InstructProtein: Aligning Human and Protein Language via Knowledge Instruction

Zeyuan Wang[1,2,3] **Qiang Zhang**[1,2]* **Keyan Ding**[1,2] **Ming Qin**[1,2,3]
Xiang Zhuang[1,2] **Xiaotong Li**[1,2] **Huajun Chen**[1,2,3]*
[1]College of Computer Science and Technology, Zhejiang University
[2]ZJU-Hangzhou Global Scientific and Technological Innovation Center
[3]AZFT Joint Lab for Knowledge Engine
{yuanzew,qiang.zhang.cs,dingkeyan,qinandming}@zju.edu.cn
{zhuangxiang, 3190104904,huajunsir}@zju.edu.cn

Abstract

Large Language Models (LLMs) have revolutionized the field of natural language processing, but they fall short in comprehending biological sequences such as proteins. To address this challenge, we propose InstructProtein, an innovative LLM that possesses bidirectional generation capabilities in both human and protein languages: (i) taking a protein sequence as input to predict its textual function description and (ii) using natural language to prompt protein sequence generation. To achieve this, we first pre-train an LLM on both protein and natural language corpora, enabling it to comprehend individual languages. Then supervised instruction tuning is employed to facilitate the alignment of these two distinct languages. Herein, we introduce a knowledge graph-based instruction generation framework to construct a high-quality instruction dataset, addressing the annotation imbalance and the absence of instructional signals in the existing protein-text corpus. In particular, the instructions inherit the structural relations between proteins and annotations in knowledge graphs, which empowers our model to engage in the causal modeling of protein functions, akin to the chain-of-thought processes in natural languages. Extensive experiments on bidirectional protein-text generation tasks show that InstructProtein outperforms state-of-the-art LLMs by a large margin. Our code is publicly available at https://github.com/HICAI-ZJU/InstructProtein.

1 Introduction

The landscape of Natural Language Processing (NLP) research, and indeed the broader Artificial Intelligence (AI) community, has recently been revolutionized by generative Large Language Models (LLMs) (Peters et al., 2018; Devlin et al., 2019; Brown et al., 2020), such as ChatGPT (Ouyang et al., 2022). The expansion of parameter size and training corpora has empowered these models to acquire versatile, general-purpose data representations that seamlessly transcend linguistic tasks encompassing comprehension and generation in a multitude of languages. Beyond natural languages (a.k.a., human languages), recent investigations have illuminated the potential of these LLMs to serve as a versatile interface for processing multimodal data, including but not limited to images, videos and speech (Chen et al., 2021; Reed et al., 2022; Gong et al., 2023; Huang et al., 2023).

However, general LLMs fall short of capturing the intricate realm of biological sequences, a domain abundant with its own unique linguistic nuances. The biological sequences, particularly proteins, represent a distinctive facet of what could be referred to as "life language", exerting a significant influence on signal transduction pathways, enzymatic catalysis, and gene regulation (Lee and Yaffe, 2016; Huber, 2001; Durek and Walther, 2008; Luzarowski et al., 2021; Jiang et al., 2022). Existing LLMs like ChatGPT or GPT4 (OpenAI, 2023) fail to accurately model the biological sequences, resulting in limitations on protein understanding and generation (AI4Science and Quantum, 2023).

To unlock the potential within LLMs for deciphering proteins, researchers have put rich efforts into developing protein language models (PLMs) (Alley et al., 2019; Elnaggar et al., 2021; Rives et al., 2021; Rao et al., 2021; Lin et al., 2023). These specialized models are tailored to ingest amino acid sequences as inputs, predict protein functionalities, or even design de novo proteins. Notwithstanding, it is crucial to highlight that while PLMs exhibit competence in comprehending amino acid sequences, they are unable to comprehend the intricacies of human languages. Recent studies (Abdine et al., 2023; Luo et al., 2023) can accept both protein sequences and textual descriptions as input, aiming to enhance protein function

*Corresponding author.

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1114–1136

prediction. Nevertheless, these endeavors to align the realms of protein and human languages are unidirectional and remain in their nascent stages; they fall short of being able to generate protein sequences based on textual instructions. In essence, there exists an unaddressed void in the current landscape of LLMs, wherein the ability to swiftly traverse between human and protein languages.

To enable an LLM to adeptly comprehend both human and protein languages, we contend that the limitations imposed by existing models primarily stem from their training corpora. Notably, many existing models are trained on either human languages or protein sequences, rendering them proficient in only one of these linguistic realms. This unilateral training approach is insufficient to imbue an LLM with a comprehensive vocabulary encompassing both languages. Moreover, it is important to recognize that the existing protein-text corpus used in previous studies (Luo et al., 2023; Abdine et al., 2023; Xu et al., 2023; Taylor et al., 2022) has its limitations. (1) The imbalance of annotations: Researchers tend to focus on well-studied proteins, leading to a significant disparity in the availability of annotations (Kustatscher et al., 2022). Training LLMs directly on such a corpus introduces model bias, which ultimately results in suboptimal performance. (2) The absence of instructional signals: Protein-related textual content is primarily comprised of descriptive narratives, often devoid of instructional signals specifically designed for training LLMs. This inherent disparity obstructs a holistic understanding of a wide range of tasks, ultimately resulting in subpar zero-shot performance (Wei et al., 2022a). **In short, the fundamental hurdle of current LLMs involves curating an elaborate training corpus that seamlessly bridges the gap between human and protein languages.**

In this work, we introduce InstructProtein, a pioneering study that aligns human and protein languages through knowledge instruction, enabling an LLM with bidirectional generation capabilities between these two languages. Specifically, to equip LLMs with the ability to understand protein language, InstructProtein adopts a two-step training approach. It initiates with pre-training on protein and natural language corpora, followed by fine-tuning with the established protein knowledge instruction dataset. To construct such an instruction dataset, we first transform raw protein-text corpora into a structured knowledge graph (KG). Inspired by the idea of chain-of-thoughts, we enrich KG with knowledge causal modeling, which involves establishing causal relationships between triples, indicating causality within annotations. We then propose a debiased sampling strategy to select KG triples, effectively addressing the issue of annotation imbalance. Finally, we mimic KG completion tasks, leverage general LLMs to convert KG triples into instructions, and conduct supervised instruction tuning. Extensive experiments have demonstrated that the introduced protein knowledge instructions significantly improve the performance of LLMs on protein understanding and design tasks. Our contributions can be summarized as follows:

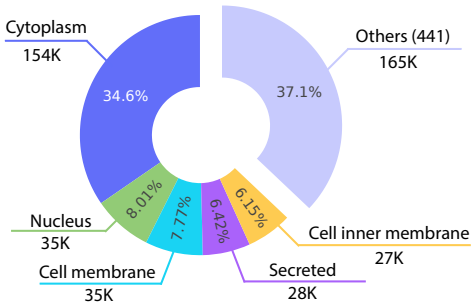


Figure 1: We visualized the top-5 subcellular location categories and their respective proportions, in comparison to the least frequently used annotations, which accounted for only 0.000224%.

1. We propose InstructProtein, an innovative LLM with bidirectional generation between protein and human languages, effectively filling the gap between the two languages.

2. We introduce a protein instruction generation framework based on knowledge graphs, resulting in the first high-quality protein instruction dataset for tuning LLMs.

3. InstructProtein outperforms state-of-the-art LLMs by a substantial margin, representing a crucial advancement toward text-guided protein function prediction and sequence design.

2 A Closer Look at Annotation Imbalance

Much of life science research is dedicated to unraveling the biological functions of proteins. While certain proteins have undergone extensive investigation, there still exist tens of thousands of proteins remaining categorized as understudied. This phenomenon implies an imbalance in protein function annotation. To clearly illustrate this problem, we take the subcellular location as an example, and show its annotation distribution in Figure 1. The

Table 1: The results of querying existing LLMs for factual knowledge. We prompt LLMs to predict subcellular location, but their results are biased to a certain category, which suggests that these LLMs have been contaminated by annotation imbalance.

Models	Prediction			
	Cytoplasm	Nucleus	Cell membrane	Others
OPT	2	115	1691	0
LLaMA	0	1806	2	0
Galactica	1807	1	0	0
Alpaca	1808	0	0	0

results reveal a notable concentration of research attention on proteins residing in the cytoplasm, while other subcellular locations lack comprehensive labeling and study. The annotation imbalance has a detrimental effect on the performance of existing LLMs. To demonstrate this, we collect the same number of proteins in each subcellular location category from UniProtKB (Consortium, 2019), resulting in 1,808 proteins in total, and prompt LLMs to predict the subcellular location. The outcomes of LLMs are presented in Table 1, from which one can observe that these LLMs are biased in a certain category, due to the annotation imbalance in the training corpus of LLMs.

3 InstructProtein

This section presents the method of InstructProtein. We first pre-train it in a self-supervised manner on natural language and protein sequence datasets respectively, and then conduct supervised tuning using the created knowledge instruction dataset.

3.1 Multilingual Pre-Training

InstructProtein is designed to comprehend both the protein and human languages. An intuitive approach involves incrementally pre-training an LLM using the protein corpus $\mathcal{P}$ and text sequences $\mathcal{T}$. Given $\mathcal{X} = \mathcal{P} \cup \mathcal{T}$ and $\{x_1, x_2, \ldots, x_n\} \in \mathcal{X}$, the training objective of a generative LLM (e.g., OPT (Zhang et al., 2022a)) is defined as

$$L(\mathcal{X}) = \sum_i \log P(x_i | x_{i-k}, \ldots, x_{i-1}; \theta), \quad (1)$$

where the prediction of each token depends on previous tokens $x_{<i}$, k is the context window size, and the conditional probability P is modeled using a neural network parameterized by θ.

3.2 Instruction Tuning

After pre-training, the model acquires an extensive comprehension of both natural language and protein sequences; however, it still falls short in alignment between these two different languages. We fill this gap through supervised instruction tuning.

3.2.1 Knowledge Instruction Generation

We propose an instruction generation method based on KGs and LLMs, aiming to construct a factual, logical, diverse, and well-balanced protein instruction dataset. Figure 2 illustrates the pipeline of three kinds of instruction generation frameworks. Conventional approaches directly utilize LLMs to generate instruction data from seed tasks or raw documents, which may introduce hallucination and bias from internal knowledge of LLMs. In the proposed method, KGs are incorporated as intermediaries to address these limitations. In specific, a KG encompassed with knowledge causal modeling is constructed to provide factual protein knowledge, based on which a debiased sampling strategy is proposed to pick KG triples. It is worth noting that LLMs simply need to accurately translate the triples into instructional data, without assuming any prior knowledge about proteins.

KG Construction. We use UniProtKB as our data source to construct the protein knowledge graph denoted as $\mathcal{G} = \{\mathcal{P}, \mathcal{R}, \mathcal{T}\}$. Here, $\mathcal{P}$, $\mathcal{R}$, and $\mathcal{T}$ are sets of protein sequences, relations, and textual annotations. Note that the textual description of proteins in UniProtKB is structured, making it easy to transform them into a knowledge graph. In pursuit of a high-quality instruction dataset, we augment KG to provide informative relationships. With chain-of-thoughts (Wei et al., 2022b), we recognize that a logical chain also exists within protein annotations. For example, the biological processes in which a protein can participate are intricately linked to its molecular function and subcellular location, with the molecular function itself being influenced by the protein's domain. To represent this causal chain of protein knowledge, we introduce a novel concept called Knowledge Causal Modeling (KCM). Specifically, a knowledge causal model comprises multiple interconnected triples organized in a directed acyclic graph, where the edge direction signifies causal relationships. This graph organizes the triples, moving from the micro-level, encompassing characteristics of protein sequences (e.g., domains), to the macro-level, encompassing biological functions. In Figure 3, we show an example of KCM retrieved from InterPro (Paysan-Lafosse et al., 2023) based on a given triple.

KG Triple Sampling. To generate instruction

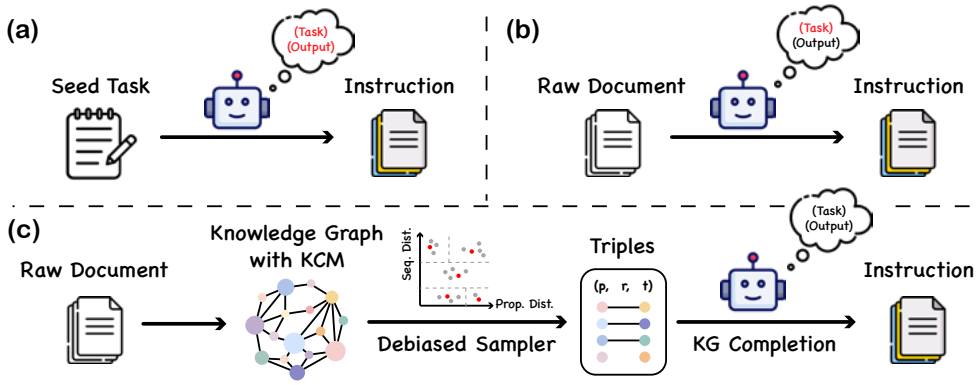


Figure 2: Overview of Instruction generation methods. The red text represents what relies on the internal knowledge of LLMs. (a) Given a seed task, prompting an LLM to produce new instruction data.(b) Utilizing LLMs to generate the instruction data corresponding to the raw documents. (c) Our KG-based instruction generation framework. We first construct a KG with knowledge causal modeling, and introduce a debiased sampler to pick informative triples, which are then translated into instruction data through the use of LLMs in conjunction with KG completion tasks.

data, we need to sample triples from the constructed KG. Considering the annotation imbalance problem in the KG, we propose a debiased sampling strategy as an alternative to uniform sampling. In specific, we first cluster proteins based on their sequence and property similarities, and then uniformly pick triples in each cluster. For sequence similarity, we employ MMseqs2 (Steinegger and Söding, 2017) to calculate the editing distance $d_s(\cdot,\cdot)$ (see Appendix B.1.1). For property similarity, since the protein properties are extensive and many of them remain unexplored, we only consider the known annotations in KG. Specifically, given an annotation t and a relation r, we denote $C_t = \{p : p \in \mathcal{P} \wedge (p,r,t) \in \mathcal{G}\}$ and $C_{/t} = \{p : p \in \mathcal{P} \wedge (p,r,t) \notin \mathcal{G}\}$ are the protein set based on the presence or absence of t. The basic idea is to maximize agreement within C_t and minimize agreement between C_t and $C_{/t}$, via optimizing protein KG embeddings. In practice, we minimize a margin-based ranking criterion over the knowledge graph:

$$\mathcal{L} = [\gamma + d_p(p_t, t+r) - d_p(p_{/t}, t+r)], \quad (2)$$

where $p_t \in C_t$, $p_{/t} \in C_{/t}$, γ is the margin, and $d_p(\cdot,\cdot)$ is a dissimilarity measure of properties, which is implemented as the ℓ_2-norm.

We define the identity threshold of sequence and property similarities as δ_p and δ_s, respectively. We denote two proteins to be similar $p_1 \simeq p_2$ as $d_s(p_1,p_2) < \delta_s$ and $d_p(p_1,p_2) < \delta_p$. $\mathcal{C} = \{C_1,\ldots,C_m\}$ represents the aggregation of proteins with m clusters, and the cluster C_i can be formulated as: $C_i = \{p : p \simeq p_{C_i}^{\text{center}}\}$, where $p_{C_i}^{\text{center}}$ is the center protein of C_i. Then, the probability of sampling a triple (p,r,t) is:

$$P((p,r,t)) = \frac{1}{m} \times \frac{1}{||C_i||} \times \frac{1}{||p||}, \quad (3)$$

where $p \in C_i$, $||C_i||$ denotes the size of C_i, and $||p||$ are the number of annotations on p.

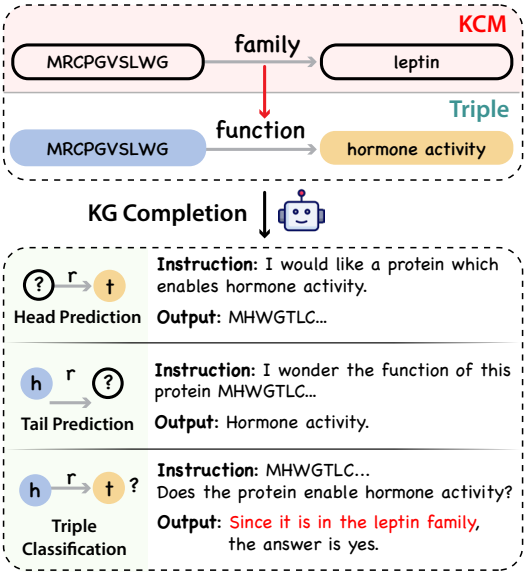


Figure 3: An example of converting a KG triple to instructions. Given a triple with corresponding KCM, we use an LLM in conjunction with KG completion tasks to generate factual, logical, and diverse instructions.

KG Triple to Instruction. By employing the debiased sampling strategy, a large number of well-balanced KG triples can be sampled. We then translate these triples into instructions. While the generation of creative tasks requires domain knowledge, the KG completion tasks offer a comprehensive template for proposing domain-specific tasks based on triples. Therefore, we simulate KG completion, and employ general LLMs (e.g., ChatGPT) to transform KG triples with retrieved KCM into

Table 2: Performance comparison between InstructProtein and baselines (open-source LLMs) on Held-In and Held-Out protein sequence understanding tasks.

Models	Params.	GO-BP		GO-MF		GO-CC		Location		MIB
		ACC	AUPR	ACC	AUPR	ACC	AUPR	Bin	Sub	
OPT	1.3B	51.83	64.76	56.10	74.50	51.94	71.90	57.52	29.06	49.40
LLaMA	7.0B	56.96	61.85	54.58	58.06	51.57	53.53	57.52	29.14	50.00
Alpaca	7.0B	61.69	65.13	59.37	73.02	57.98	61.71	57.52	18.32	50.38
Galactica	1.3B	55.11	57.08	61.30	61.93	51.17	54.54	57.52	18.32	51.58
Mol-Instructions	7.0B	50.00	49.15	50.00	47.45	50.00	47.72	57.52	18.36	50.00
BioMedGPT	10B	50.31	50.82	51.02	50.81	49.41	49.39	59.51	56.39	54.42
BioT5	252M	53.23	52.96	50.02	50.68	52.94	49.74	65.58	42.15	49.77
InstructProtein	1.3B	**71.49**	**83.16**	**85.83**	**93.68**	**79.79**	**86.37**	**85.19**	**70.79**	**62.68**

instructions, which contain an instruction describing the protein-related task about and an output result reflecting a correct response of the instruction. Figure 3 shows an example of converting the triple to instructions. The detailed implementation is depicted in Figure 7 and Table 6 in Appendix.

3.2.2 Tuning LLMs with Instructions.

Instruction tuning involves further training LLMs in a supervised manner on an instruction dataset comprising of **(instruction, output)**, bridging the gap between the LLMs' next-word prediction objective and users' goal of ensuring adherence to human instructions. With the proposed knowledge instruction dataset $\mathcal{I}$, we finetune the pre-trained LLM to align the protein and human languages. Given an instruction $Z \in \mathcal{I}$ and its tokens $\{x_1, x_2, \ldots, x_n\} \in Z$, the training objective is the same as that defined in Eq.(1).

4 Experiments

4.1 Experimental Setup

The pre-training corpus contains protein sequences from UniRef100 (Suzek et al., 2015) and sentences from PubMed abstracts. Following the methodology described in Section 3.2.1, We first constructed a protein knowledge graph based on UniProt/Swiss-Prot(Consortium, 2019). Specifically, we select nine property fields: biological process, molecular function, cellular component, family, superfamily, domain, conserved site, active site, and binding site. The resulting KG consists of 464,333 proteins as head entities, 58,725 annotations as tail entities, and 5,207,841 triples. Knowledge causal modeling is sourced from the InterPro (Paysan-Lafosse et al., 2023) and Gene Ontology (Aleksander et al., 2023) database, containing 30,446 causal relationships. We then sample 2.8 million triples and use ChatGPT to convert these triples into instructions. Particular care is required to prevent potential data contamination between training and evaluation data. We use mmseqs2 to cluster proteins with an identity surpassing the 70% threshold, then remove the clusters containing the proteins in the test set, for a total of 19,455 sequences. Detailed experimental setups and discussion on addressing potential data contamination are described in Appendix B.2.

Table 3: Performance comparison between InstructProtein and closed-source LLMs on Held-In and Held-Out protein sequence understanding tasks.

Models	GO(BP)	GO(BP)	GO (CC)	Location(Sub)
ChatGPT	44.69	48.26	43.82	15.38
Claude-2	54.73	54.04	52.56	23.15
GPT-4	47.12	57.45	51.29	18.66
InstructProtein	**71.49**	**85.83**	**79.79**	**70.79**

4.2 Protein Sequence Understanding

Datasets and Metrics. We evaluate LLMs on three widely-used protein property classification tasks: (1) Protein Function Annotation, aiming to predict the correct functions of proteins. We choose Gene Ontology (GO) dataset (Gligorijević et al., 2021), which has three branches: molecular function (MF), biological process (BP), and cellular component (CC). This is a "Held-In" task as the training instructions are derived from the GO dataset. (2) Protein Localization Prediction, which involves the prediction of the subcellular location of a given protein. We address two subproblems from DeepLoc (Almagro Armenteros et al., 2017), the subcellular localization prediction (Abbr., Sub) with 10 location categories and the binary localization prediction (Abbr., Bin) with 2 location categories. This is also a "Hold-In" task since cellular component exist in our instruction data. (3) Metal Ion Binding (MIB) Prediction, a binary classification task where the model needs to determine whether there are metal ion-binding sites in the

protein, which is a "Held-Out" task as the proteins and labels are sourced from Protein Data Bank (PDB) (wwp, 2019). We use the dataset from Hu et al. (2022). Detailed downstream task setting can be found in Appendix C

Baselines. We adopt seven state-of-the-art open-source LLMs as the baselines. OPT (Zhang et al., 2022a) and LLaMA (Touvron et al., 2023) are trained on massive text corpus. Alpaca (Taori et al., 2023) and Mol-Instructions (Fang et al., 2023) refer to LLaMA-based LLMs fine-tuned with other human/protein language datasets. Galactica (Taylor et al., 2022), BioMedGPT (Luo et al., 2023), and BioT5 (Pei et al., 2023) are domain-specific LLMs, which are trained on a large corpus of humanity's scientific knowledge, such as research papers about proteins and genes. Note that, the training corpora of Mol-Instructions, Galactica, BioMedGPT, BioT5 all contain annotations from Swiss-Prot.

Results. We present the evaluation results in Table 2. Compared with all baselines, InstructProtein achieves new state-of-the-art performance on all tasks. There are two key observations. First, InstructProtein clearly outperforms the LLMs (i.e., ChatGPT, LLaMA, Alpaca) which are stemmed from natural language training corpora. These results demonstrate that training with the corpus where proteins and natural language coexist is beneficial to LLMs, enhancing their proficiency in protein language understanding. Second, InstructProtein performs consistently better than Galactica, BioMedGPT, Mol-Instructions, and BioT5 on Held-In and Held-Out tasks, proving that the model has the ability to generalize both protein sequences and protein-related tasks. The instruction templates of Mol-Instructions are not adequately diverse, thus unable to understand the tasks in the GO and MIB benchmarks, leading to all negative predictions. It is worth noting that in the protein subcellular localization (Bin) task, there exists a severe bias in LLMs, leading to the classification of all proteins into a single group and resulting in the same accuracy of 57.52%.

Since the closed-source models such as ChatGPT, Claude-2 and GPT4 often refused to follow the instructions as illustrated in Appendix 15, we only consider the occasional cases when they responded to and report their results in Table 3. We discover that these models' predictions solely encompassed cytoplasm, nucleus, cell membrane, and secreted locations in the protein localization prediction task. This finding underscores that annotation imbalance impacts even these closed-source models, further emphasizing the significance of the high-quality instruction dataset. We provide case studies in Figure 16 in Appendix.

Table 4: Accuracy of instruction-protein pairing.

Models	Fold Rank		
	Fold	SuperFamily	Family
OPT	7.79	6.45	6.68
LLaMA	9.33	5.90	10.30
Alpaca	5.43	3.90	4.71
Galactica	11.00	10.12	10.37
Mol-Instructions	12.81	12.57	12.44
BioMedGPT	-	-	-
BioT5	14.20	12.91	27.83
InstructProtein	**55.57**	**65.07**	**79.24**

4.3 Protein Sequence Design

Generating proteins following human instructions is a highly exciting area of research. With the incorporation of the protein as part of the language capabilities in LLMs, InstructProtein is capable of generating protein sequences. However, the lack of standardized computational metrics to properly assess the quality of proteins generated by LLMs poses challenges for advancing protein generation models. In this study, we present our endeavor to build a computational evaluation framework.

4.3.1 Instruction-Protein Pairing

Datasets and Metrics. We design an instruction-protein pairing task to assess the consistency between the instruction and the generated protein. Specifically, we employ the dataset proposed by Hou et al. (2018) to provide fold-related instructions and proteins. Given a protein p and the corresponding instruction Z_0, we randomly sample other n instructions $\{Z_1, Z_2, ..., Z_n\}$ ($n = 9$ in this experiment), and the likelihood $\mathcal{L}$ of the protein given the various instructions is computed. The minimization of $\mathcal{L}(p|Z_i)$ at $i = 0$ signifies a correct pairing, and vice versa.

Results. Table 4 reports the accuracy of the instruction-protein pairing task. One can observe that InstructProtein surpasses the baselines by a large margin. BioMedGPT focuses solely on converting proteins to texts and lacks protein design capabilities. Galactica exhibits limited zero-shot performance in aligning instructions with proteins, since it is trained with narrative protein corpus. Mol-Instructions lacks pre-training on protein corpora, which makes it difficult for the model to distinguish the nuances of proteins, resulting in poor

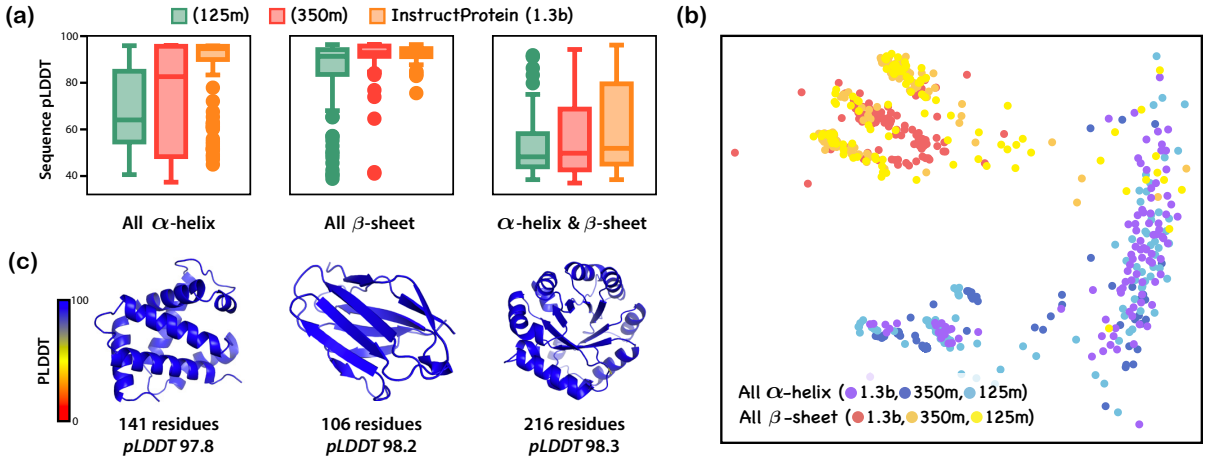


Figure 4: Visualization of structure instruction-based protein sequence de novo design. We prompt our models with different scales (125m, 350m and 1.3b) to generate three kinds of proteins (all α-helix, all β-sheet, and a combination of α-helix and β-sheet), respectively. (a) We visualize the pLDDT of generated sequences predicted by AlphaFold2 to assess the protein foldability. (b) The embeddings of sequences prompted with all α-helix and all β-sheet instructions, which are extracted from ESM2 and visualized by the MDS algorithm. (c) The structure of generated proteins with the highest confidence in each class.

results. These results confirm the superiority of our model in instruction-following for protein design.

4.3.2 Protein Sequence De Novo Design

Designing proteins with specified structures. We investigate whether InstructProtein could generate new protein sequences that are individually valid and consistent with instructions. SCOPe (Chandonia et al., 2022) classifies protein structures according to the content and organization of secondary structures, including all α-helix, all β-sheet, and the combination of α-helix and β-sheet. We sample 100 sequences from each class and assess the foldability of individual sequences by predicting their corresponding structures using ColabFold (Mirdita et al., 2022; Jumper et al., 2021) and computing the average predicted local distance difference test (pLDDT) across the whole structure (Figure 4 (a)). pLDDT increases with model scale, suggesting that scaling up the parameter size results in the generation of sequences with fewer intrinsically disordered regions. We leverage ESM2 (Lin et al., 2023) as a feature extractor to obtain the generated all α-helix and all β-sheet protein representations, which are then visualized using multi-dimensional scaling (MDS) algorithm (Kruskal, 1964) (Figure 4 (b)). We observe that the representations are divided into two groups according to instructions, indicating the instruction-following ability of the proposed model. We visualize the predicted structure of the proteins with the highest confidence in each class (Figure 4 (c)). To evaluate the novelty of the generated sequences, we utilize HHblits to search for homologs against the Uniclust30 dataset. Our analysis revealed that the highest identity of each alignment ranged from 0.313 to 0.880, with a one-standard deviation range of 0.437 to 0.732, demonstrating the generated sequences are not merely based on mutation or the combination of existing sequences but exhibit a degree of novelty. These results demonstrate that InstructProtein establishes a close correlation between natural language and protein language, verifying the effectiveness of protein de novo design based on structure-related instructions.

Designing proteins binding with specified ligands. To verify the ability to follow function-related instructions, we employ InstructProtein to design heme binders, which are proteins capable of binding to a specific compound, and visualize 3D structures of three generated proteins. In Figure 5, we present the docking result (docked by DiffDock (Corso et al., 2023)), the binding affinity (predicted by Smina (Koes et al., 2013; Trott and Olson, 2010), the lower the better), and the pLDDT score (predicted by ColabFold; the higher the absolute value, the better). We can observe the resulting proteins exhibit notable binding affinity, confirming the efficacy of InstructProtein in heme binder design. We provide more case studies and comparisons with ChatGPT, Claude-2, and Mol-Instructions in Appendix E.

4.4 Ablation Study

We conduct ablation studies on the sampling strategy and knowledge causal modeling (KCM) used in our knowledge instruction generation method.

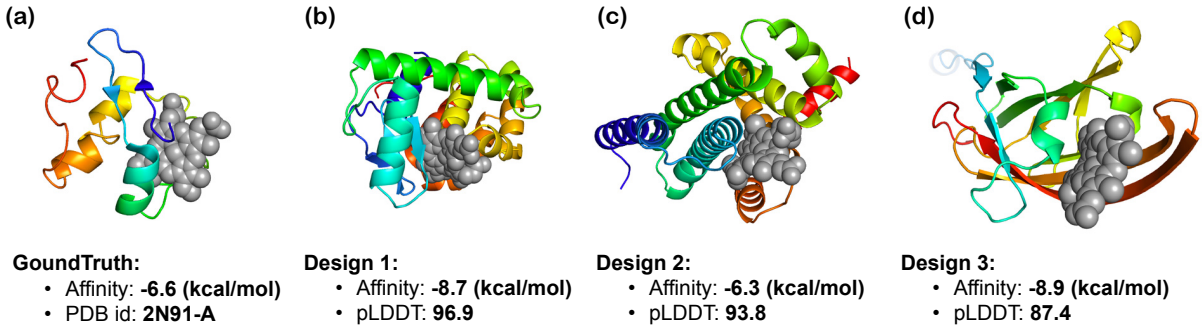


Figure 5: Visualization of functional instruction-based protein sequence de novo design. We prompt our model with the instruction "*I would like a protein that enables heme binding*". (a) is the ground-truth protein that binds with heme. (b), (c) and (d) are generated proteins with decent binding affinity.

Table 5: Ablation of Knowledge Instruction.

Sampling	KCM	Location (Sub)	GO (MF)
Unclustering	No	58.12	85.58
Seq. only	No	62.77	83.70
Seq. & Prop. (Edit)	No	66.57	84.34
Seq. & Prop. (KGE)	No	69.95	**85.92**
Seq. & Prop. (KGE)	Yes	**70.79**	85.83

From the results in Table 5, we observe that clustering similar proteins in annotation imbalance-related tasks (Location) can effectively improve model performance. However, for tasks where annotation imbalance is not significant (GO), the clustering method based on sequence alone degrade model performance, which is reasonable because this method reduces the frequency of hard samples (proteins with similar sequences but different functions). This problem can be avoided by considering both sequence and property similarities. We compare property clustering methods based on KGE distance and edit distance, and the results prove that KGE has a stronger ability to model property similarity. We also observe that the causal relationship between annotations introduced by KCM improves the performance.

5 Related Works

Large Language Models (LLMs) have achieved breakthrough performance in NLP (Brown et al., 2020; Rae et al., 2021; Hoffmann et al., 2022; Black et al., 2022; Zhang et al., 2022a; Chowdhery et al., 2022; Touvron et al., 2023). However, these LLMs are primarily tailored for human language comprehension, which limits their utility in decoding protein language. To align these two distinct languages, multimodal approaches (Taylor et al., 2022; Abdine et al., 2023; Luo et al., 2023; Fang et al., 2023) train LLMs on a corpus where natural and protein language co-exist. For example, Luo et al. (2023) integrate protein encoders into LLMs within an encoder-decoder framework. Notwithstanding, these architectures predominantly exhibit a unidirectional cross-modal capability, focusing solely on converting protein language to textual description. Taylor et al. (2022) treats protein language and human language as a unified modality. However, the lack of instruction signals and annotation imbalance hinder the alignment.

Instruction Tuning is a supervised approach to align language models with user intention (Mishra et al., 2022; Wang et al., 2022; Wei et al., 2021; Ouyang et al., 2022). It is worth noting that acquiring large-scale instruction data can be a resource-intensive and time-consuming endeavor, thereby motivating the exploration of automatic data generation techniques. A prevalent strategy (Anaby-Tavor et al., 2020; Andreas, 2020; Kaushik et al., 2019) involves augmenting existing datasets. Alternatively, several fully automatic datasets have been proposed to eliminate the need for labeled data. Schick and Schütze (2021); Ye et al. (2022) advocate for leveraging pre-trained language models to generate comprehensive labeled datasets from scratch, tailored to predefined tasks. Honovich et al. (2023a),Wang et al. (2023) and Honovich et al. (2023b) used pre-trained LLMs to automatically construct instructions by a handful of examples. Li et al. (2023) proposes to construct instruction data in a way that a LLM generates tasks based on outputs. However, these methods may introduce hallucination and bias into the instruction data. Fang et al. (2023) construct a template-based instruction dataset, which lacks diversity. Inspired by KG-enhanced LLMs (Sun et al., 2021; Liu et al., 2020; Zhang et al., 2022b), we propose knowledge instruction that can construct factual, logical, and diverse instruction datasets.

6 Conclusion

InstructProtein explores the feasibility of bidirectional generation between human and protein languages within a single large language model. Our approach involved the transformation of a raw protein-text corpus into a structured knowledge graph, from which KG triples were sampled and converted into instructions. This KG-based instruction generation method resulted in a high-quality instruction dataset, facilitating the LLM to align protein language with human language.

7 Limitations

It is important to acknowledge that there are some limitations inherent in our model. One such limitation, shared with large language models, is that InstructProtein encounters challenges with handling numerical values. In the field of protein modeling, a significant proportion is dedicated to quantitative tasks, including the determination of 3D structure, stability assessment, and fitness evaluation. Inadequate quantitative language modeling hinders the understanding of proteins and more granular controllable generation.

In the future, we will incorporate a broader spectrum of instructions, including quantitative descriptions, empowering our model to provide quantitative outputs. These developments will open up new avenues for further advancing the integration of protein and human languages, as well as expanding its practical utility in diverse applications.

Acknowledgements

We would like to thank all anonymous reviewers for their insightful and invaluable comments. This work is supported by National Natural Science Foundation of China (62302433, 62301480, U23A20496), Zhejiang Provincial "Jianbing" "Lingyan" Research and Development Program of China (2024C01135), Hangzhou West Lake Pearl Project Leading Innovative Youth Team Project (TD2023017), Zhejiang Provincial Natural Science Foundation of China (LQ24F020007) and CCF-Tencent Rhino-Bird Fund (RAGR20230122).

References

2019. Protein data bank: the single global archive for 3d macromolecular structure data. *Nucleic acids research*, 47(D1):D520–D528.

Hadi Abdine, Michail Chatzianastasis, Costas Bouyioukos, and Michalis Vazirgiannis. 2023. Prot2Text: Multimodal Protein's Function Generation with GNNs and Transformers. *arXiv preprint arXiv:2307.14367*.

Microsoft Research AI4Science and Microsoft Azure Quantum. 2023. The impact of large language models on scientific discovery: a preliminary study using gpt-4.

Suzi A Aleksander, James Balhoff, Seth Carbon, J Michael Cherry, Harold J Drabkin, Dustin Ebert, Marc Feuermann, Pascale Gaudet, Nomi L Harris, et al. 2023. The Gene Ontology knowledgebase in 2023. *Genetics*, 224(1):iyad031.

Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. 2019. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322.

José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. 2017. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395.

Ateret Anaby-Tavor, Boaz Carmeli, Esther Goldbraich, Amir Kantor, George Kour, Segev Shlomov, Naama Tepper, and Naama Zwerdling. 2020. Do Not Have Enough Data? Deep Learning to the Rescue! In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7383–7390.

Jacob Andreas. 2020. Good-Enough Compositional Data Augmentation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7556–7566.

Sidney Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. GPT-NeoX-20B: An Open-Source Autoregressive Language Model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 95–136, virtual+Dublin. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, and others Askell. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

John-Marc Chandonia, Lindsey Guan, Shiangyi Lin, Changhua Yu, Naomi K Fox, and Steven E Brenner. 2022. SCOPe: improvements to the structural classification of proteins–extended database to facilitate variant interpretation and machine learning. *Nucleic acids research*, 50(D1):D553–D559.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Pinto, Jared Kaplan, Harrison Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating Large Language Models Trained on Code.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2022. PaLM: Scaling Language Modeling with Pathways. *arXiv preprint arXiv:2204.02311*.

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

Gabriele Corso, Hannes Stärk, Bowen Jing, Regina Barzilay, and Tommi Jaakkola. 2023. Diffdock: Diffusion Steps, Twists, and Turns for Molecular Docking. *International Conference on Learning Representations (ICLR)*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *North American Chapter of the Association for Computational Linguistics*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Elie Dolgin. 2017. The most popular genes in the human genome. *Nature*, 551(7681):427–432.

Pawel Durek and Dirk Walther. 2008. The integrated analysis of metabolic and protein interaction networks reveals novel molecular organizing principles. *BMC systems biology*, 2(1):1–20.

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.

Yin Fang, Xiaozhuan Liang, Ningyu Zhang, Kangwei Liu, Rui Huang, Zhuo Chen, Xiaohui Fan, and Huajun Chen. 2023. Mol-instructions: A large-scale biomolecular instruction dataset for large language models.

Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. 2021. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168.

Tao Gong, Chengqi Lyu, Shilong Zhang, Yudong Wang, Miao Zheng, Qian Zhao, Kuikun Liu, Wenwei Zhang, Ping Luo, and Kai Chen. 2023. MultiModal-GPT: A Vision and Language Model for Dialogue with Humans.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models.

Or Honovich, Thomas Scialom, Omer Levy, and Timo Schick. 2023a. Unnatural instructions: Tuning language models with (almost) no human labor. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14409–14428, Toronto, Canada. Association for Computational Linguistics.

Or Honovich, Uri Shaham, Samuel R. Bowman, and Omer Levy. 2023b. Instruction induction: From few examples to natural language task descriptions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1935–1952, Toronto, Canada. Association for Computational Linguistics.

Jie Hou, Badri Adhikari, and Jianlin Cheng. 2018. Deepsf: deep convolutional neural network for mapping protein sequences to folds. *Bioinformatics*, 34(8):1295–1303.

Mingyang Hu, Fajie Yuan, Kevin K Yang, Fusong Ju, Jin Su, Hui Wang, Fei Yang, and Qiuyang Ding. 2022. Exploring evolution-aware &-free protein language models as protein function predictors. In *Advances in Neural Information Processing Systems*.

Shaohan Huang, Li Dong, Wenhui Wang, Yaru Hao, Saksham Singhal, Shuming Ma, Tengchao Lv, Lei Cui, Owais Khan Mohammed, Barun Patra, et al. 2023. Language Is Not All You Need: Aligning Perception with Language Models.

Armin Huber. 2001. Scaffolding proteins organize multimolecular protein complexes for sensory signal transduction. *European Journal of Neuroscience*, 14(5):769–776.

Yining Jiang, Batiste Thienpont, Vinay Sapuru, Richard K Hite, Jeremy S Dittman, James N Sturgis, and Simon Scheuring. 2022. Membrane-mediated protein interactions drive membrane protein organization. *Nature Communications*, 13(1):7373.

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.

Divyansh Kaushik, Eduard Hovy, and Zachary Lipton. 2019. Learning the Difference That Makes A Difference with Counterfactually-Augmented Data. In *International Conference on Learning Representations*.

David Ryan Koes, Matthew P Baumgartner, and Carlos J Camacho. 2013. Lessons learned in empirical

scoring with smina from the csar 2011 benchmarking exercise. *Journal of chemical information and modeling*, 53(8):1893–1904.

Joseph B Kruskal. 1964. Nonmetric multidimensional scaling: a numerical method. *Psychometrika*, 29(2):115–129.

Georg Kustatscher, Tom Collins, Anne-Claude Gingras, Tiannan Guo, Henning Hermjakob, Trey Ideker, Kathryn S Lilley, Emma Lundberg, Edward M Marcotte, Markus Ralser, et al. 2022. An open invitation to the Understudied Proteins Initiative. *Nature Biotechnology*, 40(6):815–817.

Michael J Lee and Michael B Yaffe. 2016. Protein regulation in signal transduction. *Cold Spring Harbor perspectives in biology*, 8(6):a005918.

Xian Li, Ping Yu, Chunting Zhou, Timo Schick, Luke Zettlemoyer, Omer Levy, Jason Weston, and Mike Lewis. 2023. Self-alignment with instruction backtranslation.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. 2023. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130.

Weijie Liu, Peng Zhou, Zhe Zhao, Zhiruo Wang, Qi Ju, Haotang Deng, and Ping Wang. 2020. K-BERT: Enabling Language Representation with Knowledge Graph. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 2901–2908.

Yizhen Luo, Jiahuan Zhang, Siqi Fan, Kai Yang, Yushuai Wu, Mu Qiao, and Zaiqing Nie. 2023. BioMedGPT: Open Multimodal Generative Pretrained Transformer for Biomedicine. *arXiv preprint arXiv:2308.09442*.

Marcin Luzarowski, Rubén Vicente, Andrei Kiselev, Mateusz Wagner, Dennis Schlossarek, Alexander Erban, Leonardo Perez de Souza, Dorothee Childs, Izabela Wojciechowska, Urszula Luzarowska, et al. 2021. Global mapping of protein–metabolite interactions in saccharomyces cerevisiae reveals that ser-leu dipeptide regulates phosphoglycerate kinase activity. *Communications Biology*, 4(1):181.

Milot Mirdita, Konstantin Schütze, Yoshitaka Moriwaki, Lim Heo, Sergey Ovchinnikov, and Martin Steinegger. 2022. ColabFold: Making Protein folding accessible to all. *Nature Methods*.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. 2022. Cross-Task Generalization via Natural Language Crowdsourcing Instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3470–3487.

OpenAI. 2023. GPT-4 Technical Report.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35:27730–27744.

Typhaine Paysan-Lafosse, Matthias Blum, Sara Chuguransky, Tiago Grego, Beatriz Lázaro Pinto, Gustavo A Salazar, Maxwell L Bileschi, Peer Bork, Alan Bridge, Lucy Colwell, et al. 2023. InterPro in 2022. *Nucleic Acids Research*, 51(D1):D418–D427.

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.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *North American Chapter of the Association for Computational Linguistics*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling Language Models: Methods, Analysis & Insights from Training Gopher.

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. 2019. Evaluating protein transfer learning with tape. *Advances in neural information processing systems*, 32.

Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. 2021. MSA Transformer. In *International Conference on Machine Learning*, pages 8844–8856. PMLR.

Scott Reed, Konrad Zolna, Emilio Parisotto, Sergio Gómez Colmenarejo, Alexander Novikov, Gabriel Barth-maron, Mai Giménez, Yury Sulsky, Jackie Kay, et al. 2022. A Generalist Agent. *Transactions on Machine Learning Research*. Featured Certification, Outstanding Certification.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. 2021. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118.

Timo Schick and Hinrich Schütze. 2021. Generating Datasets with Pretrained Language Models. In *Proceedings of the 2021 Conference on Empirical Meth-*

ods in Natural Language Processing, pages 6943–6951.

Martin Steinegger and Johannes Söding. 2017. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology*, 35(11):1026–1028.

Yu Sun, Shuohuan Wang, Shikun Feng, Siyu Ding, Chao Pang, Junyuan Shang, Jiaxiang Liu, Xuyi Chen, Yanbin Zhao, Yuxiang Lu, et al. 2021. ERNIE 3.0: Large-scale Knowledge Enhanced Pre-training for Language Understanding and Generation. *arXiv preprint arXiv:2107.02137*.

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

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

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 arXiv:2211.09085*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. LLaMA: Open and Efficient Foundation Language Models. *arXiv preprint arXiv:2302.13971*.

Oleg Trott and Arthur J Olson. 2010. AutoDock Vina: Improving the speed and accuracy of docking with a new scoring function, efficient optimization, and multithreading. *Journal of computational chemistry*, 31(2):455–461.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, et al. 2022. Super-NaturalInstructions: Generalization via Declarative Instructions on 1600+ NLP Tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. 2021. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022a. Finetuned Language Models are Zero-Shot Learners. In *International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Minghao Xu, Xinyu Yuan, Santiago Miret, and Jian Tang. 2023. Protst: Multi-Modality Learning of Protein Sequences and Biomedical Texts. *arXiv preprint*.

Jiacheng Ye, Jiahui Gao, Qintong Li, Hang Xu, Jiangtao Feng, Zhiyong Wu, Tao Yu, and Lingpeng Kong. 2022. ZeroGen: Efficient Zero-shot Learning via Dataset Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11653–11669.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022a. OPT: Open Pre-trained Transformer Language Models.

Taolin Zhang, Chengyu Wang, Nan Hu, Minghui Qiu, Chengguang Tang, Xiaofeng He, and Jun Huang. 2022b. Dkplm: Decomposable Knowledge-Enhanced Pre-trained Language Model for Natural Language Understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11703–11711.

A Detailed Analysis of Protein Understudying Problems

Much of life science research is dedicated to unraveling the biological functions of proteins. While certain proteins, such as the well-studied tumor suppressor p53 (Dolgin, 2017), have undergone extensive investigation, tens of thousands of proteins remain categorized as understudied. This classification implies that their biological functions are poorly elucidated, and they lack comprehensive annotation of their molecular properties.

In Figure 6, we present an analysis conducted on UniProtKB/Swiss-Prot, a highly reputable and manually curated protein knowledge repository. Figure 6 (a) depicts the relationship between the distribution of proteins and their annotation scores. These results emphasize the substantial variation in protein distribution corresponding to different annotations. This variance implies that the annotation of proteins is biased. To illustrate this problem more clearly, we analyze the subcellular location annotation. Figure 6 (b) illustrates the distribution of such annotations. The data reveals a notable concentration of research attention on proteins residing in the cytoplasm, with other subcellular locations significantly lacking in comprehensive labeling.

B Detailed Method

B.1 In-Context Examples

Knowledge Instruction relies on examples to teach language models understand how to convert information extracted from the knowledge graph into instruction data. Here we provide our example (Figure 7). We notice that when only two examples of different expressions are provided for each KGC task, the language capabilities of LLMs are activated, generating a variety of instruction data as illustrated in Table 6

B.1.1 Sequence Distance Algorithm

We denote $A = a_1 a_2 \dots a_n$ and $B = b_1 b_2 \dots b_n$ as two sequences to be aligned, where n and m are the lengths of A and B, respectively. Before calculating the editing distance, we have to determine the substitution matrix to calculate the replacement score $s(\cdot, \cdot) \in (0, 1]$ and the gap penalty scheme W_k, where k is the gap length. Then the distance matrix H can be formulated as:

$$\begin{aligned} H_{i,j} = \min\{&H_{i-1,j-1} + s(a_i, b_j); \\ &H_{i-k,j} - W_k; \\ &H_{i,j-1} - W_1; \\ &1\} \end{aligned} \tag{4}$$

where $H_{k,0} = H_{0,l} = 0$ for $0 \le k \le n$ and $0 \le l \le m$. We leverage $H_{n,m}/\max(n, m)$ as the sequence distance between A and B.

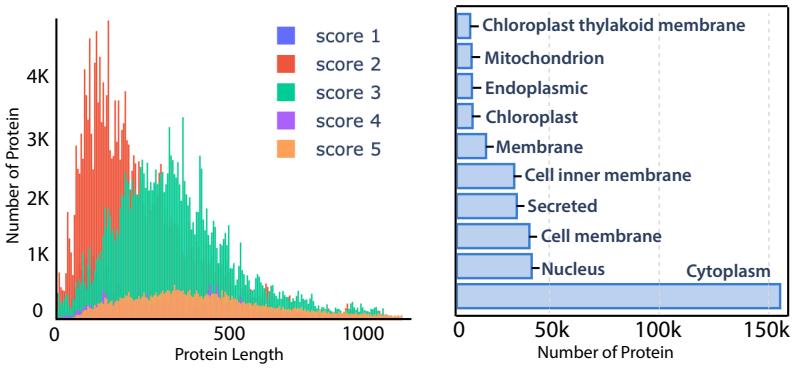
B.2 Detailed Experimental Setups

To learn protein KE embedding, following the TransE approach, we initiate embeddings for entities and relationships through a random initialization procedure. We employ the SGD optimizer with a learning rate of 1.0. The dimensions of entities and relations' embeddings are set to 200. After 1000 epochs, the loss eventually converges to 0.168. The ℓ_2 distance utilized for clustering proteins is set to 1.4. For sequence similarity, we use mmseqs2 (GPL-3.0 license) with `-cov-mode 0 -min-seq-id 0.8` parameter.

We perform incremental training on OPT-1.3b. We wrap the protein sequence with <protein> and </protein> and apply character-based tokenization, treating each amino acid as a single token. For text corpus, we tokenize them using the GPT-2 byte level BPE tokenizer. We utilize Pytorch to conduct experiments with 8 32G V100 GPUs. We use a batch size of 128 and a context length of 1,024 tokens. We adopt the Fully Sharded Data Parallel (FSDP) acceleration strategy alongside the fp16 data format. We adopt the AdamW optimizer with $\beta = (0.9, 0.98)$. We set the weight decay to 0.01 and the dropout rate to 0.1. The learning rate increases to 1e-4 for the first 5000 warming-up steps and decays linearly to 0 for the rest of the training steps. We pre-train InstructProtein for the first 40,000 steps, and fine-tune it with instruction data in the next 20,000 steps.

B.3 Detailed Data Preprocessing

An instruction data consists of a sequence of proteins, as well as textual questions and corresponding answers. We consider the data to be contaminated if the protein sequence appears in both the evaluation set and the training set. To avoid such a problem, we filtered proteins based on sequence identity. Specifically, we first collected a total of 9,373 protein sequences used in the test set for all downstream tasks. Then we clustered the proteins



(a) Protein Understudy (b) Annotation Understudy

Figure 6: The overview of the problem of understudied proteins. (a) We visualized the protein length distribution for different annotation scores. The annotation score provides a heuristic measure of the annotation content (Score 5 is associated with the best-annotated entries, and a score 1 denotes an entry with rather basic annotation.). (b) We visualized the ten most used categories in subcellular location annotations.

Example 1
KGC Task: Head Prediction
Triple: ({protein}, family, Insulin-like receptor)
Answer:
Instruction: I would like a protein in insulin-like receptor.
Output: One of the protein that meets the demand is {protein}

Example 2
KGC Task: Triple Classification
Triple: ({protein}, family, insulin-like receptor) -> Yes
Answer:
Instruction: Does {protein} belong to insulin-like receptor?
Output: Based on the record, the answer is yes.

Example 3
KGC Task: Tail Entity Prediction
Triple: ({protein}, function, insulin receptor substrate binding)
KCM: ({protein}, family, insulin-like receptor) ->
({protein}, function, insulin receptor substrate binding)
Answer:
Instruction: What function does {protein} have?
Output: Since the protein in insulin-like receptor family,
the protein enables insulin receptor substrate binding.

Example 4 Knowledge Graph Retrieval
KGC Task: Tail Entity Prediction
Triple: ({protein}, locate, nucleus)
KCM: ({protein}, family, retinoid x receptor/hnf4) ->
({protein}, locate, nucleus)

Language Model Generation
Answer:
Instruction: Where is {protein} located?
Output: Since the protein is in the retinoid x receptor/hnf4 family,
it is located in the nucleus.

Figure 7: Our data generation prompt. We provide three in-context examples with and external knowledge needed to generate the next instruction data. Purple: One of the model's generations for the given prompt.

in the training set based on sequence identity (70% as threshold), and deleted each cluster containing the sequence in the test set (a total of 19,455 protein sequences), which ensures that there is no overlap between the proteins in the training and test sets and avoid the potential for data contamination.

C Downstream Task

Similar to reading comprehension problems, we transform all data in protein sequence understanding tasks into a Question&Answer (QA) format where each item consists of a protein sequence, a question about that protein, and a list of possible answers. LLMs are required to predict which answers are true. Following Brown et al. (2020), we use a classification approach where, for example, only two outputs ("*yes*" and "*no*") are considered and the higher probability one is taken as the model's prediction.

We list the detailed definition of downstream tasks. {protein} and {label} are used as placeholders. The licenses of all datasets allow scientific research. Dataset statistics are summarized in Table 7.

Subcellular Localization Prediction is a sequence-level classification task. Each input sequence x is mapped to a label y which represents the subcellular location.

- Prompt template (InstructProtein, OPT, LLaMA, Alpha, BioMedGPT, Mol-Instructions): {protein} Instruction: What cellular components is the protein located in?
- Prompt template (Galactica): {protein} ## Subcellular Location

Table 6: Example of instructions in the Knowledge Instruction dataset. We leverage {protein} as a placeholder for the corresponing protein sequence.

Task	Instruction
Family Generation	Instruction: Can you provide me with a protein belonging to the secretoglobin family? Output: Sure, here's a protein from the secretoglobin family: {protein}.
Superfamily Prediction	Instruction: Could you please provide information about the superfamily to which {protein} belongs? Output: {protein} is likely a member of the rho GTPase activation protein superfamily.
Domain Generation	Instruction: Could you provide me with a protein that contains the nad(p)-binding rossmann-fold domain? Output: Certainly, here is a protein that contains the nad(p)-binding rossmann-fold domain: {protein}.
Family Prediction	Instruction: What family does {protein} belong to? Output: The protein belongs to the alpha-1b-glycoprotein/leukocyteimmunoglobulin-like receptor family.
Function Prediction	Instruction: protein Does the protein have the molecular function of zinc ion binding? Output: Yes, based on the information that the protein belongs to the Retinoid X receptor/HNF4 family, it indeed has the molecular function of zinc ion binding.
Process Prediction	Instruction: I'm interested in learning about the biological process associated with {protein}. Could you provide information on this? Output: Given its association with the Retinoid X receptor/HNF4 family, {protein} is likely involved in the regulation of DNA-templated transcription.
Binding Site Generation	Instruction: Provide me with a protein associated with the IQ motif EF-hand binding site. Output: Sure, the protein that interacts with the IQ motif EF-hand binding site and exhibits a protein binding molecular function is {protein}.
Family Prediction	Instruction: Does {protein} belong to nfat activation molecule 1? Output: Based on the record, the answer is yes.

- Label words (sub): {0: "plasma membrane", 1: "cytoplasm", 2: "endoplasmic reticulum", 3: "golgi", 4: "vacuole", 5: "mitochondrion", 6: "nucleus", 7: "peroxisome", 8: "chloroplast", 9: "extracellular"}
- Label words (bin): {0: ["plasma membrane", "golgi", "vacuole", "endoplasmic reticulum"], 1: ["extracellular", "peroxisome", "nucleus", "cytoplasm", "mitochondrion", "chloroplast"]}

Protein Function Annotation is a sequence-level classification task to annotation protein with functional labels. Each example consists of a protein, a label. They system must predict whether the label belongs to the protein.

- Prompt template: {protein} Instruction: Does the protein associate with label?
- Label words: {0: "No", 1: "Yes"}

Metal Ion Binding Prediction is a sequence-level classification task to predict whether a protein can bind to ion.

- Prompt template: {protein} Instruction: Does the protein associate with metal ion binding?
- Label words: {0: "No", 1: "Yes"}

Instruction-Protein Pairing Accuracy probe the insturction-following capabilities in protein generation. Protein are decoded under 10 different instructions (9 randomly sampled instructions and 1 true corresponding instruction). The system must predict which one is the most relevant instruction.

- Prompt template: Instruction: I would like a protein that is in {label}. Output: One of the protein that meets the demand is {protein}"

D Additional Experiments

D.1 Generalization ability on free-form questions

In addtion to template-based Q&A, We also experiment with creating question in a free-form manner using ChatGPT for the Subcellular Localization task (e.g., "Where within the cell can the protein be found?" or "Which parts of the cell contain the protein?"). As shown in Table 8, comparative analysis

Table 7: Dataset Statistics for downstream tasks.

Dataset	# Test	Task Type
Protein Function Annotation - *Biological Process*	104,794	Held-In
Protein Function Annotation - *Molecular Function*	22,372	Held-In
Protein Function Annotation - *Cellular Component*	38,594	Held-In
Subcellular Localization Prediction - *bin*	1,749	Held-In
Subcellular Localization Prediction - *sub*	2,773	Held-In
Metal Ion Binding Prediction	1,332	Held-Out
Instruction-Protein Pairing Accuracy - *Fold*	718	Held-Out
Instruction-Protein Pairing Accuracy - *Family*	1,272	Held-In
Instruction-Protein Pairing Accuracy - *Superfamily*	7,408	Held-In

Table 8: Performance on subcellular localization (sub) tasks on free-form questions.

Models	Subcellular Localization	
	Bin	Sub
Alpaca	57.52	18.31
Mol-Instructions	57.52	19.43
BioMedGPT	59.77	58.85
InstructProtein	**83.24**	**68.61**

Table 9: Performance on contact map prediction task.

Models	OPT	InstructProtein	TAPE
Contact Map	0.13	0.26	0.28

with instruction-tuned baselines demonstrates the model's ability to generalize across diverse questions.

D.2 Evaluation of protein representations output by InstructProtein

Because the structure of a protein determines its functions, the accuracy of a model's structure prediction directly reflects its ability to understand function. We conduct analysis using the contact map prediction task from the TAPE (Rao et al., 2019) benchmark. We compared the performance of our model, InstructProtein, with its base model OPT, as well as with the TAPE Transformer, a well-established benchmark model for structure-related tasks. We report precision at L/5 (P@L/5) in Table 9. We observe that InstructProtein significantly outperforms its base model (OPT) in understanding protein structure, demonstrating the efficacy of pre-training and instruction tuning stages in enhancing structural understanding. These findings provide valuable insights into the model's ability to reason about protein structure.

Table 10: Ablation of training stages.

Models	GO-BP	GO-MF	GO-CC
OPT (base model)	51.83	56.10	51.94
Pre-training only	50.41	55.79	54.33
Instruction tuning only	65.13	76.98	72.53
InstructProtein	71.49	85.83	79.69

D.3 Ablation study on pre-training and instruction tuning stages

To substantiate the advantages of pre-training and instruction tuning stages, we are providing additional experimental results in Table 10. The pre-training is conducted with the UniRef100 and Pubmed datasets that respectively contain proteins and biomedical literature, while the instruction tuning is conducted using the proposed Knowledge Instruction approach with aligned natural language and protein language corpora. We can observe that the improvement in performance is mainly due to the fine-tuning phase. However, the lack of pre-training for modeling protein language lead to suboptimal performance.

E More Examples

We provide more illustrative examples of protein understanding and protein de novo design to show the superiority of our InstructProtein over existing LLMs.

	Galactica	BioMedGPT	InstructProtein	Ground Truth
 **What superfamily does the protein belong to?**	Belongs to the RlmN family. ### KeywordsCytoplasm, GTP-binding, Initiation factor, Nucleotide-binding, Protein biosynthesis, RNA-binding	Belongs to the MqsS/MqsR family.	The protein is in signal transduction histidine kinase, dimerisation/phosphoacceptor domain superfamily.	1. histidine kinase/hsp90-like atpase superfamily 2. signal transduction histidine kinase, dimerisation/phosphoacceptor domain superfamily.
 **What domain does the protein have?**	The C-terminal region (residues 470-550) is predicted to be a transmembrane domain, suggesting that it may be located in the outer membrane.	In the C-terminal section; belongs to the transferase hexapeptide repeat family.	The protein has glmu, c-terminal lbh domain and moba-like ntp transferase domain.	1. glmu, c-terminal lbh domain 2. moba-like ntp transferase domain
 **What is the function of this protein?**	Part of the ABC transporter complex MetNIQ involved in methionine import. Responsible for energy coupling to the transport system.	Catalyzes the last two sequential reactions in the de novo biosynthetic pathway for UDP-N-acetylglucosamine.	The protein enables magnesium ion binding, glucosamine-1-phosphate n-acetyltransferase activity, and udp-n-acetylglucosamine diphosphorylase activity.	1. magnesium ion binding 2. glucosamine-1-phosphate n-acetyltransferase activity 3. udp-n-acetylglucosamine diphosphorylase activity
 **Does the protein have proteinase, inhibitor i2, kunitz, conserved site?**	Answer: No	Has a Kunitz-type B chain.	Based on the record, the answer is yes.	The protein has proteinase inhibitor i2, kunitz, conserved site.

Figure 8: More examples of protein understanding.

Instruction: I would like a protein that has cystatin domain. Output:

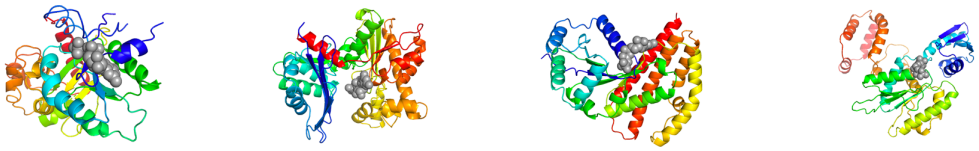
GoundTruth:
- UniProt ID: **B2Z449**

Design 1:
- Length: **165 residues**

Design 2:
- Length: **115 residues**

Design 3:
- Length: **107 residues**

Instruction: I would like a protein that enables GTP binding. Output:



GoundTruth:
- Affinity: **-6.0 (kcal/mol)**
- PDB id: **5C1S-A**

Design 1:
- Affinity: **-7.9 (kcal/mol)**
- pLDDT: **96.4**

Design 2:
- Affinity: **-8.2 (kcal/mol)**
- pLDDT: **44.4**

Design 3:
- Affinity: **-8.9 (kcal/mol)**
- pLDDT: **40.3**

Figure 9: More examples of function-instruction-based protein de novo design.

Instruction: I would like a protein that is in metallothionein family. Output:

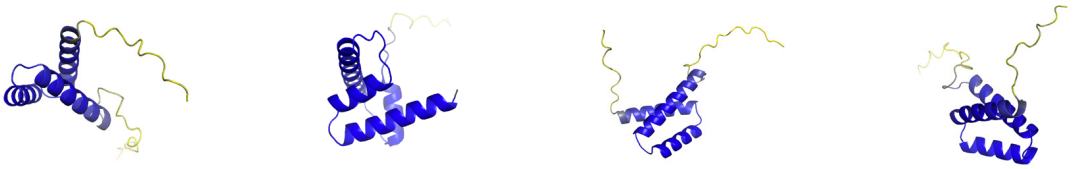
GoundTruth:
- UniProt ID: **A0A024R6T4**

Design 1:
- Length: **61 residues**

Design 2:
- Length: **61 residues**

Design 3:
- Length: **54 residues**

Instruction: I would like a protein that is in retroviral VpR/VpX protein family. Output:



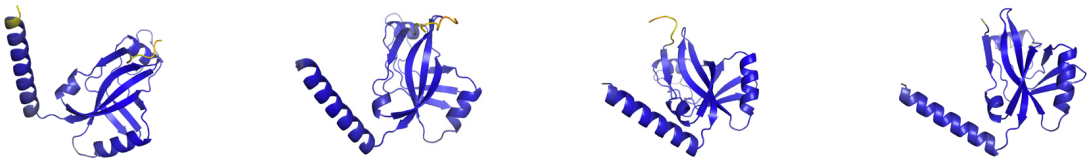
GoundTruth:
- UniProt ID: **A0A023HIS7**

Design 1:
- Length: **126 residues**

Design 2:
- Length: **96 residues**

Design 3:
- Length: **96 residues**

Instruction: I would like a protein that is in SsrA-binding protein family. Output:



GoundTruth:
- UniProt ID: **A0ALD2**

Design 1:
- Length: **153 residues**

Design 2:
- Length: **150 residues**

Design 3:
- Length: **151 residues**

Instruction: Instruction: I would like a protein that is in kappa casein family. Output:

GoundTruth:
- UniProt ID: **P02668**

Design 1:
- Length: **98 residues**

Design 2:
- Length: **145 residues**

Design 3:
- Length: **192 residues**

Figure 10: More examples of family-instruction-based protein de novo design (colored by pLDDT).

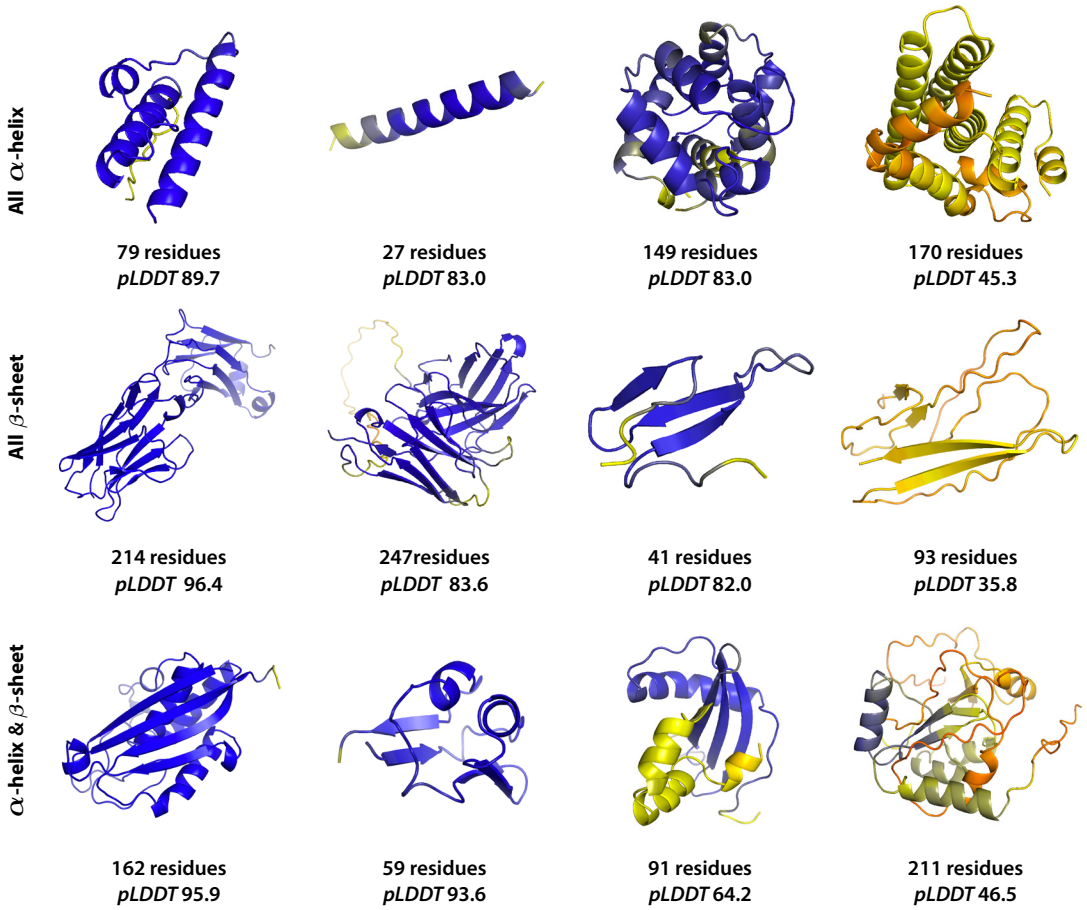


Figure 11: More examples of structure-instruction-based protein de novo design (colored by pLDDT).

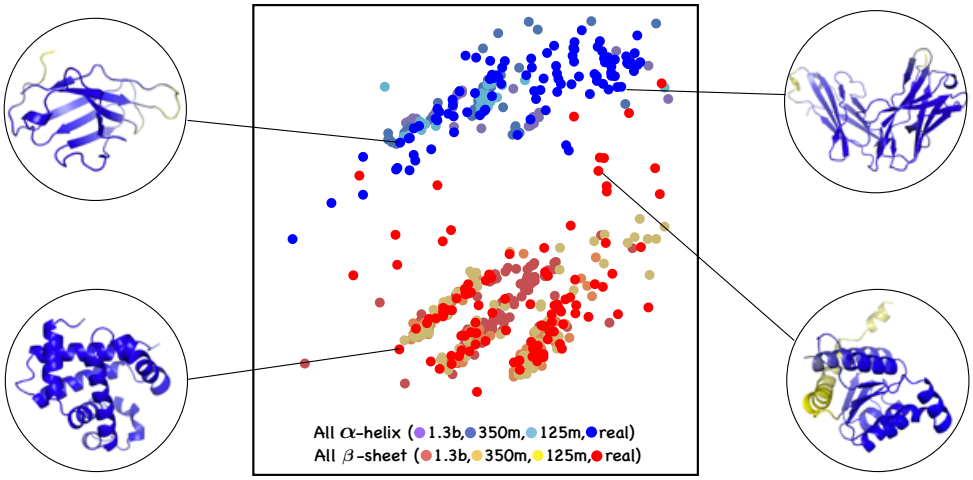


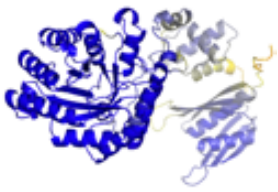
Figure 12: Comparing the embedding of proteins designed based on structural instructions and native proteins.

Prompt:
I would like a protein that enables metal ion binding.

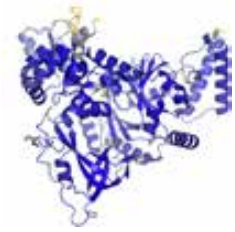
Design 1:
MNKKLVVFDTTLRDGEQSPGASMTKEEKVRIAKQLERMKVDVIEAGFAAASEGDFEAVKAIAQAIKHSTVCSLARANDK
DIARAAEALKPANSFRIHTFIATSALHMEKKLRMTPDQVYEQAVKAVRFARKFTDNVEFSPEDAGRSDPEFLCQVVEAV
INEGASTINIPDTVGYSIPSQFGQLIATIRGRVPNSDKAIWSVHCHNDLGMAVANSLAGVKIGGARQIECTINGLGERA
GNCSLEEIVMAVKTRKDYFGLDVGIDAQHILPASKLVSQITGFVVQPNKAVVGANAFAHASGIHQDGVLKARDTYEIMR
AEDVGWSANKIVLGKLSGRNAFKQRLQELGVQLESEAELNAAFMRFKDLADRKAEIFDEDIIAIVSEECALAQEQEHFK
FVSLSQRSETGEQPQAKVVFAVEGKEVTGEARGNGPVDATFNAIEGEVGSGSELLLYSVNAISTGTQAQGEVTVRLSKS
GRIVNGVGTDPDIVAASAKAYISALNKLHSKDDKLNPQRA

Design 2:
MSDTLFRQLDILRAELDQHNYRYYVDDAPSVPDAEYDRLMQELVALEAEHPELLTPDSPTQRVGGEALAAFRQVRHRIP
MLSLGNAFEDGDLEAFKRRVDRLTGELPGGPLELVCEPKLDGLAVSLLYRDGQLVRGATRGDGTTGEEVSTNVRTIRNI
PLRLHGKGWPAVLEVRGEVYMSRAGFDRLNAAQAEAGGKTFANPRNAAAGSLRQLDSKITASRPLEFCCYGIGQVSECF
DGHQQGLLARLRAWGLPINQELRHAEGLEACLAFYRKVGERRASLPYEIDGAVFKLDERARQRLGFRARAPRWAIAHKF
PAMEELTELLDVEFQVGRTGAITPVARLKPVQVAGVTVSNATLHNMDEIARLGLRPGDRVIVRRAGDVIPQVVQVVLER
RPEGARPIPLPAHCPVCGSEAVREPGEVAVRCVGGLYCEAQRKEALRHFASRRALDVEGLGDKLVDQLVDRELVGSPAD
LYRLTYEQLVGLEGFAEVSSKKLAEAIASSRQPSLARFIYALGIREVGEATAKDLARHFGTLDALMAADEEALLAVPDV
GPVVAHNIRHFFAQPHNLEVVDALLTAGVHWPQVEAVADEAPLAGQTWVLTGTLDSMSRDDAKAQLEALGARVSGSVSK
KTAAVVAGEAAGSKLAKAQELGVKVLDEEELLRLLAAP

Design 3:
MERKSKNPELSSKAFAIAYCAVCGLSRSRNQVFIHKKGLNICRQCFYKKFVSEIISEEKIRQKQRTATKTLPLKDCSDD
IKLKIDPKLKQDKTFEVIEDYEEEIKIEDTIWETNAINEISTCSVCGREEGGFFEAVVQVRAADLLYPTGAQENDELNR
QVREKLQSLRQKVATAKKEFNHIDEYAKHIQKILEEKGVSHDLVRLSTYDLAAAQSLGAVPSEGQKKPGTDEKLRIPRS
GLGLKASLISTYADKVRAARAKLKKAGDIDAQFINGIIKILVPMRPSAVHLDLLENLGCKKVLPENPQIDVLSTRKYPT
LLHGGEPLKEAPLIAPSLIQLLKPYMEGQGLVVVDSFLLSSSRVLRLLKEGALDYLIIEGKYRILSIEKGKFEIDFEGK
PDCVGGSEDLAKLAEVVGAEYLGEDALEVVKLGNQDVTKVKNVQSLEIEEIIRNESIKVKTILQEKYEDIESIIKTKER
LDKLKKEKRKKEIKKNKLTDNANKLEPIETEQELQKLESRNIPLEELLELTNELRPLILANVAERVRKELEIQKIPTEK
NE



Design 1:
Similar Protein: A0A1Z5HSR0
Metal Ion Binding: Enable



Design 2:
Similar Protein: A0A7C4E6S4
Metal Ion Binding: Enable

Design 3:
Similar Protein: None

Figure 13: More examples of designed proteins that enable metal ion binding.

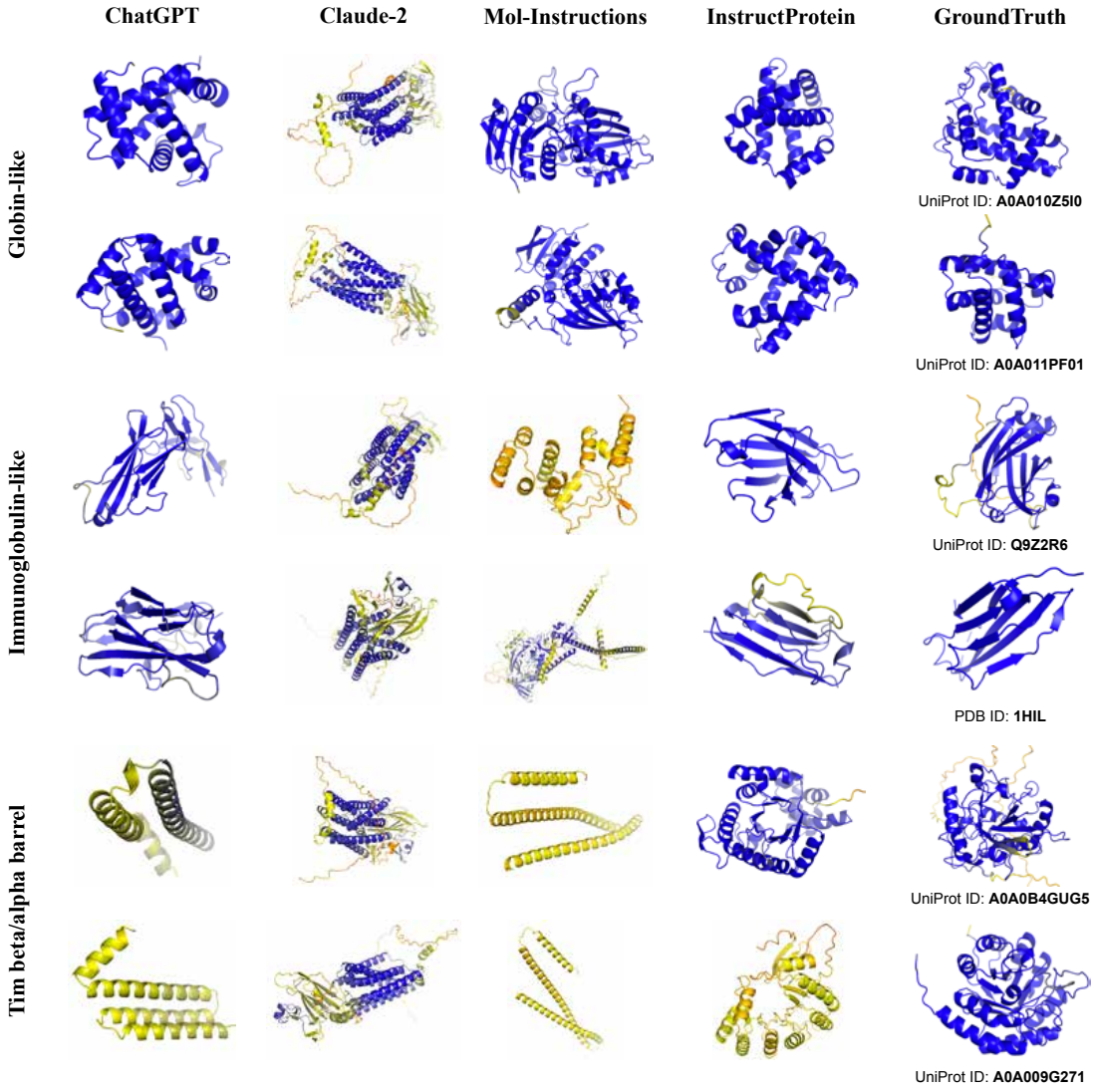


Figure 14: Visualize proteins generated by ChatGPT, Claude-2, Mol-Instructions, and InstructProtein.

Protein Design

Instruction: I would like a protein that has bifunctional inhibitor/plant lipid transfer protein/seed storage helical domain.

Output (*ChatGPT*): The combination of a bifunctional inhibitor domain, a plant lipid transfer protein (LTP) domain, and a seed storage helical domain in a single protein is a very specific and specialized request. As of my last knowledge update in January 2022, I do not have access to a database of specific proteins with this exact combination of domains.

Output (*GPT4*): To design a protein with a bifunctional inhibitor/plant lipid transfer protein/seed storage helical domain, you would typically need to combine elements from each of these distinct protein domains into a single molecule. Here's a general approach to achieving this: 1. Understand the Individual Domains; 2. Domain Fusion; 3. Consider Structural Stability and Function; 4.Computational Modeling and Simulation; 5. Synthesis and Testing

---

Protein Understanding (Family)

Instruction: What family does the protein belong to?
MAFSDLTSRTVHLYDNWIKDADPRVEDWLLMSSPLPQTILLGFYVYFVTSLGPKLMENRKPFELKKAMITYNFFIVLFSVYM
CYEFVMSGWGIGYSFRCDIVDYSRSPTALRMARTCWLYYFSKFIELLDTIFFVLRKKNSQVTFLHVFHHTIMPWTWWFGVKF
AAGGLGTFHALLNTAVHVVMYSYYGLSALGPAYQKYLWWKKYLTSLQLVQFVIVAIHISQFFFMEDCKYQFPVFACIIMSYSF
MFLLLFLHFWYRAYTKGQRLPKTVKNGTCKNKDN

Output (*ChatGPT*): To determine the classification of this protein, you would need to perform a bioinformatics analysis, which may involve using tools like BLAST or Pfam to search for homologous sequences or conserved domains. Additionally, you may need to consult with experts in the field of protein bioinformatics and structural biology for further characterization and classification.

Output (*GPT4*): However, without access to bioinformatics tools or databases in this environment, I cannot directly determine the family of the protein based on its sequence. If you have access to these tools, you can input the sequence into one of the aforementioned databases to get a detailed analysis. Alternatively, if you provide more context or information about the protein, such as its source, function, or any known characteristics, I might be able to offer more insights.

Protein Understanding (Function)

Instruction: Does the protein enable carboxyl- or carbamoyltransferase activity?
KVQLKGRDLLTLKNFTGEEIKYMLWLSADLKFRIKQKGEYLPLLQGKSLGMIFEKRSTRTRLSTETGFALLGGHPCFLTTQDIHL
GVNESLTDTARVLSSMADAVLARVYKQSDLDTLAKEASIPIINGLSDLYHPIQILADYLTLQEHYSSLKGLTLSWIGDGNNILHSIM
MSAAKFGMHLQAATPKGYEPDASVTKLAEQYAKENGTKLLLTNDPLEAAHGGNVLITDTWISMGREEEKKKRLQAFQGYQVT
MKTAKVAASDWTFLHCLPRKPEEVDDEVFYSPRSLVFPEAENRKWTIMAVMVSLLTDYSPQLQKPKF

Output (*ChatGPT*): The protein sequence you provided does not contain information about its specific enzymatic activities, such as carboxyl- or carbamoyltransferase activity. Determining the enzymatic activities of a protein typically requires experimental studies, structural analysis, and functional assays.
To understand the enzymatic activities of a protein, researchers would typically perform biochemical experiments to investigate its function and substrate specificity. Additionally, structural analysis, such as X-ray crystallography or NMR spectroscopy, can provide insights into the protein's active sites and potential catalytic mechanisms.
Based solely on the protein sequence, it is not possible to determine its enzymatic activities or functions with certainty. If you are working with this protein and need to know more about its function, I recommend conducting experimental studies or consulting with experts in the field of protein biochemistry and enzymology.

Output (*GPT4*): To determine if a protein has carboxyl- or carbamoyltransferase activity based solely on its amino acid sequence is challenging without additional context or experimental data. Carboxyltransferases and carbamoyltransferases are enzymes that catalyze the transfer of carboxyl or carbamoyl groups, respectively, to their substrates. These enzymes are typically characterized by specific active sites and motifs in their amino acid sequences that are necessary for their catalytic activity.
Without access to bioinformatics tools in this environment, I cannot directly analyze this sequence for such activities. To proceed, you would typically use the aforementioned tools, or if you have additional information about the protein (such as its source, structure, or known functions), it could help in making a more informed guess regarding its potential enzymatic activities.

Figure 15: Examples of closed-source LLMs that often refuse to answer protein-related questions.

MHHHHHHSSGVDLGTENLYFQSNAMDFPQQLEACVKQANQALSRFIAPLPFQNTPVVETMQYGALLGGKR
LRPFLVYATGHMFGVSTNTLDAPAAAVECIHAYSLIHDDLPAMDDDDLRRGLPTCHVKFGEANAILAGDALQ
TLAFSILSDANMPEVSDRDRISMISELASASGIAGMCGGQALDLDAEGKHVPLDALERIHRHKTGALIRAAVR
LGALSAGDKGRRALPVLDKYAESIGLAFQVQDDILDVVGDTATLGKRQGADQQLGKSTYPALLGLEQARKK
ARDLIDDARQALKQLAEQSLDTSALEALADYIIQRNK

User: Does the protein enable transferring alkyl or aryl groups, other than methyl groups? [Yes/No]
ChatGPT: No. The provided amino acid sequence does not contain known motifs or domains associated with enzymes that transfer alkyl or aryl groups. The sequence lacks characteristics of methyltransferases, which are enzymes involved in the transfer of methyl groups. Additionally, there are no specific functional annotations or signatures indicating a role in alkyl or aryl group transfer. Therefore, based on the given sequence, it is not likely that the protein enables the transfer of alkyl or aryl groups other than methyl groups.
GroundTruth: Yes

User: Does the protein enable (2E,6E)-farnesyl diphosphate synthas? [Yes/No]
ChatGPT: No. The absence of such motifs or homology in the provided sequence suggests that this protein is not likely to be (2E,6E)-farnesyl diphosphate synthase. It's important to note that a definitive answer would require experimental validation or more detailed bioinformatic analysis.
GroundTruth: Yes

User: Does the protein enable poly(A)-specific ribonuclease? [Yes/No]
ChatGPT: No, based on the provided amino acid sequence, there is no indication or motif suggesting that the protein enables poly(A)-specific ribonuclease. The sequence doesn't contain known motifs associated with poly(A)-specific ribonuclease activity.
GroundTruth: No

MAHHHHHHMALVSMRQLLDHAAENSYGLPAFNVNNLEQMRAIMEAADQVNAPVIVQASAGARKYAGAPF
LRHLILAAVEEFPHIPVVMHQDHGASPDVCQRSIQLGFSSVMMDGSLLEDGKTPSSYEYNVNATRTVVNFSH
ACGVSVEGEIGVLGNLETGEAGEEDGVGAAGKLSHDQMLTSVEDAVRFVKDTGVDALAIAVGTSHGAYKFT
RPPTGDVLRIDRIKEIHQALPNTHIVMHGSSSVPQEWLKVINEYGGNIGETYGVPVEEIVEGIKHGVRKVNIDT
DLRLASTGAVRRYLAENPSDFDPRKYLGKTIEAMKQICLDRYLAFGCEGQAGKIKPVSLEKMASRYAKGELN
QIVK

User: Does the protein enable fructose-bisphosphate aldolase? [Yes/No]
ChatGPT: No. The provided sequence appears to be a longer peptide with no recognizable motifs or domains associated with aldolase activity. Additionally, the sequence does not align with known fructose-bisphosphate aldolase sequences.
GroundTruth: Yes

User: Does the protein enable aldehyde-lyases? [Yes/No]
ChatGPT: No. The provided amino acid sequence does not exhibit clear motifs or conserved residues associated with aldehyde-lyases. Aldehyde-lyases are enzymes that catalyze the cleavage of carbon-carbon bonds in aldehydes. Typically, these enzymes have specific motifs and active site residues that are essential for their catalytic activity. The sequence you provided does not contain these characteristic features associated with aldehyde-lyases.
GroundTruth: Yes

User: Does the protein enable non-chaperonin molecular chaperone ATPase? [Yes/No]
ChatGPT: No. The provided amino acid sequence does not exhibit characteristics or motifs commonly associated with non-chaperonin molecular chaperone ATPase activity. Molecular chaperones, including chaperonin ATPases, typically possess specific domains and motifs involved in ATP binding and hydrolysis, as well as substrate binding. The absence of such features in the given sequence suggests that it is not likely to function as a non-chaperonin molecular chaperone ATPase.
GroundTruth: No

Figure 16: ChatGPT prediction results of whether a protein can catalyze a reaction.