Decoding the Molecular Language of Proteins with Evolla

Xibin Zhou[1†], Chenchen Han[1†], Yingqi Zhang[1‡], Jin Su[1‡], Kai Zhuang[1‡], Shiyu Jiang[1‡], Zichen Yuan[1], Wei Zheng[2], Fengyuan Dai[1], Yuyang Zhou[1], Yuyang Tao[1], Dan Wu[1], Fajie Yuan[1*]

[1]School of Engineering, Westlake University, Hangzhou, 310024, China.
[2]School of Statistics and Data Science, Nankai University, Tianjin, 300071, China.

*Corresponding author(s). E-mail(s): yuanfajie@westlake.edu.cn;
†These authors contributed equally to this work.
‡These authors also contributed equally to this work.

Abstract

Proteins, nature's intricate molecular machines, are the products of billions of years of evolution and play fundamental roles in sustaining life. Yet, deciphering their molecular language - that is, understanding how protein sequences and structures encode and determine biological functions - remains a cornerstone challenge in modern biology. Here, we introduce Evolla, an 80 billion frontier protein-language generative model designed to decode the molecular language of proteins. By integrating information from protein sequences, structures, and user queries, Evolla generates precise and contextually nuanced insights into protein function. A key innovation of Evolla lies in its training on an unprecedented AI-generated dataset: 546 million protein question-answer pairs and 150 billion word tokens, designed to reflect the immense complexity and functional diversity of proteins. Post-pretraining, Evolla integrates Direct Preference Optimization (DPO) to refine the model based on preference signals and Retrieval-Augmented Generation (RAG) for external knowledge incorporation, improving response quality and relevance. To evaluate its performance, we propose a novel framework, Instructional Response Space (IRS), demonstrating that Evolla delivers expert-level insights, advancing research in proteomics and functional genomics while shedding light on the molecular logic encoded in proteins. The online demo is available at http://www.chat-protein.com/.

1 Introduction

Proteins are nature's molecular machines, performing an astonishing diversity of tasks essential to sustaining life. Their functions are encoded in their sequences and manifested through their precisely folded three-dimensional structures [1]. Understanding protein function is one of the most fundamental challenges in modern biology, as it holds the key to unraveling cellular mechanisms, advancing drug discovery, and exploring the molecular basis of life. Despite transformative advances in experimental and computational technologies, fully decoding protein function remains an elusive goal.

Recent breakthroughs in protein structure prediction, led by AlphaFold [2–4] and its successors, have transformed our ability to predict protein structures with unprecedented accuracy. Paradoxically, however, this progress has further highlighted the gap between structural determination and functional understanding, underscoring the urgent need for novel approaches to functional annotation. This challenge is exacerbated by the sheer scale of modern protein data: while high-throughput sequencing has uncovered hundreds of

millions of protein sequences, fewer than half a million have expert-curated functional annotations [5]. This widening gap between the availability of sequence and structural data and the scarcity of functional insights has become a critical bottleneck in biological research and its applications.

Traditional computational tools, such as BLAST [6] for sequence alignment and Foldseek [7] or TM-align [8] for structural comparison, have been invaluable for identifying homologous proteins with known functions. However, these methods rely heavily on similarity to previously annotated proteins, making them less effective when analyzing proteins that lack explicit evolutionary relatives. Recent advances in deep learning-based approaches [9–18] have broadened the scope of functional prediction. Yet, these methods often fall short in generating the detailed and context-specific insights that biologists need for practical research. Furthermore, their dependence on limited annotated training data restricts both their accuracy and their capacity to generalize across the immense diversity of protein families and biological roles.

The emergence of large multimodal protein-language models offers a promising avenue to redefine traditional protein function prediction tasks. These models aim to decode the molecular "language" of proteins by leveraging protein language models for understanding protein sequences and structures, and utilizing large natural language models for generating functional insights. Over the past year, several preprint studies have emerged in this field, reflecting the growing interest in this area [19–24]. Despite this progress, the training data used in these studies remains a significant bottleneck. Most datasets are limited to just 500,000 to 3 million protein-text pairs, with fewer than 100 million word tokens, far below the scale required to fully capture the complexity and diversity of protein biology.

Here, we introduce Evolla, an 80B-parameter multimodal protein-language model that deciphers protein molecular mechanisms through natural language dialogue. Evolla features a novel architecture that seamlessly integrates three key components: a protein language model (PLM) serving as the encoder, a large language model (LLM) as the decoder, and an intermediate compression and alignment module. The model was trained on a massive corpus of 546 million AI-generated protein-question-answer triples, encompassing 150 billion word tokens - an unprecedented scale in the protein-text modeling domain. These extensive datasets were generated through an innovative pipeline that combines text-protein retrieval, prompt engineering, and LLM distillation, creating high-quality paired associations between proteins and their functional descriptions. To enhance response quality, Evolla leverages Retrieval-Augmented Generation (RAG) [25] for knowledge integration and Direct Preference Optimization (DPO) [26] for output refinement. Our comprehensive evaluations, including a novel Instructional Response Space (IRS) assessment framework, demonstrate Evolla's superior capability in generating precise, nuanced, and contextually-relevant protein functional responses. Evolla's ability to decode the molecular language of proteins opens new avenues for exploring the complexity of life at an unparalleled scale and depth.

2 Evolla model overview

Recent advancements in Vision Large Language Models [27–29] have demonstrated their effectiveness in multimodal question answering and cross-modal retrieval. Building on these breakthroughs, we introduce

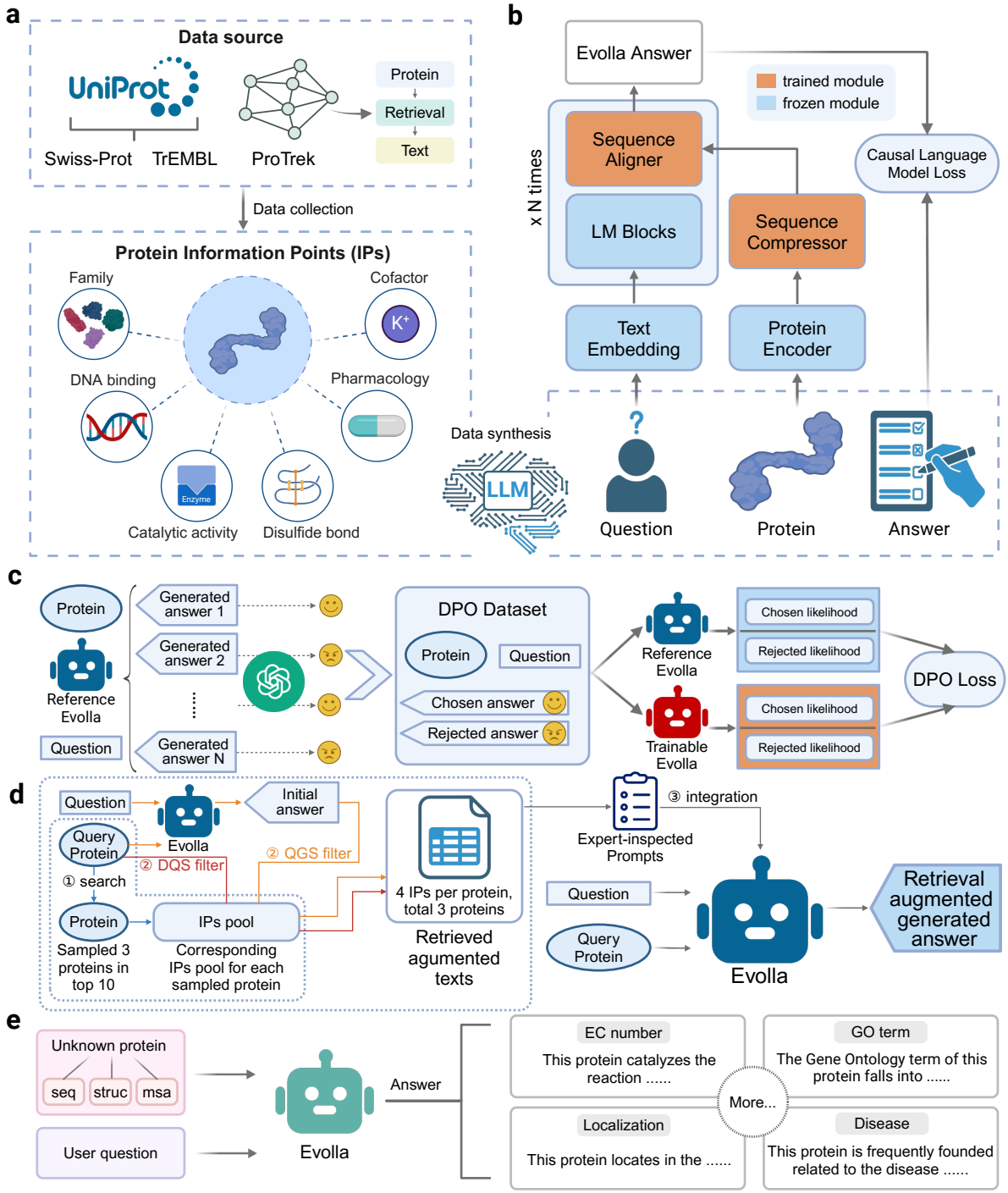


Fig. 1 The schematic overview of Evolla model. **(a)** Pipeline of AI-assisted training data generation. The dataset was sourced from the Swiss-Prot and TrEMBL databases, with protein annotations generated using ProTrek specifically for proteins in the TrEMBL dataset. We built structured, high-quality training data by collecting Protein Information Points (IPs) - concise text segments that capture specific protein characteristics. **(b)** Model training process. LLM-synthesized protein-question-answer triples served as the foundation for model training. The model, optimized through causal language modeling objectives, generates precise, protein-contextual responses to biological queries. **(c)** Direct Preference Optimization (DPO). The process consists of data generation and DPO training. The red Evolla represents the trainable model, while the blue Evolla represents the frozen reference model. **(d)** Retrieval Augmented Generation (RAG). During inference, RAG utilized direct query selection (DQS) and QA-guided selection (QGS) strategies to enhance response accuracy and relevance. **(e)** Applications of Evolla. Evolla interprets diverse protein queries, delivering precise insights derived from amino acid sequences, structural features, and multiple sequence alignments (MSAs).

Evolla, a multimodal protein-language generative model for protein functional question answering. Evolla comes in two versions: a base model of 10B parameters and a larger model of 80B parameters, the latter of which is still being trained and has not yet reached convergence. The 10B model combines a frozen 650M-parameter protein encoder, a trainable 1.7B-parameter sequence compressor and aligner, and a frozen 8B-parameter pre-trained natural language LLM. The larger 80B version extends this architecture with a

frozen 1.3B-parameter protein encoder, a trainable 8B-parameter sequence compressor and aligner, and a frozen 70B-parameter LLM. Both versions of Evolla effectively merge protein-specific knowledge with the vast general knowledge embedded in LLMs. (Figure 1b and Methods 6.1).

Evolla's protein encoder extracts high-quality representations from diverse protein data types. To identify the most suitable encoder, we benchmarked three prominent protein language models (PLMs): SaProt [12], a structure-aware PLM; ESM2 [14], a sequence-only PLM; and MSA-Transformer [30], a homology-based model (Section 6.1.1). Among these, SapProt demonstrated the best performance and was selected for training the larger version of Evolla (Figure 3a). The decoder of Evolla is a natural language LLM (e.g., LLama3 [31]) that transforms protein-specific representations into human-readable answers for protein functional questions. By leveraging the extensive general knowledge embedded in pre-trained LLMs, the decoder bridges the gap between protein-specific encodings and natural language understanding, enabling accurate and contextually relevant responses. To further enhance Evolla's performance, we integrated two key components: a sequence compressor and a sequence aligner. The sequence compressor, based on a cross-attention Transformer module [27, 32], converts variable-length protein sequences into fixed-length, high-quality representations, ensuring efficient and consistent processing by downstream components. The sequence aligner, also implemented as a cross-attention Transformer, incorporates gates [27] and residual connections [33] to seamlessly integrate protein sequence embeddings into intermediate LLM layers. This integration facilitates a smooth flow of information between protein-specific representations and the LLM's general knowledge, enhancing Evolla's ability to provide accurate, biologically meaningful insights for protein functional question-answering.

In addition to the core modules, we incorporated direct preference optimization (DPO) [26] to further fine-tune Evolla's responses based on GPT-guided preferences (Figure 1c, Section 6.2). This process involved two key stages: dataset generation and DPO training. In the data generation stage, multiple answers were generated by Evolla and scored using an advanced GPT system. Based on these evaluations, tetrads (protein, question, chosen answer, rejected answer) were constructed, with the chosen answer receiving a higher score than the rejected one, forming the DPO dataset. In the subsequent training phase, two copies of Evolla were used: a frozen reference model, whose parameters remained fixed, and a trainable model, whose parameters were updated during DPO training. The DPO loss is computed from pairwise preference comparisons between the frozen reference model and the trainable model, where pairwise preference is defined as the ratio of the likelihoods of the chosen answer to the rejected answer. A higher ratio indicates that the model assigns greater likelihood to the chosen answer, reflecting its preference for the correct answer and leading to improved predictive precision. This loss function was then used to fine-tune the trainable model, optimizing its alignment with high-quality responses while preserving the integrity of the reference model.

To further enhance inference accuracy, we employed retrieval-augmented generation (RAG) (Figure 1d, Section 6.3) with two retrieval strategies: direct query selection (DQS) and QA-guided selection (QGS). Both methods retrieve the top 10 Swiss-Prot proteins related to the query protein and randomly sample three for further processing. To provide the most valuable contexts of the query protein and enhance accuracy when generating, for each sampled protein, DQS filters the top 4 information points (IPs, basic units of annotation,

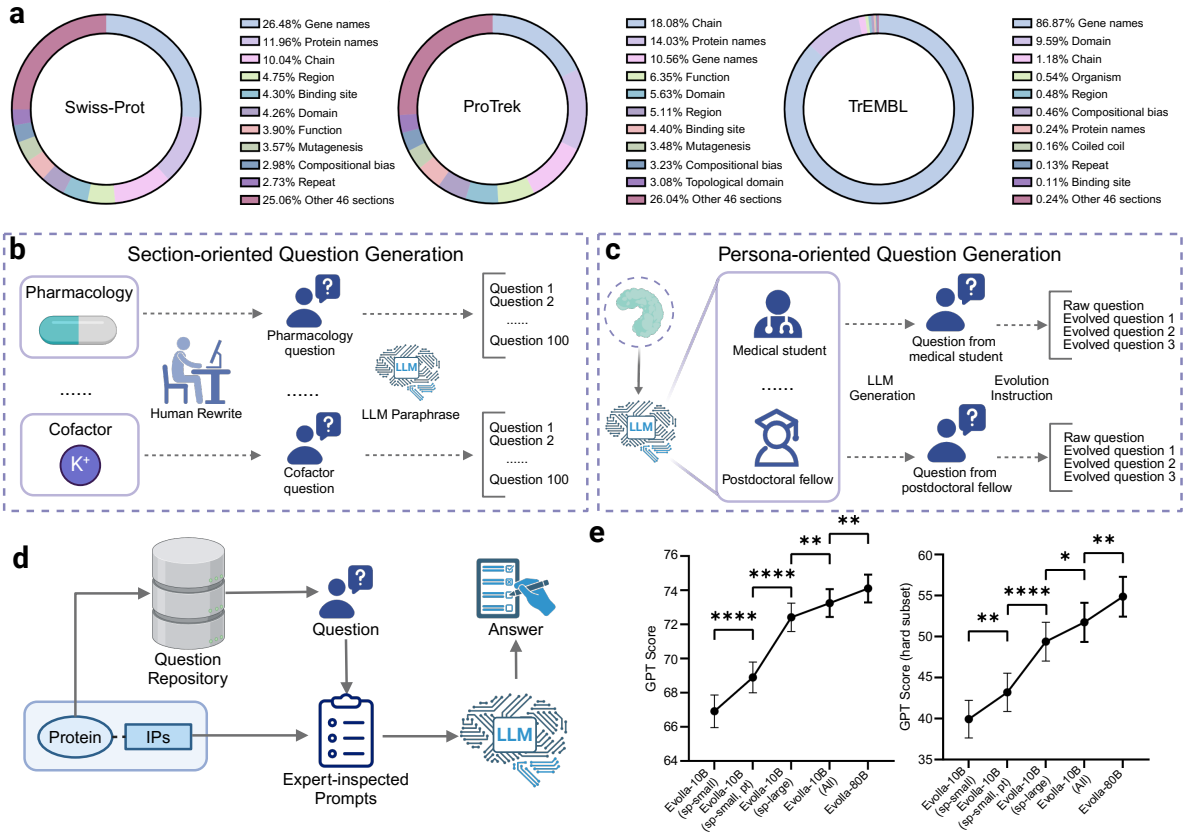


Fig. 2 AI-generated training data pipeline and data scaling performance. (a) Proportions of unique texts for different annotation sections in Swiss-Prot, ProTrek, and TrEMBL datasets. Swiss-Prot (Left) and ProTrek (Middle) exhibit balanced category proportions across diverse sections, whereas TrEMBL (Right) is dominated by *Gene names* and *Domain* section. **(b)** Schematic of section-oriented question generation. Definitions from UniProt sections are used to generate section-specific questions by human experts, which are subsequently paraphrased by a large language model (LLM) to expand the question repository. **(c)** Persona-oriented question generation pipeline. Given the associated IPs of an input protein, the descriptions of personas, along with the IPs, are fed into the LLM to generate role-specific questions based on the protein's context, which then evolve with LLM evolution instruction tuning to expand the diversity. **(d)** Overview of answer generation pipeline. Combined with IPs related to the protein, the question sampled from question repository is incorporated into expert-inspected prompts, which are then inputted into LLM to generate answers. **(e)** Model performance comparison measured by GPT scores on test set (Left) and its hard subset (Right) for various data volumes (*sp-small*: dataset of around 800,000 triples from Swiss-Prot, *pt*: dataset of around 3 million triples from ProTrek, *sp-large*: dataset of around 3 million triples from Swiss-Prot, *All*: dataset of 546 million triples). Evolla-80B, trained on dataset *All*, is still undergoing training and has not yet reached convergence.

details in Section 3 and Section 6.5) from its all corresponding IPs based on their similarity to the query protein, while QGS uses an initial answer to guide IP filtering. Expert-inspected prompts, incorporating the sampled IPs as additional context, were concatenated with the original question to generate the final answer. This combination of retrieval strategies and expert-guided prompting ensures both efficiency and accuracy in Evolla's responses.

Figure 1e illustrates the user application workflow of the Evolla model. The process begins with the input of a protein, provided as a sequence, structure, or multiple sequence alignment (MSA). Users can then ask specific questions regarding the protein's properties or functions, and Evolla generates precise, contextually relevant answers. Key applications include, but are not limited to, Enzyme Commission (EC) number classification, Gene Ontology (GO) annotation, subcellular localization prediction, and disease association analysis, among other protein function characterization tasks. These features position Evolla as a powerful and versatile tool for protein function annotation and broader biomedical research.

3 AI-generated training data boosts Evolla performance

Evolla's remarkable performance is driven by its large-scale model parameters and the unprecedented size of its training dataset. The training data is constructed from key sources (Figure 2a): Swiss-Prot [5], TrEMBL [5], and search results from the multimodal retrieval tool ProTrek [34]. Swiss-Prot provides expertly curated protein-text pairs supported by experimental evidence, while TrEMBL offers high-quality protein sequences that lack experimentally validated text descriptions. To generate reliable text descriptions for these proteins, the ProTrek tool is employed to retrieve relevant protein texts (Figure 1a).

We extracted around 14 million Information Points (IPs) - basic units of protein knowledge with precise functional descriptions - from Swiss-Prot (Release 2023_03), covering 56 annotation sections across 569,793 proteins (Section 6.5.1). For TrEMBL proteins, we employed sequence clustering at 50% identity threshold, yielding 41,229,217 unique cluster centers. Each center protein was processed through ProTrek to generate top-10 IPs by two different strategies, resulting in a comprehensive collection of 824,584,340 IPs (Section 6.5.2).

Individual IPs typically capture specific aspects of protein function but fail to provide comprehensive functional characterization. To address this limitation, we developed an innovative LLM-based augmentation pipeline that transforms these IPs into detailed natural language Q&A pairs (Section 6.6). Our pipeline incorporates two complementary question generation strategies: section-oriented and persona-oriented approaches (Figure 2b,c). The section-oriented strategy generates expert-level questions based on UniProt section definitions, while the persona-oriented approach creates diverse questions by interpreting protein information through various persona perspectives [35]. For the latter, we combined protein IPs with descriptions from a persona hub to generate initial questions, which were then refined using predefined evolution instructions to ensure diversity and depth [36]. The resulting comprehensive question repository, derived from both strategies, was integrated with corresponding IPs into expert-reviewed prompts. These enriched prompts were then processed by the LLM to generate high-quality answers (Figure 2d).

In summary, we curated a comprehensive dataset comprising over 546 million protein-question-answer triples, with around 150 billion word tokens, sourced from Swiss-Prot (16 million triples, 6.9 billion tokens) and ProTrek (530 million triples, 142 billion tokens).

4 Results

To rigorously evaluate the performance, we employed sequence clustering with a 50% identity threshold for Swiss-Prot proteins (Release 2023_03), partitioning them into training (561,187 proteins), validation (4,406 proteins, 1,000 clusters), and test sets (4,197 proteins, 1,000 clusters) (Section 6.7.1). Furthermore, we extracted a hard subset from the test set based on sequence similarity to the training set (Section 6.7.2). Notably, proteins in the hard test set share less than 30% sequence identity with those in the training set, making it a stringent benchmark for novel proteins. To further evaluate the model's performance on newly added proteins, we curated a test set comprising proteins and their corresponding annotations that

were incorporated into Swiss-Prot between Release 2023_03 and 2024_05, hereafter referred to as Swiss-Prot (NEW) (Section 6.7.3).

Since protein question-answering (protein Q&A) is a newly established task, the field lacks widely accepted evaluation metrics. Traditional question-answering (Q&A) evaluation metrics prevalent in the AI community, such as ROUGE [37] and BLEU [38], rely on lexical matching, which fails to capture the factual accuracy crucial for knowledge-intensive tasks. To address these limitations, we introduce two novel evaluation approaches. The ProTrek score quantifies semantic alignment between generated responses and target proteins by using ProTrek. The GPT score serves as our primary evaluation benchmark, leveraging the advanced LLM to assess response fidelity against expert-curated information points (IPs) (Section 6.8).

In this study, unless otherwise specified, all significance tests were performed using the Wilcoxon test. The significance levels are indicated as follows: * for $p < 0.05$, ** for $p < 0.01$, *** for $p < 0.001$, and **** for $p < 0.0001$. Unless stated otherwise, all results are presented with $\pm$ the 95% confidence intervals of the reported values.

4.1 Protein encoders

We evaluated three well-known protein encoders with Llama3-8B decoder trained on *sp-large* dataset (around 3 million triples from Swiss-Prot), each leveraging distinct protein representations: ESM2-650M (amino acid sequences), SaProt-650M (integrated amino acid and 3D structural data), and MSA-Transformer-100M (multiple sequence alignments) (Figure 3a and Section 6.1.1). Analysis of mean GPT scores revealed that SaProt-650M achieved the highest performance (72.41 ± 0.83), followed by ESM2-650M (70.78 ± 0.85) and MSA-Transformer (69.31 ± 0.87). These results highlight the advantages of jointly modeling amino acid sequences and 3D structures. Based on these findings, we adopted SaProt as the protein encoder for subsequent Evolla evaluations.

4.2 Direct preference optimization (DPO)

Preference learning techniques, particularly Direct Preference Optimization (DPO) [26], have emerged as effective methods for fine-tuning large language models (LLMs) with relatively small datasets (Section 6.2). Using Evolla with SaProt-650M encoder and Llama3-8B decoder (trained on *sp-large* dataaset) as our baseline, we implemented DPO training on a compact dataset of 37,544 tetrads, using GPT scores as preference signals (Figure 3c). This approach significantly improved the GPT score from 72.41 ± 0.83 to 78.89 ± 0.78. These results underscore the efficiency of DPO in leveraging minimal training data to achieve significant performance gains.

4.3 Retrieval augmented generation (RAG)

Here, we evaluate the performance of Evolla by applying two Retrieval-Augmented Generation (RAG) strategies: direct query selection (DQS) and QA-guided selection (QGS) (Figure 3d and 3e, Section 6.3). Our experiments revealed that Evolla equipped with both RAG strategies significantly outperformed its base model trained on *sp-large* dataset, demonstrating the effectiveness of these RAG strategies (Figure 3d).

Specifically, Evolla (sp-large) enhanced with DQS and QGS achieved mean GPT scores of 76.27 ± 0.72 and 76.91 ± 0.71, respectively, compared to 72.41 ± 0.83 for Evolla (sp-large). These results highlight that integrating RAG strategies, particularly DQS and QGS, significantly enhances response accuracy and performance, validating their efficacy in leveraging information points from similar proteins.

To assess whether the superior performance is driven solely by the retrieved information, we applied both RAG strategies (DQS and QGS) to the uni-modal Llama3-8B model and compared the results with Evolla (sp-large) (Figure 3e). The mean GPT scores were 67.38 ± 0.78 and 53.77 ± 0.79 for Llama3-8B (DQS) and Llama3-8B (QGS), respectively, both significantly lower than the Evolla (sp-large). These results indicate that Evolla, trained on protein-specific data, outperforms generic LLMs enhanced with RAG.

4.4 Scaling of model size and data volume

Here, we investigated the scaling effects associated with increasing both model size and training data volume. We evaluated Evolla-10B (SaProt-650M encoder, Llama3-8B decoder), training it on four progressively larger datasets: around 800,000 triples from Swiss-Prot (sp-small), around 800,000 triples from Swiss-Prot combined with around 3 million ProTrek-annotated triples (sp-small, pt), around 3 million triples from Swiss-Prot (sp-large), all 546 million triples (All). To assess the effectiveness of the *All* dataset, we also conducted training on Evolla-80B (SaProt-1.3B encoder and Llama3-70B decoder) using the All dataset, ensuring a more comprehensive evaluation. We then evaluated the models on the test set, including the more stringent hard subset, which serves as a more demanding performance evaluation (Figure 2e, Left and Right).

The analysis of mean GPT scores on test set revealed a clear scaling trend with respect to training data (Figure 2e Left). Evolla-10B trained on the *sp-small* dataset achieved a score of 66.91 ± 0.96. With the addition of ProTrek-annotated triples (sp-small, pt), the score improved to 68.90 ± 0.90. Training on the *sp-large* dataset further increased the score to 72.41 ± 0.83, while using the full *All* dataset resulted in a score of 73.25 ± 0.81. Finally, Evolla-80B, trained on the *All* dataset, achieved the highest score of 74.10 ± 0.81.

To further assess the model's generalization capabilities, particularly on unseen proteins in the more challenging hard subset, we conducted a comprehensive evaluation (Figure 2e Right). Evolla-10B (sp-small) achieved 39.93 ± 2.29, and Evolla-10B (sp-small, pt) improved to 43.20 ± 2.33. Training on the *sp-large* dataset further enhanced performance, reaching 49.38 ± 2.36, while Evolla-10B (All) attained 51.73 ± 2.38. With the increase in model size, Evolla-80B (All), with significantly more parameters, achieved the highest performance, reaching 54.86 ± 2.43.

We also oberserved improved performance with increasing model size (Figure 3b). Using the *sp-large* dataset, we trained the model with different combinations of protein encoders and LLMs: SaProt-35M with OPT-350M (SaProt-35M-OPT-350M), SaProt-650M with OPT-350M (SaProt-650M-OPT-350M), SaProt-650M with Llama3-8B (SaProt-650M-Llama3-8B, corresponding to Evolla-10B) and finally SaProt-1.3B with Llama3-70B (SaProt-1.3B-Llama3-70B, corresponding to Evolla-80B), which was trained on the full *All* dataset. The mean GPT scores were 53.64 ± 1.00, 62.94 ± 0.89, 72.41 ± 0.83 and 74.10 ± 0.81 for each respective combination.

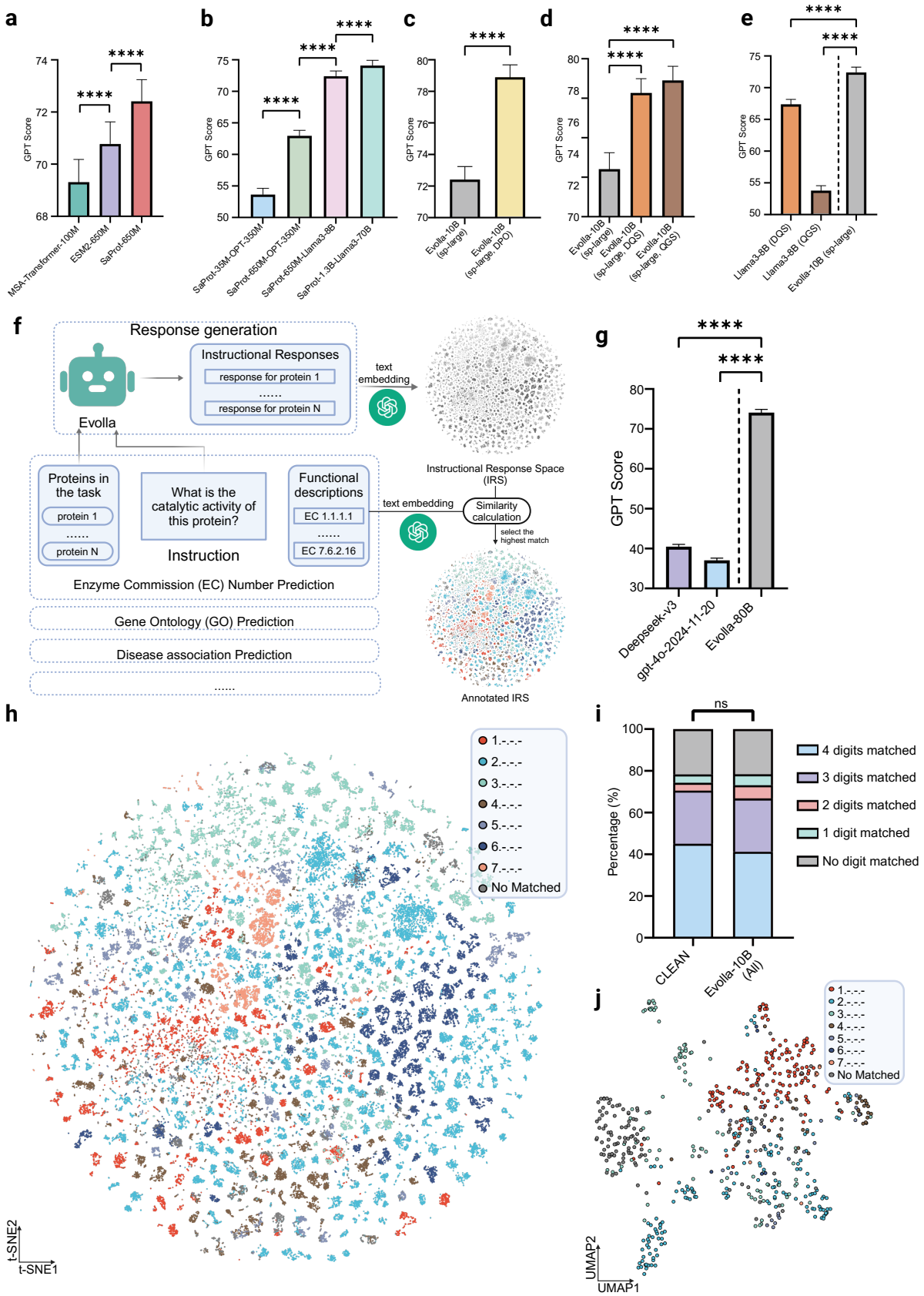


Fig. 3 Model performance and IRS evaluation. (a) Performance of different protein encoders. Measured by GPT score, **(b)** Performance of data parameter scaling. **(c)** Performance of direct preference optimization (DPO). **(d)** Performance of retrieval augmented generation (RAG) strategies, direct query selection (DQS) and QA-guided selection (QGS). **(e)** Performance comparison of Evolla and large language model enhanced by two RAG strategies (*Llama3-8B (DQS)* and *Llama3-8B (QGS)*). **(f)** Overview of IRS framework. The instructions and proteins were input into Evolla to generate responses, which, along with functional descriptions, were embedded into the IRS using a text embedding model. The most semantically similar description was then selected as the protein's annotation. **(g)** Performance comparison of Evolla and advanced general purpose LLMs (*Deepseek-v3* and *gpt-4o-2024-11-20*). **(h)** t-SNE plot of Evolla's response embeddings for enzymes in the Swiss-Prot database (Release 2023_03). Clusters represent semantically similar responses, with each cluster predominantly color-coded by the same label. **(i)** Comparison of EC number matching performance between CLEAN and Evolla-10B (All) on the enzymes of Swiss-Prot (NEW). Despite not being explicitly trained for EC number classification, Evolla achieved comparable performance to the state-of-the-art CLEAN model, with no significant difference (ns: non-significant). **(j)** UMAP plot of response embeddings for enzymes in the Swiss-Prot (NEW), color-coded by the highest matched EC numbers, as in (f).

In summary, our analysis highlights the clear scaling effects of both model size and training data volume on performance. As model size increased (from SaProt-35M-OPT-350M to Evolla-80B) and as the training data expanded (from *sp-small* to *All*), performance consistently improved across both the test set and the more challenging hard subset. Notably, Evolla-80B, trained on the full *All* dataset, achieved the highest scores, underscoring the importance of both larger models and richer datasets in achieving superior generalization capabilities.

4.5 Benchmarking Evolla against General-Purpose LLMs

To assess Evolla's overall capability, we conducted a comparative analysis against two state-of-the-art general-purpose language models: *Deepseek-v3* and *gpt-4o-2024-11-20*. Each model was evaluated using carefully designed prompts (in Table S15) containing amino acid sequences and specific functional queries, with performance measured using the previously described GPT score metric. Our Evolla-80B, featuring the SaProt-1.3B encoder and Llama3-70B decoder trained on All dataset, achieved a mean GPT score of 74.10±0.81 (Figure 3g). In contrast, *Deepseek-v3* and *gpt-4o-2024-11-20* scored 40.49±0.56 and 37.07±0.54 respectively. These results reveal a substantial performance gap, with Evolla-80B demonstrating nearly double the effectiveness of general-purpose pure natural language LLMs in protein function generation.

5 Evolla generates instructional response space (IRS)

Here, we introduce a novel framework, the Instructional Response Space (IRS), designed to evaluate Evolla's advanced understanding of specific protein properties through its generated responses (Figure 3f). As a case study, here we demonstrate its application in assessing catalytic activity, showcasing capabilities in tasks such as Enzyme Commission (EC) number prediction. We select a set of enzymes with EC numbers and pose a unified instruction regarding their catalytic activity (e.g., "What is the catalytic activity of this protein?"). We input this instruction, along with the proteins, into the Evolla model to generate natural language responses that represent the model's interpretation of each protein's catalytic function. These responses collectively form the Instructional Response Space (IRS), which encapsulates the model's understanding of the proteins' activities. To provide references, we compile detailed functional description texts for each EC number. Using OpenAI's advanced *text-embedding-3-large* model [39], we map both the generated responses (linked to the proteins) and the EC number functional descriptions (linked to the EC numbers) into a high-dimensional semantic space. Finally, we predict the EC number of each protein by evaluating the semantic alignment between the responses and the functional descriptions in this space.

To further evaluate Evolla's performance on the EC number prediction task, we utilized t-distributed Stochastic Neighbor Embedding (t-SNE) (for efficiency reasons) [40] to visualize enzymes from Swiss-Prot Release 2023_03 (Figure 3h) and unified manifold approximation and projection (UMAP) [41] for enzymes from Swiss-Prot (NEW) (Figure 3j) within the framework of the IRS. In the embedding space, each cluster represents text embeddings of generated responses with similar semantic meaning, suggesting that the corresponding proteins likely share similar functional properties. Figure 3h reveals that nearly all clusters are

uniformly colored, indicating that Evolla's generated responses effectively capture the catalytic characteristics of proteins based on task-specific instructions. This strong alignment between clusters and functional labels underscores Evolla's ability to produce semantically meaningful and biologically relevant outputs, further validating its utility in protein annotation and prediction tasks.

IRS also provides a framework for benchmarking Evolla against tools developed specifically for specific function prediction, such as CLEAN [42] for EC number prediction. To ensure a fair comparison, here we evaluated both CLEAN and Evolla on enzymes in Swiss-Prot (NEW) that were unseen by both models (Section 6.7.3 and 6.9.1). For each enzyme, the predicted EC numbers were compared to the ground truth, and the percentage of matched digits were reported. The performance difference between the two models was not statistically significant, as indicated by the Wilcoxon test (Figure 3i). CLEAN achieved a 45.0% match for all four digits (correct substrate and reaction type), while Evolla-10B (All) achieved 41.2%. For a three-digit match (correct reaction type), CLEAN reached 70.5%, while Evolla-10B (All) achieved 66.8%. Notably, Evolla's performance is achieved without any specific fine-tuning on the EC number prediction task, showcasing its strong generalization ability and reinforcing its competitive edge. More importantly, Evolla is a generative model, while CLEAN is a classification model that relies on completely different technical approaches. It is worth emphasizing that it is not an easy task for a generative model to outperform a classification model trained on a task-specific dataset.

6 Methods

6.1 Model architecture

6.1.1 Protein encoders

Proteins can be represented through diverse data modalities, including amino acid sequences, 3D structural data, and multiple sequence alignments (MSAs). To integrate these heterogeneous representations, we employ advanced deep learning models tailored to each data type, mapping them to high-dimensional feature vectors. Specifically, we utilize the state-of-the-art models ESM2-650M [14] for amino acid sequences, SaProt-650M [12] for the joint modeling of amino acid sequences and 3D structural information and MSA Transformer [30] for MSA sequences.

Since the protein structures used in our study are derived from AlphaFoldDB [43], it is important to acknowledge that some regions of these predicted structures may be unreliable. To assess the reliability of these predictions, we employed the predicted Local Distance Difference Test (pLDDT) [3], a metric commonly used to quantify the confidence in the accuracy of local structural predictions, especially for AlphaFold family. Residuals with pLDDT values below 70 are considered less reliable, as they indicate regions with higher uncertainty in their structural predictions. Consequently, we applied a masking procedure to exclude these unreliable regions from the training process. Specifically, any residual with a pLDDT score lower than 70 was masked using a placeholder #, thereby ensuring that SaProt-650M was trained only on the regions of the structure with a higher degree of confidence.

Let the original input of ESM2-650M and SaProt-650M be denoted as $X_{\text{ori}} \in \mathbb{V}^L$, where $\mathbb{V}$ represents the different vocabulary for different model and L being the length of the protein sequence. For ESM2-650M, $\mathbb{V}_{\text{ESM}}$ represents the vocabulary of amino acids ($|\mathbb{V}_{\text{ESM}}| = 20$), while $\mathbb{V}_{\text{SaProt}}$ is the vocabulary of structure-aware tokens for SaProt-650M ($|\mathbb{V}_{\text{SaProt}}| = 400$). In contrast, for MSA Transformer, the input differs in structure and is defined as $X_{\text{ori}} \in \mathbb{V}^{M \times L}$, where M denotes the depth of the multiple sequence alignment (MSA) and L is the length of the protein sequence. In this case,$\mathbb{V}$ corresponds to the amino acid vocabulary, and the additional dimension M reflects the number of aligned sequences in the MSA.

Given an input X_{ori} fed into the respective protein encoders, the output $X_p = \text{ProteinEncoder}(X_{\text{ori}}) \in \mathbb{R}^{L \times d_p}$ represents the protein embedding, where L is the length of the protein sequence and d_p is the dimensionality of the embedding. This embedding X_p captures rich, high-dimensional features and serves as the compact representation of the protein. For the MSA Transformer, the input X_{ori} and output X_p have different dimensions due to the application of mean pooling over the MSA depth dimension M. The ProteinEncoder in this case could be one of the ESM2-650M, SaProt-650M or MSA Transformer.

6.1.2 Sequence compressor

Proteins exhibit significant variability in size, with lengths ranging from tens to even thousands of amino acids. This variability poses a challenge in obtaining a consistent, fixed-length representation that effectively captures the global information. To address this challenge, we introduce the **Sequence Compressor** (SC), a cross-attention-based Transformer module designed to generate high-quality, fixed-length representations from variable-length protein embeddings. The Sequence Compressor is capable of transforming arbitrary-length protein embeddings encoded from diverse data types into fixed-length representations in the high-dimensional latent space. These uniform-length representations provide a powerful foundation for large language models (LLMs) to further interpret protein data.

Specifically, the Sequence Compressor consists of an Attention module and a Feed-Forward module (FFW) with residual connections, as show in Supplementary Figure S1(a). The Attention module receives input from the protein encoder, which processes different types of sequences. Let the output representations of the protein encoder be denoted as $X_p \in \mathbb{R}^{L \times d_p}$, where L represents the length of the sequence which is variable, and d_p is the dimensionality of the feature space. These representations serve as the keys and values in the cross-attention mechanism. Simultaneously, we introduce randomly initialized and learnable fixed-length latent tokens $X_q \in \mathbb{R}^{L_{\text{fix}} \times d_c^{\text{SC}}}$, where L_{fix} is the fixed number of tokens and d_c^{SC} is the dimensionality of the latent space. These tokens serve as the queries for the cross-attention mechanism.

The attention mechanism can be formally expressed as:

$$\text{Attention}(Q_{\text{SC}}, K_{\text{SC}}, V_{\text{SC}}) = \text{softmax}\left(\frac{Q_{\text{SC}} K_{\text{SC}}^T}{\sqrt{d_k^{\text{SC}}}}\right) V_{\text{SC}}$$

where $Q_{\text{SC}} = X_q W_Q^{\text{SC}} \in \mathbb{R}^{L_{\text{fix}} \times d_k^{\text{SC}}}$ are the queries, $K_{\text{SC}} = X_p W_K^{\text{SC}} \in \mathbb{R}^{L \times d_k^{\text{SC}}}$ are the keys, and $V_{\text{SC}} = X_p W_V^{\text{SC}} \in \mathbb{R}^{L \times d_c^{\text{SC}}}$ are the values. Here, $W_Q^{\text{SC}} \in \mathbb{R}^{d_c^{\text{SC}} \times d_k^{\text{SC}}}$, $W_K^{\text{SC}} \in \mathbb{R}^{d_p^{\text{SC}} \times d_k^{\text{SC}}}$ and $W_V^{\text{SC}} \in \mathbb{R}^{d_p^{\text{SC}} \times d_c^{\text{SC}}}$ are the learnable weight matrix to project queries, keys and values. d_k^{SC} is the dimensionality of the query and the

key vectors. The softmax function ensures that the model attends to relevant portions of the input sequence, allowing the extraction of informative features from variable-length input sequences.

After the attention mechanism has been applied, the output is passed through a residual connection, which adds the original input to the output of the attention operation:

$$O_{\text{Att}}^{\text{SC}} = \text{Attention}(Q_{\text{SC}}, K_{\text{SC}}, V_{\text{SC}}) + Q_{\text{SC}}$$

where $O_{\text{Att}}^{\text{SC}} \in \mathbb{R}^{L_{\text{fix}} \times d_c^{\text{SC}}}$ is the output of the attention mechanism with residual connection.

Following the attention operation, the output is passed through a Feed-Forward module (FFW), a multi-layer perceptron (MLP), with residual connections, which helps to stabilize training and improve the model's ability to learn complex patterns in the data.

The residual connections are defined as:

$$Z_{\text{SC}} = \text{FFW}(O_{\text{Att}}^{\text{SC}}) + O_{\text{Att}}^{\text{SC}}$$

where $O_{\text{Att}}^{\text{SC}} \in \mathbb{R}^{L_{\text{fix}} \times d_c^{\text{SC}}}$ is the input to the Feed-Forward module, and $\text{FFW}(O_{\text{Att}})$ is the transformed output of the module. This residual structure allows for the efficient flow of gradients during backpropagation and ensures that the model retains important low-level features from the input data.

The resulting output is a fixed-length latent representation $Z_{\text{SC}} \in \mathbb{R}^{L_{\text{fix}} \times d_c^{\text{SC}}}$, which captures the global context of the input protein sequence. This embedding can then be used for downstream tasks, such as protein function prediction, interaction modeling, or further analysis using large language models.

6.1.3 Sequence aligner

Large language models (LLMs) are inherently designed for understanding and generating natural language, with their architecture and training processes optimized for textual data. However, their ability to handle non-linguistic data modalities, such as protein sequences, remains a significant challenge. Proteins, for example, are composed of amino acid sequences, which follow biological rules and structures distinct from natural language. To overcome this challenge and enable LLMs to effectively interpret protein sequences, we propose the integration of a novel **Sequence Aligner (SA)** module, which is designed to inject protein-related information into the LLM processing pipeline.

The Sequence Aligner employs a cross-attention-based Transformer architecture, incorporating gates and residual connections to facilitate seamless integration of protein sequence embeddings within the intermediate layers of LLMs, as show in Supplementary Figure S1(b). This approach utilizes a cross-attention mechanism, where the fixed-length representations of the protein sequences, generated by the Sequence Compressor (SC), serve as the keys and values in the attention mechanism, while the activations of the LLM layers act as the queries. This design allows the model to directly align protein sequence data with textual information, facilitating the understanding of both modalities in a unified representation.

Specifically, let the protein sequence embeddings generated by the Sequence Compressor be represented as $Z_{\text{SC}} \in \mathbb{R}^{L_{\text{fix}} \times d_c^{\text{SC}}}$, where L_{fix} is the fixed length of compressed protein representation and d_c^{SC} is the

dimension of representations. These representations capture the structural and biological properties of the protein sequence and are used as the keys and values in the cross-attention mechanism.

At the same time, the activations of the LLM layer, denoted as $X_{\text{LLM}} \in \mathbb{R}^{L_{\text{LLM}} \times d_{\text{LLM}}}$, are employed as queries, where L_{LLM} is the length of the input sentence and d_{LLM} is the dimensionality of the output of the LLM layer. This setup ensures that the LLM's activations, which contain information about the linguistic context, are aligned with the protein sequence representations during the attention computation.

The core of the Sequence Aligner is the cross-attention mechanism, which allows the LLM to dynamically focus on relevant regions of the protein sequence embeddings while processing the textual data. Specifically, the attention mechanism computes the interaction between the queries (from the LLM activations) and the keys/values (from the protein sequence embeddings) as follows:

$$\text{Attention}(Q_{\text{SA}}, K_{\text{SA}}, V_{\text{SA}}) = \text{softmax}\left(\frac{Q_{\text{SA}} K_{\text{SA}}^T}{\sqrt{d_k^{\text{SA}}}}\right) V_{\text{SA}}$$

where $Q_{\text{SA}} = X_{\text{LLM}} W_Q^{\text{SA}} \in \mathbb{R}^{L_{\text{LLM}} \times d_k^{\text{SA}}}$ are the queries, $K_{\text{SA}} = Z_{\text{SC}} W_K^{\text{SA}} \in \mathbb{R}^{L_{\text{fix}} \times d_k^{\text{SA}}}$ are the keys, and $V_{\text{SA}} = Z_{\text{SC}} W_V^{\text{SA}} \in \mathbb{R}^{L_{\text{fix}} \times d_{\text{LLM}}}$ are the values. Here, $W_Q^{\text{SA}} \in \mathbb{R}^{d_{\text{LLM}} \times d_k^{\text{SA}}}$, $W_K^{\text{SA}} \in \mathbb{R}^{d_c^{\text{SC}} \times d_k^{\text{SA}}}$ and $W_V^{\text{SA}} \in \mathbb{R}^{d_c^{\text{SC}} \times d_{\text{LLM}}}$ are the learnable weight matrix to project queries, keys and values. d_k^{SA} is the dimensionality of the query and the key vectors.

The resulting attention output allows the LLM to selectively incorporate protein-related information into its processing, enriching its understanding of both the protein sequence and the associated textual context.

A key innovation of the Sequence Aligner is the introduction of learnable gate parameters [27], initially set to zero. This allows the LLM to gradually integrate protein data, adapting incrementally without overwhelming the model. The gates regulate the flow of protein-related information, enabling the model to learn from this data over time while maintaining stable learning dynamics.

Specifically, the protein information contribution is modulated by a gating function with residual connection, which can be expressed as:

$$O_{\text{Att}}^{\text{SA}} = \tanh\left(W_G^{\text{Att}}\right) * \text{Attention}(Q_{\text{SA}}, K_{\text{SA}}, V_{\text{SA}}) + Q_{\text{SA}}$$

where $W_G^{\text{Att}} \in \mathbb{R}$ is the learned gate parameter for cross-attention layer, and $\tanh(\cdot)$ is the tanh activation function.

The gate mechanism is also introduced in the Feed Forward module, where the output is computed as:

$$Z_{\text{SA}} = \tanh\left(W_G^{\text{FFW}}\right) * \text{FFW}(O_{\text{Att}}^{\text{SA}}) + O_{\text{Att}}^{\text{SA}}$$

where $W_G^{\text{FFW}} \in \mathbb{R}$ is the learned gate parameter for Feed Forward layer, and $\tanh(\cdot)$ denotes the tanh activation function. The output Z_{SA} of the Sequence Aligner module is then passed as input to the subsequent layer in the LLM.

By introducing these architectural components, the Sequence Aligner effectively bridges the gap between protein sequence data and natural language, enabling LLMs to better understand complex multimodal data and perform tasks that require both biological and linguistic knowledge.

6.1.4 Large language models (LLMs) decoder

To leverage the extensive knowledge embedded in Large Language Models (LLMs), we adopt LLMs of varying scales as the backbone for our approach. Specifically, we experiment with models such as OPT-350M [44] , Llama3-8B [31] and Llama3-70B [31], selecting them for their diverse capabilities and scalability in handling complex tasks. These models serve as a foundation for interpreting protein data, benefiting from their ability to generalize across a wide range of domains.

To more effectively integrate protein-specific information, we insert the sequence aligner, as previously introduced, between the layers of the LLM. This architectural modification allows for a more seamless interaction between the protein representations and the language model, enhancing the model's ability to capture domain-specific nuances and improve performance on protein-related tasks.

6.2 Direct preference optimization (DPO)

Ensuring alignment between model outputs and human intentions is a fundamental requirement for applications in specialized domains where factual accuracy is paramount [45]. Traditional evaluation metrics, such as ROUGE [37] and BLEU [38], provide quantitative evaluations but do not capture the essential dimensions of factual correctness, especially in knowledge-intensive tasks. Although expert feedback remains the gold standard for evaluation, it is neither scalable nor cost-effective. To overcome these limitations, AI systems that refer to trusted domain-specific knowledge repositories, such as Swiss-Prot, offer a scalable and reliable alternative to assess outputs. These systems enable efficient and domain-specific evaluation of generated content, bridging the gap between traditional metrics and the need for factual precision in specialized tasks.

Although Reinforcement Learning from Human Feedback (RLHF) [46] has proven effective in aligning LLMs with human preferences, its reliance on costly and time-consuming expert evaluations limits scalability, particularly in specialized domains. AI-driven feedback has emerged as a scalable and efficient alternative to address these challenges, using domain-specific knowledge to simulate expert judgment [47]. Building on this approach, we examine Direct Preference Optimization (DPO) [26], a preference learning technique designed to improve training stability and reliability of Evolla outputs. The DPO pipeline includes two key stages, preference data generation and DPO training, along with an optional checkpoint selection stage.

6.2.1 Preference data generation

A total of 45,000 (protein, question) pairs were randomly sampled from the training set. Evolla generated four candidate responses for each pair, which were evaluated using GPT scoring (Section 6.8.2), which assesses the accuracy and consistency of the responses with the protein's information points (IPs, Section 6.5).

To ensure meaningful training signals, we selected the highest and lowest scoring responses among the four generated answers with a sufficiently large score gap to form chosen and rejected answer pairs. Specifically, for GPT preference data, pairs with a score gap of 10 or greater were chosen, yielding 37,544 samples. Each data sample consists of a tetrad (protein, question, chosen answer, rejected answer), and these samples provided robust training examples for fine-tuning the models.

6.2.2 DPO training

Two Evolla replicas were involved during training: a frozen reference Evolla, whose parameters remained fixed, and a trainable Evolla, whose parameters were updated during DPO training. The frozen model served as a reference, ensuring that the updated model aligned with the preference pairs without excessively deviating from the original behavior.

The DPO loss function is defined by the following:

$$L_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(p,x,y_c,y_r)\sim\mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_c \mid p, x)}{\pi_{\text{ref}}(y_c \mid p, x)} - \beta \log \frac{\pi_\theta(y_r \mid p, x)}{\pi_{\text{ref}}(y_r \mid p, x)} \right) \right]$$

where π_θ represents the trainable Evolla and π_{ref} is the frozen reference Evolla. The training data set $\mathcal{D}$ consists of (protein, question, chosen answer, rejected answer) tetrads, denoted as p, x, y_c and y_r, respectively. The conditional likelihood $\pi(y \mid p, x)$ represents the probability of generating the entire answer $y = (y_1, y_2, ..., y_n)$ given the question x and protein p. It is defined as following:

$$\pi(y \mid p, x) = \prod_{t=1}^{n} \pi(y_t \mid y_{<t}, p, x)$$

where y_t is the t-th token in the sequence, $y_{<t}$ denotes the preceding tokens and n is the length of the entire sequence.

During training, the DPO loss was minimized with respect to the trainable model's parameters θ, the model to assign a higher likelihood to the chosen answer y_c compared to the rejected answer y_r. As a result, the trainable model π_θ was updated to better align with the preferences encoded in the dataset $\mathcal{D}$.

6.2.3 Checkpoint selection

Since the DPO loss reflects the preference alignment of chosen and rejected answers between the trainable and frozen models, it is not an ideal metric to evaluate the quality of generated responses. Instead, we selected the best checkpoint based on the quality of the generation. To manage the computational demands of generation, 500 triples were randomly sampled from the validation set to form a DPO validation subset. The generated responses for this subset were evaluated using GPT scores. During DPO training, checkpoints were saved at the end of each epoch without real-time scoring. After training, GPT scores of generated responses in the validation subset were calculated for all checkpoints, and the checkpoint performing best was selected for further testing. This approach ensured efficient and robust evaluation of generative performance.

6.3 Retrieval augmented generation (RAG)

Large Language Models (LLMs) have demonstrated remarkable capabilities across a wide range of tasks; however, they are still prone to a significant issue known as "hallucination" [48]. This phenomenon occurs when LLMs generate responses that appear factually accurate but are, in reality, fabricated or misleading. Such errors undermine the reliability and trustworthiness of LLMs, particularly in high-stakes applications.

To address this limitation, Retrieval-Augmented Generation (RAG) has been proposed as a promising solution [25]. RAG enhances LLM performance by integrating external knowledge retrieval mechanisms, sourcing the most relevant information from trusted databases. This enables the model to align its output with verifiable facts, thus improving accuracy and calibrating its responses to better match human expectations.

We proposed two strategies of retrieval-augmented generation (RAG) for Evolla: direct query selection (DQS) and QA-guided selection (QGS). Both approaches begin by retrieving the top 10 protein sequences from the Swiss-Prot database using ProTrek, based on similarity to the query protein. To enhance the diversity of the retrieved information, three proteins (excluding the query protein) are randomly selected from the top 10 hits.

To filter the IPs, we apply either DQS or QGS. In DQS, the top 4 IPs are selected for each of the three randomly chosen proteins based on their similarity to the query protein, as determined by ProTrek. In QGS, an answer is first generated using the query protein and the question, after which the top 4 IPs, ranked by similarity to the generated answer, are selected for each of the three proteins.

Finally, the retrieved proteins and their corresponding IPs are formatted according to expert-inspected prompts and concatenated with the question to generate the final answer. Representative cases are provided in Supplementary Table S16 and the overall pipeline is in Supplementary Figure S6.

6.4 Training details

Model Configuration. For Evolla-10B, we utilize SaProt-650M as the protein encoder, which takes protein sequences of up to 1024 SA tokens as input. Following this, a Sequence Compressor module is employed, consisting of 6 layers, 8 attention heads, 64 latent units, and a feedforward multiplier of 4. We initialize the LLM decoder with weights from HuggingFace's `meta-llama/Meta-Llama-3-8B-Instruct`[1], setting the maximum token length to 736. The Sequence Aligner comprises 8 cross-attention layers, each with an FFN multiplier of 4 and a 0.1 dropout rate applied to the attention probabilities, inserted after every 4 layers of the LLM decoder. For Evolla-80B, we utilize SaProt-1.3B as the protein encoder, with the same configuration for sequence compression. We initialize the LLM decoder with weights from HuggingFace's `meta-llama/Llama-3.1-70B-Instruct`[2], setting the maximum token length to 736. The Sequence Aligner consists of 20 cross-attention layers with an FFN multiplier of 1, applying a 0.1 dropout rate to the attention probabilities after every 4 layers of the LLM decoder. For both Evolla, SaProt encoder and Llama3 decoder are frozen, while the Sequence Compressor and Sequence Aligner are trainable, with a total of 1.7 billion trainable parameters for Evolla-10B and 8.2 billion trainable parameters for Evolla-80B. This configuration serves as the default setup for our experiments.

Training Strategy. We optimized Evolla-10B using DeepSpeed [49] with the AdamW optimizer [50], setting $\beta_1 = 0.9$, $\beta_2 = 0.98$, and applying L2 weight decay of 0.05. The model was trained with bf16 mixed precision [51], and the learning rate was initially warmed up from 0 to 1×10^{-5} over the first 2000 steps, followed by decay to 1×10^{-7} according to a cosine annealing schedule [52]. Training was conducted on 32

[1] https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[2] https://huggingface.co/meta-llama/Llama-3.1-70B-Instruct

NVIDIA A100 80GB GPUs. For Evolla-80B, we utilized FSDP [53] with HYBRID_SHARD on 64 NVIDIA H800 80GB GPUs. All other configurations were consistent with those used for Evolla-10B.

Generation. During generation, a temperature of 0.6 and top-p sampling with a probability of 0.9 are used, with a maximum of 512 new tokens generated per sequence.

6.5 Information Points (IPs)

6.5.1 IPs in UniProt database

Information Points (IPs) are the fundamental units of protein annotation in the UniProt database, consisting of concise textual descriptions that capture specific aspects of a protein, its residues, or segments. Each annotation typically comprises a few words or sentences and is categorized into distinct sections based on the nature of the information.

A single protein can be annotated across multiple sections and have different annotations within a single section. For instance, an enzyme might have annotations in both "Function" and "Catalytic Activity" sections, and it may also contain multiple annotations within a single section, such as "Cofactor", if it binds to more than one cofactor. Each annotation, irrespective of its section or context, is treated as an independent information point.

To construct our dataset, we collected IPs from the UniProt database, encompassing both sequence-level data (31 sections) and residue-level data (25 sections), yielding a total of 56 distinct annotation sections (Supplementary Table S1). Detailed description and examples for each sections are listed in the Supplementary File. Sequence-level IPs describe protein-wide attributes, such as "Function" or "Catalytic Activity", whereas residue-level IPs provide detailed information about specific residues or segments, including sections such as "Active Site" or "DNA Binding".

Our primary dataset was sourced from the Swiss-Prot database (Release 2023_03), which comprises 569,793 proteins annotated with a total of 14,818,151 IPs. Swiss-Prot is widely regarded as the gold standard for protein annotation due to its expert-reviewed entries, ensuring high accuracy and reliability [54].

To supplement this, we also gathered IPs from the TrEMBL database, which contains over 2.4 billion IPs. However, unlike Swiss-Prot, TrEMBL annotations are predominantly generated through automated pipelines and lack expert review [54] (Appendix A.2). This results in considerable data imbalance, making some sections dominate in the dataset (Figure 2a Right).

Given these differences, we prioritize Swiss-Prot as the primary foundation for constructing our dataset due to its high reliability and curated accuracy. Although TrEMBL provides a vast scale of IPs, its automated and unbalanced annotations limit its suitability as a primary data source when accuracy is critical. However, TrEMBL proteins remain valuable resources for expanding the dataset, offering broader coverage of protein universe.

6.5.2 IP-augmentation by ProTrek

Although the Swiss-Prot database is renowned for its expert-reviewed entries and high reliability, its dataset of approximately 570,000 proteins represents only a small fraction of the vast protein universe, which is

estimated to include billions of proteins. This significant gap poses a challenge for comprehensive protein research, particularly in the context of large-scale studies, where the diversity and breadth of the dataset are crucial for ensuring robust model training and generalization.

To address this limitation, we incorporated ProTrek [34], an advanced pre-trained tri-modal retrieval model designed to integrate sequence, structure, and text modalities, to annotate the proteins. ProTrek offers a unique capability to retrieve and associate texts from diverse protein-related datasets, enabling us to augment our dataset with additional information for underrepresented or novel proteins. This approach not only expands the scale of the dataset but also enhances its diversity by including proteins with limited experimental annotations or those identified in novel environments, such as metagenomic studies.

To efficiently manage the scale and redundancy of non-curated datasets like TrEMBL, we clustered all TrEMBL proteins using a sequence identity threshold of 50%, resulting in 41,229,217 cluster centers. These cluster centers represent unique sequence groups, effectively reducing redundancy while maintaining diversity. For these cluster centers, we utilized ProTrek to perform a comprehensive search from sequence to text modalities, followed by two distinct strategies for filtering assigned Information Points (IPs): global top 10 IPs and top 10 IPs per section. In the global top 10 IPs strategy, we retrieved the top IPs from a comprehensive global text repository, yielding a total of 412,292,170 IPs. To refine this process, we introduced the top 10 IPs per section strategy, which involved first identifying the highest-ranking IP from each of the 56 distinct sections, and subsequently selecting the top 10 IPs from these 56. This strategy also resulted in a total of 412,292,170 IPs, maintaining broad coverage across the sections. By focusing on these representative clusters and employing targeted IP filtering strategies, our approach ensures a balance between computational efficiency and the richness of the dataset, effectively optimizing both performance and diversity while maintaining a total of 824,584,340 IPs.

6.6 Information-Point-driven high-quality Q&A data construction

To effectively train the model in a question-answering paradigm, the two critical components were critical in data construction: question formulation and answer generation. For question generation, we employed both section-oriented and persona-oriented approaches to enhance the diversity and relevance of the questions. These techniques help to ensure that the questions cover a broad range of protein-related topics and are contextually rich. For answer generation, we utilized expert-inspected prompts, which combine protein-related questions with corresponding IPs. These prompts were designed to guide large language models (LLMs) in generating highly accurate and contextually appropriate responses.

In the following sections, we elaborate on the processes of question generation and answer generation, detailing the methodologies used to optimize the performance of our system.

6.6.1 Question generation by sections and personas

Question generation is a crucial first step in constructing question-answer (Q&A) pairs. Since the IPs are organized into specific sections, questions can be designed around these sections. For each section, we curated a root question based on its definition, which was then fed into GPT-3.5 and GPT-4 chat interface

to generate 50 rephrased versions each, producing a total of 100 diverse questions. This section-oriented question generation does not explicitly incorporate protein-specific IPs, leading to questions that tend to be general in nature. While such questions are useful for broad coverage, they may lack the detailed focus needed to probe deeper into the specifics of each individual protein.

To address this limitation, we incorporated persona-oriented strategy to generate questions. We utilized a persona hub [35] consisting of 10 distinct personas related to biology, ranging from high school student to medical professional (Supplementary Table S3), each with a unique knowledge level and perspective. For each question generation instance, three personas were randomly selected, and their descriptions were incorporated into the prompt (Supplementary Table S2) alongside the relevant protein IPs. This process resulted in three distinct questions, each tailored to one of the selected personas, capturing the diversity of viewpoints and expertise represented by the personas.

To further increase the diversity of the generated questions, we implemented the process of question evolution [36]. In this step, we selected three random directions from a pool of five predefined evolution strategies (Supplementary Table S4). Crucially, no new information about proteins was introduced during this phase. Instead, the evolution process focused exclusively on enhancing the existing questions by expanding their scope, complexity, or focus. As a result, this evolutionary step increased the diversity of the questions, ensuring that they covered different facets of the protein, aligning with the nuanced and multifaceted nature of protein-related information (Supplementary Table S5).

6.6.2 High quality Q&A pairs generation from IPs by LLM

An individual IP cannot reflect the comprehensive properties of a protein, so we propose to leverage LLM to synthesize and summarize this information to generate a more detailed description regarding the protein and user query.

In our approach, we employed the *glm-4-flash* model, provided by ZhipuAI [55], chosen for its fast processing speed, cost-effectiveness, and stable performance. This model was integrated with both the IPs and pre-curated questions from our question repository. The carefully designed prompts, as outlined in Supplementary Table S6, were used to guide the model in generating the corresponding answers.

Since the IPs are derived from different sources, the Swiss-Prot database and the ProTrek model, we employed different strategies to process them. For IPs originating from Swiss-Prot, we applied section-oriented questions and persona-oriented questions to guide the answer generation. The section-oriented questions focused on specific protein features, while the persona-oriented questions simulated different user perspectives. The approaches resulted in over 16 million protein-question-answer triples, comprising more than 6.9 billion text tokens in total (9,447,442 triples with 2,747,587,987 tokens for section-oriented approach and 6,695,246 triples with 4,187,323,323 tokens for persona-oriented approach). Given that we used two distinct methods to filter IPs (global top 10 and top 10 per section), we generated answers separately for each set using only the section-oriented questions. To ensure the relevance and alignment between the generated answers and proteins, we applied an empirical threshold score of 12 based on the ProTrek scoring system. This threshold was selected to represent a strong correlation between the answers and the corresponding

proteins. We then calculated the ProTrek score for each protein-answer pair and retained those with scores above the threshold. This process resulted in the generation of over 530 million high-confidence triples with over 142 billion tokens (530,406,949 triples with 142,634,317,457 text tokens), covering more than 40 million proteins.

In summary, we leveraged a massive dataset consisting of over 546 million protein-question-answer triples, totaling approximately 150 billion text tokens to train Evolla (546,549,637 triples and 149,569,228,767 tokens). This comprehensive and diverse dataset provided a solid foundation for training the model, enabling it to learn nuanced relationships and complex patterns in protein-language interactions.

6.7 Dataset Partitioning

Dataset partitioning plays a critical role in evaluating the performance and generalization of machine learning models. In this study, we partitioned the dataset based on protein sequence identity, a widely adopted and robust practice in bioinformatics [6, 56]. Partitioning by sequence identity ensures that training, validation and test sets are sufficiently distinct, minimizing potential bias caused by sequence similarity while enabling a more accurate assessment of the model's ability to generalize to novel proteins. This approach also reflects real world scenarios, where the model is expected to predict the properties or annotations of previously unseen proteins.

6.7.1 Partition of training, validation, test set

The proteins in the Swiss-Prot database (Release 2023_03) were partitioned into training, validation, and test sets. The training set was used exclusively for model training, while the validation set was employed for calculating validation loss and determining early stopping criteria. The test set was reserved for final model evaluation. To ensure distinct sets and minimize overlap, *MMseqs2* [57], a widely-used and efficient sequence clustering tool, was applied to group the proteins based on sequence identity. A threshold of 50% sequence identity was used, resulting in 127,465 distinct cluster centers. This ensures that any protein in a given cluster has more than 50% sequence identity with other members of the same cluster, and less than 50% sequence identity with proteins in any other cluster. To guarantee robust representation in both the validation and test sets, 1,000 randomly selected clusters were assigned to each set, while the remaining clusters were allocated to the training set. This resulted in 4,406 proteins in the validation set, 4,197 proteins in the test set, and 561,187 proteins in the training set. Once the protein sets were defined, the corresponding text annotations and Q&A pairs were assigned to the appropriate sets, ensuring consistency in data distribution across all sets.

6.7.2 Imbalance of training set and fine-grained partition of test set

To further explore the class imbalance within the training set, we applied more stringent clustering criteria to the training data. Specifically, we clustered the proteins in the training set using a 30% sequence identity threshold, which resulted in 65,130 distinct clusters. Notably, the top 988 clusters (1.52% of the total) contained proteins that accounted for half of all proteins in the training set. This observation reveals a significant

imbalance in the protein distribution within the Swiss-Prot training set, likely reflecting inherent biases in the curation process or historical research priorities reflected in the Swiss-Prot database (Supplementary Figure S5).

In light of this clustering imbalance, we sought to further partition the test set based on its similarity to the training set, with the hypothesis that proteins from well-represented, highly annotated clusters would be easier for the model to learn. To achieve this, we performed a similarity search using *MMseq2* [57] on the test set against the training set, applying the same 30% sequence identity threshold. Proteins from the test set that matched any protein from the top 988 clusters in the training set were classified as the "easy set", due to their higher likelihood of having rich annotations. Proteins matching sequences from the remaining 64,142 clusters were categorized into the "medium set", while proteins without matches in the training set were assigned to the "hard set". Proteins that matched both top and remaining clusters were also classified into the "easy set".

6.7.3 Recent set in Swiss-Prot database as Swiss-Prot (NEW)

To evaluate the model's performance on novel, unseen proteins, we utilized the recent release of the Swiss-Prot database (Release 2024_05). By comparing this new release to the previous version (Release 2023_03), we identified the set of proteins exclusive to the new release, which consisted of 2,452 proteins. This Swiss-Prot (NEW) set provides a valuable benchmark for testing the model's ability to generalize to proteins that were not present in the original training data, simulating real-world scenarios where the model encounters newly discovered or updated proteins.

6.8 Quantitative Evaluation Metrics

6.8.1 ProTrek Score and ProTrek pass rate

ProTrek [34] is a tri-modal protein language model designed to enable the retrieval of protein sequences, structures, and textual descriptions from one another (Supplimentary Figure S2 Left). Using ProTrek, we compute the similarity between a given protein sequence P_i and its associated textual description T_i as follows:

$$\text{Similarity}(P_i, T_i) = \frac{\text{Enc}_p(P_i) * \text{Enc}_t(T_i)}{\tau}$$

Here, $\text{Enc}_p(P_i) \in \mathbb{R}^h$ represents the protein encoder, which maps the protein sequence P_i to an h-dimensional embedding. Similarly, $\text{Enc}_t(T_i) \in \mathbb{R}^h$ is the text encoder, which maps the text description T_i into an h-dimensional embedding. $\tau \in \mathbb{R}$ is the temperature coefficient, which controls the scaling of the similarity score to adjust its range and sensitivity.

This formulation allows for the computation of a similarity score by projecting both the protein and textual descriptions into a shared embedding space. The dot product of their embeddings is then normalized by the temperature coefficient τ, yielding a final, scaled similarity score.

To compare the generated responses with ground truth answers, we introduced the ProTrek pass rate, a metric designed to quantify the quality of model-generated answers relative to ground truth. The ProTrek pass rate is defined as the percentage of generated answers whose ProTrek score reaches at least 90% of the ProTrek score of the corresponding ground truth answer.

6.8.2 GPT score

In this study, we selected the *gpt-4o-mini-2024-07-18* model as our expert single-modal large language model (LLM) to serve as the judge for protein analysis tasks (Figure S2 Right). This model was chosen based on its balanced performance in terms of both accuracy and cost-effectiveness. According to the ranking released by [58], it is positioned 8th on the Chatbot Arena LLM Leaderboard on October 6, 2024, highlighting its competitive capabilities in large language model benchmarks. For the purpose of protein analysis, we treat the model's reviewed textual descriptions as reliable factual data. This allows the model to evaluate generated data against these established facts, ultimately providing a quantitative score for the analysis.

Expert-inspected prompts were used to incorporate the generated texts and the facts, ensuring the model could provide accurate and insightful scores (Supplementary Table S7).

We hypothesize that this approach leverages the model's linguistic understanding and factual accuracy to create a robust evaluation framework for protein analysis tasks, where the integration of domain-specific knowledge and machine-generated insights plays a pivotal role in performance assessment.

6.8.3 Effectiveness of ProTrek score and GPT score

To assess the effectiveness and reliability of the ProTrek and GPT scores, we conducted a rigorous evaluation using a sample of 77 protein function descriptions from Swiss-Prot. For each description, we generated five corresponding texts, each representing a different level of relevance to the original description, with decreasing relevance (Appendix A.3). A biology student, with expertise in protein function annotation, was then invited to label the texts based on their relevance to the original descriptions. These labels were treated as gold-standard annotations for comparison purposes.

We employed a variety of metrics to rank the relevance of the five generated texts, including the ProTrek score, GPT score, and traditional text metrics such as BLEU-4 [38], ROUGE-L [37], and METEOR [59]. Spearman's rank correlation coefficients were computed to quantify the relationship between the rankings derived from the different metrics and the gold-standard labels, and the results are summarized in Supplementary Figure S3. Our analysis revealed that both the GPT score and ProTrek score, as well as ROUGE-L, outperformed the other two metrics (BLEU-4 and METEOR) in terms of their ability to reliably rank the texts according to their relevance to the original description. Notably, the difference in performance among these three top-performing metrics (GPT score, ProTrek score, and ROUGE-L) was not statistically significant, suggesting that all three offer comparable efficacy in ranking the relevance of the texts.

To further evaluate the performance and robustness of these metrics, we visualized the pairwise relationships between the metrics using a pairplot, shown in Figure S4. For ROUGE-L, although it was able to distinguish between texts of varying relevance indicated by Supplementary Figure S3, the distribution of

scores appeared somewhat tangled. This created challenges in applying an empirical threshold to effectively differentiate relevant from non-relevant texts, particularly when relevant texts exhibited high semantic similarity but low character-level overlap. In contrast, both the GPT score and ProTrek score showed clear demarcation between relevant and non-relevant texts, with distinct boundaries that suggest superior performance in ranking the relevance of function descriptions. These results indicate that the GPT score and ProTrek score are more robust and reliable than ROUGE-L, especially when dealing with texts that share high semantic similarity but low surface-level similarity.

6.9 Instructional Response Space (IRS)

6.9.1 Enzyme Commission number prediction

The Enzyme Commission (EC) number, a standardized four-digit identifier, is foundational for the systematic classification of enzymes [60]. In this study, we curated a comprehensive and up-to-date dataset of EC numbers from the International Union of Biochemistry and Molecular Biology (IUBMB) Enzyme database (Update 2024_03) [60], encompassing 6,822 entries. The dataset includes 1,933 EC numbers from EC 1, 2,058 from EC 2, 1,349 from EC 3, 763 from EC 4, 319 from EC 5, 243 from EC 6, and 97 from EC 7. To enrich the dataset, we retrieved detailed annotations for each EC number from the KEGG Enzyme database (Update 2024_09) [61]. These annotations span multiple categories, including Name, Class, Sysname, Reaction (IUBMB), Reaction (KEGG), Substrate, Product, Comment, Pathway, Orthology, and Genes. This curated dataset serves as input for generating standardized enzyme descriptions using LLM, employing a structured prompt framework that integrates key categories including EC number, Name, Class, Sysname, Reaction (IUBMB), Substrate, Product, and Comment (prompts in Supplementary Table S8). The descriptions are generated by most advanced text embedding model *gpt-4o-2024-11-20* and the EC numbers, their annotations and generated descriptions are exemplified in Supplementary Table S10. This generated textual EC descriptions serve as the candidate description set for IRS framework.

To provide a comprehensive map of the enzyme universe and rigorously evaluate the performance of Evolla by the Instructional Response Space (IRS) framework, we collected two distinct sets of proteins: enzymes from Swiss-Prot (Release 2023_03) and Swiss-Prot (NEW) (Section 6.7.3). For both datasets, we filtered enzymes with at least one EC number annotation and availability in AlphaFoldDB to leverage the SaProt model. This process yielded 231,428 enzymes from Swiss-Prot and 602 enzymes from the Swiss-Prot (NEW). Each enzyme was input into Evolla with the instruction, "What is the catalytic activity of this protein?" to generate corresponding responses. The generated responses were embedded into a high-dimensional text representational space using OpenAI's advanced *text-embedding-3-large* model. To identify the most relevant EC descriptions, we employed Facebook AI Similarity Search (FAISS) [62], a state-of-the-art tool for efficient similarity search in large datasets. The similarity between response embeddings and EC descriptions was computed using Euclidean distances, producing a ranked list of relevant EC descriptions for each response. To annotate the generated responses and their corresponding enzymes, we applied the

section type	sections
protein level	Function, Miscellaneous, Caution, Catalytic activity, Cofactor, Activity regulation, Biophysicochemical properties, Pathway, Involvement in disease, Allergenic properties, Toxic dose, Pharmaceutical use, Disruption phenotype, Subcellular location, Post-translational modification, Subunit, Domain (non-positional annotation), Sequence similarities, RNA Editing, Tissue specificity, Developmental stage, Induction, Biotechnology, Polymorphism, GO annotation, Proteomes, Protein names, Gene names, Organism, Taxonomic lineage, Virus host
residue level	Active site, Binding site, Site, DNA binding, Natural variant, Mutagenesis, Transmembrane, Topological domain, Intramembrane, Signal peptide, Propeptide, Transit peptide, Chain, Peptide, Modified residue, Lipidation, Glycosylation, Disulfide bond, Cross-link, Domain, Repeat, Compositional bias, Region, Coiled coil, Motif

Supplementary Table S1 The 56 sections collected from the UniProt database include both protein-level and residue-level annotations.

maximum separation strategy proposed by CLEAN [42], which ensures robust classification by maximizing the separation between matched and unmatched candidates.

7 Conclusion

In this work, we present Evolla, an 80B-parameter protein-language generative model that advances our ability to decode and interpret the molecular language of proteins. The key innovation lies in its unprecedented training scale: an AI-synthesized corpus of 546 million protein question-answer pairs encompassing 150 billion word tokens—surpassing existing literature by 2-3 orders of magnitude. This extraordinary scale, combined with our protein-language model architecture and enhanced by Direct Preference Optimization (DPO) and Retrieval-Augmented Generation (RAG), enables Evolla to generate precise, contextually nuanced insights into protein functionality while maintaining scientific accuracy.

By successfully bridging the gap between protein sequences, structures, and their functional understanding through natural language dialogue, Evolla establishes a new paradigm for protein function interpretation at scale. Our work not only contributes to the fields of proteomics and functional genomics but also provides researchers with a powerful tool to unlock the fundamental mysteries encoded in protein molecules, advancing our exploration of life's complexity at the molecular level.

Data availability

The Evolla-10B weights can be downloaded from https://huggingface.co/westlake-repl/Evolla-10B.

Code availability

Evolla is open-sourced under the MIT license. The code repository is available at https://github.com/westlake-repl/Evolla. The Evolla webserver is located at http://www.chat-protein.com/.

Acknowledgement

We would like to extend our heartfelt thanks to the Zhipu BigModel Open Platform for their invaluable support in data generation.

Prompt
You receive a few sentences, each describing the same protein you are analysing. In addition, specific amino acid features within the protein are given, along with detailed spans or specific amino acid sites. The spans are in the form of tuples, represented as (x, y) with integer numbers ranging from 0 to the length of the protein. These values correspond to the start amino acid index x and the end amino acid index y. The sites are in the form of tuples, represented as (A, x, B), where A means the original amino acid, x means the index of the site and B means the mutated amino acid. If A and B are the same amino acid, some important note will be in the site. The single amino acids are represented as (x,), where x means the index of the amino acid. You will play three different roles, and based on the provided protein description, each role will ask a question. Make sure the questions do not overlap and are asked from the perspective of each role. #Role 1#: {0} #Role 2#: {1} #Role 3#: {2} The generated questions should be formatted as follows: Question 1: [Question from Role 1] Question 2: [Question from Role 2] Question 3: [Question from Role 3] When using the information from the caption and sites information, directly analyse the protein, and do not mention that the information source is the caption or the sites.Don't include the name of the protein, refer it as "this protein" '#The Given Description#', 'given description', 'description', 'detailed information' are not allowed to appear in #Question#. #Here is the Given Description#: {} #Questions#:

Supplementary Table S2 Prompt for generation of persona-oriented questions.

Personas	description
clinical biochemist	You are a clinical biochemist who analyzes proteins in biological fluids to diagnose and monitor diseases. Your questions focus on the clinical significance of protein markers, the development of diagnostic assays, and the interpretation of protein profiles in patient samples.
high school student	You are a high school student with a budding interest in biology. Your questions are basic and seek to understand the fundamental concepts of proteins, such as their role in the body, how they are made, and why they are important for health.
undergraduate biology student	You are an undergraduate biology student learning about the structure and function of proteins. Your questions reflect your coursework, including inquiries about protein synthesis, the relationship between amino acids and protein structure, and the basics of protein folding.
postdoctoral fellow	You are a postdoctoral fellow specializing in a particular aspect of protein science. Your questions are highly specialized and may involve complex experimental designs, cutting-edge technologies, and the integration of multiple research disciplines.
medical student	You are a medical student learning about the clinical relevance of proteins. Your questions are about the role of proteins in disease, how protein abnormalities can lead to medical conditions, and the potential for protein-based treatments in healthcare.
structural biologist	You are a structural biologist using techniques like X-ray crystallography and cryo-EM to determine protein structures. Your questions are about the relationship between protein structure and function, the dynamics of protein folding, and the implications of structural data for drug design.
geneticist	You are a geneticist examining the genetic basis of protein function and variation. Your questions explore the inheritance patterns of protein-coding genes, the effects of genetic polymorphisms on protein activity, and the role of proteins in genetic disorders.
synthetic biologist	You are a synthetic biologist designing and engineering proteins for specific applications. Your questions are about protein optimization, the creation of novel protein functions, and the use of synthetic proteins in biotechnology and medicine.
science communicator	You are a science communicator who translates complex protein research for the general public. Your questions are about the best ways to explain protein science in an accessible manner, the use of analogies and metaphors, and the impact of public understanding on scientific progress.
middle school student	You are a middle school student with a curiosity about the natural world. Your questions are very simple and direct, focusing on the basic properties of proteins, how they are related to DNA, and why they are important for living organisms.

Supplementary Table S3 Personas and its descriptions in persona hub.

Evolutions	Prompts
Adding constraints	Please add one more constraints/requirements into #The Given Prompt#
Deepening and expanding	If #The Given Prompt# contains inquiries about certain issues, the depth and breadth of the inquiry can be increased.
Concept specification	Please replace general concepts with more specific concepts.
Multi-step reasoning	If #The Given Prompt# can be solved with just a few simple thinking processes, you can rewrite it to explicitly request multiple-step reasoning.
Generating rare prompts	Please draw inspiration from the #Given Prompt# to create a brand new prompt.This new prompt should belong to the same domain as the #Given Prompt# but be even more rare. The LENGTH and complexity of the #Created Prompt# should be similar to that of the #Given Prompt#.

Supplementary Table S4 Evolutions and their corresponding prompts.

Prompt
I want you act as a Prompt Rewriter. Your objective is to rewrite a given prompt into a more complex version to make those famous AI systems (e.g., chatgpt and GPT4) a bit harder to handle. But the rewritten prompt must be reasonable and must be understood and responded by humans. Your rewriting cannot omit the non-text parts such as the table and code in #The Given Prompt#:. Also, please do not omit the input in #The Given Prompt#. You SHOULD complicate the given prompt using the following methods. Make sure that the rewritten prompts do not overlap: # Method 1#: {0} # Method 2#: {1} # Method 3#: {2} The rewritten prompts should be formatted as follows: Evol_Question 1: [Evol_Question from Method 1] Evol_Question 2: [Evol_Question from Method 2] Evol_Question 3: [Evol_Question from Method 3] You should try your best not to make the #Rewritten Prompt# become verbose. Note that the #Rewritten Prompt# should be different from the #Given prompt#. '#The Given Prompt#', '#Rewritten Prompt#', 'given prompt' and 'rewritten prompt' are not allowed to appear in #Rewritten Prompt#. #The Given Prompt#: {} #Rewritten Prompt#:

Supplementary Table S5 Prompt to evolve the question without providing any protein related information.

Prompt
You are an AI biology assistant that can analyze a single protein. You receive a few sentences, each describing the same protein you are analysing. In addition, specific amino acid features within the protein are given, along with detailed spans or specific amino acid sites. The spans are in the form of tuples, represented as (x, y) with integer numbers ranging from 0 to the length of the protein. These values correspond to the start amino acid index x and the end amino acid index y. The sites are in the form of tuples, represented as (A, x, B), where A means the original amino acid, x means the index of the site and B means the mutated amino acid. If A and B are the same amino acid, some important note will be in the site. The single amino acids are represented as (x,), where x means the index of the amino acid. Using the provided caption and sites information, answer the question related to this protein in a detailed manner. Instead of directly mentioning the sites information, utilize this data to analyse the protein using natural language. Include details like site functions, relation to other proteins, contributions of sites to functions. When using the information from the caption and sites information, directly analyse the protein, and do not mention that the information source is the caption or the sites. Don't include the name of the protein, refer it as "this protein". Always answer as if you are directly analysing the protein. When the question cannot be answered based on the information, donnot mention the information source is the caption or the sites. These are the sentences about the protein. {Information points} {Question}

Supplementary Table S6 Prompt to generate answers with IPs and questions.

Prompt
As an expert biologist, you are assigned to check one paragraph is aligned with facts or not. You will receive some facts, and one paragraph. Score the paragraph between 0 to 100. Give reasons for those aligned ones and not aligned ones. The score should be the format of {"score": score} Here's the facts: {Information points} Here's the paragraph: {paragraph}

Supplementary Table S7 Prompt to calculate GPT scores.

Prompt
Summarize the following text using the specified template format. Ensure the output is formatted exactly as shown in the template. Only output the text and must start with the EC number. Template format: EC 1.4.3.4: The enzyme, Monoamine oxidase, is an oxidoreductase that catalyzes the oxidative deamination of monoamines, which are neurotransmitters and biogenic amines. It utilizes oxygen as an electron acceptor to convert primary, secondary, and some tertiary amines into aldehydes, ammonia, and hydrogen peroxide. The enzyme is a flavoprotein found in the outer mitochondrial membrane and plays a significant role in the metabolism of neurotransmitters such as serotonin, dopamine, and norepinephrine. It differs from other amine oxidases by its ability to oxidize secondary and tertiary amines. Monoamine oxidase is inhibited by specific compounds like clorgyline and pargyline, which makes it distinct from similar enzymes like diamine oxidase. Text information to be summarized: [detailed information of EC number]

Supplementary Table S8 Prompt for EC number description generation.

Prompt
As a biology professor, you will receive a paragraph that describes a protein. You need to understand the content of the paragraph and think carefully about what information the paragraph contains about the protein. Next, you need to generate five paragraphs in order of how much information you contain about the protein. Based on the relevance to this protein, the first paragraph is very relevant, the second is relevant, the third is slightly relevant, the fourth is not related to the protein, and the fifth is not related at all. Here is the paragraph. {paragraph} When generating paragraphs, describe the protein directly, without referring to the original paragraph or confusing the reader.

Supplementary Table S9 Prompt to generate text with different levels of relevance. Five levels of relevance will be generated and presented in order.

EC number	annotations	descriptions
EC 1.1.1.6	EC number: EC 1.1.1.6 Name: glycerol dehydrogenase; glycerin dehydrogenase; NAD+-linked glycerol dehydrogenase Class: Oxidoreductases; Acting on the CH-OH group of donors; With NAD+ or NADP+ as acceptor BRITE hierarchy Sysname: glycerol:NAD+ 2-oxidoreductase Reaction(IUBMB): glycerol + NAD+ = glycerone + NADH + H+ [RN:R01034] Substrate: glycerol [CPD:C00116]; NAD+ [CPD:C00003] Product: glycerone [CPD:C00184]; NADH [CPD:C00004]; H+ [CPD:C00080] Comment: Also acts on propane-1,2-diol.	EC 1.1.1.6: The enzyme, glycerol dehydrogenase, is an oxidoreductase that catalyzes the oxidation of glycerol to glycerone with the concomitant reduction of NAD+ to NADH and the release of H+. It acts on the CH-OH group of donors utilizing NAD+ as an electron acceptor. The enzyme is known for its activity on glycerol and also acts on other substrates like propane-1,2-diol. Its systematic name is glycerol:NAD+ 2-oxidoreductase, and its reaction plays a critical role in the metabolism of glycerol and similar compounds.
EC 3.1.1.2	EC number: EC 3.1.1.2 Name: arylesterase; A-esterase (ambiguous); paraoxonase (ambiguous); aromatic esterase Class: Hydrolases; Acting on ester bonds; Carboxylic-ester hydrolases BRITE hierarchy Sysname: Sysname aryl-ester hydrolase Reaction(IUBMB): a phenyl acetate + H2O = a phenol + acetate [RN:R07342] Substrate: phenyl acetate [CPD:C15583]; H2O [CPD:C00001] Product: phenol [CPD:C15584]; acetate [CPD:C00033] Comment: Acts on many phenolic esters. The reactions of EC 3.1.8.1 aryldialkylphosphatase, were previously attributed to this enzyme. It is likely that the three forms of human paraoxonase are lactonases rather than aromatic esterases [7,8]. The natural substrates of the paraoxonases are lactones [7,8], with (+/-)-5-hydroxy-6E,8Z,11Z,4Z-eicostetraenoic-acid 1,5-lactone being the best substrate [8].	EC 3.1.1.2: The enzyme, Arylesterase, is a hydrolase that acts on ester bonds to catalyze the hydrolysis of phenolic esters into phenols and acetates using water. The reaction involves converting phenyl acetate and water into phenol and acetate. While the enzyme was previously thought to carry out reactions of aryldialkylphosphatase (EC 3.1.8.1), it is now understood that the three forms of human paraoxonase are likely lactonases rather than aromatic esterases. The natural substrates of paraoxonases are lactones, with the best substrate being (+/-)-5-hydroxy-6E,8Z,11Z,4Z-eicosatetraenoic-acid 1,5-lactone.

Supplementary Table S10 Examples of EC number, their annotations and descriptions.

Rank	Text
raw text	Receptor tyrosine kinase which mediates the pleiotropic actions of insulin. Binding of insulin leads to phosphorylation of several intracellular substrates, including, insulin receptor substrates (IRS1, 2, 3, 4), SHC, GAB1, CBL and other signaling intermediates. Each of these phosphorylated proteins serve as docking proteins for other signaling proteins that contain Src-homology-2 domains (SH2 domain) that specifically recognize different phosphotyrosine residues, including the p85 regulatory subunit of PI3K and SHP2. Phosphorylation of IRSs proteins lead to the activation of two main signaling pathways: the PI3K-AKT/PKB pathway, which is responsible for most of the metabolic actions of insulin, and the Ras-MAPK pathway, which regulates expression of some genes and cooperates with the PI3K pathway to control cell growth and differentiation. Binding of the SH2 domains of PI3K to phosphotyrosines on IRS1 leads to the activation of PI3K and the generation of phosphatidylinositol-(3, 4, 5)-triphosphate (PIP3), a lipid second messenger, which activates several PIP3-dependent serine/threonine kinases, such as PDPK1 and subsequently AKT/PKB. The net effect of this pathway is to produce a translocation of the glucose transporter SLC2A4/GLUT4 from cytoplasmic vesicles to the cell membrane to facilitate glucose transport. Moreover, upon insulin stimulation, activated AKT/PKB is responsible for: anti-apoptotic effect of insulin by inducing phosphorylation of BAD; regulates the expression of gluconeogenic and lipogenic enzymes by controlling the activity of the winged helix or forkhead (FOX) class of transcription factors. Another pathway regulated by PI3K-AKT/PKB activation is mTORC1 signaling pathway which regulates cell growth and metabolism and integrates signals from insulin. AKT mediates insulin-stimulated protein synthesis by phosphorylating TSC2 thereby activating mTORC1 pathway. The Ras/RAF/MAP2K/MAPK pathway is mainly involved in mediating cell growth, survival and cellular differentiation of insulin. Phosphorylated IRS1 recruits GRB2/SOS complex, which triggers the activation of the Ras/RAF/MAP2K/MAPK pathway. In addition to binding insulin, the insulin receptor can bind insulin-like growth factors (IGFI and IGFII). When present in a hybrid receptor with IGF1R, binds IGF1 (By similarity). In adipocytes, inhibits lipolysis (PubMed:27322061)
Very relevant	The insulin receptor is a receptor tyrosine kinase that mediates the pleiotropic actions of insulin. Upon insulin binding, it phosphorylates several intracellular substrates, including insulin receptor substrates (IRS1-4), SHC, GAB1, and CBL. These phosphorylated proteins serve as docking sites for signaling molecules with Src-homology-2 (SH2) domains, such as the p85 regulatory subunit of PI3K and SHP2. Activation of IRS proteins leads to two main signaling pathways: the PI3K-AKT/PKB pathway, which facilitates metabolic actions like glucose transport by translocating GLUT4 to the cell membrane, and the Ras-MAPK pathway, which regulates gene expression, cell growth, and differentiation. In adipocytes, the insulin receptor inhibits lipolysis, promoting lipid storage.
relevant	The insulin receptor plays a crucial role in glucose homeostasis by activating signaling pathways upon insulin binding. It triggers the PI3K-AKT/PKB pathway, leading to increased glucose uptake in cells, and the Ras-MAPK pathway, influencing cell growth and gene expression. Additionally, it can bind insulin-like growth factors when forming hybrid receptors with IGF1R.
Slightly relevant	The insulin-like growth factor 1 receptor (IGF1R) is a receptor tyrosine kinase that binds IGF1 and IGF2, playing a significant role in growth and development. It shares structural similarities with the insulin receptor and can form hybrid receptors that influence cellular proliferation and differentiation.
Not related to the protein	Hemoglobin is a protein found in red blood cells responsible for transporting oxygen from the lungs to tissues throughout the body. It consists of four subunits, each containing an iron-bound heme group that binds oxygen molecules, facilitating efficient gas exchange.
Not related at all	The Eiffel Tower is a wrought-iron lattice tower located on the Champ de Mars in Paris, France. Constructed for the 1889 Exposition Universelle, it has become an iconic symbol of France and a globally recognized architectural landmark.

Supplementary Table S11 An example of the correlation levels related to the raw text. The correlations are very relevant, relevant, slightly relevant, not relevant to this protein, not relevant at all.

	Ndata	ALL			Hard		
		ProTrek Score	ProTrek Pass Rate	GPT Score	ProTrek Score	ProTrek Pass Rate	GPT Score
Evolla-10B (sp-large)	20M	15.029/15.074	84.26%	72.411	12.224/12.747	61.88%	49.383
+GPT Preference	1M	15.027/15.074	84.07%	78.892	12.316/12.747	63.47%	52.444

Supplementary Table S12 Performance comparison of Evolla-10B (sp-large) model and its fine-tuned version by GPT preference data.

	Ndata	ALL			Hard		
		ProTrek Score	ProTrek Pass Rate	GPT Score	ProTrek Score	ProTrek Pass Rate	GPT Score
Evolla-10B (sp-large)	20M	15.029/15.074	84.26%	72.410	12.224/12.747	61.88%	49.382
+DQS	-	15.597/15.074	90.02%	76.27	13.640/12.747	79.64%	57.953
+QGS	-	15.244/15.074	87.39%	76.89	12.439/12.747	67.47%	58.286

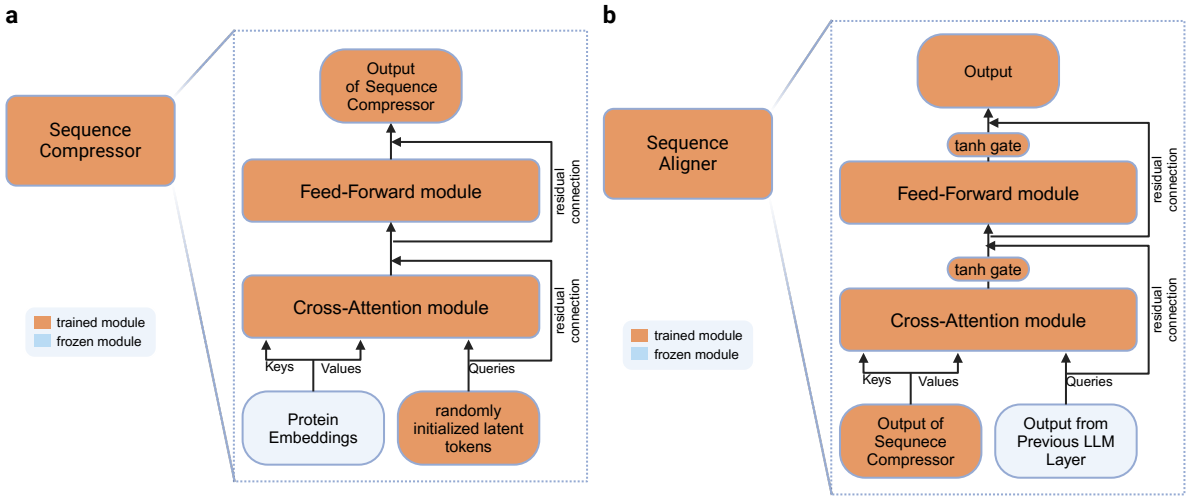
Supplementary Table S13 Comparison of DQS and QGS methods on Evolla-10B (sp-large).

Appendix A Appendix

A.1 Detailed performance of DPO and RAG

A.1.1 Testing Result of DPO

The selected checkpoint was evaluated on the test set. The evaluation used the two metrics to measure the model's ability to generate high-quality and biologically meaningful responses (GPT scores for accuracy and reliability, and ProTrek scores for biological relevance).



Supplementary Figure S1 Sequence Compressor (a) and Sequence Aligner (b).

Table S12 presents an analysis of alignment with GPT Preference for the Evolla-10B (sp-large) model, evaluated using metrics including ProTrek Score, ProTrek Pass Rate, and GPT Score, on the ALL and Hard test sets.

For GPT Preference, the Evolla-10B (sp-large) model exhibits substantial gains in GPT Score on both the ALL and Hard test sets. On the ALL set, the GPT Score improves from 72.411 to 78.892, while the ProTrek Score (15.027/15.074) and ProTrek Pass Rate (84.07%) remain stable. On the Hard set, GPT Preference increases the GPT Score from 49.383 to 52.444, alongside a slight improvement in ProTrek Pass Rate (61.88% to 63.47%) and ProTrek Score (12.224/12.747 to 12.316/12.747).

In summary, GPT Preference leads to consistent improvements in GPT Scores for the Evolla-10B (sp-large) in both All and Hard set, while maintaining stable ProTrek Scores. These findings demonstrate the effectiveness of preference-based fine-tuning in improving alignment with specific evaluation criteria.

A.1.2 Testing Result of RAG

The prompt template for similar protein descriptions is provided in Table S16. These methods are designed to improve retrieval performance by utilizing different retrieval strategies, as summarized in Table S13.

Table S13 presents a comparative analysis of the DQS and QGS methods for the Evolla-10B (sp-large) model. The evaluation is based on several metrics, including ProTrek Score, ProTrek Pass Rate, and GPT Score, which are assessed on both the ALL and Hard test sets.

For the Evolla-10B (sp-large) model, incorporating DQS results in significant improvements in both ProTrek Score (from 15.029 to 15.597) and ProTrek Pass Rate (from 84.26% to 90.02%), along with an increase in GPT Score (from 72.410 to 76.27). On the Hard test set, DQS shows a notable boost, with ProTrek Score rising from 12.224 to 13.640 and GPT Score improving from 49.382 to 57.953. However, QGS outperforms DQS in certain aspects, particularly in GPT Score, where it reaches 76.89 on the ALL set, surpassing DQS's 76.27. While QGS shows more modest improvements in ProTrek Pass Rate, its superior GPT Score suggests that QA-driven sequence retrieval improves the relevance of the retrieved sequences.

Database	N_{IP}	N_{IP}^{w}	N_{IP}^{wo}	Ratio($N_{\text{IP}}^{w}/N_{\text{IP}}$)
Swiss-Prot	8,189,139	378,943	7,810,196	4.63%
TrEMBL	1,420,812,825	1,418,087,919	2,724,906	99.81%

Supplementary Table S14 Proportion of automatically tagged information points among assertion-tagged points. N_{IP} represents the number of IPs with assertions, N_{IP}^{w} represents the number of IPs with automatic assertions, and N_{IP}^{wo} represents the number of IPs without automatic assertions.

Overall, both DQS and QGS show improvements in retrieval performance, especially on the Hard set. DQS leads to stronger improvements in ProTrek Pass Rate, indicating better alignment with the target proteins. On the other hand, QGS excels in GPT Score, demonstrating the effectiveness of QA-driven retrieval in improving the accuracy and relevance of the generated answers.

A.2 Automatically annotated TrEMBL

In the UniProt database, most IPs, including those from Swiss-Prot and TrEMBL, are associated with "evidence tags" that describe the source of the information, such as published scientific experiments, orthologous proteins, or other databases. Each piece of evidence is formally classified by a code from the Evidence and Conclusion Ontology (ECO) [63], which ensures standardized documentation.

We also collected all automatic assertions from ECO in order to confirm the assumption, focusing on subclasses of ECO_0000501, which represents "evidence used in automatic assertion". These subclasses formed a comprehensive set of all automatic assertions. Using this set, we analyzed the proportion of automatically versus non-automatically labeled information points in Swiss-Prot and TrEMBL, as shown in Table S14. Only 4.63% of the 8,189,139 evidence-labeled information points in Swiss-Prot were automatically annotated, while 99.81% of the 1,420,812,825 evidence-labeled information points in TrEMBL were automatically annotated.

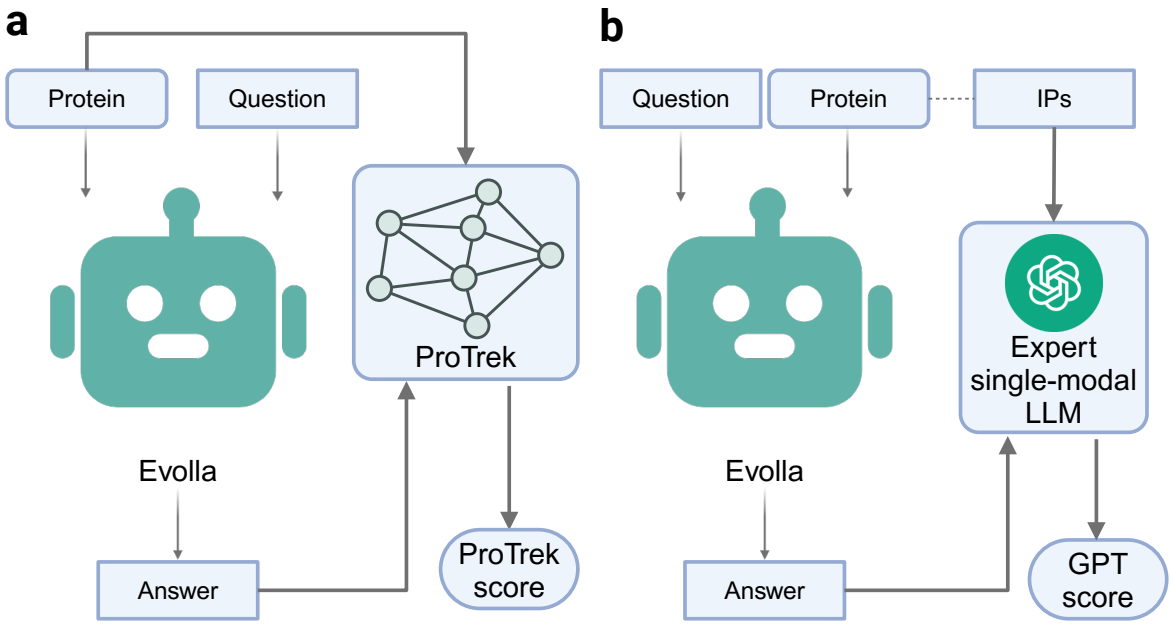
This high proportion of automatic annotations in TrEMBL indicates that most of its data lacks manual curation. As a result, training a model directly on TrEMBL's information points risks the model learning incorrect or biased patterns due to potential inconsistencies in the automated annotations. This can lead to a decline in the model's performance when applied to more rigorously curated datasets, such as those from Swiss-Prot.

A.3 Performance of GPT score and ProTrek score

Although natural language processing has had a significant impact on many areas, open-ended text evaluation still faces considerable challenges, particularly due to the limitations of text metrics based on character-level comparisons [64]. These limitations arise from the inability of such metrics to fully capture the semantic richness and contextual nuances of open-ended text.

To address these issues, we propose to use GPT score and ProTrek score as complementary solution.The GPT score can effectively evaluate the relevance and quality of open-ended text, while the ProTrek score provides a robust method for comparing protein sequences with textual descriptions.

We randomly sampled descriptions of 77 proteins from SwissProt functions, each containing more than 200 words to ensure comprehensive coverage of the protein's functional details. For each description, we



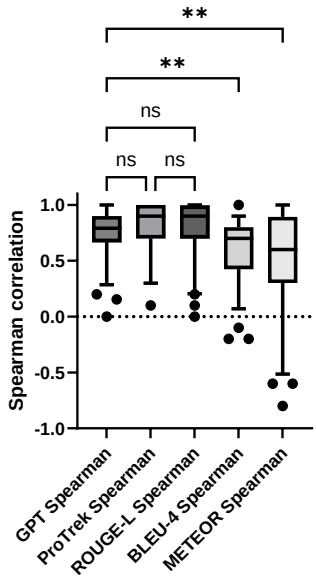
Supplementary Figure S2 ProTrek score pipeline (a) and GPT score pipeline (b).

used the state-of-the-art large language model, *o1-preview*, which ranks first on the Chatbot Arena LLM Leaderboard as of October 6, 2024, to rewrite the text. Five variations of the text were generated, each with decreasing relevance to the original description. The prompt used for this generation process is detailed in Table S9, and an example of these variations is shown in Table S11.

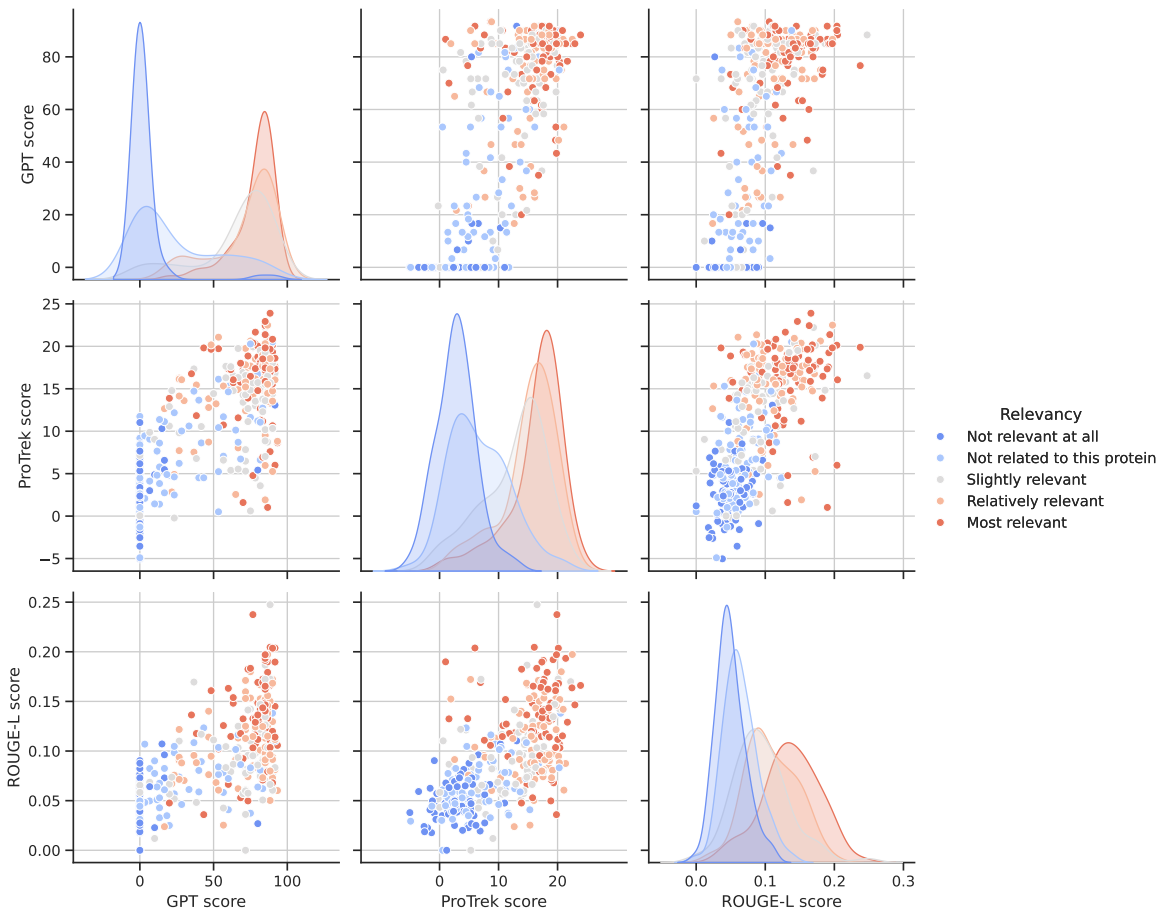
A biology student was invited to evaluate the relevance of five rewritten sentence variants, and the human ranking results were treated as the gold standard. The evaluation involved presenting the student with an original sentence alongside five rewritten versions, which he ranked based on their relevance to the original. Categories included "Very relevant," "Relevant," "Slightly relevant," "Not related to this protein," and "Not relevant at all". For analytical purposes, we consolidated "Very relevant," "Relevant," and "Slightly relevant" into a single "Relevant" category, while "Not related to this protein" and "Not relevant at all" were grouped as "Irrelevant".

To quantify similarity between each rewritten sentence and the original, we employed both GPT and ProTrek scores, alongside traditional metrics such as BLEU-4, ROUGE-L, and METEOR. To ensure a fair assessment of semantic similarity, original descriptions were omitted from the information points during evaluation. We then calculated Spearman correlation coefficients between each metric ranking and the human ranking, shown in Figure S3. Results indicated that ROUGE-L, GPT score, and ProTrek score demonstrated similar and notably high spearman correlations.

Further analysis focused on the three best metrics (ROUGE-L, GPT score, and ProTrek score) in Figure S4, revealing that GPT and ProTrek scores effectively distinguished between relevant and irrelevant pairs. These findings underscore the utility of GPT and ProTrek scores in reliably assessing semantic relevance, providing a robust framework for relevance classification in protein-related text analysis.



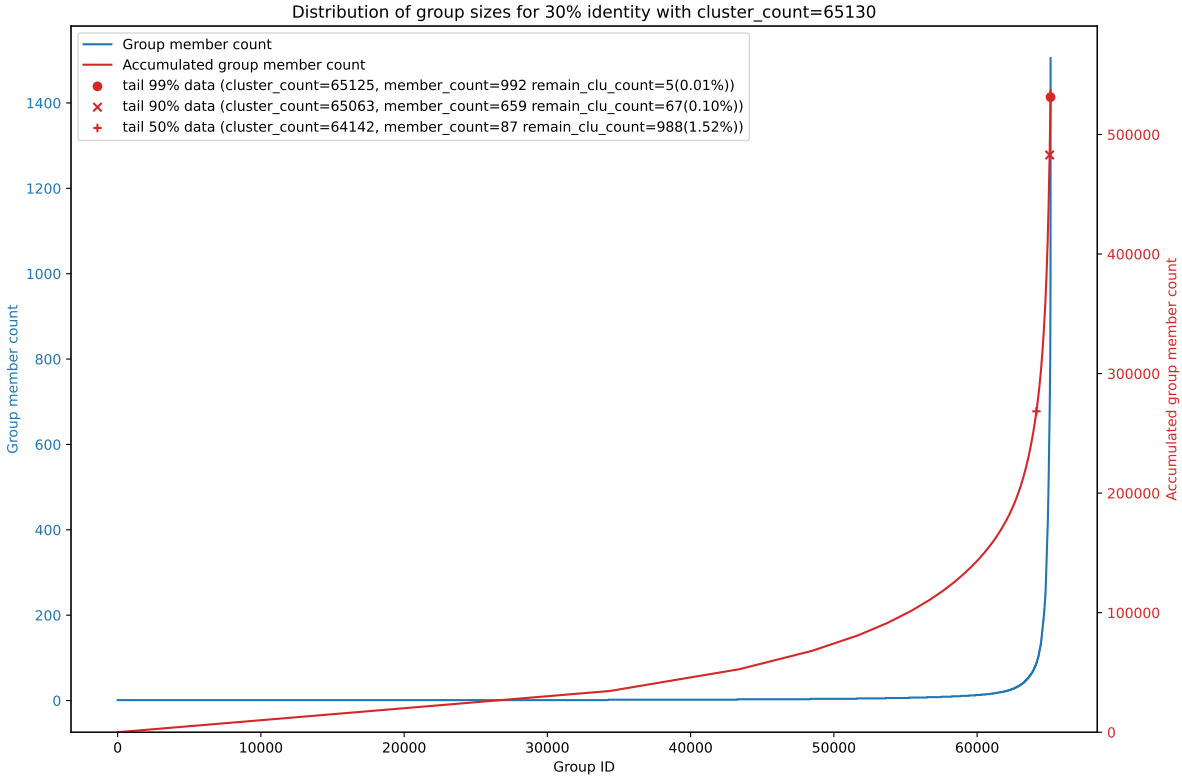
Supplementary Figure S3 Barplot of spearman correlation among GPT score, ProTrek score, ROUGE-L, BLEU-4 and METEOR.



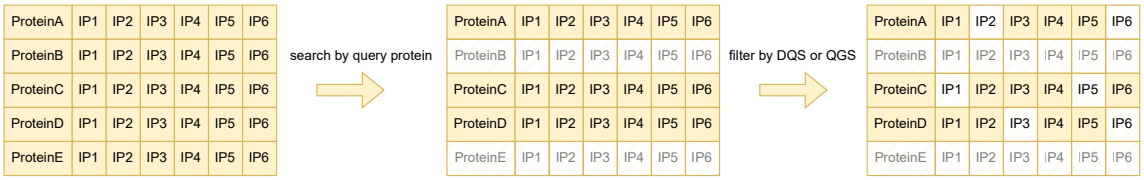
Supplementary Figure S4 Pairplot of distributions for GPT score, ProTrek score and ROUGE-L.

References

[1] Anfinsen, C. B. The formation and stabilization of protein structure. *Biochemical Journal* **128**, 737 (1972).



Supplementary Figure S5 Cluster plot in training set of Swiss-Prot database (Release 2023_03).



Supplementary Figure S6 Overall pipeline of RAG strategy.

[2] Abramson, J. *et al.* Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature* 1–3 (2024).

[3] Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *nature* **596**, 583–589 (2021).

[4] Senior, A. W. *et al.* Improved protein structure prediction using potentials from deep learning. *Nature* **577**, 706–710 (2020).

[5] Uniprot: the universal protein knowledgebase in 2021. *Nucleic acids research* **49**, D480–D489 (2021).

[6] Altschul, S. F. *et al.* Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research* **25**, 3389–3402 (1997).

[7] Van Kempen, M. *et al.* Fast and accurate protein structure search with foldseek. *Nature Biotechnology* **42**, 243–246 (2024).

[8] Zhang, Y. & Skolnick, J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research* **33**, 2302–2309 (2005).

As an advanced language model with extensive training and knowledge, I would like you to perform a comprehensive analysis of the following amino acid sequence. Please consider the following aspects in your analysis: - Protein Structure Prediction: Provide insights into the possible secondary and tertiary structures that this amino acid sequence might form. - Functional Domains: Identify any known functional domains or motifs within the sequence that could indicate the protein's function. - Post-Translational Modifications: Highlight potential sites for post-translational modifications and discuss their possible effects on protein activity. - Biological Pathways: Explain how this protein might interact within specific biological pathways or processes. - Disease Associations: Discuss any known associations between this amino acid sequence or similar sequences and particular diseases or medical conditions. - Evolutionary Conservation: Analyze the evolutionary conservation of this sequence across different species to infer its importance. - Potential Interactions: Suggest possible interactions with other proteins or molecules based on the sequence characteristics. - Please ensure that your analysis is detailed and supported by relevant biological concepts and knowledge. Additional Requirement: After the summary of your analysis, please answer the specific question I will provide. **Amino Acid Sequence:** {sequence} **Question:** {question}
As an advanced language model with extensive training and knowledge, please perform a thorough analysis of the following amino acid sequence. Your analysis should cover the following aspects: - Protein Structure Prediction: Discuss the potential secondary and tertiary structures that this amino acid sequence may form. - Functional Domains: Identify any known functional domains or motifs within the sequence that could suggest the protein's function. - Post-Translational Modifications: Highlight possible sites for post-translational modifications and explain their potential impact on protein activity. - Biological Pathways: Describe how this protein might interact within specific biological pathways or processes. - Disease Associations: Explore any known links between this amino acid sequence or similar sequences and particular diseases or medical conditions. - Evolutionary Conservation: Assess the evolutionary conservation of this sequence across various species to determine its significance. - Potential Interactions: Suggest possible interactions with other proteins or molecules based on the sequence characteristics. Additional Requirement: After summarizing your analysis, please answer the specific question I will provide. **Amino Acid Sequence:** {sequence} **Question:** {question}
Please act as a highly trained language model with extensive knowledge and conduct a detailed examination of the following amino acid sequence. Address the following components in your analysis: - Protein Structure Prediction: Provide insights into the possible secondary and tertiary structures formed by this amino acid sequence. - Functional Domains: Detect any known functional domains or motifs within the sequence that may indicate the protein's role. - Post-Translational Modifications: Identify potential sites for post-translational modifications and discuss how they might affect protein function. - Biological Pathways: Explain the potential interactions of this protein within specific biological pathways or processes. - Disease Associations: Discuss any associations between this amino acid sequence or similar sequences and particular diseases or health conditions. - Evolutionary Conservation: Evaluate the evolutionary conservation of this sequence across different species to infer its importance. - Potential Interactions: Recommend possible interactions with other proteins or molecules based on the sequence features. Extra Requirement: Following your analysis summary, please respond to a specific question I will pose. **Amino Acid Sequence:** {sequence} **Question:** {question}

Supplementary Table S15 Three representative prompts for general purpose large language models to answer specific questions based on protein informations.

[9] Bileschi, M. L. *et al.* Using deep learning to annotate the protein universe. *Nature Biotechnology* **40**, 932–937 (2022).

[10] Zhou, N. *et al.* The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. *Genome biology* **20**, 1–23 (2019).

Similar proteins:

Entry: B7IA77
GO annotation: With respect to biological process, the GO term involving glycyl-tRNA aminoacylation encompasses this protein.
Catalytic activity: The action of this protein is essential for the reaction: ATP + glycine + tRNA(Gly) = AMP + diphosphate + glycyl-tRNA(Gly).
Protein names: Within its field, this protein is catalogued as GlyRS.
GO annotation: The GO term for this protein integrating molecular function includes glycine-tRNA ligase activity.
Similar proteins: Glycine–tRNA ligase beta subunit, Glycine–tRNA synthetase alpha subunit.

Entry: Q6LW15
Protein names: GlyRS is the given name for this protein.
Catalytic activity: In the reaction: ATP + glycine + tRNA(Gly) = AMP + diphosphate + glycyl-tRNA(Gly), this protein serves a catalytic function.
GO annotation: Regarding biological process, the gene ontology label assigned to this protein encapsulates glycyl-tRNA aminoacylation.
GO annotation: With correlation to molecular function, the GO term designated for this protein includes glycine-tRNA ligase activity.
Similar proteins: Glycine-tRNA ligase from Homo sapiens, Glycine–tRNA ligase delta subunit.

Entry: Q12TC1
GO annotation: The GO term for this protein involving biological process incorporates glycyl-tRNA aminoacylation.
Catalytic activity: The reaction: ATP + glycine + tRNA(Gly) = AMP + diphosphate + glycyl-tRNA(Gly) is expedited by the enzymatic action of this protein.
GO annotation: As regards molecular function, the GO terminology for this protein envelopes glycine-tRNA ligase activity.
Protein names: This protein goes under the identification of GlyRS.
Similar proteins: Glycine-tRNA ligase gamma subunit, Glycine–tRNA synthetase epsilon subunit.

Question: Which enzymatic operation does this protein specifically carry out?

Supplementary Table S16 RAG prompt for protein B4T945, consisting of three retrieved similar proteins and four relevant descriptions formatted as subsection name followed by description. The prompt precedes the question.

[11] Radivojac, P. *et al.* A large-scale evaluation of computational protein function prediction. *Nature methods* **10**, 221–227 (2013).

[12] Su, J. *et al.* Saprot: Protein language modeling with structure-aware vocabulary (2024). URL https://openreview.net/forum?id=6MRm3G4NiU.

[13] Su, J. *et al.* Saprothub: Making protein modeling accessible to all biologists. *bioRxiv* 2024–05 (2024).

[14] Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).

[15] Hayes, T. *et al.* Simulating 500 million years of evolution with a language model. *bioRxiv* 2024–07 (2024).

[16] Elnaggar, A. *et al.* Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* **44**, 7112–7127 (2021).

[17] Hu, M. *et al.* Exploring evolution-aware &-free protein language models as protein function predictors. *Advances in Neural Information Processing Systems* **35**, 38873–38884 (2022).

[18] He, Y. *et al.* Protein language models-assisted optimization of a uracil-n-glycosylase variant enables programmable t-to-g and t-to-c base editing. *Molecular Cell* **84**, 1257–1270 (2024).

[19] Abdine, H., Chatzianastasis, M., Bouyioukos, C. & Vazirgiannis, M. Prot2text: Multimodal protein's function generation with gnns and transformers **38**, 10757–10765 (2024).

[20] Lv, L. *et al.* Prollama: A protein large language model for multi-task protein language processing. *arXiv e-prints* arXiv–2402 (2024).

[21] Luo, Y. *et al.* Biomedgpt: An open multimodal large language model for biomedicine. *IEEE Journal of Biomedical and Health Informatics* (2024).

[22] Wang, C., Fan, H., Quan, R. & Yang, Y. Protchatgpt: Towards understanding proteins with large language models. *arXiv preprint arXiv:2402.09649* (2024).

[23] Char, S., Corley, N., Alamdari, S., Yang, K. K. & Amini, A. P. Protnote: a multimodal method for protein-function annotation. *bioRxiv* 2024–10 (2024).

[24] Liu, N. *et al.* Evollama: Enhancing llms' understanding of proteins via multimodal structure and sequence representations. *arXiv preprint arXiv:2412.11618* (2024).

[25] Fan, W. *et al.* A survey on rag meeting llms: Towards retrieval-augmented large language models 6491–6501 (2024).

[26] Rafailov, R. *et al.* Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* **36** (2024).

[27] Alayrac, J.-B. *et al.* Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems* **35**, 23716–23736 (2022).

[28] Liu, H., Li, C., Wu, Q. & Lee, Y. J. Visual instruction tuning. *Advances in neural information processing systems* **36** (2024).

[29] Li, J., Li, D., Savarese, S. & Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models 19730–19742 (2023).

[30] Rao, R. M. *et al.* Msa transformer 8844–8856 (2021).

[31] Dubey, A. *et al.* The llama 3 herd of models. *arXiv preprint arXiv:2407.21783* (2024).

[32] Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems* (2017).

[33] He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition 770–778 (2016).

[34] Su, J., Zhou, X., Zhang, X. & Yuan, F. Protrek: Navigating the protein universe through tri-modal contrastive learning. *bioRxiv* 2024–05 (2024).

[35] Ge, T. *et al.* Scaling synthetic data creation with 1,000,000,000 personas. *arXiv preprint arXiv:2406.20094* (2024).

[36] Xu, C. *et al.* Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244* (2023).

[37] Lin, C.-Y. Rouge: A package for automatic evaluation of summaries 74–81 (2004).

[38] Papineni, K., Roukos, S., Ward, T. & Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation 311–318 (2002).

[39] OpenAI. New embedding models and api updates (2023). URL https://openai.com/index/new-embedding-models-and-api-updates/. Accessed: 2024-12-30.

[40] Van der Maaten, L. & Hinton, G. Visualizing data using t-sne. *Journal of machine learning research* **9** (2008).

[41] McInnes, L., Healy, J. & Melville, J. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426* (2018).

[42] Yu, T. *et al.* Enzyme function prediction using contrastive learning. *Science* **379**, 1358–1363 (2023).

[43] Varadi, M. *et al.* Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research* **50**, D439–D444 (2022).

[44] Zhang, S. *et al.* Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068* (2022).

[45] Bhattacharya, M., Pal, S., Chatterjee, S., Lee, S.-S. & Chakraborty, C. Large language model to multimodal large language model: A journey to shape the biological macromolecules to biological sciences and medicine. *Molecular Therapy Nucleic Acids* **35** (2024). URL https://doi.org/10.1016/j.omtn.2024.102255.

[46] Ouyang, L. *et al.* Training language models to follow instructions with human feedback. *Advances in neural information processing systems* **35**, 27730–27744 (2022).

[47] Lee, H. *et al.* Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback (2024). URL https://arxiv.org/abs/2309.00267. 2309.00267.

[48] Huang, L. *et al.* A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Transactions on Information Systems* (2023).

[49] Rasley, J., Rajbhandari, S., Ruwase, O. & He, Y. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters 3505–3506 (2020).

[50] Loshchilov, I., Hutter, F. *et al.* Fixing weight decay regularization in adam. *arXiv preprint arXiv:1711.05101* **5** (2017).

[51] Micikevicius, P. *et al.* Mixed precision training. *arXiv preprint arXiv:1710.03740* (2017).

[52] Loshchilov, I. & Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[53] Zhao, Y. *et al.* Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277* (2023).

[54] Gasteiger, E., Jung, E. & Bairoch, A. Swiss-prot: connecting biomolecular knowledge via a protein database. *Current issues in molecular biology* **3**, 47–55 (2001).

[55] GLM, T. *et al.* Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793* (2024).

[56] Madden, T. The blast sequence analysis tool. *The NCBI handbook* **2**, 425–436 (2013).

[57] Steinegger, M. & Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* **35**, 1026–1028 (2017).

[58] Zheng, L. *et al.* Judging llm-as-a-judge with mt-bench and chatbot arena (2023). 2306.05685.

[59] Banerjee, S. & Lavie, A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments 65–72 (2005).

[60] NC-IUBMB. Enzyme nomenclature: Recommendations of the nomenclature committee of the international union of biochemistry and molecular biology on the nomenclature and classification of enzymes by the reactions they catalyse (2024). URL https://iubmb.qmul.ac.uk/enzyme/.

[61] Kanehisa, M., Furumichi, M., Tanabe, M., Sato, Y. & Morishima, K. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic acids research* **45**, D353–D361 (2017).

[62] Douze, M. *et al.* The faiss library (2024). URL https://arxiv.org/abs/2401.08281. 2401.08281.

[63] Nadendla, S. *et al.* Eco: the evidence and conclusion ontology, an update for 2022. *Nucleic Acids Research* **50**, D1515–D1521 (2022).

[64] Novikova, J., Dušek, O., Curry, A. C. & Rieser, V. Why we need new evaluation metrics for nlg. *arXiv preprint arXiv:1707.06875* (2017).