



BetaDescribe: Providing rich descriptions from protein sequences

Edo Dotan^{a,b}, Iris Lyubman^b, Marcelo Ehrlich^b , Eran Bacharach^b , Tal Pupko^{b,1} , and Yonatan Belinkov^{a,c,1}

Affiliations are included on p. 8.

Edited by Eugene V. Koonin, National Institutes of Health, Bethesda, MD; received December 26, 2025; accepted May 20, 2026

Deciphering protein function is fundamental to advancements in medicine and biotechnology. However, conventional experimental characterization remains resource-intensive. Public large language models (LLMs), though proficient in natural language processing, often fail to accurately interpret and predict the functional and structural properties of proteins, limiting their utility in bioinformatics. To address this gap, we introduce BetaDescribe, designed to generate detailed and rich textual descriptions of proteins, including their 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. BetaDescribe starting point was the LLAMA2 model, which was trained on trillions of tokens. Our model was next trained on datasets containing both biological and English text, which allowed the incorporation of biological knowledge. In addition to the description generator, BetaDescribe comprises multiple validator models and a judge, which together enable accurate ranking of alternative generated descriptions. 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 further 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.

Protein2Text | biological large models | protein language models | natural language processing | protein description prediction

Since the discovery of the first protein sequence (1), 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 the specific functions of proteins is crucial for advancing fields like medicine, agriculture, and biotechnology. By comprehensively understanding how proteins operate, scientists can develop targeted therapies for diseases, engineer crops with desirable traits, and design novel biomaterials. Furthermore, elucidating protein functionality is essential for uncovering the complexities of life itself, offering insights into evolution, cellular dynamics, and organismal behavior. Thus, the pursuit of deciphering protein functionality stands as a cornerstone in the quest to comprehend the intricacies of biological systems.

Unlocking the functionality of proteins poses a formidable challenge due to the intricate and multifaceted nature of these biomolecules. Researchers often embark on complex wet lab experiments, navigating through a labyrinth of variables and interactions. Designing experiments to elucidate protein functionality requires meticulous planning, innovative techniques, and sophisticated instrumentation. From studying protein–protein interactions to deciphering their three-dimensional structures, each aspect demands tailored experimental approaches. Moreover, the dynamic nature of proteins adds another layer of complexity, as their functions can be influenced by environmental factors, posttranslational modifications (PTM), and interactions with other molecules. In addition, when facing an unknown protein, typically, researchers need to design and conduct multiple experiments, each targeting a specific hypothesis. Thus, experimental determination of the functionality of a new protein may take a few 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 (2) such as computer vision (3), natural language processing (NLP; 4), speech (5), and structural biology (6). Biological sequences,

Significance

Determining protein function remains a major bottleneck in biology, particularly for proteins lacking close homologs with known annotations. This work introduces BetaDescribe, a protein-to-text modeling framework that generates rich, human-readable functional descriptions directly from protein sequences. The proposed architecture relies on a specialized multicomponent framework featuring protein-to-text generation, multiple validators, and a judge model. We show that this integrated system significantly boosts performance and ensures description reliability. By integrating biological and natural language knowledge, BetaDescribe captures functional signals beyond sequence similarity and highlights regions critical for protein activity. This approach provides a complementary strategy to traditional annotation methods, enabling deeper exploration of protein function across uncharacterized regions of the proteome universe.

Author contributions: E.D., M.E., E.B., T.P., and Y.B. designed research; E.D., I.L., E.B., T.P., and Y.B. performed research; E.D., T.P., and Y.B. contributed new reagents/analytic tools; E.D., I.L., M.E., E.B., T.P., and Y.B. analyzed data; and E.D., I.L., M.E., E.B., T.P., and Y.B. wrote the paper.

The authors declare no competing interest.

This article is a PNAS Direct Submission.

Copyright © 2026 the Author(s). Published by PNAS. This article is distributed under Creative Commons Attribution-NonCommercial-NoDerivatives License 4.0 (CC BY-NC-ND).

¹To whom correspondence may be addressed. Email: talp@tauex.tau.ac.il or belinkov@technion.ac.il.

This article contains supporting information online at <https://www.pnas.org/lookup/suppl/doi:10.1073/pnas.2537345123/-/DCSupplemental>.

Published June 29, 2026.

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 (7, 8). 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, in contrast to biological sequences, evaluating texts written in natural languages, like English, is relatively straightforward; a simple read-through by a native speaker can reveal errors, convey meaning, and provide insights into the text's quality and coherence. Nevertheless, similar properties of protein sequences and English text allow adapting NLP-based techniques for protein analyses (9–12).

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 (13) or stability (14, 15). On the classification side, deep learning models have been developed to predict protein subcellular localization (16–18), infer structural characterization (19), identify type III effectors (20), classify antibiotic resistance genes properties (21), or recognize antimicrobial activity (22). 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 (23–30).

More recently, protein language models were successfully trained to predict structures from sequences. Landmark models such as AlphaFold3 (31) and ESM3 (32) leverage large-scale pretraining on protein sequence corpora to learn complex sequence–structure relationships. 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 (11) 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 (33–36) 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 (37), trained on trillions of English tokens. While this pretraining corpus likely encompasses public scientific repositories (such as PubMed and Wikipedia), providing a broad baseline of biological information, the model initially lacks the specialized, high-resolution expertise required for precise protein functional annotation. To bridge this gap, 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. BetaDescribe acts as a bridge between protein sequences and their rich functional descriptions.

A Deep Learning Framework for Generating Protein Descriptions.

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* (Fig. 1). The generator creates rich and detailed candidate textual descriptions for each protein. The validators predict a few 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 the generation of text is much more complex than its validation). In addition, we harnessed GPT4 (38) to serve as the judge. Similar techniques to generate and validate solutions have been proposed in the context of code generation (39). As the final output, BetaDescribe provides a set of possible descriptions, ranked by their likelihood.

Results

To validate BetaDescribe, we perform a multitiered analysis focusing on accuracy, comparative utility, and biological relevance. We first benchmark BetaDescribe against sequence-based search tools across three distinct protein categories (defined by E-value), using both automated linguistic metrics and blinded expert evaluation. We then compare BetaDescribe to general-purpose LLMs. To demonstrate practical utility, we apply our framework to uncharacterized viral and bacterial proteins. Finally, we utilize *in silico* mutagenesis and domain-association analyses to investigate

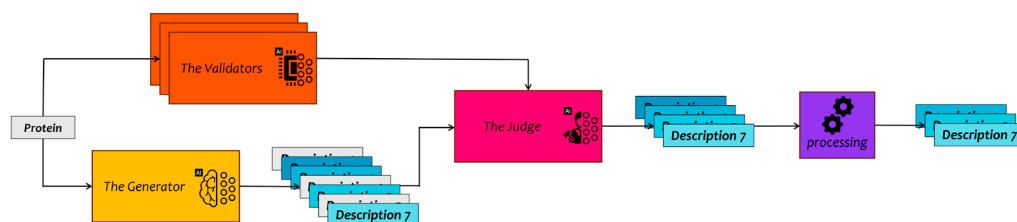


Fig. 1. BetaDescribe workflow. The generator (7 billion parameters, LLAMA2 based, fine-tuned on UniRef sequences) processes the protein sequences and creates multiple candidate descriptions. The validators (150 million parameters, ESM2 based, fine-tuned on UniProt labels), independently, provide simple textual properties of the protein. The judge (GPT4) 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.

Table 1. Performance of BetaDescribe on Category 1 proteins, i.e., test proteins without BlastP hits when searched against the training data

	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

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. Exact match checks for identical strings. ChrF and SacreBLEU score string similarity based on shared characters and words. Cosine similarity assesses semantic meaning by measuring the distance between text embeddings.

whether the model’s predictions are grounded in biologically significant protein regions.

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 (40, 41). 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, ~ 2.5 × 10⁶ proteins). In our analysis, we compared BetaDescribe predictions to the best BlastP hit (with the lowest E-value) when searching the training set.

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 (SI Appendix, Supplementary Information S1). Accordingly, we excluded all proteins with an identical sequence in the training and test data from all further analyses. To automatically evaluate the predicted descriptions, we utilized four metrics: a) exact match; b) Character n-gram F score (ChrF; 42); c) SacreBLEU (43), which is an improvement of the Bilingual Evaluation Understudy (BLEU) method used to evaluate machine translation systems; d) the cosine similarity. ChrF and SacreBLEU score string similarity based on shared characters and words. Cosine similarity assesses semantic meaning by measuring the distance between text embeddings (Materials and Methods). In general, across all measures of accuracy, the differences between the first, second, and third predictions were relatively small (Table 1). Out of the 151 proteins in Category 1, only 133 (88%) were retained after the validator and the judge filtered out invalid descriptions. The percentages of cases with two or three alternative valid descriptions were 75% and 62%, respectively.

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, A0A8C0XGC0 in the test set is a 50 amino acid-long protein with the following UniProt description: “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 (SI Appendix, Supplementary Information S2).

In the case of protein A0A1E3Q8Q4, the accuracy was 0.35, 0.081, and 0.61 for ChrF, SacreBLEU, and cosine similarity scores, respectively. The UniProt description of this 84-amino acid-long protein 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 (SI Appendix, Supplementary Information S2), 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 (SI Appendix, Supplementary Information S1).

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., 41), 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 SI Appendix, Supplementary Information S3). The ChrF and the SacreBLEU scores were not significantly different. We additionally tested PSI-Blast (44) and HMMER (45), both of which underperformed compared to BetaDescribe (SI Appendix, Supplementary Information S3). 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 (SI Appendix, Supplementary Information S3). With a 0.6 cosine similarity cutoff, out of the 147 predictions, BetaDescribe (Prediction 1) was accurate in 55 predictions and BlastP was accurate in 27 cases (SI Appendix, Supplementary Information S4).

The following case study highlights the benefits of providing several alternative descriptions. In the test set, the protein C6Y4A7 is part of the endoplasmic reticulum membrane protein complex (EMC; Table 2). In this case, the third prediction of BetaDescribe yielded the highest accuracy, while the first and second predictions captured some aspects of the “true” protein description (ER-related function). Quantitatively, the first, second, and third predictions achieved cosine similarity scores of 0.72, 0.78, and 0.79, respectively. 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.

Table 2. Case study of descriptions generated by BetaDescribe for test protein C6Y4A7

	FUNCTION\$	SUBCELLULAR LOCATION\$	SUBUNIT\$	SIMILARITY\$
UniProt Description	The EMC seems to be required for efficient folding of proteins in the endoplasmic reticulum (ER)	Endoplasmic reticulum membrane	Component of the ER membrane protein complex (EMC)	Belongs to the membrane magnesium transporter (TC 1.A.67) family.
BetaDescribe (Prediction 1)	Involved in the early part of the secretory pathway	Golgi apparatus membrane	–	Belongs to the KISH family
BetaDescribe (Prediction 2)	Required for the assembly of the V0 complex of the vacuolar ATPase (V-ATPase) in the endoplasmic reticulum	Endoplasmic reticulum membrane	–	Belongs to the VMA21 family
BetaDescribe (Prediction 3)	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	Endosome membrane	–	Belongs to the membrane magnesium transporter (TC 1.A.67) family.
BlastP Nearest	Catalyzes the deamination of dCTP to dUTP	–	Homotrimer	Belongs to the dCTP deaminase family

Providing descriptions when blastP hits are significant. Due to the substantial computational demands of the BetaDescribe pipeline, we conducted the following analysis on a representative subset of the ~2.5 million proteins in Category 3. To this end, we sampled 1,000 proteins that exhibited sequence homology to 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 (SI Appendix, Supplementary Information S3). 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 observed a strong correlation between the accuracy of BlastP predictions and cases in which BlastP and BetaDescribe produced concordant annotations, suggesting that agreement with BetaDescribe can serve as an indicator of increased confidence in BlastP predictions (SI Appendix, Supplementary Information S5).

Analyzing the source of errors. In some cases, the judge rejects a description generated by the generator, based on input from the validators (Fig. 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 descriptions are correct, but the judge erroneously

determines that they are incompatible. SI Appendix, Supplementary Information S6 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 that were part of the top predictions versus those that were not. These results show that, for example, the judge erroneously accepts incompatible descriptions (when the validator and the generator disagree) and rejects compatible ones in 19% of the cases.

Manual evaluation. We manually evaluated a subset of the predicted descriptions to test the agreement between cosine similarity scores and expert assessments. Specifically, 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, 3 an acceptable match, 2 a poor match and 1 an inadequate match (SI Appendix, Supplementary Information S7). Notably, the manual evaluation was conducted in a blind manner; the experts were unaware of whether a prediction originated from BlastP or BetaDescribe.

As shown in Fig. 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

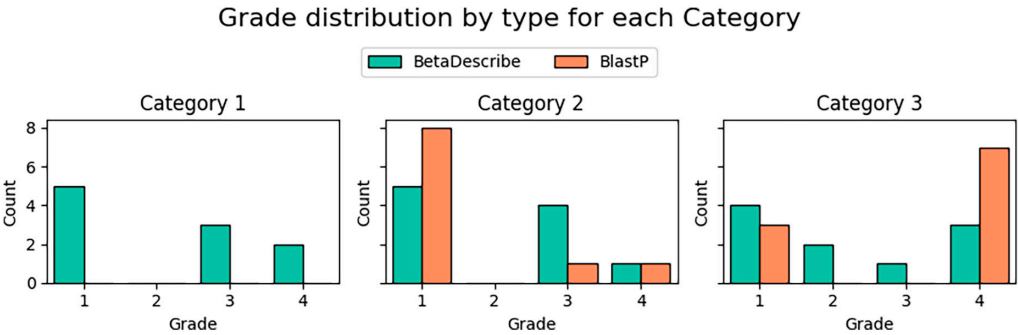


Fig. 2. Histogram reporting the manual evaluation results. The subplots correspond to the different categories. Each bar represents the count of items assigned to a specific grade (1–4), grouped by the description type (BetaDescribe or BlastP), in which a grade of 4 represents highly accurate descriptions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

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, indicating that most of its outputs were judged as inadequate in this mid-similarity setting. By contrast, in Category 3, where sequence similarity is high, BlastP performs considerably 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 reliant on sequence similarity than BlastP. While BlastP excels in high-similarity cases (Category 3), its performance drops sharply when sequence similarity is lower. In contrast, BetaDescribe maintains a more stable performance across categories, albeit with greater variance, reflecting its generative and potentially more exploratory nature. 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 (*SI Appendix, Supplementary Information S8*).

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 (38), Gemini (46), 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, we provided the protein sequence, as well as three examples for the required output (*SI Appendix, Supplementary Information S9*). 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.

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-dependent RNA polymerase subunits. We also analyzed a recently identified bacterial protein involved in the immune system.

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 (ORF) and the protein's domain structure, which are typical for nidoviruses (47). 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 (Table 3).

SnRV-Env (UniProt ID: UPI000010DFE3). The second example is the envelope (Env) protein of the snakehead retrovirus (SnRV). The genome of this fish virus was sequenced about 30 y ago (48) 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 (48). 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 membranar localization (*SI Appendix, Supplementary Information S10*).

Table 3. Prediction for the TGV-S protein by BetaDescribe (Predictions 1 and 2) and BlastP

BetaDescribe (Prediction 1)	FUNCTIONS <i>Envelope glycoprotein that forms spikes at the surface of virion envelope. Essential for the initial attachment to heparan sulfate moieties 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</i> <i>Virion membrane, SUBUNIT</i> <i>Homotrimer; disulfide-linked. Binds to heparan sulfate proteoglycans. Interacts with gH/gL heterodimer, SIMILARITY</i> <i>Belongs to the herpesviridae glycoprotein B family.</i>
BetaDescribe (Prediction 2)	FUNCTIONS <i>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 three conformational states: prefusion native state, prehairpin intermediate state, and postfusion 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</i> <i>Cell membrane. SUBUNIT</i> <i>The mature envelope protein (Env) consists of a trimer of SU-TM heterodimers attached by noncovalent interactions or by a labile interchain disulfide bond</i>
BlastP (protein Q28042, E-value $< 10^{-5}$)	FUNCTIONS <i>Binds to oocyte zona pellucida in vivo. May play a role in the fertilization process and/or early embryonic development, SUBCELLULAR LOCATION</i> <i>Cytoplasmic vesicle, secretory vesicle, SIMILARITY</i> <i>Belongs to the glycosyl hydrolase 18 family.</i>

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 (49). 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 (50). By and large, when discovered, TiLV's ten main proteins showed no significant sequence similarity to other known viral proteins (51). Among these ten, proteins 1–3, were predicted to serve as subunits of TiLV polymerase (52), a prediction that was recently validated experimentally and structurally (53). 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 (*SI Appendix, Supplementary Information S10*). In contrast, BetaDescribe correctly assigned polymerase activity for all three proteins. For Protein 1, the top prediction of BetaDescribe correctly assigned RNA-dependent RNA polymerase activity, while for proteins 2 and 3, BetaDescribe top predictions inaccurately assigned the polymerase activity to be DNA-dependent. 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 (*SI Appendix, Supplementary Information S10*). Interestingly, an endonuclease-like domain was identified in Protein 3, but its functionality remains elusive (53); such activity is included in an alternative BetaDescribe prediction.

H5TRP0. To further demonstrate the utility of BetaDescribe's ability in analyzing nonviral 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 CRISPR, CRISPR-Cas system (54), which is an adaptive immune system in prokaryotes that targets foreign DNA or RNA sequences with high specificity (55). 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). In contrast, BetaDescribe generated two out of three predictions that correctly linked it to CRISPR activity (Table 4). 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 (56).

Association of the Prediction with Functionally Important Protein Regions. Functionally important protein regions can be identified through mutagenesis experiments (57). 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 (*SI Appendix, Supplementary Information S11*).

We report preliminary results for this approach on the extensively studied human insulin protein (P01308; an additional example, the protein RecA, is provided in *SI Appendix, Supplementary Information S11*). 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 (58). Fig. 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.

To further investigate whether BetaDescribe's description generation relies on known functional regions, we analyzed 165 well-characterized proteins from ProteinGym (59) and retrieved their annotated domains from UniProt. Of the 165 proteins, 110 had 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

Table 4. Prediction for the H5TRP0 protein by BetaDescribe (Predictions 1–3) and BlastP

BetaDescribe (Prediction 1)	FUNCTIONS <i>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).</i>
BetaDescribe (Prediction 2)	FUNCTIONS <i>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),</i> COFACTOR <i>Mg(2+), or Mn(2+) required for ssDNA cleavage activity., & iron-sulfur cluster,</i> SIMILARITY <i>Belongs to the CRISPR-associated exonuclease Cas4 family.</i>
BetaDescribe (Prediction 3)	FUNCTIONS <i>Involved in the transposition of the insertion sequence IS5,</i> SIMILARITY <i>Belongs to the transposase 11 family.</i>
BlastP (protein A0A916UE10, E-value of 1)	FUNCTIONS <i>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,</i> CATALYTIC ACTIVITY <i>Hydrolysis of proteins in presence of ATP.,</i> SUBUNIT <i>Homoheptamer. Organized in a ring with a central cavity,</i> SUBCELLULAR LOCATION <i>Cytoplasm,</i> INDUCTION <i>By heat shock,</i> SIMILARITY <i>Belongs to the peptidase S16 family.</i>

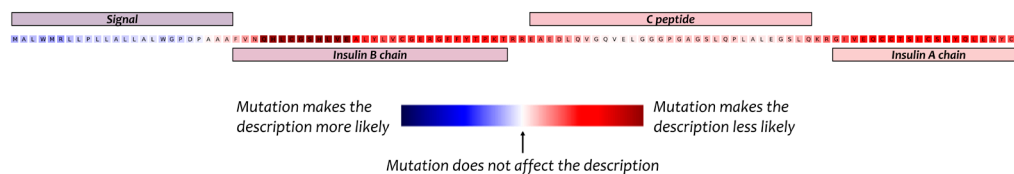


Fig. 3. Identifying functionally important regions for the preproinsulin protein. The four main regions of the insulin are marked: Signal, Insulin B chain, C peptide, and Insulin A chain.

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. *SI Appendix, Supplementary Information S11* also discusses the effect of tokenization on our mutagenesis analysis.

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 particularly valuable for proteins with low sequence homology to known protein families. 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.

The primary contribution of this work lies in the architectural modularity of the BetaDescribe framework, rather than the performance of any single generative backbone. By decoupling the sequence-to-text task from independent validation and adjudication, our system (combining the generator, validators, and the judge) introduces a critical layer of quality control often absent in end-to-end models. This structural separation of concerns ensures that the framework remains model-agnostic and inherently scalable. As the field of protein language modeling continues to advance (driven by the emergence of superior pretrained models, higher-quality datasets, refined architectures, and optimized training procedures) these increasingly powerful generators can be seamlessly integrated as “drop-in” replacements within our system. Ultimately, the BetaDescribe architecture provides a robust, future-proof infrastructure for functional annotation, ensuring that generative gains are consistently grounded by rigorous, automated oversight.

The functional annotation of “hard” proteins, those characterized by low sequence homology to known families, remains a formidable challenge for automated systems. Community-wide initiatives, most notably the Critical Assessment of Functional Annotation (CAFA) challenges (60–62), have systematically highlighted the limitations of traditional alignment-based methods in these sparse regions of the protein space. Emerging generative-based solutions offer a transformative approach to this problem. Unlike homology-based inference, these deep learning frameworks are capable of identifying latent structural and functional motifs, providing a more robust mechanism for predicting the roles of experimentally uncharacterized sequences.

The relationship between BlastP and BetaDescribe resembles the one of traditional information retrieval (such as Google) and generative information retrieval (such as DSI transformer; 63), which works by feeding a query into a neural network and the network returns the document that matches this query. Accordingly, BetaDescribe allows an immediate and intuitive way to obtain descriptions for a given protein rather than a list of homologous proteins. Thus, BetaDescribe serves as a complementary tool for function prediction based on remote homology algorithms. In the future, we expect the users to be able to repeatedly communicate with such LLM-based predictors, and influence the prediction by providing additional information, such as data from ongoing experiments.

Explaining model predictions is common in the NLP domain (64). As exemplified by the preproinsulin and the RecA analysis (*SI Appendix, Supplementary Information S11*), 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.

A key limitation of our automatic evaluation is the small and highly redundant sample of Category 1 proteins analyzed. Because 69 out of the 151 sequences share identical descriptions, the dataset's diversity is heavily constrained. Consequently, these results may not fully reflect the method's performance across a broader selection of distinct proteins. While our manual evaluation provides deep, detailed qualitative insights, it covers only a small subset of examples and cannot replace a comprehensive automatic evaluation.

Materials and Methods

The Generator. The generator is the central model in BetaDescribe. It is a decoder-only model (65) 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 (37). 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.

Training. Starting from the original Llama2 model (37), 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 S1, $\sim 2.9 \times 10^{10}$ 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 (66). 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 S1 contained both English and protein sequences, the English text was not directly connected to the protein sequences. In Dataset S2, 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 (67). For example, protein "AOA0VOV610" with the following protein sequence, "MGREDKTTWKSNNYFLKLV...1," is trained to predict the following: **"FUNCTIONS** 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 S3). 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 (68). We tested two pretraining models and different values for the learning rate for the processing of biological data. For the specific models, training hyperparameters, and dataset configurations, see SI Appendix, Supplementary Information S12.

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 (69). For our generator, we used three alternative prompts (SI Appendix, Supplementary Information S13). 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 (70, 71). 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 (SI Appendix, Supplementary Information S13).

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 diverse descriptions could provide additional insights. By increasing the temperature hyperparameter (specifically utilizing a value of 1.0 in our implementation), the generator can predict descriptions that are not present in our training set.

Architecture. The model architecture is based on the LLAMA2 model with its default tokenization (37). 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. (37) for additional details on the architecture.

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 multilabeled; and 3) presence of enzymatic activity (a binary-classification task). The starting point for each validator is the ESM2 base model (10), 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 (67). Of note, the training and validation sets are strict subsets of the generator training data and validation data, respectively. This ensures that validator models are never exposed to evaluation sequences during training. For the models, tokenizers, training hyperparameters, and datasets, see SI Appendix, Supplementary Information S14.

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 (Fig. 1). Specifically, we used a combination of rule-based decision and prompt-based queries to GPT4 (38) to reject unlikely descriptions (SI Appendix, Supplementary Information S15).

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 ChrF (SI Appendix, Supplementary Information S7). 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 (72, 73). 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 (74).

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 (67). 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. First, the exact match binary metric assigns a value of 1 if the true and inferred descriptions are the exact same string, and 0 otherwise. Next, we utilized two metrics commonly used for evaluating machine-translated text: the ChrF score (42) and BLEU (75), as implemented in SacreBLEU (43). These metrics assign a score between 0 and 1, indicating the degree of similarity between the strings based on shared characters and words. Furthermore, we included the cosine similarity metric, which measures the distance between the embeddings of the descriptions, i.e., the numerical representations, of the descriptions processed by the generator. Unlike the three previous string-based approaches, cosine similarity assesses the semantic meaning of the descriptions (SI Appendix, Supplementary Information S7 provides detailed metric explanations).

Data, Materials, and Software Availability. Models and code data have been deposited in BetaDescribe-code and Github (<https://github.com/technion-cs-nlp/BetaDescribe-code>) (76).

ACKNOWLEDGMENTS. T.P. and Y.B. have received funding from the Israel Science Foundation (Grants 2818/21 and 448/20, respectively).

Author affiliations: ^aThe Henry and Marilyn Taub Faculty of Computer Science, Technion-Israel Institute of Technology, Haifa 3200003, Israel; ^bThe Shmunis School of Biomedicine and Cancer Research, George S. Wise Faculty of Life Sciences, Tel Aviv University, Tel Aviv 69978, Israel; and ^cKempner Institute, Harvard University, Cambridge, MA 02134

1. F. Sanger, E. O. P. Thompson, The amino-acid sequence in the glycol chain of insulin. 2. The investigation of peptides from enzymic hydrolysates. *Biochem. J.* **53**, 366 (1953).
2. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. *Nature* **521**, 436–444 (2015).
3. A. Vouliodimos, N. Doulimas, A. Doulimas, E. Protopapadakis, Deep learning for computer vision: A brief review. *Comput. Intell. Neurosci.* **2018**, 7068349 (2018).
4. T. Young, D. Hazarika, S. Poria, E. Cambria, Recent trends in deep learning based natural language processing. *arXiv [Preprint]* (2018). <http://arxiv.org/abs/1708.02709> (Accessed 29 June 2025).
5. A. B. Nassif, I. Shahin, I. Attili, M. Azzeh, K. Shaalan, Speech recognition using deep neural networks: A systematic review. *IEEE Access* **7**, 19143–19165 (2019).
6. J. Jumper *et al.*, Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
7. E. Simon, K. Swanson, J. Zou, Language models for biological research: A primer. *Nat. Methods* **21**, 1422–1429 (2024).
8. E. Dotan, G. Jaschek, T. Pupko, Y. Belinkov, Effect of tokenization on transformers for biological sequences. *Bioinformatics* **4**, btae196 (2024).
9. T. Hayes *et al.*, Simulating 500 million years of evolution with a language model. *Science* **387**, 850–858 (2025).
10. Z. Lin *et al.*, Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
11. E. Nijkamp, J. A. Ruffolo, E. N. Weinstein, N. Naik, A. Madani, ProGen2: Exploring the boundaries of protein language models. *Cell Syst.* **14**, 968–978.e3 (2023).
12. C. Zhang, Q. Liu, L. Freddolino, StarFunc: Fusing template-based and deep learning approaches for accurate protein function prediction. *Genom. Proteom. Bioinform.*, qzq018 (2026).
13. Z. Wang *et al.*, LM-GVP: An extensible sequence and structure informed deep learning framework for protein property prediction. *Sci. Rep.* **12**, 6832 (2022).
14. E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, G. M. Church, Unified rational protein engineering with sequence-based deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
15. J. Gong *et al.*, THPLM: A sequence-based deep learning framework for protein stability changes prediction upon point variations using pretrained protein language model. *Bioinformatics* **39**, btad646 (2023).
16. A. Elnaggar *et al.*, ProtTrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **44**, 7112–7127 (2022).
17. Y. Jiang *et al.*, MULocDeep: A deep-learning framework for protein subcellular and suborganellar localization prediction with residue-level interpretation. *Comput. Struct. Biotechnol. J.* **1**, 4825–4839 (2021).
18. Y. Jiang *et al.*, MULocDeep web service for protein localization prediction and visualization at subcellular and suborganellar levels. *Nucleic Acids Res.* **51**, W343–W349 (2023).
19. J. Hou, B. Adhikari, J. Cheng, DeepSF: deep convolutional neural network for mapping protein sequences to folds. *Bioinform.* **34**, 1295–1303 (2018).
20. N. Wagner *et al.*, Natural language processing approach to model the secretion signal of type III effectors. *Front. Plant Sci.* **13**, 1024405 (2022).
21. Y. Li *et al.*, HMD-ARG: Hierarchical multi-task deep learning for annotating antibiotic resistance genes. *Microbiome* **9**, 40 (2021).
22. D. Veltri, U. Kamath, A. Shehu, Deep learning improves antimicrobial peptide recognition. *Bioinformatics* **34**, 2740–2747 (2018).
23. Y. Cao, Y. Shen, TALE: transformer-based protein function annotation with joint sequence-label embedding. *Bioinformatics* **37**, 2825–2833 (2021).
24. V. Gligorijević *et al.*, Structure-based protein function prediction using graph convolutional networks. *Nat. Commun.* **12**, 3168 (2021).
25. M. Kulmanov, M. A. Khan, R. Hoehndorf, J. Wren, DeepGO: Predicting protein functions from sequence and interactions using a deep ontology-aware classifier. *Bioinformatics* **34**, 660–668 (2018).
26. M. Littmann, M. Heinzinger, C. Dallago, T. Olenyi, B. Rost, Embeddings from deep learning transfer GO annotations beyond homology. *Sci Rep* **11**, 1160 (2021).
27. T. Sanderson, M. L. Bileschi, D. Belanger, L. J. Colwell, ProtInfer, deep neural networks for protein functional inference. *eLife* **12**, e80942 (2023).
28. N. Strothoff, P. Wagner, M. Wenzel, W. Samek, UDSMPot: Universal deep sequence models for protein classification. *Bioinformatics* **36**, 2401–2409 (2020).
29. A. Sureyya Rifaioğlu, T. Doğan, M. Jesus Martin, R. Cetin-Atalay, V. Atalay, DEEPred: Automated protein function prediction with multi-task feed-forward deep neural networks. *Sci. Rep.* **9**, 7344 (2019).
30. Q. Yuan *et al.*, GPSFun: Geometry-aware protein sequence function predictions with language models. *Nucleic Acids Res.* **52**, W248–W255 (2024).
31. J. Abramson *et al.*, Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* **630**, 493–500 (2024).
32. T. Hayes *et al.*, Simulating 500 million years of evolution with a language model. *Science* **387**, 850–858 (2025).
33. H. Abidine, M. Chatzianastasis, C. Bouyioukos, M. Vazirgiannis, Prot2Text: Multimodal protein's function generation with GNNs and transformers. *AAAI* **38**, 10757–10765 (2024).
34. L. Zhuo *et al.*, "ProtLLM: An interleaved protein-language LLM with protein-as-word pre-training" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L. W. Ku, A. Martins, V. Srikumar, Eds. (Association for Computational Linguistics, Bangkok, Thailand, 2024), pp. 8950–8963.
35. Z. Liu *et al.*, "ProtT3: Protein-to-text generation for text-based protein understanding" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, L. W. Ku, A. Martins, V. Srikumar, Eds., Association for Computational Linguistics, Bangkok, Thailand, 2024, pp. 5949–5966.
36. X. Fei *et al.*, "Prot2Text-V2: Protein function prediction with multimodal contrastive alignment" in *Advances in Neural Information Processing System 38*, C. Zhang *et al.*, Eds. (Advances in Neural Information Processing Systems, San Diego, CA, 2025).
37. H. Touvron *et al.*, Llama 2: Open foundation and fine-tuned chat models. *arXiv [Preprint]* (2023). <http://arxiv.org/abs/2307.09288> (Accessed 29 June 2025).
38. OpenAI, GPT-4 technical report. *arXiv [Preprint]* (2023). <https://arxiv.org/abs/2303.08774v3> (Accessed 29 June 2025).
39. P. Haluptzok, M. Bowers, A. T. Kalai, "Language models can teach themselves to program better" in *International Conference on Learning Representations (ICLR, Kigali, Rwanda, 2023)*.
40. S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman, Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
41. G. Moreno-Hagelsieb, K. Latimer, Choosing BLAST options for better detection of orthologs as reciprocal best hits. *Bioinformatics* **24**, 319–324 (2008).
42. M. Popović, "chrF: character n-gram F-score for automatic MT evaluation" in *Proceedings of the Tenth Workshop on Statistical Machine Translation*, O. Bojar *et al.*, Eds. (Association for Computational Linguistics, Lisbon, Portugal, 2015), pp. 392–395.
43. M. Post, "A call for clarity in reporting BLEU scores" in *Proceedings of the Third Conference on Machine Translation: Research Papers*, O. Bojar *et al.*, Eds. (Association for Computational Linguistics, Brussels, Belgium, 2018), pp. 186–191.
44. S. F. Altschul *et al.*, Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
45. S. R. Eddy, Accelerated profile HMM searches. *PLoS Comput. Biol.* **7**, e1002195 (2011).
46. Gemini Team, P. Georgiev *et al.*, Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv [preprint]* (2024). <http://arxiv.org/abs/10.48550/arXiv.2403.05530> (Accessed 29 June 2025).
47. S. Karnieli *et al.*, Discovery of an unrecognized nidovirus associated with granulomatous hepatitis in rainbow trout. *iScience* **26**, 106370 (2023).
48. D. Hart, G. N. Frerichs, A. Rambaut, D. E. Onions, Complete nucleotide sequence and transcriptional analysis of snakehead fish retrovirus. *J. Virol.* **70**, 3606–3616 (1996).
49. M. Eyring *et al.*, Identification of a novel RNA virus lethal to Tilapia. *J. Clin. Microbiol.* **52**, 4137–4146 (2014).
50. M. D. Jansen, H. T. Dong, C. V. Mohan, Tilapia lake virus: A threat to the global tilapia industry? *Rev. Aquac.* **11**, 725–739 (2019).
51. E. Bacharach *et al.*, Characterization of a novel orthomyxo-like virus causing mass die-offs of tilapia. *mBio* **7**, e00431 (2016).
52. R. Abu Rass *et al.*, Inferring protein function in an emerging virus: Detection of the nucleoprotein in Tilapia Lake virus. *J. Virol.* **96**, e0175721 (2022).
53. B. Arragain, M. Pelosse, A. Thompson, S. Cusack, Structural and functional analysis of the minimal orthomyxovirus-like polymerase of Tilapia Lake Virus from the highly diverged Amnoonviridae family. *Nat. Commun.* **14**, 8145 (2023).
54. G. Bigelyte *et al.*, Innate programmable DNA binding by CRISPR-Cas12m effectors enable efficient base editing. *Nucleic Acids Res.* **52**, 3234–3248 (2024).
55. Y. Ishino, H. Shinagawa, K. Makino, M. Amemura, 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**, 5429–5433 (1987).
56. T. Wiegand *et al.*, TnpB homologues exapted from transposons are RNA-guided transcription factors. *Nature* **631**, 439–448 (2024).
57. C. A. Hutchison *et al.*, Mutagenesis at a specific position in a DNA sequence. *J. Biol. Chem.* **253**, 6551–6560 (1978).
58. D. F. Steiner, S. Y. Park, J. Støy, L. H. Philipson, G. I. Bell, A brief perspective on insulin prediction. *Diabetes Obes. Metab.* **11**, 189–196 (2009).
59. P. Notin *et al.*, ProteinGym: Large-scale benchmarks for protein fitness prediction and design. *Adv. Neural Inf. Process. Syst.* **15**, 64331–64379 (2023).
60. I. Kahanda, C. S. Funk, F. Ullah, K. M. Verspoor, A. Ben-Hur, A close look at protein function prediction evaluation protocols. *Gigascience* **14**, 41 (2015).
61. N. Zhou *et al.*, The CAFE challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome Biol.* **19**, 244 (2019).
62. Y. Jiang *et al.*, An expanded evaluation of protein function prediction methods shows an improvement in accuracy. *Genome Biol.* **17**, 184 (2016).
63. Y. Tay *et al.*, "Transformer memory as a differentiable search index" in *Advances in Neural Information Processing Systems 35*, A. Agarwal, D. Belgrave, K. Cho, A. Oh, Eds. (Advances in Neural Information Processing Systems, New Orleans, LA, 2022).
64. P. Atanasova, J. G. Simonsen, C. Lioma, I. Augenstein, "A diagnostic study of explainability techniques for text classification" in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, B. Webber, T. Cohn, Y. He, Y. Liu, Eds. (Online: Association for Computational Linguistics, 2020), pp. 3256–3274.
65. A. Radford, K. Narasimhan, T. Salimans, I. Sutskever, Improving language understanding by generative pre-training. OpenAI [Preprint] (2018). https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf. Accessed 29 June 2025.
66. B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, the UniProt Consortium, UniRef clusters: A comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).
67. The UniProt Consortium, UniProt: The universal protein knowledgebase. *Nucleic Acids Res.* **45**, D158–D169 (2017).
68. C. Tan *et al.*, "A survey on deep transfer learning" in *Artificial Neural Networks and Machine Learning (ICANN 2018)*, V. Kůrková, Y. Manolopoulos, B. Hammer, L. Iliadis, I. Maglogiannis, Eds. (Springer International Publishing, Cham, 2018), pp. 270–9.
69. P. Sahoo *et al.*, A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv [Preprint]* (2024). <http://arxiv.org/abs/10.48550/arXiv.2402.07927> (Accessed 29 June 2025).
70. D. H. Ackley, G. E. Hinton, T. J. Sejnowski, A learning algorithm for Boltzmann machines. *Cogn. Sci.* **9**, 147–169 (1985).
71. G. Hinton, O. Vinyals, J. Dean, Distilling the knowledge in a neural network. *arXiv [Preprint]* (2015). <http://arxiv.org/abs/1503.02531> (Accessed 29 June 2025).
72. V. D. Blondel, J. L. Guillaume, R. Lambiotte, E. Lefebvre, Fast unfolding of communities in large networks. *J. Stat. Mech.* **2008**, P10008 (2008).
73. F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, D. Parisi, Defining and identifying communities in networks. *Proc. Natl. Acad. Sci. U.S.A.* **101**, 2658–2663 (2004).
74. A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using networkX. (2008).
75. K. Papineni, S. Roukos, T. Ward, W. J. Zhu, "BLEU: A method for automatic evaluation of machine translation" in *ACL '02*, P. Isabelle, E. Charniak, D. Lin, Eds. (Association for Computational Linguistics, Philadelphia, PA, 2002), pp. 311–318.
76. E. Dotan *et al.*, Code and Models from: "BetaDescribe: Providing Rich Descriptions from Protein Sequences." GitHub. <https://github.com/technion-cs-nlp/BetaDescribe-code>. Deposited 20 June 2025.