Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Subject Section

Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information

Seonwoo Min [1], Seunghyun Park [2], Siwon Kim [1], Hyun-Soo Choi [3], and Sungroh Yoon [1,4]*

[1]Department of Electrical and Computer engineering, Seoul National University, Seoul 08826, South Korea
[2]Clova AI Research, NAVER Corp., Seongnam 13561, South Korea
[3]AIX Center, SK Telecom, Seoul 04539, South Korea
[4]Interdisciplinary Program in Bioinformatics, ASRI, INMC, and Institute of Engineering Research, Seoul National University, Seoul 08826, South Korea

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Bridging the exponentially growing gap between the number of unlabeled and labeled proteins, a couple of works have adopted semi-supervised learning for protein sequence modeling. They pre-train a model with a substantial amount of unlabeled data and transfer the learned representations to various downstream tasks. Nonetheless, the current pre-training methods mostly rely on a language modeling task and often show limited performances. Therefore, a complementary protein-specific task for pre-training is necessary to better capture the information contained within unlabeled protein sequences.
Results: In this paper, we introduce a novel pre-training scheme called **PLUS**, which stands for **P**rotein sequence representations **L**earned **U**sing **S**tructural information. PLUS consists of masked language modeling and a complementary protein-specific pre-training task, namely same family prediction. PLUS can be used to pre-train various model architectures. In this work, we mainly use PLUS to pre-train a recurrent neural network (RNN) and refer to the resulting model as PLUS-RNN. It advances state-of-the-art pre-training methods on six out of seven tasks, *i.e.*, (1) three protein(-pair)-level classification, (2) two protein-level regression, and (3) two amino-acid-level classification tasks. Furthermore, we present results from our ablation studies and interpretation analyses to better understand the strengths of PLUS-RNN.
Availability: The codes and pre-trained models are available at https://github.com/mswzeus/PLUS/
Contact: sryoon@snu.ac.kr
Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Proteins consisting of linear chains of amino acids are one of the most versatile molecules in living organisms. They serve vital functions in prevalent biological mechanisms, *e.g.*, transmitting nerve pulses, storing and transporting other molecules, and providing immune protection (Berg *et al.*, 2006). The versatility of proteins is generally attributed to their diverse structures. Proteins naturally fold up into three-dimensional structures depending on the sequence of amino acids. Then, the structures have a direct impact on their functions.

With the advent of next-generation sequencing technologies, obtaining protein sequences has become relatively more accessible. Nonetheless, annotating a sequence with meaningful attributes still requires time-consuming and resource-intensive processes. Bridging the exponentially growing gap between the number of unlabeled and labeled protein sequences, a variety of *in silico* approaches have been widely adopted for predicting their numerous characteristics (Holm and Sander, 1996).

Sequence alignment is one of the key techniques in the computational protein biology. Alignment-based methods compare protein sequences using carefully designed scoring matrices (Eddy, 2004) or Hidden Markov

Models (HMMs) (Söding *et al.*, 2005). A correct alignment can group similar sequences together, provide information on conserved regions, and help us investigate uncharacterized proteins. However, not only does its computational complexity increase exponentially with the number of proteins, but it also shows difficulties in identifying distantly related proteins. Homologous proteins sharing a common evolutionary ancestor can have high sequence-level variations (Creighton, 1993). Therefore, simply comparing sequence similarities often fails to capture global structural and functional similarities of proteins (Bepler and Berger, 2019).

Building upon the success of deep learning, a number of works have also proposed deep learning algorithms for computational protein biology (Min *et al.*, 2017). Some of them use raw protein sequences and solely rely on the deep learning. Others may take in additional features. While they have advanced state-of-the-arts (SOTA) for various tasks, development of the deep learning algorithms requires highly task-specific processes: (1) It requires training a randomly initialized task-specific model from scratch. (2) It demands careful considerations on the selection of a model architecture and its hyperparameters tailored for each task. (3) Additional features also vary for each task such as alignment-based or known structural traits (Rao *et al.*, 2019).

Semi-supervised learning, which leverages both unlabeled and labeled data, has been one of the long-standing goals of broad machine learning community (Chapelle *et al.*, 2009). A semi-supervised learning algorithm pre-trains a universal model with a substantial amount of unlabeled data. Then, it transfers the learned representations and fine-tunes the model with a small amount of labeled data for each supervised task. The crux of semi-supervised learning is how to define a proper pre-training task. For example, recently, bidirectional encoder representations from Transformers (BERT) has been a new sensation in natural language processing (NLP) (Devlin *et al.*, 2018). BERT enabled more effective use of unlabeled text by proposing novel pre-training tasks for NLP, *i.e.*, masked language modeling (MLM) and next sentence prediction (NSP). The tasks guide a model to learn contextualized representations of words and relationship between sentences.

Now the natural question is, can protein biology also take advantage of semi-supervised learning? According to the linguistic hypothesis (AlQuraishi, 2019), naturally occurring proteins are not just random. Evolutionary pressure constrains them to a learnable manifold where indispensable structures and functions are maintained. Thus, by observing many proteins even without any annotations, we should be able to obtain an implicit understanding of the language of proteins. For instance, a couple of works have recently proposed pre-training methods for protein representations (Bepler and Berger, 2019; Alley *et al.*, 2019; Strodthoff *et al.*, 2019). They adopted language modeling (LM) from NLP and showed that pre-training helps for various downstream protein tasks. However, as tasks assessing protein embeddings (TAPE) have recently shown in their benchmark results (Rao *et al.*, 2019), the current pre-training methods are still often outperformed by task-specific models. It could be because LM alone is not enough, and a complementary protein-specific task for pre-training is necessary to better capture the information contained within the proteins.

In this paper, we introduce a novel pre-training scheme for protein sequence modeling called **PLUS**, which stands for **P**rotein sequence representations **L**earned **U**sing **S**tructural information. PLUS consists of MLM and an additional complimentary protein-specific pre-training task, namely same family prediction (SFP). SFP leverages computationally clustered protein families (Finn *et al.*, 2014) and trains a model to predict whether a pair of proteins belongs to a same family. PLUS can be used to pre-train various model architectures including a bidirectional recurrent neural network (BiRNN) and the Transformer (TFM). The resulting models are referred to as PLUS-RNN and PLUS-TFM, respectively. In this work, considering their sequential modeling capability and computational complexity, we mainly use PLUS-RNN. Afterwards, the pre-trained universal model can be fine-tuned on a variety of downstream tasks without training a randomly initialized task-specific models from scratch. It advances the pre-training SOTA methods on six out of seven protein biology tasks, *i.e.*, (1) three protein(-pair)-level classification, (2) two protein-level regression, and (3) two amino-acid-level classification tasks. Finally, we present results from our ablation studies and interpretation analyses to give a better understanding of the strengths of PLUS-RNN.

2 Related Works

2.1 Pre-training natural language representations

Pre-training natural language representations has been the basis of NLP research for a long time. A number of approaches have been proposed and their shared main component is LM. Traditional word2vec uses a skip-gram model which is directly trained to predict surrounding words given a representation of a center word (Mikolov *et al.*, 2013). The key idea is that ideal representations must convey syntactic and semantic information, and thus the representation of a token must be able to predict other tokens around. Note that in such formulation, all it needs is a sequence of tokens without any additional labels.

While early approaches learned context-independent representations, embeddings from language models (ELMo) generalized them to learn contextualized representations by adopting forward and reverse RNNs (Peters *et al.*, 2018). Given a sequence of tokens, the forward RNN sequentially processes the sequence left-to-right, and it is trained to predict the next token given its history. The reverse RNN is similar but processes the sequence in reverse, right-to-left. After the pre-training, hidden states of both RNNs are merged into a single vector representation for each token. Thus, unlike the previous word2vec, the same token can be transformed into different representations based on its contexts.

The major limitation of ELMo is that each RNN is trained using unidirectional LM and simply combined afterwards. Since valuable information often comes from both directions, unidirectional LM is inevitably sub-optimal. In contrast, BERT first proposed to pre-train bidirectional natural language representations using a multi-layer bidirectional TFM (Devlin *et al.*, 2018). The key element of the TFM is a self-attention layer composed of multiple individual attention heads (Vaswani *et al.*, 2017). Given an input sequence $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$, an attention head computes the output sequence $\mathbf{z} = [\mathbf{z}_1, \cdots, \mathbf{z}_n]$. Each token is a weighted sum of values, computed by a weight matrix $\mathbf{W}^V$:

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} (\mathbf{x}_j \mathbf{W}^V). \tag{1}$$

Each attention coefficient α_{ij} is the output of a softmax function applied on the dot products of the query with all keys, computed by $\mathbf{W}^Q$ and $\mathbf{W}^K$:

$$\alpha_{ij} = \frac{\exp(\mathbf{e}_{ij})}{\sum_{k=1}^{n} \exp(\mathbf{e}_{ik})}, \qquad \mathbf{e}_{ij} = \frac{(\mathbf{x}_i \mathbf{W}^Q)(\mathbf{x}_j \mathbf{W}^K)^T}{\sqrt{d_z}}, \tag{2}$$

where d_z is the the output token dimension. The self-attention layer directly performs $O(1)$ computations for all the pairwise tokens, whereas a recurrent layer requires $O(n)$ sequential computations for the farthest pair. It allows easier traversal for forward and backward signals, and thus, enables better capturing of long-range dependencies (Vaswani *et al.*, 2017).

The main contribution of BERT is that it introduced novel pre-training tasks for a multi-layer bidirectional model. Since its bidirectional conditioning allows each token to indirectly see itself, it cannot be pre-trained with the conventional LM. Instead, BERT resolved the problem by proposing an MLM task. It simply masks some input tokens at random and trains the model to predict them from the contexts. In addition, BERT adopts an NSP task which enables learning sentence relationships by training a model to predict whether a given pair of sentences is consecutive.

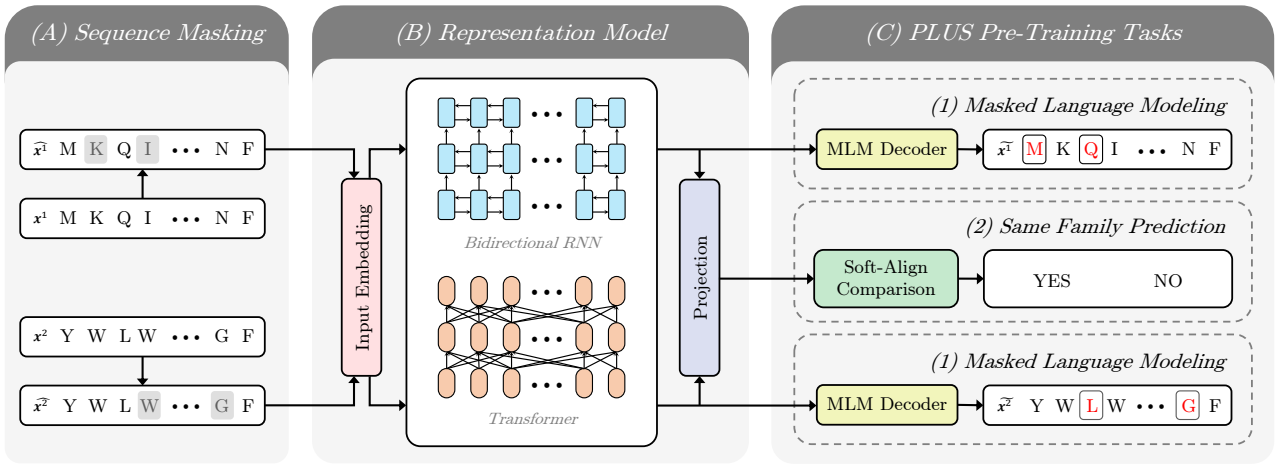

Fig. 1. Overview of PLUS pre-training scheme for protein sequence modeling. (A) We mask 15% of amino acids in each protein sequence at random. (B) PLUS trains a model to transform amino acids into sequences of bidirectional representations. (C) PLUS consists of two pre-training tasks. Masked language modeling (MLM) trains a model to predict the masked amino acids given their contexts. Same family prediction (SFP) trains a model to predict whether a pair of proteins belongs to a same protein family.

2.2 Pre-training protein sequence representations

Taking advantage of similarities to NLP, there is a long history of NLP-based methods adapted to learn protein sequence representations. Early approaches have focused on learning context-independent representations. For example, ProtVec (Asgari and Mofrad, 2015) and doc2vec (Yang *et al.*, 2018) generate non-overlapping 3-mers from protein sequences and pre-train their representations based on a skip-gram model from word2vec.

The most closely related previous works to our paper are P-ELMo (Bepler and Berger, 2019) and UniRep (Alley *et al.*, 2019), which learn contextualized protein representations. P-ELMo proposed a two-phase algorithm. First, it trains tied forward and reverse RNNs using the conventional LM with an unlabeled dataset. Then on top of them, it adopts another BiRNN and trains it with an additional small labeled dataset. Note that the latter supervised training deviates from the goal of pre-training, *i.e.*, utilizing low human-effort and large unlabeled datasets. UniRep used a unidirectional RNN with multiplicative long short-term memory (mLSTM) hidden units (Krause *et al.*, 2016) and trained the model using the conventional LM.

The current protein pre-training methods still have some major limitations and often lag behind task-specific models (Rao *et al.*, 2019). First, as in the previous methods in NLP, they only learn unidirectional representations from an unlabeld dataset. It is obvious that unidirectional representations are sub-optimal for numerous protein biology tasks, where it is crucial to assimilate global information from both directions. Second, they solely rely on LM to learn from unlabeled protein sequences. While LM is a simple and effective task, complementary pre-training task tailored for each data modality has been often the key to further improve the quality of representations. For instance, in NLP, BERT adopted the NSP task; a lite BERT (ALBERT) devised a complementary sentence order prediction (SOP) task to model inter-sentence coherence and showed consistent performance improvements for downstream tasks (Lan *et al.*, 2019).

3 Methods

We introduce PLUS, a novel pre-training scheme for protein sequence modeling (Figure 1). Consisting of MLM and complimentary protein-specific SFP pre-training tasks, PLUS can help a model to learn strucurally contextualized bidirectional representations. In the following, we will explain the details of the pre-training dataset, the model architectures, and the pre-training and fine-tuning procedures.

3.1 Pre-training dataset

As in P-ELMo, we use Pfam release 27.0 (Finn *et al.*, 2014) as the pre-training dataset. It contains total 21,827,419 protein sequences clustered into 16,479 families. Each protein family is computationally constructed by comparing sequence similarity of proteins using sequence alignments and HMMs. Due to the loose connection between sequence and structure similarities, the family labels only provide weak structural information (Elofsson and Sonnhammer, 1999). Nonetheless, we empirically show that the magnitude of the dataset complements the weakness and can help the model to learn structurally contextualized representations.

We use the training and test sets divided by a random 80/20% split and filter out the sequences shorter than 20 amino acids. Additionally, for the training set, we also remove the families containing less than 1,000 proteins. It results in 14,670,860 sequences from 3,150 families used for the following PLUS pre-training. Note that we have not done any ablation studies for the filtering conditions and other conditions may improve the results. The training and test datasets are available at our repository.

3.2 Model architecture

PLUS can be used to pre-train various model architectures including BiRNN and TFM. The resulting models are referred to as PLUS-RNN and PLUS-TFM, respectively. In this work, we mainly use PLUS-RNN based on its two advantages over PLUS-TFM. First, it is more effective for learning sequential nature of proteins. The self-attention layer of TFM performs dot products between all pairwise tokens regardless of their positions within the sequence (Equation 2). In other words, it gives equal opportunity to local and long-range contexts to determine the representations. While it facilitates learning long-range dependencies, its downside is that it completely ignores *locality bias* within a sequence. This is particularly problematic for protein biology, where local amino acid motifs often have more significant structural and functional implications (Bailey *et al.*, 2006). On the contrary, RNN sequentially processes a sequence, and local contexts are naturally more emphasized.

Second, PLUS-RNN provides lower computational complexity. Although it depends on the model hyperparameters, TFMs generally demand a larger number of parameters than RNNs (Vaswani *et al.*, 2017). Furthermore, the computations between all pairwise tokens in the self-attention layer place a huge computational burden scaling quadratically

with the input sequence length. Considering that pre-training typical TFMs handling 512 tokens already requires tremendous resources (Devlin *et al.*, 2018), it is computationally difficult to use TFMs to deal with longer protein sequences even up to a couple of thousand amino acids.

Given a protein sequence $\mathbf{x} = [\mathbf{x}_1, \cdots, \mathbf{x}_n]$ where $\mathbf{x}_i \in \{21 \text{ amino acid types}\}$[1], PLUS-RNN transforms it into a sequence of representations. First, an input embedding layer EM embeds each amino acid into a d_e-dimensional dense vector:

$$\mathbf{e} = [\mathbf{e}_1, \cdots, \mathbf{e}_n], \quad \mathbf{e}_i = \mathrm{EM}(x_i). \tag{3}$$

Then, a BiRNN of L-layers obtains bidirectional representations as a function of the entire sequence. We use long short-term memory (LSTM) as the basic unit of the BiRNN (Hochreiter and Schmidhuber, 1997). In each layer, it computes d_h-dimensional forward and backward hidden states ($\overrightarrow{\mathbf{h}}_i^l$ and $\overleftarrow{\mathbf{h}}_i^l$) and combines them into a hidden state $\mathbf{h}_i^l$ with a non-linear transformation:

$$\begin{aligned}
\overrightarrow{\mathbf{h}}_i^l &= \sigma(\overrightarrow{\mathbf{W}}_x^l \mathbf{h}_i^{l-1} + \overrightarrow{\mathbf{W}}_h^l \mathbf{h}_{i-1}^l + \overrightarrow{\mathbf{b}}^l), \\
\overleftarrow{\mathbf{h}}_i^l &= \sigma(\overleftarrow{\mathbf{W}}_x^l \mathbf{h}_i^{l-1} + \overleftarrow{\mathbf{W}}_h^l \mathbf{h}_{i+1}^l + \overleftarrow{\mathbf{b}}^l), \\
\mathbf{h}_i^l &= \sigma(\mathbf{W}_h^l [\overrightarrow{\mathbf{h}}_i^l; \overleftarrow{\mathbf{h}}_i^l] + \mathbf{b}^l) \quad for \quad l = 1, \cdots, L,
\end{aligned} \tag{4}$$

where $\mathbf{h}_i^0 = \mathbf{e}_i$; $\mathbf{W}$ and $\mathbf{b}$ are weight and bias vectors, respectively. We use the final hidden states $\mathbf{h}_i^L$ as high-dimensional representations $\mathbf{r}$ of each amino acid:

$$\mathbf{r} = [\mathbf{r}_1, \cdots, \mathbf{r}_n], \quad \mathbf{r}_i = \mathbf{h}_i^L. \tag{5}$$

We adopt an additional projection layer to obtain smaller d_z-dimensional representations $\mathbf{z}$ of each amino acid with a linear transformation:

$$\mathbf{z} = [\mathbf{z}_1, \cdots, \mathbf{z}_n], \quad \mathbf{z}_i = \mathrm{Proj}(\mathbf{r}_i). \tag{6}$$

During pre-training, in order to reduce computational complexity, we use $\mathbf{r}$ and $\mathbf{z}$ for the MLM and SFP tasks, respectively. During fine-tuning, we can either use $\mathbf{r}$ or $\mathbf{z}$, depending on which performs the best on the development set or based on computational constraints.

In this work, we primarily use two model sizes while fixing the input embedding dimension d_e and the projection dimension d_z as 21 and 100, respectively:

- **PLUS-RNN$_{\text{BASE}}$** : $L = 3$, $d_h = 512$, # of Parameters = 15M
- **PLUS-RNN$_{\text{LARGE}}$**: $L = 3$, $d_h = 1024$, # of Parameters = 59M

The former hyperparameters are chosen to match the BiRNN in P-ELMo. However, since P-ELMo also uses the forward and reverse RNNs, PLUS-RNN$_{\text{BASE}}$ has less than half number of parameters of P-ELMo (32M).

3.3 Pre-training procedure

Now, we explain the pre-training procedure of PLUS (Figure 1). In contrast to the previous approaches, it learns bidirectional representations based on two pre-training tasks, *i.e.*, MLM and SFP, designed to assimilate global structural information. For the complete pre-training (PT) loss, we use λ_{PT} to control their relative importance. To the best of our knowledge, this is also the first work to pre-train a BiRNN with MLM.

[1] 20 proteinogenic and 1 unspecified amino acids

3.3.1 Task #1: Masked Language Modeling (MLM)

For the MLM task, we generally follow the procedures used in BERT. Given a protein sequence $\mathbf{x}$, we randomly select 15% of the input amino acids. Then, for each selected amino acid $\mathbf{x}_i$, we perform one of the following random masking actions. For 80% of the time, we replace $\mathbf{x}_i$ with the token denoting the unspecified amino acid. For 10% of the time, we randomly replace $\mathbf{x}_i$ with one of the 20 proteinogenic amino acids. Finally, for the left 10%, we keep $\mathbf{x}_i$ intact. The purpose of the last one is to bias the representations towards the true amino acids.

Given a masked protein sequence $\hat{\mathbf{x}}$, PLUS-RNN produces bidirectional representations and the MLM decoder computes log probabilities for $\widetilde{\mathbf{x}}$ over 20 amino acid types. The MLM task trains the model to maximize the probabilities corresponding to the masked ones. As PLUS-RNN is asked to predict randomly masked amino acids given their contexts, the MLM task enables the model to learn bidirectional contextual representations throughout the entire protein sequence.

3.3.2 Task #2: Same Family Prediction (SFP)

Considering that additional pre-training tasks are often the key to further improve the quality of representations, we devise a complementary protein-specific pre-training task. The SFP task leverages computationally clustered weak family labels from the Pfam dataset. It trains a model to predict whether a given protein pair belongs to a same protein family. Despite its simplicity, we empirically show that the SFP complements the MLM and helps capturing global structural information of proteins.

In order to pre-train PLUS-RNN with the SFP pre-training task, we sample two protein sequences $\mathbf{x}^1$ and $\mathbf{x}^2$ from the Pfam dataset. For 50% of the time, the two sequences are sampled from a same protein family. For the other 50%, they are randomly sampled from different protein families. Note that, in contrast to BERT pre-training, we do not need to consider the lengths of the input sequences during the sampling process, since we use the BiRNN instead of the TFM.

PLUS-RNN transforms a protein pair into sequences of representations $\mathbf{z}^1 = [\mathbf{z}_1^1, \cdots, \mathbf{z}_{n_1}^1]$ and $\mathbf{z}^2 = [\mathbf{z}_1^2, \cdots, \mathbf{z}_{n_2}^2]$. Then, we use soft-align comparison (Bepler and Berger, 2019) to compute their similarity score $\hat{c}$ as a negative weighted sum of $l1$-distances between every $\mathbf{z}_i^1$ and $\mathbf{z}_j^2$ pair:

$$\hat{c} = -\frac{1}{C} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \omega_{ij} \left\| \mathbf{z}_i^1 - \mathbf{z}_j^2 \right\|_1, \quad C = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \omega_{ij}, \tag{7}$$

where the weight ω_{ij} of each $l1$-distance is computed by

$$\begin{aligned}
\omega_{ij} &= 1 - (1 - \alpha_{ij})(1 - \beta_{ij}), \\
\alpha_{ij} &= \frac{\exp(-\left\| \mathbf{z}_i^1 - \mathbf{z}_j^2 \right\|_1)}{\sum_{k=1}^{n_2} \exp(-\left\| \mathbf{z}_i^1 - \mathbf{z}_k^2 \right\|_1)}, \\
\beta_{ij} &= \frac{\exp(-\left\| \mathbf{z}_i^1 - \mathbf{z}_j^2 \right\|_1)}{\sum_{k=1}^{n_1} \exp(-\left\| \mathbf{z}_k^1 - \mathbf{z}_j^2 \right\|_1)}.
\end{aligned} \tag{8}$$

Intuitively, we can understand the soft-align comparison as computing an *expected alignment score*, where the expectations are over all possible alignments. We suppose that the smaller the distance between representations are, the more likely the pair of amino acids will be aligned. Then, we can consider α_{ij} as a probability that $\mathbf{z}_i^1$ is aligned to $\mathbf{z}_j^2$ considering all the amino acids from $\mathbf{z}^2$ (vice versa for β_{ij}). As a result, $\hat{c}$ is the expected alignment score over all possible alignments with probabilities ω_{ij}. Note that the negative signs are for converting distances into scores, and thus, a higher value of $\hat{c}$ indicates the pair of protein sequences is structurally more similar.

Given the similarity score, the output layer finally computes a probability that the pair belongs to a same protein family. The SFP task trains the model to minimize cross-entropy loss between the true label and the predicted probability. As PLUS-RNN is asked to produce higher similarity scores for proteins from the same families, the SFP task enables the model to better assimilate global structural information.

3.4 Fine-tuning procedure

The fine-tuning procedure of PLUS-RNN follows the conventional usage of BiRNN-based prediction models. For each downstream task, we only add one hidden and one output layers on top of the pre-trained model. Then, all the parameters are fine-tuned with task-specific datasets and loss functions. For the complete fine-tuning (FT) loss, we use λ_{FT} to control the relative importance of classification and regularization losses.

For tasks involving a protein pair, we use the same computations used in the SFP pre-training task. Specifically, we only replace the SFP output layer with a new output layer. For single protein-level tasks, we adopt an additional attention layer to aggregate variable-length representations into a single vector (Bahdanau *et al.*, 2014). Then, the aggregated vector is fed into hidden and output layers. For amino-acid-level tasks, representations of each amino acid are fed into hidden and output layers.

4 Experiments

All the models are implemented in PyTorch (Paszke *et al.*, 2017) and trained on either NVIDIA V100 or P40 GPUs. Due to the space limitations, please refer to the Appendix A.1 for details on pre-training and fine-tuning procedures. The data and codes used for the experiments are available at our repository. In the following, we will explain compared baselines, pre-training results, and fine-tuning results on seven benchmark tasks.

4.1 Baselines

For comparative evaluations, we use several baselines. First, in all of the seven protein biology tasks, we provide two alternative pre-training method benchmarks, *i.e.*, P-ELMo and PLUS-TFM. The former shares similar model architecture with PLUS-RNN$_{\text{BASE}}$, so it can show the effectiveness of the pre-training scheme. The latter is also pre-trained with PLUS, so it can show the effectiveness of BiRNN compared to TFM for protein sequences. The model architecture of PLUS-TFM is analogous to BERT$_{\text{BASE}}$ model consisting of 110M parameters. Due to its huge computational burden scaling quadratically with the input sequence length, following the procedures used in BERT, we pre-train PLUS-TFM only using the protein pairs shorter than 512 amino acids.

Second, for the TAPE tasks (Stability, Fluorescence, and SecStr), we also provide their pre-training method benchmarks: P-ELMo, UniRep, TAPE-TFM, TAPE-RNN, and TAPE-ResNet. We note that these comparisons are in their favor, since they were pre-trained with more than twice the number of protein sequences (32,207,059 sequences from Pfam release 32.0). The TAPE baselines can further demonstrate that PLUS-RNN outperforms various pre-training SOTA methods.

Finally, we benchmark PLUS-RNN against task-specific SOTA models without pre-training. They include deep learning, HMMs, or alignment based models exploiting additional features varying for each task. The additional features include scoring matrices with evolutionary information, HMM transition probabilities, and structural traits. Furthermore, if no previous deep learning based model exists for a given task, we provide RNN$_{\text{BASE}}$ and RNN$_{\text{LARGE}}$ models without pre-training instead. The comparison with the task-specific SOTA models can show in which type of tasks the pre-training scheme is most effective and help us understand its current limitations.

Table 1. Results on pre-training tasks

Method	(M)LM (acc)	SFP (acc)
PLUS-TFM	0.37	**0.98**
PLUS-RNN$_{\text{BASE}}$	0.33	0.96
PLUS-RNN$_{\text{LARGE}}$	0.37	0.97
P-ELMo*	0.29	-
P-ELMo†	0.28	-
UniRep†	0.32	-
TAPE-TFM†	**0.45**	-
TAPE-RNN†	0.40	-
TAPE-ResNet†	0.41	-

* Our experiments (Pfam 27.0). † Excerpted from TAPE (Pfam 32.0).

4.2 Pre-training results

Table 1 shows test accuracies on the MLM and SFP pre-training tasks. Only the models pre-trained with PLUS are evaluated for the SFP task. We should be careful for comparing the results from our experiments and TAPE, since they used different test datasets. Nonetheless, we can still indirectly compare them considering the following: (1) The test datasets are both randomly sampled proteins from different versions of Pfam dataset (27.0 for PLUS and 32.0 for TAPE). (2) P-ELMo was evaluated in both datasets and showed similar LM accuracies. It indicates that the difference between the two datasets is negligible.

We can see that some models have lower LM accuracies than the others. However, the lower LM capability does not exactly correspond to performance in the fine-tuning tasks. This discrepancy has been previously observed in TAPE, and it can be also observed in the following sections. In terms of SFP, all the models pre-trained with PLUS show high accuracies. This is because it could be a quite easy task compared to the LM. Since the Pfam families are constructed based only on the sequence similarities, a pair of analogous sequences would probably be from a same family. Despite its simplicity, we empirically show that the SFP complements the MLM by encouraging the models to compare protein representations during the pre-training.

4.3 Fine-tuning results

4.3.1 Benchmark tasks

We evaluate PLUS on seven protein biology tasks. We note and acknowledge that the benchmark datasets were curated, pre-processed, and split by the cited corresponding papers. Due to the space limitations, we only provide concise task definitions and evaluation metrics in the main manuscript. Please refer to the Appendix A.2 for details on data and compared models for each task.

Homology Homology is a protein-pair-level classification task (Fox *et al.*, 2013). The goal is to classify structural similarity level of a protein pair into *Family*, *SuperFamily*, *Fold*, *Class*, and *None*. We report accuracy of predicted similarity level and Spearman correlation ρ between predicted similarity scores and true similarity levels. Furthermore, we also provide area under the precision recall curve (AUPR) at each similarity level.

Solubility Solubility is a protein-level classification task (Khurana *et al.*, 2018). The goal is to predict whether a protein is *soluble* or *insoluble*. We report accuracy for this task.

Localization Localization is a protein-level classification task (Armenteros *et al.*, 2017). The goal is to classify a protein into ten subcellular locations. We report accuracy for this task.

Stability Stability is a protein-level regression task (Rocklin *et al.*, 2017). The goal is to predict a real-valued proxy for the intrinsic stability. This task is from TAPE and we report Spearman correlation ρ.

Fluorescence Fluorescence is a protein-level regression task (Sarkisyan *et al.*, 2016). The goal is to predict a log-fluorescence intensity. This task is from TAPE and we report Spearman correlation ρ.

Table 2. Summarized results on protein biology benchmark tasks

	Protein(-pair)-level Classification			Protein-level Regression		Amino-acid-level Classification	
Method	Homology (acc)	Solubility (acc)	Localization (acc)	Stability (ρ)	Fluorescence (ρ)	SecStr (acc8)	Transmembrane (acc)
PLUS-TFM	0.96	**0.72**	0.69	0.76	0.63	0.59	0.82
PLUS-RNN$_{\text{BASE}}$	0.96	0.70	0.69	**0.77**	0.67	0.61	**0.89**
PLUS-RNN$_{\text{LARGE}}$	**0.97**	0.71	**0.70**	**0.77**	**0.68**	**0.62**	**0.89**
Pre-training SOTA	0.95	0.64	0.54	0.73	**0.68**	0.61	0.78
Task-specific SOTA	0.93	0.77	0.78	0.73	0.67	0.72	0.80

For each task, the best pre-training method is in **bold**. It is **bold and underlined** if it is the best including task-specific SOTA.

Table 3. Detailed Homology prediction results

	Overall		Per-Level AUPR			
Method	acc	ρ	Class	Fold	Superfamily	Family
PLUS-TFM	0.96	**0.70**	0.94	0.91	0.95	**0.67**
PLUS-RNN$_{\text{BASE}}$	0.96	0.69	0.94	0.90	0.94	0.66
PLUS-RNN$_{\text{LARGE}}$	**0.97**	**0.70**	**0.95**	**0.92**	**0.96**	0.66
P-ELMo*	0.95	0.69	0.90	0.88	0.94	0.65
P-ELMo†	0.95	0.69	0.91	0.90	0.95	0.65
NW-align†	0.78	0.22	0.31	0.41	0.58	0.53
HHalign†	0.79	0.23	0.40	0.62	0.86	0.52
TMalign†	0.81	0.37	0.55	0.85	0.83	0.57
RNN$_{\text{BASE}}$	0.93	0.66	0.86	0.80	0.89	0.62
RNN$_{\text{LARGE}}$	0.83	0.52	0.66	0.46	0.52	0.39

* Results from our implementation. † Excerpted from P-ELMo.

Table 4. Detailed SecStr prediction results

	CB513		CASP12		TS115	
Method	acc8	acc3	acc8	acc3	acc8	acc3
PLUS-TFM	0.59	0.73	0.57	0.71	0.65	0.77
PLUS-RNN$_{\text{BASE}}$	0.61	0.75	0.60	0.72	0.66	0.78
PLUS-RNN$_{\text{LARGE}}$	**0.62**	**0.77**	**0.60**	**0.73**	**0.68**	**0.79**
P-ELMo*	0.61	**0.77**	0.54	0.68	0.63	0.76
P-ELMo†	0.58	0.73	0.57	0.70	0.65	0.76
UniRep†	0.57	0.73	0.59	0.72	0.63	0.77
TAPE-TFM†	0.59	0.73	0.59	0.71	0.64	0.77
TAPE-RNN†	0.59	0.75	0.57	0.70	0.66	0.78
TAPE-ResNet†	0.58	0.75	0.58	0.72	0.64	0.78
RaptorX‡	0.71	0.83	0.66	0.79	0.72	0.82
NetSurfP-2.0‡	0.72	0.85	0.70	0.82	0.75	0.86

* Results from our implementation. † Excerpted from TAPE.
‡ Excerpted from (Klausen *et al.*, 2019).

Secondary structure (SecStr) SecStr is an amino-acid-level classification task (Klausen *et al.*, 2019). The goal is to classify each amino acid into eight or three classes describing its local structure. This task is from TAPE and we report both three-way and eight-way classification accuracies (acc8/acc3) for this task.

Transmembrane Transmembrane is an amino-acid-level classification task (Tsirigos *et al.*, 2015). The goal is to detect segments of an amino acid sequence which cross the cell membrane. We report accuracy for this task.

4.3.2 Results summary

Table 2 presents summarized results for the seven benchmark tasks. To be concise, we show SOTA results from two categories: previous pre-training models (*i.e.*, P-ELMo, UniRep, TAPE-TFM, TAPE-RNN, and TAPE-ResNet) and task-specific models without pre-training. Detailed results for Homology and SecStr are provided in the following subsections. Please refer to the Appendix A.2 for detailed results for the other tasks.

We can see that PLUS-RNN$_{\text{LARGE}}$ model outperforms the pre-training SOTA models on six out of seven protein biology benchmark tasks. Considering that some pre-training methods showed higher LM capabilities, we can speculate that the performance improvements are contributed to the protein-specific SFP pre-training task. In the ablation studies, we further explain the relative importance of each aspect of PLUS-RNN. Although PLUS-TFM has almost twice as many as parameters than PLUS-RNN$_{\text{LARGE}}$ (110M vs. 59M), it shows inferior performances in most tasks. We suppose that this is due to its disregard of *locality bias*, which could be particularly problematic for protein biology.

Now, we compare PLUS-RNN$_{\text{LARGE}}$ with task-specific SOTA models. While the former performs significantly better on some tasks, it still lags far behind for the others. The results demonstrate that tailored models with additional features provide powerful advantages which still could not be learned from the pre-training. One classic example is position specific scoring matrices (PSSMs) generated from multiple sequence alignments. We conjecture that simultaneous observation of multiple proteins during the alignment could facilitate capturing evolutionary information within the proteins. In contrast, the current LM-based pre-training methods use millions of proteins but still exploit each one individually. The relatively small performance improvement by PLUS could also be explained by that the SFP task still only utilizes pairwise protein information. We expect exploiting multiple proteins during the pre-training might be the key to push performance past the task-specific SOTA models.

4.3.3 Homology and SecStr results

For further analyses, we present detailed evaluation results for Homology and SecStr tasks. We chose the two tasks because they are representative protein biology tasks relevant to global and local structures, respectively. The improved results of the former can lead to discovery of new enzymes and antibiotic resistant genes (Tavares *et al.*, 2013). The latter is important for understanding the function of proteins for those evolutionary structural information are not available (Klausen *et al.*, 2019).

The detailed Homology prediction results are presented in Table 3. The results show that PLUS-RNN$_{\text{LARGE}}$ outperforms both P-ELMo and task-specific models. In contrast to the RNN$_{\text{LARGE}}$ which shows overfitting due to the limited labeled training data, PLUS pre-training enables us to take advantage of the large model architecture. The correlation differences among PLUS-RNN$_{\text{LARGE}}$ (0.697), PLUS-RNN$_{\text{BASE}}$ (0.693), and P-ELMo (0.685) are small but statistically significant with p-values less than 10^{-15} (Steiger, 1980). The per-level AUPR results help us further look into which level of structural information is more captured by the proposed pre-training. The largest performance improvement of PLUS comes at the higher *Class* level rather than the lower *Family* level. It indicates that even though the Pfam family labels tend to be structurally correlated with the Homology task *Family* levels (Creighton, 1993), it is not the decisive factor for the performance improvement. Instead, PLUS pre-training incorporates the weak structural information and facilitates inferring higher-level global structure similarities.

Table 5. Ablation studies on Homology and SecStr tasks

	λ_{PT}		λ_{FT}		Homology						SecStr	
Method	MLM	SFP	MLM	CLS	acc	ρ	Class	Fold	Superfamily	Family	acc8	acc3
PLUS-RNN$_{\text{BASE}}$	0.7	0.3	0.3	0.7	**0.96**	**0.70**	**0.95**	**0.91**	**0.96**	**0.72**	**0.66**	**0.78**
RNN$_{\text{BASE}}$	-	-	0.3	0.7	0.93	0.67	0.88	0.81	0.92	0.68	0.61	0.73
(PT-A)	-	1.0	0.3	0.7	0.94	0.68	0.91	0.85	0.93	0.70	0.62	0.74
(PT-B)	0.5	0.5	0.3	0.7	**0.96**	0.69	**0.95**	**0.91**	0.95	0.70	0.66	0.77
(PT-C)	1.0	-	0.3	0.7	**0.96**	0.69	0.93	0.89	0.95	0.70	0.65	0.77
(FT-A)	0.7	0.3	-	1.0	0.94	0.68	0.91	0.85	0.93	0.70	0.65	0.77
(FT-B)	0.7	0.3	0.5	0.5	**0.96**	0.69	**0.95**	**0.91**	0.95	0.70	**0.66**	**0.78**

Note: We use the development sets for the ablation studies.

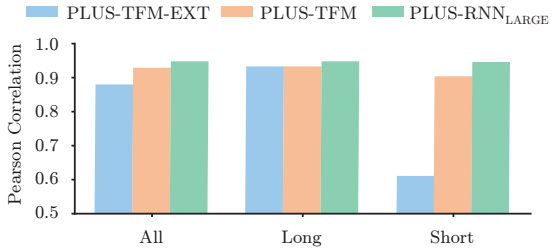

Fig. 2. Homology prediction results for different lengths.

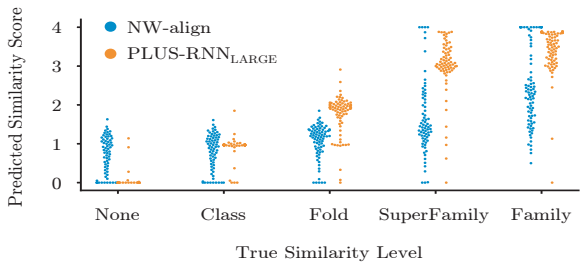

Fig. 3. Scatter plot of predicted similarity scores and true similarity levels.

The detailed SecStr prediction results are presented in Table 4. CB513, CASP12, and TS115 denote SecStr test datasets. Again, the results show that PLUS-RNN$_{\text{LARGE}}$ consistently outperforms all the other pre-training SOTA methods. It demonstrates that the SFP task complements the LM task during the pre-training and helps learning improved structurally contextualized representations. On the other hand, PLUS-RNN$_{\text{LARGE}}$ still lags far behind task-specific SOTA models using alignment-based features. We suppose it can be explained by the following two reasons. First, as previously stated, PLUS only utilizes pairwise information rather than multiple proteins simultaneously during the pre-training. Second, the SFP task requires understanding global structures, and local structures are relatively negligible. Therefore, we believe devising an additional pre-training task relevant to local structural information would be able to improve the performance on the SecStr task.

4.4 Ablation studies

In the following, we show results from various ablation studies on the Homology and SecStr tasks to better understand the strengths and each aspect of the PLUS framework. We use PLUS-RNN$_{\text{BASE}}$ as the baseline model unless explicitly stated otherwise. Note that we use the development sets for the ablation studies.

We explore the effect of using different λ_{PT} controlling the relative importance of the MLM and SFP pre-training tasks (Table 5). The results clearly show that pre-training is always helpful. Between the two pre-training tasks, removing the MLM task hurts the prediction performance more than removing the other. Nonetheless, the SFP task in addition to the MLM task consistently improves the prediction performance in all structural levels. This coincides with the expected results that the MLM task plays the primary role, and the SFP task complements the former by encouraging the models to compare pairwise protein representations.

During the fine-tuning, we can also simultaneously train a model for the MLM task as well as the classification (CLS) task. We explore the effect of using different λ_{FT} controlling the relative importance of them (Table 5). The results show that the adoption of MLM task consistently improves the prediction performance for both Homology and SecStr benchmark tasks. We suppose that the MLM task serves as a form of regularization and improves the generalization performance of the models.

Finally, we compare the Homology prediction performances of PLUS-TFM and PLUS-RNN$_{\text{LARGE}}$ for protein pairs of different lengths (Figure 2). Since PLUS-TFM was pre-trained only using protein pairs shorter than 512 amino acids, we denote *Long* for protein pairs longer than 512 amino acids and *Short* otherwise. Then, we evaluate PLUS-TFM for the *Long* protein pairs in two ways: (1) We simply use the protein pairs as they are. (2) We truncate them to 512 amino acids. The former is denoted as PLUS-TFM-EXT (as in extended) and the latter is denoted as PLUS-TFM.

PLUS-RNN$_{\text{LARGE}}$ consistently provides competitive performances regardless of the protein lengths. On the other hand, PLUS-TFM-EXT deteriorates for the *Long* protein pairs, and PLUS-TFM shows less performance degradation. The results show the limitation of TFM models using the limited context size of 512 amino acids. Although the number of *Long* protein pairs is relatively small (13.4%) in the Homology development dataset, it is indispensable to deal with long protein sequences for analyzing complex proteins that are found in nature. Since this is due to the computational burden of TFM scaling quadratically with the input sequence length, we expect recently proposed adaptive attention span (Sukhbaatar *et al.*, 2019) may be able to help improve PLUS-TFM.

4.5 Qualitative analyses

To better understand the strengths of PLUS-RNN, we provide its qualitative analyses. We use the Homology task and interpret how the learned protein representations help inferring the global structural similarities of proteins.

In order to compare two proteins, PLUS-RNN uses soft-align to compute their similarity score $\hat{c}$ (Equation 7). Even though there is one more computation by the output layer for the Homology prediction output, we can use the similarity scores to interpret PLUS-RNN. Note that using the penultimate layer for the model interpretation is a widely adopted approach in the machine learning community (Zintgraf *et al.*, 2017).

Figure 3 shows the scatter plot of predicted similarity scores and true similarity levels. For comparison, we also show the NW-align results with the BLOSUM62 scoring matrix (Eddy, 2004). The plot shows that NW-align often produces low similarity scores for protein pairs from the same *family*. This is because of high sequence-level variations, resulting in dissimilar sequences having similar structures. In contrast, most of the similarity scores of protein pairs from the same *family* have a high value.

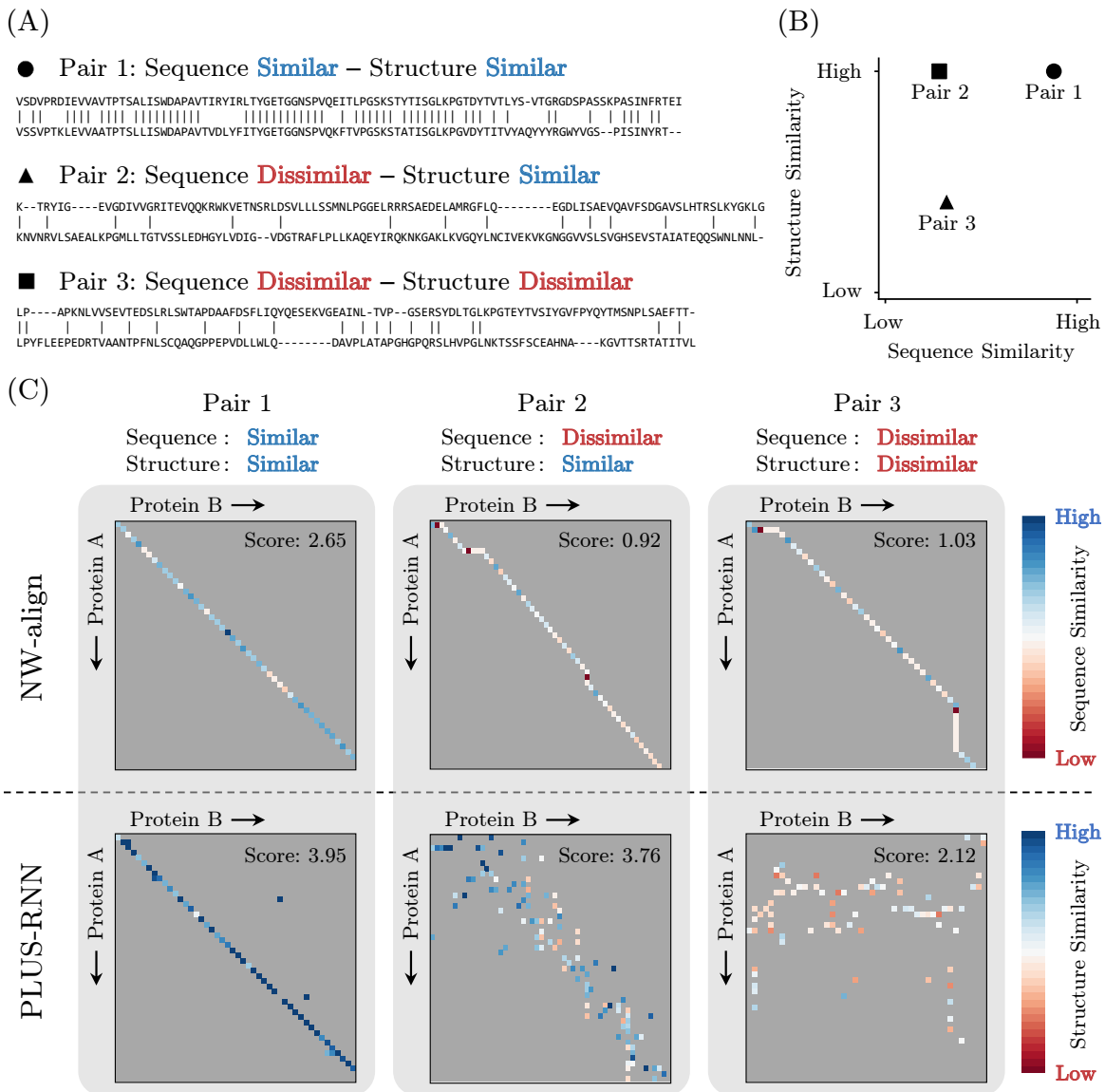

Fig. 4. Homology interpretation (A) We look into three types of protein pairs: a sequence similar - structure similar pair, a sequence dissimilar - structure dissimilar pair, and a sequence dissimilar - structure dissimilar pair. (B) The sequence and structure similarities of each pair are defined by NW-align scores and Homology dataset labels, respectively. (C) The heatmaps of NW-align of raw amino acids and soft-alignment of PLUS-RNN representations for the three pairs. Due to the space limitations, we only show the top left quadrant of the heatmaps.

Furthermore, we look into three types of protein pairs: (1) a sequence similar - structure similar pair, (2) a sequence dissimilar - structure dissimilar pair, and (3) a sequence dissimilar - structure dissimilar pair (Figure 4(A) and (B)). Note that sequence similar - structure dissimilar pair does not exist in the Homology datasets. The sequence and structure similarities are defined by NW-align scores and Homology dataset labels, respectively. The pairs having similar structures are chosen from the same *family*, and those having dissimilar structures are chosen from the same *fold*. Figure 4(C) shows the heatmaps of NW-align of raw amino acids and soft-alignment of PLUS-RNN representations (ω_{ij} in equation 7) for the three pairs. Due to the space limitations, we only show the top left quadrant of the heatmaps. Each cell in the heatmap indicates the corresponding amino acid pairs from protein A and B. Blue denotes high sequence similarity in NW-align and high structure similarity in PLUS-RNN.

First, we compare the pairs having similar structures (the first and second columns in Figure 4(C)). The heatmaps show that NW-align successfully aligns the similar sequence pair with the score of 2.65. However, it fails for the dissimilar sequence pair with the score of 0.92. It supports that comparing the raw sequence similarities cannot identify the correct structure similarities. On the other hand, soft-alignment of PLUS-RNN representations are successful for both similar and dissimilar sequences with the scores of 3.95 and 3.76. Next, we compare the second and the third pairs. Although only the second pair has similar structures, NW-align fails for both and even gives higher score of 1.03 to the third pair. In contrast, regardless of the sequence similarities, the soft-alignment of PLUS-RNN representations correctly degenerates only for the third pair with dissimilar structures with the score of 2.12. Therefore, the interpretation results verify that the learned representations from PLUS-RNN are structurally contextualized and performs better for inferring the global structure similarities.

5 Concluding Remarks

In this work, we presented PLUS, a novel pre-training scheme for bidirectional protein sequence representations. Consisting of the MLM and the protein-specific SFP pre-training tasks, it can better capture structural information contained within the proteins. PLUS can be used to pre-train various model architectures. In this work, considering the sequential modeling capability and computational complexity, we mainly used PLUS-RNN. It advances the previous SOTA pre-training methods on six out of seven protein biology tasks. Furthermore, to better understand its strengths, we also provided the results from our ablation studies and qualitative interpretation analyses.

We are excited about the future of PLUS. We expect the gap between the number of unlabeled and labeled proteins will continue to exponentially grow, and the pre-training method will play even larger roles. Based on the strengths and weaknesses of PLUS, we plan to extend the work in several directions. First, considering that it is especially powerful for inferring global structural information, we are also interested in more exquisite prediction of protein structures (Kryshtafovych *et al.*, 2019). Second, although the pre-training helps, it still lags behind task-specific models for some tasks. We suppose this is because of its weaknesses on learning local structural and evolutionary information. We believe there is still huge room for improvements and exploiting multiple proteins during the pre-training, likewise in the alignment, could be the key (Poplin *et al.*, 2018).

Acknowledgements

This work was supported by the Brain Korea 21 Plus Project in 2020.

References

Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M., and Church, G. M. (2019). Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, **16**(12), 1315–1322.

AlQuraishi, M. (2019). Alphafold at casp13. *Bioinformatics*, **35**(22), 4862–4865.

Armenteros, J. J. A., Sønderby, C. K., Sønderby, S. K., Nielsen, H., and Winther, O. (2017). Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, **33**(21), 3387–3395.

Asgari, E. and Mofrad, M. R. (2015). Continuous distributed representation of biological sequences for deep proteomics and genomics. *PloS one*, **10**(11), e0141287.

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bailey, T. L., Williams, N., Misleh, C., and Li, W. W. (2006). Meme: discovering and analyzing dna and protein sequence motifs. *Nucleic acids research*, **34**(suppl_2), W369–W373.

Bepler, T. and Berger, B. (2019). Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations*.

Berg, J. M., Tymoczko, J. L., and Stryer, L. (2006). Biochemistry. 5th. *New York: WH Freeman*, **38**(894), 76.

Chapelle, O., Scholkopf, B., and Zien, A. (2009). Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]. *IEEE Transactions on Neural Networks*, **20**(3), 542–542.

Creighton, T. E. (1993). *Proteins: structures and molecular properties*. Macmillan.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eddy, S. R. (2004). Where did the blosum62 alignment score matrix come from? *Nature biotechnology*, **22**(8), 1035–1036.

Elofsson, A. and Sonnhammer, E. (1999). A comparison of sequence and structure protein domain families as a basis for structural genomics. *Bioinformatics (Oxford, England)*, **15**(6), 480–500.

Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., *et al.* (2014). Pfam: the protein families database. *Nucleic acids research*, **42**(D1), D222–D230.

Fox, N. K., Brenner, S. E., and Chandonia, J.-M. (2013). Scope: Structural classification of proteinsâĽ"extended, integrating scop and astral data and classification of new structures. *Nucleic acids research*, **42**(D1), D304–D309.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, **9**(8), 1735–1780.

Holm, L. and Sander, C. (1996). Mapping the protein universe. *Science*, **273**(5275), 595–602.

Khurana, S., Rawi, R., Kunji, K., Chuang, G.-Y., Bensmail, H., and Mall, R. (2018). Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, **34**(15), 2605–2613.

Klausen, M. S., Jespersen, M. C., Nielsen, H., Jensen, K. K., Jurtz, V. I., Soenderby, C. K., Sommer, M. O. A., Winther, O., Nielsen, M., Petersen, B., *et al.* (2019). Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, **87**(6), 520–527.

Krause, B., Lu, L., Murray, I., and Renals, S. (2016). Multiplicative lstm for sequence modelling. *arXiv preprint arXiv:1609.07959*.

Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K., and Moult, J. (2019). Critical assessment of methods of protein structure prediction (casp)âĽ"round xiii. *Proteins: Structure, Function, and Bioinformatics*, **87**(12), 1011–1020.

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. (2019). Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Min, S., Lee, B., and Yoon, S. (2017). Deep learning in bioinformatics. *Briefings in bioinformatics*, **18**(5), 851–869.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Poplin, R., Chang, P.-C., Alexander, D., Schwartz, S., Colthurst, T., Ku, A., Newburger, D., Dijamco, J., Nguyen, N., Afshar, P. T., *et al.* (2018). A universal snp and small-indel variant caller using deep neural networks. *Nature biotechnology*, **36**(10), 983–987.

Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., and Song, Y. S. (2019). Evaluating protein transfer learning with tape. In *Advances in neural information processing systems*.

Rocklin, G. J., Chidyausiku, T. M., Goreshnik, I., Ford, A., Houliston, S., Lemak, A., Carter, L., Ravichandran, R., Mulligan, V. K., Chevalier, A., *et al.* (2017). Global analysis of protein folding using massively parallel design, synthesis, and testing. *Science*, **357**(6347), 168–175.

Sarkisyan, K. S., Bolotin, D. A., Meer, M. V., Usmanova, D. R., Mishin, A. S., Sharonov, G. V., Ivankov, D. N., Bozhanova, N. G., Baranov, M. S., Soylemez, O., *et al.* (2016). Local fitness landscape of the green fluorescent protein. *Nature*, **533**(7603), 397–401.

Söding, J., Biegert, A., and Lupas, A. N. (2005). The hhpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, **33**(suppl_2), W244–W248.

Steiger, J. H. (1980). Tests for comparing elements of a correlation matrix. *Psychological bulletin*, **87**(2), 245.

Strodthoff, N., Wagner, P., Wenzel, M., and Samek, W. (2019). Udsmprot: Universal deep sequence models for protein classification. *bioRxiv*, page 704874.

Sukhbaatar, S., Grave, E., Bojanowski, P., and Joulin, A. (2019). Adaptive attention span in transformers. In *ACL*.

Tavares, L. S., Silva, C. d. S. F. d., Souza, V. C., Silva, V. L. d., Diniz, C. G., and Santos, M. D. O. (2013). Strategies and molecular tools to fight antimicrobial resistance: resistome, transcriptome, and antimicrobial peptides. *Frontiers in microbiology*, **4**, 412.

Tsirigos, K. D., Peters, C., Shu, N., Käll, L., and Elofsson, A. (2015). The topcons web server for consensus prediction of membrane protein topology and signal peptides. *Nucleic acids research*, **43**(W1), W401–W407.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yang, K. K., Wu, Z., Bedbrook, C. N., and Arnold, F. H. (2018). Learned protein embeddings for machine learning. *Bioinformatics*, **34**(15), 2642–2648.

Zintgraf, L. M., Cohen, T. S., Adel, T., and Welling, M. (2017). Visualizing deep neural network decisions: Prediction difference analysis. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.