SPRoBERTa: protein embedding learning with local fragment modeling

Lijun Wu, Chengcan Yin, Jinhua Zhu, Zhen Wu, Liang He, Yingce Xia, Shufang Xie, Tao Qin and Tie-Yan Liu

Corresponding author: Lijun Wu, Microsoft Research AI4Science, Beijing, 100080, China. E-mail: lijwu@microsoft.com

Abstract

Well understanding protein function and structure in computational biology helps in the understanding of human beings. To face the limited proteins that are annotated structurally and functionally, the scientific community embraces the self-supervised pre-training methods from large amounts of unlabeled protein sequences for protein embedding learning. However, the protein is usually represented by individual amino acids with limited vocabulary size (e.g. 20 type proteins), without considering the strong local semantics existing in protein sequences. In this work, we propose a novel pre-training modeling approach SPRoBERTa. We first present an unsupervised protein tokenizer to learn protein representations with local fragment pattern. Then, a novel framework for deep pre-training model is introduced to learn protein embeddings. After pre-training, our method can be easily fine-tuned for different protein tasks, including amino acid-level prediction task (e.g. secondary structure prediction), amino acid pair-level prediction task (e.g. contact prediction) and also protein-level prediction task (remote homology prediction, protein function prediction). Experiments show that our approach achieves significant improvements in all tasks and outperforms the previous methods. We also provide detailed ablation studies and analysis for our protein tokenizer and training framework.

Keywords: local fragment representation, protein tokenizer, protein pre-training

Introduction

Proteins are essential parts of organisms and participate in almost every process within cells. For example, enzyme proteins catalyze biochemical reactions and structure proteins are important to maintain cell shape. We humans also need proteins to provide the necessary amino acids that we cannot synthesize. Hence, understanding protein is critical to understanding human beings, which increases the importance of protein representation modeling with machine learning methods. Over the past decades, the size of protein database has grown exponentially. In contrast, obtaining meaningful annotations for the protein sequences in database requires huge cost, and the functionally and structurally annotated proteins are only a fraction of the large-scale generated proteins. Therefore, there is a great need of protein analysis methods that can leverage abundant unlabeled protein sequences.

Recently, the scientific community embraces the self-supervised pre-training approaches (such as BERT [1], RoBERTa [2]) from natural language processing (NLP) to improve the representation learning for protein sequences. The common procedure is to first pre-train large neural language models across millions of unlabeled protein sequences to learn universal protein embeddings, and then fine-tune the pre-trained models on different downstream protein tasks.

However, directly taking the protein sequences as text sentences to pre-train the language model has some drawbacks, and there are many differences between protein sequences and text sentences. First, from the semantic information aspect, ‘word’ (several sequential characters) is usually considered as the minimal semantic unit in NLP and each word can convey a certain type of meaning even without other contexts. As for protein...
sequences, there are several important amino acid-level tasks such as secondary structure prediction (SSP), and also amino acid pair-level tasks such as contact prediction. Therefore, the minimal unit should be processed on each amino acid. However, modeling the amino acid itself may not contain enough information for understanding the global protein sequence with it only. A token for the type of amino acids is much less informative than a token for the type of words. Therefore, the basic unit of amino acid in the common protein pre-training works ignores the significant difference on semantic richness between protein and text. Second, from the pre-training view, the vocabulary in the protein pre-training models is about two dozen types of amino acids, while in NLP, the vocabulary size usually contains more than 10,000 words. Hence, it is obvious that each ‘word’ can represent more distinguishable information compared with amino acid, which can ease the training of language models since the masked language modeling (MLM) in pre-training is indeed a conditional prediction task based on the context information from other tokens. Third, the self-attention module in Transformer [3] model is a global attention over the protein sequence, which may not well model the strong local context since the abundant amino acids distract the attention over the long protein sequence.

To address the above differences, it is important to incorporate the local fragment information when building the protein pre-training model to benefit the context representation. One solution is that we can use k-mers [4], which are usually adopted in genomics or DNA sequences, to represent the local fragments in protein sequences. However, the k-mers are fixed local patterns without specific semantic meanings and flexible modeling ability. To alleviate the above drawbacks, in this paper, we explicitly model the local fragment of the protein sequence with an unsupervised protein tokenizer (learned from data) to help the pre-training model enhance the protein embeddings. Specifically, we utilize SentencePiece [5], a subword encoding algorithm which is widely used in NLP, to split the protein sequence into different pieces. The algorithm scans the large amount of protein sequences by a unigram language model and successive amino acids with higher frequency will be split as local fragment. Intuitively, statistically frequent protein sequence patterns may link with intrinsic rules of permutation of amino acids in nature proteins. Compared with single amino acids, these sequence patterns provide more local context information. Furthermore, we introduce a novel pre-training framework to integrate the local protein patterns and amino acids as input. In addition to the MLM on amino acid, we propose to predict the masked local fragments as another pre-training task, which we found is very effective. Our pre-training model is named as SPRoBERTa since the architecture is based on the well-known RoBERTa [2] training framework.

To evaluate the effectiveness of our SPRoBERTa, we pre-train the Transformer [3] encoder model on Pfam dataset and then fine-tune the model on different tasks, including the amino acid-level prediction (SSP), the amino acid pair-level prediction (contact prediction) and the protein sequence-level prediction (remote homology prediction, protein function prediction). On all experiments, we find our pre-trained and fine-tuned model boosts the task performance significantly, which can demonstrate the great value of introducing the local protein patterns into protein representation models.

Related work

Protein representation learning

Our work aims to obtain powerful representation for protein sequences with pre-training methods. In this line, [6, 7] are among the first works to apply deep learning to obtain protein representations from unlabeled protein sequences. Later, following the principle of pre-training and then fine-tuning in the NLP field [8, 9], UDSMProt [10], P-ELMO [7] and Seq-Vec [11] all take this way to empower the protein representation. However, they all take the ELMo [8] based model as backbone. Since BERT [1] based pre-training models can well capture the bidirectional context information with MLM task, TAPE [12] takes the same model for pre-training protein sequences. It further creates a benchmark for evaluating the pre-trained representations on different downstream protein tasks. ESM [13–15] series works pre-train the protein model with diverse data and more elaborate settings, and they demonstrate strong learning ability of the pre-trained protein representations. Recently, more works adopt the similar training methods [16–19] with advanced architecture or data resources, and they all show the benefits of protein pre-training models.

Subword encoding

Subword encoding is effective in NLP field to solve the open vocabulary problem (rare and unknown words); the intuition is that various word classes are composed of smaller units than words. Generally speaking, it can extract the combination of few continuous characters so that each token is not too coarse-grained to represent one word compared with the word-level vocabulary. Widely used subword encoding algorithms include Byte-Pair Encoding (BPE) [20], SentencePiece [5, 21] and WordPiece [22], and they all split sentences into subword units by statistics of consequent characters.

The most related works to ours is [23] and [24], both of which leverage the BPE encoding to learn the protein embeddings, in which the protein sequence is tokenized by subword token instead of the amino acids. The differences between them are that [24] trains on a larger dataset and bigger model compared with [23]. Our work is different from them. Concretely, our method can encode both the amino acid input and the local fragment input, while their works only focus on the subword input. Besides, our approach can support amino acid-level, amino acid pair-level and also protein sequence-level tasks, while they can only support the protein sequence-level task since no amino acid information existed in the input. Furthermore, to support the different protein tasks, we introduce a novel integrated framework at the input stage with both amino acid and local fragment representations.

Methods

In this section, we introduce the details of our SPRoBERTa framework. We first introduce the RoBERTa pre-training and its utilization on protein sequences. Then we describe the local fragment tokenization. Finally, the newly added pre-training tasks and the overall framework of integrated amino acid and local fragment representations pre-training are presented.

RoBERTa pre-training

Our pre-training framework is based on RoBERTa [2], a well-known and effective pre-training method in NLP field. RoBERTa is a widely used Transformer encoder [3] based architecture, in which the self-attention module, and the feed-forward network contribute to the strong modeling ability of the sequences. Besides, the bidirectional context modeling of Transformer encoder makes it great to capture the global information of the input sequence.

Specifically, for an input sentence $x = \{x_1, x_2, \ldots, x_n\}$, where $n$ is the length of the sequence, and each $x_i$ is a token in the sequence. The protein sequence $x = \{[CLS], x_1, x_2, \ldots, x_n\}$. We omit $[CLS]$ here.
for short description, but [CLS] token is important and will be used for fine-tuning. The input representation of each token \( x_i \) contains two kinds of embeddings, one is the token embedding \( e_x \) (retrieved from an embedding table), and the other one is the position embedding \( p_i \) for each token position in the sequence (learned or fixed). The two embeddings are then summed to make the final input representation \( e_i = e_x + p_i \). With \( e_i \) for each token, the Transformer encoder will take \( e_i \) as input, and transforms it to the hidden representation \( h_i \) by self-attention module and feed-forward network along with residual connection [25]:

\[
h_i = FFN(h_i) + h_i, \quad h_i = SA(e_i) + e_i, \tag{1}
\]

where \( FFN \) is the feed-forward network and \( SA \) is the self-attention mechanism, which is defined by

\[
FFN(x) = \max(0,xW_1 + b_1)W_2 + b_2, \tag{2}
\]

\[
SA(q, k, v) = \text{softmax} \left( \frac{qW_s \times kW_s}{\sqrt{d_k}} \right)(vW_v), \tag{3}
\]

where \( W_{1,2,q,k,v} \) and \( b_{1,2} \) are parameters and \( \sqrt{d_k} \) is the hidden dimension. For self-attention, the \( q, k, v \) are the tokens \( x_i \) in the sequence. The processed hidden state \( h_i \) can be finally used for prediction (We omit the multi-head operation for attention, and the layer normalization operation.).

RoBERTa pre-training adopts the MLM task. Several tokens in the input sequence are selected and replaced with the special token [MASK]. The objective of MLM is to reconstruct the original tokens for these [MASK] positions. The implementation is that 15% of the input tokens are dynamically masked for each sequence. Then the sequence with the masked tokens are fed into the Transformer encoder, and the hidden states \( h \) are obtained after the encoder processing as described above. Mathematically speaking, the loss function of MLM pre-training task is

\[
L_{\text{MLM}}(x) = - \sum_{m=1}^{M} \log p(x_m|\hat{x}), \tag{4}
\]

where \( x_m \) is the masked token and \( \hat{x} \) is the sequence with masked tokens.

In the scenario of protein pre-training, the input sequence \( x \) is the protein sequence, and each token \( x_i \) is the specific amino acid from the near 20 types of amino acids. The pre-training task is then to recover the masked amino acid in the protein sequence conditioned on other not masked amino acids. This protein pre-training task is the most commonly used method nowadays [12, 13, 17].

**Local fragment tokenization**

As we discussed in the introduction, there are some drawbacks for only involving the amino acid representations for protein pre-training. Therefore, we attempt to introduce semantic-rich representation methods for protein pre-training.

In this work, we leverage the SentencePiece [5] tokenizer, a widely adopted subword tokenization method in NLP, for protein local fragment extraction. SentencePiece is a subword segmentation algorithm based on a unigram language model and it is capable of extracting multiple subword segmentations with probabilities. The unigram language model aims to find the most probable segmentation for an input sentence if there are multiple segmentation candidates. Through iterative searching and optimization of the unigram language model with EM optimizing each loss for the subword, those top ones with small losses will be kept. The process iteratively goes until the desired vocabulary size is reached. The specific algorithm can be found in the original paper [5, 21]. We directly take the open-source toolkit for implementation.

For protein sequence, the SentencePiece algorithm first learns a unigram language model on FASTA sequences; the algorithm stops until the desired local fragment vocabulary size is reached (we study the different size of vocabulary in Section 5.1). Then the learned model is applied on all FASTA sequence for local fragment segmentation. The separated sequences are then used for pre-training.

To better show the local fragment extraction, we take one FASTA sequence as an example and present the results after segmentation. The different representations for this protein sequence are shown in Figure 2. For the amino acid representation, it is simply split by the space token between amino acids, while for the local fragment presentation, several continuous amino acids are combined, and there are different local fragments, such as 'QRI' with three amino acids, 'CT' with two amino acids and 'M' with one amino acid. Take Cysteine (C) as an example; we show the relative frequency (deep color means high frequency) of its related tokens in the two different tokenized vocabularies in Figure 2. It is obvious that the different local fragments contain 'C' which will be used for training. Each kind of local fragment may have its own meaning after learning, and we will show some analysis about its relationship to the secondary structure in Section 5.3.

**Local fragment recovering task**

As we introduced above, the traditional pre-training task for RoBERTa is the MLM, and for protein pre-training, it is amino acid MLM accordingly. That is, the model aims to predict the types of the individually masked amino acids given the whole protein sequence (Eqn. 4). Since we incorporate the local fragment into the presentation to induce the model to learn relationship among these local fragments, we propose another pre-training task, which is the local fragment recovering task, which tries to recover the masked local fragment and adapts the masking strategy to avoid leaking information.

Concretely, we mask the local fragment tokens with continuous amino acids instead of masking individual amino acids. Then the model takes the protein sequence with masked local fragment representations as input, after encoding with RoBERTa model, the goal is to recover the masked local fragments at the masked positions. Mathematically, suppose the protein sequence is now represented as \( x = (s_i)_{i=1}^{k} \), where \( k \) is the length of the local fragments in \( x \); the model is to minimize the following loss function:

\[
L_{\text{MLM}_s}(x) = - \sum_{m=1}^{M} \log p(s_m|\hat{x}), \tag{5}
\]

where \( s_m \) denotes the masked local fragment tokens and \( \hat{x} \) is the sequence with masked tokens.

**SПрoBERTa framework**

The overall framework of our SПрoBERTa framework is shown in Figure 1. From the figure, the input protein contains two parts, the
Figure 1. The overall framework of our SPRoBERTa. Take a virtual protein sequence as an example here, the protein ‘YMKDRV’ will be tokenized to ‘YM K D R V’ by amino acid representation, and ‘YM KDR V’ by local fragment representation. The two representations will be added together with two different position embeddings, then the Transformer model will take them as input to pre-train two different tasks, amino acid masked language modeling and local fragment recovering.

amino acid tokens and the local fragment tokens, corresponding to the amino acid input embeddings and the local fragment embeddings, respectively. Besides, as for the position embeddings to indicate the different positions of the tokens in the sequence, we design two different types. For the amino acid tokens, each position has a unique position embedding which can be learned or fixed. For the local framework tokens, we need to specially design the corresponding position embeddings. As we mentioned that our method can support both amino acid-level task and the protein sequence-level task, we need to maintain both the amino acid and the local fragment information in our model. Therefore, for the local fragment, we need to make it of the same length as amino acid representations. Specifically, we repeat the same local fragment k times, where k is the amino acid tokens in the continuous local fragment. Correspondingly, for the local fragment, we introduce a relative position embedding for the repeated local fragment representations to distinguish these same local fragment tokens. For example, if the protein sequence is “YMKDRV”, the segmented local fragments are “YM K D R V”. We repeat “YM” two times, “KDR” three times and “V” once, as shown in the figure. The local fragment position embeddings are then $p_1, p_2$ for two “YM” tokens and $p_1, p_2, p_3$ for repeated “KDR” tokens. Then, the amino acid embedding, local fragment embedding and the two kinds of position embeddings are summed as input for the SPRoBERTa model.

For the masking strategy in our pre-training model, we design a coordinated strategy for amino acid MLM and local fragment recovering. Concretely, we mask over the local fragment tokens, and the corresponding amino acids in the local fragment will also be masked. The objective is to predict the masked amino acids and the local fragment tokens, and the overall loss function is $L = L_{MLM_{aa}} + L_{MLM_{LF}}$. Take the example in the figure for illustration; the repeated ‘KDR’ tokens and the ‘K’, ‘D’, ‘R’ amino acids are masked, the model needs to recover ‘KDR’ and ‘K’, ‘D’, ‘R’ accordingly. In this way, the amino acid and local fragment information can both be learnt and kept for fine-tuning.

Experiments
In this section, we evaluate our SPRoBERTa model by pre-training and fine-tuning on different protein analysis tasks. We first introduce the pre-training settings, then the fine-tuning tasks, and report the results on these different protein downstream tasks.

Pre-training dataset
Same as previous works [12, 24], we use Pfam [26] dataset for pre-training. Pfam dataset contains 31 million protein domains and is extensively used in Bioinformatics. The protein sequences are clustered into evolutionary-related families. For a fair comparison, we take the processed version of Pfam data from [12], and we randomly split the data into training, validation and test sets. Further, we also pre-train on UniRef50 [27] dataset, which is verified to be a better pre-training dataset for comparison [13].
Fine-tuning tasks

To evaluate the SPRoBERTa pre-training, we fine-tune on different protein tasks. Most benchmark datasets are evaluated by TAPE [12] except the protein function prediction task.

- **SSP** is a sequence annotation task where each amino acid is mapped to a label in {Helix, Strand, Other}. SSP is highly related to the protein structure for understanding the protein function. The training/test data are from NetSurfP-2.0 [28]/CB513 [29]. The evaluation is the per amino acid prediction accuracy.

- **Contact Prediction** is an amino acid pair-level task that each pair of amino acids in the protein is mapped to a label \{0, 1\} to indicate whether the pair is 'in contact' (< 8 Å apart) or not. It is important for the full 3D protein structure prediction. For contact prediction, we use two different evaluation datasets. One is the ProteinNet dataset used in TAPE [12] and test data are from CASP12. The other one is from ESM [13]. We report precision of the L/S most likely contacts for the short-, medium- and long-range contacts.

- **Remote Homology Prediction** is a protein sequence-level classification task; each protein is mapped to a label in {1,...,1195}, where the label means different protein folds. The training and test data are from [31], the classification accuracy is reported.

- **Protein Function Prediction** or Gene Ontology (GO) term prediction is a protein sequence-level classification task. The GO terms are organized into three categories: molecular function (MF), biological process (BP) and cellular component (CC). We follow DeepFRI [32] to process the data. The area under the Precision-Recall curve (AUPRC) is reported for this task.

Pre-training and fine-tuning settings

We use the RoBERTa [2] base setting for our protein pre-training. Specifically, there are 12 Transformer encoder layers, the embedding size is 768, the feed-forward hidden units are 3072 and the attention heads are 12. The detailed configurations are as follows. We use batch size with 4096 tokens and we keep the training protein with max length 768. The Adam [33] optimizer is used for training with initial learning rate 0.0006, the learning schedule is polynomial_decay, dropout is set as 0.1 and weight decay is 0.01. The total training step is about 200k.

For fine-tuning, we use the same hyperparameters for SSP, contact prediction, remote homology and protein function prediction tasks. We use Adam [33] as the optimizer with peaking learning rate 0.00001. The learning rate schedule is polynomial_decay and weight decay is 0.01. We also clip the gradient norm with value 1.0. We fine-tune the models for 100 epochs and use early stopping with 20 epochs as the patience.

For the input vocabulary of amino acid and local fragment, we use open-sourced SentencePiece toolkit (https://github.com/google/sentencepiece) from Google for local fragment tokenization. We train the unigram language model with default configuration, and the vocabulary size for the local fragments we set is (10k, 30k, 50k). As for the amino acid vocabulary, it contains the most common 20 types and several special tokens.

Compared methods

We mainly compare our method with the following methods on SSP, contact prediction and remote homology prediction:

- **TAPE** [12] is the first benchmark of protein embedding pre-training. It consists of 12 Transformer encoder layers with a hidden size of 512 units and eight attention heads.

- **RoBERTa Base** [2] is the backbone model that we used, which is an optimized variant of BERT [1]. Here, we pre-train the RoBERTa Base model on the protein data and compare the results. Note that this model only trains on the amino acid representation.

- **BPE Longformer** [24] is the most related work to ours in that the BPE algorithm is used for protein tokenization. It sets the BPE vocabulary for protein sequence as 10k, and we set the same size for fair comparison.

- **CPCProt** [34] is a self-supervised contrastive learning model of protein representations by mutual information maximization. It is also based on TAPE pre-training data.

- **Profile Prediction** is a new pre-training task introduced by [35]. The alignment profile is calculated from the multiple sequence alignment (MSA) [36] proteins, which is supposed to be more informative than the raw protein sequence.
Table 1. Main results of our SPRoBERTa model and other compared methods. The evaluation metric for contact prediction is the average precision of L/5 most likely predicted medium&long-range contacts on TAPE test set, long-range on ESM test set, which are same as TAPE paper [12]. RoBERTa Base* is our reproduced backbone model that trained only on the amino acid representation. ESM-1b* has multiple stacked convolutional blocks on the pre-trained model for fine-tuning, which leads to high performance. The results for baseline methods are reported from the original paper, and '-' means this task is not fine-tuned in the paper.

<table>
<thead>
<tr>
<th>Methods</th>
<th>SSP</th>
<th>CP</th>
<th>RH</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TAPE</td>
<td>ESM</td>
<td></td>
</tr>
<tr>
<td>TAPE [12]</td>
<td>0.730</td>
<td>0.360</td>
<td>0.232</td>
</tr>
<tr>
<td>RoBERTa Base* (Pfam)</td>
<td>0.752</td>
<td>0.376</td>
<td>0.214</td>
</tr>
<tr>
<td>RoBERTa Base* (UniRef50)</td>
<td>0.782</td>
<td>0.567</td>
<td>0.370</td>
</tr>
<tr>
<td>BPE Longformer [24]</td>
<td>0.669</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CPCProt [34]</td>
<td>0.700</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Profile Prediction [35]</td>
<td>0.740</td>
<td>0.330</td>
<td>-</td>
</tr>
<tr>
<td>ESM-1b* [13]</td>
<td></td>
<td></td>
<td>0.569</td>
</tr>
<tr>
<td>ProteinBERT [37]</td>
<td>0.740</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>OntoProtein [38]</td>
<td>0.820</td>
<td>0.400</td>
<td>-</td>
</tr>
<tr>
<td>SPRoBERTa-10k (Pfam)</td>
<td>0.763</td>
<td>0.462</td>
<td>0.256</td>
</tr>
<tr>
<td>SPRoBERTa-30k (Pfam)</td>
<td>0.767</td>
<td>0.516</td>
<td>0.271</td>
</tr>
<tr>
<td>SPRoBERTa-50k (Pfam)</td>
<td>0.769</td>
<td>0.474</td>
<td>0.262</td>
</tr>
<tr>
<td>SPRoBERTa-10k (UniRef50)</td>
<td>0.818</td>
<td>0.605</td>
<td>0.395</td>
</tr>
<tr>
<td>SPRoBERTa-30k (UniRef50)</td>
<td>0.816</td>
<td>0.632</td>
<td>0.391</td>
</tr>
<tr>
<td>Alignment Baseline [12]</td>
<td>0.800</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

For protein function prediction, we separately compared the results with the following methods.

- ESM-1b [13] is a huge Transformer encoder with 34 layers trained on UniRef50 dataset. Note that when fine-tuning on the downstream tasks, ESM-1b stacked with multiple network blocks on the pre-trained model.
- ProteinBERT [37] is a BERT-based modeling with GO annotations, which contains both local and global representations. The pre-training is on UniRef90 dataset.
- OntoProtein [38] also takes GO into protein pre-training with a contrastive learning method. It pre-trains on a new dataset ProteinKG25 created by the authors.

For protein function prediction, we separately compared the results with the following methods.

- DeepFRI [32] is a structure-based encoder which employs an LSTM model. It further constructs a residue graph with a three-layer graph GCN.
- ProtBERT-BFD [17] is a BERT-large model trained on 2.1 billion protein sequences from BFD [39].
- LM-GVP [40] is an extended model of GVP [41], which prepends ProtBERT [17] before GVP to utilize the sequence representation.
- GearNet [42] is a geometric structure pre-training model for protein representation. It utilizes a protein graph encoder with multi-view contrastive learning.

Main results

We report the main results of SSP, contact prediction, remote homology prediction in Table 1 and the results of GO term prediction in Table 2. For the baseline numbers, all of them are taken from the original papers, except the RoBERTa Base and the SPRoBERTa results, which are trained by us.

For SSP, with pre-training on Pfam dataset, the TAPE [12] accuracy is about 0.730, and our reproduced RoBERTa Base achieves 0.752. Other baseline methods are even worse than our RoBERTa Base model. Profile Prediction [35] leverages the MSA-related profile information for pre-training, but the accuracy is even worse than our RoBERTa Base. For our SPRoBERTa models with different vocabulary size, they obtain more than 0.76 accuracy, which surpass the RoBERTa Base model by more than 2.0% accuracy. Compared with TAPE, our SPRoBERTa achieves near 4.0% accuracy improvement. Our 0.769 accuracy is also closer to the Alignment Baseline result 0.80. With the better pre-training dataset UniRef50, the SSP accuracy further increases from 0.782 (RoBERTa Base*) to be 0.818, which even surpasses the alignment baseline result that requires large number of MSA [43] sequences. These results clearly prove the effectiveness of our SPRoBERTa pre-training.

For contact prediction, the precision results comparison are as follows. Upon Pfam pre-training data, on TAPE dataset, the medium&long-range contacts precision of TAPE [12] model is 0.360, and our reproduced RoBERTa Base model achieves 0.376 performance, which is slightly better. As for Profile Prediction [10], it obtains 0.330 precision, which is worse than TAPE baseline. For our SPRoBERTa models, we can achieve 0.462 precision when local fragment vocabulary is 10k, and this is much better than RoBERTa Base by more than 8.6% precision improvement (near 10% precision improvement than TAPE). When vocabulary size is 30k and 50k, we can obtain further significant improvements to get a 0.516 and 0.474 precision score. Similarly, for long-range contact on ESM dataset, our SPRoBERTa models surpass RoBERTa Base model by 4.0~6.0% precision score, which is also a large margin improvement. With UniRef50 pre-training data, the precision further improves to 0.632 and 0.395, which are also outstanding performances (though far away from the ESM-1b* model with 34-layer large parameters and highly careful grid search). These strong performances demonstrate that our local fragment representation based pre-training helps capture pairwise relations.

For remote homology prediction, we also achieve consistent improvements. The TAPE [12] baseline is about 0.210 accuracy, and our RoBERTa Base reproduces similar 0.217 accuracy. BPE Longformer [24] obtains about 4.0% gain with accuracy 0.256. Our SPRoBERTa model achieves a comparable result of 0.258 accuracy on Pfam pre-training data and 0.304 on UniRef50 pre-training set, which are better than previous works. Compared with BPE Longformer, the advantage of our SPRoBERTa is that our pre-trained model can be uniformly applied on different tasks including amino acid-level, pair-level and sequence-level, but BPE Longformer can only work on sequence-level task.
Table 3. Results of our SPRoBERTa model with 10k, 30k and 50k local fragment vocabulary size on SSP, contact prediction (CP), remote homology (RH) and GO term prediction (GO). We report the detailed precision results of medium&long-range, short-/medium-/long-range contacts for TAFE contact prediction and long-range contact for ESM contact prediction for fair comparisons.

<table>
<thead>
<tr>
<th>#Vocabulary</th>
<th>SSP</th>
<th>CP (TAFE)</th>
<th>CP (ESM)</th>
<th>RH</th>
<th>GO</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>M&amp;L</td>
<td>S</td>
<td>M</td>
<td>L</td>
</tr>
<tr>
<td>RoBERTa Base</td>
<td>0.752</td>
<td>0.376</td>
<td>0.377</td>
<td>0.329</td>
<td>0.286</td>
</tr>
<tr>
<td>SPRoBERTa-10k</td>
<td>0.763</td>
<td>0.462</td>
<td>0.370</td>
<td>0.400</td>
<td>0.346</td>
</tr>
<tr>
<td>SPRoBERTa-30k</td>
<td>0.767</td>
<td>0.516</td>
<td>0.390</td>
<td>0.398</td>
<td>0.403</td>
</tr>
<tr>
<td>SPRoBERTa-50k</td>
<td>0.769</td>
<td>0.474</td>
<td>0.353</td>
<td>0.378</td>
<td>0.376</td>
</tr>
</tbody>
</table>

Finally, for GO term prediction in Table 2, we can also see that SPRoBERTa can achieve consistent improvements on all category datasets with 3.0 – 6.0% gain. Though our results cannot catch up with the super large ESM-1b model, the superior performance compared with other methods can demonstrate the effectiveness of our SPRoBERTa, and our method is complementary to their works for better results.

Analysis

Beyond the above results, we provide several analysis to have a better understanding of our SPRoBERTa, including vocabulary comparison, ablation study, structure relations and so on. These study experiments are conducted on the pre-training models on the Pfam dataset.

Different local fragment vocabulary size

We have shown the main results of different vocabulary size (10k, 30k and 50k) on three different level tasks. To have a detailed comparison, we provide more results on these tasks. Specifically, we evaluate the fine-tuned SPRoBERTa model on three tasks, and we report the detailed number of short-/medium-/long-range contacts for contact prediction on TAFE dataset, and the long-range contact on ESM dataset. We also report on our backbone baseline RoBERTa Base model to have a clearer comparison. The results are shown in Table 3.

From the table, we can first see that when comparing with RoBERTa Base model, our SPRoBERTa pre-training model with local fragment representation improves each task with non-trivial margin, which can demonstrate the effectiveness of local fragment representation. Then for the specific different local fragment vocabulary size comparison, we can find that except the remote homology result is best on 10k vocabulary, for other tasks, including each of the specific range contacts, they prefer larger local fragment vocabulary size, 30k and 50k. When expanding the vocabulary from 30k to 50k, the improvement on SSP and other tasks is only marginal, e.g. 0.767 with 30k and 0.769 with 50k; the results of different range contact prediction even decrease, e.g. 0.390 with 30k and 0.353 with 50k; most of the tasks achieve best result with 30k vocabulary size. Therefore, it shows that 30k local fragments are relatively good for these tasks. The above results can prove that the local fragment representation indeed benefits the representation of the protein sequence both on token (local) level and sequence (global) level.

NLP vocabulary versus protein vocabulary

The pre-training and fine-tuning principle is developed from NLP, and then other fields take this effective method to achieve great success. Similar to our protein pre-training, we take the RoBERTa [2] model as our backbone. Therefore, it is interesting and necessary to compare the NLP with protein pre-training. In this section, we compare these two fields from vocabulary, e.g. the vocabulary for text sentences and protein sequences. Specifically, we count the vocabulary frequency of character-level and subword-level for text sentences, and the amino acid-level and local fragment-level for protein sequences. The text sentences are from the English wiki data (about 4.5M), and the protein sequences are from Pfam dataset (about 31M). We plot the vocabulary frequency in Figures 3 and 4. Figure 3 shows the frequency of top-60 characters and 20 amino acids, and Figure 4 shows the frequency of the subword of English text and local fragment of protein. For better presentation and comparison, we also show the bar of top-10 and bottom-10 frequency subword tokens of English text and local fragment vocabulary of protein.

From the figures, we can have several observations. (1) The amino acid vocabulary is small with 20 types, and the frequency of each type of the amino acid is relatively similar (blue line in Figure 3). For example, the highest frequency amino acid ‘L’ (Leucine) is about 8 times the lowest frequency amino acid ‘C’ (Cysteine). Compared with character vocabulary (green line Figure 3) of English text, it is obvious that the character vocabulary has a clear long-tail phenomenon so that the model can be confident of distinguishing the type of word consisting of characters at each position, while it is hard for amino acid to some extent. (2) The local fragment vocabulary (blue line in Figure 4) of protein is also long-tail (the top frequency ones are more than 100 times than bottom frequency ones, the blue bar in the bottom

Figure 3. Vocabulary statistics comparison of characters of English text sentences and amino acids of protein sequences. For better presentation, the character vocabulary for English text is the top-60 frequency characters. We also plot the L (leucine) and C (Cysteine) frequencies for comparison.
right of Figure 4), as well as the subword vocabulary (green line and the top left green bar in Figure 4) of English text. Therefore, it makes the local fragment tokens with top frequency in protein sequences easy to learn, and less for the bottom frequency tokens. The above observations clearly show differences between protein sequences and NLP texts, and the integration of local fragment vocabulary helps alleviate the differences so to benefit our protein pre-training.

Local fragment versus secondary structure

We have mentioned that our local fragment representation can reflect the structure information of protein. Therefore, in this section, we study the relationship between our learned local fragment tokens and the protein structure information. The results are presented in Table 4. Specifically, we look at the secondary structure information on CB513 dataset (three types of labels), and then map our local fragment tokens to the secondary structure label. We investigate the following statistics. (1) The uniqueness of the local fragment token and the secondary structure label (uniqueness mapping). For example, the ‘QRI’ local fragment token is only mapped to ‘121’ secondary structure label without other possible label choice. The uniqueness mapping ratio is calculated by comparing unique secondary structure label of vocabulary tokens with the full vocabulary tokens, e.g. 0.582 (2545/8270). (2) The consistency of the secondary structure label and the local fragment token (consistency mapping). For instance, ‘QRI’ is mapped to ‘111’ which means that each of the amino acid (‘Q’, ‘R’, ‘I’) in the local fragment token has the same secondary structure label ‘1’. The consistency mapping ratio is calculated by comparing local fragment tokens in the dataset that are consistent with secondary structure labels for all local fragment tokens, e.g. 0.683 (33292/48724). For the local fragment vocabulary, we study on the 10k and 30k SentencePiece operated vocabularies.

From Table 4, we can see that the consistency mapping ratio of the local fragment tokens is over 0.6, which is a relatively high ratio and means that the continuous amino acids in the local fragment token have the same secondary fragment label. Hence, grouping these amino acids to be a local fragment token is reasonable to represent the same structure information. Besides, the uniqueness mapping ratio of the local fragment tokens is 0.308(10k) and 0.582(30k), which means these local fragment tokens have unique secondary structure labels without any ambiguous label information. Therefore, grouping these amino acids to be a unit of local fragment token can benefit the learning of specific structure knowledge to enhance the protein representation. To better illustrate the relationship between local fragment token and the secondary structure label, we put several examples in Table 5. For example, ‘EMLR’ only maps to the secondary structure label ‘0000’, and ‘QM’ can be mapped to multiple label choices, ‘22’, ‘11’, ‘00’ and ‘12’.

Ablation study

We introduce the local fragment tokenization and the local fragment recovering pre-training task in our SPRoBERTa framework. To better understand the effectiveness of each component, we conduct ablation study in this section. Specifically, we study them in pre-training and fine-tuning stages differently, which include the following situations:

- Pre-training with/without local fragment recovering.
the pre-training stage, and this task must take the local fragment representation as input. We study first ablation as pre-training with or without the local fragment recovering task to evaluate its contribution.

- Fine-tuning with/without local fragment representation. The second ablation is to test the effect of local fragment representation during fine-tuning stage. In this way, the fine-tuning model only takes the amino acid representation without the local fragment representation as input. Note that this fine-tuning is exactly the same as our RoBERTa Base model fine-tuning.

The evaluation is conducted on the three tasks as in Table 1. We take the 10k, 30k, 50k vocabulary SPRoBERTa models as the evaluation model for the local fragment recovering study, and we simply take the 10k model for local fragment representation study. The results are shown in Table 6 and Table 7. From the tables, we can observe: (1) The local fragment recovering task is important for the pre-training model to obtain good representations. When removing the local fragment recovering task in the pre-training stage, we can find the results of all the tasks dropped significantly. For example, the contact prediction (TAPE) accuracy decreases from 0.462 to 0.447 for SPRoBERTa-10k, and for 30k model, it even drops from 0.516 to 0.416 with a large margin 10%. Similar decrease can be found on other tasks. This well demonstrates the importance and effectiveness of local fragment recovering pre-training task. (2) The local fragment representation also provides necessary information for protein learning. Though SSP task slightly differs when we remove the local fragment representation during fine-tuning (0.763/0.756 when fine-tuning w/w/o local fragment representation), it has huge impact to other protein tasks. For example, the accuracy of TAPE contact prediction is 0.408 without local fragment representation, which is far away from the model that fine-tuning with local fragment representation (0.462). Therefore, we can conclude that the local fragment representation brings important information to represent the protein, which is complementary to the amino acid representation. Another interesting observation is that even fine-tuning without the local fragment presentation, the model also surpasses the RoBERTa Base model on all tasks (also in Table 7).

Since this fine-tuning ablation is exactly the same as our RoBERTa Base fine-tuning, it means that our local fragment recovering task with the local fragment representation also enhances the amino acid representation learning during pre-training.

### Limitations

Though our SPRoBERTa framework can improve the protein representations, it also has several limitations. First, our method utilizes two vocabulary embedding matrices for protein sequence, e.g., the amino acid embedding and the local fragment embedding. Due to the large size of local fragment vocabulary, the model size is largely increased, which needs more computational cost. Second, our SPRoBERTa does not reduce the length of the protein sequence, which still faces the problem of long protein sequence modeling. Hence, these limitations are important directions for future works.

### Conclusion

Protein modeling has become more important for understanding protein function, which helps in the understanding of human beings. In this work, we propose a SPRoBERTa pre-training framework to model the protein representation. Different from previous works, which only take the amino acid representation or subword representation, our model takes both the amino acid and local fragment representations as integrated input. We also propose a local fragment recovering pre-training task on the local fragment tokens. Our pre-trained model can be fine-tuned on different protein analysis tasks and the experimental results show that our SPRoBERTa framework outperforms previous work with non-marginal performance improvements, along with detailed analysis of our model. In the future, we intend to build more advanced tokenizers for protein sequence. Besides, with the great success of AlphaFold2 [44], incorporating more structure-aware information in pre-training becomes promising.

### Key Points

- We introduce an unsupervised tokenizer into the protein pre-training model to address the limited amino acid tokenization, which can extract semantic local fragments in the protein sequence.
- We present a novel pre-training framework for protein embedding with different pre-training tasks, which effectively integrates the local fragments and amino acids in a protein sequence representation.
- We evaluate our pre-trained model by fine-tuning on different protein tasks, and the significant improved experimental results strongly prove the effectiveness of our approach.

### Acknowledgments

The authors thank the anonymous reviewers for their valuable suggestions.
References


