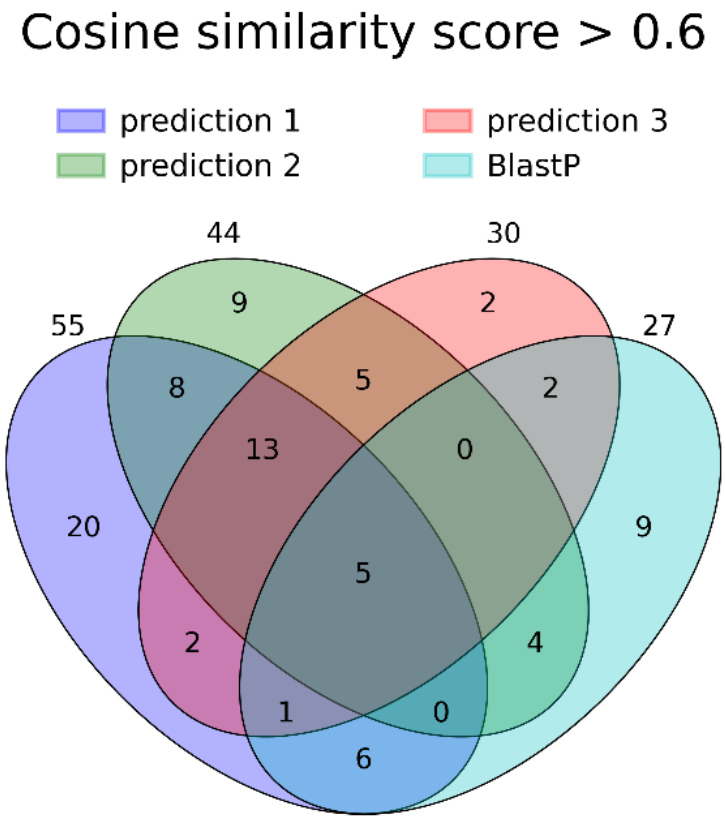


Figure 6: The Venn diagram displays the count and overlap of proteins with high cosine similarity scores (above 0.6). For example, five proteins achieved high cosine similarity scores for the three BetaDescribe predictions and the BlastP-based description. Additionally, 20, 9, 2, and 9 proteins had high cosine similarity scores exclusively in predictions 1, 2, 3, or using BlastP, respectively. After normalization, these counts correspond to 37.4%, 40.7%, 34.5%, and 18.3%, respectively.

‘‘**FUNCTION$** Catalyzes the deamination of dCTP to dUTP, **CATALYTIC ACTIVITY$** dCTP + H(+) + H2O = dUTP + NH4(+), **PATHWAY$** Pyrimidine metabolism; dUMP biosynthesis; dUMP from dCTP (dUTP route): step 1/2, **SUBUNIT$** Homotrimer,**SIMILARITY$** Belongs to the dCTP deaminase family.’’

A.10 Supplementary Information S10: BlastP Accuracy Increases if Congruent with BetaDescribe

We divided the Category 3 proteins by their E-value score into ten categories (one category of 235 proteins with a hit E-value of 0, and nine other bins with an equal number of proteins, 75 or 76). For each category, we divided the proteins into high and low values based on the cosine similarity score between the BlastP prediction and Prediction 1 of BetaDescribe. Figure 7 reports the average cosine similarity score of the BlastP hit. When the BlastP and BetaDescribe predictions are congruent, i.e., a cosine similarity score above the median for

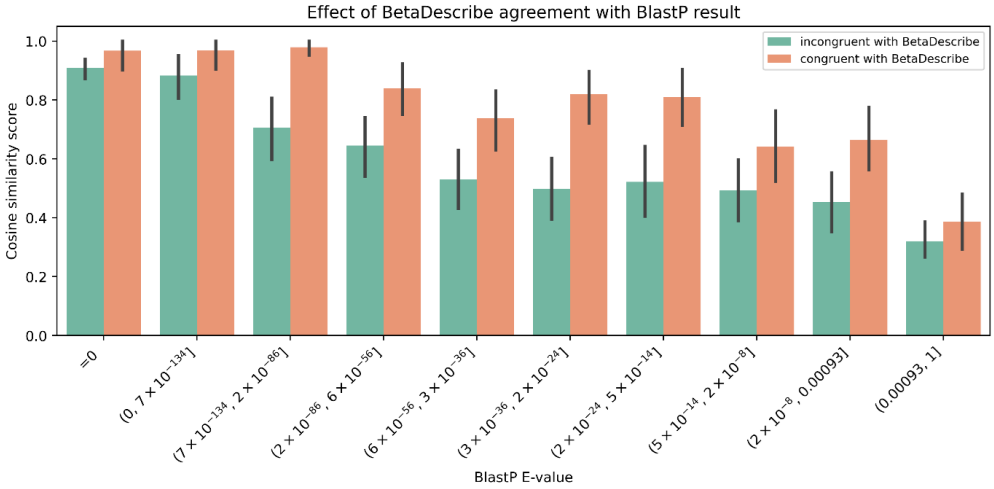


Figure 7: BlastP-based predictions are more accurate when they are congruent with BetaDescribe's predictions. BlastP scores on Category 3 proteins, as a function of their E-value score as yielded by BlastP. In each bin, we divided the BlastP predictions into two groups: those that are congruent with BetaDescribe's prediction 1 (cosine similarity score above the median) and those not.

that bin, the BlastP predictions are more accurate compared to the case where BlastP and BetaDescribe predictions are incongruent (t-test; $p < 0.001$).

A.11 Supplementary Information S11: Ablation Tests

A.11.1 Effect of Pretraining on Performance

The final training of our generator is termed Stage 3 (See main text and Supplementary Information S1). The Preliminary Stage, Stage 1, and Stage 2 are pretraining used to incorporate English and protein knowledge within our model. Here, we evaluated whether such pretraining contributed to performance. Specifically, we evaluated the performance of the model if instead of using pretraining, we initiated Stage 3 with randomly initialized weights. This model was trained on approximately 8.3 billion tokens, equivalent to $2,000$ steps. The model struggled to generate any properly structured descriptions and had a failure rate of 98.5% (no function predicted in the description). Among the 24 validation proteins for which the model successfully predicted the function, eight had the exact protein sequence in the training set. In contrast, the model that was trained on Stages 1 and 2, using the same $2,000$ steps, failed to predict only 34.3% of the test set, clearly demonstrating the importance of pretraining.

A.11.2 The Validators Performance

We evaluated the performance of the validators on a validation set comprising 40,000 proteins. Table 6 reports the precision error rate (1 minus precision) for the subcellular localization validator and the error rate for the higher taxonomy and enzymatic activity validators. The subcellular localization prediction is a multi-label classification task and thus, the accuracy is extremely high (0.99993). Hence, we report the precision error for that validator. As can be seen, the validators' error reaches a plateau, and stabilizes around 0.011, 0.0035, and 0.027 for the subcellular localization, higher taxonomy level, and enzymatic activity predictions, respectively. This indicates that enzyme classification is more challenging than the higher taxonomy level classification. These low error rates explain the importance of including the validator as part of the pipeline: the low error rates allow the judge, relying on the validator's predictions, to reliably reject unlikely generated descriptions.

Table 6: Performance of the different validators as a function of the number of training steps. For subcellular localization, we report the precision error rate (one minus precision). For the two other categories, we report error rates (one minus accuracy). Values are calculated using a validation set of 40,000 proteins.

Steps	120K	240K	360K	480K	600K	720K	840K	960K	1,080K	1,200k
Subcellular localization	0.023	0.02	0.017	0.015	0.014	0.014	0.013	0.012	0.012	0.012
Higher taxonomy level	0.0133	0.0091	0.0077	0.0063	0.0057	0.005	0.0046	0.004	0.0037	0.0035
Enzymatic activity	0.049	0.042	0.035	0.034	0.033	0.031	0.034	0.028	0.029	0.027

Table 7: Comparing the evaluation of our subcellular localization validator to MULocDeep. The validation set contains 36, 101, and 122, proteins from fungi, viridiplantae, and metazoan species, respectively.

Metric	BetaDescribe Validator	MULocDeep
F1	0.37	0.38
Precision	0.37	0.35
Recall	0.4	0.42
Accuracy	0.31	0.26

Next, we compared the performance of our subcellular localization validator to MULocDeep (Jiang et al., 2021; 2023b). While our validator can predict 388 locations, MULocDeep predicts 94 classes. Thus, we chose to use their dataset for the comparison, while manually mapping our label set to their label set. For example, if our model predicted "trans-golgi network" we used the "golgi apparatus" class (as "trans-golgi network" was not part of the validation set). We used the MULocDeep webserver to receive the predictions (Jiang et al., 2023b). Overall, the performance is similar, where MULocDeep receives slightly higher F1 and recall, and our validator receives slightly higher precision and accuracy (Table 7).

A.11.3 The Judge Performance

To evaluate the judge, we manually curated the following dataset. First, we randomly sampled 200 descriptions from the training dataset, each with a subcellular localization. Each description is matched with a pair of properties, which mimic the validator output. An example of such a pair of properties is "viruses" and "endoplasmic reticulum". We manually modified the properties of some of these pairs, so that 100 descriptions are matched with "true" properties, i.e., the properties match the description provided by UniProt. For the remaining 100 descriptions we modified at least one of the properties so that the validator and the descriptions are incongruent. We expect the judge to correctly accept the first 100 samples and reject the others. For example, consider the following description:

FUNCTION$** Inhibits post-transcriptional processing of cellular pre-mRNA, by binding and inhibiting two cellular proteins that are required for the 3'-end processing of cellular pre-mRNAs: the 30 kDa cleavage and polyadenylation specificity factor/CPSF4 and the poly(A)-binding protein 2/PABPN1. In turn, unprocessed 3' end pre-mRNAs accumulate in the host nucleus and are no longer exported to the cytoplasm. [...], **SUBUNIT$ Homodimer. Interacts with host TRIM25 (via coiled coil); this interaction specifically inhibits TRIM25 multimerization and TRIM25-mediated RIGI CARD ubiquitination. Interacts with human EIF2AK2/PKR, CPSF4, IVNS1ABP and PABPN1, **SUBCELLULAR LOCATION$** Host nucleus, **DOMAIN$** The dsRNA-binding region is required for suppression of RNA silencing, **SIMILARITY$** Belongs to the influenza A viruses NS1 family.

With the real properties of "viruses" and the subcellular localization of the "host nucleus", we except the judge to correctly accept the description. We expect the judge to reject the description with false properties, such as, a higher taxonomy level of "eukaryote" and a subcellular location of "endoplasmic reticulum". Out of the 100 descriptions with the correct properties from the training set, GPT4 and Claude (3.5 Sonnet), correctly classified 81 and

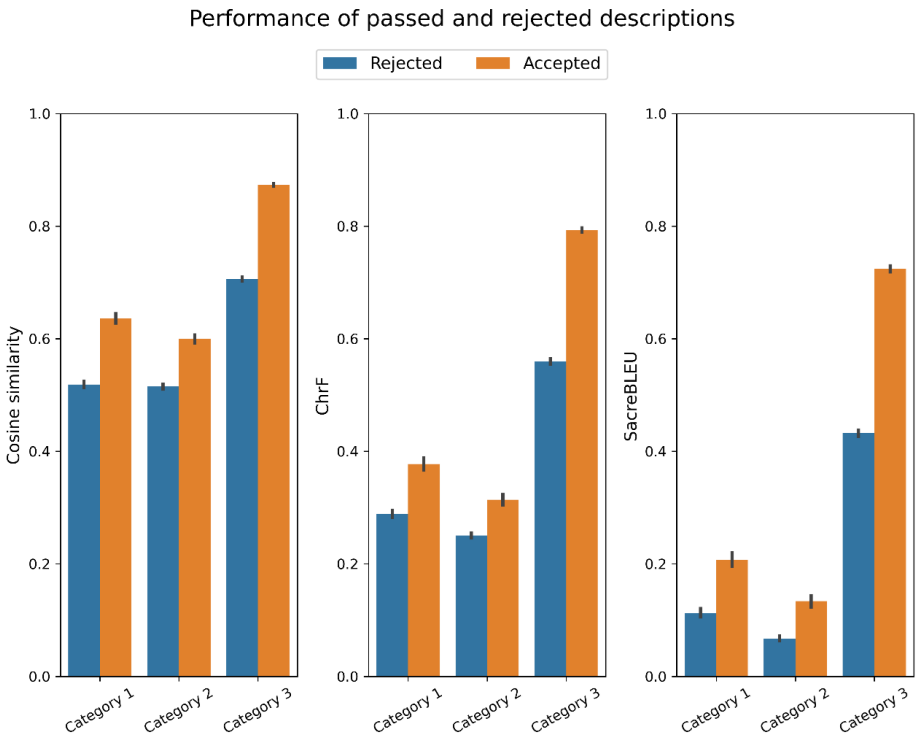


Figure 8: Evaluation of predictions accepted and rejected by the judge. We divided the descriptions generated for Categories 1, 2, and 3 based on the verdict from the judge. We then evaluated the cosine similarity, ChrF, and SacreBLEU scores compared to the true labels.

97, respectively. Out of the 100 descriptions with erroneous (altered) properties, GPT4 and Claude correctly rejected 87 and 68, respectively. Thus, the accuracy of GPT4 is 0.84, with an F1 score of 0.84, while the accuracy of Claude is 0.83 with an F1 score of 0.85.

A.11.4 Effect of Rejecting Description on Overall Performance

To evaluate the effect of the rejection step by the judge, we conducted the following analysis, which uses the predictions from the validators. We used the generated descriptions for our validation set (Categories 1, 2, and 3), and split the descriptions into two groups based on the verdict given by our judge (passed / rejected). Then, we calculated our evaluation metrics on each group independently (Figure 8). In all tested cases, the performance of the passed descriptions was significantly better compared to the rejected ones (t-test; $p < 0.00001$). This suggests that the judge, combined with the validators, effectively filters out incorrect descriptions.

A.11.5 Effect of Diverse Set of Descriptions

To evaluate the effectiveness of selecting a subset of diverse descriptions using community detection, we compared the performance of descriptions within the top three options compared to those outside. We divided the descriptions for our validation set (Categories 1, 2, and 3) into two groups: those chosen to the top three options and those outside them. We then calculated our evaluation metrics on each group independently (Figure 9). The results demonstrate that descriptions within the top three communities achieved significantly higher performance across all metrics compared to descriptions outside these communities

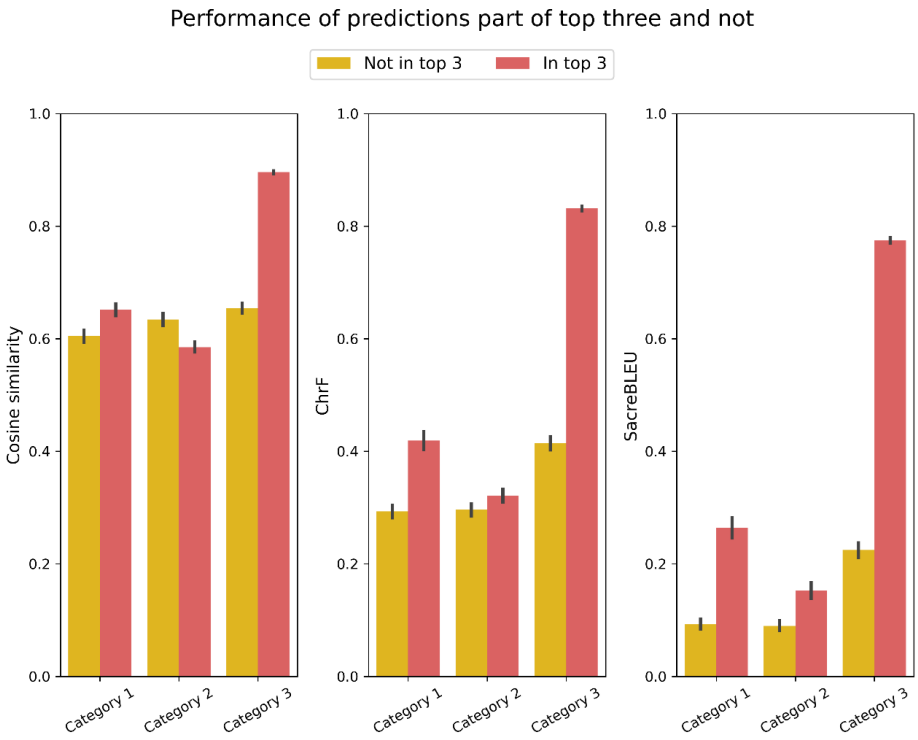


Figure 9: Evaluation of descriptions within and outside the top three options. We divided the descriptions based on their membership in the top three options. We then evaluated the cosine similarity, ChrF, and SacreBLEU scores compared to the true labels.

(t-test; $p < 0.015$) on Categories 1 and 3. This is particularly evident in Category 3, where the differences in cosine similarity, ChrF, and SacreBLEU scores are most pronounced. For Category 2, the results are mixed. The cosine similarity of descriptions outside the top-ranked options is higher than that of those within the top options, whereas the ChrF and SacreBLEU scores show the opposite trend (t-test; $p < 0.015$). These results suggest that for a small subset of proteins, our approach does not improve the provided descriptions.

A.12 Supplementary Information S12: A Strong Correlation Between the Manual Evaluation Results and the Cosine Similarity Metric

Figure 10 shows the linear regression between cosine similarity and the manual grades. The results show a highly significant correlation ($p < 0.0001$), with an R^2 of 0.827. Recognizing that linear regression can be sensitive to outliers, we repeated the analysis excluding predictions with cosine similarity above 0.95. Even under this constraint, the correlation remained significant ($p < 0.001$), with an R^2 of 0.613.

A.13 Supplementary Information S13: Public LLMs Predictions of the Function of Proteins Given a Protein Sequence as Input

A.13.1 Public LLM Predictions for Protein Functions

We next evaluated the performance of predicting protein functions using public LLMs. We tested three LLMs trained on general knowledge: GPT4 (OpenAI et al., 2024), Gemini (Gemini Team et al.), and Claude (https://www.anthropic.com/news/claude-3-5-sonnet). We asked these LLMs to predict the function of 30 test proteins from Category 3. In the prompt,

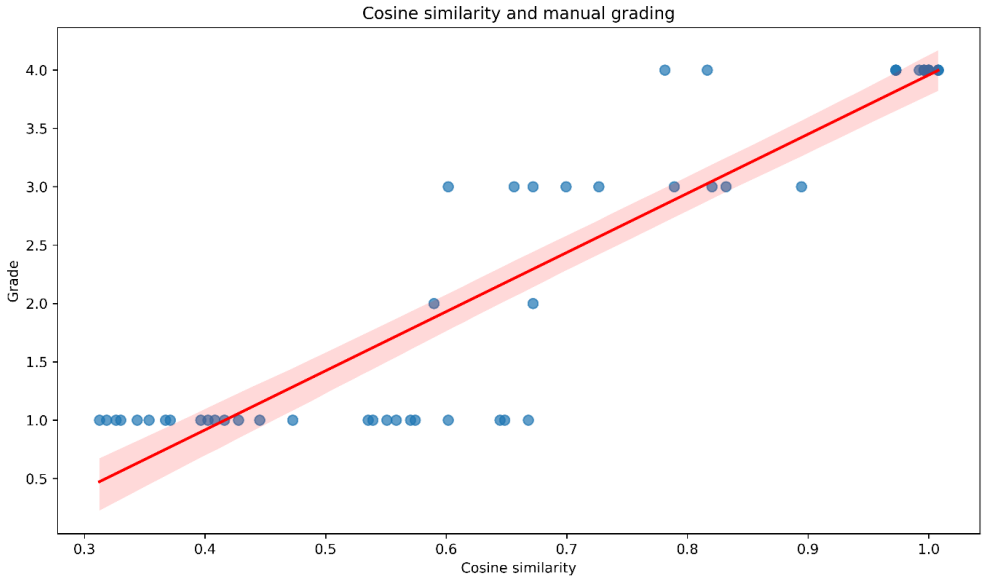


Figure 10: The linear regression between cosine similarity scores (x-axis) and manual grades (y-axis). Cosine similarity measures the distance between the embeddings of predicted and reference descriptions (ranging from -1 to 1). Manual grades were assigned by domain experts where: 1 indicates an inadequate match, 2 indicates a poor match, 3 indicates an acceptable match, and 4 indicates a perfect match.

we provided the protein sequence, as well as three examples for the required output (see below). The cosine similarity score of the prediction and the true description was comparable among the three LLMs, with Claude, GPT4, and Gemini values of 0.49, 0.45, and 0.43, respectively. GPT4 and Gemini scores were not significantly different from scores obtained when the description for a given protein was randomly sampled from the training set ($p = 0.37$, 0.89, respectively). For comparison, BetaDescribe's cosine similarity score was 0.86. For only two proteins, the public LLMs provided a cosine similarity score above 0.6: A0A0D2C110 and A0A0D6M674. The highest cosine similarity score was predicted by Claude for protein A0A0D2C110, with the "true" description: ''**FUNCTION$** Has a role in the initiation of DNA replication. Required at S-phase checkpoint, **SUBCELLULAR LOCATION$** Nucleus, **SIMILARITY$** Belongs to the SLD2 family.'' For this protein Claude provided the following description: ''**FUNCTION$** Transcriptional regulator involved in chromatin remodeling and gene expression regulation, particularly during development and cellular differentiation, **SUBCELLULAR LOCATION$** Nucleus, **SIMILARITY$** Belongs to the ARID (AT-rich interaction domain) family of DNA-binding proteins''

A.13.2 Specific Prompts used to Evaluate Public LLMs

We tested how public LLMs perform compared to BetaDescribe, i.e., how well they can describe the function of a given protein when provided its sequence as input. We used the public versions as of October 2024 of the following models: GPT4 (OpenAI et al., 2024), Gemini 1.5 Flash (Gemini Team et al., 2024), and Claude 3.5 Sonnet. The following prompt was used for this experiment:

"What is the function of the protein with the following sequence:

(amino_acid_chain)

Use the following output format.

Example 1 of the output format:

FUNCTION$** Catalyzes the reduction of fatty acyl-CoA to fatty alcohols, **CATALYTIC ACTIVITY$ a long-chain fatty acyl-CoA + 2 H(+) + 2 NADPH = a long-chain primary fatty alcohol + CoA + 2 NADP(+), **SIMILARITY$** Belongs to the fatty acyl-CoA reductase family.

Example 2 of the output format:

FUNCTION$** Nuclease required for the repair of DNA interstrand cross-links (ICL). Acts as a 5'-3' exonuclease that anchors at a cut end of DNA and cleaves DNA successively at every third nucleotide, allowing to excise an ICL from one strand through flanking incisions, **CATALYTIC ACTIVITY$ Hydrolytically removes 5'-nucleotides successively from the 3'-hydroxy termini of 3'-hydroxy-terminated oligonucleotides., **COFACTOR$** Mg(2+), **SUBCELLULAR LOCATION$** Nucleus, **SIMILARITY$** Belongs to the FAN1 family.

Example 3 of the output format:

FUNCTION$** Directs RNA polymerase II nuclear import, **SUBCELLULAR LOCATION$ Cytoplasm, **SIMILARITY$** Belongs to the IWR1/SLC7A6OS family."

Supplementary Data 5 provides the resulting prediction for each protein. We note that some of their answers contain unrelated information (such as suggestions to use bioinformatics tools). Thus, we manually processed their answers to extract only the meaningful descriptions for the evaluation metrics.

A.14 Supplementary Information S14: Descriptions Provided by BetaDescribe and BlastP Search for Unknown Proteins

A.14.1 SnRV-Env (UniProt ID: UPI000010DFE3)

The envelope (Env) protein of the snakehead retrovirus (SnRV). The genome of this fish virus was sequenced about 30 years ago (Hart et al., 1996) but no experimental data support the functionality of this protein. The SnRV-Env functionally is predicted by the genomic localization of the env gene, typical to retroviruses (downstream to the gag-pol ORF) and the presence of a predicted leader peptide and a transmembrane domain (Hart et al., 1996). The Env protein is targeted to the plasma membrane, and plays essential roles in receptor binding, membrane fusion, and viral entry into the host cells, and thus elucidating its function is important for understanding virus-cell interactions. Querying the protein sequence against the training set yielded a hit with an E-value of 0.5 to a protein, A0A0R1ZYJ7 with an unrelated function of pseudouridine synthesis. BetaDescribe provided two predictions: the first inferred functions related to viral envelope proteins and the second predicted membranal localization (Table 9).

A.14.2 Descriptions for Three TiLV Proteins (UniProt IDs: UPI0007A102A5, UPI0007A10278, UPI0007A0F427)

TiLV (Tilapia lake virus) is a negative-stranded RNA virus first identified in 2014 in northern Israel (Eyngor et al., 2014). The virus infects both wild and farmed tilapia populations. Since its discovery, TiLV has been detected in various regions across Asia, Africa, and South America, threatening the food security of millions of people (Jansen et al., 2009). By and large, when discovered, TiLV's ten main proteins showed no significant sequence similarity to other known viral proteins (Bacharach et al., 2016). Among these ten, proteins 1 – 3, were predicted to serve as subunits of TiLV polymerase (Abu Rass et al., 2022), a prediction that was recently validated experimentally and structurally (Arragain et al., 2023). These sequences and their associated functions are not included in the training or the test data. Among these three proteins, the predictions based on BlastP correctly identified only Protein 1 as polymerase (see Tables 10). In contrast, BetaDescribe correctly assigned polymerase activity for all three proteins. For Protein 1, the top prediction of BetaDescribe correctly assigned RNA-depended RNA polymerase (RdRP) activity, while for proteins 2 and 3, BetaDescribe top predictions inaccurately assigned the polymerase activity to be DNA-depended. Of note, some of BetaDescribe's alternative predictions are likely to be false, e.g., the proteasome connection in the second prediction for Protein 1 (see below). Interestingly, an endonuclease-like domain was identified in Protein 3, but

its functionality remains elusive (Arragain et al., 2023); such activity is included in an alternative BetaDescribe prediction.

A.14.3 H5TRP0

To further demonstrate the utility of BetaDescribe's ability in analyzing non-viral proteins, we provided a query for the bacterial protein H5TRP0. We selected this protein because its function was described only recently after the training and test datasets were created. This protein, termed Cas12m, operates within the Clustered Regularly Interspaced Short Palindromic Repeats (CRISPR) system (Bigelyte et al., 2024), which is an adaptive immune system in prokaryotes that targets foreign DNA or RNA sequences with high specificity (Ishino et al., 1987). Searching for the most related protein in the training set yielded an insignificant hit to an unrelated protein with serine protease activity (A0A916UE10, E-value of 1; Table 13). In contrast, BetaDescribe generated two out of three predictions that correctly linked it to CRISPR activity (Table 13). Although the third description is not directly Cas-associated, recent studies reveal that the RNA-guided DNA binding and cleavage activity of Cas12 originates from transposon-encoded nuclease TnpB. This nuclease promotes transposon survival and spread and performs similar reactions to Cas12 (Wiegand et al., 2024).

A.14.4 Predictions Generated by BetaDescribe and BlastP-based descriptions

Below are the predictions generated by BetaDescribe as well as the BlastP-based prediction for six proteins for which the function was not experimentally validated: TGV-S (Table 8), SnRV-Env (Table 9), protein 1 (Table 10), protein 2 (Table 11), protein 3 (Table 12) and H5TRP0 (Table 13).

A.15 Supplementary Information S15: Quantifying the Functionally Importance of Protein Regions

We employed *in-silico* alanine scanning mutagenesis, using a sliding-window approach, with a window size of ten amino acids and a shift of one amino acid, i.e., overlapping windows. To quantify the importance of substituting the residues in a specific sequence window to alanine, we used the negative log-likelihood metric, which provides a fit between a sequence and a description (see below). Thus, for each amino acid, there are ten values (except the ones in the edges of the protein sequence) of negative-log likelihood scores (one for each participation in a window). We averaged the negative-log likelihood for each amino acid, and normalized the scores via a power transformation (Yeo & Johnson, 2000), which averages the scores to zero. Next, we subtracted the normalized value of the wild-type protein sequence, such that the average score of the wild-type sequence becomes zero. We note that we did not train a model to predict the impact of mutagenesis experiments instead, we used an unsupervised approach that relies on the previous training of BetaDescribe generator to analyze the mutated sequences.

A.15.1 Negative Log-Likelihood

In NLP, negative log-likelihood (NLL) is a measurement of how well a model predicts text. It is commonly used to evaluate language models by determining how "surprised" the model is by specific tokens. A lower negative log-likelihood indicates the model is more confident in its predictions, assigning higher probabilities to the actual next tokens in the sequence. Models are trained to minimize this value, and thus improve language understanding and generation accuracy. For the following n tokens: $w_1, w_2, ..., w_n$, the negative log-likelihood is defined as:

$$NLL = -\frac{1}{n}\sum_{i=11}^{n}\log(p(w_i)$$

Where $p(w_i)$ is the probability to generate the token w_i given the previous tokens: $w_1...w_{i-1}$.

Table 8: Prediction for the TGV-S protein by: BetaDescribe (predictions 1 and 2), and BlastP.

BetaDescribe (Prediction 1)	"**FUNCTION$** Envelope glycoprotein that forms spikes at the surface of virion envelope. Essential for the initial attachment to heparan sulfate moities of the host cell surface proteoglycans. Involved in fusion of viral and cellular membranes leading to virus entry into the host cell. Following initial binding to its host receptors, membrane fusion is mediated by the fusion machinery composed at least of gB and the heterodimer gH/gL. May be involved in the fusion between the virion envelope and the outer nuclear membrane during virion egress, **SUBCELLULAR LOCATION$** Virion membrane, **SUBUNIT$** Homotrimer; disulfide-linked. Binds to heparan sulfate proteoglycans. Interacts with gH/gL heterodimer, **SIMILARITY$** Belongs to the herpesviridae glycoprotein B family."
BetaDescribe (Prediction 2)	"**FUNCTION$** The surface protein (SU) attaches the virus to the host cell by binding to its receptor. This interaction triggers the refolding of the transmembrane protein (TM) and is thought to activate its fusogenic potential by unmasking its fusion peptide. Fusion occurs at the host cell plasma membrane, & The transmembrane protein (TM) acts as a class I viral fusion protein. Under the current model, the protein has at least 3 conformational states: pre-fusion native state, pre-hairpin intermediate state, and post-fusion hairpin state. During viral and target cell membrane fusion, the coiled coil regions (heptad repeats) assume a trimer-of-hairpins structure, positioning the fusion peptide in close proximity to the C-terminal region of the ectodomain. The formation of this structure appears to drive apposition and subsequent fusion of viral and target cell membranes. Membranes fusion leads to delivery of the nucleocapsid into the cytoplasm, **SUBCELLULAR LOCATION$** Cell membrane. **SUBUNIT$** The mature envelope protein (Env) consists of a trimer of SU-TM heterodimers attached by noncovalent interactions or by a labile interchain disulfide bond"
BlastP (protein Q28042, $E-value < 10^{-5}$)	"**FUNCTION$** Binds to oocyte zona pellucida in vivo. May play a role in the fertilization process and/or early embryonic development, **SUBCELLULAR LOCATION$** Cytoplasmic vesicle, secretory vesicle, **SIMILARITY$** Belongs to the glycosyl hydrolase 18 family."

A.15.2 Additional Example for Identifying Functionally Important Protein Regions

RecA of *Escherichia coli* (P0A7G6), the founding member of the bacterial RecA protein family, is essential for initiating DNA break repair, activating the SOS response, enabling translation synthesis, and promoting the spread of antibiotic resistance genes. For its various roles, RecA has a few functional regions: N-terminal domain, "Make ATP Work" (MAW) motif, A site, B site, and DNA binding sites (McGrew & Knight, 2003). Previous studies showed that mutating these regions may lead to severe defects in RecA function (Lee & Wang, 2009; Leite et al., 2019; McGrew & Knight, 2003). As reported in Figure 11, BetaDescribe indicated that mutations in some of the functional sites, such as the N-terminal domain, MAW motif, and A site, increase the negative log-likelihood of the description. In other words, BetaDescribe was mainly influenced by mutations in certain functional regions, and significantly less so by mutations in non-functional regions. However, not all functional regions were identified as important. The B site as well as some binding sites were not classified as important. This analysis suggests that the BetaDescribe model can be used in an unsupervised approach to identify regions of importance.

A.15.3 Rigorous in-silico Mutagenesis Analysis

To further investigate whether BetaDescribe's description generation relies on known functional regions, we analyzed 165 well-characterized proteins from ProteinGym (Notin et al., 2023) and retrieved their annotated domains from UniProt. Of the 165 proteins, 110 had

Table 9: Prediction for the SnRV-Env protein by: BetaDescribe (predictions 1 and 2), and BlastP.

BetaDescribe (Prediction 1)	**FUNCTION$** The leader peptide is a component of released, infectious virions and is required for particle budding, & The transmembrane protein (TM) acts as a class I viral fusion protein. Under the current model, the protein has at least 3 conformational states: pre-fusion native state, pre-hairpin intermediate state, and post-fusion hairpin state. During viral and target cell membrane fusion, the coiled coil regions (heptad repeats) assume a trimer-of-hairpins structure, positioning the fusion peptide in close proximity to the C-terminal region of the ectodomain. The formation of this structure appears to drive apposition and subsequent fusion of viral and target cell membranes. Membranes fusion leads to delivery of the nucleocapsid into the cytoplasm, **SUBCELLULAR LOCATION$** Endoplasmic reticulum membrane.
BetaDescribe (Prediction 2)	**FUNCTION$** Acts as a ligand for KLRK1, **SUBCELLULAR LOCATION$** Cell membrane, **SIMILARITY$** Belongs to the NKG2D ligand family.
BlastP (protein A0A0R1ZYJ7, E-value of 0.5)	**FUNCTION$** Responsible for synthesis of pseudouridine from uracil, **CATALYTIC ACTIVITY$** a uridine in RNA = a pseudouridine in RNA, **SIMILARITY$** Belongs to the pseudouridine synthase RluA family.

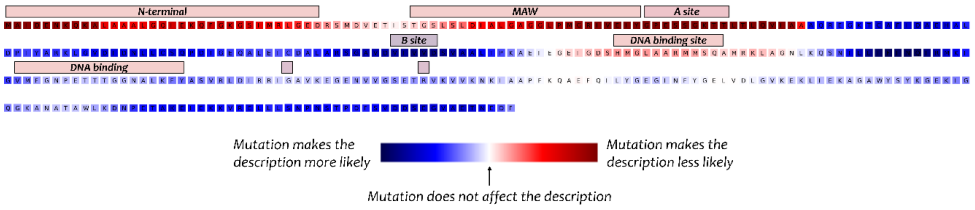


Figure 11: Functionally Important regions within the RecA protein as identified by BetaDescribe. The importance of each region is computed by a window-sliding, *in-silico* mutagenesis approach. The negative log-likelihood of each residue is evaluated to quantify functional importance. The values are normalized such that zero represents the average conservation score of the wild-type protein.

at least one annotated domain (e.g., "CN hydrolase", "PDZ1", "SH3" or "DRBM1"). For each protein, we identified the important positions by calculating if the position increases or decreases the fit of the descriptions (measured by BetaDescribe's negative-log likelihood) and examined whether the positions are located within the annotated domains. Our results indicate that mutating positions in known domains affects the fit significantly more than mutating positions outside of these domains (Chi-Square test; $p = 1.986 \times 10^{-126}$). To further examine how this effect varies with model confidence, we divided proteins into three equally-sized categories based on BetaDescribe's wild-type fit, i.e., the negative-log likelihood of the original description. If BetaDescribe is confident in a description for a specific protein, it should have a low negative-log likelihood. Significant association was observed across the low and medium categories: $p = 2.487 \times 10^{-111}$, and $p = 9.127 \times 10^{-20}$, respectively, while the high negative-log likelihood category did not show significant association ($p = 1.0$). This lack of signal at high negative-log likelihood suggests that BetaDescribe's negative-log likelihood score is a meaningful proxy for the biological relevance of its outputs. These findings indicate that BetaDescribe learned to identify functionally relevant regions despite not being explicitly trained on domain annotations or mutation data, highlighting the capacity of the model to capture biologically meaningful features from sequence-description pairs. Supplementary Information S15, also discuss the effect of tokenization on our mutagenesis analysis.

Table 10: Prediction for the protein 1 by: BetaDescribe (predictions 1, 2 and 3), and BlastP.

BetaDescribe (Prediction 1)	**FUNCTION$** Probably involved in the RNA silencing pathway and required for the generation of small interfering RNAs (siRNAs), **CATALYTIC ACTIVITY$** a ribonucleoside 5'-triphosphate + RNA(n) = diphosphate + RNA(n+1), **SIMILARITY$** Belongs to the RdRP family.
BetaDescribe (Prediction 2)	**FUNCTION$** Hydrolase that can remove 'Lys-48'-linked conjugated ubiquitin from proteins, **CATALYTIC ACTIVITY$** Thiol-dependent hydrolysis of ester, thioester, amide, peptide and isopeptide bonds formed by the C-terminal Gly of ubiquitin (a 76-residue protein attached to proteins as an intracellular targeting signal)., **SIMILARITY$** Belongs to the MINDY deubiquitinase family. FAM188 subfamily.
BetaDescribe (Prediction 3)	**FUNCTION$** The proteasome is a multicatalytic proteinase complex which is characterized by its ability to cleave peptides with Arg, Phe, Tyr, Leu, and Glu adjacent to the leaving group at neutral or slightly basic pH. The proteasome has an ATP-dependent proteolytic activity. This subunit is involved in antigen processing to generate class I binding peptides, **CATALYTIC ACTIVITY$** Cleavage of peptide bonds with very broad specificity., **SUBCELLULAR LOCATION$** Nucleus.
BlastP (protein O41353, E-value of 0.000871)	**FUNCTION$** RNA-dependent RNA polymerase which is responsible for replication and transcription of virus RNA segments. The transcription of viral mRNAs occurs by a unique mechanism called cap-snatching. 5' methylated caps of cellular mRNAs are cleaved after 10-13 nucleotides by PA. In turn, these short capped RNAs are used as primers by PB1 for transcription of viral mRNAs. During virus replication, PB1 initiates RNA synthesis and copy vRNA into complementary RNA (cRNA) which in turn serves as a template for the production of more vRNAs, **CATALYTIC ACTIVITY$** a ribonucleoside 5'-triphosphate + RNA(n) = diphosphate + RNA(n+1), **SUBUNIT$** RNA polymerase is composed of three subunits: PA, PB1 and PB2, **SIMILARITY$** Belongs to the influenza viruses polymerase PB1 family.

Table 11: Prediction for the protein 2 by: BetaDescribe (prediction 1), and BlastP.

BetaDescribe (Prediction 1)	**FUNCTION$** DNA-dependent RNA polymerase catalyzes the transcription of DNA into RNA using the four ribonucleoside triphosphates as substrates, **CATALYTIC ACTIVITY$** a ribonucleoside 5'-triphosphate + RNA(n) = diphosphate + RNA(n+1), **SIMILARITY$** Belongs to the RNA polymerase beta' chain family.
BlastP (protein A0A357NC89, E-value of 3.4)	**FUNCTION$** Synthesizes alpha-1,4-glucan chains using ADP-glucose, **CATALYTIC ACTIVITY$** [(1- 4)-alpha-D-glucosyl](n) + ADP-alpha-D-glucose = [(1- 4)-alpha-D-glucosyl](n+1) + ADP + H(+), **PATHWAY$** Glycan biosynthesis; glycogen biosynthesis, **SIMILARITY$** Belongs to the glycosyltransferase 1 family. Bacterial/plant glycogen synthase subfamily.

A.15.4 Discussing the Effect of the Tokenization on the Mutagenesis Analysis

We analyzed the tokenization of the 165 protein sequences used in our mutagenesis analysis. We found that the average token length (the number of amino acids per token) is 1.54, indicating that on average, tokens are only slightly longer than single amino acids. This suggests that although some multi-letter groupings do occur, the tokenizer generally maintains relatively fine-grained resolution over the sequences. When fine-tuning pretrained language models on a new domain, it is a standard and effective practice to reuse the original tokenizer rather than retrain it. In practice, this means that the model can learn to associate existing

Table 12: Prediction for the protein 3 by: BetaDescribe (predictions 1, 2 and 3), and BlastP.

BetaDescribe (Prediction 1)	**FUNCTION$** DNA-dependent RNA polymerase catalyzes the transcription of DNA into RNA using the four ribonucleoside triphosphates as substrates. Specific core component of RNA polymerase III which synthesizes small RNAs, such as 5S rRNA and tRNAs, **SUBCELLULAR LOCATION$** Nucleus, **SUBUNIT$** Component of the RNA polymerase III (Pol III) complex consisting of 17 subunits, **SIMILARITY$** Belongs to the eukaryotic RPC3/POLR3C RNA polymerase subunit family.
BetaDescribe (Prediction 2)	**FUNCTION$** Interacts with EME1 to form a DNA structure-specific endonuclease with substrate preference for branched DNA structures with a 5'-end at the branch nick. Typical substrates include 3'-flap structures, D-loops, replication forks and nicked Holliday junctions. May be required in mitosis for the processing of stalled or collapsed replication fork intermediates. May be required in meiosis for the repair of meiosis-specific double strand breaks subsequent to single-end invasion (SEI), **SUBCELLULAR LOCATION$** Nucleus, **COFACTOR$** Mg(2+), **SUBUNIT$** Interacts with EME1, **SIMILARITY$** Belongs to the XPF family.
BetaDescribe (Prediction 3)	**FUNCTION$** Decapping enzyme for NAD-capped RNAs: specifically hydrolyzes the nicotinamide adenine dinucleotide (NAD) cap from a subset of RNAs by removing the entire NAD moiety from the 5'-end of an NAD-capped RNA, **SUBCELLULAR LOCATION$** Nucleus, **COFACTOR$** a divalent metal cation, **SIMILARITY$** Belongs to the DXO/Dom3Z family.
BlastP (protein A0A1F6CNS5, E-value of 1.8)	**FUNCTION$** Associates with the EF-Tu.GDP complex and induces the exchange of GDP to GTP. It remains bound to the aminoacyl-tRNA.EF-Tu.GTP complex up to the GTP hydrolysis stage on the ribosome, **SUBCELLULAR LOCATION$** Cytoplasm, **SIMILARITY$** Belongs to the EF-Ts family.

tokens with new meanings (in this case, amino acid fragments) during training, without needing to completely retrain the tokenizer. Research supports the idea that models can repurpose tokens semantically even when tokenizers were originally optimized for other modalities like English text. For instance, domain-adaptation studies in biomedical and clinical settings have shown that tokenizers trained on general English corpora work well in new domains without retraining (Gururangan et al., 2020; Lee et al., 2025). Finally, although the tokenizer's subword grouping might affect granularity, the important regions identified by BetaDescribe are much longer than the average number of amino acids per token. This suggests that the model is effectively capturing meaningful sequence-level perturbations, and that tokenization is not obscuring position-level importance in a significant way.

A.16 Supplementary Information S16: An example in which the BlastP-based Description is More Accurate than that of BetaDescribe

In nine out of the 147 Category 2 proteins, the BlastP-based description had a higher cosine similarity score (above 0.6) than all predictions by BetaDescribe. Protein A0A3G3C7W0 exemplifies such a case. The description from UniProt is as follows:

''**FUNCTION$** Probable toxin that inhibits ion channels, **SUBCELLULAR LOCATION$** Secreted, **DOMAIN$** The cysteine framework is VI/VII (C-C-CC-C-C), & The presence of a 'disulfide through disulfide knot' structurally defines this protein as a knottin, **PTM$** Mostly non-hydroxylated.''

The BlastP-based description in the training set is: ''**FUNCTION$** Insecticidal toxin that reversibly and voltage-independently blocks both mid-low- (M-LVA) and high-voltage-activated (HVA) calcium channels (Cav) in cockroach DUM neurons. Also causes a modest block of insect sodium channel currents (Nav). Induces potent

Table 13: Prediction for the H5TRP0 by: BetaDescribe (predictions 1, 2 and 3), and BlastP.

BetaDescribe (Prediction 1)	**FUNCTION$** CRISPR (clustered regularly interspaced short palindromic repeat) is an adaptive immune system that provides protection against mobile genetic elements (viruses, transposable elements and conjugative plasmids). CRISPR clusters contain spacers, sequences complementary to antecedent mobile elements, and target invading nucleic acids. CRISPR clusters are transcribed and processed into CRISPR RNA (crRNA).
BetaDescribe (Prediction 2)	**FUNCTION$** CRISPR (clustered regularly interspaced short palindromic repeat) is an adaptive immune system that provides protection against mobile genetic elements (viruses, transposable elements and conjugative plasmids). CRISPR clusters contain sequences complementary to antecedent mobile elements and target invading nucleic acids. CRISPR clusters are transcribed and processed into CRISPR RNA (crRNA), **COFACTOR$** Mg(2+) or Mn(2+) required for ssDNA cleavage activity., & iron-sulfur cluster, **SIMILARITY$** Belongs to the CRISPR-associated exonuclease Cas4 family.
BetaDescribe (Prediction 3)	**FUNCTION$** Involved in the transposition of the insertion sequence IS5, **SIMILARITY$** Belongs to the transposase 11 family.
BlastP (protein A0A916UE10, $E-valueof1$)	**FUNCTION$** ATP-dependent serine protease that mediates the selective degradation of mutant and abnormal proteins as well as certain short-lived regulatory proteins. Required for cellular homeostasis and for survival from DNA damage and developmental changes induced by stress. Degrades polypeptides processively to yield small peptide fragments that are 5 to 10 amino acids long. Binds to DNA in a double-stranded, site-specific manner, **CATALYTIC ACTIVITY$** Hydrolysis of proteins in presence of ATP., **SUBUNIT$** Homohexamer. Organized in a ring with a central cavity, **SUBCELLULAR LOCATION$** Cytoplasm, **INDUCTION$** By heat shock, **SIMILARITY$** Belongs to the peptidase S16 family.

excitatory symptoms, followed by flaccid paralysis leading to death in house crickets (By similarity), **SUBCELLULAR LOCATION$** Secreted, **DOMAIN$** The presence of a 'disulfide through disulfide knot' structurally defines this protein as a knottin, **MISCELLANEOUS$** This toxin comes from a female specimen. It is observed that propeptide sequences coming from female specimen have only limited homology with the male paralogs, but the reason is unknown, **SIMILARITY$** Belongs to the neurotoxin 08 (Shiva) family. 01 (omega toxin) subfamily.''

The descriptions generated by BetaDescribe is: ''**FUNCTION$** Has antibacterial activity, **SUBCELLULAR LOCATION$** Secreted, **SIMILARITY$** Belongs to the beta-defensin family.''

''**FUNCTION$** Involved in gametogenesis and steroidogenesis, **SUBCELLULAR LOCATION$** Secreted, **SUBUNIT$** Heterodimer of an alpha and a beta chain, **SIMILARITY$** Belongs to the glycoprotein hormones subunit beta family.''

''**FUNCTION$** Has antibacterial activity,''