

---

# Pre-training Protein Language Models with Label-Agnostic Binding Pairs Enhances Performance in Downstream Tasks

---

**Modestas Filipavicius \***

Department of Biosystems Science and Engineering  
ETH Zurich  
Zurich, 8004  
Switzerland  
mfilipav@ethz.ch

**Matteo Manica**

IBM Research Zurich  
Rueschlikon  
tte@zurich.ibm.com

**Joris Cadow**

IBM Research Zurich  
Rueschlikon  
dow@zurich.ibm.com

**Maria Rodriguez Martinez**

IBM Research Zurich  
Rueschlikon  
mrm@zurich.ibm.com

## Abstract

Life science practitioners are drowning in unlabeled protein sequences. Natural Language Processing (NLP) community has recently embraced self-supervised learning as a powerful approach to learn representations from unlabeled text, in large part due to the attention-based context-aware Transformer models. In a transfer learning fashion, expensive pre-trained universal embeddings can be rapidly fine-tuned to multiple downstream prediction tasks. In this work we present a modification to the RoBERTa model by inputting a mixture of binding and non-binding protein sequences (from STRING database) during pre-training with the Masked Language Modeling (MLM) objective. Next, we compress protein sequences by 64% with a Byte Pair Encoding (BPE) vocabulary consisting of 10K tokens, each 3-4 amino acids long. Finally, to expand the model input space to even larger proteins and multi-protein assemblies, we pre-train Longformer models that support 2,048 tokens. Our approach produces excellent fine-tuning results for protein-protein binding prediction, TCR-epitope binding prediction, cellular-localization and remote homology classification tasks. We suggest that the Transformer’s attention mechanism contributes to protein binding site discovery. Further work in token-level classification for secondary structure prediction is needed. Code available at: [https://github.ibm.com/PaccMann/paccmann\\_proteomics](https://github.ibm.com/PaccMann/paccmann_proteomics)

## 1 Introduction

Inferring protein properties from the primary amino acid sequence is of particular importance in the light of sequencing technology advances resulting in a vast number of proteins with unknown characteristics. UniProt database, as of 2020\_011 release, contains 180M protein sequences, and only 560K of them (or 0.31%) are labeled [11]. Those lucky few functionally and structurally characterized proteins belong to a highly exclusive group of families with immediate industrial and

---

\*Use footnote for providing further information about author (webpage, alternative address)—*not* for acknowledging funding agencies.

therapeutic potential, such as GFPs, viral Env proteins or more recently CAS nucleases. Therefore, most of the proteins stay in the dark matter of the proteome – especially those from Prokarya and viruses, intrinsically disordered or post-translationally modified proteins [37].

As early as 1970s Anfinsen proposed that the protein structure and functionality is encoded by its primary amino acid sequence [5]. A prevailing hypothesis today claims that under evolutionary forces, protein sequence space was sampled to form a closed set of structural and functional motifs, since there was a strong evolutionary pressure to reuse and recycle these components [4]. Thus, even the unlabeled raw protein sequences should be enough to implicitly understand the language of proteins.

Recently, self-supervised learning established itself as a powerful method for learning useful information from unlabeled sequences, especially attention-based Transformer models [43]. They work by first pre-training a large context-aware attention model on millions of unlabeled text lines with a proxy task such as predicting the next word in a sentence given all previous words [31, 32, 33] or predicting masked-out words from their context [12, 26]. In the second step, the model is fine-tuned specifically for each downstream task in a supervised manner with labeled datasets.

What if we treated protein sequences as sentences and individual (or several continuous) amino acids as words? Throughout 2019 and 2020 we saw a surge of such NLP-based techniques applied to protein sequences, which showed preliminary potential to extract useful biological information from massive unlabeled datasets [35, 34, 28, 41, 2, 7, 17, 14, 29].

### **Our contributions to transformer-based protein language modeling (LM):**

- Demonstrate that pre-training a RoBERTa language model, solely with the masked language modeling (MLM) objective, on a mixture of binding and random protein pairs results in superior downstream protein classification performance.
- Compresses the protein sequence space by 64% by expanding the 20 amino acid character vocabulary to 10K sub-word tokens by Byte-Pair Encoding (BPE) algorithm [40].
- Prepare new pre-training and protein-protein binding prediction datasets from STRING database [42].
- Pre-train a Longformer LM [6] with a maximum sequence length of up to 2,048 tokens, accommodating long protein sequences or multi-protein assemblies.

## **2 Related work**

UniRep was the first study to revolutionize the protein representation learning field [2]. In autoregressive manner a multiplicative LSTM was pre-trained on 22M Pfam [13] sequences to learn a 1,900-dimensional representation for each amino acid, which successfully predicted protein secondary structure. Next, Bepler Berger (2019) achieved SOTA results for secondary structure prediction by jointly pre-training bidirectional LSTM on protein sequence pairs, and supervising the model with the structure similarity prediction between those sequences and amino acid contact prediction within the sequence. Inspired by ULMFit’s [19] success in applying the "pre-train first then fine-tune" procedure to NLP, UDSMProt [41] pre-trained a first protein transfer learning model with a multi-layer LSTM.

More recent protein LM efforts picked up on a NLP trend of using context-dependant LM architectures: ELMo and BERT. ELMo [31] has been applied in P-ELMO [7], SeqVec [17], PLUS-RNN [28]. ELMo learns context-dependent representations by predicting the next token separately in forward and reverse directions with LSTM [18]. Later these two representations are combined, however they struggle when bidirectional context is needed. BERT [12] solved this problem by pre-training Transformer encoder units with masked language modeling (MLM) objective. This architecture was used for proteins by ESM [35], TAPE [34] and ProtTrans [14].

In parallel to our work, ProBERTa [29] was trained using BPE for token representation (vocabulary with 10,000 tokens). We coincidentally chose the same size for BPE vocabulary and trained with RoBERTa. However, our model is different in two key aspects. First, we pre-trained on much bigger datasets with at least 10M sequences, as opposed to 0.5M sequences from SwissProt. Second, our model was much bigger with 12 layers and 12 heads. Together, bigger models trained on larger

Hyperparameter	Pfam	String1(2)Seq	StringLongformer1(2)Seq	SwissProt
Number of Tokens	512	512	2048	512
Number of Layers	12	12	12	<b>6</b>
Hidden size	768			
FFN inner hidden size	3072			
Attention heads	12			
Attention head size	64			
Dropout	0.1			
Attention Dropout	0.1			
Warmup Steps	4K	20K	512	5K
Peak Learning Rate	1e-4	1e-4	7e-5	1e-4
Weight Decay	0.01			
Total Batch Size	128	128	16	128
Per GPU Batch Size	16	16	1	16
Number GPUs	4			
GA Steps	2	2	4	2
Tokens per Batch	65,536			

Table 1: Hyperparameters for pretraining Protein RoBERTa. If a hyperparameter showed only once, assume it is applied to all models.

datasets just for a few epochs achieve better performance than training shallow and narrow models on small datasets for many steps [25, 16].

In terms of accommodating long protein sequences, ProtBert [14] trained Transformer-XL and XLNet autoregressive models which support long sequences by re-using hidden states of the previous sub-sequences. Conversely, we use an auto-encoding language modeling which learns bidirectional contexts, as opposed to the auto-regressive approach above, which is better suited for sequence pair tasks like binding prediction.

### 3 Methods

#### 3.1 Pre-training

Our suggested pre-training architecture is a derivative of RoBERTa [26], and is depicted in **Figure 1**. For more information about the architecture, consult [43, 12, 26]. Masked LM pretraining involved masking out 15% of input tokens at random and predicting them. Unless specified, we opted for a deep ( $L = 12$  layers) and wide ( $H = 12$  attention heads) architecture. The remaining pre-training parameters are displayed in **Table 1**. AdamW was used as optimizer [27], with  $\epsilon = 1e-6$ ,  $\beta_1 = 0.9$ ,  $\beta_2 = 0.98$ , and linear weight decay. We used four Tesla P100 SXM2 GPUs with 16Gb memory, ensuring to put  $2^{16}$  tokens per effective batch, and mixed precision training (fp16) [1] helped to reduce memory consumption and speed-up training.  $Tokens\ per\ batch = Sequence\ length \times \#GPUs \times GPU\ batch\ size \times \#GA\ steps$ , for RoBERTa models:  $Tokens\ per\ batch = 512 \times 4 \times 16 \times 2 = 2^{16}$ . All of our protein language models are pre-trained and fine-tuned using RoBERTa architecture implemented in Transformers library [45] and PyTorch 1.3.1 [30].

To accommodate protein sequences of 2,048 tokens long, we used the **Longformer** architecture [6], a derivative RoBERTa which utilizes a predefined length attention window within which the expensive quadratic complexity self-attention is computed. We follow implementation from [6], with an attention window of 512 tokens. Depending on the task, some tokens are assigned global attention – such tokens attend to all sequence tokens, and all tokens attend to them. For instance, in sequence classification tasks the CLS token pools attention from all the other sequence tokens. Since pretraining from scratch is computationally expensive, we start with RoBERTa checkpoints from earlier experiments trained on up 512 token-long sequences. Initially training with the shorter sequences, and towards the end with expensive longer ones is a common practice [12, 6].

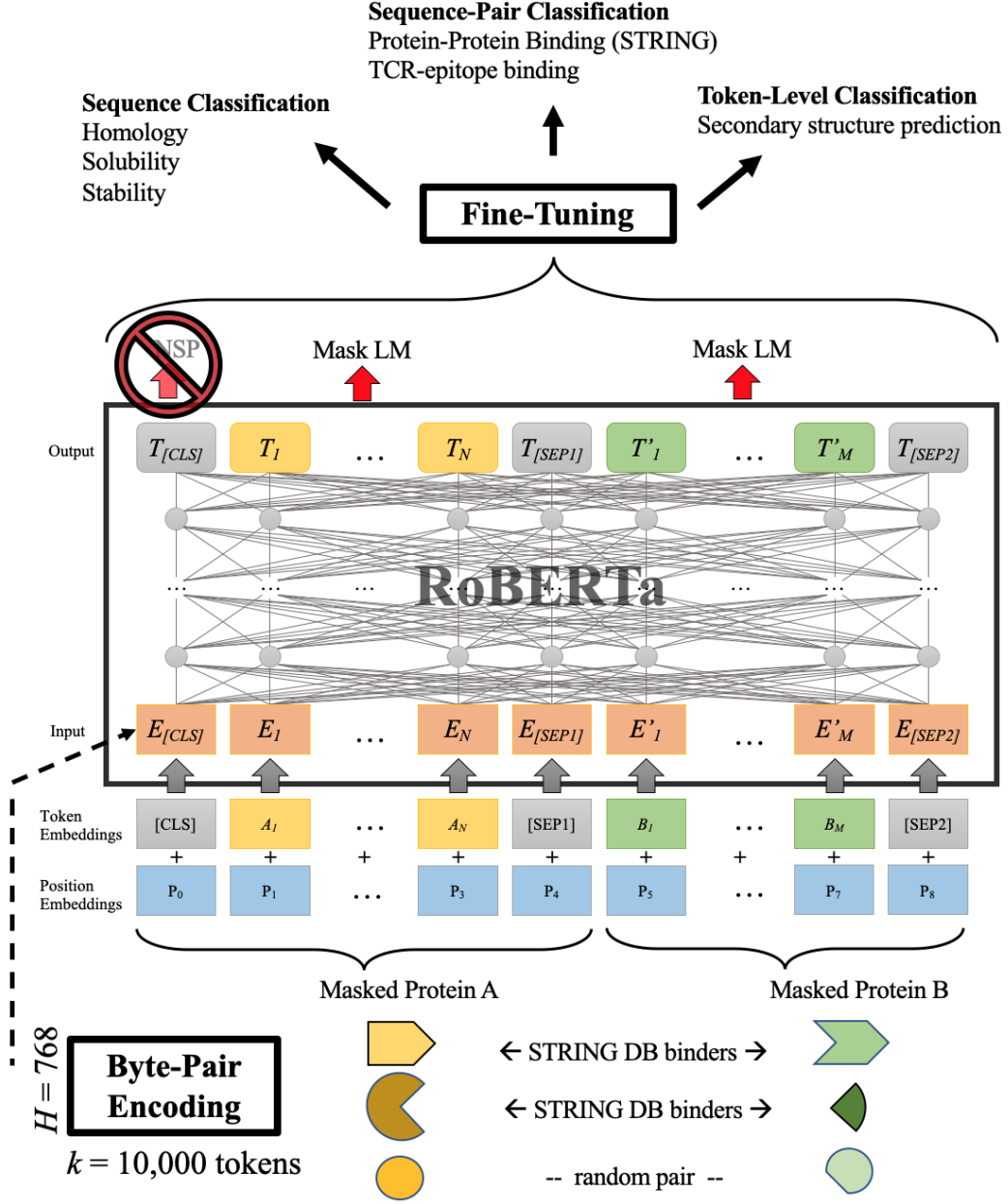


Figure 1: Proposed architecture is pre-trained by a mixture of binding and non-binding protein sequences, using only the MLM objective. Byte-pair encoding with a 10k token vocabulary enables inputting 64% longer protein sequences compared to character level embedding.  $E_i$  and  $T_i$  represent input and contextual embeddings for token  $i$ . CLS is a special token for classification-task output, while SEP separates two non-consecutive sequences.

### 3.2 Input sequence representation with Byte-Pair Encoding (BPE)

BPE [40] is a sub-word segmentation algorithm commonly used in Neural Machine Translation, and more recently in GPT2 [33] and RoBERTa [26] LMs. In a deterministic manner BPE creates a vocabulary of predefined size from subwords by merging the most frequently occurring subword pairs, until the desired vocabulary size is reached.

Datasets	Pfam	String	String2Seq	StringLF	StringLF2Seq	SwissProt
CE train	5.01	4.70	4.59	4.91	5.16	4.58
CE val	5.81	5.83	5.22	6.52	6.30	4.49
CE Pfam holdout	7.40	7.72	<b>6.41</b>	8.03	6.75	6.56
Train data size	31M	10M	5M	9.53M	4.76M	504K
Eval data size	100K	150K	150K	150K	150K	56K
Batch size	128	128	128	16	16	128
Train Epochs	2	3.4	6.5	0.030	0.070	45.3
Final LR	1.5e-5	3.8e-7	7.4e-8	6.6e-5	6.2e-5	5.5e-5
Training Steps	561K	300K	321K	66K	63K	178K
Round 1 steps	389.6K	144K	158K			
Round 2 steps	171.5K	155K	163K			

Table 2: Hyperparameters and language model pre-training results for different pre-training datasets. Cross Entropy (CE) loss is a natural log of perplexity. Round 2 for String1Seq and String2Seq conducted because LR was not decreasing fast enough, LR was  $9.84e-5$  and  $9.64e-5$  respectively. In round 2 the learning rate was linearly decreased to 0. SwissProt number of epochs was set to 100. Val perplexity evaluated on 56K sequences. Pfam for round one was trained with linearly decreasing learning schedule, but but for expected 100 epochs, only 1.6 of which were completed. Stage two continued training from a checkpoint but with 2 expected epochs, 0.4 of which were completed, with the final LR  $1.48e-5$ . Pfam holdout set consists of 44,311 sequences.

Tasks	Localization	Solubility	Homology	PPB	TCR	SSP
Reference	[3]	[21]	[9]	<i>ours and</i> [42]	<i>Anonymous</i>	[22]
Train	9,977	62,478	12,311	2,662,711	124,486	9,279
Dev	1,107	6,942	734	63,589	13,832	2316
Test	1,000	2,001	2,000	150,000	10K	581

Table 3: Fine-tuning datasets. PPB is Protein-Protein Binding, TCR is T-cell receptor and its epitope binding, SSP is secondary structure prediction.

### 3.3 Pre-training data

We pretrained on protein sequences from SwissProt [11], Pfam [13] and STRING [42] databases, for more information see **Table 2**. Most notably, we prepared two pretraining datasets which contain protein sequence pairs from STRING, half of them are strong binders (binding interaction score is at least 700, in 0-1,000 confidence scale), while the other half are random pairs. String2Seq model has two binding pairs that together are shorter than 512 BPE token limit, while StringLF2Seq model accommodates two sequences which together are up to 2,048 tokens-long.

### 3.4 Fine-tuning

Pre-training can be assessed with cross-entropy loss on MLM objective, but the ultimate assessment comes from the downstream performance on protein prediction tasks. We have attempted three types of tasks. First, single protein sequence classification, for which we plug in task-specific inputs into the pre-trained model, with a single modification of a supplementary output layer which takes in the representation from CLS to give class SoftMax probabilities. Similarly, in protein sequence pair classification the relationship between the two sequences separated by a SEP token is learned by the CLS token as well. Finally, token level classification, all tokens except CLS pass through an extra SoftMax layer to give a class probability for a token. This is equivalent to Named Entity Recognition (NER). See **Table 4** and **Appendix 5** for fine-tuning data description and hyperparameters. Parameter search was performed for learning rate  $\{1e-5, 3e-5, 5e-5\}$ , batch size  $\{8, 16, 32\}$  and learning rate weight decay  $\{0.05, 0.1\}$ .

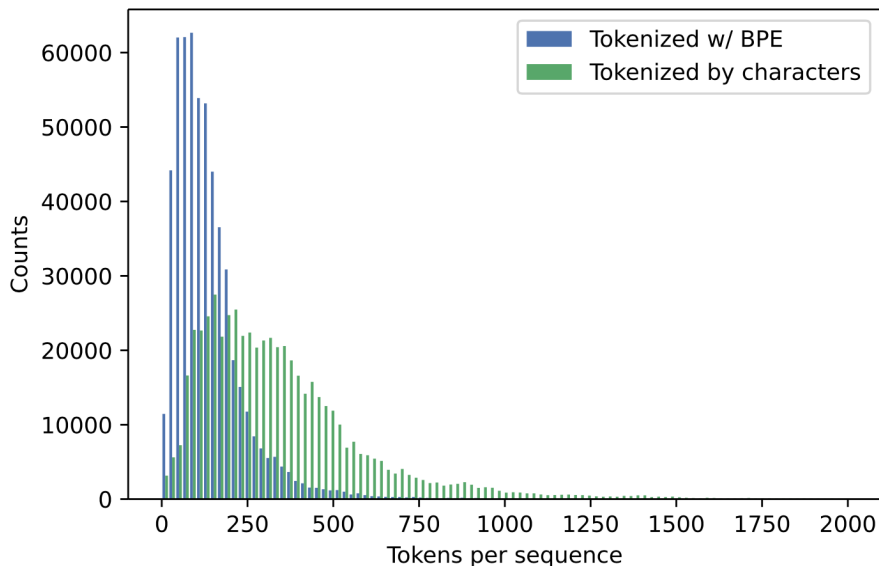


Figure 2: BPE vocabulary with 10,000 tokens captures 98.4% of SwissProt sequences within the 512 token constraint often used for BERT-like models compared to capturing 82% if tokenized by single characters.

## 4 Results

### 4.1 BPE allows efficient sequence compression for sequence classification tasks

BPE algorithm creates a predefined size vocabulary from subwords by merging the most frequently occurring subword pairs in a bottom-up fashion, until the vocabulary size is reached [40]. Amino acid tokens are listed in the order of decreasing value frequency, for an example see **Appendix Figure 4**. The most frequent two-letter tokens are "LL", "AA", "AL" and "VL", three letter – "ALL", "AAL", "LLL". These tokens feature some of the most common amino acids. Most importantly, as seen in **Figure 2**, BPE more efficiently compresses protein sequences compared to character-level encoding, with an average sequence length down from 360 to 130 tokens – more than 64% compression. As a result, only 1.58% of BPE-tokenized SwissProt sequences exceed the 512 token limit, often seen in BERT-like models, as opposed to a staggering 18.2% for regular character-level splitting. For a 1,000 token limit, the loss is reduced from 3.3% to 0.23%.

Why is the efficient compression so important? First off, most of the studies in the literature discard protein subsequence which exceeds the 500 or 1,000 amino acid limit, thus losing valuable information. Secondly, for sequence-pair classification tasks, such as PPB prediction, fitting both sequences within the model length constraint is of paramount importance. BPE tokenization for the first time enables us to present two long protein sequences to the Transformer models during the pre-training and fine-tuning stages.

Choosing the vocabulary size parameter  $k$  for BPE algorithm was a heuristic process. We have tested  $k = 100, 1000, 10000, 30000$ , and observed that at  $k = 10000$  BPE vocabulary mostly consists of 3 and 4-characters-long tokens, as seen in **Appendix Figure 5**. We postulated that such length distribution should accurately capture the frequent protein subsequence motifs, and be advantageous to the secondary structure prediction problem.

### 4.2 Pre-trained embeddings encode amino acid biochemical properties

We visualized the 768-dimensional embeddings for single characters corresponding to 21 amino acids 10K-long BPE vocabulary. T-SNE plot in **Appendix Figure 6** confirms that amino acids cluster according to their charge, hydrophilicity and size, in accordance with [2, 34, 35]. Note that we are still exploring how to visualize the 10K protein-sub-word vocabulary.

Tasks	Localization	Solubility	Homology	PPB	TCR	SSP
Reference	[3]	[21]	[9]	<i>ours and</i> [42]	<i>Anonymous</i>	[22]
Train	9,977	62,478	12,311	2,662,711	124,486	9,279
Dev	1,107	6,942	734	63,589	13,832	2316
Test	1,000	2,001	2,000	150,000	10K	581

Table 4: Fine-tuning datasets. PPB is Protein-Protein Binding, TCR is T-cell receptor and its epitope binding, SSP is secondary structure prediction.

Tasks	Localization	Solubility	Homology	PPB	TCR	SSP
Result	<b>0.793</b>	0.583	0.256	<b>0.983</b>	<b>0.729*</b>	0.669*
SOTA result	0.78	0.77	0.26	NA	0.70	0.791**
Classes	10	2	1,195	2	2	3
LR	5e-5	5e-5	5e-5	3e-5	5e-5	1e-5
Batch Size	16	8	16	32	32	8
Epochs	15	15	15	2	5	5
Optim steps	9,360	58,575	11,550	166K	19,455	5,800

Table 5: Fine-tuning results, reported in accuracy. All of the best results achieved with String2Seq model, unless specified by "\*". \*SwissProt; \*\*3-class secondary structure prediction, using SwissProt pretraining, SoTA result from 12-layer BERT from [35], their best 34 layer model had accuracy of 0.89. We ran our fine-tuned models against the test sets only once. Development runs with a unique set of parameters were not replicated. Hyperparameters for best performing models are shown.

### 4.3 MLM pre-training with protein binding pairs outperforms single sequences

We present our RoBERTa-like model pre-training results in **Table 2**. In short, the models are pre-trained across three datasets: SwissProt, Pfam and String, two input representations: single sequences (models Pfam, String, SwissProt) or binding sequence-pairs (models String2Seq, StringLF2Seq). Two architectures are pretrained: RoBERTa [26] and Longformer [6] which expect up to  $T = 512$  and  $T = 2,048$  tokens for model input, respectively.

To test a hypothesis that pre-training on a mixture of binding/non-binding protein sequence-pairs would result in a superior pre-training and downstream fine-tuning task performance (especially for PPB tasks), we have built a novel protein-protein binding dataset from high quality ( $> 700$ ) protein binding pairs found in STRING database from 5K organisms and 41M pairs. We selected 2.5M binding and 2.5M non-binding pairs to form our **String2Seq** dataset. The two binders in the sequence pair were separated by EOS and SEP tokens, which in our RoBERTa tokenization scheme share the same token - " $</s>$ ". **String** pretraining dataset consisted of all unique protein sequences found in String2Seq dataset.

As seen in **Table 2**, String2Seq model had a lower MLM CE loss than String on Pfam holdout dataset. We hypothesize that the pre-training with a mixture of sequences resulted in learning more accurate secondary and tertiary structure representations, which facilitate protein-protein binding, and by extension, are responsible for the local protein structure. Note, that learning happened exclusively through MLM objective, during AdamW optimization there was no gradient update from sequence-pair relationship information. That is, the model was agnostic whether the two sequences were randomly selected or were strong binders.

### 4.4 Sequence-pair pre-trained models outperform single sequence models for fine-tuning tasks

The ultimate test for learning protein language is to evaluate the language model on downstream task, as [34] and [41] demonstrated that low CE values for MLM task do not always correspond to excellent downstream task performance. We attempted 5 tasks, results for them are summarized in **Table 5**. Our most shocking finding is the apparent increase in fine-tuning performance after pre-training on 2

sequence pairs only with MLM, without explicit sequence-pair classification objective (like Next Sentence Prediction (NSP) in BERT or Sentence Order Prediction (SOP) in Albert [24]).

String2Seq model performed the best for all tasks except for the secondary structure prediction (SSP) and T-cell receptor target prediction (TCR). For **localization**, the second best model was String1Seq, with accuracy of 0.780. Our model has overtaken the previous result of 0.78 achieved by Bidirectional-LSTM network [3].

Our best model also performed well on the **remote homology** task that maps protein folds to 1,195 families. This task challenges the model to detect structural similarities among proteins with radically different evolutionary histories. Test sequences are from different protein families than those found in the train set. It is worth noting that the label distribution is extremely unequal, with top 6 most frequent class labels account for 22% of all data (See **Appendix Figure 7**). String1Seq performed at 0.224 accuracy. The previous best result from TAPE study [34] achieved 0.26 accuracy, with pre-trained and fine-tuned LSTM, while their BERT implementation was 0.05 lower.

Not surprisingly, our model did extremely well in **protein-protein binding** (PPB) task. In part, because it was pre-trained on some of the sequences found in the training set of the task. However, we took care to construct evolutionarily diverse validation and test datasets which do not contain any data used for pre-training the model.

As of 2020-10-01 we are not aware of other methods that propose such pre-training scheme. One study [28] explored using NSP-like pre-training objective that predicts belonging to the same Pfam family group (3,150 families), but we argue that this task is too easy and propose using a new String Binding Predictor objective (for further information see Future Directions Appendix 5).

Regarding the low performance on **SSP** task, we were able to test only SwissProt model, that we developed early as one of the baselines. Towards the end of this project we have experienced bugs in the testing script for our sequence annotation script.

#### 4.5 Visualizing attention patterns reveals specialized layers in protein LM embedding

Transformers are attention-driven models, and we can visualize the multi-head self-attention for the input sequence representation across the layers and heads [10, 36, 44]. With such information, we can explain if the model learns the syntax and semantics necessary to perform well on prediction tasks. Initial layers typically encode positional relations, middle – dependency relations, final - unique and global patterns.

For simplicity purposes, we visualize the layers and attention heads for the first alpha helix of TIM-barrel, a *de novo* designed four-fold symmetry protein [20] (see **Appendix Figure 8B**). As seen in the the two far-left panels on **Figure 3**, pre-training on strong protein-protein binders from STRING DB results in drastically different attention patterns. For example, observe the increase in diffuse and unstructured patterns in layers 9, 10 and 11 (teal, blue, orange), or extreme reliance on the (BOS) token, especially in layers 4-7. We speculate that these layers are responsible for secondary structure information.

Most notably, both single and pair-sequence pre-trained models develop attention mechanism that tracks approximately every 4th amino acid, which is responsible for helical secondary structure via hydrogen-bond formation between carboxy and amide groups. In our tokenization scheme this corresponds to attending one or two preceding and proceeding tokens. Attention heads with such behavior were observed in Layers 3-9 in our models. Further attention signatures for other pre-trained models are shown in **Appendix Figure 10**.

##### 4.5.1 Fine-tuning alters attention patterns

Fine-tuning has diversified the attention patterns, see **Figure 8A**) right panels. Most notably, sequence-pair pre-training produced embedding layers and heads with extreme focus on the BOS token, as seen in the bottom-left plot layers 4-7. After fine-tuning for protein-protein binding with STRING binders, the attention focused on "VEQL" and "KC" tokens, which have charged Q and K residues, and covalent cysteine bond member C. Also, the reliance on BOS and EOS tokens decreases as we fine-tune the model.



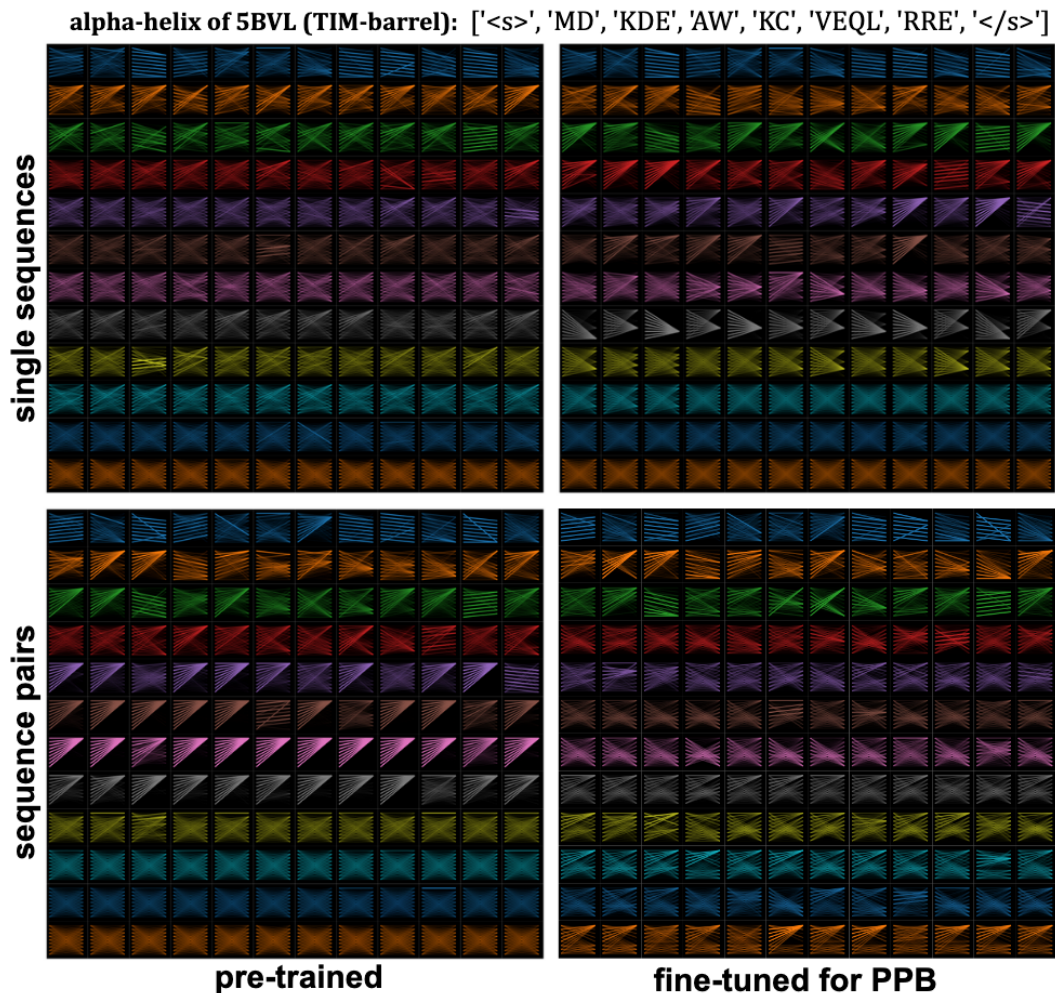


Figure 3: Self-attention for the first alpha-helix tokens of a synthetic protein 5BVL [20]. Self-attention shown for all combinations of layers (rows 0-11, top-to-bottom, same color) and attention heads (columns 0-11). Visualized with BertViz [44].

Next, we observed that fine-tuning alleviated the non-specific attention as observed in sequence pairs pretrained model. After fine-tuning, L11 had at least 3 attention heads that were informative and did not focus solely on BOS token.

## 5 Conclusions

In this study we demonstrated that pre-training a RoBERTa language model with MLM objective on a mixture of binding and random protein pairs results in superior downstream protein classification performance. Next, we compressed the protein sequence space by 64% by expanding the 20 amino acid character vocabulary to 10K sub-word tokens by BPE algorithm [40]. Further, we prepared new pre-training and protein-protein binding prediction datasets from STRING database [42].

Finally to our knowledge, we are the first ones to train a BERT-like model that supports extremely long protein sequences. Although only 3/10,000 proteins are longer than 2,048 tokens (as generated by BPE), those proteins belong to the dark proteome and could benefit from better functional annotation. Also, having a big input representation size is crucial for predicting protein-pair interactions, or even interactions between 3-20 proteins as is common in transcription machinery or when the homo/hetero-protomer complexes assemble.

The language modeling community increasingly advocates for training a large model (in terms of layers and attention heads) on a big dataset for fewer epochs to achieve better downstream results, rather than passing through a small dataset for multiple epochs with a small model [25]. With this dogma in mind, we were somewhat cautious when evaluating the recent protein embeddings [29], that take a similar RoBERTa approach to protein LM, since they trained on a small SwissProt dataset with a shallow and narrow model.

All in all, the study is still largely exploratory and breadth-oriented, and for future studies we aim to focus in depth on tackling the token-level classification tasks such as secondary structure prediction and contact point-prediction which require the model to understand secondary and tertiary structure. We also aim to explore the benefits of adding an explicit next sequence prediction objective. For further future directions see **Appendix Future Directions 5**.

## References

- [1] *A PyTorch Extension: Tools for easy mixed precision and distributed training in Pytorch*, (accessed September, 2020). <https://github.com/NVIDIA/apex>.
- [2] Ethan C Alley, Grigory Khimulya, Surojit Biswas, Mohammed AlQuraishi, and George M Church. Unified rational protein engineering with sequence-based deep representation learning. *Nature methods*, 16(12):1315–1322, 2019.
- [3] José Juan Almagro Armenteros, Casper Kaae Sønderby, Søren Kaae Sønderby, Henrik Nielsen, and Ole Winther. Deeploc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, 33(21):3387–3395, 2017.
- [4] Mohammed AlQuraishi. End-to-end differentiable learning of protein structure. *Cell systems*, 8(4):292–301, 2019.
- [5] Christian B Anfinsen. Principles that govern the folding of protein chains. *Science*, 181(4096):223–230, 1973.
- [6] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- [7] Tristan Bepler and Bonnie Berger. Learning protein sequence embeddings using information from structure. *arXiv preprint arXiv:1902.08661*, 2019.
- [8] Kaj Bostrom and Greg Durrett. Byte pair encoding is suboptimal for language model pretraining. *arXiv preprint arXiv:2004.03720*, 2020.
- [9] John-Marc Chandonia, Naomi K Fox, and Steven E Brenner. Scope: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic acids research*, 47(D1):D475–D481, 2019.
- [10] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What does bert look at? an analysis of bert’s attention. *arXiv preprint arXiv:1906.04341*, 2019.
- [11] UniProt Consortium. Uniprot: a worldwide hub of protein knowledge. *Nucleic acids research*, 47(D1):D506–D515, 2019.
- [12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [13] Sara El-Gebali, Jaina Mistry, Alex Bateman, Sean R Eddy, Aurélien Luciani, Simon C Potter, Matloob Qureshi, Lorna J Richardson, Gustavo A Salazar, Alfredo Smart, et al. The pfam protein families database in 2019. *Nucleic acids research*, 47(D1):D427–D432, 2019.
- [14] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhownik, et al. Prottrans: Towards cracking the language of life’s code through self-supervised deep learning and high performance computing. *arXiv preprint arXiv:2007.06225*, 2020.

- [15] Wenzhong Guo, Jianwen Wang, and Shiping Wang. Deep multimodal representation learning: A survey. *IEEE Access*, 7:63373–63394, 2019.
- [16] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. Don’t stop pretraining: Adapt language models to domains and tasks. *arXiv preprint arXiv:2004.10964*, 2020.
- [17] Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC bioinformatics*, 20(1):723, 2019.
- [18] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [19] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.
- [20] Po-Ssu Huang, Kaspar Feldmeier, Fabio Parmeggiani, D Alejandro Fernandez Velasco, Birte Höcker, and David Baker. De novo design of a four-fold symmetric tim-barrel protein with atomic-level accuracy. *Nature chemical biology*, 12(1):29–34, 2016.
- [21] Sameer Khurana, Reda Rawi, Khalid Kunji, Gwo-Yu Chuang, Halima Bensmail, and Raghendra Mall. Deepsol: a deep learning framework for sequence-based protein solubility prediction. *Bioinformatics*, 34(15):2605–2613, 2018.
- [22] Michael Schantz Klausen, Martin Closter Jespersen, Henrik Nielsen, Kamilla Kjaergaard Jensen, Vanessa Isabell Jurtz, Casper Kaae Soenderby, Morten Otto Alexander Sommer, Ole Winther, Morten Nielsen, Bent Petersen, et al. Netsurfp-2.0: Improved prediction of protein structural features by integrated deep learning. *Proteins: Structure, Function, and Bioinformatics*, 2019.
- [23] Taku Kudo and John Richardson. Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. *arXiv preprint arXiv:1808.06226*, 2018.
- [24] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942*, 2019.
- [25] Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joseph E Gonzalez. Train large, then compress: Rethinking model size for efficient training and inference of transformers. *arXiv preprint arXiv:2002.11794*, 2020.
- [26] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*, 2019.
- [27] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [28] Seonwoo Min, Seunghyun Park, Siwon Kim, Hyun-Soo Choi, and Sungroh Yoon. Pre-training of deep bidirectional protein sequence representations with structural information. *arXiv preprint arXiv:1912.05625*, 2019.
- [29] Ananthan Nambiar, Maeve Elizabeth Heflin, Simon Liu, Sergei Maslov, Mark Hopkins, and Anna Ritz. Transforming the language of life: Transformer neural networks for protein prediction tasks. *BioRxiv*, 2020.
- [30] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. dAlché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

- [31] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.
- [32] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. URL [https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language\\_understanding\\_paper.pdf](https://s3-us-west-2.amazonaws.com/openai-assets/researchcovers/languageunsupervised/language_understanding_paper.pdf), 2018.
- [33] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners.
- [34] Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Peter Chen, John Canny, Pieter Abbeel, and Yun Song. Evaluating protein transfer learning with tape. In *Advances in Neural Information Processing Systems*, pages 9686–9698, 2019.
- [35] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *bioRxiv*, 2019.
- [36] Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A primer in bertology: What we know about how bert works. *arXiv preprint arXiv:2002.12327*, 2020.
- [37] Jennifer L Ross. The dark matter of biology. *Biophysical Journal*, 111(5):909–916, 2016.
- [38] Philippe Schwaller, Benjamin Hoover, Jean-Louis Reymond, Hendrik Strobelt, and Teodoro Laino. Unsupervised Attention-Guided Atom-Mapping. 5 2020.
- [39] D. Sehnal, A. S. Rose, J. Koča, S. K. Burley, and S. Velankar. Mol\*: Towards a common library and tools for web molecular graphics. In *Proceedings of the Workshop on Molecular Graphics and Visual Analysis of Molecular Data*, MolVA '18, page 29–33, Goslar, DEU, 2018. Eurographics Association.
- [40] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.
- [41] Nils Strodthoff, Patrick Wagner, Markus Wenzel, and Wojciech Samek. Udsmpot: Universal deep sequence models for protein classification. *bioRxiv*, page 704874, 2019.
- [42] Damian Szklarczyk, Annika L Gable, David Lyon, Alexander Junge, Stefan Wyder, Jaime Huerta-Cepas, Milan Simonovic, Nadezhda T Doncheva, John H Morris, Peer Bork, et al. String v11: protein–protein association networks with increased coverage, supporting functional discovery in genome-wide experimental datasets. *Nucleic acids research*, 47(D1):D607–D613, 2019.
- [43] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.
- [44] Jesse Vig. A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*, 2019.
- [45] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

## Appendix

### Fine-tuning dataset preparation

**Protein-protein binding prediction** We used the same high confidence binding data from STRING DB. However, this time, the labeled training data was composed of 2M short ( $< 512$  tokens) and 0.667M long ( $> 512$  and  $\leq 2048$  tokens) protein binding pairs, in a 3:1 ratio. Development and test

STRING split	Train	Dev	Test
loose total avail.	54.4M (0.89)	-	22.1M
loose used	4.00M (0.75)	0.200M	0.600M
strict total avail.	54.4M (0.89)	-	248K (0.37)
strict used	2.67M (0.75)	63.9K	60.0K

Table 6: Preparing STRING dataset for model fine-tuning. In parentheses we show what fraction of sequence pairs after tokenization are less than 512 tokens-long (sequences  $A$  and  $B$ , plus special tokens), the remainder is between 512 and 20148 tokens.

sets had 63.9K and 60.0K sequences, respectively. The dataset was composed of 1:1 binding:non-binding protein pairs <sup>2</sup>.

**T-cell receptor binding prediction** The task predicts for a given T-cell Receptor (TCR) the most likely peptide fragments that bind the TCR (peptide fragments from from a larger antigen epitope that’s fragmented/digested by the T-cell and subsequently presented on its surface). TCRs here are represented by their very short CDR3 regions which are only 5-20 aa. There are 124,486 TCR and binding epitope pairs, and 13,832 pairs in evaluation dataset.

**Remote Homology** Sequence classification task to assign a protein to one of 1,995 possible folds. Data from [15]. Protein fold classes follow Zipf’s law, very few folds take up most of 12,311 sequences in training dataset. Top 6 folds take up more than 21 % of all data.

**Solubility** Binary task to predict if proteins are soluble or not. Final training data included 28,972 soluble and 40,448 insoluble proteins, approximately balanced at 7:10 soluble:insoluble ratio.

### Extra attention results

Here we give a glimpse into the attention mechanisms of our single-sequence String DB pretrained model, String1Seq, 9. Initial Layer 0 (L0) exhibits attention to previous and next tokens. Layer L1 mostly, focuses on BOS token, while L2 on a mixture of BOS and EOS tokens. One general pattern we observed is that the reliance on BOS and EOS tokens decreases as we fine-tune the model and as we progress upwards through its layers. For example, attention on previous tokens is strong in L3H9, which is probably instrumental for the model’s secondary structure understanding, but unlike in L1-2, BOS and EOS tokens are largely ignored.

Starting with L3-4 we observe several attention modes per head, while in the previous layers one mode was always dominating. For instance, in L4H2 there is a strong self attention towards the first half of sequence, [‘MD’, ‘KDE’, ‘AW’, ‘KC’], while in L4H11, first half of sequence focuses on the next token and second half of the sequence on the first half. Similarly L5 has a mixture of heads that look onto the last tokens, or focus on [‘AW’, ‘KC’]. We show all token-token attention patterns for L6H6 in the right-most panel, interestingly, the first 4 protein tokens exclusively focus on the last two tokens [‘VEQL’, ‘RRE’], while the last two tokens focus only on [‘MD’, ‘KDE’, ‘AW’, ‘KC’]. This uncovers another pattern, of learning to pay attention separately to the two halves of the sequence. L7-9 demonstrate multiple modes in most of their attention heads. Interestingly, in L9H10, we see the opposite of L6H6 - the set of the first four tokens and the final two tokens pay attention only to their respective subsets. Layer 11 is dominated by this pattern, except that the first half pays attention to BOS token.

### Future directions

As is common in science, by the end of this study there are more unanswered questions than at the beginning. Here we present a few directions where the project could be extended.

<sup>2</sup>See `scripts.string_prepare_finetuning_dataset`

```

{
  "<s>": 0, "<pad>": 1, "</s>": 2, "<unk>": 3, "<mask>": 4, "!" : 5, "\"": 6, "#": 7, "$": 8,
  ..
  "A": 37, "B": 38, "C": 39, "D": 40, ..., "x": 259, "Ñ": 260,
  ..
  "LL": 261, "AA": 262, "AL": 263, "VL": 264, "GL": 265, "EL": 266, "SL": 267, "GG": 268, "SS":
  269, "EE": 270, "TL": 271, "DL": 272,
  ..
  "HE": 412, "ALL": 413, "MS": 414, "GE": 415, "FP": 416, "HP": 417, "TK": 418, "TY": 419, "AAL":
  420, "VC": 421, "NF": 422, "ES": 423, "DF": 424, "AW": 425, "IV": 426, "MD": 427, "ED": 428,
  "QN": 429, "TV": 430, "LLL": 431, "QK": 432,
  ..
  "RGG": 793, "HM": 794, "NNNN": 795, "RSL": 796, "TGL": 797, "NAL": 798, "VRR": 799,
  ..
  "APPLE": 5734, "YVLPWQMSFWGATVI": 5735, "TLSP": 5736,
  ..
  "EIKK": 8490, "PHIKPEWYFLFAYAILRSIPNKL": 8491, "ALVH": 8492,
  ..
  "EIAR": 8565, "YVLPWQMSFWGATVITNLLSAIPYIG": 8566, "KLPL": 8567, "SGTP": 8568,
  ..
  "IQSG": 10238, "NDDG": 10239, "QQQQQQQQQQQQQQ": 10240, "ANPLNTP": 10241,
  ..
  "KGIS": 10259, "KGKN": 10260
}

```

Figure 4: BPE vocabulary is a dictionary with token strings arranged in the decreasing frequency serving as keys, and token IDs as the values. Here the most frequent two-letter tokens are "LL", "AA", "AL" and "VL", three letter - "ALL", "AAL", "LLL". Unusually long tokens are observed at 5735, 8491 and 8566 positions

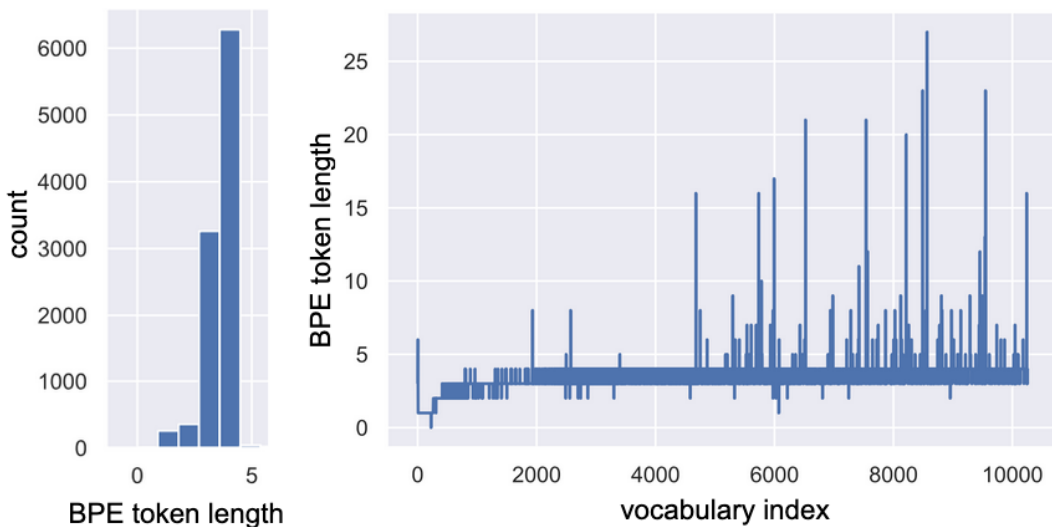


Figure 5: Histogram of BPE token lengths. On the left panel we omitted low frequency token lengths (longer than 5). More than 6,000 tokens are 4-characters-long, 3,000 tokens consist of 3 characters, and only less than 200 tokens are 1 or 2 characters-long. Some rare unusually long tokens appear on the right figure, such as 23 character PHIKPEWYFLFAYAILRSIPNKL and 27 character YVLPWQMSFWGATVITNLLSAIPYIG at indices 8491 and 8566 respectively.

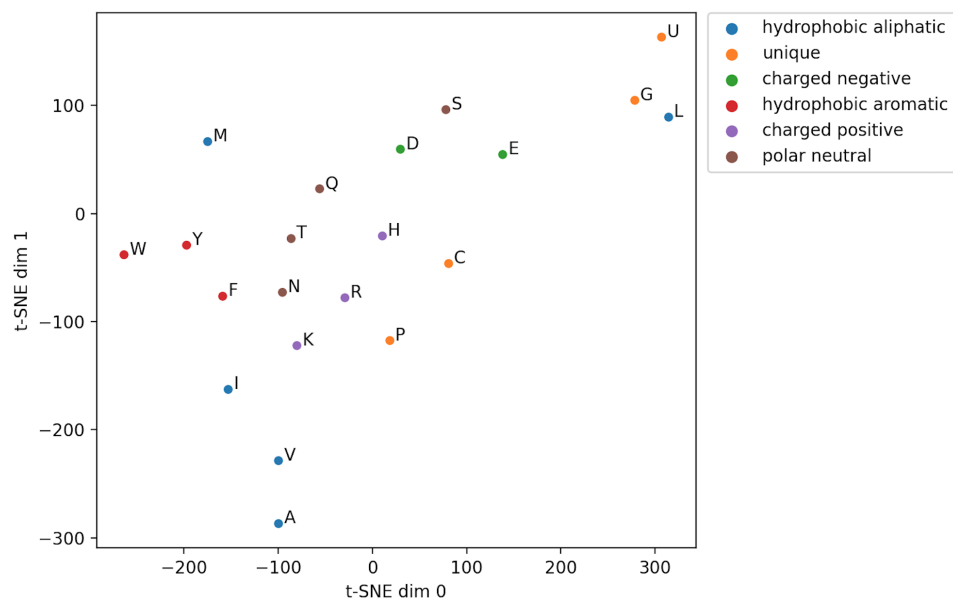


Figure 6: t-SNE visualization of 768-dimensional embedding space for 21 single character BPE tokens that encode amino acids. Parameters for t-SNE compression: learning rate: 40, perplexity: 5

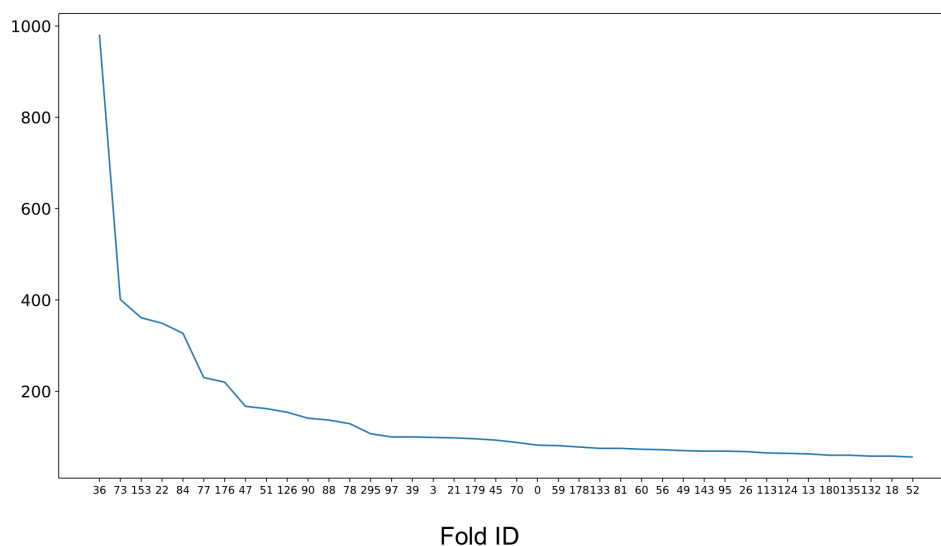


Figure 7: Top 40 most common protein fold classes among 12,311 proteins in the remote homology dataset.



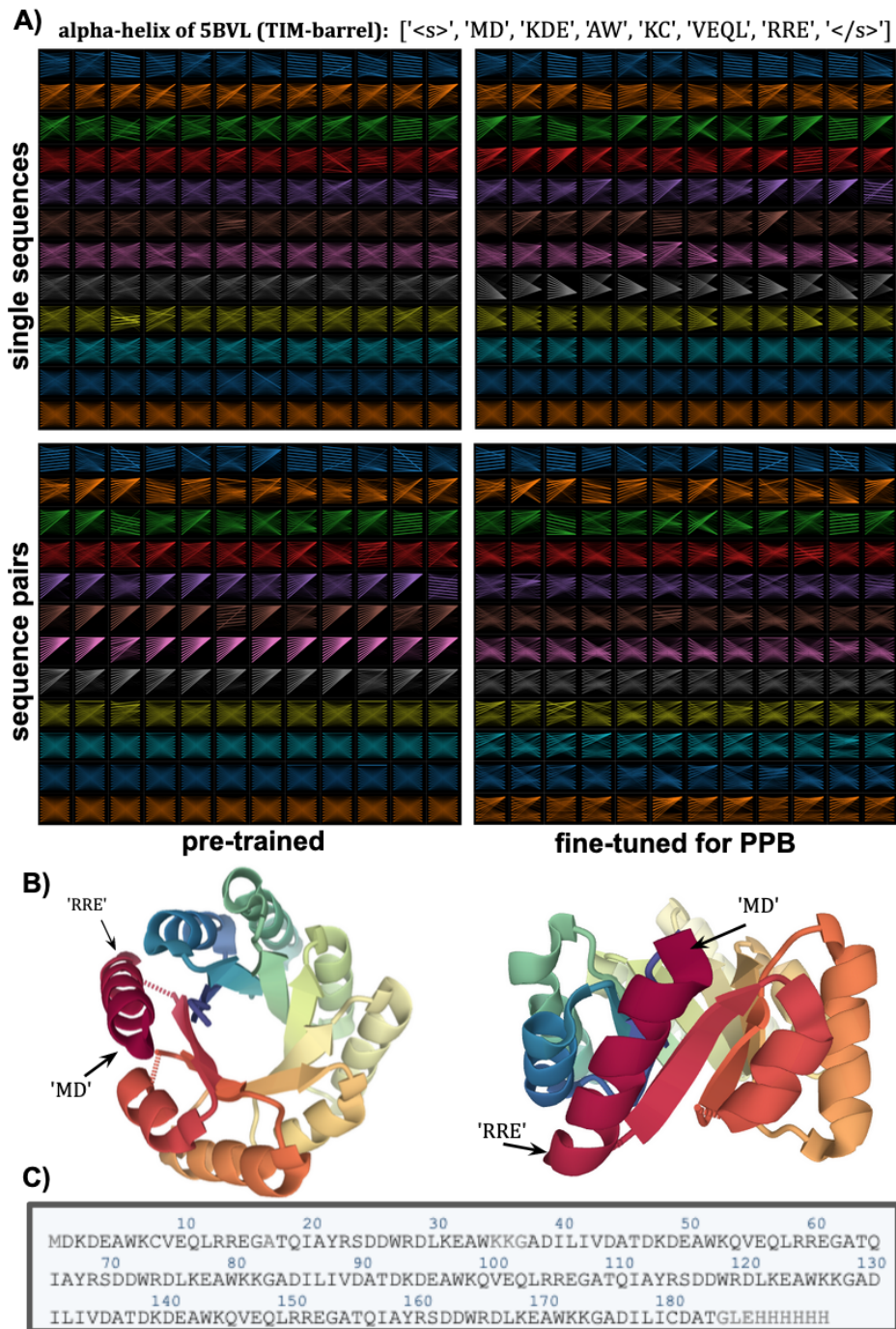


Figure 8: Visualizing self-attention between pre-trained and fine-tuned (protein-protein binding task) RoBERTa models for alpha-helix of a synthetic protein 5BVL [20]. **A)** Self-attention for 8 tokens shown for all combinations of model layers (rows 0-11, top-to-bottom, same color) and attention heads (columns 0-11). Visualisation done with BertViz [44]. **B)** 3D visualization (using Mol\* [39]) of 5BVL from the bottom (left) and side (right) views. The 15aa-long alpha-helix, for which the attention is shown in A), is colored in bordeaux. **C)** Amino acid sequence for 5BVL. Alpha helix of interest corresponds to positions 0 – 15, and beta-sheet that follows is at 19 – 2.



alpha-helix of 5BVL (TIM-barrel): ['<s>', 'MD', 'KDE', 'AW', 'KC', 'VEQL', 'RRE', '</s>']

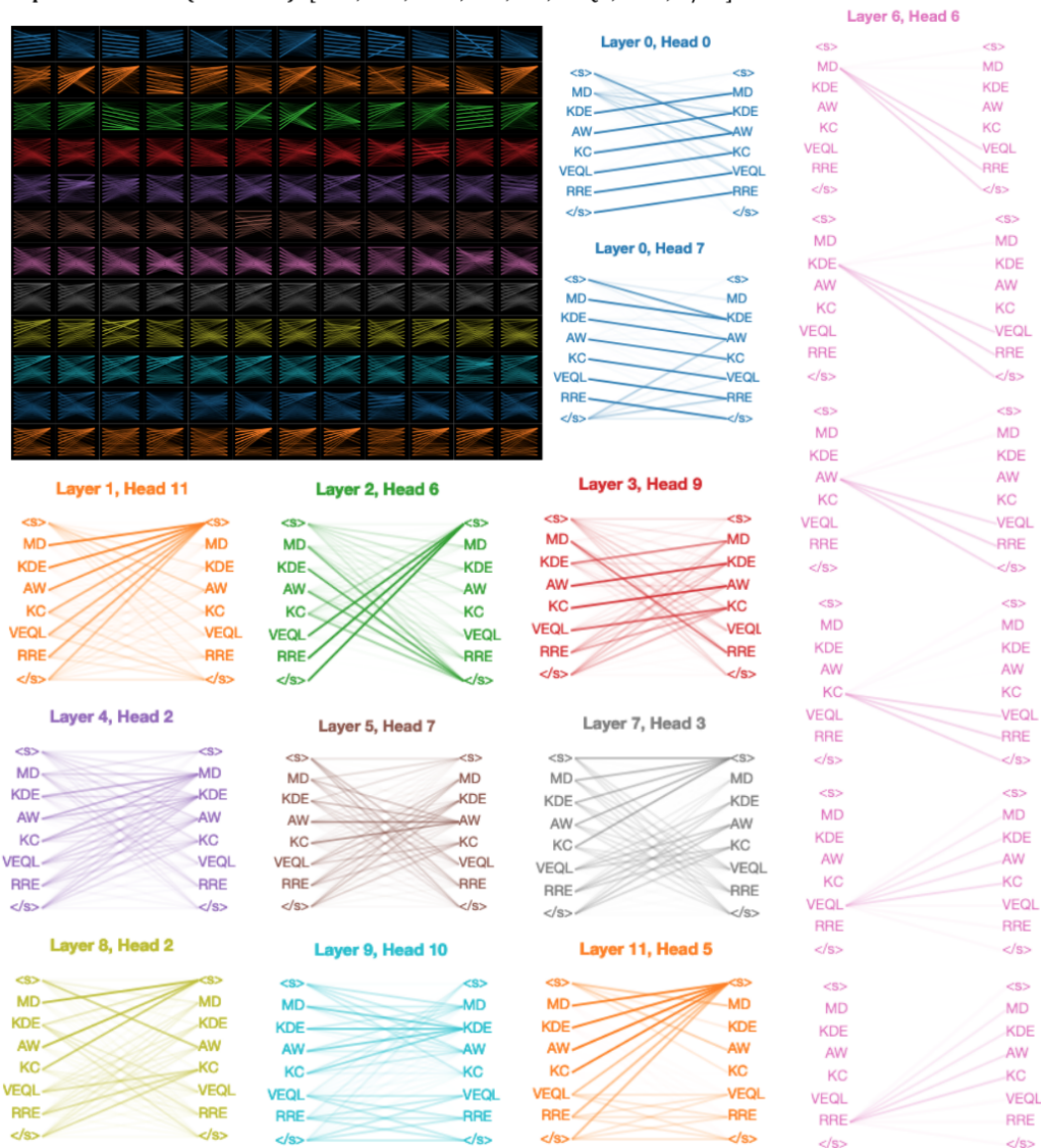


Figure 9: In-depth look at self-attention patterns after fine-tuning for protein binding prediction. Showing alpha-helix of a synthetic protein 5BVL [20]. **A** Self-attention for 8 tokens shown for all combinations of model layers (rows 0-11, top-to-bottom, same color) and attention heads (columns 0-11). Visualized with BertViz [44]

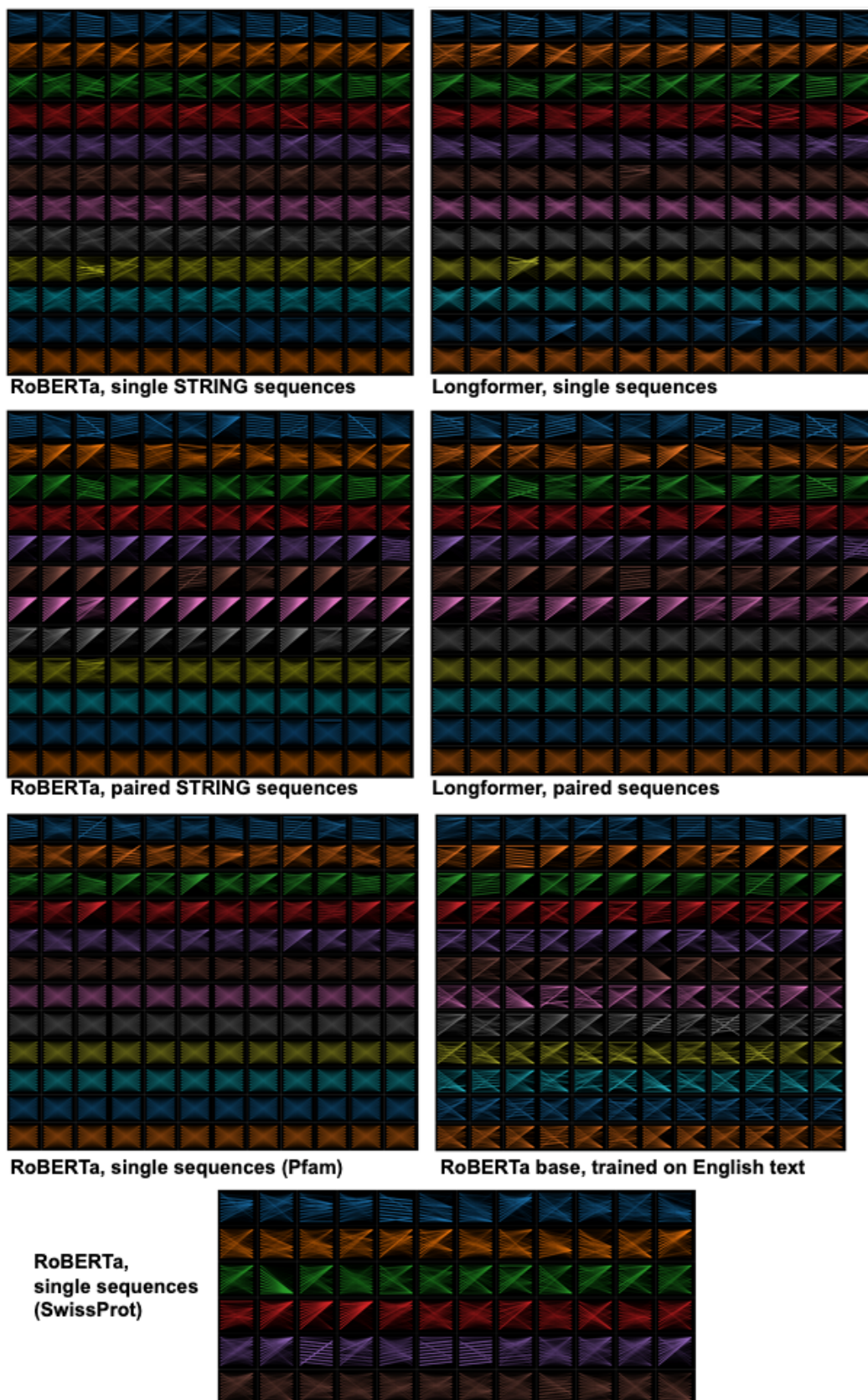


Figure 10: Attention mechanisms across layers and self-attention heads for other pretrained models. Visualized with BertViz [44]

### Secondary structure prediction on Byte-Pair encoded sequences

So far we have used a 20 amino acid vocabulary when doing a 3-class secondary structure predictions. Training a protein model with 20-25 amino acid vocabulary is straightforward, we treat each character as a token which has a single corresponding label, and predict new labels in the test set. But what if we have 10k tokens from BPE, and assume that a set of longer, multi-character BPE tokens will carry more structural information to predict secondary structure. To utilize the full 10k vocabulary we have to overcome a problem: input sequence tokens (after BPE encoding) will break the continuity of labels. See this problem illustrated in **Figure 11**. Secondary structure labels often form 2-7 amino-acid-long sequences, which we call label islands, see **Figure 12**. We are proposing to explore pre-tokenizing all the training data with BPE tokenizer and creating a new label-island aware vocabulary, and prune the BPE vocabulary by removing the tokens that do not appear in the new vocabulary. Then during evaluation, we would hope that the test sequences are correctly parsed, and label continuity is preserved after tokenizing with the new pruned vocabulary. Alternatively, we could split the sequence into tri-grams.

<b>Input sequence</b>	A C A B D E E E A C C C A E A A B
<b>2° struct labels</b>	0 0 0 0 0 2 2 0 1 1 1 1 2 2 0 0 2
<b>Label island-aware input</b>	ACABD EE E ACCC AE AA B
<b>Tokenized label islands</b>	ACA,BD EE E ACC,C AE AA B
<b>Vocabulary</b>	{ACA, BD, EE, E, ACC, C, AE, AA, B}

Figure 11: Label island continuity problem.

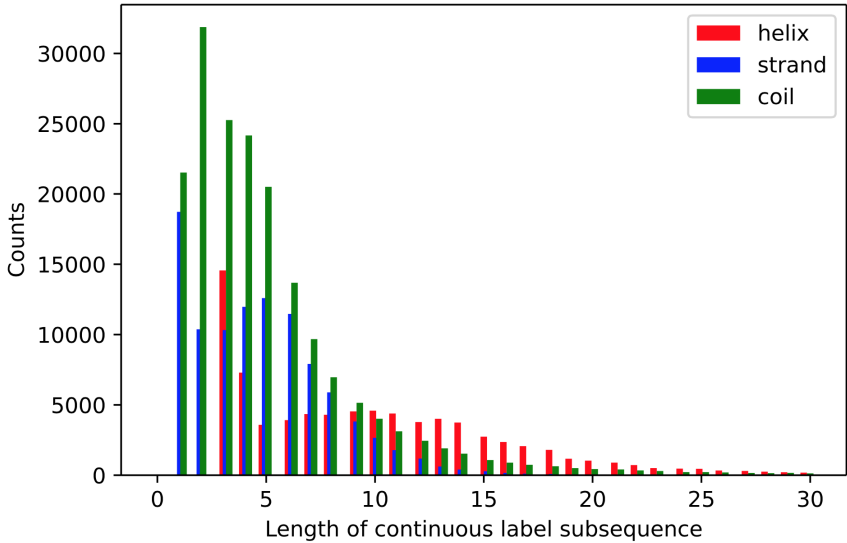


Figure 12: Secondary structure label space continuity

### Searching for a suitable Next Sentence Prediction objective in the protein domain

We propose further exploration if and what supplementary intrinsic pretraining objective (besides the masked language modeling, or MLM) could be beneficial for the secondary structure and PPI prediction fine-tuning tasks. For example, Min et al (2020) [28] proposed the Same Family Prediction objective. However as we saw in [34, 35], clustering proteins into Pfam families is a relatively easy task with > 90% accuracies. Therefore, would propose using Protein-Protein Binding (PPB) objective, because the model would have to explicitly learn the feature space responsible for

detecting binding epitopes. The model implementation is quite simple, we could simply apply a binary mask on our input, that signifies which tokens belong to which sequence, as is done in BERT [12] model.

### **Investigate new uses for extremely large model input space (2,048 tokens) made available by Longformer pre-training**

We estimate that a model with  $T = 2048$  can accommodate more than 99.99% protein sequences. In theory, if we assume that self-attention can fully understand protein secondary and tertiary structure, with such a large model we could predict the binding between two (1,022 each), four (511) or eight (255) proteins. But first, we still have to validate the Longformer on all downstream tasks.

### **Attempt TCR task in a different way**

T-cell receptor binding prediction is extremely hard to model. We propose to introduce a new protein sequence task akin to question answering in NLP. We could train on full TCR sequences, not just CDR3, with an additional objective which determines the exact span where the short CDR3 lies in the long TCR sequence.

### **Visualization and quantification of binding-site predictions**

We assume that by training on protein sequence pairs, the self-attention heads learn across the layers which subsequence patterns lead to the two-sequence binding and recognize what patterns constitute the binding sites. By visualising combinations of  $L$  layers and  $A$  self-attention heads, we try to retrieve the correct binding sites. Instead of the simple binding pair visualization by BertViz, we are curious to utilize RXNMapper [38] to discover the binding sites, and give probabilities for binding site prediction.

### **Novel encoding strategies**

BPE is a deterministic subword encoder, which produces tokens that do not always have a morphological meaning. According to [8], Unigram language modeling [23] is more advantageous than BPE when pre-training language models. Whereas BPE creates new subword tokens while  $|V| < k$ , ULM algorithm starts with a superset of the final vocabulary, and while  $|V| > k$  ( $V$  is a set of all substrings occurring more than once in the set of strings  $D$ ) proceeds to prune out tokens, by inferring a LM parameters  $\theta$ . Using Viterbi algorithm, the optimal segmentation is determined (this segmentation has maximal likelihood with inferred  $\theta$ ). Therefore, we propose using Unigram LM in the next iteration of this project.