2021 43rd Annual International Conference of the
IEEE Engineering in Medicine & Biology Society (EMBC)
Oct 31 - Nov 4, 2021. Virtual Conference

# Deep Learning Proteins using a Triplet-BERT network

Mark Lennox[1], Neil Robertson[2], Barry Devereux[3]

***Abstract*— Modern sequencing technology has produced a vast quantity of proteomic data, which has been key to the development of various deep learning models within the field. However, there are still challenges to overcome with regards to modelling the properties of a protein, especially when labelled resources are scarce. Developing interpretable deep learning models is an essential criterion, as proteomics research requires methods to understand the functional properties of proteins. The ability to derive quality information from both the model and the data will play a vital role in the advancement of proteomics research. In this paper, we seek to leverage a BERT model that has been pre-trained on a vast quantity of proteomic data, to model a collection of regression tasks using only a minimal amount of data. We adopt a triplet network structure to fine-tune the BERT model for each dataset and evaluate its performance on a set of downstream task predictions: plasma membrane localisation, thermostability, peak absorption wavelength, and enantioselectivity. Our results significantly improve upon the original BERT baseline as well as the previous state-of-the-art models for each task, demonstrating the benefits of using a triplet network for refining such a large pre-trained model on a limited dataset. As a form of white-box deep learning, we also visualise how the model attends to specific parts of the protein and how the model detects critical modifications that change its overall function.**

## I. Introduction

Deep learning models such as deep neural networks (DNN) are becoming increasingly popular in bioinformatics as they can handle large datasets, require minimal feature engineering and are capable of handling complex relationships within the data. Deep learning has proven that it can model a variety of complex processes within biology, as these models provide predictions without any explicit knowledge of the specific physical and biological mechanisms. However, a substantial amount of labelled data is usually required during the development stages. These resources are often not available for certain protein design and engineering tasks, which is inconvenient when modelling critical properties within a protein [48, 47, 46]. Computational bioinformatics and protein modelling require new approaches to develop robust deep learning models that can combat the lack of labelled data. A majority of the deep learning techniques applied in bioinformatics research originate from applications within image classification [21, 13, 35] and language modelling [31, 41, 8]. State-of-the-art approaches in the fields of natural language processing (NLP) [31, 32, 8], and computer vision (CV) [19, 43, 38], now commonly employ a technique known as *pre-training*. These methods have revealed that DNNs can still retain their performance as these methods produce robust models with a limited number of training examples.

The method of pre-training requires a deep learning model to be trained on a separate task before being fine-tuned to a different dataset [16]. The utility of pre-training was first demonstrated within the field of computer vision [7, 49], as large convolutional neural networks were initially trained on vast image datasets, before being fine-tuned to specific tasks [21, 40, 37]. In NLP, state-of-the-art language models use vast corpora of text to perform unsupervised, or self-supervised pre-training [31, 32, 8]. Recent approaches within protein sequence analysis have employed similar methods during training [48, 33, 30]. However, pre-training is still costly and time-consuming to perform as it requires a considerable amount of computational resources, and so has had a slow adoption rate within computational biology.

Computer vision was again at the forefront of modern machine learning with the application of *metric-learning* for modelling limited datasets. In metric learning, the original inputs to a DNN are transformed into a feature space that can be used to compare and match examples based on a distance metric (i.e. Euclidean distance, cosine-distance) [45, 17, 28]. Examples of such deep metric-learning include the use of siamese networks [19], triplet-networks [9], and matching networks [43]. In this work, we aim to determine if both pre-training and metric learning can be implemented simultaneously to develop a deep learning model that is suitable for modern protein sequence analysis.

In this paper, we will consider a pre-trained BERT model [33], which is based on a large corpus of unlabeled protein sequences with the goal of re-purposing this model by fine-tuning it using a triplet network for a set of downstream tasks. During training, triplets (i.e. anchor, positive and negative) of the protein sequences will be used along with weight-sharing within the BERT model to cluster the data based on a triplet loss [15]. The BERT model will be used to produce a vector representation for the anchor, positive and negative protein respectfully. During training, a protein is considered to be a positive example to the anchoring protein if its labelled value (i.e. measured property) is closer in absolute value to the anchor's label when compared to the negative instance. Throughout the tuning process, new triplets are formed as the BERT model undergoes semi-supervised training, and begins clustering individual cases within the dataset.

Determining the critical properties of a protein is one of the most challenging aspects of any downstream task. Traditionally, many of these properties are discovered by examining the physical structure of the protein. However,

*This work was not supported by any organization

[1]EEECS, Queen's University Belfast, mlennox05@qub.ac.uk
[2]EEECS, Queen's University Belfast, n.robertson@qub.ac.uk
[3]EEECS, Queen's University Belfast, b.devereux@qub.ac.uk

978-1-7281-1178-0/21/$31.00 ©2021 IEEE

this is often a very time-consuming and expensive process. Another option is to encode each amino acid with a basic set of physical properties (e.g. its charge or hydrophobicity). Inevitably many physical properties can be missed or poorly represented by such feature engineering, which then leads to overfitting and inadequately modelling of the downstream task. Many have considered encoding the primary structure of the protein (i.e. the sequence of amino acids) [18, 39, 44], where a vector of real numbers represents each amino acid, and are optimised in a deep learning model. Since deep learning is often used to avoid feature engineering, these models can capture sophisticated features by analysing the original sequence of amino acids.

As vast resources of proteomic data become available (e.g. the UniProt database [5]), it provides an opportunity to use large amounts of unlabelled (with respect to the downstream task of data) data to perform large-scale pre-training. This could be a vital step forward for proteomic research as it is a far less expensive and time-consuming alternative. Rao et al. displayed the potential of pre-training and fine-tuning for protein sequence analysis by introducing the Tasks Assessing Protein Embeddings (TAPE) [33]. They benchmarked the current state-of-the-art models on a set of five protein tasks that included a variety of domains within proteomics. This included a Transformer model [41], a ResNet model [50], and a long short-term memory (LSTM) model [14] of their own design. In addition to these three models, they also benchmarked two previously proposed architectures; another a bidirectional LSTM model [2], and a unidirectional mLSTM [20, 1]. Rao et al. experiments concluded with the transformer model outperforming every other model tested concerning its accuracy, perplexity and exponentiated cross-entropy [33]. These results were not surprising as transformer-style architectures have quickly become the new standard for many NLP tasks [41, 8, 6].

Given that Rao et al. have shown that both a transformer is a strong candidate for modelling a protein sequence [33], we aim to build upon their work by testing this pre-trained model on four downstream tasks introduced by Yang et al. [48]. In this way, we aim to investigate whether such pre-trained embeddings can help predict relevant properties for a set of downstream tasks. We aim to improve this pre-trained model with the use of a triplet style network to fine-tune the model to incorporate additional relevant information about the protein for each specific downstream task. A drawback to Rao et al. work was that they only tested their transformer network using a character-based encoding [33]. However, state-of-the-art transformer language models now commonly use a subword encoding algorithm before embeddings a sentence [31, 32, 8]. Subword algorithms such as byte-pair encoding algorithm (BPE) [10], or unigram encoding algorithm [22] can provide more extensive vocabulary during training. These encoding algorithms also have the added benefit of reducing the length of the input sequence. This could also reduce the time and cost required to model protein sequences without using excessive amounts of padding.

In an extension to the Rao et al. work, Vig et al. explored how this BERT model [33] was capable of discerning structural and functional properties about the protein [42]. Vig et al. proved that the model was able to model long-range dependencies within the sequence of amino acids. It was also able to deduce information about the protein based on the folding structure, target binding sites, and additional complex biophysical properties [42]. They concluded that the specific heads within the model attended to individual amino acids, as the attention similarity matrix was positively correlated to the expected substitution scores (i.e. BLOSUM62) for each amino acid. Vig et al. noted that the deeper layers of the BERT model focused relatively more attention on binding sites and contacts [42]. In contrast, information about the secondary structure (i.e. low- to mid-level concepts) within the protein was targeted evenly across each of the layers.

Another example of using pre-training was by Yang et al. [48]. They applied both pre-training and an n-gram encoding strategy to analyse a set of proteins. Their approach consisted of using a tri-gram encoding to each protein sequence analysing the set of tri-grams with a Doc2Vec model [23]. The method encodes the protein in a trivial fashion, which makes it susceptible to poorly represented (i.e. infrequent) tri-grams that can later affect pre-training. Lennox et al. improved on this work by testing the use of subword algorithms on protein sequences [26]. Both approaches are unfavourable as they implement Doc2Vec models, which return a single vector representation for the entire protein. This makes it difficult to interpret each vector representation when querying specific modifications in the protein.

Metric learning is still uncommon within computational biology deep learning even though it has become standard practice in computer vision [19, 9, 43]. There have been many improvements to deep metric learning from its introduction with the siamese style networks [3, 4, 12]. One of the most notable instances was by Hoffer et al. with the triplet network [15].

In this work, we aim to use such a triplet style training procedure to improve the encodings produced by the BERT model of Rao et al. [33]. We extract embedded representations for four protein property prediction tasks [48] using the pre-trained BERT model [33], and then fine-tune our BERT model to each task. The tasked covered in this investigation contain proteins from various families, and library designs that were not included in Rao et al. work [33]. We show that the predictive power of models trained using these embeddings exceeds those trained on the previous state-of-the-art methods. This approach can be an accurate and efficient alternative as it does not require alignment or any additional structural data about the protein. A series of visualisation techniques will be used to present the critical relationships with the data, and how the BERT model attends to specific amino acids in the protein.

## II. Materials and Methods

Previous applications of pre-training [48, 26] and deep metric learning [24] have shown a clear benefit to applying either technique to analyse protein data. One key drawback to these approaches is the limited window sizes to which they encode segments of the protein. This can be detrimental to the model's performance as it is unable to capture long-range dependencies within the protein and therefore encodes less information about the protein's final structure. Our approach improves upon these examples by using a BERT-style model that is capable of encoding the complete protein in a bidirectional fashion. Past work has justified the importance of either pre-training the model or using metric learning. There is still room for improvement by bringing both approaches together by utilising a state-of-the-art pre-trained network than has been fine-tuned using a triplet style network. Since the pre-trained BERT model of Rao et al. has not been set up to handle a subword encoding (e.g. BPE or Unigram), we aim instead to set a stable baseline for the application of both pre-training and deep metric learning that will only use a character-based encoding.

### *A. Modelling Scheme*

The Triplet-BERT network employed in this investigation is outlined in Figure 1. For each task, the proteins are encoded by a BERT model [33], which has been re-trained on a set of protein sequences used in TAPE investigation. These proteins were collected from the recently curated Pfam database [11], which holds approximately thirty-one million protein domains and forms the corpus used to train large sequence models as featured in TAPE [33]. The architects of the Pfam database have organised the proteins into clusters that share evolutionary-related groups, also known as families. In summary, the BERT model consists of 12-layers with a hidden size of 512 units and eight attention heads, leading to a $\sim 38M$ - parameter model. It was trained using masked-token prediction [8]. Every layer of the BERT model is frozen except for the last layer, which will allow the model to be easily tuned to each task. The features produced by the final layer of the model are then pooled to form the vector representation for each protein. Initially the model will encode a triplet of proteins, $(x_a, x_p, x_n)$, whereby $x_a$, $x_p$, and $x_n$ denote the anchor, positive and negative proteins respectively. The BERT model will then output the following:

$$\begin{aligned} a &= f(x_a) \\ p &= f(x_p) \\ n &= f(x_n) \end{aligned} \tag{1}$$

$$\begin{aligned} D(a,p) &= \|a - p\| \\ D(a,n) &= \|a - n\| \end{aligned} \tag{2}$$

$$L(a,p,n) = \frac{1}{2}\{\max(0, m + D(a,p) - D(a,n)\}. \tag{3}$$

In this example, we are applying inter-domain learning because the weights of the BERT model are shared. The model is represented by the encoding function $f$, which is applied to each branch of the triplet network. Once the BERT model has encoded the triplet, it is then passed through one final dense, and L2-normalisation layer [36]. The triplet of encodings can then be used to train the BERT to rank the triplet based on the anchoring protein via the triplet-loss (Equation. 3, where $m = 0.1$). Once the BERT model has been fine-tuned, we will use its final generated encodings to build a simple regression model to predict the given properties of each task. As in past examples, Gaussian process (GP) regression models (Matérn kernels with $\nu = 5/2$) [34] will be used to model the properties along with the same train-test split for each task to remain unbiased to the experiments carried out in previous work [48, 26, 24].

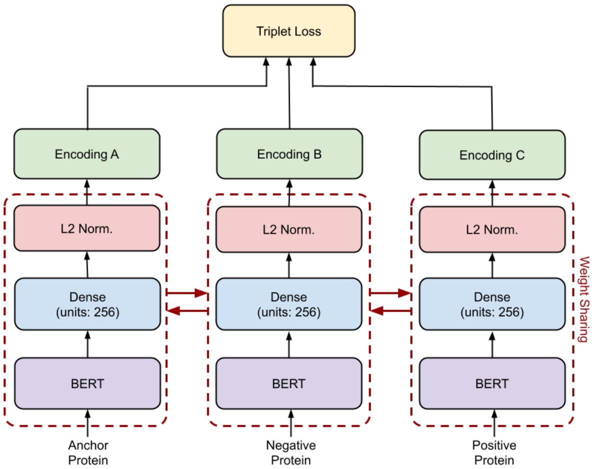


Fig. 1: Overview of the Triplet-BERT Approach.

### *B. Tasks*

To thoroughly evaluate the performance of using our approach, we included four downstream tasks in this investigation that cover a range of potential properties in which deep learning could be applied. These four tasks are the same as those used in previous work [48, 26, 24].

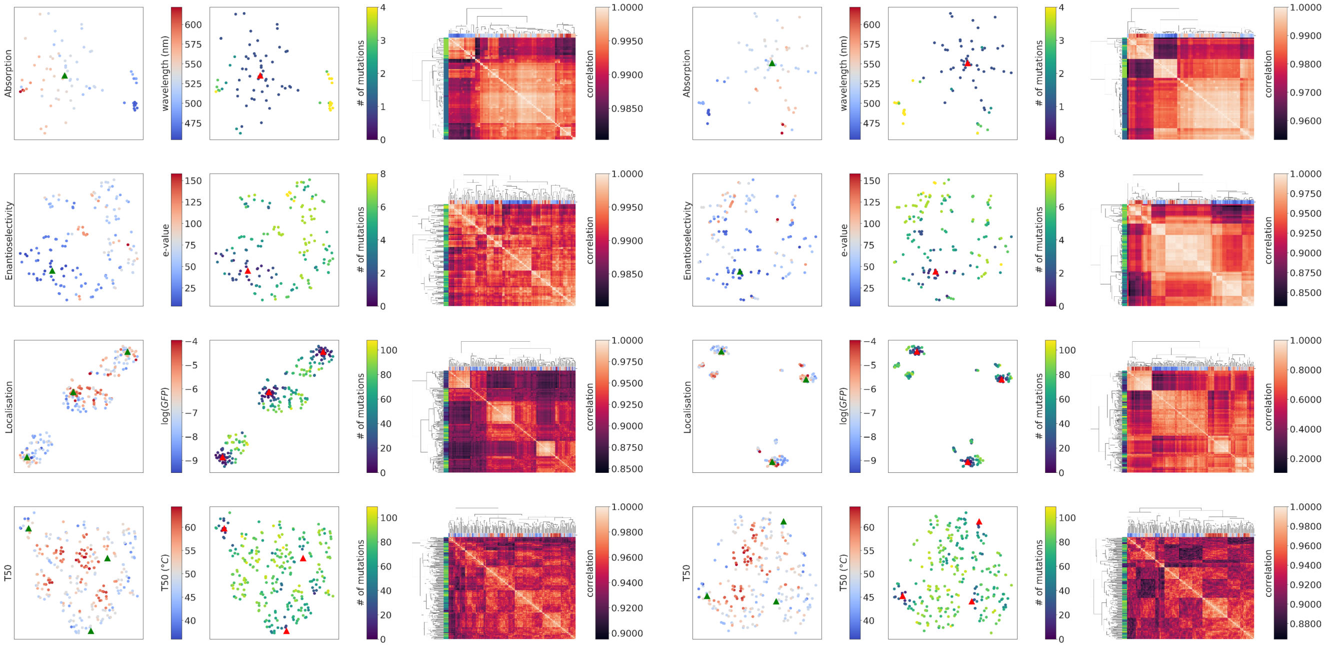
## III. Results and Discussion

In this study, we began by evaluating the original BERT model as an example of a pre-training strategy. This model was used as a baseline to our investigation on a set of downstream tasks. We then built on this approach by testing the advantages of combining both pre-training and deep metric learning to the same tasks, as shown in Table I. Just as in past studies, we run a five-fold cross-validation, whereby we adopted an eighty-twenty split of the generated triplets from the training data to train and validate the performance of the model for each dataset. Doing so provided a stable training setup during the fine-tuning stage of development. Unsurprisingly, the triplet-tuned version of the BERT model easily outperformed the original pre-trained baselines along with the other examples that included both CNN and Doc2Vec based models with an improved mean absolute error (MAE) score in each task. Our results indicate that the fine-tuning stage does alter the latent space produced by the original model, and tailors it to each specific downstream task improving the final representation of each protein. The real value in applying pre-training is observed

TABLE I: Results (MAE) for the four protein downstream tasks.

| Model | Encoding | Vocabulary Size | Absorption | Enantioselectivity | Localisation | T50 |
|---|---|---|---|---|---|---|
| BERT (Triplet) (Ours) | Character | 20 | **14.06 (1.257)** | **3.85 (0.586)** | **0.50 (0.03)** | **2.36 (0.139)** |
| BERT (Non-Triplet) [33] | Character | 20 | 16.57 (1.765) | 7.57 (0.767) | 0.70 (0.043) | 2.47 (0.121) |
| CNN (Triplet) [25] | Character | 20 | 17.14 (1.487) | 5.93 (0.741) | 0.63 (0.042) | 2.58 (0.13) |
| CNN (Non-Triplet) [25] | Character | 20 | 25.28 (2.266) | 8.01 (1.034) | 0.67 (0.052) | 3.32 (0.163) |
| Doc2Vec [27] | Unigram | 2000 | 26.41 (2.268) | 6.77 (1.018) | 0.65 (0.056) | 2.98 (0.166) |
| | | 4000 | 18.09 (1.740) | 6.90 (0.777) | 0.76 (0.041) | 2.80 (0.13) |
| | | 8000 | 20.92 (2.073) | 8.58 (0.894) | 0.86 (0.046) | 2.59 (0.172) |
| | | 16000 | 24.05 (2.013) | 7.07 (0.964) | 0.77 (0.052) | 3.33 (0.201) |
| | | 32000 | 21.98 (2.058) | 9.53 (0.978) | 0.76 (0.049) | 2.96 (0.18) |
| Doc2Vec [27] | BPE | 2000 | 23.83 (2.323) | 10.38 (1.06) | 0.66 (0.049) | 2.70 (0.184) |
| | | 4000 | 20.80 (2.097) | 9.76 (0.939) | 0.67 (0.045) | 3.01 (0.165) |
| | | 8000 | 18.46 (1.852) | 6.72 (0.808) | 0.75 (0.046) | 2.75 (0.13) |
| | | 16000 | 20.64 (1.850) | 6.08 (0.829) | 0.73 (0.048) | 2.76 (0.174) |
| | | 32000 | 24.27 (2.193) | 7.03 (0.95) | 0.67 (0.052) | 2.80 (0.175) |
| Doc2Vec [48] | Tri-gram | 8000 | 23.30 (2.129) | 9.14 (1.018) | 0.73 (0.047) | 2.91 (0.198) |
| Doc2Vec [27] | Character | 20 | 46.08 (3.718) | 12.55 (1.733) | 0.81 (0.091) | 4.32 (0.286) |

Notes: Mean Absolute Error (*MAE*) between the actual test values and the predicted test values.



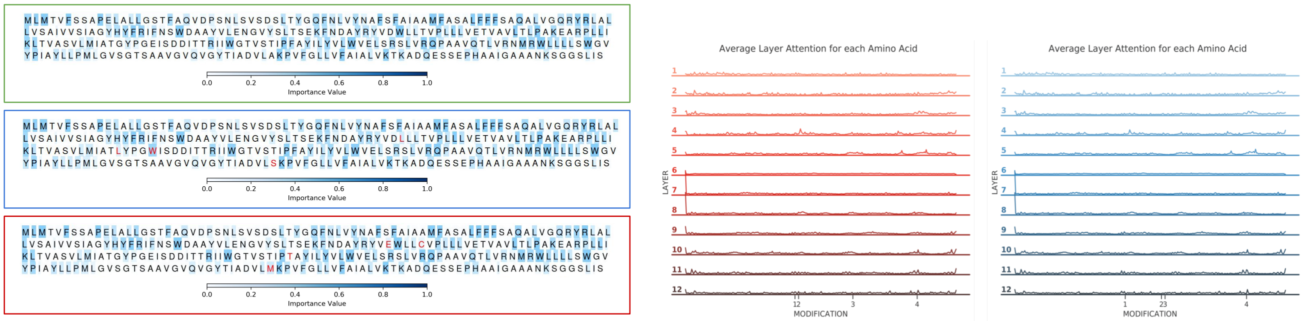
(a) Original BERT model

(b) Triplet-BERT model

Fig. 2: A set of t-SNE and cluster plots for both versions of the BERT model, thereby visualising the correlations within each learned embedding space (e.g. the number of modifications present and the functional property of each protein) (see text for details).

when the model can successfully encode a protein without any prior knowledge of biochemistry. Only during the pre-training stage does the model begin to learn these complex relationships between the amino acids within the protein sequence. Through proper fine-tuning can these pre-trained embeddings be improved by using deep metric learning to model subtle mutations within a set of amino acids.

The encodings produced by both strategies are visualised using a set of t-distributed stochastic neighbour embedding (t-SNE) [29] plots along with cluster maps, as shown in Figures 2a - 2b (with all t-SNE projections using a perplexity of 30) were produced for each downstream task. Figure 2a depicts the encodings of the original BERT model for each downstream task. While Figure 2b represented the final encodings produced once the BERT model had been tuned using our triplet network approach. By considering the combination of both plots, it is easier to envision how the BERT model perceives each protein sequence when using either strategy and observe the contribution of each mutation in the final feature vector representation.



(a)

(b)

Fig. 3: (a) A set of attention maps highlighting the importance of each amino-acid with three proteins from the peak absorption task. The parent protein is outlined in green, and its least and most absorbent versions are outlined in blue and red respectfully. Any modifications are represented by red lettering. (b) Average Attentions within the BERT model. The least and most absorbent mutated versions of the parent protein are coloured blue and red respectfully.

For the absorption task, we see how there is less of an order to the original encoding when compared to the triplet-tuned counterparts, as shown in Figures 2a-2b. In Figure 2b, it is far easier to determine which modifications will have a more significant effect on the proteins absorption value as the triplet-tuned encodings become more tailored to the task. Figures 2a-2b presents the model's ability to capture even the most minor modifications to the original parent protein, regardless of the length of the sequence. The high correlations observed in the cluster map in Figure 2a reflect the fact that all the modified proteins were based on one protein and indicated two main clusters within the dataset. However, in Figure 2b the cluster map based on the triplet-tuned encodings provide a more detailed depiction of how the proteins are correlated to one another as we observe smaller sub-clusters within the dataset.

In the enantioselectivity task, we again see the best performance from the triplet-BERT model, as shown in Table I. Still, both versions of the BERT model were capable of detecting any modifications present within the parent protein. However, when considering the cluster maps in Figures 2a-2b we can see the triplet tuned encodings provide more distinct clusters when compared to the originals. In Figures 2a-2b, we observe that the triplet encodings incorporate information with regards to the measured property and the modifications present within the protein. In the absorption task, we notice that an increase in the number of modifications could lead to either an increase or decrease in absorption values. However, for enantioselectivity, the more modifications that are present in the protein, the higher its expected e-value will be for this particular dataset.

When examining the plasma membrane localisation task, we again observed the best results from the triplet-BERT encodings, as shown in Table I. When visualising both sets of encodings in Figures 2a-2b, we can see how the BERT model easily clusters the three families tested in this specific task. Interestingly in Figure 2b, the triplet encoding clusters these families further away from one another. With smaller sub-clusters appearing for the proteins that have a higher localisation value. The number of mutations appears to have an opposite effect to that of absorption and enantioselectivity. From Figures 2a-2b, we see that the fewer mutations present within this protein leads to a higher overall localisation value. Just as in the case of the absorption and enantioselectivity, we again notice a far sharper cutoff between suspected groups within the data when using the triplet encodings for Figure 2b. In considering the cluster maps in Figures 2a-2b, it becomes easier to recognise which parent protein is more or less receptive to the task as the model becomes better at detecting the relevant modifications.

In the final task, we again observed the triplet-BERT model producing the best encodings for modelling thermostability (i.e. T50) values, as outlined in Table I. When we visualise the encodings produced by each strategy in Figures 2a-2b, the proteins that possess the highest T50 values are clustered into the centre of each plot. Unlike in the localisation task, the encodings produced by the BERT model for the thermostability task are not initially separated into three distinct clusters based on the parent proteins. Instead, we see a series of smaller groups with most of the proteins with high thermostability values congregating in the centre. Similarly to absorption and enantioselectivity, the number of mutations present in the protein is positively correlated to the thermostability value. Similarly to the localisation task, when we consider the cluster maps in Figures 2a-2b, we can see that triplet-BERT model is better at clustering the proteins with higher thermostability values together.

To reinforce the utility and interpretability of this approach, we have also included a set of plots in Figure 3 that focus on a few examples from the peak absorption task. In Figure 3a, we have mapped the attention weights of the final layer onto the parent protein and two mutated (i.e. the most and least absorbent) versions of this protein. In Figure 3a, we can see which parts of the protein and what specific mutations (i.e. red lettering) contribute the most to the final vector representation. Moreover, in Figure 3b, by taking an average over each head of the BERT model, we can ascertain the critical parts of the protein within each layer.

## IV. Conclusion

In this work, we have illustrated how pre-training can be utilised for robust modelling of a protein's functional properties, and with some additional fine-tuning through the use of a triplet-network, these models can be further improved. From the results, the triplet-BERT network produced more detailed encodings in each downstream task when compared to the original pre-trained BERT encodings and previous baselines. When using both strategies of pre-training and metric learning, we observed state-of-the-art results for all downstream tasks when compared to using just one of these approaches. This investigation has shown that deep learning can still be applied to produce state-of-art regardless of the limited number of examples within the dataset. More specifically, we have highlighted the potential for pre-training and metric learning within the field of proteomics. By visualising the intermediate features generated by the BERT model, we also provided insight into the function of a protein as we measured the impact of specific modifications featured in all four downstream tasks.

As modern sequencing technology continues to improve proteomics to provide more data on the properties of a protein, it will become paramount to link these extensive resources to specific tasks through the use of techniques such as pre-training and metric learning. In future work, we postulate that subword encodings could improve the encodings generated during pre-training by the BERT model. This will allow the network to learn a far more substantial vocabulary for each protein and will reduce the overall sequence length of the protein, which in turn will reduce the time and cost required to perform pre-training.

## References

[1] Ethan C Alley et al. "Unified rational protein engineering with sequence-only deep representation learning". In: *bioRxiv* (2019), p. 589333.

[2] Tristan Bepler and Bonnie Berger. "Learning protein sequence embeddings using information from structure". In: *arXiv preprint arXiv:1902.08661* (2019).

[3] Jane Bromley et al. "Signature verification using a" siamese" time delay neural network". In: *Advances in neural information processing systems*. 1994, pp. 737–744.

[4] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. IEEE. 2005, pp. 539–546.

[5] UniProt Consortium. "UniProt: a hub for protein information". In: *Nucleic acids research* 43.D1 (2014), pp. D204–D212.

[6] Zihang Dai et al. "Transformer-xl: Attentive language models beyond a fixed-length context". In: *arXiv preprint arXiv:1901.02860* (2019).

[7] Jia Deng et al. "Imagenet: A large-scale hierarchical image database". In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.

[8] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[9] Xingping Dong and Jianbing Shen. "Triplet loss in siamese network for object tracking". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 459–474.

[10] Philip Gage. "A new algorithm for data compression". In: *The C Users Journal* 12.2 (1994), pp. 23–38.

[11] Sara El-Gebali et al. "The Pfam protein families database in 2019". In: *Nucleic acids research* 47.D1 (2018), pp. D427–D432.

[12] Raia Hadsell, Sumit Chopra, and Yann LeCun. "Dimensionality reduction by learning an invariant mapping". In: *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. Vol. 2. IEEE. 2006, pp. 1735–1742.

[13] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[14] Sepp Hochreiter and Jürgen Schmidhuber. "Long short-term memory". In: *Neural computation* 9.8 (1997), pp. 1735–1780.

[15] Elad Hoffer and Nir Ailon. "Deep metric learning using triplet network". In: *International Workshop on Similarity-Based Pattern Recognition*. Springer. 2015, pp. 84–92.

[16] Jeremy Howard and Sebastian Ruder. "Universal language model fine-tuning for text classification". In: *arXiv preprint arXiv:1801.06146* (2018).

[17] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. "Deep metric learning for visual tracking". In: *IEEE Transactions on Circuits and Systems for Video Technology* 26.11 (2015), pp. 2056–2068.

[18] Kishore Jaganathan et al. "Predicting splicing from primary sequence with deep learning". In: *Cell* 176.3 (2019), pp. 535–548.

[19] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition". In: *ICML deep learning workshop*. Vol. 2. 2015.

[20] Ben Krause et al. "Multiplicative LSTM for sequence modelling". In: *arXiv preprint arXiv:1609.07959* (2016).

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[22] Taku Kudo. "Subword regularization: Improving neural network translation models with multiple subword candidates". In: *arXiv preprint arXiv:1804.10959* (2018).

[23] Quoc Le and Tomas Mikolov. "Distributed representations of sentences and documents". In: *International conference on machine learning*. 2014, pp. 1188–1196.

[24] Mark Lennox, Neil Robertson, and Barry Devereux. "Deep Metric Learning for Proteomics". In: *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2020, pp. 308–313.

[25] Mark Lennox, Neil Robertson, and Barry Devereux. "Deep Metric Learning for Proteomics". In: *Submitted to: The International Conference on Machine Learning and Applications* (2020).

[26] Mark Lennox, Neil Robertson, and Barry Devereux. "Expanding the Vocabulary of a Protein: Application of Subword Algorithms to Protein Sequence Modelling". In: *2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE. 2020, pp. 2361–2367.

[27] Mark Lennox, Neil Robertson, and Barry Devereux. "Expanding the Vocabulary of a Protein: Application of Subword Algorithms to Protein Sequence Modelling". In: *The Annual International Conference of the IEEE Engineering in Medicine and Biology Society* (2020).

[28] Jiwen Lu et al. "Multi-manifold deep metric learning for image set classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1137–1145.

[29] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[30] Seonwoo Min et al. "Pre-Training of Deep Bidirectional Protein Sequence Representations with Structural Information". In: *arXiv preprint arXiv:1912.05625* (2019).

[31] Matthew E Peters et al. "Deep contextualized word representations". In: *arXiv preprint arXiv:1802.05365* (2018).

[32] Alec Radford et al. "Improving language understanding by generative pre-training". In: *URL https://s3-us-west-2. amazonaws. com/openai-assets/researchcovers/languageunsupervised/language understanding paper. pdf* (2018).

[33] Roshan Rao et al. "Evaluating Protein Transfer Learning with TAPE". In: *arXiv preprint arXiv:1906.08230* (2019).

[34] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer School on Machine Learning*. Springer. 2003, pp. 63–71.

[35] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems*. 2015, pp. 91–99.

[36] Florian Schroff, Dmitry Kalenichenko, and James Philbin. "Facenet: A unified embedding for face recognition and clustering". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 815–823.

[37] Karen Simonyan and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition". In: *arXiv preprint arXiv:1409.1556* (2014).

[38] Jake Snell, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 4077–4087.

[39] Tanlin Sun et al. "Sequence-based prediction of protein protein interaction using a deep-learning algorithm". In: *BMC bioinformatics* 18.1 (2017), p. 277.

[40] Christian Szegedy et al. "Inception-v4, inception-resnet and the impact of residual connections on learning". In: *Thirty-first AAAI conference on artificial intelligence*. 2017.

[41] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems*. 2017, pp. 5998–6008.

[42] Jesse Vig et al. "BERTology Meets Biology: Interpreting Attention in Protein Language Models". In: *arXiv preprint arXiv:2006.15222* (2020).

[43] Oriol Vinyals et al. "Matching networks for one shot learning". In: *Advances in neural information processing systems*. 2016, pp. 3630–3638.

[44] Leyi Wei et al. "Prediction of human protein subcellular localization using deep learning". In: *Journal of Parallel and Distributed Computing* 117 (2018), pp. 212–217.

[45] Kilian Q Weinberger and Lawrence K Saul. "Distance metric learning for large margin nearest neighbor classification". In: *Journal of Machine Learning Research* 10.Feb (2009), pp. 207–244.

[46] Zachary Wu et al. "Machine learning-assisted directed protein evolution with combinatorial libraries". In: *Proceedings of the National Academy of Sciences* 116.18 (2019), pp. 8852–8858.

[47] Kevin K Yang, Zachary Wu, and Frances H Arnold. "Machine-learning-guided directed evolution for protein engineering". In: *Nature methods* 16.8 (2019), pp. 687–694.

[48] Kevin K Yang et al. "Learned protein embeddings for machine learning". In: *Bioinformatics* 34.15 (2018), pp. 2642–2648.

[49] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems*. 2014, pp. 3320–3328.

[50] Fisher Yu, Vladlen Koltun, and Thomas Funkhouser. "Dilated residual networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 472–480.