Journal of Cheminformatics

RESEARCH Open Access

PETA: evaluating the impact of protein transfer learning with sub-word tokenization on downstream applications

Yang Tan[1,2,3,4†], Mingchen Li[1,2,3,4†], Ziyi Zhou[2], Pan Tan[2,3], Huiqun Yu[1*], Guisheng Fan[1*] and Liang Hong[2,3,4*]

Abstract Protein language models (PLMs) play a dominant role in protein representation learning. Most existing PLMs regard proteins as sequences of 20 natural amino acids. The problem with this representation method is that it simply divides the protein sequence into sequences of individual amino acids, ignoring the fact that certain residues often occur together. Therefore, it is inappropriate to view amino acids as isolated tokens. Instead, the PLMs should recognize the frequently occurring combinations of amino acids as a single token. In this study, we use the byte-pair-encoding algorithm and unigram to construct advanced residue vocabularies for protein sequence tokenization, and we have shown that PLMs pre-trained using these advanced vocabularies exhibit superior performance on downstream tasks when compared to those trained with simple vocabularies. Furthermore, we introduce PETA, a comprehensive benchmark for systematically evaluating PLMs. We find that vocabularies comprising 50 and 200 elements achieve optimal performance. Our code, model weights, and datasets are available at https://github.com/ginnm/ProteinPretraining.

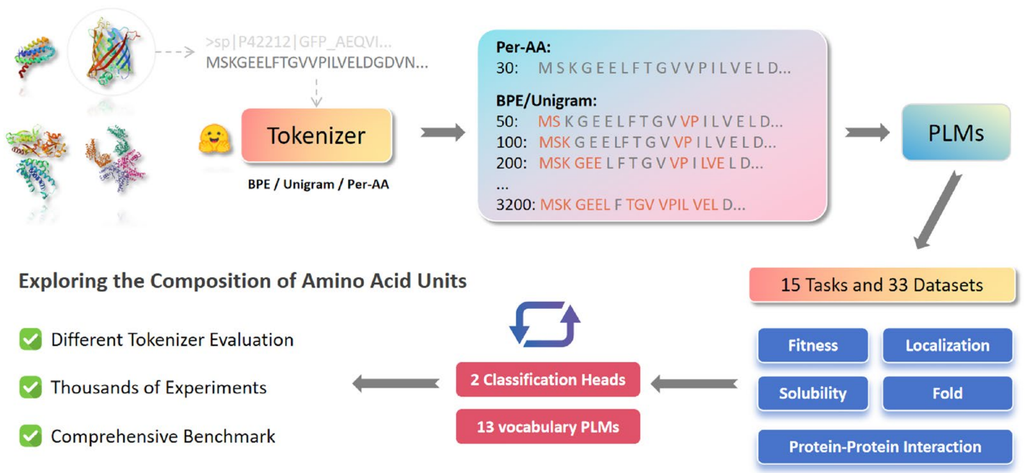
Scientific contribution This study introduces advanced protein sequence tokenization analysis, leveraging the byte-pair-encoding algorithm and unigram. By recognizing frequently occurring combinations of amino acids as single tokens, our proposed method enhances the performance of PLMs on downstream tasks. Additionally, we present PETA, a new comprehensive benchmark for the systematic evaluation of PLMs, demonstrating that vocabularies of 50 and 200 elements offer optimal performance.

Keywords Protein language model, Protein tokenization, Vocabulary size, Evaluation benchmark

[†] Yang Tan and Mingchen Li have contributed equally to this work.

*Correspondence:
Huiqun Yu
yhq@ecust.edu.cn
Guisheng Fan
gsfan@ecust.edu.cn
Liang Hong
hongl3liang@sjtu.edu.cn
Full list of author information is available at the end of the article

Graphical Abstract


>sp|P42212|GFP_AEQVI...
MSKGEELFTGVVPILVELDGDVN...
Tokenizer
BPE / Unigram / Per-AA
Per-AA:
30: M S K G E E L F T G V V P I L V E L D...
BPE/Unigram:
50: MS K G E E L F T G V VP I L V E L D...
100: MSK G E E LF T G V VP I L V E L D...
200: MSK GEE L F T G V VP I LVE L D...
...
3200: MSK GEEL F TGV VPIL VEL D...
PLMs
15 Tasks and 33 Datasets
Fitness
Localization
Solubility
Fold
Protein-Protein Interaction
2 Classification Heads
13 vocabulary PLMs
Exploring the Composition of Amino Acid Units
✓ Different Tokenizer Evaluation
✓ Thousands of Experiments
✓ Comprehensive Benchmark

Introduction

Proteins play a pivotal role in sustaining life forms and have found extensive applications in human endeavors, including gene editing [1, 2], drug discovery [3], and enzymatic catalysis [4]. Furthermore, gaining insights into protein properties or enhancing their functionality holds significant practical value, such as enhancing the function of the original protein [5] or annotating an unknown sequence [6]. Protein engineering typically follows two common approaches: laboratory-based experiments and computation-based methods. The former includes structural analysis [7], expression purification [8] and direct evolution [9], while valuable, are time-consuming and heavily reliant on domain-specific knowledge. This limitation falls short of meeting the evolving demands of both the scientific community and industry. Conversely, the computation-based modeling strategy relies on machine-learning or physics-based methods that are often not particularly accurate but are cost-effective and time-saving. Thanks to the advancements in protein sequencing technology [10], new avenues have opened up for training large-scale protein models capable of capturing a more comprehensive understanding. For instance, ESM series [11–13] to leverage the UniProt database [14] which contains over 200 million protein sequences or its subsets for training purposes.

In recent years, there has been significant development in pre-trained protein language models [11–13, 15–19]. A protein will be tokenized into a sequence of bio-tokens (per-AA or multi-aa) and then use a pre-trained Transformer to convert this sequence into dense vectors, which serve as representations of the protein. Typically, these models tokenize the protein sequence by naturally dividing it into its constituent amino acids. The vocabulary size for amino acids is approximately 20, contrasting with natural language models, which often encompass thousands of words or sub-words. Many tokenization algorithms [20, 21] have been effectively employed in language models to replace character-level tokenization by grouping frequently co-occurring characters into words and show different performance in pre-trained human language models [22, 23]. However, we discovered that, in the domain of proteins, there had been no systematic research evaluating how different tokenization algorithms impact protein language models. In this paper, we draw inspiration from [24] and aim to develop a universal amino acid coding approach capable of delivering robust performance across various protein-related tasks, while harnessing the benefits of knowledge sharing and transfer as shown in Fig 1. A recent study [25] has shown reducing vocabulary size will decrease the model's performance and distort evolutionary information, and we conduct a more comprehensive study on increasing the alphabets based on pre-training. To facilitate a thorough assessment and take cues from the success of benchmark datasets in domains like computer vision and natural language processing, e.g., ImageNet [26] and GLUE [27], we have meticulously curated a collection of 33 datasets categorized into 15 distinct tasks. These datasets are integral to advancing the realm of deep learning in protein comprehension. Our PETA benchmark encompasses five groups of tasks, including protein fitness prediction, protein localization prediction, protein-protein interaction prediction, protein solubility prediction, and protein

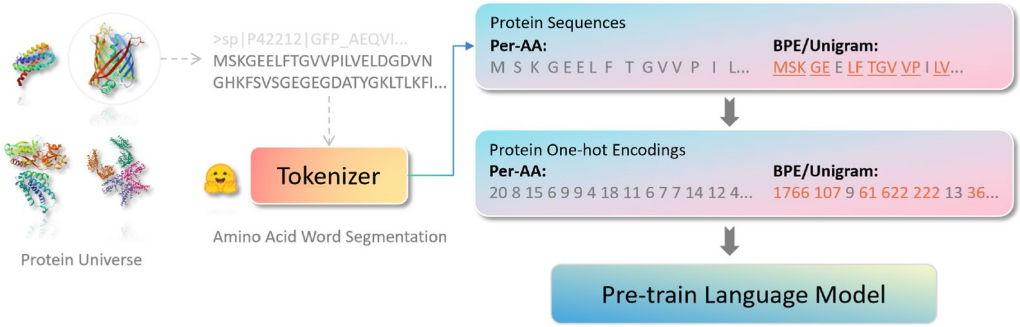


Fig. 1 The protein sequence is formed into a new discrete token sequence through different word segmentation methods. As the size of the vocabulary increases, the amino acid composition of a single token becomes more complex

fold prediction. For each individual dataset, we evaluate the performance of three types of tokenizers, two new residue-pair tokenizers are used to train five models with distinct vocabularies respectively and one per-amino-acid (Per-AA) model acts as baseline. Two different pooling mechanisms and three random seeds are employed in downstream tasks to mitigate potential classification biases. We anticipate that our comprehensive analysis of protein tokenizers and the PETA benchmark will serve as a pivotal milestone for the continued advancement of protein language model pre-training.

Our contributions are as follows:

- Creation of the PETA Benchmark: We meticulously curate the PETA benchmark, a comprehensive collection of 33 datasets categorized into 15 distinct protein-related tasks. This benchmark spans 5 diverse aspects of protein research. It provides a standardized evaluation framework for protein language models.
- Protein Tokenization Analysis: We summarize how the amino acid coding approach enhances the effectiveness of protein language models across diverse protein-related tasks. By addressing the influence of amino acid combinations, the research offers valuable insights into the optimization of protein language models.
- Comprehensive Experiments: We have pre-trained 13 protein language models with 3 types of tokenizers, and thousands of downstream experimental evaluations are conducted to ensure the validity of the results. The model weights, code, etc. are completely open-source in the community.

Related work

Protein representation learning

Representation learning harnesses knowledge acquired from large-scale corpora to generalize across various tasks. Early approaches primarily employed machine learning techniques from natural language processing, such as word2vec [28] and doc2vec [29], to extract features from protein sequences [30–32]. Recently, deep learning has exhibited tremendous potential by enabling models with increased capacity and deeper encoders, capable of handling millions or billions of protein sequences. ESM-1V [12], SESNet [33], and ECNet [34], which focus on predicting mutation fitness. Additionally, ESM-1b [13] and ESM-2 [11] employ mask language modeling. ProtTrans [18] pre-trains language models under various architectures [35–39], while XTrimo [40] aligns its pre-trained architecture with GLM [41]. Ankn [19] uses an asymmetric encoder and decoder framework and different mask probabilities to improve the pre-training performance. CPCProt [42] leverages a contrastive predictive coding loss, whereas ProGen [15, 16], UniRep [43], ProXLNet [18], ProtGPT2 [17], and Tranception [44] are pre-trained using next amino acid prediction tasks. Although many of these approaches share common objectives with natural language processing, there are also innovations like ProteinBERT [45] and CARP [46] which employ convolutional networks for downstream tasks. Some works delve into protein multiple sequence alignments (MSAs) [44, 47, 48], while others take structure-based approaches to extract topology information for inverse folding [49–51], protein design [52, 53], and protein engineering [54]. Notably, LM-GVP [55], MIF-ST [50], and ProtSSN [56] integrate sequence and structural information to enhance the learning of effective protein representations [57, 58]. In this benchmark, our primary focus revolves around evaluating the performance of language models utilizing different tokenization strategies.

Protein modeling benchmarks

A comprehensive benchmark has shown great influence in the traditional computer science community and driven the research direction of different works [26, 27, 59–61]. However, it is worth noting that the field of computing protein engineering still lacks a large-scale benchmarking framework. In contrast, the biennial Critical Assessment of Protein Structure Prediction (CASP) [62] has emerged as a gold standard for assessing advancements in protein structure prediction. In tandem with CASP, the Critical Assessment of Functional Annotation (CAFA) challenge [63] has been established to evaluate the prediction of protein functions. Several notable works, such as DeepSequence [64], Envision [65], and ProteinGym [44], focus on measuring very different functional fitness variations in response to diverse protein modifications, including substitutions and insertions/deletions. Techniques like deep mutational scanning (DMS) [66] and other protein engineering methods are used to build up these datasets. On the other hand, works like SoluProtMutDB [67], SKEMPI [68], and ProThermDB [69] concentrate on assessing specific properties concerning single amino acid variations (SAVs). Additionally, FLIP [70] offers various data partitioning methods across three protein landscapes for evaluating fitness prediction. The TAPE benchmark [71] encompasses five tasks, with three focusing on structure prediction and the remaining two targeting fitness prediction. PEER [72] encompasses seventeen biologically relevant tasks spanning five aspects of protein understanding. ProteinGLUE [73] comprises seven downstream tasks designed for self-supervised protein representation learning. DeepLoc [74, 75] provides datasets for subcellular localization classification. The STRING database [76] annotates protein-protein interactions (PPIs) with seven types of interactions. TDA [77] generates protein-related datasets and tasks tailored for drug discovery. ESOL website [78] aggregates solubility scores for ensemble E.coli proteins.

Methods

We designed PETA to answer two important questions:

- Is residue-wise tokenizer good enough for protein language model pre-training?
- How do different vocabulary sizes influence the representation ability on downstream tasks?

Most of the works choose one tokenizer aligned with previous research without much concern, to answer the first question, we utilize three amino acid segmentation strategies including residue-wise and sub-word tokenizers. For the second question, we design larger vocabulary size arranged from {50, 100, 200, 800, 1600, 3200} for the Unigram and BPE segmentation methods. The model trained under per-AA is the baseline, it has a vocabulary size of 20 common amino acids and many works adopt this [11–13, 18, 49]. It is worth noting that these vocabulary sizes do not contain special tokens, such as $< mask >$ or $< pad >$. In general, we utilize three tokenization methods, two types of classification heads, and two model pipelines to solve different tasks in the PETA benchmark. The framework of PETA is shown in Fig 2.

Amino acid segmentation

In this study, we utilize three classic sequence segmentation methods: per-amino-acid encoding (Per-AA), byte pair encoding (BPE) [20], and unigram language modeling (Unigram) [21] as shown in Fig 1. Per-AA focuses on individual amino acid units, enabling high-resolution analysis of subtle variations. BPE offers flexibility by segmenting sequences into subunits, effectively capturing structural information, while Unigram, based on character-level statistics, captures global sequence characteristics. These diverse methods collectively enhance our comprehensive analysis of protein sequences, each serving a unique role in addressing specific analytical requirements.

Byte pair encoding

BPE segments sequences into the most frequent subunits or tokens. This method iteratively merges the most frequently adjacent pairs of characters into a single token, thus reducing the vocabulary size and simplifying the model's complexity. In the context of protein sequences, BPE helps in identifying and encoding common motifs and structural domains that are crucial for functional characterization. By applying BPE, we can efficiently manage and interpret large datasets, as it bridges the gap between amino acid-level granularity and holistic sequence representation. This method is widely used in various natural language models such as GPT [79] and BERT [36].

Unigram language modeling

Unigram modeling simplifies text segmentation by independently calculating the likelihood of each word based on how often it appears in the data. Unlike the BPE method, which looks at the frequency of pairs of adjacent characters, Unigram creates a list of words by finding the most likely combination of tokens to form a language model. This approach is especially useful for analyzing protein sequences because it can identify rare, yet important, amino acids or patterns that other methods might miss. Additionally, Unigram's probabilistic approach allows it to adjust the vocabulary flexibly according to the context of the sequence, making it adaptable to new or uncommon variations. This flexibility makes it suitable for working with multiple languages or in situations with limited language data [80].

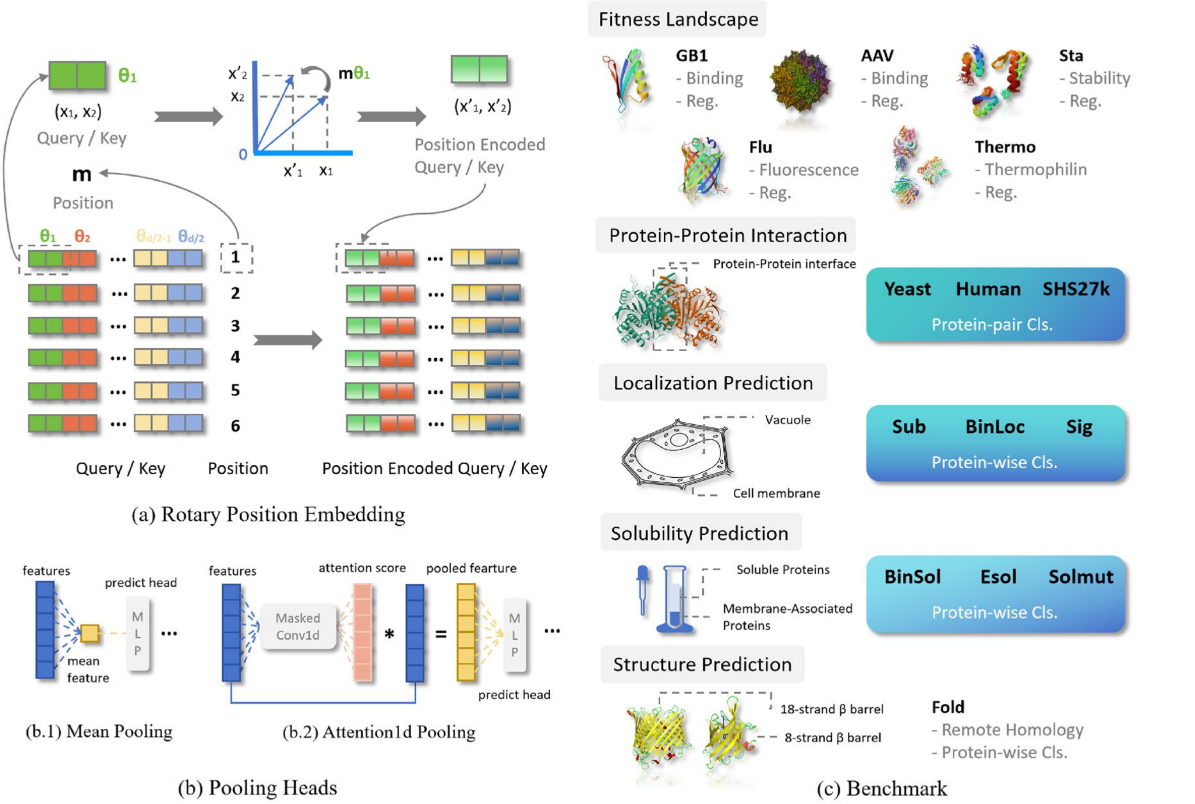


Fig. 2 The framework of PETA. (a) Pre-trained models use rotary position embedding, which possesses favorable theoretical properties and is an absolute positional encoding applicable to linear Attention. (b) We employed two distinct classification heads, namely mean pooling and attention1d pooling. The former is the most commonly used method at present, while the latter is relatively more complex. (c) Our benchmark comprises 15 downstream tasks, which can be categorized into five groups. Some of these downstream tasks include multiple datasets or data splitting methods, amounting to a total of 33 datasets

Pre-training protein language models

Model architecture

Our pre-training architecture employs RoFormer [81], an autoencoding model that adopts a BERT-like structure augmented with rotary positional embeddings, as illustrated in Fig. 2 (a). These rotary positional embeddings effectively harness positional information within sequences. Detailed hyperparameter configurations are delineated in "Experimental Setups" section. Initially, protein sequences are tokenized and transformed into one-hot encoded representations. These representations are subsequently fed into RoFormer's encoder, which generates sets of hidden states that maintain the length consistency with the input tokenized sequence. Finally, these hidden states are transformed into a vector with a dimensionality corresponding to the vocabulary size, upon which a softmax function is applied to yield the reconstruction probability density distribution.

Pre-training objective

We employ the mask language modeling (MLM) objective for pre-training our models [35]. Given an input sequence, a subset of tokens is selected at random and replaced with a special mask token. The model is then trained to predict these masked tokens based on the unmasked context tokens. The loss function for this objective can be defined as:

$$L_{MLM} = E_{x \sim X} E_M \sum_{x} -\log p(x_i | x_{/M}) \tag{1}$$

here, x is a sequence from the dataset X, and $x_{/M}$ represents the sequence with masked tokens removed. $p(x_i|x_{/M})$ is the conditional probability of predicting the correct token x_i given the context $x_{/M}$. The aim is to minimize the negative log-likelihood of the true token at each masked index i, which in this case are amino acids, given the unmasked sequence as context. Intuitively, the model must learn to identify the dependencies between the masked and unmasked tokens to successfully predict the masked positions.

Language modeling perplexity

We use *Perplexity (PPL)* to evaluate the performance of the MLM, computed as:

$$\mathrm{PPL} = \exp(-\frac{1}{N}\sum_{i=1}^{N}\log p(x_i|x)) \tag{2}$$

where N is the number of masked tokens, as well as x_i is the *ith* token of sequence. To account for potential unfair comparisons arising due to varying vocabulary sizes across different models, we introduce the metric of *Normalized Perplexity (NPPL)* range from 1 to positive infinity. The formula for Normalized Perplexity is as follows:

$$\mathrm{NPPL} = \exp\,(-\frac{1}{N \times V}\sum_{i=1}^{N}\log p(x_i|x)) \tag{3}$$

where V is the vocabulary size.

Pre-training data

We train all models on UniRef90 [82], a comprehensive protein sequence database that contains approximately 138 million sequences from diverse life forms. This large-scale database serves as a robust training set for capturing the underlying patterns and intricacies associated with protein sequences. For model validation, a subset of 100,000 sequences is reserved from the UniRef90 database. The validation set serves multiple purposes: monitoring the training process by observing perplexity during pre-training to ensure correct model behavior, and selecting hyperparameters such as batch size and learning rate via grid search, where we split the validation set into subsets A and B (in an 8:2 ratio) for training and evaluation respectively. Additionally, it is used to compute the metric of normalized perplexity to assess the effectiveness of pre-training results. This setup ensures that the trained models are subjected to a variety of sequence patterns, thereby facilitating a more robust understanding of protein sequences. By reserving a significant number of sequences for validation, we also ensure an unbiased assessment of the model performance.

Classification head

To test the potential bias caused by different classification methods, the *Mean Pooling* and *Attention1d Pooling* are adopted under our evaluation, as shown in Fig 2 (b). The former is trained on the average of features aligned with the first dimension, using MLP and ReLU activation to make a prediction. The latter is trained on an attention mechanism and a 1D convolution layer to map different weights to embed and predict the label.

Model pipeline

Protein-wise tasks Leaning a function $y = f_\theta(x)$ that maps protein x to a label y, where f_θ is parameterized by a sequence-based encoder and a classification head defined upon the residue-wise or residue-pair protein embedding.

Protein-pair tasks Leaning a function $y = f_\theta(x)$ that maps a pair of proteins (x_1, x_2) to a label y, where f_θ is parameterized by a pair of siamese sequence-based encoders and a classifier defined upon the sum of the embeddings of two proteins.

Benchmark tasks

The PETA benchmark includes 15 tasks within 5 groups and 33 datasets in total, mainly focusing on protein-wise and protein-pair tasks. We have curated tasks from influential protein engineering applications and made updates to certain datasets to ensure their relevance and accuracy, a summary of the downstream dataset statistics is shown in Table 1.

Fitness prediction

This set of tasks aims to forecast functional attributes of proteins, which may be either discrete or continuous in nature.

GB1 fitness prediction assesses fitness scores among mutations within the GB1 landscape from [86]. Given a protein sequence x, we map it to a regression value $y \in \mathrm{R}$, where a fitness score of 1 represents the wild-type and 0 indicates non-binding affinity. In our analysis, we utilize all the dataset splits proposed in FLIP [70], encompassing "one-vs-rest", "two-vs-rest", "three-vs-rest", "low-vs-high" and "Sampled". For example, "one-vs-rest" indicates the wild type and single mutants are assigned to train and validation, while the rest are assigned to test. More details about dataset splits can be found in supplementary materials Section.

Impact: GB1 serves as the binding domain of protein G [87], an immunoglobulin binding protein found in Streptococcal bacteria [88]. This task stands as a gold standard for investigating epistatic interactions.

AAV fitness prediction entails the evaluation of fitness values associated with Adeno-associated virus (AAV) capsid proteins [89]. Given a protein sequence x, we establish a mapping to a regression value $y \in \mathrm{R}$, focusing on the mutational window spanning positions 561 to 588 from [90]. We adopt all the dataset splits from FLIP [70], which includes "Mut-Des", "Des-Mut", "one-vs-rest", "two-vs-rest", "seven-vs-rest", "low-vs-high"

Table 1 Benchmark task details. Each task, along with its task name, category, the count of datasets or splits, the source of the dataset, and evaluation metric are shown below

Task name	Category	Count	Source	Metric
Fitness Prediction				
GB1 fitness (GB1)	Reg.	5	FLIP [70]	Spearman's ρ
AAV fitness (AAV)	Reg.	7	FLIP [70]	Spearman's ρ
Thermostability (Thermo)	Reg.	3	FLIP [70]	Spearman's ρ
Fluorescence (Flu)	Reg.	1	TAPE [71]	Spearman's ρ
Stability (Sta)	Reg.	1	TAPE [71]	Spearman's ρ
Protein-Protein Interaction Prediction				
Yeast PPI (Yeast)	Cls.	1	PETA	Accuracy
Human PPI (Human)	Cls.	1	PETA	Accuracy
SHS PPI (SHS27k)	Cls.	1	PETA	Accuracy
Localization Prediction				
Subcellular localization (Sub)	Cls.	3	pro-loc [83], DeepLoc-2 [75]	Accuracy
Binary localization (BinLoc)	Cls.	1	pro-loc [83]	Accuracy
Sorting signal (Sig)	Cls.	1	DeepLoc-2 [75]	Accuracy
Solubility Prediction				
Binary solubility (BinSol)	Cls.	1	DeepSol [84]	Accuracy
E.coli solubility (Esol)	Reg.	1	GraphSol [85]	MSE
Mutation solubility (Solmut)	Reg.	3	PETA	Spearman's ρ
Fold Prediction				
Fold Prediction (Fold)	Cls.	3	TAPE [71]	Accuracy

Reg.: regression; Cls.: classification; MSE: mean square error; Spearman's ρ: Spearman Correlation

and "Sampled". More details about dataset splits can be found in supplementary materials Section 2.1.2

Impact: AAV proteins are responsible for facilitating the integration of a DNA payload into a target cell by the virus [91]. This task specifically addresses the prediction of fitness for an extended sequence subjected to mutations at select positions.

Thermostability prediction involves the analysis of protein melting curves, which are acquired through a mass spectrometry-based assay and meticulously sourced from [92]. In this endeavor, we focus on a protein sequence x, which is drawn from a vast pool of 48,000 proteins spanning 13 diverse species. Our objective is to predict a thermostability score $y \in \mathrm{R}$. For this analysis, we have employed the dataset splitting strategies "Human", "mixed_split", and "Human_cell" as outlined in FLIP [70]. More details about dataset splits can be found in the supplementary materials Section 2.1.3.

Impact: Thermostable proteins [93, 94] demonstrate an ability to endure higher temperature conditions for extended periods or function at an accelerated rate. This task aligns closely with applications in protein engineering, particularly within industrial settings, where the enhanced stability of proteins can yield substantial benefits.

Fluorescence prediction primarily focuses on forecasting the fitness of mutants of the green fluorescent protein (GFP) [95], as documented by [96]. In this context, we are presented with a GFP mutant sequence x and aim to predict the corresponding fluorescence intensity $y \in \mathrm{R}$. We leverage the dataset and split methodology derived from TAPE [71], which involves training the model on lower-order mutants and subsequently evaluating it on higher-order mutants.

Impact: Green fluorescent protein can mark particular proteins in an organic structure by its green fluorescence [97], this makes it easier for researchers to observe. This task bears significance in uncovering mutational patterns that either enhance or diminish such vital biological properties.

Stability prediction endeavors to assess the resilience of proteins within their natural environment. It involves taking a protein sequence, denoted as x, and predicting its corresponding experimental stability score, denoted as $y \in \mathrm{R}$. In this pursuit, we leverage the dataset curated from [98] and employ the partitioning method introduced in TAPE [71]. The training dataset comprises proteins sourced from four rounds of experimental design, while the test dataset encompasses proteins that are Hamming distance-1 neighbors of the top candidate proteins.

Impact: The design of stable proteins in the face of mutations plays a pivotal role in the field of protein engineering [99]. This work is instrumental in various applications, such as ensuring the effective delivery of drugs before they degrade.

Protein-protein interaction prediction

Predicting protein-protein interactions (PPI) is pivotal for deciphering the intricate molecular networks underpinning cellular functions and disease mechanisms, guiding targeted therapeutic interventions.

Yeast PPI prediction involves the prediction of whether two yeast proteins interact with each other. When presented with two proteins, denoted as x_1 and x_2 from yeast, the classifier assigns a binary label $y \in 0, 1$ to signify the presence or absence of interaction between them. To accomplish this task, we utilize the yeast PPI dataset sourced from [100]. In this dataset, half of the instances represent positive cases, selected from the DIP_20070219 database of interacting proteins [101], with stringent criteria that exclude proteins with fewer than 50 amino acids or exhibiting ≥40% sequence identity on the full dataset. The negative cases are generated by randomly pairing proteins that lack evidence of interaction, and these pairs are further filtered based on their sub-cellular locations.

Impact: The yeast dataset serves as a widely recognized benchmark [102, 103] for assessing model performance, and yeast PPI prediction substantially enhances our comprehension of cellular processes by unveiling intricate protein interactions and providing crucial insights into the functional roles of proteins within yeast cells.

Human PPI prediction task predicts whether two human proteins interact or not. When provided with protein sequences x_1 and x_2 from humans, the predictor generates a binary label $y \in 0, 1$ to indicate the presence or absence of interaction between them. We adopt the dataset from [104], comprising positive protein pairs obtained from the Human Protein Reference Database (HPRD) [105] and negative protein pairs sourced from different cellular compartments with no documented interaction [106]. To ensure data quality, self-interactions and duplicate interactions were removed, resulting in the creation of two datasets, namely "AB" and "CD." The "AB" dataset encompasses the entire dataset, while the "CD" dataset comprises selected proteins with identities below 25%. For evaluation, we exclusively employ the "AB" split strategy in this task.

Impact: Human PPI prediction holds immense practical significance in clinical research. Notably, insights into protein interactions linked to diseases enhance our understanding of human disease mechanisms, paving the way for innovative therapeutic strategies [107, 108].

SHS PPI prediction is to classify the type of interaction between a given protein pair. Given two protein sequences, x_1 and x_2, the model aims to predict a label y where $y \in \{0, 1..., 6\}$. These interaction types encompass categories such as "reaction", "activation", and "catalysis", among others. Our analysis utilizes a dataset derived from interaction pairs specific to Homo sapiens, sourced from the STRING database [76]. We adopt preprocess strategies as recommended by [109] where the suboptimal health status (SHS) dataset is divided into two subsets: "SHS27k" and "SHS148k". For computational efficiency, our study focuses solely on the "SHS27k" subset. The data is partitioned into training, validation, and test sets at a random split ratio of 8:1:1.

Impact: Understanding and categorizing the precise interactions between protein pairs is pivotal in unraveling intricate cellular mechanisms and shedding light on complex biological pathways. This knowledge not only aids in defining drug efficacy through network-based "drug-disease proximity measures" [110] but also plays a crucial role in interpreting the outcomes of genome-wide association screens [111].

Localization prediction

Identifying the localization or local-related biological mechanism of proteins within various cellular compartments is of paramount importance in the process of functional annotation.

Subcellular localization prediction aims to dig out the specific cellular location of a given natural protein. Given a protein sequence denoted as x, the model assigns it to multiple possible localizations $y \in 0, 1, ..., 9$, which may include designations such as ”Nucleus" and "Cytoplasm", among others. To accomplish this task, we utilize two datasets from DeepLoc-1 [74] and DeepLoc-2 [75]. For the DeepLoc-1 dataset, we apply the split methodology introduced by [83]. Regarding the DeepLoc-2 dataset, its original split strategy involves 5-fold cross-validation from SwissProt. In our implementation, we employ the first three partitions as training data, the fourth as validation data, and ultimately evaluate the model's performance on the last partition and the independent test dataset of human protein atlas (HPA) [112].

Impact: The subcellular localization of proteins plays a crucial role in deciphering the fundamental mechanisms of diseases linked to abnormal subcellular localization [113, 114]. Notably, some proteins are recognized for their ability to localize within multiple cellular compartments, underscoring the intricate and pertinent nature of this research domain.

Binary localization prediction constitutes a sub-problem of the aforementioned task. The model's responsibility is to decide whether a given protein x should be categorized as "membrane-bound" or "soluble," denoted as $y \in 0, 1$. The datasets for training and testing are drawn from DeepLoc-1 [74], which includes an additional label system where "S" represents soluble, "M" corresponds to membrane-bound, and "U" signifies unknown localization. To train the model, we employ the same data partitioning method as introduced by [83], while excluding data points labeled as "U".

Impact: Predicting protein localization as either membrane-bound or soluble is vital for deciphering cellular functions, particularly in signal transduction and transport [115]. It plays a pivotal role in drug discovery, enabling the design of targeted therapies against membrane proteins.

Sorting signal prediction elucidates the intricate process of subcellular localization by identifying biological mechanisms within sorting signal sequences that guide proteins to specific subcellular structures or organelles. When presented with a short sequence x, the model assigns it to one of nine classes denoted as $y \in 0, 1, ..., 8$, encompassing designations such as "Signal Peptide (SP)" and "Mitochondrial Transit Peptide (MT)", among others. This constitutes a multi-label classification task, and we employ the dataset sourced from DeepLoc-2 [75]. As the original dataset lacks an official split strategy, we perform a random split with a train/validation/test ratio of 8:1:1.

Impact: Protein sorting signals facilitate the precise intracellular localization of proteins, thereby sustaining cellular homeostasis and the integrity of subcellular compartments [116, 117]. These signals typically entail interactions with partner proteins or sorting complexes. It is significant to investigate protein sorting signals for comprehending the intracellular localization and functional intricacies of proteins.

Solubility prediction

This group of tasks is to predict the protein solubility, which is critical for optimizing protein expression, purification, and drug development processes.

Binary solubility prediction aims to determine whether a protein is soluble or insoluble. When presented with a protein denoted as x, the model assigns it to a binary label, $y \in 0, 1$. The dataset and data partitioning approach is drawn from DeepSol [84], where protein sequences exhibiting a sequence identity of $\geq 30\%$ to any sequence in the test set are excluded from the training set. This task shares similarities with binary localization prediction but explicitly focuses on modeling solubility.

Impact: Protein solubility is pivotal for swiftly and efficiently selecting appropriate protein samples, saving resources and time, particularly in biotechnology, drug development, and laboratory protein purification [118, 119]. It improves experiment success rates and resource allocation while advancing scientific research.

E.coli solubility prediction involves forecasting the solubility value of E. coli proteins using an ensemble database, downloadable from the eSOL website [78]. When provided with a sequence from E. coli, the model predicts a regression value, denoted as $y \in \mathrm{R}$. Solubility, in this context, is defined as the ratio of the supernatant fraction to the total fraction, as determined in physiochemical experiments referred to as PURE [120]. We utilize the training and validation datasets sourced from GraphSol [85] and further partition the validation dataset into separate validation and test sets.

Impact: E. coli, as a prevalent host organism for protein expression, demands precise solubility predictions to optimize recombinant protein production, purification, and subsequent functional studies [121]. Such predictions, based on experimental data and computational models, facilitate the selection of suitable protein candidates for diverse applications [122], ranging from structural biology to drug discovery and industrial processes.

Mutation solubility prediction measures the impact of mutations on protein solubility. Given a mutated protein sequence denoted as x, the model predicts the solubility change $y \in \mathrm{R}$ relative to the wild-type sequence. This task encompasses three distinct protein mutation datasets, with mutations occurring at single points within proteins such as "Beta-lactamase TEM (blat)", "Chalcone Synthase (cs)" and "Levoglucosan Kinase (lgk)". These datasets were sourced from SoluProtMutDB [67] which provides manually curated and reliable data in the standardized format. Data points where recorded mutations did not align with the original sequence were excluded, and the training, validation, and test datasets were partitioned in an 8:1:1 ratio.

Impact: Low protein solubility is a significant hurdle in industrial processes and is implicated in numerous human diseases [123]. Investigating the impact of mutations on protein solubility not only sheds light on the mechanisms underpinning disease development but also enhances the application of protein engineering in various industrial domains.

Fold prediction

While AlphaFold [124] and RoseTTAFold [125] have made significant strides in structure prediction, the fold prediction task still remains a rigorous assessment to evaluate the representation quality of the sequence model.

Fold prediction is the automated classification of protein sequences into one of 1,195 known protein folds, facilitating the modeling of the sequence-structure relationship. Given any sequence x, the objective is to predict the fold label $y \in 0, 1, ..., 1194$, determined by the backbone coordinates of the corresponding protein structure. This task utilizes the dataset from [126], originally derived from the SCOP 1.75 database [127]. Notably, this dataset meticulously addresses homologous sequence redundancy between test and training datasets through two distinct strategies: a three-level redundancy reduction at fold/superfamily/family levels and sequence identity reduction.

Impact: Fold prediction is essential for unraveling the intricate relationship between a protein's primary sequence and its three-dimensional structure, with profound implications for fields ranging from structural biology to drug design [128].

Experiments

Experimental setups

We perform the pre-training of our models on 8 A100-80GB GPUs, using a data-parallel distribution strategy. The global batch size is set to 1024 (local batch size is set to 32), and the maximum sequence length is constrained to 1024 tokens. We employ dropout regularization with a rate of 0.1 during the pre-training phase to mitigate overfitting. The architecture comprises 12 encoder layers, with each layer having a hidden size of 768 and an intermediate size of 3072. The multi-head attention mechanism contains 12 heads, each with a dimensionality of 64. For model optimization, we utilize the Adam optimizer, with a learning rate initialized at 1e-4. The maximum number of training steps is set to 530,000. The learning rate schedule involves a warm-up mechanism for the first 2000 iterations, following which the learning rate is linearly decayed to zero. The Adam hyperparameters are configured as follows: epsilon is 1e-8, $\beta_1 = 0.9$ and $\beta_2 = 0.98$. Gradient clipping is applied with a maximum value of 5.0 to prevent exploding gradients. Our implementation leverages the PyTorch framework in conjunction with the Hugging Face library, aligning with best practices for efficient and scalable training of language models.

In the case of supervised tasks, all pre-trained model weights are kept fixed to ensure a fair evaluation of their representation capabilities. Classifiers are trained using a batch size of 256, a learning rate of 0.001, and the Adam optimizer. Early stopping is employed with a patience threshold of 20 epochs, with a maximum of 100 epochs for training. It is important to note that these hyperparameters were adopted without adjustments, drawing reference from [70]. Each individual experiment underwent training three times using different random seeds, and the final results represent the average scores obtained.

Pre-training results

Following the pre-training phase, all models achieved a reduction in loss to an acceptable level, demonstrating effective learning from the training data. Table 2 and Table 3 present the perplexity and normalized perplexity metrics calculated on the test set for both Byte Pair Encoding (BPE) and Unigram models, respectively. For the base model employing a per-amino-acid (Per-AA) tokenization strategy, the *Perplexity* value is 7.78, and the corresponding *Normalized Perplexity* is 1.06.

In the supplementary materials, Figures S1 to S30 show the loss curves, as well as the perplexity and normalized perplexity curves for the pre-trained models. It is important to note that evaluations were performed at intervals of 10,000 steps. These figures collectively demonstrate that all models have converged to a reasonable range, substantiating their effectiveness in learning the underlying data distribution.

Table 2 Perplexity and Normalized Perplexity on the validation set for the BPE model

Tokenization	**BPE**						
Vocabulary size	50	100	200	400	800	1600	3200
Perplexity	9.51	13.66	23.67	34.87	49.64	72.39	105.01
Normalized Perplexity	1.05	1.03	1.02	1.01	1.00	1.00	1.00

Table 3 Perplexity and Normalized Perplexity on the validation set for the Unigram model

Tokenization	**Unigram**						
Vocabulary size	50	100	200	400	800	1600	3200
Perplexity	8.26	12.95	26.98	62.39	81.11	115.90	220.77
Normalized Perplexity	1.04	1.03	1.02	1.01	1.01	1.00	1.00

Benchmark results overview

To provide researchers with insights into how the augmentation of vocabulary size in PLMs affects embedding quality, we conducted a systematic evaluation. The scores in Table 4 represent a given vocabulary size, how many datasets or splits, on average performance, surpassing the traditional Per-AA method (with a vocabulary size of 20). For example, We have 33 datasets or splits in total, and 22 of them outperform the baseline method using a vocabulary size of 20 when using a vocabulary size of 50. This counting method is obtained by comparing the average scores on each dataset, for the extended vocabulary models. The average score on each dataset was computed from the mean of 12 experiment results (2 tokenization methods x 2 classification heads x 3 random seeds). For the baseline models, the average score on each dataset was derived from the mean of 6 experiments (2 classification heads x 3 random seeds). More detailed results of each experimental setting can be found in the supplementary materials Table S1 to S54.

Our experimental findings have led to several key insights:

- Significant Impact of Vocabulary Size. Extensive experimentation has unequivocally demonstrated that vocabulary size profoundly influences protein representation, albeit with varying degrees of impact across different types of downstream tasks. Notably, in every dataset associated with fold prediction, an inverse relationship was observed wherein enhancements in vocabulary size correlated with negative optimization.
- Existence of an Optimal Vocabulary Threshold. Contrasting with language models utilized in NLP, PLMs with an excessively large vocabulary size can potentially exert detrimental effects on downstream tasks. Specifically, when the vocabulary size surpasses 800, the majority of tasks are performed suboptimally compared to the baseline model that employs per-AA segmentation.

Downstream tasks

Fitness Prediction. Table 5 showcases results for five distinct tasks under the Fitness Prediction and the evaluation metric is *Spearman correlation*. It is worth highlighting that the datasets for GB1, AAV, and Thermo have different splits and the details are shown in Fig 3. A key observation from the data is the mixed effects of adding more words to the vocabulary. Compared to the per-AA tokenization, the performance of most splits is improved with an expanded vocabulary. However, AAV was an exception, experiencing a significant drop in performance as the vocabulary grew, with decreases ranging from 3% to 7.4%. In contrast, the performance of

Table 4 The number of datasets or splits whose average score exceeds the baseline model of 20 vocabulary size

Vocabulary	Sum (33)	Fit (17)	PPI (3)	Loc (5)	Sol (5)	Fold (3)
50	22	10	3	5	4	0
100	19	8	2	5	4	0
200	20	9	3	4	4	0
800	16	5	3	4	4	0
1,600	15	5	2	4	4	0
3,200	15	5	2	4	4	0

Fit: protein fitness; PPI: protein-protein interaction; Loc: protein localization; Sol: protein solubility; Fold: protein fold

Table 5 Performance on Fitness Prediction and Localization Prediction

Vocabulary	Fitness Prediction					Localization Prediction				
	GB1(5) ↑	AAV(7) ↑	Thermo(3) ↑	Flu(1) ↑	Sta(1) ↑	Sub-1(1) ↑	BinLoc(1) ↑	Sub-2(1) ↑	Sub-hpa(1) ↑	Sig(1) ↑
20	48.6$_{(0.8)}$	**34.8**$_{(0.3)}$	61.3$_{(0.2)}$	39.3$_{(1.0)}$	**51.3**$_{(0.3)}$	**94.6**$_{(0.1)}$	91.3$_{(0.3)}$	92.2$_{(0.1)}$	89.0$_{(0.1)}$	95.7$_{(0.1)}$
50	**50.3**$_{(0.9)}$	**31.8**$_{(0.8)}$	**61.8**$_{(0.2)}$	39.9$_{(2.3)}$	**53.0**$_{(0.5)}$	**94.8**$_{(0.0)}$	**91.7**$_{(0.3)}$	**92.6**$_{(0.0)}$	**89.1**$_{(0.2)}$	**96.1**$_{(0.0)}$
100	**53.4**$_{(0.4)}$	**31.3**$_{(0.2)}$	**62.1**$_{(0.1)}$	38.3$_{(2.0)}$	**51.6**$_{(0.7)}$	**94.7**$_{(0.1)}$	**91.3**$_{(0.1)}$	**92.3**$_{(0.0)}$	89.0$_{(0.0)}$	**95.9**$_{(0.2)}$
200	**50.4**$_{(0.1)}$	30.6$_{(1.0)}$	**61.6**$_{(0.1)}$	**42.0**$_{(1.9)}$	49.5$_{(0.5)}$	94.5$_{(0.1)}$	**91.3**$_{(0.2)}$	**92.2**$_{(0.0)}$	**89.4**$_{(0.0)}$	**95.9**$_{(0.1)}$
800	46.2$_{(1.2)}$	28.8$_{(0.2)}$	60.6$_{(0.3)}$	41.5$_{(1.7)}$	46.4$_{(0.1)}$	94.4$_{(0.1)}$	90.9$_{(0.3)}$	92.1$_{(0.0)}$	**89.4**$_{(0.1)}$	95.6$_{(0.1)}$
1,600	46.5$_{(0.9)}$	27.4$_{(0.1)}$	61.3$_{(1.0)}$	**43.8**$_{(0.9)}$	43.2$_{(1.6)}$	94.4$_{(0.1)}$	90.5$_{(0.2)}$	92.1$_{(0.0)}$	**89.2**$_{(0.1)}$	95.6$_{(0.1)}$
3,200	46.9$_{(1.1)}$	29.9$_{(0.9)}$	60.8$_{(0.1)}$	**43.0**$_{(1.9)}$	44.5$_{(1.0)}$	94.3$_{(0.2)}$	90.9$_{(0.1)}$	92.1$_{(0.0)}$	89.0$_{(0.1)}$	95.6$_{(0.2)}$

Each value indicates the *mean*$_{(std)}$ score across all experiments within the same vocabulary size. The values colored with 🟨 are higher than the Per-AA method. Datasets marked with (*) indicate the number of dataset splits. For instance, **GB1** encompasses five different dataset splits within the same dataset. The score with a vocabulary size of 50 reflects results across 60 experiments (5×2×2×3, representing the number of dataset splits, tokenization methods, classification heads, number of random seed experiments)

† The top three are highlighted by **First**, **Second**, Third.

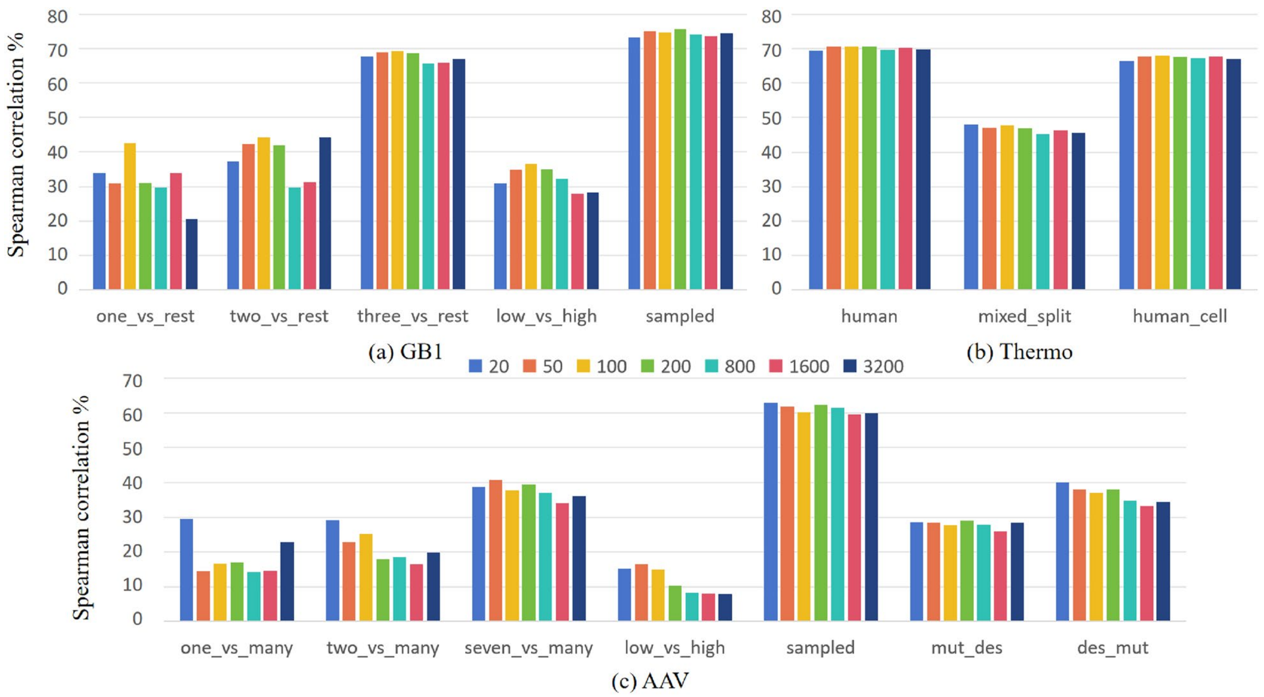


Fig. 3 Detail performances of the GB1, Thermo, and AAV datasets across different vocabulary sizes

Flu benefited from a larger vocabulary. Interestingly, the average performance for GB1 and Stab began to decline after the vocabulary size reached 200, even falling below the baseline set at a vocabulary size of 20. Thermo is not sensitive to vocabulary size changes, fluctuating approximately 0.5% up or down.

Localization Prediction. The right side of Table 5 presents results from five datasets under the category of Localization Prediction, and the monitored metric is *Accuracy*. Across all datasets and partitioning methodologies, the task of subcellular localization prediction consistently achieves a remarkable classification accuracy exceeding 90. This accuracy remains quite stable, with any changes in performance staying within 1% despite differences in vocabulary size. Drawing from these experimental insights, it is evident that language models handle both multi-class and single-class prediction tasks for protein localization relatively easy. Moreover, the vocabulary size seems to have minimal influence on the prediction outcomes for protein localization tasks.

Protein-Protein interaction Prediction. Table 6 summaries the 3 datasets from PPI Prediction, and the metric is *Accuracy*. The table clearly shows that using more specific words helps in identifying the relationships between protein pairs. Additionally, datasets that are harder to classify show higher performance enhancements. For instance, in the **Yeast** dataset, the model with a vocabulary size of 1, 600 exhibited a 3.4% average score increase compared to the model with a vocabulary size of 20. In SHS27k, every increase in vocabulary size resulted in performance improvements, with the biggest improvement being of 2.6%. In contrast, while the Human dataset

Table 6 Performance on PPI Prediction, Solubility Prediction and Fold Prediction

Vocabulary	PPI Prediction			Solubility Prediction			Fold Prediction		
	Yeast(1) ↑	SHS27k(1) ↑	Human(1) ↑	Esol(1) ↓	BinSol(1) ↑	Solmut(3) ↑	superfamily(1) ↑	family(1) ↑	fold(1) ↑
20	69.9$_{(0.7)}$	51.9$_{(0.8)}$	92.7$_{(0.2)}$	**0.044**$_{(0.008)}$	67.0$_{(0.5)}$	18.7$_{(0.7)}$	**51.4**$_{(0.4)}$	**93.0**$_{(0.5)}$	**26.7**$_{(0.4)}$
50	**72.5**$_{(0.7)}$	52.6$_{(0.2)}$	**93.0**$_{(0.2)}$	**0.045**$_{(0.002)}$	69.1$_{(0.1)}$	**21.7**$_{(0.7)}$	**50.5**$_{(0.1)}$	**92.9**$_{(0.2)}$	**26.8**$_{(0.6)}$
100	72.0$_{(0.9)}$	52.0$_{(0.3)}$	92.5$_{(0.2)}$	0.048$_{(0.000)}$	69.6$_{(0.3)}$	20.8$_{(0.3)}$	43.7$_{(0.7)}$	**91.9**$_{(0.3)}$	22.6$_{(0.6)}$
200	**72.8**$_{(0.5)}$	**53.5**$_{(0.3)}$	**92.9**$_{(0.1)}$	0.047$_{(0.001)}$	70.2$_{(0.1)}$	20.1$_{(1.4)}$	**43.9**$_{(0.4)}$	91.5$_{(0.2)}$	**26.4**$_{(1.9)}$
800	72.0$_{(0.6)}$	**53.4**$_{(0.3)}$	**92.8**$_{(0.2)}$	**0.046**$_{(0.001)}$	**70.2**$_{(0.7)}$	**21.0**$_{(0.7)}$	43.5$_{(0.5)}$	89.4$_{(0.5)}$	24.0$_{(0.2)}$
1,600	**73.3**$_{(0.6)}$	**53.1**$_{(0.7)}$	92.5$_{(0.6)}$	0.048$_{(0.001)}$	**70.4**$_{(1.3)}$	**21.3**$_{(1.3)}$	41.8$_{(0.5)}$	89.0$_{(0.4)}$	23.1$_{(1.4)}$
3,200	72.4$_{(0.5)}$	52.0$_{(0.2)}$	92.2$_{(0.2)}$	0.047$_{(0.000)}$	69.9$_{(0.9)}$	19.7$_{(1.1)}$	40.6$_{(0.2)}$	88.0$_{(0.4)}$	23.3$_{(0.4)}$

Each value indicates the *mean$_{(std)}$* score across all experiments within the same vocabulary size. The values colored with ▨ are higher than the Per-AA method. Datasets marked with (*) indicate the number of dataset splits

† The top three are highlighted by **First**, **Second**, Third.

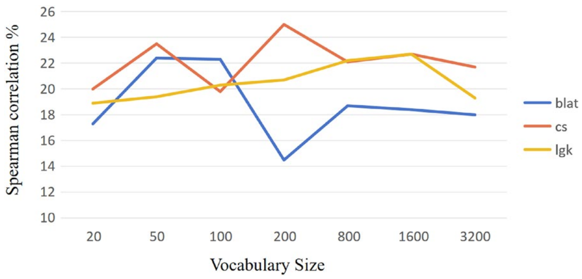


Fig. 4 A detailed exposition is provided on the performance results of three distinct protein solubility mutation datasets: Beta-lactamase TEM (blat), Chalcone Synthase (cs), and Levoglucosan Kinase (lgk) across varying vocabulary sizes

showed improvements across the board, the maximum increase was a mere 0.2%.

Solubility Prediction. The middle section of Table 6 displays the findings for Solubility Prediction. While the evaluation metric for the three datasets in Solmut is the *Spearman correlation*, **Esol** employs *MSE*, BinSol and DeepSol use *Accuracy* as their metric. Predicting solubility regression values for Esol proves to be relatively straightforward, the lower MSE scores indicate better performance, and this remains consistent across models with different vocabulary sizes. BinSol observed the opposite situation to Esol, the average score increased by 2% to 3%, which is relatively significant. An analysis of the Solmut datasets indicates that models with expanded vocabulary sizes have the potential to improve performance by 1% to 5% as shown in Fig. 4. Although most instances show enhancement, occasional instability is detected. Such variability could be attributed to the alterations in the inherent characteristics of proteins post-mutation.

Fold Prediction. The results for Fold Prediction are detailed on the right side of Table 6, and the metrics is *Accuracy*. In fold prediction, an important pattern is observed: as the vocabulary size enlarges, there is a clear drop in performance across all datasets. To illustrate, the superfamily split sees the most significant drop, falling by 1.3% to 11.2%, while the family and fold splits experience declines of 5% and 3.8% respectively. This trend highlights a consistent drop in performance as the larger vocabulary size is. This decline could be due to the combination of

Table 7 The average results of different downstream task groups under the same vocabulary with varying tokenization methods are presented

Vocab.	BPE					Unigram				
	Fit	PPI	Loc	Sol	Fold	Fit	PPI	Loc	Sol	Fold
20	**47.1**$_{(0.2)}$	71.5$_{(0.5)}$	**93.2**$_{(0.1)}$	42.9$_{(0.2)}$	**57.2**$_{(0.1)}$	**47.1**$_{(0.1)}$	71.5$_{(0.5)}$	**93.2**$_{(0.1)}$	42.9$_{(0.2)}$	**57.2**$_{(0.2)}$
50	**47.1**$_{(0.1)}$	72.2$_{(0.2)}$	**93.4**$_{(0.2)}$	**45.6**$_{(0.3)}$	**55.8**$_{(0.1)}$	**47.7**$_{(0.3)}$	**73.1**$_{(0.5)}$	**93.3**$_{(0.1)}$	45.3$_{(1.0)}$	**57.6**$_{(0.3)}$
100	**47.1**$_{(0.2)}$	72.1$_{(0.3)}$	**93.2**$_{(0.1)}$	44.6$_{(0.2)}$	53.4$_{(0.6)}$	**47.6**$_{(0.2)}$	72.2$_{(0.4)}$	93.0$_{(0.1)}$	**45.8**$_{(0.8)}$	52.1$_{(0.2)}$
200	**47.4**$_{(0.5)}$	**73.1**$_{(0.1)}$	**93.2**$_{(0.2)}$	**45.4**$_{(1.1)}$	**54.9**$_{(0.8)}$	46.2$_{(0.3)}$	**73.1**$_{(0.5)}$	**93.1**$_{(0.1)}$	44.9$_{(0.4)}$	**53.0**$_{(0.4)}$
800	45.4$_{(0.5)}$	**72.8**$_{(0.5)}$	93.0$_{(0.2)}$	**45.6**$_{(0.3)}$	52.3$_{(0.2)}$	44.1$_{(0.1)}$	**72.6**$_{(0.3)}$	92.8$_{(0.2)}$	**45.6**$_{(0.6)}$	52.3$_{(0.1)}$
1,600	45.2$_{(0.7)}$	**73.2**$_{(0.8)}$	92.9$_{(0.2)}$	**45.7**$_{(1.0)}$	51.5$_{(0.1)}$	43.7$_{(0.4)}$	**72.7**$_{(0.4)}$	92.7$_{(0.1)}$	**45.9**$_{(0.3)}$	51.0$_{(0.3)}$
3,200	45.2$_{(0.3)}$	72.3$_{(0.2)}$	92.8$_{(0.1)}$	44.3$_{(0.2)}$	50.9$_{(0.2)}$	44.8$_{(0.2)}$	72.2$_{(0.4)}$	92.9$_{(0.2)}$	45.3$_{(1.2)}$	50.4$_{(0.1)}$

Each score represents the average score of all experiments within that task group, encompassing different tasks, datasets, classification heads, and random seeds. The values colored with 🟨 are higher than the Per-AA method. *Abbreviations,* **Vocab.**: vocabulary size; **Fit**: protein fitness; **PPI**: protein-protein interaction; **Loc**: protein localization; **Sol**: protein solubility; **Fold**: protein fold

† The top three are highlighted by **First**, **Second**, Third.

Table 8 The average results of different downstream task groups under the same vocabulary with varying pooling heads are presented

Vocab.	Mean Pooling					Attention1d Pooling				
	Fit	PPI	Loc	Sol	Fold	Fit	PPI	Loc	Sol	Fold
20	**41.5**$_{(0.2)}$	70.1$_{(0.7)}$	**92.8**$_{(0.2)}$	40.7$_{(1.2)}$	**56.8**$_{(0.2)}$	52.2$_{(0.3)}$	72.8$_{(0.7)}$	93.4$_{(0.1)}$	45.0$_{(1.5)}$	**57.2**$_{(0.2)}$
50	**41.1**$_{(0.2)}$	**70.6**$_{(0.3)}$	**93.0**$_{(0.2)}$	41.8$_{(0.1)}$	**56.2**$_{(0.2)}$	**53.7**$_{(0.5)}$	74.8$_{(0.4)}$	**93.7**$_{(0.1)}$	**49.0**$_{(0.6)}$	**57.2**$_{(0.5)}$
100	**41.5**$_{(0.2)}$	**70.1**$_{(0.3)}$	92.4$_{(0.0)}$	42.0$_{(0.1)}$	52.6$_{(0.4)}$	**53.1**$_{(0.2)}$	74.2$_{(0.1)}$	**93.8**$_{(0.2)}$	**48.4**$_{(0.7)}$	52.9$_{(0.2)}$
200	**41.2**$_{(0.3)}$	**70.8**$_{(0.3)}$	**92.5**$_{(0.1)}$	**42.7**$_{(0.7)}$	**54.3**$_{(1.0)}$	**52.4**$_{(0.3)}$	75.3$_{(0.3)}$	**93.7**$_{(0.1)}$	47.6$_{(0.9)}$	**53.6**$_{(0.4)}$
800	39.3$_{(0.4)}$	70.0$_{(0.2)}$	92.2$_{(0.2)}$	**42.9**$_{(0.7)}$	52.0$_{(0.4)}$	50.1$_{(0.3)}$	**75.5**$_{(0.1)}$	**93.6**$_{(0.1)}$	48.3$_{(1.5)}$	52.6$_{(0.4)}$
1,600	39.2$_{(0.3)}$	69.7$_{(0.2)}$	92.0$_{(0.1)}$	**43.2**$_{(0.5)}$	51.4$_{(0.4)}$	49.7$_{(0.6)}$	**76.1**$_{(0.6)}$	93.6$_{(0.2)}$	**48.4**$_{(0.8)}$	51.2$_{(0.5)}$
3,200	39.8$_{(0.1)}$	68.7$_{(0.6)}$	92.1$_{(0.1)}$	41.5$_{(0.3)}$	50.6$_{(0.1)}$	50.2$_{(0.1)}$	**75.7**$_{(0.2)}$	**93.6**$_{(0.1)}$	48.1$_{(0.7)}$	50.7$_{(0.1)}$

Each score represents the average score of all experiments within that task group, encompassing different tasks, datasets, classification heads, and random seeds. The values colored with 🟨 are higher than the Per-AA method. Vocab.: vocabulary size; Fit: protein fitness; PPI: protein-protein interaction; Loc: protein localization; Sol: protein solubility; Fold: protein fold

† The top three are highlighted by **First**, **Second**, Third.

multiple amino acid tokens during the encoding process, which might hide important details of the local structures. Consequently, this increase in vocabulary sizes could be negatively impacting the ability to predict structure-related tasks.

Ablation study

From Tables 7 and 8, we can observe positive optimizations in most datasets and negative optimizations in some tasks due to the expansion of the vocabulary. Importantly, these optimizations, whether positive or negative, are independent of the tokenization method used, the type of classification head, and the random seed. The changes are solely attributed to the variations in vocabulary size. It is worth noting that, we remove Esol from Solubility Prediction due to the incomparable scale of the values.

Analysis of tokenizers. From Table 7, it can be observed that the discrepancies arising from different tokenization methods are minimal across various downstream tasks. The main source of performance variation stems from the impact of vocabulary size on model representation. Across different tasks, as the vocabulary size increases, the model performance exhibits a bell-shaped curve, showing an initial increase followed by a decline.

Impact of pooling heads. From Table 8, it can be observed that, when freezing the pre-trained model parameters and only tuning the pooling head, the performance is highly correlated with the choice of the classification head. When using the same vocabulary size, the attention1d pooling method outperforms the mean pooling method. Additionally, similar to the results in Table 7, as the vocabulary size increases, the model's representational capacity across various downstream tasks tends to decline.

Conclusion

In this paper, we introduce PETA, a vocabulary study optimized for protein language models across a broad range of datasets. To mitigate potential biases arising from different tokenization methods, classification heads, and random seeds, for each fixed vocabulary size, we employed both BPE and Unigram tokenization methods, two classification heads (mean pooling and attention1d pooling), and experiments with three different random seeds on each dataset. Ultimately, we found that expanding the vocabulary size to some extent (50-200) generally enhances performance on downstream tasks. However, once the vocabulary size surpasses 800, the model's representational power exhibits a broad decline across most tasks. We hope that this work and benchmark will influence the future protein language model community and contribute positively to human health, environmental development, and biomedicine.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s13321-024-00884-3.

Supplementary file 1: Figure S1-S2. Loss, perplexity and normalized perplexity curve under per-AA tokenization. Figure S3-S16. Loss, perplexity and normalized perplexity curve under BPE tokenization. Figure S17-S30. Loss, perplexity and normalized perplexity curve under Unigram tokenization. Table S1. GB1 dataset train/val/test split. Table S2-S5. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on GB1. Table S6. AAV dataset train/val/test split. Table S7-S10. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on AAV. Table S11. Meltome dataset train/val/test split. Table S12-S15. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Meltome. Table S16. Fluorescence dataset train/val/test split. Table S17-S18. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Fluorescence. Table S19. Stability dataset train/val/test split. Table S20-S21. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Stability. Table S22. Deeploc-1 dataset train/val/test split. Table S23-S24. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Deeploc-1. Table S25. Deeploc_binary dataset train/val/test split. Table S26-S27. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Deeploc_binary. Table S28. Deeploc-2 dataset train/val/test split. Table S29-S30. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Deeploc-2. Table S31. Deep_signal dataset train/val/test split. Table S32-S33. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Deep_signal. Table S34. PPI dataset train/val/test split. Table S35-S38. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on PPI. Table S39. DeepSol dataset train/val/test split. Table S40-S41. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on DeepSol. Table S42. Esol dataset train/val/test split. Table S43-S44. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Esol. Table S45. Solmut dataset train/val/test split. Table S46-S49. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Solmut. Table S50. Fold dataset train/val/test split. Table S51-S54. Detail results of Unigam/BPE tokenization and mean/attention1d pooling on Fold. Table S55-S62. Detail comparison with ESM2 models.

Acknowledgements

This work was supported by Research Programme of National Engineering Laboratory for Big Data Distribution and Exchange Technologies, Shanghai Municipal Special Fund for Promoting High Quality Development (No. 2021-GYHLW-01007), the National Natural Science Foundation of China (11974239, 12104295), the Innovation Program of Shanghai Municipal Education Commission (2019-01-07-00-02-E00076), Shanghai Jiao Tong University Scientific and Technological Innovation Funds (21X010200843), the Student Innovation Center at Shanghai Jiao Tong University, and Shanghai Artificial Intelligence Laboratory.

Author contributions

Yang Tan: Conceptualization of this study, Methodology, Data curation, Implementation, writing & editing. Mingchen Li: Conceptualization of this study, Methodology, Data curation, Implementation, writing & editing. Pan Tan: Review & editing. Ziyi Zhou: Review & editing. Huiqun Yu: Supervision, review & editing. Guisheng Fan: Supervision, review & editing. Liang Hong: Supervision, review & editing.

Data availability

Data and code released on GitHub. https://github.com/ginnm/ProteinPretraining

Declarations

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details

[1]School of Information Science and Engineering, East China University of Science and Technology, Shanghai 200237, China. [2]Shanghai National Center for Applied Mathematics (SJTU Center), & Institute of Natural Science, Shanghai Jiao Tong University, Shanghai 200240, China. [3]Shanghai Artificial Intelligence Laboratory, Shanghai 200240, China. [4]Chongqing Artificial Intelligence Research Institute of Shanghai Jiao Tong University, Chongqing 200240, China.

Received: 21 March 2024 Accepted: 13 July 2024
Published online: 02 August 2024

References

1. Doudna JA, Charpentier E (2014) The new frontier of genome engineering with crispr-cas9. Science 346(6213):1258096
2. Hsu PD, Lander ES, Zhang F (2014) Development and applications of crispr-cas9 for genome engineering. Cell 157(6):1262–1278
3. Scott DE, Bayly AR, Abell C, Skidmore J (2016) Small molecules, big targets: Drug discovery faces the protein-protein interaction challenge. Nat Rev Drug Dis 15(8):533–550
4. Lee HC (2006) Structure and enzymatic functions of human cd38. Mol Med 12(11):317–323
5. Joo H, Lin Z, Arnold FH (1999) Laboratory evolution of peroxide-mediated cytochrome p450 hydroxylation. Nature 399(6737):670–673
6. Sharan R, Ulitsky I, Shamir R (2007) Network-based prediction of protein function. Mol Syst Biol 3(1):88
7. Feng Y, De Franceschi G, Kahraman A, Soste M, Melnik A, Boersema PJ, De Laureto PP, Nikolaev Y, Oliveira AP, Picotti P (2014) Global analysis of protein structural changes in complex proteomes. Nat Biotechnol 32(10):1036–1044
8. Lesley SA (2001) High-throughput proteomics: protein expression and purification in the postgenomic world. Protein Expression Purif 22(2):159–164
9. Arnold FH (1998) Design by directed evolution. Accounts Chem Res 31(3):125–131
10. Ma B (2015) Novor: real-time peptide de novo sequencing software. J Am Soc Mass Spectr 26(11):1885–1894
11. Lin Z, Akin H, Rao R, Hie B, Zhu Z, Lu W, Smetanin N, Verkuil R, Kabeli O, Shmueli Y (2023) Evolutionary-scale prediction of atomic-level protein structure with a language model. Science 379(6637):1123–1130
12. Meier J, Rao R, Verkuil R, Liu J, Sercu T, Rives A (2021) Language models enable zero-shot prediction of the effects of mutations on protein function. Adv Neural Inf Process Syst 34:29287–29303
13. Rives A, Meier J, Sercu T, Goyal S, Lin Z, Liu J, Guo D, Ott M, Zitnick CL, Ma J (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci 118(15):2016239118
14. U, Consortium (2019) Uniprot: a worldwide hub of protein knowledge. Nucl Acids Res 47(1):506–515
15. Madani A, Krause B, Greene ER, Subramanian S, Mohr BP, Holton JM, Olmos Jr JL, Xiong C, Sun ZZ, Socher R, et al (2023) Large language models generate functional protein sequences across diverse families. Nat Biotechnol, 1–8
16. Nijkamp E, Ruffolo J, Weinstein EN, Naik N, Madani A (2022) Progen2: Exploring the boundaries of protein language models. arXiv preprint arXiv:2206.13517
17. Ferruz N, Schmidt S, Höcker B (2022) Protgpt2 is a deep unsupervised language model for protein design. Nat Commun 13(1):4348
18. Elnaggar A, Heinzinger M, Dallago C, Rehawi G, Wang Y, Jones L, Gibbs T, Feher T, Angerer C, Steinegger M (2021) Prottrans: toward understanding the language of life through self-supervised learning. IEEE Trans Pattern Anal Mach Intell 44(10):7112–7127
19. Elnaggar A, Essam H, Salah-Eldin W, Moustafa W, Elkerdawy M, Rochereau C, Rost B (2023) Ankh: Optimized protein language model unlocks general-purpose modelling. bioRxiv, 2023–01
20. Gage P (1994) A new algorithm for data compression. C Users J 12(2):23–38
21. Bengio Y, Ducharme R, Vincent P (2000) A neural probabilistic language model. Advances in neural information processing systems. 13.
22. Rust P, Pfeiffer J, Vulić I, Ruder S, Gurevych I (2020) How good is your tokenizer? on the monolingual performance of multilingual language models. arXiv preprint arXiv:2012.15613
23. Choo S, Kim W (2023) A study on the evaluation of tokenizer performance in natural language processing. Appl Artif Intell 37(1):2175112
24. Asgari E, McHardy AC, Mofrad MR (2019) Probabilistic variable-length segmentation of protein sequences for discriminative motif discovery (dimotif) and sequence embedding (protvecx). Sci Rep 9(1):3577
25. Ieremie I, Ewing RM, Niranjan M (2024) Protein language models meet reduced amino acid alphabets. Bioinformatics 40(2):061
26. Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L (2009) Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 . Ieee
27. Wang A, Singh A, Michael J, Hill F, Levy O, Bowman SR (2018) Glue: A multi-task benchmark and analysis platform for natural language understanding. arXiv preprint arXiv:1804.07461
28. Mikolov T, Sutskever I, Chen K, Corrado GS, Dean J (2013) Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems **26**
29. Le Q, Mikolov T (2014) Distributed representations of sentences and documents. In: International Conference on Machine Learning, pp. 1188–1196. PMLR
30. Wang J, Yang B, Revote J, Leier A, Marquez-Lago TT, Webb G, Song J, Chou K-C, Lithgow T (2017) Possum: a bioinformatics toolkit for generating numerical sequence feature descriptors based on pssm profiles. Bioinformatics 33(17):2756–2758
31. Mejía-Guerra MK, Buckler ES (2019) A k-mer grammar analysis to uncover maize regulatory architecture. BMC Plant Biol 19(1):1–17
32. Wan F, Zeng J (2016) Deep learning with feature embedding for compound-protein interaction prediction. Biorxiv, 086033
33. Li M, Kang L, Xiong Y, Wang YG, Fan G, Tan P, Hong L (2023) Sesnet: sequence-structure feature-integrated deep learning method for data-efficient protein engineering. J Cheminf 15(1):1–13
34. Luo Y, Jiang G, Yu T, Liu Y, Vo L, Ding H, Su Y, Qian WW, Zhao H, Peng J (2021) Ecnet is an evolutionary context-integrated deep learning framework for protein engineering. Nat Commun 12(1):5743
35. Lan Z, Chen M, Goodman S, Gimpel K, Sharma P, Soricut R (2019) Albert: A lite bert for self-supervised learning of language representations. arXiv preprint arXiv:1909.11942
36. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805
37. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser Ł, Polosukhin I (2017) Attention is all you need. Advances in neural information processing systems **30**
38. Raffel C, Shazeer N, Roberts A, Lee K, Narang S, Matena M, Zhou Y, Li W, Liu PJ (2020) Exploring the limits of transfer learning with a unified text-to-text transformer. J Mach Learning Res 21(1):5485–5551
39. Ye H, Chen Z, Wang D-H, Davison B (2020) Pretrained generalized autoregressive model with adaptive probabilistic label clusters for extreme multi-label text classification. In: International Conference on Machine Learning, pp. 10809–10819. PMLR
40. Chen B, Cheng X, Geng Y-a, Li S, Zeng X, Wang B, Gong J, Liu C, Zeng A, Dong Y, et al (2023) Xtrimopglm: Unified 100b-scale pre-trained transformer for deciphering the language of protein. bioRxiv, 2023–07
41. Du Z, Qian Y, Liu X, Ding M, Qiu J, Yang Z, Tang J (2021) Glm: General language model pretraining with autoregressive blank infilling. arXiv preprint arXiv:2103.10360
42. Lu AX, Zhang H, Ghassemi M, Moses A (2020) Self-supervised contrastive learning of protein representations by mutual information maximization. BioRxiv, 2020–09
43. Alley EC, Khimulya G, Biswas S, AlQuraishi M, Church GM (2019) Unified rational protein engineering with sequence-based deep representation learning. Nat Methods 16(12):1315–1322

44. Notin P, Dias M, Frazer J, Hurtado JM, Gomez AN, Marks D, Gal Y (2022) Tranception: Protein fitness prediction with autoregressive transformers and inference-time retrieval. In: International Conference on Machine Learning, pp. 16990–17017. PMLR
45. Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M (2022) Proteinbert: a universal deep-learning model of protein sequence and function. Bioinformatics 38(8):2102–2110
46. Yang KK, Fusi N, Lu AX (2022) Convolutions are competitive with transformers for protein sequence pretraining. bioRxiv, 2022–05
47. Rao RM, Liu J, Verkuil R, Meier J, Canny J, Abbeel P, Sercu T, Rives A (2021) Msa transformer. In: International Conference on Machine Learning, pp. 8844–8856. PMLR
48. Biswas S, Khimulya G, Alley EC, Esvelt KM, Church GM (2021) Low-n protein engineering with data-efficient deep learning. Nat Methods 18(4):389–396
49. Hsu C, Verkuil R, Liu J, Lin Z, Hie B, Sercu T, Lerer A, Rives A (2022) Learning inverse folding from millions of predicted structures. In: International Conference on Machine Learning, pp. 8946–8970 . PMLR
50. Yang KK, Zanichelli N, Yeh H (2022) Masked inverse folding with sequence transfer for protein representation learning. bioRxiv, 2022–05
51. Jing B, Eismann S, Suriana P, Townshend RJ, Dror R (2020) Learning from protein structure with geometric vector perceptrons. arXiv preprint arXiv:2009.01411
52. Dauparas J, Anishchenko I, Bennett N, Bai H, Ragotte RJ, Milles LF, Wicky BI, Courbet A, Haas RJ, Bethel N (2022) Robust deep learning-based protein sequence design using proteinmpnn. Science 378(6615):49–56
53. Zheng Z, Deng Y, Xue D, Zhou Y, Ye F, Gu Q (2023) Structure-informed language models are protein designers. bioRxiv, 2023–02
54. Zhou B, Zheng L, Wu B, Tan Y, Lv O, Yi K, Fan G, Hong L (2024) Protein engineering with lightweight graph denoising neural networks. J Chem Inf Modeling 64(9):3650–3661
55. Wang Z, Combs SA, Brand R, Calvo MR, Xu P, Price G, Golovach N, Salawu EO, Wise CJ, Ponnapalli SP (2022) Lm-gvp: an extensible sequence and structure informed deep learning framework for protein property prediction. Sci Rep 12(1):6832
56. Tan Y, Zhou B, Zheng L, Fan G, Hong L (2023) Semantical and topological protein encoding toward enhanced bioactivity and thermostability. bioRxiv, 2023–12
57. Tan Y, Li M, Zhou B, Zhong B, Zheng L, Tan P, Zhou Z, Yu H, Fan G, Hong L (2024) Simple, efficient and scalable structure-aware adapter boosts protein language models. arXiv preprint arXiv:2404.14850
58. Li M, Tan Y, Ma X, Zhong B, Yu H, Zhou Z, Ouyang W, Zhou B, Hong L, Tan P (2024) Prosst: Protein language modeling with quantized structure and disentangled attention. bioRxiv, 2024–04
59. Wang A, Pruksachatkun Y, Nangia N, Singh A, Michael J, Hill F, Levy O, Bowman S (2019) Superglue: A stickier benchmark for general-purpose language understanding systems. Advances in neural information processing systems **32**
60. Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13, pp. 740–755. Springer
61. Kay W, Carreira J, Simonyan K, Zhang B, Hillier C, Vijayanarasimhan S, Viola F, Green T, Back T, Natsev P, et al (2017) The kinetics human action video dataset. arXiv preprint arXiv:1705.06950
62. Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J (2021) Critical assessment of methods of protein structure prediction (casp)-round xiv. Proteins: Structure, Function, and Bioinformatics 89(12), 1607–1617
63. Zhou N, Jiang Y, Bergquist TR, Lee AJ, Kacsoh BZ, Crocker AW, Lewis KA, Georghiou G, Nguyen HN, Hamid MN (2019) The cafa challenge reports improved protein function prediction and new functional annotations for hundreds of genes through experimental screens. Genome Biol 20(1):1–23
64. Riesselman AJ, Ingraham JB, Marks DS (2018) Deep generative models of genetic variation capture the effects of mutations. Nat Methods 15(10):816–822
65. Gray VE, Hause RJ, Luebeck J, Shendure J, Fowler DM (2018) Quantitative missense variant effect prediction using large-scale mutagenesis data. Cell Syst 6(1):116–124
66. Fowler DM, Fields S (2014) Deep mutational scanning: a new style of protein science. Nat Methods 11(8):801–807
67. Veleckỳ J, Hamsikova M, Stourac J, Musil M, Damborsky J, Bednar D, Mazurenko S (2022) Soluprotmutdb: a manually curated database of protein solubility changes upon mutations. Comput Struct Biotechnol J 20:6339–6347
68. Moal IH, Fernández-Recio J (2012) Skempi: a structural kinetic and energetic database of mutant protein interactions and its use in empirical models. Bioinformatics 28(20):2600–2607
69. Nikam R, Kulandaisamy A, Harini K, Sharma D, Gromiha MM (2021) Prothermdb: thermodynamic database for proteins and mutants revisited after 15 years. Nucleic Acids Res 49(D1):420–424
70. Dallago C, Mou J, Johnston KE, Wittmann BJ, Bhattacharya N, Goldman S, Madani A, Yang KK (2021) Flip: Benchmark tasks in fitness landscape inference for proteins. bioRxiv, 2021–11
71. Rao R, Bhattacharya N, Thomas N, Duan Y, Chen P, Canny J, Abbeel P, Song Y (2019) Evaluating protein transfer learning with tape. Advances in neural information processing systems **32**
72. Xu M, Zhang Z, Lu J, Zhu Z, Zhang Y, Chang M, Liu R, Tang J (2022) Peer: a comprehensive and multi-task benchmark for protein sequence understanding. Adv Neural Inf Processing Syst 35:35156–35173
73. Capel H, Weiler R, Dijkstra M, Vleugels R, Bloem P, Feenstra KA (2022) Proteinglue multi-task benchmark suite for self-supervised protein modeling. Sci Rep 12(1):16047
74. Almagro Armenteros JJ, Sønderby CK, Sønderby SK, Nielsen H, Winther O (2017) Deeploc: prediction of protein subcellular localization using deep learning. Bioinformatics 33(21):3387–3395
75. Thumuluri V, Almagro Armenteros JJ, Johansen AR, Nielsen H, Winther O (2022) Deeploc 2.0: Multi-label subcellular localization prediction using protein language models. Nucleic Acids Research 50(W1), 228–234
76. Szklarczyk D, Morris JH, Cook H, Kuhn M, Wyder S, Simonovic M, Santos A, Doncheva NT, Roth A, Bork P, et al (2016) The string database in 2017: Quality-controlled protein-protein association networks, made broadly accessible. Nucleic acids research, 937
77. Huang K, Fu T, Gao W, Zhao Y, Roohani Y, Leskovec J, Coley CW, Xiao C, Sun J, Zitnik M (2021) Therapeutics data commons: Machine learning datasets and tasks for drug discovery and development. arXiv preprint arXiv:2102.09548
78. Niwa T, Ying B-W, Saito K, Jin W, Takada S, Ueda T, Taguchi H (2009) Bimodal protein solubility distribution revealed by an aggregation analysis of the entire ensemble of Escherichia coli proteins. Proc Natl Acad Sci 106(11):4201–4206
79. Ouyang L, Wu J, Jiang X, Almeida D, Wainwright C, Mishkin P, Zhang C, Agarwal S, Slama K, Ray A (2022) Training language models to follow instructions with human feedback. Adv Neural Inf Proc Syst 35:27730–27744
80. Kudo T, Richardson J (2018) Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint arXiv:1808.06226
81. Su J, Lu Y, Pan S, Murtadha A, Wen B, Liu Y (2021) Roformer: Enhanced transformer with rotary position embedding. arXiv preprint arXiv:2104.09864
82. Suzek, B.E., Wang, Y., Huang, H., McGarvey, P.B., Wu, C.H., Consortium, U. (2015) Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. Bioinformatics 31(6):926–932
83. Stärk H, Dallago C, Heinzinger M, Rost B (2021) Light attention predicts protein location from the language of life. Bioinf Adv 1(1):035
84. Khurana S, Rawi R, Kunji K, Chuang G-Y, Bensmail H, Mall R (2018) Deepsol: a deep learning framework for sequence-based protein solubility prediction. Bioinformatics 34(15):2605–2613
85. Chen J, Zheng S, Zhao H, Yang Y (2021) Structure-aware protein solubility prediction from sequence through graph convolutional network and predicted contact map. J Cheminf 13(1):1–10
86. Wu NC, Dai L, Olson CA, Lloyd-Smith JO, Sun R (2016) Adaptation in protein fitness landscapes is facilitated by indirect paths. Elife 5:16965
87. McCallister EL, Alm E, Baker D (2000) Critical role of β-hairpin formation in protein g folding. Nat Struct Biol 7(8):669–673
88. Sauer-Eriksson AE, Kleywegt GJ, Uhlén M, Jones TA (1995) Crystal structure of the c2 fragment of streptococcal protein g in complex with the fc domain of human igg. Structure 3(3):265–278

89. Girod A, Wobus CE, Zádori Z, Ried M, Leike K, Tijssen P, Kleinschmidt JA, Hallek M (2002) The vp1 capsid protein of adeno-associated virus type 2 is carrying a phospholipase a2 domain required for virus infectivity. J Gen Virol 83(5):973–978
90. Bryant DH, Bashir A, Sinai S, Jain NK, Ogden PJ, Riley PF, Church GM, Colwell LJ, Kelsic ED (2021) Deep diversification of an aav capsid protein by machine learning. Nat Biotechnol 39(6):691–696
91. Vandenberghe L, Wilson J, Gao G (2009) Tailoring the aav vector capsid for gene therapy. Gene Ther 16(3):311–319
92. Jarzab A, Kurzawa N, Hopf T, Moerch M, Zecha J, Leijten N, Bian Y, Musiol E, Maschberger M, Stoehr G (2020) Meltome atlas-thermal proteome stability across the tree of life. Nat Methods 17(5):495–503
93. Yeoman CJ, Han Y, Dodd D, Schroeder CM, Mackie RI, Cann IK (2010) Thermostable enzymes as biocatalysts in the biofuel industry. Adv Appl Microbiol 70:1–55
94. Haki G, Rakshit S (2003) Developments in industrially important thermostable enzymes: a review. Biores Technol 89(1):17–34
95. Labas YA, Gurskaya N, Yanushevich YG, Fradkov A, Lukyanov K, Lukyanov S, Matz M (2002) Diversity and evolution of the green fluorescent protein family. Proc Natl Acad Sci 99(7):4256–4261
96. Sarkisyan KS, Bolotin DA, Meer MV, Usmanova DR, Mishin AS, Sharonov GV, Ivankov DN, Bozhanova NG, Baranov MS, Soylemez O (2016) Local fitness landscape of the green fluorescent protein. Nature 533(7603):397–401
97. Willig KI, Kellner RR, Medda R, Hein B, Jakobs S, Hell SW (2006) Nanoscale resolution in gfp-based microscopy. Nat Methods 3(9):721–723
98. Rocklin GJ, Chidyausiku TM, Goreshnik I, Ford A, Houliston S, Lemak A, Carter L, Ravichandran R, Mulligan VK, Chevalier A (2017) Global analysis of protein folding using massively parallel design, synthesis, and testing. Science 357(6347):168–175
99. Shoichet BK, Baase WA, Kuroki R, Matthews BW (1995) A relationship between protein stability and protein function. Proc Natl Acad Sci 92(2):452–456
100. Guo Y, Yu L, Wen Z, Li M (2008) Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences. Nucleic Acids Res 36(9):3025–3030
101. Salwinski L, Miller CS, Smith AJ, Pettit FK, Bowie JU, Eisenberg D (2004) The database of interacting proteins: 2004 update. Nucleic acids research 32(suppl_1), 449–451
102. Hashemifar S, Neyshabur B, Khan AA, Xu J (2018) Predicting protein-protein interactions through sequence-based deep learning. Bioinformatics 34(17):802–810
103. Yu H, Braun P, Yıldırım MA, Lemmens I, Venkatesan K, Sahalie J, Hirozane-Kishikawa T, Gebreab F, Li N, Simonis N (2008) High-quality binary protein interaction map of the yeast interactome network. Science 322(5898):104–110
104. Pan X-Y, Zhang Y-N, Shen H-B (2010) Large-scale prediction of human protein-protein interactions from amino acid sequence based on latent topic features. J Proteome Res 9(10):4992–5001
105. Peri S, Navarro JD, Amanchy R, Kristiansen TZ, Jonnalagadda CK, Surendranath V, Niranjan V, Muthusamy B, Gandhi T, Gronborg M (2003) Development of human protein reference database as an initial platform for approaching systems biology in humans. Genome Res 13(10):2363–2371
106. Rhodes DR, Tomlins SA, Varambally S, Mahavisno V, Barrette T, Kalyana-Sundaram S, Ghosh D, Pandey A, Chinnaiyan AM (2005) Probabilistic model of the human protein-protein interaction network. Nat Biotechnol 23(8):951–959
107. Rual J-F, Venkatesan K, Hao T, Hirozane-Kishikawa T, Dricot A, Li N, Berriz GF, Gibbons FD, Dreze M, Ayivi-Guedehoussou N (2005) Towards a proteome-scale map of the human protein-protein interaction network. Nature 437(7062):1173–1178
108. Yu H, Tardivo L, Tam S, Weiner E, Gebreab F, Fan C, Svrzikapa N, Hirozane-Kishikawa T, Rietman E, Yang X (2011) Next-generation sequencing to generate interactome datasets. Nat Methods 8(6):478–480
109. Chen M, Ju CJ-T, Zhou G, Chen X, Zhang T, Chang K-W, Zaniolo C, Wang W (2019) Multifaceted protein-protein interaction prediction based on siamese residual rcnn. Bioinformatics 35(14):305–314
110. Guney E, Menche J, Vidal M, Barábasi A-L (2016) Network-based in silico drug efficacy screening. Nat Commun 7(1):10331
111. Hillenmeyer S, Davis LK, Gamazon ER, Cook EH, Cox NJ, Altman RB (2016) Stams: string-assisted module search for genome wide association studies and application to autism. Bioinformatics 32(24):3815–3822
112. Thul PJ, Åkesson L, Wiking M, Mahdessian D, Geladaki A, Ait Blal H, Alm T, Asplund A, Björk L, Breckels LM (2017) A subcellular map of the human proteome. Science 356(6340):3321
113. Delmolino LM, Saha P, Dutta A (2001) Multiple mechanisms regulate subcellular localization of human cdc6. J Biol Chem 276(29):26947–26954
114. Millar AH, Carrie C, Pogson B, Whelan J (2009) Exploring the function-location nexus: using multiple lines of evidence in defining the subcellular location of plant proteins. Plant Cell 21(6):1625–1631
115. Gimpelev M, Forrest LR, Murray D, Honig B (2004) Helical packing patterns in membrane and soluble proteins. Biophys J 87(6):4075–4086
116. Kanner EM, Friedlander M, Simon SM (2003) Co-translational targeting and translocation of the amino terminus of opsin across the endoplasmic membrane requires gtp but not atp. J Biol Chem 278(10):7920–7926
117. Nielsen H, Tsirigos KD, Brunak S, Heijne G (2019) A brief history of protein sorting prediction. Protein J 38:200–216
118. Davis GD, Elisee C, Newham DM, Harrison RG (1999) New fusion protein systems designed to give soluble expression in Escherichia coli. Biotechnol Bioeng 65(4):382–388
119. Trainor K, Broom A, Meiering EM (2017) Exploring the relationships between protein sequence, structure and solubility. Curr Opin Struct Biol 42:136–146
120. Shimizu Y, Kanamori T, Ueda T (2005) Protein synthesis by pure translation systems. Methods 36(3):299–304
121. Costa S, Almeida A, Castro A, Domingues L (2014) Fusion tags for protein solubility, purification and immunogenicity in Escherichia coli: the novel fh8 system. Front Microbiol 5:63
122. Hebditch M, Carballo-Amador MA, Charonis S, Curtis R, Warwicker J (2017) Protein-sol: a web tool for predicting protein solubility from sequence. Bioinformatics 33(19):3098–3100
123. Musil M, Konegger H, Hon J, Bednar D, Damborsky J (2018) Computational design of stable and soluble biocatalysts. Acs Catalysis 9(2):1033–1054
124. Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Žídek A, Potapenko A (2021) Highly accurate protein structure prediction with alphafold. Nature 596(7873):583–589
125. Baek M, DiMaio F, Anishchenko I, Dauparas J, Ovchinnikov S, Lee GR, Wang J, Cong Q, Kinch LN, Schaeffer RD (2021) Accurate prediction of protein structures and interactions using a three-track neural network. Science 373(6557):871–876
126. Hou J, Adhikari B, Cheng J (2018) Deepsf: deep convolutional neural network for mapping protein sequences to folds. Bioinformatics 34(8):1295–1303
127. Fox NK, Brenner SE, Chandonia J-M (2014) Scope: structural classification of proteins-extended, integrating scop and astral data and classification of new structures. Nucleic Acids Res 42(D1):304–309
128. Chen D, Tian X, Zhou B, Gao J, et al (2016) Profold: Protein fold classification with additional structural features and a novel ensemble classifier. BioMed research international **2016**

Publisher’s Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.