# Explaining embedding results for scoring alignments

Riley Gavigan
Thesis Supervisor: Lucian Ilie, Course Instructor: Nazim Madhavji
Department of Computer Science, University of Western Ontario, London, N6A 5B7, Ontario, Canada

February 1, 2024

**Abstract**

*Abstract Content Here*

## 1 Introduction

Proteins are one of the four molecules of life. Finding similarities among protein sequences is essential in identifying protein structure and function. This is done by computing alignments between sequences. The BLAST program[1] is one of the most widely used tools in science [1]. An essential part of BLAST is the scoring function; the most widely used functions are provided by the BLOSUM matrices [2].

Sequence similarity is essential in sequence analysis for DNA, RNA, and peptide sequences [3]. Peptide sequence alignment is the most complex case, with a language of 20 common amino acids forming a theoretically countably infinite amount of unique peptide sequences shown in Equation 1 by taking the Cartesian product.

$$Theoretical\ Limit = \prod_{k=1}^{\infty} |A| = \prod_{k=1}^{\infty} 20 = 20 \times 20 \times \dots \tag{1}$$

While there is theoretically a countably infinite number of peptide sequences, the observed sequences in living organisms are constrained by biological, genetic, and functional factors. For example, the average eukaryotic protein size is $353 \pm 62.5$ residues [4].

The $E$-score protein alignment scoring method [5] outperforms state-of-the-art methods, supported by comparing ProtT5 [6] $E$-score results with BLOSUM45 [2, 5].

$E$-score uses Transformer models to produce contextual embeddings for the residues in peptide sequences. Model information is available in Table 1. These models are based off of their Natural Language Processing (NLP) equivalents [7, 8, 9, 10, 11].

Contextual embeddings are embeddings produced by the self-attention mechanism in the Transformer architecture [12], which is shown in Figure 1. Similar to word embeddings in NLP, they describe the position of a residue in a high-dimensional vector space.

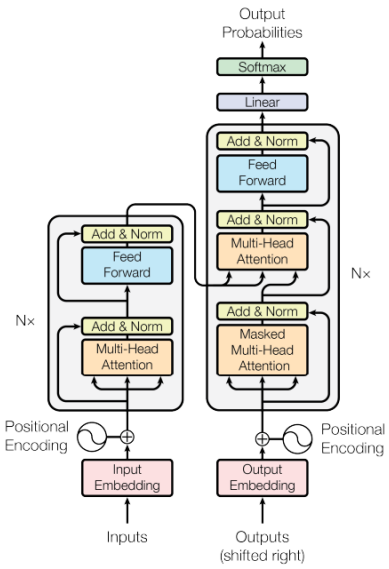


**Figure 1:** *Transformer model architecture [12]. Encoder is on the left; decoder is on the right.*

Contextual embeddings produced for protein sequences have many important applications in biology, including structure prediction [13, 14, 15] and function prediction [16, 17, 18].

The $E$-score alignment method is another application for these embeddings, outperforming the state-of-the-art methods [5] by completely changing the way alignments are computed.

The embedding vector produced for each protein residue varies based on the model that was used. For example, the embedding for a protein sequence of 310 residues using ProtT5 will have the dimensions [310, 1024]. The embedding dimensions are outlined in Table 1. The dimensionality of the embedding vectors represents the number of features encoded in the embedding, and is a fixed value for a given model.

The embeddings produced by a model for a protein

[1] Exceeds 108,000 citations, according to Google Scholar.

**Table 1:** *Transformer models available in the E-score method; n = number of residues. ProtT5, ProtBert, ProtAlbert, and ProtXLNet come from ProtTrans [6]. ESM1b and ESM2 come from the Meta Fundamental AI Research Protein Team [11].*

| Model | Architecture | Embedding Dim | Pre-Trained Dataset | Training Method |
|---|---|---|---|---|
| ProtT5 | Encoder-Decoder | n * 1024 | UniRef50 | Text-to-Text |
| ESM1b | Encoder | n * 1280 | UniRef50 | Masked Language Modeling |
| ESM2 | Encoder | n * 1280 | UniRef50 | Masked Language Modeling |
| ProtBert | Encoder | n * 1024 | UniRef100 | Masked Language Modeling |
| ProtAlbert | Encoder | n * 4096 | UniRef100 | Masked Language Modeling |
| ProtXLNet | Decoder | n * 1024 | UniRef100 | Permutation Language Modeling |

$P$, calculated in Equation 2, are used as the input to calculate the cosine similarity.

$$E(P) = GetEmbeddings(Model = ProtT5) \quad (2)$$

Calculating the cosine similarity between two vectors $A = (A_i)_{i=1..n}$ and $B = (B_i)_{i=1..n}$ is shown in Equation 3.

$$CosSim(A, B) = cos(\theta) \equiv \frac{A \cdot B}{\|A\|\|B\|} \equiv \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}} \quad (3)$$

$E$-score is calculated by taking the cosine similarity between the embedding vector for two residues $(i, j)$, shown in Equation 4 where $P_1$ and $P_2$ are proteins [5].

$$E\text{-}score(i, j) = CosSim(E(P_1)_i, E(P_2)_j) \quad (4)$$

In calculating sequence alignment using the $E$-score method, the cosine similarity results were mostly mostly less than $\frac{\pi}{2}$. It was also determined that ProtT5 performed better than the other models [5].

In this paper, I explain significant qualitative and quantitative differences and similarities between the models in Table 1. Combined with visualization and analysis of embedding vector and cosine similarity distributions, I propose the contributing factors to better $E$-score performance.

Inference is applied to describe the procedure and techniques for fine-tuning ProtT5 and other models to produce better embeddings for alignment.

# 2 Materials and Methods

## Transformers

T5 is an encoder-decoder model; it uses both the encoder and decoder of the Transformer architecture. Explain how this excels and how it relates to E-Score and why T5 performs best. Mention the dimension of embeddings. Mention the training procedure.

BERT and ALBERT are auto-encoder models; they only use the encoder of the Transformer architecture. Explain how this excels and how it relates to their performance. Mention the dimension of embeddings. Mention the training procedure.

XLNet is an auto-regressive model; it only uses the decoder of the Transformer architecture. Explain how this excels and how it relates to their performance. Mention the dimension of embeddings. Mention the training procedure.

ESM1b and ESM2 are also auto-encoder models; they only use the encoder of the Transformer architecture and are based on RoBERTa. Mention the dimension of embeddings. Mention the training procedure.

Compare BERT, ALBERT, ESM1b and ESM2 because they are all auto-encoding models. Their differences and similarities, and their E-score results observed.

## Embedding vectors

### Data

Embedding vectors will be analyzed between the available models on this representative data.

### Normalizing vector results

### Visualizing normalized results

### Analyzing visualizations and data

## Cosine similarity

### Data

Cosine similarity will be analyzed between embedding vectors between different models on this representative data.

**Visualizing cosine similarity**

**Analyzing visualizations and data**

### Fine-tuning ProtT5

**Data**

ProtT5 will be fine-tuned with representative data from...

**PEFT and LoRA**

**Compute Power**

**Intuition and reasoning**

## 3 Results

### Embedding distributions

### Cosine similarity

### Model performance

### Fine-tuning

## 4 Discussion and Conclusion

## 5 Acknowledgements

Dr. Lucian Ilie provided thesis supervision and guidance. This research built upon the initial *E*-score method research [5].

## 6 Supplementary information

The proposal for this research was completed and approved November 2023, and can be found online[2].

A supplementary file containing Supplementary Tables 1-N is available online[3].

## References

[1] S. F. Altschul et al. "Basic local alignment search tool". In: *Journal of molecular biology* (1990). DOI: 10.1016/S0022-2836(05)80360-2.

[2] S. Henikoff and J. G. Henikoff. "Amino acid substitution matrices from protein blocks". In: *Proceedings of the National Academy of Sciences* (1992). DOI: 10.1073/pnas.89.22.10915.

[3] Dan Ofer, Nadav Brandes, and Michal Linial. "The language of proteins: NLP, machine learning protein sequences". In: *Computational and Structural Biotechnology Journal* 19 (2021), pp. 1750–1758. ISSN: 2001-0370. DOI: https://doi.org/10.1016/j.csbj.2021.03.022.

[4] Yannis Nevers et al. "Protein length distribution is remarkably uniform across the tree of life". In: *Genome Biology* 24.1 (2023), p. 135. ISSN: 1474-760X. DOI: 10.1186/s13059-023-02973-2. URL: https://doi.org/10.1186/s13059-023-02973-2.

[5] S. Ashrafzadeh et al. "Scoring alignments by embedding vector similarity". In: (2023). DOI: 10.1101/2023.08.30.555602.

[6] A. Elnaggar et al. "ProtTrans: Towards Cracking the Language of Lifes Code Through Self-Supervised Deep Learning and High Performance Computing". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2021). DOI: 10.1109/TPAMI.2021.3095381.

[7] C. Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: (2020). DOI: 10.48550/arXiv.1910.10683.

[8] J. Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: (2018). DOI: 10.48550/arXiv.1810.04805.

[9] Zhenzhong Lan et al. *ALBERT: A Lite BERT for Self-supervised Learning of Language Representations*. 2020. arXiv: 1909.11942 [cs.CL].

[10] Z. Yang et al. "XLNet: Generalized autoregressive pretraining for language understanding". In: *Advances in neural information processing systems* (2019). DOI: 10.48550/arXiv.1906.08237.

[11] A. Rives et al. "Biological Structure and Function Emerge from Scaling Unsupervised Learning to 250 Million Protein Sequences". In: *PNAS* (2019). DOI: 10.1101/622803.

[12] A. Vaswani et al. "Attention is all you need". In: (2017). DOI: 10.48550/arXiv.1706.03762.

[13] Andrew W. Senior et al. "Improved protein structure prediction using potentials from deep learning". In: *Nature* 577.7792 (2020), pp. 706–710. DOI: 10.1038/s41586-019-1923-7. URL: https://doi.org/10.1038/s41586-019-1923-7.

[14] Jianyi Yang et al. "Improved protein structure prediction using predicted inter-residue orientations". In: *bioRxiv* (2019). DOI: 10.1101/846279.

[2] https://www.rileygavigan.com/e-score-proposal.pdf
[3] https://www.rileygavigan.com/e-score-data.pdf

[15] John Jumper et al. "Highly accurate protein structure prediction with AlphaFold". In: *Nature* 596.7873 (2021), pp. 583–589. ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2.

[16] Maxat Kulmanov and Robert Hoehndorf. "DeepGOPlus: improved protein function prediction from sequence". In: *Bioinformatics* 36.2 (2019), pp. 422–429. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz595.

[17] Vladimir Gligorijević et al. "Structure-based protein function prediction using graph convolutional networks". In: *Nature Communications* 12.1 (2021), p. 3168. ISSN: 2041-1723. DOI: 10.1038/s41467-021-23303-9.

[18] Boqiao Lai and Jinbo Xu. "Accurate Protein Function Prediction via Graph Attention Networks with Predicted Structure Information". In: *bioRxiv* (2021). DOI: 10.1101/2021.06.16.448727.