S-PLM: Structure-aware Protein Language Model via Contrastive Learning between Sequence and Structure

Duolin Wang[1], Mahdi Pourmirzaei[1], Usman L Abbas[2], Shuai Zeng[1], Negin Manshour[1], Farzaneh Esmaili[1], Biplab Poudel[1], Yuexu Jiang[1], Qing Shao[2]*, Jin Chen[3]* and Dong Xu[1]*

[1] Department of Electrical Engineering and Computer Science and Christopher S. Bond Life Sciences Center, University of Missouri, Columbia, MO 65211, USA

[2]Chemical & Materials Engineering, University of Kentucky, Lexington, KY 40506, USA

[3]Department of Medicine and Department of Biomedical Informatics and Data Science, University of Alabama at Birmingham, Birmingham, AL 35294, USA

*To whom correspondence should be addressed. Emails: qshao@uky.edu; jchen5@uab.edu; xudong@missouri.edu.

Abstract: Large protein language models (PLMs) present excellent potential to reshape protein research by encoding the amino acid sequences into mathematical and biological meaningful embeddings. However, the lack of crucial 3D structure information in most PLMs restricts the prediction capacity of PLMs in various applications, especially those heavily depending on 3D structures. To address this issue, we introduce S-PLM, a 3D structure-aware PLM utilizing multi-view contrastive learning to align the sequence and 3D structure of a protein in a coordinate space. S-PLM applies Swin-Transformer on AlphaFold-predicted protein structures to embed the structural information and fuses it into sequence-based embedding from ESM2. Additionally, we provide a library of lightweight tuning tools to adapt S-PLM for diverse protein property prediction tasks. Our results demonstrate S-PLM's superior performance over sequence-only PLMs, achieving competitiveness in protein function prediction compared to state-of-the-art methods employing both sequence and structure inputs.

Introduction

Proteins play a crucial role in resolving the health, energy, and environmental challenges that society is facing. An essential task is to gain information about proteins of interest in a fast and accurate manner. Along this line, computational predictions of protein properties from protein sequences play important roles. Protein language models (PLMs) can explore the underlying patterns within protein sequences and predict properties based on the underlying sequence patterns[1]. The current paradigm for PLM development and deployment includes two stages: (1) train a PLM to convert the amino acid sequence into a mathematical representation (embedding) by means of the masked language modeling (MLM), where the model predicts masked amino acids in the sequence based on the surrounding context[2–4]; and (2) fine-tune the pretrained PLMs with the protein property[2] data to perform specific protein prediction tasks. Several PLMs have been developed following this paradigm, such as the ProtBert[3] and ESM2[5] models. These PLMs have shown encouraging results for protein property predictions if well fine-tuned on specific data and demonstrated their potential for new knowledge discovery and analyses[6–8].

One challenge in developing PLMs is enriching critical biophysical information into the embeddings. It is well known that a protein's function relies on its 3D structure. However, most

PLMs are trained solely on amino acid sequences, thereby constraining their predictive capabilities, especially those heavily depending on 3D structures of proteins. Some methods attempted to enrich the evolutionary or function information in the sequence-based embedding. One method includes multiple sequence alignments (MSAs) into PLMs, such as AlphaFold's Evoformer[9] and MSA Transformer[10]. ProteinBERT incorporated gene ontology (GO) annotations into the MLM pre-training scheme[11]. It utilized a denoising autoencoder to pre-train with corrupted protein sequences and GO annotations and performed comparatively well in many protein tasks despite its relatively smaller size. These methods indeed enriched the information in the sequence-based embedding and boost the performance of the PLMs. However, none of these methods incorporated the critical structural information in an indirect manner.

Another emerging method is to develop a joint embedding with both sequence and structural inputs. Chen *et al.* proposed a self-supervised learning-based method for structure representation to leverage the available pre-trained PLMs[12]. Zhang et al. systematically explored the joint embedding for proteins based on ESM2 and three distinct structure encoders[13]. Hu et al. also developed a joint embedding for proteins based on coupling information from protein sequence and structure[14]. These joint embeddings show better performance in quite a few protein property prediction tasks, highlighting the importance of including structure information in the protein representation. However, the utilization of these joint-embedding models requires both sequence and structure as input. Even with prediction tools such as AlphaFold, it remains challenging to gain reliable structures for some specific proteins. Furthermore, computationally these methods take an extra step to obtain predicted structures, which requires extra effort and time.

We propose an alternative approach. Instead of developing a joint embedding that needs both sequence and structural inputs, we develop a sequence-based embedding that incorporates structural information. Multi-view contrastive learning has emerged as a promising technique for this purpose[15,16]. By focusing on learning a coordinated embedding that integrates and leverages information from different views, this approach enables the model to capture rich and complementary features from diverse perspectives. During training, the contrastive loss function pushes the representations of similar views (sequence and structure of the same protein) to be close to each other in the embedding space while simultaneously separating representations of dissimilar views (between different proteins) further apart. As a result, multi-view contrastive learning effectively captures underlying semantic structures and patterns in the data. A head-to-head comparison has demonstrated that multi-view contrastive learning can provide a more effective strategy than the cross-view representation learning[16].

To this end, we propose S-PLM, a 3D structure-aware PLM that enables the sequence-based embedding to carry the structural information through multi-view contrastive learning. One advantage of S-PLM compared to those joint-embedding models is that S-PLM only needs amino acid sequences as the input during inference after being trained. This sequence-only input avoids the overhead of using predicted protein structures. It is worth mentioning that two other recent successful cases have illustrated the potential of contrastive learning in enriching information in sequence-based embedding. One is the ProtST, which injected biomedical text into a PLM by aligning these two modalities through a contrastive loss[17]. After training, their PLM contained enhanced protein property information and demonstrated superiority over previous models on diverse protein representations and classification benchmarks. Another one is the CLEAN[18] model, which utilized the information from the EC number to enhance the ability of PLM to predict enzymatic functions based on a protein sequence. Although S-PLM can accommodate either

structure or sequence as the input, this article will only focus on demonstrating the advantages of S-PLM in encoding protein sequences and leveraging structural information to improve sequence representations for various protein prediction tasks. The sequence encoder of S-PLM was implemented based on a pretrained ESM2 model, offering extensibility to incorporate new protein properties incrementally without forgetting previous knowledge[19]. Through several unsupervised protein clustering tasks and several structure-dominated supervised protein prediction tasks, S-PLM outperformed the baseline ESM2 model significantly.

To apply pretrained PLMs to specific protein prediction tasks, a typical transfer learning approach involves fine-tuning the PLMs with domain data to update all model parameters. However, full fine-tuning may not work well for large PLMs due to limited task-specific training data or severe computational and memory costs. To address this challenge, a set of lightweight tuning strategies has emerged, making specific and parameter-efficient modifications to large pre-existing models. These strategies, such as fine-tune top layers, Adapter Tuning[19], and Low-rank adaptation (LoRA)[20], selectively update targeted parameters while keeping others frozen, significantly reducing computational and memory requirements and the needs of task-specific training data while achieving comparable or superior performance. However, there is limited existing research applying these methods to PLMs[21,22]. This work explored several lightweight tuning strategies for diverse protein prediction tasks utilizing S-PLM. Through further lightweight tuning, S-PLM achieved competitive performance in Enzyme Commission number[23] and Gene Ontology[24] prediction tasks, comparable to the state-of-art methods employing both sequence and structure inputs. A library of lightweight tuning methods has been made available at: https://github.com/duolinwang/S-PLM/.

Results

Structure-aware protein language model (S-PLM)

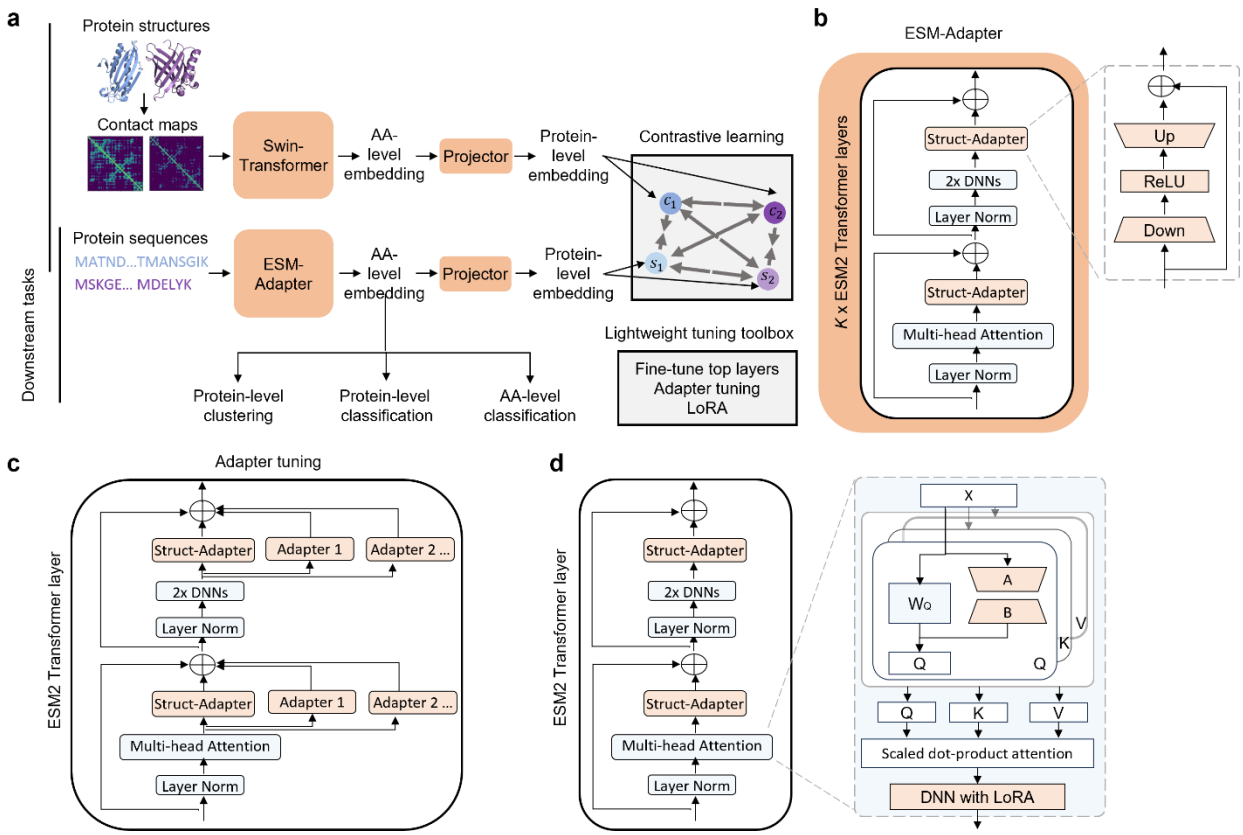

Figure 1. The framework of S-PLM and lightweight tuning strategies for downstream supervised learning. ***a***, The framework of S-PLM. During pretraining, the model inputs the amino acid sequences and contact maps simultaneously. The protein sequences are transformed into amino-acid level (AA-level) embeddings through ESM-Adapter, while the contact maps are transformed into protein-level embeddings through Swin-Transformer. Through respective projector layers, the S-PLM model is trained through contrastive learning on the protein-level embeddings from each modality. After pre-training, the ESM-Adapter that generates the AA-level embeddings before the projector layer is used for downstream tasks, such as protein-level clustering, classification, and AA-level classification tasks. The entire ESM-Adapter model can be fully frozen or learnable through lightweight tuning. ***b***, Architecture of the ESM-Adapter. The sequence encoder of S-PLM, ESM-Adapter, is tuned on the top *K* Transformer layers of ESM2, each of which adds a compact adapter module (Struct-Adapter). During the pre-training, the ESM2 backbone model is frozen and only the Struct-Adapter modules are trainable. ***c***, Adapter tunning for supervised downstream tasks is implemented by integrating a list of additional paralleled adapters on the Struct-Adapter modules. ***d***, LoRA tuning for supervised downstream tasks is implemented by adding trainable rank decomposition matrices into the multi-head attention layer of top *K* Transformer layers of ESM2.

Figure 1a depicts the framework of S-PLM. The pretraining architecture of S-PLM consists of two encoders, one to encode protein sequence and the other to encode protein 3D structures. In this study, the one-letter amino acid sequences are utilized as the input of protein sequences. The backbone C_α contact maps are used to represent the protein 3D structures because inter-residue distances contain comprehensive and essential information about protein structure. During pretraining, S-PLM inputs both amino acid sequences and backbone C_α contact maps. The

protein sequence information is converted into amino-acid level embedding through a sequence encoder (ESM2-Adapter), while the contact map information is transformed into a protein-level embedding through a structure encoder (Swin-Transformer[25]). Then through each corresponding projector layer, sequences and contact maps are converted into separate protein-level embeddings (Method 1 and 2). Finally, the S-PLM model is trained using contrastive learning as the contrastive loss is applied to a batch of sequences and contact maps. The objective of the S-PLM model is to maximize the alignment of the embeddings for the sequence and structure from the same protein and clearly separate the embeddings for the sequences and structures between different proteins (de-alignment). Inspired by the SimCLR method[26], our work adapts the CLIP[27] approach for contrastive language-image pretraining. Besides the CLIP's alignment and de-alignment across different modalities, our model also accounts for de-alignment within the same modality. For instance, as shown in Fig. 1a, our model also emphasizes dissimilarity between embeddings of sequences ($s_1 \leftrightarrow s_2$) and embeddings of contact maps ($c_1 \leftrightarrow c_2$) from different proteins (Method 3).

The sequence encoder of S-PLM was implemented as a structure adapter on the ESM2 model using adapter tuning[19]. Our adapter tuning involves integrating adapter modules into the top *K* Transformer layer of the ESM2 model. As shown in Fig. 1b, the adapter modules (Struct-Adapter) consist of a bottleneck structure and a skip-connection, positioned twice in one Transformer layer: after the multi-head attention projection and after the two feed-forward layers (Method 5). Using adapter tuning to implement the sequence encoder of S-PLM offers several advantages. Firstly, the integrated adapter module is compact. It contains much fewer parameters than the original Transformer modules of ESM2, which alleviates the training burden. Secondly, it is extensible for incremental training to add new protein features for the future extension of our model, such as protein function, without catastrophic forgetting of the previous ones. As for the current S-PLM pretraining, it retains the sequence representation capabilities of the ESM2 model because the entire ESM2 backbone model remains unchanged.

During the inference stage, S-PLM has the flexibility to accept either sequence or contact map as input and produces corresponding embeddings at various levels tailored to specific downstream tasks (e.g., protein-level clustering, protein-level classification and AA-level classification). This versatility allows S-PLM to adapt and provide suitable representations based on the specific input data and requirements of the given task. In this paper and the subsequent results, the S-PLM model primarily focuses on generating sequence embeddings from protein sequences. Therefore, the pre-trained sequence encoder of S-PLM that generates the AA-level embeddings before deploying projectors was used for downstream tasks. The entire sequence encoder can be fully frozen or learnable. To fully exploit the potential of S-PLM in supervised protein prediction tasks, we have developed several lightweight tuning strategies based on the sequence encoder of S-PLM, all incorporated into the lightweight tuning toolbox, including the fine-tune top layers, Adapter tuning (Fig. 1c), and LoRA tuning (Fig. 1d) (Method 5).

Contrastive learning rearranges the alignment between the sequence and structure embeddings

We initially assessed how contrastive learning influences the alignment between sequence and structure embeddings. We randomly selected 25 proteins from an independent test set and projected their corresponding sequence and structure embeddings onto a 2D t-SNE plot processed with and without contrastive learning. As shown in Fig. 2 left, without contrastive learning, the sequence-based embeddings (obtained from pre-trained ESM2-t33_650M_UR50D)

and the structure-based embeddings (obtained from pre-trained Swin-Transformer) were distributed in widely separated spaces. Embeddings from the same modality were closer than those from different modalities, indicating no information exchange between the two modalities. In contrast (Fig. 2 right), contrastive learning brought sequence and structure embedding of the same protein closer, suggesting successful rearrangement of sequence-based embeddings to incorporate protein structure information (and vice versa). We also grouped the proteins in the test dataset into different bins according to the sequence length. Supplementary Fig. S1 shows the same t-SNE plots, revealing the alignment of structure and sequence embeddings within specific sequence length ranges. This experiment further substantiates the efficacy of contrastive learning in aligning sequence and structure embeddings, affirming that the observed alignment is not influenced by sequence length.

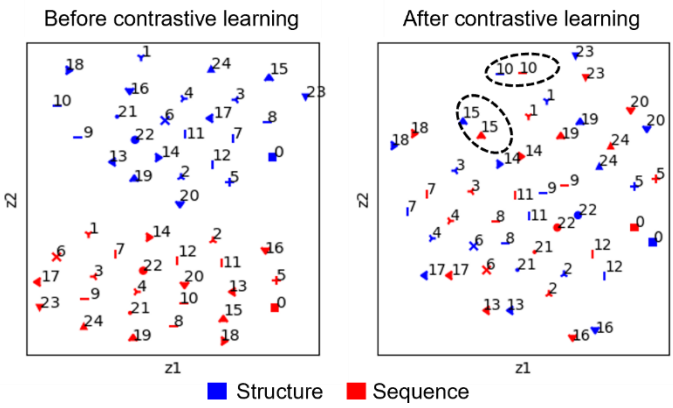

Figure 2. Comparison of structure and sequence embeddings before and after contrastive learning using t-SNE. A node in blue indicates a structure-based 2D-t-SNE embedding for a protein. A node in red indicates a sequence-based 2D-t-SNE embedding for a protein. The number beside each node indicates the index of the protein. Different indices and shapes indicate different proteins. The structure-based and sequence-based embeddings from the same protein have the same shape and protein index.

S-PLM injects the structural information into the sequence latent space

To investigate whether S-PLM can inject the structural information into the sequence latent space, we evaluated the sequence representations for the CATH protein domains[28]. This experiment primarily emphasizes structural information; hence, we deliberately chose a single representative sequence from each CATH superfamily. This approach ensures clearer visualization and reduces computational burden. We utilized the CATHS40 dataset, whose proteins have maximal 40% sequence identity. Our analysis focused exclusively on the Class, Architecture, and Topology levels of the CATH hierarchy, deliberately excluding the homologous superfamily at the last level, which is more sequence-similarity-driven clusters. We visualized and benchmarked the sequence representations produced by S-PLM against models that depend on sequence alone, including ESM2 (ESM2_t33_650M_UR50D), the pretrained PLM on which S-PLM is based, and another structure-aware model PromptProtein[29] (Fig. 3a). In Fig. 3a, each row shows the t-SNE visualization of protein embeddings from top five most represented categories of one hierarchy. It shows that sequence representation from S-PLM separates CATH structural classes more clearly than the embeddings from other two models.

We further utilized the Calinski-Harabasz index (CHI)[30] to quantitatively assess the capability of embeddings derived from different methods in distinguishing CATH structural categories. The CHI score quantifies the ratio between the sum of between-cluster dispersion and the sum of within-cluster dispersion. We computed the CHI using the ground-truth CATH categories to define clusters, using sequence embeddings to calculate both between- and within-cluster dispersion. As shown in Fig. 3b, S-PLM achieved much higher CHI across all CATH hierarchy levels than ESM2 and PromptProtein. Given that these CATH catigories were established using protein structures, this analysis strongly suggests that the sequence embedding produced by the developed S-PLM exhibits an inherent awareness of protein structures, surpassing the other two sequence-only PLMs in effectively distinguishing proteins with diverse structural characteristics. The poor performance of the sequence-only ESM2 also indicated their limitations in explicitly acquiring protein structure knowledge.

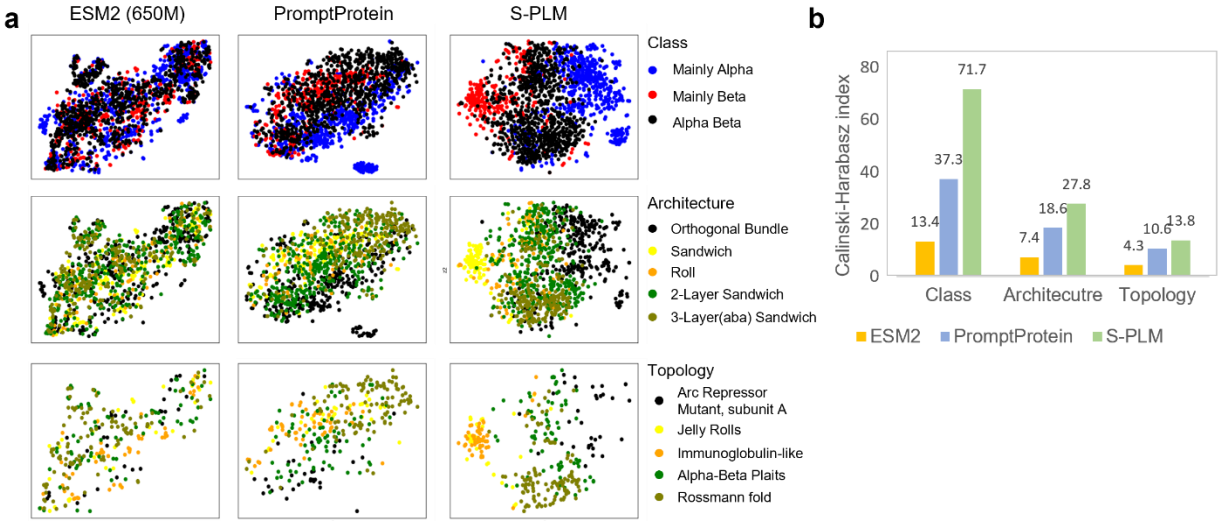

Figure 3. Visualization and benchmark of protein embeddings for three CATH structural hierarchies (Class, Architecture and Topology). **a**, t-SNE visualization of protein embeddings from top five most represented categories from one hierarchy. Embeddings produced by ESM2 and PromptProtein are shown side by side with S-PLM. **b**, Utilizing the Calinski-Harabasz index to quantitatively assess the capability of embeddings derived from different methods in clustering CATH structural categories.

Clustering enzymes via S-PLM

Recently, Huang et al. introduced a structure-based protein clustering approach for discovering deaminase functions and identifying novel deaminase families[31]. Their work applied AlphaFold2 to predict protein structures and subsequently clustered the entire deaminase protein family based on the predicted structure similarities through structure alignment. They discovered new functions of the deaminase proteins and found new deaminases; such findings cannot be obtained by mining amino acid sequences. Their study compared the proposed structure-based clustering approach (MSTA) with a multiple-sequence-alignment (MSA) based method. The findings demonstrated that the MSTA-based method significantly outperformed the MSA approach in clustering the deaminase protein families. In this study, we investigated the effectiveness of our S-PLM model in clustering the deaminase family by comparing and

benchmarking the sequence embeddings from S-PLM against MSA, MSTA and two PLMs that depend on sequence alone, i.e., the ESM2 and PromptProtein. We utilized the identical sequence data of their study to generate representations for each query protein sequence. For this task, we utilized the S-PLM and obtained a 1280-dimensional embedding for each protein. Subsequently, we employed t-SNE to reduce the vector to a two-dimensional representation and applied K-Means clustering method to the reduced dimension. The Adjusted Rand Index (ARI) was computed by comparing K-Means clustering assignments with known deaminase family annotations. We conducted the t-SNE visualization and calculated the ARI in the same way for all the benchmarked methods. Based on the comparison (Fig. 4 a,b), S-PLM achieved very close results to MSTA (ARI 0.87 vs. 0.93) while surpassing the performance of the sequence-based alignment method (ARI of MSA: 0.55) and the other two PLMs reliant solely on sequence information (ARI: 0.63 and 0.46, respectively).

Kinases are a specific type of enzymes that facilitate the transfer of phosphate groups to proteins in a process known as phosphorylation, which plays a critical role in many biology processes[32]. In this study, we utilized kinases from established kinase groups and assessed the sequence embeddings generated by our S-PLM model for clustering purposes. We extracted 336 kinases, categorized into 9 kinase groups, along with their respective kinase domain sequences from GPS5.0[33]. Subsequently, sequence embeddings were generated for each kinase using its corresponding kinase domain sequences. For comparison, we also obtained sequence embeddings from other two PLMs: ESM2 (ESM2-t33_650M_UR50D) and PromptProtein. From the t-SNE plots (Fig. 4c) and the ARI comparison (Fig. 4d), the sequence embeddings produced by our S-PLM showed superior division of kinase groups with a significantly high ARI (0.72) compared to 0.28 for ESM2 and 0.28 for PromptProtein. This is likely because the S-PLM model incorporates protein structure information essential for determining kinase groups. Taken together, S-PLM provides effective sequence representation for enzyme clustering.

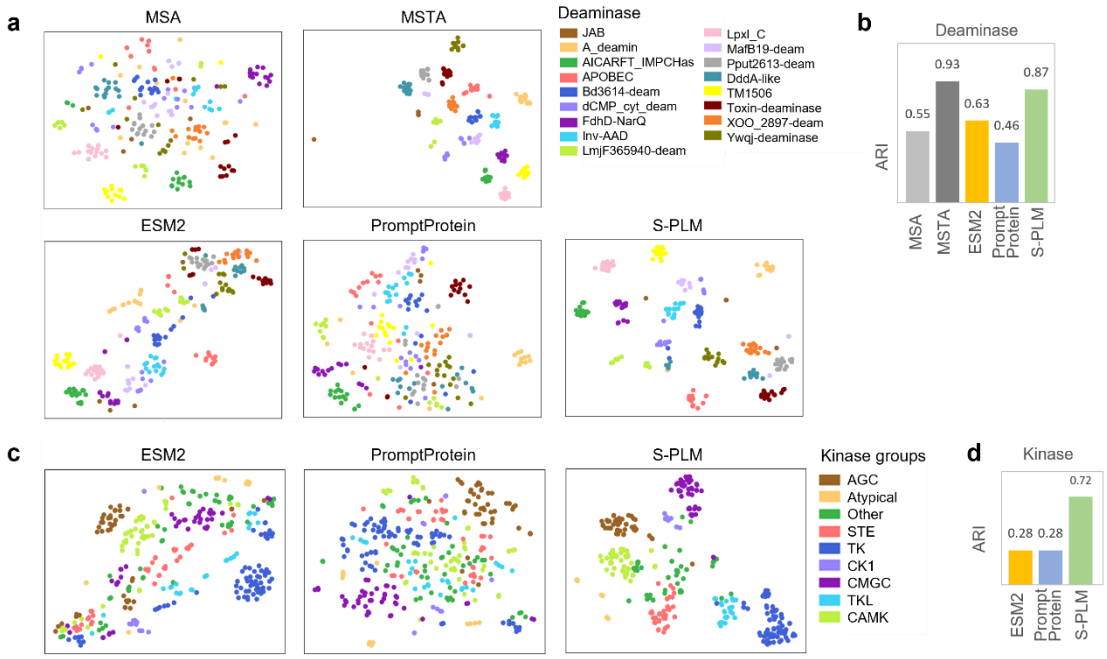

Figure 4. Clustering enzymes via S-PLM. **a**, The t-SNE visualization of protein embeddings from 242 deaminases proteins from alignment-based approaches: multiple-sequence-alignment (MSA)

and multiple-structure-alignment (MSTA); sequence representation methods: ESM2, PromptProtein, and S-PLM. **b**, Quantitative benchmarks of the ability of S-PLM to cluster the deaminase proteins compared with the methods mentioned in **a**. Adjusted Rand Index (ARI) was computed by comparing K-Means clustering assignments with known family annotations for the deaminase proteins. **c**, The t-SNE visualization of protein embeddings from human kinase groups via pre-trained PLM models: ESM2, PromptProtein, and S-PLM. Different kinase groups are distinguished by different colors. **d**, Quantitative benchmarks of the ability of S-PLM to cluster the human kinases compared with other sequence-only methods mentioned in **c**. Adjusted Rand Index (ARI) was computed by comparing K-Means clustering assignments with kinase group labels for the kinase domains.

S-PLM outperforms ESM2 for protein fold and enzyme reaction classification

We evaluated S-PLM for supervised protein fold and enzyme reaction classification, comparing the results with the base ESM2 encoder. The same classification layer was applied on top of each encoder, respectively. The comparison comprises two scenarios. In the first scenario, we treated both S-PLM and ESM2 as protein representation models, freezing all parameters of each encoder and allowing only the classification layers to be trainable. This approach ensures a fair comparison, as the trainable layers in both S-PLM and ESM2 were the same. In the second scenario, we fine-tuned the top K layers of both the S-PLM and ESM2 models, allowing the encoders to be trainable along with the classification layers. It is important to note that, given the different architectures of S-PLM and ESM2, achieving absolute fairness in this comparison is impossible, as the optimal configurations for both models may differ. Consequently, to align more closely with ESM2, all configurations were pre-selected based on ESM2's optimal performance, including the experiments in the first scenario. For scenario two, the key hyperparameter of ESM2 is K, *i.e.,* how many top Transformer layers are trainable. For Fold classification, we tried K=5,6,7 and for Enzyme reaction, we tried K= 1,2. Details regarding the model configurations can be found in Supplementary Table S1.

The comparative results were shown in Table 1. It demonstrated that the proposed S-PLM model consistently outperformed the base ESM2 models for all the classification tasks when all encoder layers were frozen (ESM2-fix *vs.* S-PLM-fix). When an equivalent number of transformer layers were fine-tuned (ESM2-finetune top K *vs.* S-PLM-finetune top K), S-PLM-finetune outperformed in all tasks except for Family classification. Despite a slight 0.2% decrease in Family classification, S-PLM-finetune demonstrated a notable 2%-3% improvement in Superfamily and Fold predictions. This highlights the significance of structural information in Fold and Superfamily predictions, while sequence information prevails in family prediction. We noted that the superior performance of S-PLM-finetune does not stem from more trainable parameters compared to ESM2-finetune. For instance, S-PLM-finetune top5 (116M) has fewer parameters than ESM2-finetune top6 (120M), yet it outperformed in Fold (37.74% vs. 36.63%) and Superfamily prediction (77.59% vs. 75.84%). In enzyme reaction classification, our model achieved 86.71% test accuracy, outperforming ESM2-finetune top2 (84.50%) with fewer parameters (23M vs. 40M). Taken together, these results support the benefits of integrating structural information into our S-PLM model, particularly for improving predictions in protein fold and enzyme reaction classifications, where structure features are crucial.

Table 1: Comparing S-PLM with ESM2 for protein fold and enzyme reaction classification.

Method	Protein classification			#Params
	Fold	Superfamily	Family	
ESM2-fix	27.86	57.26	98.11	1M
S-PLM-fix	**34.72**	**62.50**	**98.36**	1M
ESM2-finetune top5	34.96	76.24	**98.74**	99M
S-PLM-finetune top5	**37.74**	**77.59**	98.51	116M
ESM2-finetune top6	36.63	75.84	**98.82**	120M
S-PLM-finetune top6	**37.33**	**77.51**	98.35	139M
ESM2-finetune top7	36.49	76.56	**98.66**	140M
S-PLM-finetune top7	**37.19**	**77.99**	98.43	162M

Method	Enzyme reaction	#Params
ESM2-fix	78.84	0.5M
S-PLM-fix	**82.11**	0.5M
ESM2-finetune top1	86.53	20M
S-PLM-finetune top1	**86.71**	23M
ESM2-finetune top2	84.50	40M
S-PLM-finetune top2	**86.02**	46M

The performance is evaluated by accuracy (%). #Params represents number of trainable parameters. Performance is evaluated by accuracy (%). #Params represents number of trainable parameters. *K* in "top*K*" indicates how many top Transformer layers are trainable.

Lightweight tuning strategies help S-PLM to achieve the state-of-the-art performance for supervised protein predictions

In order to showcase the capabilities of S-PLM in supervised protein prediction tasks, we applied the developed lightweight tuning strategies, including fine-tuning top layers, adapter tuning (Fig. 1c), and LoRA (Fig. 1d), on the sequence encoder of S-PLM. We benchmarked S-PLM with these lightweight tuning strategies on GO term, EC number, and secondary structure predictions with state-of-the-art methods. We employed the same training, validation, and test sets for all these tasks, and the only differences are our models and the training strategies. The GO term and EC number prediction served as protein-level supervised classification tasks, which are evaluated in terms of the F_{max}, while the secondary structure (SS) prediction served as the residue-level supervised classification task evaluated by accuracy (%). The comparison results are shown in Table 2. The best and suboptimal results for each task were shown in bold and underlined symbols. Here, "-tune" indicates the results were trained by one of our lightweight tuning strategies, and the results for all three training strategies for each task and their model descriptions are provided in Supplementary Table S2. For GO and EC number predictions, the results of ESM2, CoupleNet, GearNet, and ESM-GearNet were taken from the publication[13]. For SS prediction, the results of ESM-1b[*] and DML[34] were taken from the PEER benchmark paper[35].

We can see from the comparison that our S-PLM-tune model achieved competitive performance in all these tasks; some of the results were the best (GO-BP:0.495, GO-MF:0.686 and SS prediction:87.48) and some of the results were suboptimal (For GO-CC, CoupleNet: 0.494 *vs.* S-PLM-tune: 0.484 and for EC number prediction, ESM-GearNet: 0.890 *vs.* S-PLM-tune:0.888). Noted that for the methods under comparison, some of the best results were

achieved using both protein sequences and structures as input (CoupleNet and ESM-GearNet). In contrast to CoupleNet and ESM-GearNet, our S-PLM-tune relies solely on protein sequences, and it achieves compatible results or even superior results, showcasing the broader practical utility of our model.

Table 2: Benchmarking on GO, EC, and secondary structure (SS) prediction.

Method	Inputs	Performance			
Protein-level					
		GO (F_{max})			EC (F_{max})
		BP	MF	CC	
ESM2 [13]	Seqs.	0.460	0.661	0.445	0.880
S-PLM-tune	Seqs.	**0.495**	**0.685**	<u>0.484</u>	<u>0.888</u>
CoupleNet [13]	Seqs.+Struct.	0.467	0.669	**0.494**	0.866
GearNet [13]	Struct.	0.356	0.503	0.414	0.730
ESM-GearNet [13]	Seqs.+Struct.	0.488	0.681	0.464	**0.890**
Residue-level SS (Accuracy%)					
ESM-1b* [35]	Seqs.	83.14			
DML [35]	Seqs.	<u>86.41</u>			
S-PLM-tune	Seqs.	**87.48**			

"Seqs." and "Struct." indicate the usage of protein sequence and structure in the model, respectively. [*] means the results are taken from literature. The best and suboptimal results for each task were shown in bold and underlined symbols.

Discussion and conclusion

In this work, we proposed S-PLM, a structure-aware protein language model via contrastive learning of sequence and 3D structure of proteins. Distinguishing from joint-embedding-based methods, S-PLM independently encodes protein sequences and 3D structures. This unique feature allows S-PLM to make predictions using only protein sequences, reducing the dependence on the protein's 3D structure. This aspect is particularly important when obtaining reliable protein structure is difficult or time-consuming. By employing a structure-adapter on the pretrained ESM2 model, our method generated sequence representations that seamlessly incorporate 3D structure information while maintaining the original sequence representation capabilities of ESM2. Importantly, this adapter is extensible; in the future, if new protein attributes need to be included, such as protein function and protein-protein interactions, the existing ESM2 and structure-adapter can remain unchanged. Simply adding a new adapter and training on the updated data will suffice.

The results of our experiments in this study highlight the efficacy of S-PLM across a range of downstream use cases. Firstly, S-PLM demonstrates the ability to align sequence and structure embeddings of the same protein effectively while keeping other embeddings from other proteins further apart. Secondly, compared to other PLMs that only use sequences, S-PLM shows impressive awareness of protein structures, as evidenced by its superior performance in clustering CATH domains. Thirdly, in enzyme clustering tasks, S-PLM outperforms both traditional sequence-alignment-based methods and other PLM methods significantly. Finally, together with S-PLM model, we developed a library of lightweight training strategies that can be applied to train S-PLM. Even without protein structure, S-PLM achieves competitive performance in protein supervised prediction tasks, comparable to the state-of-the-art methods that employing both

sequence and structure inputs. These findings collectively highlight the potential of S-PLM as an alternative protein representation model tool for protein analysis and prediction tasks.

This work also has some limitations and challenges. One challenge lies in the scale of S-PLM, like other large PLMs, requiring substantial amounts of training data for prediction tasks. Currently, we trained S-PLM on Alphafold2-SwissProt which used a relatively small database compared with other methods. The overall performance is expected to be enhanced when trained on larger protein structure databases with higher accuracy. Another limitation of the current S-PLM is its potential impact on the residue-level representation, as our contrastive loss function was applied to the averaged representation of all residues. To address this, one potential improvement is to introduce additional contrastive loss on the residue-level representations[36]. These areas present promising avenues for further exploration and improvement in the future.

Methods

1 Sequence encoder

Our sequence encoder was developed based on the pretrained ESM2 model[5]. Given the constraints of computational resources and model capacity, we selected ESM2-t33_650M_UR50D as our base PLM model, which has 650 million parameters. In particular, the input protein sequence was firstly tokenized by one-hot encoded for each amino acid, and then 33 layers of transformer encoders were applied, and the embedding dimension for each position was 1280. In this procedure, a BEGIN token (“<cls>”) and END token (“<eos>”) was added to the sequence and went through the transformer together with the amino acid tokens, and a “<pad>” token was used for padding sequences. Through the transformer layers, the outputs were a 1280D vector per-residue, the embedding of the begin and end tokens, as well as the embeddings for padding sequences. The embeddings for each residue were used for residue-level representations of a protein for downstream tasks. The average of per-residue embeddings, excluding the padding tokens was used for protein-level embedding for the contrastive-learning training and downstream tasks. Then, two projector layers were applied to the protein-level embedding that transformed the dimension into the final output protein-level embedding, which is 256D. The projector layers were two linear layers.

2 Structure encoder

We utilized the contact map to represent the protein 3D structure as it contains complete protein structure information, possesses inherent invariance capabilities, and is straightforward for implementation. Therefore, our structure encoder is specifically designed to encode the protein contact maps. Since the contact map representation resembles an image, the wide-adopted networks for image-related takes can be employed as the encoder. The ResNet50[37], Segment Anything Model (SAM)[38], and Swin-Transformer[25] models were under consideration due to their widespread popularity as image networks or their superior performance in various competitions. By evaluating them with pre-trained weights on the CATH domains with known protein structures (Supplementary Fig. S2), the Swin-Transformer (swinv2-tiny-patch4-window8-256) was finally applied because of its application enables more effective feature extraction from the contact map representation. To meet the requirement of the image network, which expects three input channels, we transformed our contact map into a representation with three channels. Given one sequence, the raw contact map was generated by calculating the coordinate distance between the C_α atoms for each amino acid for one sequence. In general, a contact map is a binary matrix

with a value of 1 if the pairwise distance is within a chosen threshold, indicating contact between the residues; otherwise, assign a value of 0.

In our case, we applied a distance threshold for each channel and converted the raw contact map into a continuous similarity matrix. Specifically, the distance threshold for each channel is d_i: [22Å, 22Å, 22Å], the same value that used in Alphafold2. The value of similarity matrix ranges from 0 to 1, with 1 indicating the shortest pairwise distance and 0 indicating the longest distance. The final contact map representation can be famulated as the following:

$$(d_i - CLIP(C_{seq}, 0, d_i))/d_i \tag{1}$$

where CLIP is a function that can change the C_{seq} values that are higher than d_i into d_i. By averaging the embeddings of the representation layer of Swin-Transformer and excluding the padding representation, we got the protein-level embedding of structure for the contrastive-learning training. Then, two projector layers were applied to the protein-level embedding that transformed the dimension into the final output protein-level embedding, which is 256D, the same as the final output protein-level embedding from sequence.

3 Multi-view contrastive learning

The objective of contrastive learning in this study is to attract the sequence embeddings and structure embeddings from the same protein closer and repel all the embeddings from different proteins further apart in the latent space. To achieve this, we applied a multi-view contrastive loss function to the protein-level embeddings obtained from the last projection layer of the sequence and structure encoders. The multi-view contrastive loss function was modified based on the *NT-Xent* (the normalized temperature-scaled cross entropy loss) in SimCLR[26]. In contrast to the SimCLR paper, the positive pair is only defined for sequence embedding (E^i_{seq}) and structure embedding (E^i_{str}) from the same protein (*i*) in our approach. Then the multi-view contrastive loss function for the positive pair (E^i_{seq},E^i_{str}) was defined in the following equation.

$$L_{E^i_{seq},E^i_{str}} = -\log \frac{\exp\left(\frac{sim\left(E^i_{seq},E^i_{str}\right)}{\tau}\right)}{\sum_{k=1}^{N} 1_{k\neq i}\exp\left(\frac{sim\left(E^i_{seq},E^k_{str}\right)}{\tau}\right)+1_{k\neq i}\exp\left(\frac{sim\left(E^i_{seq},E^k_{seq}\right)}{\tau}\right)+1_{k\neq i}\exp\left(\frac{sim\left(E^i_{str},E^k_{str}\right)}{\tau}\right))} \tag{2}$$

where $1_{k\neq i} \in \{0,1\}$ is an indicator function evaluating to 1 if $k \neq i$ and τ denotes a temperature parameter; *N* is the batch size. The $sim(x, y)$is a function to quantify the similarity between embeddings x and y, defined as follows:

$$sim(x, y) = x^T y/\|x\|\|y\| \tag{3}$$

This loss function helped to maximize the alignment of the embeddings for protein *i* in the two views (E^i_{seq} , E^i_{str}) and minimize the alignment between protein *i* and the other proteins ($k \neq i$). The final contrastive loss was calculated for all positive pairs within a mini-batch of samples.

4 Preparation of training dataset for contrastive learning and training process

We prepared the training database based on the proteins in the Swiss-Prot library. The amino acid sequences of the proteins were gained from the Swiss-Prot library and saved in the FASTA format. The 3D structures of the proteins were gained from the alphaFold2 repository. The C_α-C_α contact maps for individual proteins were determined using in-house Python code based on the alphaFold2 predicted 3D structures. We randomly selected 500 K proteins from the Swiss-Prot library for training and 41.5 K proteins for validation. The training was performed on a single A100

GPU using the computational facility of the University of Missouri. The batch size is 20 due to the GPU RAM limit. S-PLM was trained for more than 20 epochs.

5 Lightweight tuning on sequence encoder

Lightweight tuning strategy in this paper refers to training approaches that make specific and often parameter-efficient modifications to a pre-existing model, reducing the computational resources and memory needed compared to training a model from fully fine-tuning. We implemented the fine-tune top layers, LoRA, and adapter tuning on the sequence encoder of S-PLM as shown in Fig. 1 for downstream protein sequence prediction tasks. The details for each strategy are as follows:

Fine-tune top layers: In the ESM2 backbone model we used, there are 33 transformer layers in total. Here, "fine-tune top layers" refers to fine-tuning only the top $K \leq 33$ transformer layers and freezing the remainder. Here K is a hyperparameter in our configuration.

LoRA: LoRA refers to Low-Rank adaptation[20], which freezes the pre-trained model weights and injects trainable rank decomposition matrices into each layer of the Transformer architecture. It can greatly reduce the number of trainable parameters for downstream tasks. For a pretrained weight matrix $W_0 \in \mathbb{R}^{dxk}$, its updates ΔW can be represented as a low-rank decomposition $\Delta W = BA,\ where\ B \in \mathbb{R}^{dxr}, A \in \mathbb{R}^{rxk}$, and the rank $r \ll \min(d, k)$. During training, W_0 is frozen, A and B contain trainable parameters. For an input x , the LoRA forward pass yields:

$$h = W_0 + \Delta W x = W_0 + BAx$$

In the implementation of LoRA, the is scaled by $\frac{\alpha}{r}$, where α is a constant in r. In our implementation, the low-rank decomposition matrices were applied to the query, key, value, and output projection matrices in the self-attention module of the top *K* Transformer layers of ESM2 (Fig. 1d). Here, K, α, and r are hyperparameters in our configuration.

Adapter tuning: Adapter tuning involves integrating adapter modules into the Transformer layer of the ESM2 model. The adapter modules were implemented according to [19]. It is positioned twice in one Transformer layer of ESM2: after the self-attention projection and after the two feed-forward. Each adapter module consisted of a bottleneck structure and a skip-connection. The bottleneck structure compressed the input data into a reduced-dimensional space and then reconstructs the data to restore it to the original input dimension. The bottleneck structure enabled the adapter module with few parameters relative to the attention and feedforward layers in the original Transformer. The integration of the ESM2 model with the structure-aware adapter modules (Struct-Adapter) is illustrated in Fig. 1b. Unlike the original adapter tuning[19], which applied the adapter modules into all Transformer layers, we specifically insert them into the top K Transformer layers of ESM2. Here K is a hyperparameter in our configuration. Adapter tunning for supervised downstream tasks is implemented by integrating a list of additional paralleled adapters on the Struct-Adapter modules, as shown in Fig. 1c. The hyperparameter used and the number of trainable parameters for each strategy for all the downstream tasks are shown in Supplementary Table S2.

6 Data preprocessing for CATH superfamily clustering

The sequence data with CATH annotations were downloaded from the CATH database (release v4_3_0)[28] and the cath-dataset-nonredundant-S40-v4_3_0 dataset whose proteins of maximally

40% sequence identity was used. We only kept sequences that have records of known protein structures in the PDB database. To reduce computational load and for clearer visualization, from one CATH superfamily, only one represented sequence with the longest sequence length was selected. We considered all the top three represented categories of Class (1553 sequences), top five represented categories of Architecture (1049 sequences), and top five represented categories of Topology (306 sequences) from the CATH hierarchy. The selected CATH sequences were provided in Supplementary Data 1.

7 Downstream supervised protein prediction tasks

The downstream supervised protein prediction tasks include four protein-level prediction tasks: the protein fold classification, enzyme reactions classification, Gene Ontology (GO) term prediction, and enzyme commission (EC) numbers prediction; and one residue-level prediction task: protein secondary structure prediction.

7.1 Fold classification

Protein fold prediction is utilized to accurately determine the fold class label of a given protein. We use the same dataset splits from [39]. The dataset comprises a total of 16,712 proteins, classified into 1,195 distinct fold classes. In their dataset split, three test sets are used: ‘Fold’, ‘Superfamily’, and ‘Family’. In the 'Fold' set, training is conducted, excluding proteins from the same superfamily. In the 'Superfamily' set, proteins from the same family are excluded from training. In the 'Family' set, proteins belonging to the same family are included in the training process.

7.2 Enzyme reaction classification

The goal of enzyme reaction classification is to determine the class of enzyme-catalyzed reactions for a protein, utilizing all four levels of Enzyme Commission (EC) numbers [40]. Following the methodology described in [41], the dataset was split into training, validation, and test sets, comprising 37,428 proteins categorized across 384 four-tiered Enzyme Commission (EC) numbers. All proteins have less than 50% sequence similarity across the data splits.

7.3 Gene ontology term prediction

The objective of Gene Ontology (GO) term prediction is to ascertain the association of a protein with a specific GO term, including three tasks: biological process (BP), molecular function (MF), and cellular component (CC). Each task is formulated as a multi-label classification problem. We use the same dataset splits from [42] where the test set has up to 95% (30%, 40%, 50%, 70% and 95%) sequence similarity with the training set.

7.4 Enzyme commission number prediction

The objective of this task is to identify the 538 distinct third and fourth level Enzyme Commission (EC) numbers, which characterize the catalytic functions of various proteins in biochemical reactions. It is formulated as a multi-label classification problem. We use the same dataset splits from [42], where the test set has up to 95% (30%, 40%, 50%, 70% and 95%) sequence similarity with the training set.

7.5 Secondary structure prediction

The goal of this task is to predict the local structures of protein residues into three secondary structure labels (*i.e.,* coil, strand, or helix). We used the same dataset splits in the reference[35], where the Klausen's dataset[43] was adopted as the training set and the CB513 dataset[44] was used as the test set. There are no two proteins that have greater than 25% sequence identity and the test set has less than 25% sequence identity against the training set. Same as in the reference[35], we truncated those sequences with more than 1022 residues by keeping the first 1022 residues. The statistics of the dataset for each task is shown in Supplementary Table S3. The source code and data can be found at https://github.com/duolinwang/S-PLM/ .

Acknowledgments

D.W. and D.X. are partially supported by the National Institutes of Health (grant R35GM126985). U.A., Q.S., and J.C. would like to thank AI in Medicine (AIM), University of Kentucky (NCATS UL1TR001998, NCI P30 CA177558). U.A. and Q.S. would also like to thank the financial support from the Start-up Fund of the University of Kentucky and Alzheimer Association AARG-23-1144638. Q.S. also acknowledges the financial support of National Science Foundation (2154996). We thank the University of Kentucky Center for Computational Sciences and Information Technology Services Research Computing for their support and use of the Morgan Compute Cluster and associated research computing resources. This work also used Delta-GPU at NCSA through allocation CIS230053 from the Advanced Cyberinfrastructure Coordination Ecosystem: Services & Support (ACCESS) program, which is supported by National Science Foundation grants #2138259, #2138286, #2138307, #2137603, and #2138296.

References

1. Ofer, D., Brandes, N. & Linial, M. The language of proteins: NLP, machine learning & protein sequences. Comput Struct Biotechnol J 19, 1750–1758 (2021).

2. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint at https://doi.org/10.48550/arXiv.1810.04805 (2018).

3. Elnaggar, A., Heinzinger M, Dallago C., Rehawi G., Wang Y., Jones L., Gibbs T., Feher T., Angerer C., Steinegger M., Bhowmik D. & Rost B. ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning. IEEE Trans Pattern Anal Mach Intell 44, 7112–7127 (2022).

4. Lin, Z., Akin H., Rao R., Hie B., Zhu Z., Lu W., Smetanin N., Verkuil R., Kabeli O., Shmueli Y., Dos Santos Costa A., Fazel-Zarandi M., Sercu T., Candido S. & Rives A. Evolutionary-scale prediction of atomic-level protein structure with a language model. Science vol. 379 1123–1130 (2023).

5. Rives, A., Meier J., Sercu T., Goyal S., Lin Z., Liu J., Guo D., Ott M., Zitnick CL., Ma J. & Fergus R. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci U S A 118, (2021).

6. Vu, M. H., Akbar, R., Robert, P. A., Swiatczak, B., Sandve, G. K., Greiff, V. & Haug, D. T. T. Linguistically inspired roadmap for building biologically reliable protein language models. Nature Machine Intelligence 5, 485–496 (2023).

7. Wang, B., Xie, Q., Pei, J., Chen, Z., Tiwari, P., Li, Z., & Fu, J. Pre-trained language models in biomedical domain: A systematic survey. ACM Computing Surveys 56, 1–52 (2023).

8. Bepler, T. & Berger, B. Learning the protein language: Evolution, structure, and function. Cell Syst 12, 654-669.e3 (2021).

9. Jumper, J., Evans R., Pritzel A., Green T., Figurnov M., Ronneberger O., Tunyasuvunakool K., Bates R., Žídek A., Potapenko A., Bridgland A., Meyer C., Kohl SAA., Ballard A.J ., Cowie A., Romera-Paredes B., Nikolov S., Jain R., Adler J., Back T., Petersen S., Reiman D., Clancy E., Zielinski M., Steinegger M., Pacholska M., Berghammer T., Bodenstein S., Silver D., Vinyals O., Senior AW., Kavukcuoglu K., Kohli P. & Hassabis D. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589 (2021).

10. Rao, R. M., Liu, J., Verkuil, R., Meier, J., Canny, J., Abbeel, P., Sercu, T. & Rives, A. MSA Transformer. in Proceedings of the 38th International Conference on Machine Learning (eds. Meila, M. & Zhang, T.) vol. 139 8844--8856 (PMLR, Proceedings of Machine Learning Research, 2021).

11. Brandes, N., Ofer, D., Peleg, Y., Rappoport, N. & Linial, M. ProteinBERT: a universal deep-learning model of protein sequence and function. Bioinformatics vol. 38 2102–2110 (2022).

12. Chen, C. S., Zhou, J., Wang, F., Liu, X. & Dou, D. Structure-aware protein self-supervised learning. Bioinformatics 39, (2023).

13. Zhang, Z., Wang, C., Xu, M., Chenthamarakshan, V., Lozano, A. C., Das, P. & Tang, J. A Systematic Study of Joint Representation Learning on Protein Sequences and Structures. Preprint at http://arxiv.org/abs/2303.06275 (2023).

14. Hu, B., Tan, C., Xia, J., Zheng, J., Huang, Y., Wu, L., Liu, Y., Xu, Y. & Li, S. Z. Learning Complete Protein Representation by Deep Coupling of Sequence and Structure. Preprint at http://biorxiv.org/lookup/doi/10.1101/2023.07.05.547769 (2023)

15. Ramesh, A., Nichol, A., Chu, C. & Chen, M. Hierarchical Text-Conditional Image Generation with CLIP Latents. Preprint at http://arxiv.org/abs/2204.06125 (2022).

16. Tian, Y., Krishnan, D. & Isola, P. Contrastive multiview coding. in Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16 776–794 (Springer, 2020).

17. Xu, M., Yuan, X., Miret, S. & Tang, J. Protst: Multi-modality learning of protein sequences and biomedical texts. Preprint at https://doi.org/10.48550/arXiv.2301.12040 (2023).

18. Yu, T., Cui, H., Li, J. C., Luo, Y., Jiang, G. & Zhao, H. Enzyme function prediction using contrastive learning. Science 379, 1358–1363 (2023).

19. Houlsby, N., Giurgiu, A., Jastrzebski, S. & Morrone, B. Parameter-Efficient Transfer Learning for NLP. preprint at https://doi.org/10.48550/arXiv.1902.00751

20. Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S. & Chen, W. LoRA: Low-Rank Adaptation of Large Language Models. Preprint at http://arxiv.org/abs/2106.09685 (2021).

21. Sledzieski, S., Kshirsagar, M., Baek, M., Berger, B., Dodhia, R. & Ferres, J. M. L. Democratizing Protein Language Models with Parameter-Efficient Fine-Tuning. Preprint at https://doi.org/10.1101/2023.11.09.566187 (2023)

22. Zeng, S., Wang, D. & Xu, D. PEFT-SP: Parameter-Efficient Fine-Tuning on Large Protein Language Models Improves Signal Peptide Prediction. Preprint at https://doi.org/10.1101/2023.11.04.565642 (2023).

23. Barrett, A. J. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). Enzyme Nomenclature. Recommendations 1992. Supplement 4: corrections and additions (1997). Eur J Biochem 250, 1–6 (1997).

24. The Gene Ontology project in 2008. Nucleic Acids Res 36, D440-444 (2008).

25. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S. & Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. in Proceedings of the IEEE/CVF international conference on computer vision, 10012–10022 (2021).

26. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. in International conference on machine learning 1597–1607 (PMLR, 2020).

27. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. Learning transferable visual models from natural language supervision. in International conference on machine learning 8748–8763 (PMLR, 2021).

28. Sillitoe, I.P., Bordin, N., Dawson, N.L., Waman, V.P., Ashford, P., Scholes, H.M., Pang, C.S., Woodridge, L., Rauer, C., Sen, N., Abbasian, M., Cornu, S.L., Lam, S.D., Berka, K., Vareková, I.H., Vareková, R.S., Lees, J.G., & Orengo, C.A. CATH: increased structural coverage of functional space. Nucleic acids research 49, D266–D273 (2021).

29. Wang, Z., Zhang, Q., Hu, S., Yu, H., Jin, X., Gong, Z., & Chen, H. Multi-level Protein Structure Pre-training via Prompt Learning. The Eleventh International Conference on Learning Representations. 2022.

30. Caliński, T. & Harabasz, J. A dendrite method for cluster analysis. Communications in Statistics-theory and Methods 3, 1–27 (1974).

31. Huang, J., Lin, Q., Fei, H., He, Z., Xu, H., Li, Y., Qu, K., Han, P., Gao, Q., Li, B., Liu, G., Zhang, L., Hu, J., Zhang, R., Zuo, E., Luo, Y., Ran, Y., Qiu, J., Zhao, K.T., & Gao, C. Discovery of deaminase functions by structure-based protein clustering. Cell 186, 3182-3195.e14 (2023).

32. Coussens, L.M., Parker, P.J., Rhee, L., Yang-Feng, T.L., Chen, E.Y., Waterfield, M.D., Francke, U., & Ullrich, A. Multiple, distinct forms of bovine and human protein kinase C suggest diversity in cellular signaling pathways. Science 233, 859–866 (1986).

33. Wang, C., Xu, H., Lin, S., Deng, W., Zhou, J., Zhang, Y., Shi, Y., Peng, D., & Xue, Y. GPS 5.0: An Update on the Prediction of Kinase-specific Phosphorylation Sites in Proteins. Genomics Proteomics Bioinformatics 18, 72–80 (2020).

34. Yang, W., Liu, Y. & Xiao, C. Deep metric learning for accurate protein secondary structure prediction. Knowledge-Based Systems 242, 108356 (2022).

35. Xu, M., Zhang, Z., Lu, J., Zhu, Z., Zhang, Y., Ma, C., Liu, R., & Tang, J. PEER: a comprehensive and multi-task benchmark for protein sequence understanding. Advances in Neural Information Processing Systems 35, 35156–35173 (2022).

36. Birnbaum, F., Jain, S., Madry, A. & Keating, A. E. Jointly Embedding Protein Structures and Sequences through Residue Level Alignment. Machine Learning for Structural Biology Workshop, NeurIPS (2023).

37. He, K., Zhang, X., Ren, S. & Sun, J. Deep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition 770–778 (2016).

38. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W., Dollár, P., & Girshick, R.B. Segment anything. arXiv preprint arXiv:2304.02643 (2023).

39. Hou, J., Adhikari, B. & Cheng, J. DeepSF: deep convolutional neural network for mapping protein sequences to folds. Bioinformatics vol. 34, 1295–1303 (2017).

40. Omelchenko, M. V., Galperin, M. Y., Wolf, Y. I. & Koonin, E. V. Non-homologous isofunctional enzymes: A systematic analysis of alternative solutions in enzyme evolution. Biology Direct 5, 31 (2010).

41. Hermosilla, P., Schäfer, M., Lang, M., Fackelmann, G., Vázquez, P., Kozlíková, B., Krone, M., Ritschel, T., & Ropinski, T. Intrinsic-Extrinsic Convolution and Pooling for Learning on 3D Protein Structures. Preprint at https://doi.org/10.48550/arXiv.2007.06252 (2021).

42. Gligorijević, V., Renfrew, P.D., Kosciólek, T., Leman, J.K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B.C., Fisk, I., Vlamakis, H., Xavier, R.J., Knight, R., Cho, K., & Bonneau, R. Structure-based protein function prediction using graph convolutional networks. Nature Communications 12, 3168 (2021).

43. Klausen, M.S., Jespersen, M.C., Nielsen, H.B., Kjærgaard Jensen, K., Jurtz, V.I., Sønderby, C.K., Sommer, M.O., Winther, O., Nielsen, M., Petersen, B., & Marcatili, P. NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning. Proteins 87, 520–527 (2019).

44. Cuff, J. A. & Barton, G. J. Evaluation and improvement of multiple sequence methods for protein secondary structure prediction. Proteins 34, 508–519 (1999).