Vilnius University

Mathematics and Informatics Faculty

Insitute of Informatics

Bioinformatics study program

Protein thermostability prediction using sequence representations from protein language models

Author: Ieva Pudžiuvelytė

Supervisor: Kliment Olechnovič, PhD

Course thesis

Vilnius, 2022

Contents

1 Introduction

The variety of proteins is no less diverse than the variety of organisms. Just as the latter set is divided into domains, there are different attempts to classify proteins into distinct subsets. One way is to consider the heat-resistance property of biological macromolecules, which is an important trait for practical applications in biotechnology. It is often required to find proteins that not only perform the required function, yet also are thermostable - their structure does not denature at the higher temperature.

Determination of protein's of intereset thermostability can be made experimentally in the wet lab; by searching for its sequence in the proteome of the known thermophilic organism; by developing methods to predict thermostability from the protein's sequence or structural model. The third choice overcomes downsides of the first two methods: high expenses and slow progress, and limitation of knowledge about organisms recorded in databases.

Earlier studies show that protein's sequence and structural properties influence the thermostability of the macromolecule [1]. Furthermore, one of the most recent achievements in the field of deep learning are transformer architecture-based language models or, particularly, protein language models that have not yet been used to classify proteins based on their thermostability. Therefore, it was decided to apply protein representations from the protein language model to make inferences about thermostability of the biological macromolecules.

There are transformer architecture-based language models trained in an unsupervised fashion to predict probabilities of elements in sequences [2]. One of the methods to train the model in an unsupervised way is masked-language modeling (MLM). The method includes masking some of the tokens from the input setting a goal for the model, which is being trained, to correctly predict the masked parts of the input with a reference only to the information in the context. Simultaneously, the process of training creates sequences' embeddings – the real numbered vectors that represent semantic connections of language components. These representations can be transferred as input to specific application models trained using a supervised learning method to complete the defined task, such as the classification problem. Unsurprisingly, the transition between two types of learning has a name of 'transfer learning'.

This separation is practically useful because the computationally-heavy task to train the language model can be excluded from the development of the application model.

Since proteins can be represented in amino acid sequences assumed as a particular language, there are protein language models – models trained on protein sequences – that provide embeddings as output. The multi-dimensional vectors are transferred for the application neural network as its input to observe the results and decide whether the computed representations are suitable to solve the specific biological task.

This work presents a novel way to predict thermal stability of proteins. The solution is a feed-forward neural network (FNN). To train the FNN, the evolutionary scale model 1b (ESM-1b)[3] is used to generate embeddings for proteins of organisms with annotated growth temperatures [4]. The model takes the generated embedding to predict the thermostability class of the input protein.

2 Theory

The main objective of the work is to apply embeddings from protein language models to create a classifier that could determine to which thermostability class the protein sequence belongs.

Thermostability classes were created by setting the threshold of 65 °C - proteins that are stable at temperatures strictly lower than the given threshold should be assigned to class '0' and the remaining ones compose the class '1'.

2.1 Protein language models

Protein language models are transformer models trained on protein sequences. The transformer is a model, which is made of encoder-decoder architecture that relies entirely on self-attention [5]. The encoder part provides continuous representations of the input composed of sequences of symbols, meanwhile the decoder part generates output for each symbol in the input sequence. For this principle of architecture, transformers can be trained in an unsupervised fashion and be applies to natural language processing (NLP) tasks at which they produce state-of-the-art results [6].

Since amino acid sequences can be considered as a particular language, transformer architectures were applied to solve tasks related to protein biology or molecule modeling *in silico*. Attention mechanisms in models of transformer architecture, taking BERT-like model as a example [7], are capable to capture the folding structure, binding sites, and complex biophysical properties of proteins.

2.2 ESM-1b embeddings

Due to the novelty of embeddings, a considerably good performance of protein language models, and a recently emerged availability of embeddings, it was decided to develop a neural network model that would take protein embeddings as input and give the thermostability class label as output.

ESM-1b is one of evolutionary scale models trained by Facebook Research [3]. The model has 33 layers and 650 million parameters. The model was trained in an unsupervised fashion on UniRef50 data set (accessed March 28, 2018)[8]. In order to ensure determinism in the validation set, authors removed protein sequences that were longer than 1024 amino acids.

The authors made a script to extract model's embeddings available in the repository "Evolutionary Scale Modeling". The script allows to choose from which model and layer embeddings will be taken, what embeddings (mean, per amino acid, or beginning of the sequence token) to keep. In the result of using the script, a 1280 dimensional vector for each protein is generated.

The fact that sequences longer than 1024 amino acids were removed from the validation data set for ESM-1b model's training implies to the limitation of model's embeddings, which cannot be generated for sequences longer than 1024 amino acids.

3 Methods

Generally, the development of a tool for protein thermostability predictions consisted of the following steps:

1. Collecting sets of sequences
2. Calculating ESM-1b embeddings for the sets of sequences
3. Processing the set of generated embeddings
4. Visualising ESM-1b embeddings
5. Training and validating the neural network model of the chosen architecture
6. Testing the trained neural network model
7. Presenting the results of the model

The workflow will be reviewed in terms of these points.

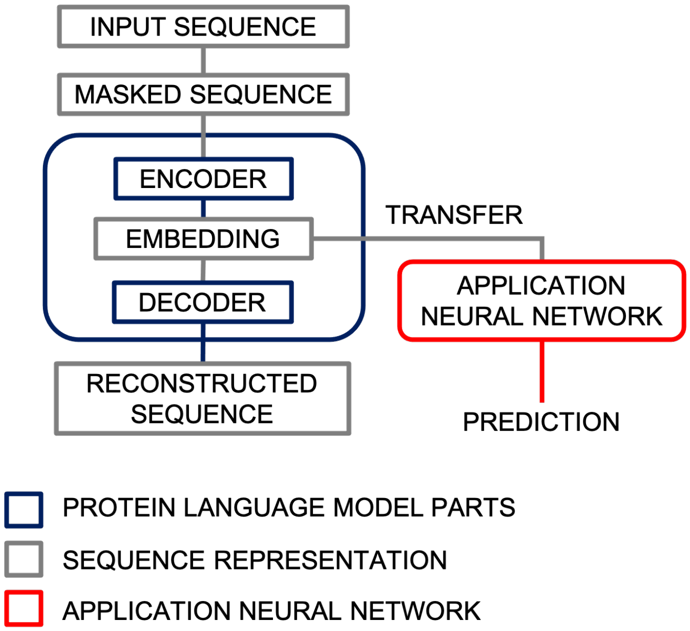

Figure 1: The scheme of embeddings from protein language model usage in the application neural network model

3.1 Collecting data

Initially, the steps prior to the neural network model construction were done using a small data set that was composed of proteomes of two organisms: a mesophilic bacteria *Escherichia coli* (UP000000625) and a thermophilic archaeon *Sulfolobus solfataricus* (UP000001974). The growth temperature of *E. coli* is 37 °C [9] and 80 °C for *S. solfataricus* [10]. This data set was named '001' and used only for embeddings visualisation.

The visualisation of '001' stimulated to check whether embeddings are distinguished by the thermostability property or the life domain has a significant impact to the data clusterisation. Subsequently, '002' data set was a collection of 2 mesophilic archaea and 2 thermophilic bacteria proteomes.

Mesophilic archaea:

- *Methanobrevibacter oralis* (UP000077428)
- *Nitrosopumilus maritimus* strain SCM1 (UP000000792)

Thermophilic bacteria:

- *Aquifex aeolicus* (strain VF5) (UP000000798)
- *Thermotoga maritima* (strain ATCC 43589 / DSM 3109 / JCM 10099 / NBRC 100826 / MSB8) (UP000008183)

After receiving the results of the first neural network model trained on '002' data set, it was decided to train and test the model on the first real data set. '003' data set was a subset of the data set of 21458 annotated organisms [4].

Requirements for '003' data set were: the data set needed to be balanced and a single taxonomy identifier could be apparent only in either training, validation, or testing data set.

To make the data set balanced, 216595 sequences from 51 proteomes taken for the class '0' and 212729 sequences from 111 proteomes of the class '1' (Table 1, Table 2).

Proportions that were chosen to divide classes of '003' data set were 70%, 15%, and 15% for training, validation, and testing sets respectively.

Table 1: Numbers of different proteomes that compose training, validation, and testing data sets

	Class '0'	Class '1'
Training	32	77
Validation	8	17
Testing	11	17

Table 2: Numbers of amino acid sequences that compose training, validation, and testing data sets

	Class '0'	Class '1'
Training	145128	143868
Validation	33204	32616
Testing	38263	36245

3.2 Calculating embeddings

The final layer of ESM-1b model was used to generate protein embeddings taken as input to the classification neural network. For training, validation, and initial testing stages embeddings averaged over the full sequence were chosen. These embeddings were vectors of 1280 dimensions.

3.3 Processing generated embeddings

As it was mentioned in the previous section, ESM-1b embeddings can be calculated only for sequences that are not longer than 1024 amino acids. Therefore, after the generation of embeddings, sequences without embeddings were filtered out from the initial data set. The final data set consisted of 423127 embeddings (212144 of class '0' and 210983 of class '1') (Table 3).

Table 3: Numbers of embeddings that compose training, validation, and testing data sets

	Class '0'	Class '1'
Training	141602	142707
Validation	32793	32363
Testing	37749	35913

These embeddings were saved to NPZ and TSV files. NPZ files are binary files that are used to load sequence representations to the model. TSV files with embedding vectors were created for a human-readable record and analysis with other tools. The TSV files were made headerless with columns representing the following information:

- #0 - Taxonomy ID of the organism, to which the sequence belongs
- #1 - Accession ID of the sequence
- #2 - Length of the sequence
- #3 - Temperature label
- #4 - #1284 Components of embeddings

3.4 Visualising embeddings

Calculated embeddings were visualised using principal component analysis (PCA) from *Scikit-Learn Python* library (version 0.24.2) (Figure 2). The plots demonstrated that embeddings form clusters that might be divided to distinct classes, which promised the high performance of the trained neural network.

Additionally, *Python* minimum-distortion embedding (*PyMDE*) library (version 0.1.5) was used to visualise data sets of embeddings (Figure 3), which showed the 3D image of data clusters.

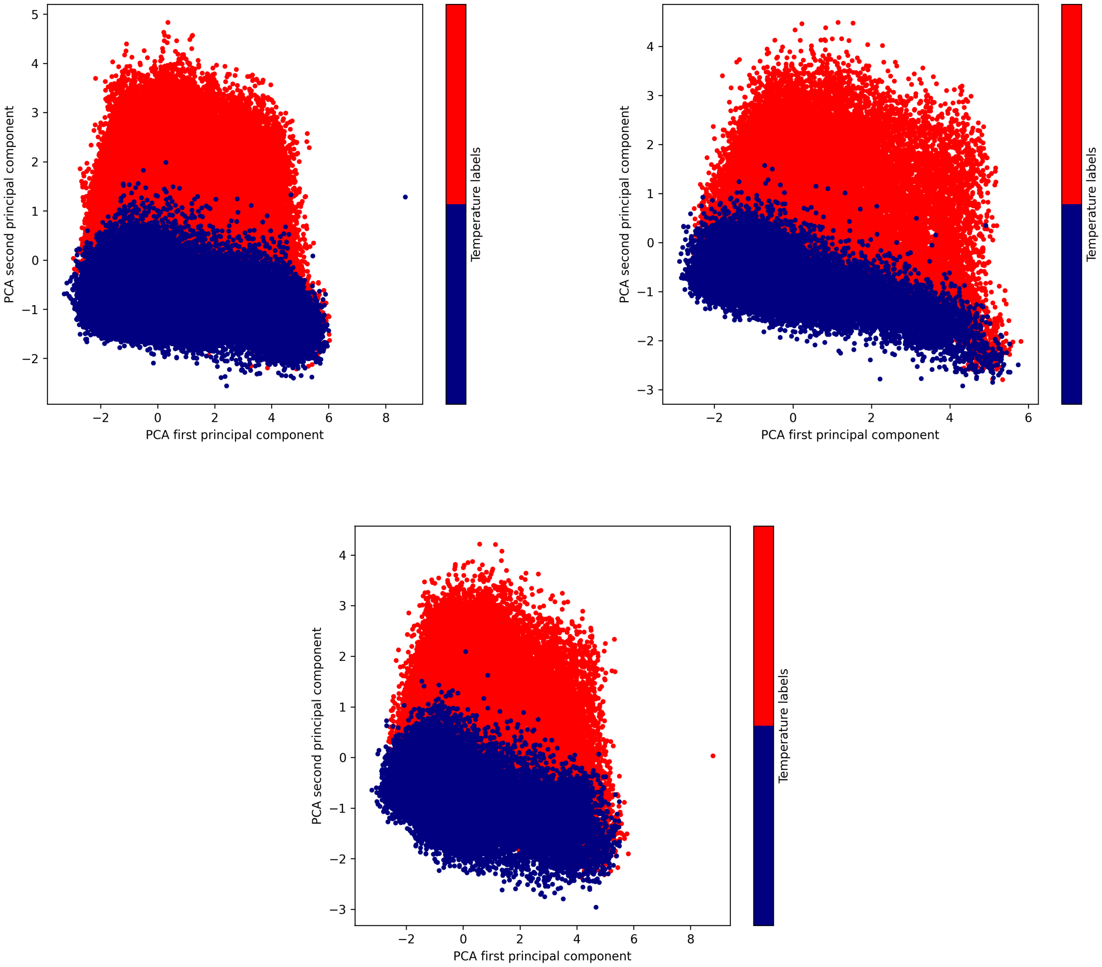

Figure 2: PCA visualisations of '003' model data sets' embeddings: training set in the upper left, validation set in the upper right, and testing set in the lower center

3.5 Training and validation of the neural network

At first, it was decided to try training the neural network built of the most simple architecture - a single layer perceptron (SLP). The input of the layer is 1280-dimensional - it takes the input of protein embeddings - and the output is a binary label that represents the thermostability class. The activation and loss functions were sigmoid and binary cross entropy respectively. Adam optimizer [11] with learning rate of 0.0001 was chosen.

The model was trained and validated in 5 epochs taking mini-batch size of 24 embeddings. The ROC curves (Figure 4) did not indicate a significant progress of the training process - the area under the curve was considerably large since the first epoch of the training.

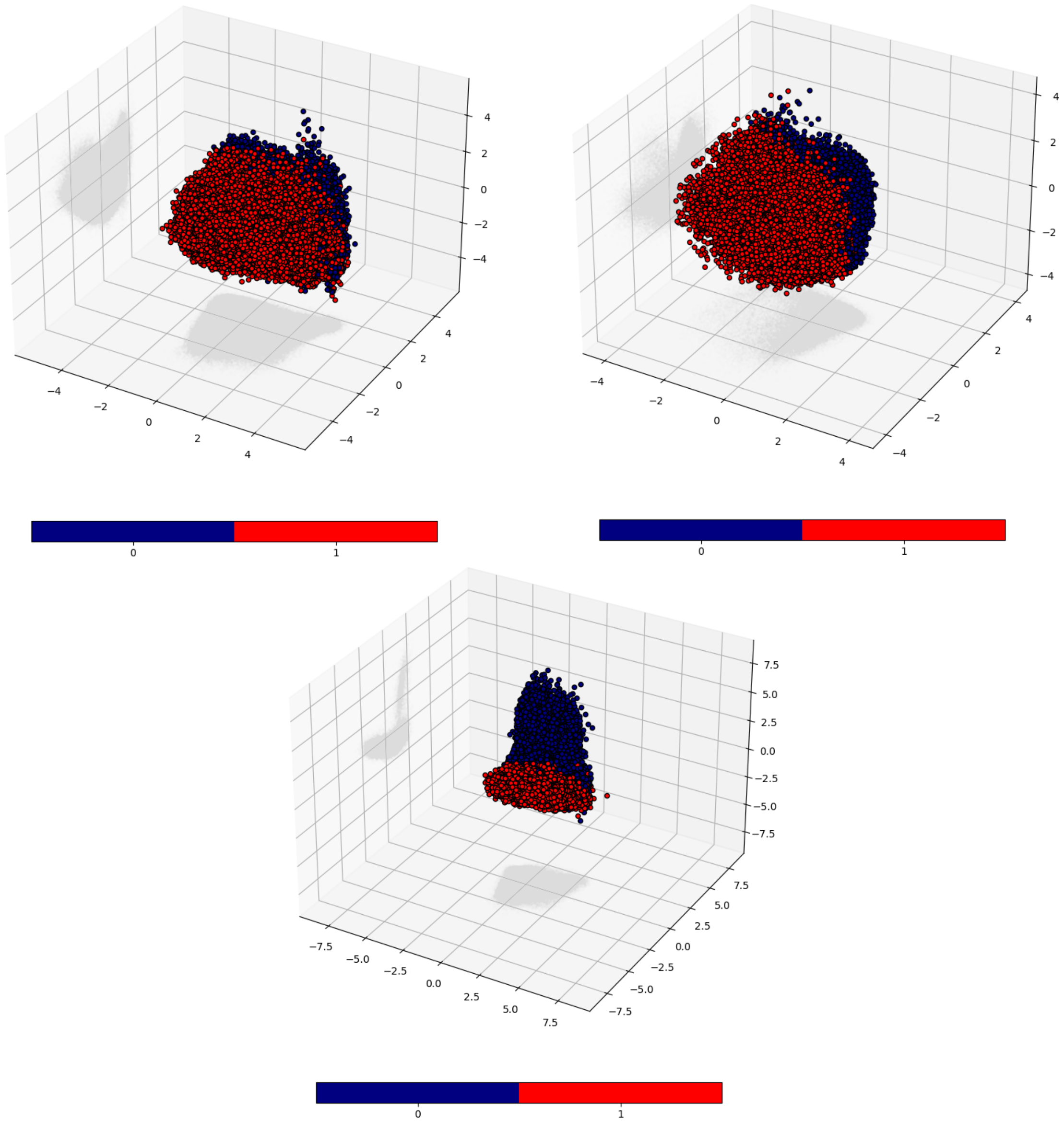

Figure 3: *PyMDE* PCA visualisations of '003' model data sets' embeddings: training set in the upper left, validation set in the upper right, and testing set in the lower center

Confusion matrices presented in this work are composed of values in the following order reading from left to right: the first row consists of true negative (TN) and false positive (FP) values, the second row consists of false negative (FN) and true positive (TP) values. Therefore, the values of the confusion matrix after the fifth training epoch showed that most of the mistakes were made by assigning false positive labels to sequences.

Values in confusion matrix allow to calculate metrics: accuracy, precision, and recall. Formulas for calculation of these metrics are given below (Eq. 1-3). The AUC is calculated

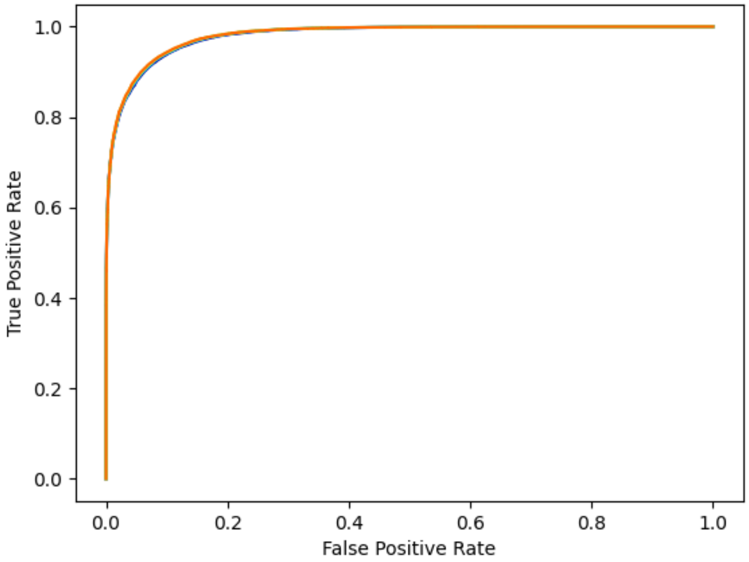

Figure 4: ROC curves for each validation epoch - the orange one denotes the curve of the fifth epoch

Table 4: Confusion matrix after the fifth validation epoch

	0	1
0	32793	3636
1	1582	30770

by integrating the ROC curve.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

Table 5: Model performance metrics after the fifth validation epoch

Accuracy	0.91989
Precision	0.89432
Recall	0.9511
ROC AUC	0.92

4 Results

The model was tested with the testing subset of '003' data set that was not exposed to the model before the testing stage.

The ROC curve (Figure 5) was steep enough to show a good performance of the model. The confusion matrix (Table 6) of the testing stage showed balanced number of mistakes (FP and FN values were approximately equal). Area under testing ROC curve was 0.924, accuracy and precision metrics exceeded those after the fifth training epoch (Table 7). In addition to the metrics given for the fifth training epoch, Matthew's correlation coefficient (MCC) (4) was calculated as well.

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4}$$

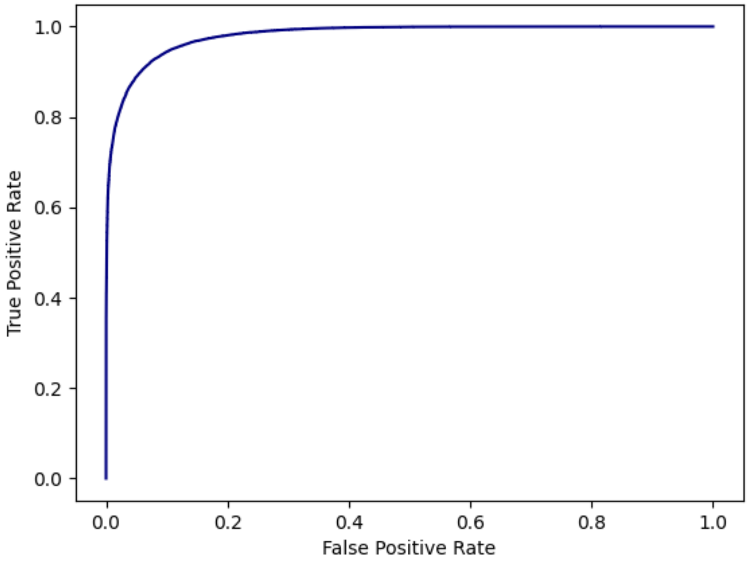

Figure 5: ROC curve after the testing stage

Table 6: Confusion matrix in the testing phase

	0	1
0	34933	2813
1	2788	33122

Table 7: Comparison of model performance metrics between results of the fifth epoch of the validation stage and testing phase

	Validation 5th	Testing
Accuracy	0.91989	0.924
Precision	0.89432	0.922
Recall	0.9511	0.922
ROC AUC	0.92	0.924
MCC	-	0.845

5 Conclusions

In summary, the objective to develop a tool that predicts the thermostability class of the input protein was achieved. Nevertheless, the workflow pointed several areas to consider for improvement and future development:

- Overcoming the limitation of sequence length of 1024 amino acids
- Experimentation with more sophisticated model architectures
- Testing the trained model with per token embeddings and getting thermostability predictions for each amino acid in the sequence
- Picking a different subset of the annotated growth temperatures data set
- Taking a different kind of embeddings as input

References

[1] H. P. Modarres, M. Mofrad, and A. Sanati-Nezhad, “Protein thermostability engineering,” *RSC advances*, vol. 6, no. 116, pp. 115252–115270, 2016.

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

[3] A. Rives, J. Meier, T. Sercu, S. Goyal, Z. Lin, J. Liu, D. Guo, M. Ott, C. L. Zitnick, J. Ma, *et al.*, “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences,” *Proceedings of the National Academy of Sciences*, vol. 118, no. 15, 2021.

[4] M. K. M. Engqvist, “Growth temperatures for 21,498 microorganisms,” Feb. 2018.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” *Advances in neural information processing systems*, vol. 30, 2017.

[6] J. Vig and Y. Belinkov, “Analyzing the structure of attention in a transformer language model,” *arXiv preprint arXiv:1906.04284*, 2019.

[7] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher, and N. F. Rajani, “Bertology meets biology: Interpreting attention in protein language models,” *arXiv preprint arXiv:2006.15222*, 2020.

[8] B. E. Suzek, Y. Wang, H. Huang, P. B. McGarvey, C. H. Wu, and U. Consortium, “Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches,” *Bioinformatics*, vol. 31, no. 6, pp. 926–932, 2015.

[9] J. Jang, H.-G. Hur, M. J. Sadowsky, M. Byappanahalli, T. Yan, and S. Ishii, “Environmental escherichia coli: ecology and public health implications—a review,” *Journal of applied microbiology*, vol. 123, no. 3, pp. 570–581, 2017.

[10] M. Zaparty, D. Esser, S. Gertig, P. Haferkamp, T. Kouril, A. Manica, T. K. Pham, J. Reimann, K. Schreiber, P. Sierocinski, *et al.*, "“hot standards” for the thermoacidophilic archaeon sulfolobus solfataricus," *Extremophiles*, vol. 14, no. 1, pp. 119–142, 2010.

[11] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.