ProteinRAP: Constructing Retrieval Augmented Prompts to Assist Large Language Models in Protein Understanding

Anonymous ACL submission

Abstract

Large language models (LLMs) have demonstrated remarkable success in Natural Language Processing (NLP), primarily due to their emergent abilities derived from extensive pre-training. These pre-trained LLMs can handle numerous tasks without additional supervised fine-tuning, facilitating their transfer to various problems. However, when applied to the "language of life"—proteins, LLMs often fall short in capturing the complex relationships between amino acid sequences and their functions, resulting in suboptimal performance in related tasks. To address this issue, this study introduces **ProteinRAP**, a novel method leveraging Retrieval-Augmented Prompts (RAPs) to enhance LLM performance on protein tasks without extensive retraining. ProteinRAP comprises Protein-Text CLIP, which utilizes contrastive learning for cross-modal retrieval, and an optimized prompt learning strategy. Through RAP construction, LLMs exhibit significant improvements in protein understanding. Evaluations on both general and protein-specific LLMs in protein understanding tasks highlight existing methods' limitations. ProteinRAP markedly boosts performance, achieving up to 87.7% improvement over general LLMs and matching state-of-the-art results without additional training.

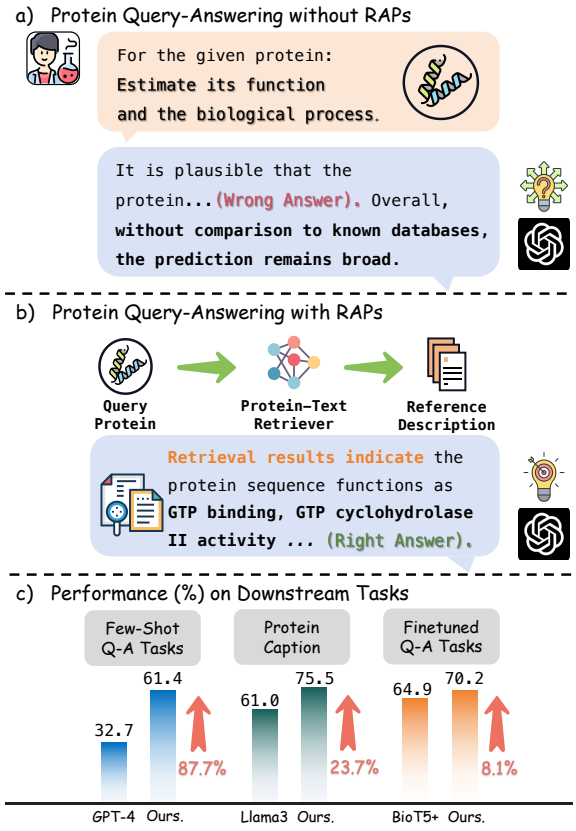

Figure 1: (a) General LLMs face challenges in protein understanding tasks. (b) Retrieval mechanisms enable LLMs to produce accurate answers. (c) Retrieval-augmented approaches achieve significant performance improvements across diverse tasks.

1 Introduction

In recent years, pre-trained large language models such as GPT4 (Achiam et al., 2023), Llama3 (Dubey et al., 2024), Qwen (Bai et al., 2023), and Deepseek (Liu et al., 2024a) have emerged as a new paradigm in the field of natural language processing (NLP). (Zhang et al., 2023; Chang et al., 2024) This shift is largely due to their remarkable performance on few-shot and zero-shot tasks (Wei et al., 2022; Kojima et al., 2022). The underlying mechanism enabling this capability is the models' ability to perform in-context learning from specific prompts. (Brown et al., 2020) By providing predefined instructions and question formats as input, these models can infer and provide answers to tasks with zero or few samples without the need for parameter updates.

In the biological domain, and particularly in protein science, pre-trained LLMs have shown suboptimal performance in few-shot and zero-shot tasks (Tan et al., 2024). While proteins can be represented as sequences of amino acids, LLMs struggle to capture the relationship between these sequences and their biological functions due to the

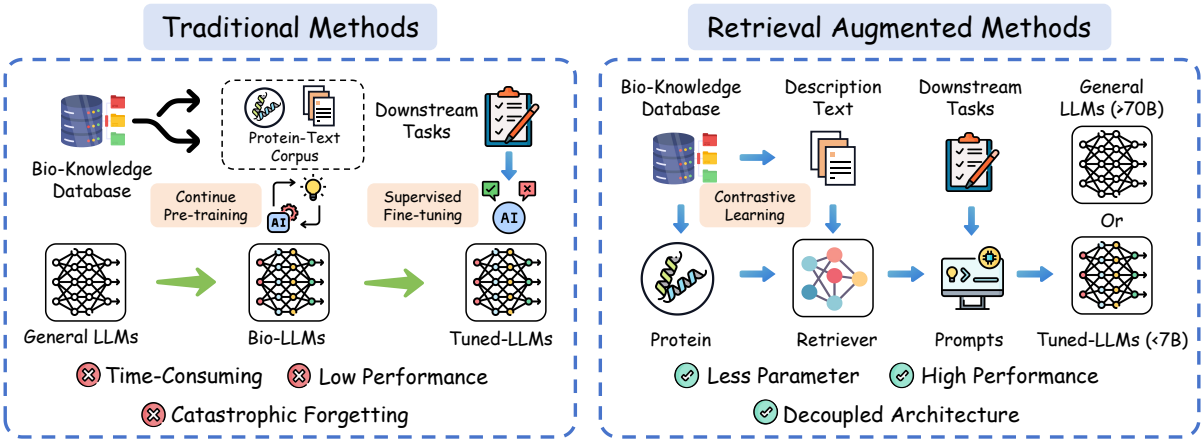

Figure 2: A comparison of retrieval-augmented methods and traditional approaches. Traditional methods re-train LLMs on protein sequences, whereas retrieval-augmented approaches leverage contrastive learning to train a retriever. By injecting retrieved knowledge into prompts, the retrieval-augmented method boosts LLM performance on protein-related tasks.

structural differences between protein sequences and natural language. To address this issue, various protein-specific models have been developed, such as ESM (Hayes et al., 2025), Galactica (Taylor et al., 2022), ProtTrans (Elnaggar et al., 2021), ProteinBERT (Brandes et al., 2022), and ProGen2 (Nijkamp et al., 2023), which integrate protein sequences in their pre-training datasets. Though these models excel in protein property prediction and design, they fail to process natural language instructions effectively. Alternative approaches involve continued pre-training and supervised fine-tuning using protein databases (Fang et al., 2024), or employ protein encoders and cross-modal projectors for alignment (Liu et al., 2024c; Wang et al., 2024; Liu et al., 2024b). Despite mitigating some issues, these methods require significant computational resources as LLM parameters grow, and suffer from challenges such as catastrophic forgetting (Wu et al., 2024b; Luo et al., 2023), where the model's original domain performance declines, and adaptability issues requiring parameter updates per task (Zhao et al., 2024).

Leveraging evolutionary insights that homologous proteins tend to perform similar functions (Hilbert et al., 1993), we propose a retrieval-enhanced prompt technique to enhance LLM performance on protein-related tasks. Our approach uses contrastive learning to develop a protein-text multi-modal retriever, called Protein-Text CLIP. This model retrieves similar samples from protein databases to construct Retrieval-Augmented Prompts (RAPs). Our experiments demonstrate that RAPs significantly improve LLM performance across various scales.

In an evolutionary context, similar proteins are often homologous and frequently perform similar functions in the life sciences (Hilbert et al., 1993). This insight prompts the use of alignment and retrieval approaches to accomplish protein understanding tasks. Compared to traditional retrieval augmentation methods, protein retrieval augmentation involves two distinctly different modalities: protein FASTA sequences and textual annotations. Existing methods predominantly rely on sequence alignment or retrieval techniques for protein attribute prediction (Ma et al., 2023), rather than addressing open-ended questions such as protein instruction-based querying (Fang et al., 2024).

Based on all the above, in this study, we introduce ProteinRAP, a method using retrieval-enhanced prompts to enhance LLM capabilities in protein understanding tasks. Firstly, we develop the Protein-Text CLIP model, leveraging contrastive learning for cross-modal retrieval. For different downstream tasks, this model retrieves similar samples from the corresponding protein database and then constructs retrieval augmented prompts (RAPs). RAPs are then used in LLMs through in-context learning, integrating retrieved annotations with the query sequence to enhance task performance. Downstream experiments showed that this approach significantly improves LLMs' prediction accuracy across various protein tasks without requiring further model training. The contributions of this work can be summarized as follows:

1. We conduct a comprehensive evaluation of general LLMs and mixed protein-text LLMs on protein captioning and understanding tasks. Our analysis highlights the significant disadvantage of existing methods, particularly in the protein-text generation domain, underscoring the need for more targeted approaches.

2. We propose a novel paradigm named ProteinRAP, which includes the development of an efficient protein-text retriever. This method is the first to employ retrieval-augmented techniques for open-ended answer generation in protein-related tasks. Furthermore, we design specialized prompts tailored for protein tasks and conduct exhaustive evaluations and ablation studies on the retrieval method. This advances the development of retrieval-enhanced approaches in the protein domain substantially.

3. Our findings demonstrate remarkable improvements in various tasks, achieving an 87.7% improvement on general-purpose LLMs and a 23.7% increase in the protein caption over the previous state-of-the-art (SOTA) method, and the protein understanding task sees an 8.1% improvement. Notably, the RAP-based methodology achieves results comparable to SOTA models in a training-free manner, highlighting its efficacy and practical applicability.

2 Related Works

This section provides an overview of research efforts in three interconnected domains: protein language modeling, protein-text cross-modal learning, and prompt engineering techniques.

2.1 Protein Language Models (PLMs)

Protein language models (PLMs) leverage the success of Transformers in NLP to represent protein sequences as biological languages. **Encoder-Based Models** (Hayes et al., 2025; Brandes et al., 2022; Elnaggar et al., 2021; Cao and Shen, 2021) extraction of protein sequence and structural features using bidirectional attention, **Decoder-Based Models** (Madani et al., 2023; Nijkamp et al., 2023; Lv et al., 2024; Ferruz et al., 2022) focus on protein sequence generation. **Encoder-Decoder Models** (Chen et al., 2024; Elnaggar et al., 2021) broadened the scope with large-scale pre-training. These models have achieved excellent performance in protein attribute prediction and protein design. However, PLMs cannot integrate textual information, which is critical for downstream tasks involving cross-modal reasoning.

2.2 Mixed Protein-Text Language Models

To overcome the limitation of separate protein and textual modeling, researchers have developed mixed protein-text models that aim to bridge biological and linguistic domains, which can be mainly divided into three categories: **Contrastive Learning Based Methods** (Xu et al., 2023; Liu et al., 2023, 2024c; Wu et al., 2024a) employs contrastive learning to align protein sequence with their textual annotations, **Text-Augmented Pre-training Methods** (Ferruz et al., 2022; Taylor et al., 2022; Lv et al., 2024; Pei et al., 2023; Zhuo et al., 2024; Liu et al., 2024b) expand the pre-training corpora to include protein sequences, and **Multi-Modal Fusion Methods** (Liu et al., 2024c; Abdine et al., 2024; Wang et al., 2024) adopt protein encoders to extract sequence embeddings, and then align them to LLMs through projector layers. However, as LLMs increase in parameter size, retraining demands significant time and computational resources, while fine-tuning can result in catastrophic forgetting.

2.3 Protein Related Retrieval-Based Methods

In the field of protein understanding, retrieval and comparison-based methods are extensively utilized. **Multi-Sequence Alignment Models** (Rao et al., 2021; Jumper et al., 2021; Li et al., 2024) leverage multi-sequence alignment techniques to enhance deep learning model performance in protein attribute and structure prediction. An alternative approach, **Single-Sequence Alignment Method** (Ma et al., 2023), offers improvements in model performance while increasing speed by modifying the alignment process from multiple to single sequences. Additionally, **Retrieval-Enhanced Prediction Models** (Shaw et al., 2024) utilize retrieval-enhanced techniques specifically for protein attribute prediction tasks, and ProlLM (Jin et al., 2024) applies thought chain retrieval to enhance the efficacy of protein interaction predictions. Despite their advancements, these methods predominantly concentrate on attribute prediction tasks and do not adequately address more complex challenges such as protein annotation.

3 Methodology

The overall pipeline of our methods is shown in Fig. 4. To leverage the gap between protein sequences and bio-textual description, a CLIP-like model is

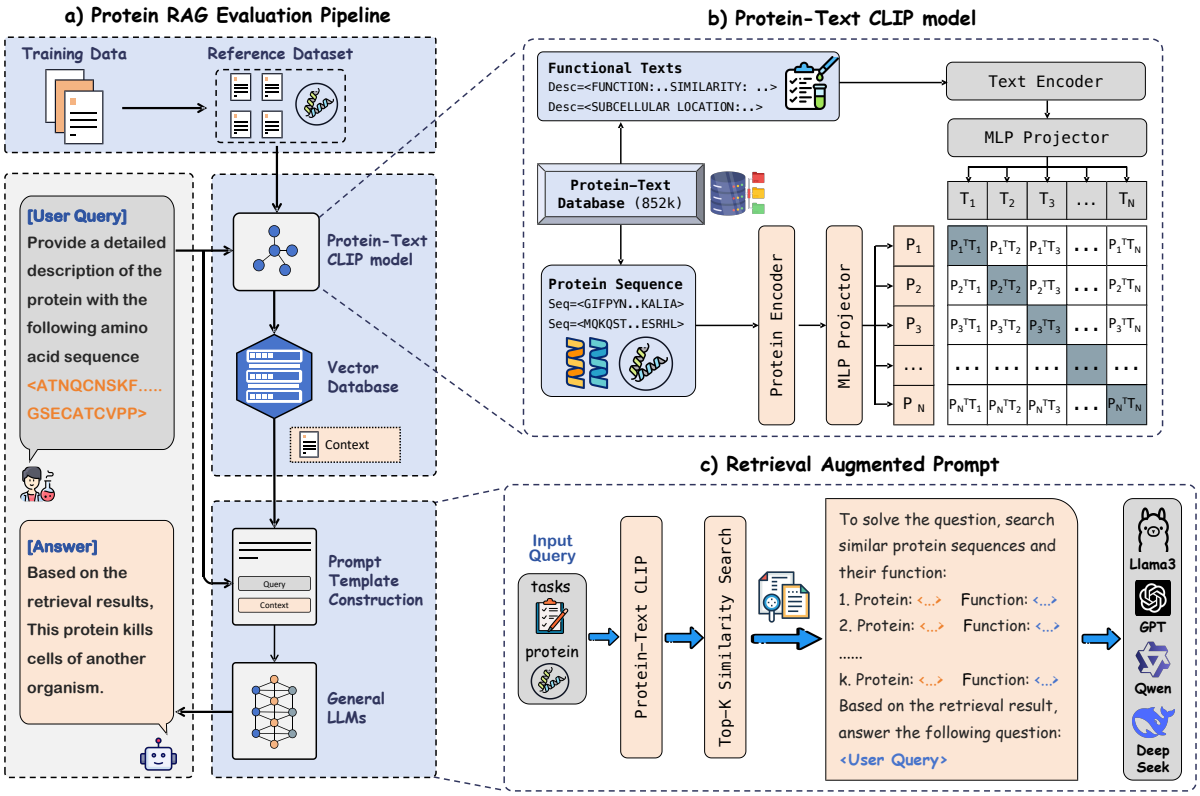

Figure 3: Overview of the Protein-Text CLIP training and retrieval-augmented prompting framework. **(a)** Protein-Text CLIP is trained using protein sequences and textual descriptions from the Swiss-Prot dataset, aligning protein embeddings with text embeddings in a shared space. **(b)** Given a user query with a protein sequence, top-K similar entities are retrieved using Protein-Text CLIP. A knowledge-augmented prompt is created and processed by advanced language models (e.g., Llama 3, GPT-4) to generate detailed biological insights.

trained to perform a bidirectional search between protein and text. For downstream tasks, we first use this model to retrieve the most similar sequences and their description in the training dataset, constructing RAP with retrieval results. Pre-trained LLMs can use RAP to predict the final answer.

3.1 Protein-Text CLIP

The Contrastive Language-Image Pre-Training (CLIP) model (Radford et al., 2021) has achieved remarkable success in cross-modal retrieval within the visual domain. The original CLIP architecture employs separate image and text encoders, trained with contrastive learning on (image, text description) pairs. Specifically, for each training batch containing k pairs $\{P_i, T_i\}$, the image encoder extracts features F_p^i from all P_i, and the text encoder extracts features F_t^i from all T_i. The optimization objective is to maximize the similarity between matching pairs (F_p^i and F_t^i) while minimizing the similarity between non-matching pairs (F_p^i and F_t^j for $i \neq j$).

Building on this framework, we introduce the Protein-Text CLIP model, which adapts the CLIP paradigm to the protein-text domain. To leverage existing pre-trained models and reduce computational overhead, we utilize ESM-C (ESM Team, 2024) as the protein encoder and BioGPT (Luo et al., 2022) as the text encoder. A multi-layer perceptron (MLP) is employed to project the embeddings from ESM-C and BioGPT into a shared feature space of identical dimensionality, facilitating effective similarity computation. The overall architecture of Protein-Text CLIP is illustrated in Figure 4 (b). Unlike to existing work ProteinCLIP (Wu et al., 2024a), we utilize the different protein encoder and text encoder, and train the whole model instead of only the projector part.

3.1.1 Architecture and Training

Protein-Text CLIP adopts a dual-encoder architecture inspired by the original CLIP model, tailored for protein and text modalities. The protein encoder ESM-C 600M and the text encoder BioGPT generate feature vectors of 1152 and 768 dimensions, respectively. Both encoders are linked to MLP pro-

jection heads that map their outputs into a unified 512-dimensional embedding space for cross-modal similarity computation. The model is trained on a combined dataset from Swiss-Prot (Bairoch and Apweiler, 2000) and ProteinKG25 (Zhang et al., 2022). Detailed information about the dataset can be seen in Section 4.1.

3.1.2 Loss Function

To align the protein and text embeddings, we adopt a symmetric contrastive loss function inspired by the original CLIP model. This involves computing cross-entropy losses in both protein-to-text and text-to-protein directions and averaging them. The loss function is defined as Equation 1, 2 and 3.

$$\mathcal{L}_{\text{p2t}} = \frac{1}{2k} \sum_{i=1}^{k} \log \left(\frac{\exp\left(\text{sim}(F_p^i, F_t^i)/\tau\right)}{\sum_{j=1}^{k} \exp\left(\text{sim}(F_p^i, F_t^j)/\tau\right)} \right) \quad (1)$$

$$\mathcal{L}_{\text{t2p}} = \frac{1}{2k} \sum_{i=1}^{k} \log \left(\frac{\exp\left(\text{sim}(F_t^i, F_p^i)/\tau\right)}{\sum_{j=1}^{k} \exp\left(\text{sim}(F_t^i, F_p^j)/\tau\right)} \right) \quad (2)$$

$$\mathcal{L} = \mathcal{L}_{\text{p2t}} + \mathcal{L}_{\text{t2p}} \quad (3)$$

In Equation 1,2, τ represents a learnable temperature parameter, and sim denotes the cosine similarity between the projected embeddings. This symmetric loss ensures that both protein-to-text and text-to-protein alignments are optimized, enhancing the robustness of the cross-modal representations.

3.2 Retrieval-Augmented Prompt Construction

The ProteinRAP framework constructs task-specific prompts through a hybrid representation learning and retrieval process, as illustrated in Figure 4 (c). For each training sample $x_i \in \mathcal{D}_{\text{train}}$ containing protein sequence P_i and associated text description T_i, we compute dual-modality embeddings using Protein-Text CLIP:

$$\begin{aligned} \mathbf{F}_p^i &= \text{CLIP}_{\text{protein}}(P_i) \in \mathbb{R}^d \\ \mathbf{F}_t^i &= \text{CLIP}_{\text{text}}(T_i) \in \mathbb{R}^d \end{aligned} \quad (4)$$

The mixed embedding $\mathbf{M}_i$ is computed through modality fusion:

$$\mathbf{M}_i = \alpha \mathbf{F}_p^i + (1 - \alpha)\mathbf{F}_t^i \quad (5)$$

where $\alpha \in [0, 1]$ controls the sequence-text balance. These mixed embeddings are indexed using Faiss (Johnson et al., 2019) with exact inner-product search (IndexFlatIP), which guarantees precise retrieval for moderate-scale biological datasets. The knowledge database $\mathcal{B} = \{(\mathbf{M}_i, x_i)\}_{i=1}^{N}$ maps embeddings to original samples.

Retrieval operates through cosine similarity computed as normalized inner products:

$$\text{sim}(\mathbf{M}_i, \mathbf{M}_j) = \frac{\mathbf{M}_i \cdot \mathbf{M}_j}{\|\mathbf{M}_i\| \|\mathbf{M}_j\|} \quad (6)$$

During training, for each x_i we retrieve its k-nearest neighbors from $\mathcal{B}$ using cosine similarity:

$$\mathcal{N}_k(x_i) = \underset{\mathbf{M}_j \in \mathcal{B} \setminus \{\mathbf{M}_i\}}{\text{top-}k} \text{sim}(\mathbf{M}_i, \mathbf{M}_j) \quad (7)$$

Test samples $x' \in \mathcal{D}_{\text{test}}$ retrieve neighbors from $\mathcal{B}$ using the same similarity metric. The final prompt $\mathcal{P}(x)$ for input x combines the original sample with retrieved instances:

$$\mathcal{P}(x) = [x; \mathcal{N}_k(x)] \quad (8)$$

where $[\cdot;\cdot]$ denotes context concatenation. Implementation details and prompt templates are provided in Supplementary A.

3.3 Instruction Tuning and RAP In-Context Learning

Instruction tuning with constructed prompts enables LLMs to effectively utilize Retrieval-Augmented Prompts (RAPs) for summarizing and extracting answers from retrieval results, which is simpler than learning implicit protein features directly from sequences. To enhance few-shot prediction capabilities using models exceeding 70 billion parameters, RAPs leverage strong in-context learning abilities. In ProteinRAP, relevant textual descriptions are retrieved for a given protein sequence query Q. Let $R(Q) = \{T_1, T_2, \ldots, T_n\}$ be these descriptions. The prompt $\mathcal{P}(Q)$ is:

$$\mathcal{P}(Q) = \text{"[Retrievals]: "}\ T_1\ T_2\ \ldots\ T_n\ \text{"[Query]: "}Q \quad (9)$$

This structure integrates retrievals into the prompt, enriching the model with relevant context and enhancing prediction accuracy. In-context learning, whereby models use embedded examples within the prompt, aids in guiding the responses. The LLM processes $\mathcal{P}(Q)$, which includes both the query and retrievals, to produce the prediction $\hat{y}$:

$$\hat{y} = \text{LLM}(\mathcal{P}(Q)) \quad (10)$$

Here, $\hat{y}$ is the output prediction, benefiting from query-driven augmentation during prompt construction.

4 Experiment

In this section, we evaluate the performance of Protein-Text CLIP on protein-text retrieval tasks. Moreover, on two open-ended answer generation tasks, Protein Caption and Protein Understanding, we trained and tested the performance of existing LLMs, and demonstrated the performance of retrieval-based methods under instruction fine-tuning and context learning

4.1 Protein-Text Dataset

In this section, we introduce the datasets used in protein retrieval, protein caption, and protein understanding tasks, including Swiss-Prot, ProteinKG25, and Mol-Instruction. Statistical information on these datasets is provided in Appendix C.

Swiss-Prot (Bairoch and Apweiler, 2000) The Swiss-Prot database is a high-quality, manually curated protein database that provides comprehensive annotations for proteins, including functional descriptions, catalytic activities, biological processes, and subcellular localization. In this study, we adopted the annotation processing methodology from ProtT3 (Liu et al., 2024c), focusing on three key attributes: **FUNCTION**, **SUBCELLULAR LOCATION**, and **SIMILARITY**. These curated attributes were extracted to form (protein, text) pairs for model training.

ProteinKG25 (Zhang et al., 2022) The ProteinKG25 dataset is a comprehensive knowledge graph derived from the Gene Ontology database. This dataset encodes protein-related information in the form of triples, representing relationships between proteins and their associated attributes or terms. Utilizing the annotation processing methodology from ProtT3 (Liu et al., 2024c), we aggregated all triples corresponding to the same protein and transformed them into free-text descriptions using predefined text templates.

Mol-Instruction (Fang et al., 2024) The Mol-Instruction dataset is a specialized instructional dataset designed to address the limitations of LLMs in the biomolecular domain. It comprises three key components: molecule-oriented instructions, protein-oriented instructions, and biomolecular text instructions. In this study, we utilize the protein-oriented subset, mainly focus on four tasks, **protein function**, **general function**, **domain motif** and **catalytic activity**.

4.2 Protein-Text Bi-directional Retrieval

We extract the (text, protein) data from the ProteinKG25 and Swiss-Prot datasets to train the Protein-Text CLIP model. In order to compare with existing methods, we tested the retrieval performance in batch and in the whole test dataset on ProteinKG25. Following (Liu et al., 2024c), we use the accuracy and Recall@20 as evaluation metrics. Besides, we employ ProtST (Xu et al., 2023), ProteinCLAP (Liu et al., 2023) and ProtT3 stage 1 (Liu et al., 2024c) as baselines.

Results Table 3 shows our results: we observed that in the whole test set, our method improved by about 21% in accuracy and 5.7% in recall@20 compared with the previous best method, demonstrating the superiority of our model in cross-modal retrieval.

4.3 Protein Captioning

The protein caption task involves generating descriptive textual annotations for given protein sequences, thereby enhancing the understanding and analysis of protein functions and characteristics. We utilize the Swiss-Prot dataset (Bairoch and Apweiler, 2000) to create (protein sequence, text description) pairs for both training and evaluation. Following (Liu et al., 2024c), BLEU (Papineni et al., 2002), ROUGE (Lin, 2004), METEOR (Banerjee and Lavie, 2005) and Exact Matching are used as metrics. Details of these evaluation metrics can be found in Appendix D. In this task, we have trained the most advanced LLMs by full parameter tuning and LoRA tuning as baselines, also compared them with the existing methods. Specifically, we perform full parameter tuning on Galactica (Taylor et al., 2022), BioGPT (Luo et al., 2022), Llama3.3-1B, Llama3.2-3B (Dubey et al., 2024), utilize LoRA fine-tuning on Llama3.1-8B (Dubey et al., 2024) and ProLLaMA-7B (Lv et al., 2024), compare with ProtT3 (Liu et al., 2024c).

For our approach, we evaluate the results of the fine-tuning model ProteinRAP-1B and the general large-scale model (GPT-4o) (Achiam et al., 2023) with RAP. ProteinRAP-1B uses Llama-3.2-1B-Instruct (Dubey et al., 2024) as the base model, trained in one epoch on the RAP format training dataset and evaluated with the same format. GPT-4o with RAP is a training-free method that performs in-context learning directly from retrieval prompts to predict the target answer.

Model	Exact.	BLEU-2	BLEU-4	ROUGE-1	ROUGE-2	ROUGE-L	METEOR	Average.
Galactica-1.3B	11.2	23.6	20.5	40.4	39.5	29.2	37.7	28.8
BioGPT-347M	0.0	9.4	7.2	28.18	27.5	13.3	26.5	16.0
ProLLaMA-7B*	0.0	4.5	3.4	12.8	6.2	11.7	21.4	8.5
Llama3.3-1B	22.0	60.3	57.9	54.7	44.2	53.1	61.8	50.5
Llama3.2-3B	34.0	68.9	67.1	65.0	58.5	63.7	69.8	61.0
Llama3.1-8B*	8.7	20.9	17.8	39.0	37.7	25.9	35.9	26.5
ProtT3-1.3B	25.7	55.0	51.4	63.6	56.5	62.1	63.6	53.9
GPT-4o w/ RAP	38.2	71.9	70.4	83.5	80.1	82.6	81.8	72.6 (19% ↑)
ProteinRAP-1B*	**46.5**	**81.4**	**80.5**	**80.8**	**77.1**	**79.9**	**81.9**	**75.5** (23% ↑)

Table 1: Performance (%) comparison of Swiss-Prot (Bairoch and Apweiler, 2000) protein caption tasks. "*" stands for LoRA finetuning. **Bold** indicates the best performance, underline indicates the second-best performance. ($x\%$ ↑) represents the performance improvement over existing methods.

Model	Protein Function		General Function		Domain Motif		Catalytic Activity		Average.
	R-L	METEOR	R-L	METEOR	R-L	METEOR	R-L	METEOR	
Galactica-1.3B	7.1	8.6	48.2	46.2	55.3	57.3	30.2	31.4	35.53
BioGPT-347M	50.9	51.8	49.7	45.1	**55.4**	57.1	54.2	50.5	51.83
ProLLaMA-7B*	48.6	53.2	20.3	35.0	46.7	57.0	39.3	50.6	43.83
Llama-3.2-1B*	46.5	47.1	45.1	39.9	49.9	53.9	52.6	51.4	48.30
Llama-3.1-8B*	52.1	54.4	54.2	50.4	51.2	56.7	59.6	61.1	54.96
BioT5-Plus-252M	56.6	62.2	68.0	67.7	53.4	62.0	71.8	77.6	64.91
GPT-4o w/ RAP	58.8	65.5	74.8	73.0	44.0	43.3	72.4	76.6	63.55 (2% ↓)
ProteinRAP-1B*	62.0	69.6	76.1	76.6	54.0	**62.2**	75.7	83.6	69.97 (7.1% ↑)
ProteinRAP-8B*	**62.8**	**70.4**	**76.8**	**77.2**	53.4	61.0	**76.1**	**84.0**	**70.21** (8.1% ↑)

Table 2: Performance (%) comparison of different models across four protein understanding tasks. "R-L" stands for ROUGE-L metric, "*" stands for LoRA finetuning. **Bold** indicates the best performance, underline indicates the second-best performance. ($x\%$ ↑) represents the performance improvement over existing methods.

Model	Batched (64)		Test Set (10k)	
	Acc	R@20	Acc	R@20
ProtST	70.8	98.5	5.5	41.6
ProteinCLAP	**93.2**	99.2	53.4	91.2
ProtT3	92.3	98.9	55.8	91.7
Our Method	92.1	**99.5**	**67.6**	**97.0**

Table 3: Protein-to-text retrieval performance (%) (Acc, R@20) on the ProteinKG25(Zhang et al., 2022) dataset.

Results Table 1 presents the results. We observed that (1) The models using protein data in the pre-training stage (Galactica, BioGPT, ProLLaMA) performed worse than the Llama series models, which may be due to their lack of text processing ability. (2) LLMs with full parameter fine-tuning outperform the LoRA model, pointing out that learning from protein sequences needs more trainable parameters. (3) Using RAP can significantly enhance the effect of LLM on this task. Our method achieves 23 % and 19 % improvement of existing methods in instruction fine-tuning and in-context learning respectively.

4.4 Protein Understanding

The Protein Understanding task is designed to evaluate the ability of models to accurately follow instructions related to protein-specific queries, which consists of three components: [instructions], [input], and [output]. Unlike the Protein Caption task, the Protein Understanding task is more challenging, requiring LLMs to simultaneously handle both the protein sequences and the instructions to generate final answers. Mol-Instruction dataset mentioned in Section 4.1 is used, employing ROUGE-L (Lin, 2004) and METEOR (Banerjee and Lavie, 2005) as evaluation metrics for each task, using the average scores across all tasks to assess the models' capabilities.

Similar to the Protein Caption in Section 4.3, we trained and evaluated the performance of various LLMs, with baselines including LLama3 (Dubey et al., 2024), ProLLaMA (Lv et al., 2024), Galactica (Taylor et al., 2022), and BioT5 plus (Pei et al., 2024). For our method, we tested the fine-tuning performance of RAPs on the 1B and 8B LLama

models and explored the effects of using prompts for in-context reasoning with GPT-4o.

Result The experimental results, as shown in Table 2, allow us to draw the following conclusions: (1) In the protein understanding task, LLMs pretrained with protein data exhibit performance comparable to the Llama3 models, indicating that these models have strong capabilities in processing protein sequences and text simultaneously. (2) For the Llama3 models, increasing the model scale leads to better performance in the protein understanding task, regardless of whether RAPs are used. (3) Methods based on in-context learning can achieve performance similar to previous best models without additional training, while the instruction finetuning model achieves an average improvement of 8.1%.

5 Ablation Study

In this part, we analyzed the ablation of the Protein-Text CLIP module and retrieval enhancement prompt. Specifically, we studied the following tasks from the perspective of model training and method implementation **(1) Is RAPs generic on existing open source LLMs?** As shown in Table 4 , We have studied the improvement of three accessible LLMS in protein understanding tasks by raps. The results show that after enhancing raps, the general large-scale model can achieve great improvement in all tasks, which shows the universality of our method.

(2) Can RAPs still perform well in the case of insufficient retrieval samples? As shown in Table 5, we used a more difficult division on the general function task and only predicted 80% of the test data from 20% of the training data. Experiments show that the baseline method has a huge decline when using more difficult partition methods, while the proteinRAP can still maintain a good performance.

(3) in RAPs, how does the number of retrieved entries K affect the performance of the model? As shown in Figure 4, we visualized the performance of LLMS' different K on four tasks. The results show that increasing the number of K can slightly increase the performance of the model, but at the same time, due to too many samples, LLMS will be misled by the wrong samples, and the effect will decline on some cases

Model	PF	GF	DM	CA	Avg.
Llama3 70B					
- w/o RAP	27.0	22.1	34.4	36.0	29.8
- w/ RAP	56.7	66.8	45.7	66.6	**58.9** (97.6% ↑)
GPT-4o					
- w/o RAP	27.3	26.8	36.0	41.0	32.7
- w/ RAP	58.8	70.6	44.0	72.4	**61.4** (87.7% ↑)
DeepSeek V3					
- w/o RAP	25.5	19.8	30.3	36.5	28.0
- w/ RAP	51.1	30.7	46.9	62.6	**47.8** (70.7% ↑)

Table 4: ROUGE-L Performance (%) Comparison of Large Models with and without RAP Across Four Tasks, "PF", "GF", "DM", and "CA" stands for "general function", "domain motif", "catalytic activity", and "protein function" tasks respectively.

Model	B-2	B-4	R-1	R-2	R-L
ProteinRAP-1B					
- Original Split	74.7	71.3	77.0	68.6	76.1
- Train:Test = 5:5	73.0	69.1	74.2	65.3	73.5
- Train:Test = 2:8	66.3	62.0	68.4	58.1	66.9
Llama-3.2-1B					
- Original Split	48.2	44.3	56.3	45.3	54.6
- Train:Test = 5:5	47.1	42.8	53.3	41.4	51.2
- Train:Test = 2:8	20.4	14.9	35.6	21.1	32.7

Table 5: Performance (%) Comparison of ProteinRAP-1B and Llama-3.2-1B Models Under Different Data Splits, in which "B-2" and "B-4" means BLUE-2, BLUE-4 metrics, "R-1", "R-2" and "R-L" means ROUGE-1, ROUGE-2 and ROUGE-L metrics.

6 Conclusions

In this study, we introduced a novel approach, ProteinRAP, in the domain of protein science by leveraging retrieval-augmented prompts to enhance the capabilities of LLMs in protein-related tasks. Through comprehensive evaluations, we demonstrated that our retrieval-enhanced paradigm closes the performance gap between general LLMs and models specifically pre-trained with protein data. Our findings indicate that ProteinRAP significantly outperforms existing methods in protein captioning and understanding, achieving remarkable improvements even in a training-free setup. These results underscore the potential of retrieval-augmented methodologies to enable efficient and scalable solutions for complex biological tasks without the need for extensive parameter tuning. By showcasing the utility of cross-modal retrieval and prompt engineering, this work sets a new direction for future explorations in enhancing LLMs' applicability in specialized domains such as protein science.

7 Limitations

While ProteinRAP demonstrates substantial improvements, it has several limitations. **Weakness in Protein Design Tasks** remains a challenge, as the method performs well in understanding, it has suboptimal results in protein design tasks, which is shown in Appendix A. **Retrieval Methodology Limitations** hinder performance when high-quality data is lacking, and optimal base model selection requires further study. Furthermore, **Limited Exploration of Other Scientific Entities** indicates that our approach has yet to extend beyond protein sequences to entities such as DNA and RNA. We will improve the method in the later feature work to solve these limitations.

8 Potential Risks

In this study, the proposed ProteinRAP method focuses on enhancing the protein understanding capabilities of LLMs by using retrieval-augmented prompts. While this approach does not involve human subjects, it is crucial to consider the potential risks associated with its application. Similar to other LLM-focused research, ProteinRAP could be misused to generate inaccurate or misleading descriptions of protein properties, which may have implications for scientific research and applications. Additionally, the enhanced understanding capabilities of LLMs in the biological domain might inadvertently contribute to the development of harmful applications, such as engineered pathogens, if not properly regulated. Therefore, we encourage researchers and practitioners employing this method to remain vigilant about these risks and to implement appropriate safeguards to minimize potential misuse.

References

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. 2024. Prot2text: Multimodal protein's function generation with gnns and transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10757–10765.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *ArXiv preprint*, abs/2303.08774.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *ArXiv preprint*, abs/2309.16609.

Amos Bairoch and Rolf Apweiler. 2000. The swiss-prot protein sequence database and its supplement trembl in 2000. *Nucleic acids research*, 28(1):45–48.

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72, Ann Arbor, Michigan. Association for Computational Linguistics.

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. 2022. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38(8):2102–2110.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Yue Cao and Yang Shen. 2021. Tale: Transformer-based protein function annotation with joint sequence–label embedding. *Bioinformatics*, 37(18):2825–2833.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45.

Bo Chen, Xingyi Cheng, Pan Li, Yangli-ao Geng, Jing Gong, Shen Li, Zhilei Bei, Xu Tan, Boyan Wang, Xin Zeng, et al. 2024. xtrimopglm: unified 100b-scale pre-trained transformer for deciphering the language of protein. *ArXiv preprint*, abs/2401.06199.

UniProt Consortium. 2019. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *ArXiv preprint*, abs/2407.21783.

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom

Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. 2021. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127.

ESM Team. 2024. Esm cambrian: Revealing the mysteries of proteins with unsupervised learning.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2024. Mol-instructions: A large-scale biomolecular instruction dataset for large language models. In *The Twelfth International Conference on Learning Representations*. OpenReview.net.

Noelia Ferruz, Steffen Schmidt, and Birte Höcker. 2022. Protgpt2 is a deep unsupervised language model for protein design. *Nature communications*, 13(1):4348.

Thomas Hayes, Roshan Rao, Halil Akin, Nicholas J Sofroniew, Deniz Oktay, Zeming Lin, Robert Verkuil, Vincent Q Tran, Jonathan Deaton, Marius Wiggert, et al. 2025. Simulating 500 million years of evolution with a language model. *Science*, page eads0018.

Martina Hilbert, Gerald Böhm, and Rainer Jaenicke. 1993. Structural relationships of homologous proteins as a fundamental principle in homology modeling. *Proteins: Structure, Function, and Bioinformatics*, 17(2):138–151.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Mingyu Jin, Haochen Xue, Zhenting Wang, Boming Kang, Ruosong Ye, Kaixiong Zhou, Mengnan Du, and Yongfeng Zhang. 2024. ProLLM: Protein chain-of-thoughts enhanced LLM for protein-protein interaction prediction. In *First Conference on Language Modeling*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 7(3):535–547.

John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. 2021. Highly accurate protein structure prediction with alphafold. *nature*, 596(7873):583–589.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Pan Li, Xingyi Cheng, Le Song, and Eric P Xing. 2024. Retrieval augmented protein language models for protein structure prediction. *bioRxiv*, pages 2024–12.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. 2024a. Deepseek-v3 technical report. *ArXiv preprint*, abs/2412.19437.

Nuowei Liu, Changzhi Sun, Tao Ji, Junfeng Tian, Jianxin Tang, Yuanbin Wu, and Man Lan. 2024b. Evollama: Enhancing llms' understanding of proteins via multimodal structure and sequence representations. *arXiv preprint arXiv:2412.11618*.

Shengchao Liu, Yanjing Li, Zhuoxinran Li, Anthony Gitter, Yutao Zhu, Jiarui Lu, Zhao Xu, Weili Nie, Arvind Ramanathan, Chaowei Xiao, et al. 2023. A text-guided protein design framework. *ArXiv preprint*, abs/2302.04611.

Zhiyuan Liu, An Zhang, Hao Fei, Enzhi Zhang, Xiang Wang, Kenji Kawaguchi, and Tat-Seng Chua. 2024c. ProtT3: Protein-to-text generation for text-based protein understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5949–5966, Bangkok, Thailand. Association for Computational Linguistics.

Renqian Luo, Liai Sun, Yingce Xia, Tao Qin, Sheng Zhang, Hoifung Poon, and Tie-Yan Liu. 2022. Biogpt: generative pre-trained transformer for biomedical text generation and mining. *Briefings in bioinformatics*, 23(6):bbac409.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *ArXiv preprint*, abs/2308.08747.

Liuzhenghao Lv, Zongying Lin, Hao Li, Yuyang Liu, Jiaxi Cui, Calvin Yu-Chian Chen, Li Yuan, and Yonghong Tian. 2024. Prollama: A protein large language model for multi-task protein language processing. *arXiv e-prints*, pages arXiv–2402.

Chang Ma, Haiteng Zhao, Lin Zheng, Jiayi Xin, Qintong Li, Lijun Wu, Zhihong Deng, Yang Lu, Qi Liu, and Lingpeng Kong. 2023. Retrieved sequence augmentation for protein representation learning. *bioRxiv*, pages 2023–02.

Ali Madani, Ben Krause, Eric R Greene, Subu Subramanian, Benjamin P Mohr, James M Holton, Jose Luis Olmos, Caiming Xiong, Zachary Z Sun, Richard Socher, et al. 2023. Large language models generate functional protein sequences across diverse families. *Nature echnology*, 41(8):1099–1106.

Erik Nijkamp, Jeffrey A Ruffolo, Eli N Weinstein, Nikhil Naik, and Ali Madani. 2023. Progen2: exploring the boundaries of protein language models. *Cell systems*, 14(11):968–978.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Qizhi Pei, Lijun Wu, Kaiyuan Gao, Xiaozhuan Liang, Yin Fang, Jinhua Zhu, Shufang Xie, Tao Qin, and Rui Yan. 2024. BioT5+: Towards generalized biological understanding with IUPAC integration and multi-task tuning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1216–1240, Bangkok, Thailand. Association for Computational Linguistics.

Qizhi Pei, Wei Zhang, Jinhua Zhu, Kehan Wu, Kaiyuan Gao, Lijun Wu, Yingce Xia, and Rui Yan. 2023. BioT5: Enriching cross-modal integration in biology with chemical knowledge and natural language associations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1102–1123, Singapore. Association for Computational Linguistics.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Roshan Rao, Jason Liu, Robert Verkuil, Joshua Meier, John F. Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. 2021. MSA transformer. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR.

Peter Shaw, Bhaskar Gurram, David Belanger, Andreea Gane, Maxwell L Bileschi, Lucy J Colwell, Kristina Toutanova, and Ankur P Parikh. 2024. Protex: A retrieval-augmented approach for protein function prediction. *bioRxiv*, pages 2024–05.

Yang Tan, Ruilin Wang, Banghao Wu, Liang Hong, and Bingxin Zhou. 2024. Retrieval-enhanced mutation mastery: Augmenting zero-shot prediction of protein language model. *ArXiv preprint*, abs/2410.21127.

Ross Taylor, Marcin Kardas, Guillem Cucurull, Thomas Scialom, Anthony Hartshorn, Elvis Saravia, Andrew Poulton, Viktor Kerkez, and Robert Stojnic. 2022. Galactica: A large language model for science. *ArXiv preprint*, abs/2211.09085.

Chao Wang, Hehe Fan, Ruijie Quan, and Yi Yang. 2024. Protchatgpt: Towards understanding proteins with large language models. *ArXiv preprint*, abs/2402.09649.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kevin E Wu, Howard Chang, and James Zou. 2024a. Proteinclip: enhancing protein language models with natural language. *bioRxiv*, pages 2024–05.

Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024b. Continual learning for large language models: A survey. *ArXiv preprint*, abs/2402.01364.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. 2023. Protst: Multi-modality learning of protein sequences and biomedical texts. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38749–38767. PMLR.

Ningyu Zhang, Zhen Bi, Xiaozhuan Liang, Siyuan Cheng, Haosen Hong, Shumin Deng, Qiang Zhang, Jiazhang Lian, and Huajun Chen. 2022. Ontoprotein: Protein pretraining with gene ontology embedding. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. 2023. Instruction tuning for large language models: A survey. *ArXiv preprint*, abs/2308.10792.

Zehui Zhao, Laith Alzubaidi, Jinglan Zhang, Ye Duan, and Yuantong Gu. 2024. A comparison review of transfer learning and self-supervised learning: Definitions, applications, advantages and limitations. *Expert Systems with Applications*, 242:122807.

Le Zhuo, Zewen Chi, Minghao Xu, Heyan Huang, Jianan Zhao, Heqi Zheng, Conghui He, Xian-Ling Mao, and Wentao Zhang. 2024. ProtLLM: An interleaved protein-language LLM with protein-as-word pre-training. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8950–8963, Bangkok, Thailand. Association for Computational Linguistics.

A ProteinRAP in Protein Design Tasks

A.1 Task Overview

The protein design task, derived from the Mol-Instructions dataset (Fang et al., 2024), requires

models to synthesize protein sequences that meet specific functional and structural constraints. Models must interpret complex instructions detailing properties like enzymatic specificity, metal ion binding, and solubility optimization, and output corresponding amino acid sequences. This task has critical applications in drug design, synthetic biology, and enzyme engineering.

A.2 Experimental Results

We evaluated ProteinRAP alongside other baseline models, including Galactica-1.3B, Llama variants, ProLLaMA, and general RAP models, on the Mol-Instructions Protein Design task. Performance was evaluated using metrics such as BLEU (2/4-gram), METEOR, and ROUGE. Table 7 details the results.

A.3 Result Analysis and Conclusion

The results reveal that retrieval-augmented prompts (RAP) provide limited improvements in protein design tasks, as LLMs struggle to effectively interpret and utilize retrieved protein sequence information compared to textual data. In contrast, BioT5+, which underwent unsupervised pretraining on protein-specific datasets, significantly outperforms RAP-based and instruction-tuned autoregressive models across most metrics. This underscores the importance of domain-specific pretraining for understanding complex protein data. Future work should explore combining unsupervised pretraining on protein data with RAP approaches to further enhance task performance.

B Model Training Details

B.1 Protein-Text CLIP

Model Architecture The Protein-Text CLIP model consists of two primary components: a protein sequence encoder and a text encoder. The protein encoder is based on ESMC (600M) (Hayes et al., 2025) for protein sequence understanding, while the text encoder leverages BioGPT (Luo et al., 2022) to process textual descriptions. Both encoders generate high-dimensional embeddings, which are then projected into a shared 512-dimensional latent space using linear projections (see Table 8). Ablation about protein encoder and text encoder can be seen in Table 6.

During training, the model employs contrastive learning to align protein and text representations. Specifically, mean-pooled embeddings from both modalities are normalized and passed through separate projection layers. The resulting embeddings are used to compute a similarity score, scaled by a learnable temperature parameter σ, and optimized using cross-entropy loss.

Hyper-Parameters We used the hyper-parameters summarized in Table 8 to train Protein-Text CLIP. Mixed precision training with bf16 was enabled to accelerate large-scale computations on GPUs. During training, we combine two datasets: SwissProt and OntoProtein, for both protein and text inputs.

Training Procedure: The model was trained on 4 GPUs using the "Accelerate" library. Protein-text pairs were tokenized and encoded separately for training. During inference, embeddings were extracted to compute recall at various thresholds (`recall@k`) using FAISS indexing. The loss function alternates between optimizing logits for protein-text alignment and text-protein alignment.

The reported evaluation metrics include recall@1, recall@10, and recall@20. These metrics provide quantitative measures of the model's ability to retrieve correct text descriptions for a given protein sequence.

B.2 Large Language Model

Model Architecture We leverage a large pre-trained causal language model for protein-related tasks, fine-tuned using instruction-tuning techniques. The training process builds upon the Llama3 (Dubey et al., 2024) framework, with additional lightweight parameter-efficient finetuning (PEFT) using the LoRA (Low-Rank Adaptation) mechanism (Hu et al., 2022).

The overall architecture consists of a transformer-based auto-regressive model fine-tuned on protein-text tasks. LoRA fine-tuning is applied to selected projection layers (e.g., **q_proj**, **k_proj**, **v_proj**, **o_proj**), allowing modification of only a small subset of the model's parameters to efficiently adapt to domain-specific tasks.

Hyper-Parameters The LLM fine-tuning process utilizes hyper-parameters shown in Table 9. Training is conducted with DeepSpeed-enabled distributed GPUs, utilizing mixed-precision (bf16) and memory optimization techniques. LoRA significantly reduces memory requirements by freezing the majority of model weights and introducing lightweight low-rank updates. The cosine learning rate schedule with warm-up ensures stable convergence.

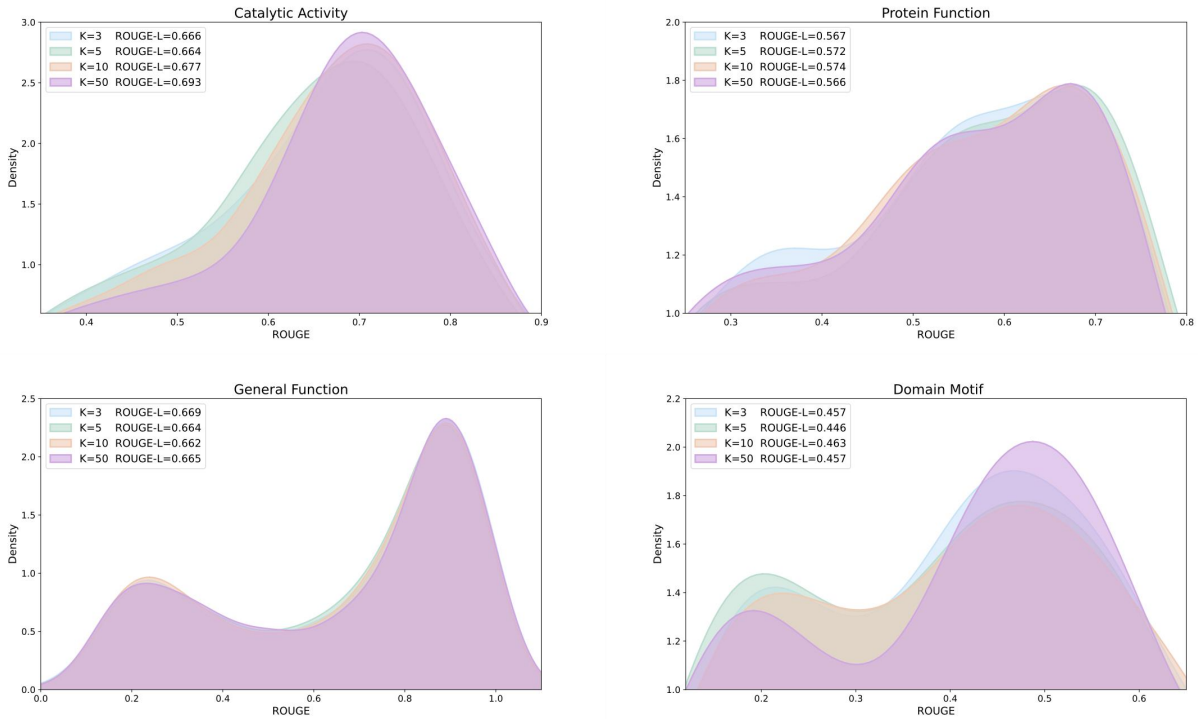

Figure 4: Ablation study of retrieval numbers of four tasks.

Protein Encoder	Text Encoder	In Batch (64)		In Test Set (10k)	
		R@1	R@10	R@1	R@10
ESM-C 300M	BioMedBERT	0.87	0.99	0.17	0.58
ESM-C 300M	BioGPT	0.90	0.99	0.22	0.66
ESM-C 600M	BioMedBERT	0.89	0.99	0.20	0.63
ESM-C 600M	BioGPT	**0.90**	**0.99**	**0.23**	**0.67**

Table 6: Ablation Study in Protein Encoder and Text Encoder selection.

Model	**BLEU-2**	**BLEU-4**	**METEOR**	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
Galactica-1.3B	8.57	3.98	15.57	32.63	14.57	25.56
Llama-3.1-8B-Instruct	8.55	3.73	18.97	47.48	22.54	39.02
Llama-3.2-1B-Instruct	8.26	3.59	17.83	**48.21**	**23.68**	37.69
ProLLaMA Stage 1	5.25	2.25	12.73	18.38	8.86	15.23
BioT5+ (ROUGE-L only)	-	-	-	-	-	**63.44**
ProteinRAP	**13.91**	**6.00**	**24.78**	47.48	22.95	38.60

Table 7: Model performance on the Mol-Instructions Protein Design task.

Evaluation The evaluation follows multi-metric assessment using BLEU, Meteor, and ROUGE scores. During inference, sampling parameters for text generation include a top-p threshold of 0.9, temperature of 0.6, and max output length of 512 tokens. The model effectively handles protein-oriented tasks such as catalytic activity annotation and protein design, demonstrating high alignment between predicted and ground-truth outputs.

C Additional Dataset and Details

The datasets used in our study consist of three main parts.

(a) The Swiss-Prot dataset includes proteins and their text descriptions. It contains a training set of 430,595 entries with an average protein length of

Hyper-parameter	Value
Protein encoder	ESMC,
Text encoder	BioGPT
Protein feature dimension	1152
Text feature dimension	768
Batch size	32
Learning rate	4e-5
Number of epochs	1
Mixed precision	bf16
Max protein sequence length	1024
Max text sequence length	512
Projection dimension	512
Optimizer	AdamW
Scheduler	linear decay
Logit scale initialization	2.6592
Training Epochs	50
Approximate training duration	2 days

Table 8: Hyper-parameter settings used for Protein-Text CLIP.

Hyper-parameter	Value
Learning rate for LoRA	1e-4
Learning rate for full parameter	4e-5
Batch size per device	2
Gradient accumulation steps	8
LoRA rank	8
LoRA α	32
LoRA dropout	0.05
Max sequence length	1024 tokens
RAP max sequence length	4096 tokens
Number of epochs	1
Optimizer	AdamW
LR scheduler type	Cosine
Warm-up ratio	0.1
Weight decay	1e-2
Mixed precision	bf16
Gradient checkpointing	Enabled
Devices	8 A100-80GB
Approximate training duration	2 hours per task
DeepSpeed config	Zero-2

Table 9: Hyper-parameter settings during training.

336 and an average text length of 48. The validation set comprises 10,000 entries, with average protein and text lengths of 358 and 59, respectively. The test set also consists of 10,000 entries, with average lengths of 357 for proteins and 60 for text.

(b) The ProteinKG25 dataset also features proteins and their text descriptions. The training set has 422,315 entries, with average protein and text lengths of 338 and 101, respectively. The validation set, containing 10,000 entries, has average lengths of 360 for proteins and 104 for text. Similarly, the test set includes 10,000 entries, with average protein and text lengths of 360 and 107, respectively.

(c) For protein property prediction tasks, the dataset contains various aspects such as protein function with 110,689 entries and 3,494 molecular instructions (PMol). Catalytic activity is represented by 51,573 entries with 1,601 PMol. Domain/Motif has 43,700 entries with 1,400 PMol, and functional description involves 83,939 entries with 2,633 PMol.

C.1 Protein Retrieval

Protein Retrieval aims to perform bidirectional retrieval between protein sequences and textual descriptions using datasets such as SwissProt and ProteinKG25. A pretrained Protein-Text CLIP model is employed, evaluated with Recall@k. The task includes: protein-to-text retrieval: Given a protein sequence, retrieve its corresponding textual description. This task benchmarks the ability of models to bridge protein and text representations.

C.2 Protein Caption

Protein Caption generates functional, subcellular, and molecular similarity descriptions for proteins. Using SwissProt annotations. This task enables functional characterization of unknown proteins. A detailed breakdown, along with related query-answer tasks, is shown in Table 10.

D Details on Metrics

We evaluate the model using several commonly used evaluation metrics adapted to protein description generation and understanding tasks. Here, we detail these metrics, including their calculation method, significance, and specific usage.

Exact Match: This metric measures the proportion of predictions that exactly match the ground truth. It is typically used for retrieval tasks and provides an intuitive understanding of prediction accuracy.

Recall@k: This metric evaluates whether the correct entity appears in the top-k retrieved items. For a prediction system:

BLEU: (Papineni et al., 2002) BLEU, or BiLingual Evaluation Understudy, is a metric often used to measure the fluency and correspondence of machine-generated sequences against reference de-

scriptions. Employing n-grams, we compute the overlap:

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

where BP is a brevity penalty, w_n are the weights typically equal for all n-grams, $\sum_{n=1}^{N} w_n = 1$, and p_n is the precision for n-grams.

ROUGE: (Lin, 2004) Recall-Oriented Understudy for Gisting Evaluation (ROUGE) measures the quality of machine-generated text by comparing its overlap with a reference set of word sequences. Specifically, it evaluates:

- ROUGE-N (e.g., ROUGE-1, ROUGE-2): Measures n-gram overlap.
- ROUGE-L: Based on the longest common subsequence, it considers both recall and precision to compute an F1 score.

METEOR: (Banerjee and Lavie, 2005) METEOR considers synonyms and linguistic variations, providing a more semantically oriented evaluation metric than BLEU or ROUGE. It is calculated using unigram precision and recall, often integrating linguistic features like stemming and synonymy. The simplified formula presented here is:

$$\text{METEOR} = \frac{10m}{(9k + p + 10m)},$$

where m is the number of aligned unigrams, k is the fragmentation penalty, and p indicates precision.

E Prompt Construction Detail

E.1 Prompt Template

In the construction of prompts for protein-related tasks, we employ distinct templates tailored to the specific nature of the task: protein function analysis or protein design. Each template is structured to include an introductory statement, a task description, retrieved examples from the database, and specific instructions for the task at hand. These components ensure a comprehensive understanding and execution of the given instructions.

E.2 Prompt Case

To better illustrate the application of these templates, a Retrieval Augmented Prompt sample for the general function task is provided. This example showcases how retrieved examples and task-specific instructions are integrated to enhance the problem-solving process.

F License

In this section, we provide an overview of the licensing terms for several models and datasets utilized in this study, detailing their respective usage conditions.

Swiss-Prot Database (Bairoch and Apweiler, 2000)

The Swiss-Prot Database is distributed under the UniProt Consortium's license, which allows free access for research and non-commercial purposes. Users must attribute the source and agree not to distribute the database without prior permission from the consortium.

UniProt Database (Consortium, 2019)

The UniProt Database is available under the Creative Commons Attribution (CC BY 4.0) License. This license permits users to share and adapt the data for any purpose, provided appropriate credit is given, a link to the license is provided, and indication of any changes made is specified.

Mol-Instructions Dataset (Fang et al., 2024)

Released under the Creative Commons Attribution-NonCommercial 4.0 International License (CC BY-NC 4.0). This license permits use, sharing, and adaptation of the dataset for non-commercial purposes, with appropriate attribution and indication of changes. Commercial use requires additional permissions.

LLaMA 3 (Dubey et al., 2024)

The LLaMA 3 model is released under the LLaMA Community License. This license permits use, modification, and distribution, with specific conditions such as prohibitions against using the model for training other language models. For commercial use, compliance with Meta's Acceptable Use Policy is mandatory, and entities with over 700 million monthly active users must obtain a separate license from Meta.

BioT5+ Model (Pei et al., 2024)

The BioT5+ model is available under the MIT License. This allows for free use, modification, and distribution, including for commercial purposes, as long as the original copyright notice and permission notice are retained. The software is provided "as is," with no warranties or guarantees, and the authors disclaim liability for any issues arising from its use.

GPT-4o (Achiam et al., 2023)

GPT-4o is proprietary software developed by OpenAI. Access is provided through OpenAI's services, such as ChatGPT and the Azure OpenAI Service, under subscription plans. The model is not open-source and is governed by OpenAI's terms of service and usage policies.

DeepSeek V3 (Liu et al., 2024a)

DeepSeek V3 is distributed under the DeepSeek License (v1.0, Oct 23, 2023). It grants a free, global, irrevocable license for modification and distribution, with strict restrictions on military use, harm, misinformation, discrimination, and unauthorized data processing. Users must enforce these limits in derivative works. Disclaimers of warranties and liability are included, and any legal matters are subject to the jurisdiction of Chinese law, specifically in Hangzhou.

Table 10: Protein Query-Answer and Caption Task Categories with Examples

Task Category	Example
Protein Caption	**Instruction:** Provide a detailed description of the protein with the following amino acid sequence, including its FUNCTION, SUBCELLULAR LOCATION, and SIMILARITY. **Input:** An amino acid sequence, e.g., "MAWRSGLCETD-SRTLKQFLQEECMWKLVGK..." **Output:** FUNCTION: Plays a critical role in cytoplasmic virus egress... SUB-CELLULAR LOCATION: Virion tegument, Host cytoplasm, Host nucleus... SIMILARITY: Belongs to the herpesviridae cytoplasmic envelopment protein 2 family...
General Function	**Instruction:** Could you evaluate the protein with this amino acid sequence and present a summary of its features? **Input:** An amino acid sequence, e.g., "MKSVHSSPQNTSHTIMTFYPT-MEEFADFNT..." **Output:** A brief summary highlighting general functions, e.g., "Histone demethy-lase that specifically demethylates 'Lys-9' of histone H3, thereby playing a central role in histone code."
Protein Function	**Instruction:** Could you analyze the protein corresponding to the amino acid sequence and offer insights on its function, the biological processes it might participate in, and its location within the cell? **Input:** An amino acid sequence, e.g., "MNPKKLVIASRESLLAMWQAKHIQ-GRLKAL..." **Output:** Description of function, biological processes, and cellular localization, e.g., "Hydroxymethylbilane synthase activity; implicated in heme biosynthetic process; localized in cytoplasm."
Catalytic Activity	**Instruction:** Given the protein sequence below, please analyze and describe the catalytic activity of the corresponding enzyme, specifically the chemical reaction it catalyzes. **Input:** An amino acid sequence, e.g., "MKPVHIVSSAQMRWADMQTMQK-TPSRTLME..." **Output:** Chemical reaction catalyzed by the enzyme, e.g., "(6S)-NADPHX + ADP = AMP + H(+) + NADPH + phosphate."
Domain/Motif	**Instruction:** Please examine the following protein sequence and predict any domains or motifs you can discern. **Input:** An amino acid sequence, e.g., "MKSIEVHTDGSCLGNPGPGGWAALL-RYNGR..." **Output:** Identified domains or motifs, e.g., "RNase H type-1 domains."

A Sample of Retrieval Augmented Prompt

```
You are an assistant that helps with protein function analysis.
Task: Analyze the protein with the following sequence and describe its properties:
Target Protein: ```
MTTPTPLRSVTVNTPPPYTIAIGPGLLHDPPRLAATIRGRHALILSDSEVAPRYAAQLHETLLRARPDLHLNVFTLPAGETSKSLENFGAAIAQLATLGATRDA
CLFALGGGVIGDLAGFTAACWMRGIDYVQVPTTLLAMVDSSVGGKTAVDIPQGKNMVGAFHPPRAVIADTDTLATLPLRELRAGLSEVIKYGAIRDPVFFHWLQ
TTREALLARDPAALAQAIARSCEHKADIVGRDPLEKGERVLLNLGHTFGHAIETTQGYSTPGSNNLNHGEAVAVGMVLAARLSNTLGLAPAEDTETLKNLLDAY
GLPTVLPSGLTPEMLLERMRLDKKNIAGRLRLVLWRGIGHAEAVPDVDEAAVRQILAN
```
Below are similar proteins retrieved from a database along with their functions:

Example 0: Protein: ```
MAKFELYAEVDVSISGHQYPIIICRNGLIDPELINRFITSKQVLIVTNRTVAPLYLGHLQSGLPSKQCDVVILEDGEEHKNQRSLFTIYDSLIQNKHHRDTSII
ALGGGVIGDMAGFAASTYQRGVRFIQLPTTLLAQVDASVGGKTAINHPAGKNMIGSFYQPQAVIIDLNTLKTLPEREFRAGIAEMIKYALLVGGPFFERIQAVL
QQGLTVHSPELPLLIAECCQVKAKIVEQDERESGLRALLNLGHTFAHALETYTDYKKWLHGEAVAIGLYCAAVLSEKKGLLDKPIVDQVEKMLIHAGLPHKIPN
SIDLIQLRELMSLDKKIKNNCLRFVMIKKPGACYIDDSVTEDCLHNTLINVVEGEQK
``` Answer: [A short report on the protein with the given amino acid sequence highlights: Catalyzes the
conversion of 3-deoxy-D-arabino-heptulosonate 7-phosphate (DAHP) to dehydroquinate (DHQ).]

Example 1: Protein: ```
MNAIESIEVALDTLPENRSYSIHIGQGLLSRMDLLLPHLPGKKAAIVTNTTIAPLYLEKLRSALAEHHVETFAITLPDGERYKHWETLNLIFDALLEHRCERRT
PLIALGGGVIGDLTGFAAATYLRGVPFIQIPTTLLAQVDSSVGGKTGINHPLGKNMIGAFYQPQLVLTDSATLTTLPDRELRAGIAEIIKYGLIYDADFFDWLE
QHMNSLLARDPAAVNYAIRRSCEIKAEIVSLDERESGLRALLNLGHTFGHAIENAMGYGAWLHGEAVAAGTLMAADLSRRLQRITSQEVDRIRYLFENTGLPVK
GPRISPERYLESMQLDKKVKEGAIRFILLDSIGKASPGDTVPTPLLLETLSACVADA
``` Answer: [A concise description of the protein with the specified amino acid sequence includes:
Catalyzes the conversion of 3-deoxy-D-arabino-heptulosonate 7-phosphate (DAHP) to dehydroquinate (DHQ).]

Example 2: Protein: ```
MKTERVNVNVNNQPYPIYIGENLLQDKSLLQRHVKGRQVMIVSNETIAAFYLDPLKAIYQDFQCDTFILPDGEQYKTLEYWERILHKLACNHHRDTTLIALGGG
VVGDITGFAAACYQRGVDFIQVPTTLLAQVDASIGGKTAVNHPVGKNLIGAFHQPKAVIIDLNTLNTLPEREFKAGMAEIVKAALIKDEKFFTDLENKMSDLLQ
RNFIFLQAVIKRAAEIKRDIVNADEKERSGERALLNLGHTFAHAIERLLGYGQWLHGEAVSAGLVLAAQLSHRKNLLDFESLQRICRLLTQISLPIHFPKSINA
DELLSAMYMDKKVANERLHLILLEDLGHAVVSDQVDDRELKSFLENG
``` Answer: [A concise description of the protein with the specified amino acid sequence includes:
Catalyzes the conversion of 3-deoxy-D-arabino-heptulosonate 7-phosphate (DAHP) to dehydroquinate (DHQ).]

Example 3: Protein: ```
MNRPGWILLYGPPGVGKTTLGRWLAARLELPFYDLDERIQQVNGRTIPQIFQEEGESGFRQREKSALKELLTLPPGVAALGGGALLDGDNRQLAERCGTVLCLT
AGLQTLLERLGEASQTRPLLKGEDGWQARLSALLEARREHYASFETRLPTDGRTLDETGGEALCALGIFPIRGMERPYRMMVHNGILELAADHLNEIGRSRTAA
LVCDSNTARLYAEKVEKPLTAAGWRVRRCVVPAGEAHKTLQTTADLWAQFVEGGLERGSLVVALGGGVVGDMSGFAAAAFLRGVDWVNLPTTLLAMVDASIGGK
TGVDLPQGKNLVGAFHPPRLVLADPLVLSTLPIGEVRSGMAEVIKHGVIGDPALLDACADGAQGLSGGWEWLVRRAAAVKVRVIEADPYEQGLREVLNFGHTLG
HALEKSSGYRLRHGEAVAIGMVAETRLAERLGIAERGLSGRLAAILSRWGLPVDPPAGLSAEQIRAGLTVDKKRRDGQLRFSLPHRAGQVLHGVIVPAEEALRE
VIG
``` Answer: [A concise description of the protein with the specified amino acid sequence includes:
Catalyzes the specific phosphorylation of the 3-hydroxyl group of shikimic acid using ATP as a
cosubstrate.]

Example 4: Protein: ```
MATPLFHADLTVHTQSHDYPIVITENAIAENSSMASQVAPYITGRQVLIVTNETVAPLYLKALQEELEAQFTVQVCVLPDGEQYKNQSSINQIYDVLMAVHFNR
DVTLIALGGGVIGDMTGFAAASFMRGVNFIQIPTTLLSQVDSSVGGKTGINHPQGKNMIGAFWQPQMVLADMSTLKTLPARELSAGLAEVIKYALIMDAEFLTW
LEHNLPAMMALDLAVLGEAVKRCCQYKADVVAQDERESGVRALLNFGHTFGHVIETHEGYGSWLHGEAVAAGMVQAAELSQKIGWLTSDEVACVKRILSLANLP
ITPPPIEVQTALDLMGHDKKVKHGQIRLILLKSLGEAVLTNDFDPHLLTDVLATHAP
``` Answer: [A brief overview of the protein with the provided amino acid sequence is as follows:
Catalyzes the conversion of 3-deoxy-D-arabino-heptulosonate 7-phosphate (DAHP) to dehydroquinate (DHQ).]
Please analyze and infer the possible function of the target protein based on the given information.
Refer to the functions of similar proteins and perform logical reasoning.
```

```
Ground Truth: Here is a summary of the protein with the given amino acid sequence: Catalyzes the
conversion of 3-deoxy-D-arabino-heptulosonate 7-phosphate (DAHP) to dehydroquinate (DHQ).
```

Figure 5: A Retrieval Augmented Prompt sample on the general function task.

Protein Function Analysis Template
Introductory Statement: You are an assistant that helps with protein function analysis.
Task Description: Task: {Task_Instruction} Target Protein: {Target_Protein}
Retrieved Examples: Below are similar proteins retrieved from a database along with their functions: Example 1: Protein: {Protein1} Answer: [{Function1}] Example 2: Protein: {Protein2} Answer: [{Function2}] Example 3: Protein: {Protein3} Answer: [{Function3}] Example 4: Protein: {Protein4} Answer: [{Function4}] Example 5: Protein: {Protein5} Answer: [{Function5}]
Instruction for Analysis: Please analyze and infer the possible function of the target protein based on the given information. Refer to the functions of similar proteins and perform logical reasoning.

Table 11: Template for Protein Function Analysis

Protein Design Template
Introductory Statement: You are an assistant that helps with protein design.
Task Description: Task: {Task_Instruction} Functional Description: {Functional_Description}
Retrieved Examples: Below are similar tasks retrieved from a database along with their answer: Example 1: Description: {Description1} Answer: [{Design1}] Example 2: Description: {Description2} Answer: [{Design2}] Example 3: Description: {Description3} Answer: [{Design3}] Example 4: Description: {Description4} Answer: [{Design4}] Example 5: Description: {Description5} Answer: [{Design5}]
Instruction for Design: Please design the target protein based on the given information.

Table 12: Template for Protein Design