# Explaining embedding results for scoring alignments

Final Progress Report

by

Riley Gavigan

CS 4490Z
Thesis Supervisor: Lucian Ilie
Course Instructor: Nazim Madhavi

Department of Computer Science
University of Western Ontario, London, N6A 5B7, Ontario, Canada
March 28, 2024

# Glossary

**amino acid** molecules that combine to form proteins, containing both an amino and a carboxyl group 14, 15

**peptide** a compound consisting of two or more amino acids linked in a chain, the carboxyl group of each acid being joined to the amino group of the next 11, 14, 15

**residue** a single unit that makes up a polymer, such as an amino acid in a polypeptide chain 8, 11, 14

**transformer** a type of neural network architecture used to solve the problem of transduction or transformation of input sequences into output sequences in deep learning applications using self-attention 8, 10, 11, 16

# Abbreviations

**ALBERT** A Lite BERT 10

**BERT** Bidirectional Encoder Representations from Transformers 10

**BLAST** Basic Local Alignment Search Tool 10

**BLOSUM** BLOcks SUbstitution Matrix 10

**CDD** Conserved Domain Database 14

**ENNA** Evolutionary Neural Network Algorithm 15

**LLM** Large Language Models 11, 16

**LoRA** Low-Rank Adaptation 11, 19

**MSA** Multiple Sequence Alignments 14, 16, 19

**NLP** Natural Language Processing 5, 8, 13, 19

**RoBERTa** Robustly Optimized BERT 10

**T5** Text-To-Text Transfer Transformer 10

# Structured Abstract

### Context and motivation

The *E*-score protein alignment scoring method (Ashrafzadeh et al., 2023) outperforms state-of-the-art methods, supported by comparing ProtT5 (Elnaggar et al., 2021) *E*-score results with BLOSUM45 (Henikoff & Henikoff, 1992).

This research aimed to understand *E*-score results, building upon the observation that mean cosine similarity results between two embeddings are not evenly distributed.

By understanding the underlying causes of the observed results, we can improve the *E*-score method. Insights can be used to fine-tune the transformer models (Elnaggar et al., 2021; Rives et al., 2019) and performance of embeddings.

### Research questions

- What properties of embeddings produce better cosine similarity results?
- Why do cosine similarity results primarily fall within a positive range?
- How can models be fine-tuned to produce better embeddings?

### Principal ideas

Positive cosine similarity results imply the produced embeddings are mostly similar. Comparing different embedding types provides insight into their distributions. Through these comparisons, conclusions about properties that improve *E*-score results were drawn.

### Research methodology

This research is a data science investigation to obtain insight about the embeddings and cosine similarity results in the *E*-score method.

### Anticipated results

This study primarily aimed to obtain insight and knowledge for the *E*-score method, specifically:

- Knowledge about the distributions of different embedding types
- Knowledge about the cosine similarity between embeddings
- Insight to fine-tune and improve models

### Novelty

By building upon a novel method for scoring protein alignments using cosine similarity (Ashrafzadeh et al., 2023), novel conclusions about embeddings and cosine similarity were made, leading to further research that can improve embeddings and models.

### Impact

Improvements in transformer models for the *E*-score alignment scoring method can be made through the insight this research found. Improvements may also be applicable to Natural Language Processing (NLP) Models such as T5 (Raffel et al., 2020).

### Progress and completed work

Insight into properties behind embedding type distributions was obtained. From these properties, cosine similarity results were explained. These properties were explained through conducted research and simulation in combination with insight from biochemical background research.

### Limitations

No limitations are known to exist in this research.

# Table of Contents

# 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 *E*-score method is a method computes alignments between sequences using contextual embeddings produced by transformer models (Ashrafzadeh et al., 2023). This method uses several different transformer models based off of NLP models (Devlin et al., 2018; Lan et al., 2020; Liu et al., 2019; Raffel et al., 2020; Z. Yang et al., 2019).

These transformer models produce embeddings when provided protein sequences. Understanding the values and distributions of these embeddings between each model is one focus of this research (O1).

*E*-score uses cosine similarity to compute similarity between pairs of embeddings for scoring alignments. This research analyzes the distributions of observed cosine similarity results for natural and random protein sequences (O2, O3).

Combining embedding distribution and cosine similarity results with biochemical understandings of proteins is used to draw conclusions about model performance and *E*-score results. Specifically, explanations about why some models outperform other models are derived (O1, O2).

Using inference about the proposed factors contributing to *E*-score performance, I describe the procedure and techniques for fine-tuning ProtT5 to produce better embeddings for the *E*-score method (O4).

Significant results from this research include:

- Significant positive correlation between higher embedding value variance and improved *E*-score performance for a given model.
- Significant positive correlation between average cosine similarity results approaching 0 and improved *E*-score performance for a given model.

Novel implications about model flexibility and fine-tuning models to better adapt to the frequency of residues (or words in NLP) provide significant insight into improving performance of different models for not only *E*-score, but for any method using transformers.

## 1.1 Report structure

Chapter 2 provides a reader with background on important concepts and details discussed later in the thesis. Chapter 3 outlines the objectives of the research. Chapter 4 outlines the materials and methods used in the research conducted on the *E*-score method. Chapter 5 provides the results from analysis performed in the data science investigation. Chapter 6 discusses the results, their implications, limitations, and generalizations. Chapter 7 concludes the study by addressing the research questions outlined in the thesis proposal, and discusses impact and novelty of the results. Chapter 8 discusses potential future work and novel lessons learned from this research.

# Background and Related Work

## 2.1 Natural Language Processing

Natural Language Processing is the branch of artificial intelligence that deals with computers understanding text and spoken words (Khurana et al., 2023). One significant advancement was the introduction of transformers (Vaswani et al., 2017). Before Transformers, methods such as word2vec (Mikolov et al., 2013) and GloVe (Pennington et al., 2014) generated contextually-independent embedding vectors for words. Transformer models introduced contextual embeddings generated through self-attention (Vaswani et al., 2017).

Information about each model serving as foundation for *E*-score models:

- Text-To-Text Transfer Transformer (T5): Text-to-text approach. Input and output are both text strings. Relies of transfer learning for downstream fine-tuning (Raffel et al., 2020). GLUE benchmark average: 88.7
- Bidirectional Encoder Representations from Transformers (BERT): Bidirectional training using masked language modeling for a deeper sense of context from sequential reading (Devlin et al., 2018).
- A Lite BERT (ALBERT): A lightweight version of BERT that uses parameter-reduction techniques to reduce training time and memory limitations (Lan et al., 2020). GLUE benchmark average: 87.3
- Robustly Optimized BERT (RoBERTa): A stronger version of BERT that was trained longer; removed next-sentence pretraining; and trained with larger mini-batches and learning rates (Liu et al., 2019). GLUE benchmark average: 86.4
- XLNet: Designed to overcome the pretrain-finetune discrepency BERT suffered from, outperforming BERT significantly on 20 tasks (Z. Yang et al., 2019). GLUE benchmark average: 87.5

## 2.2 *E*-score

Finding similarities among protein sequences is essential in identifying protein structure and function. This is done by computing alignments between sequences. The Basic Local Alignment Search Tool (BLAST) program[1] is one of the most widely used tools in science (Altschul et al., 1990). An essential part of BLAST is the scoring function; the most widely used functions are provided by the BLOcks SUbstitution Matrix (BLOSUM) (Henikoff and Henikoff, 1992).

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

The $E$-score protein alignment scoring method (Ashrafzadeh et al., 2023) is another one of these scoring functions that outperforms state-of-the-art methods. $E$-score's improved performance was supported by comparing ProtT5 (Elnaggar et al., 2021) results with BLOSUM45 (Ashrafzadeh et al., 2023; Henikoff and Henikoff, 1992). $E$-score uses transformer models to produce contextual embeddings for the residues in peptide sequences. Model information is available in Table 5.3.

Contextual embeddings describe the position of a residue in a high-dimensional vector space. Contextual embeddings have many important applications in biology, including structure prediction (Jumper et al., 2021; Senior et al., 2020; J. Yang et al., 2019) and function prediction (Gligorijević et al., 2021; Kulmanov and Hoehndorf, 2019; Lai and Xu, 2021). The $E$-score alignment method is another application for these embeddings, outperforming the state-of-the-art methods (Ashrafzadeh et al., 2023) by completely changing the way alignments are computed.

The embedding vector produced for each protein residue varies based on the model. Embedding dimensions and pre-training dataset are outlined in the research code repository. The dimensionality of the embedding vectors represents the number of features encoded in the embedding, and is a fixed value for each model.

Calculating the cosine similarity between two vectors $A = (A_i)_{i=1..n}$ and $B = (B_i)_{i=1..n}$: $\frac{A \cdot B}{\|A\|\|B\|}$. $E$-score is calculated by taking the cosine similarity between the embedding vectors from two residues.

In calculating sequence alignment using the $E$-score method, the cosine similarity results were mostly mostly less than $\frac{\pi}{2}$. ProtT5 had the best performance (Ashrafzadeh et al., 2023).

## 2.3 Analysis and Research Gap

There is no research analyzing results and properties contributing to improved embedding performance for comparable models to the $E$-score method using protein transformers. Fine-tuning Large Language Models (LLM) is a powerful technique to leverage pre-trained models and adapt them to perform better at a specific task or tasks. Fine-tuning can be improved upon using insights such as those taken from this research. The purpose of fine-tuning is to avoid the need to pre-train a model from scratch for a task; instead relying on powerful pre-trained models and modifying them to better suit the task.

Supervised learning involves providing the model with a labeled dataset, and the model will learn to map the input to the output by minimizing its loss function (Mohri et al., 2012/2018). Reinforcement learning involves providing a reward signal to the model when it generates a desired output, and the model learns to generate the desired output for a task by maximizing the reward signal (Sutton and Barto, 2018). Both of these tasks can be leveraged along with novel conclusions from this research to better fine-tune models for $E$-score and for other tasks that follow similar procedures to draw unique conclusions. Fine-tuning techniques such as Low-Rank Adaptation (LoRA), a technique that freezes the pre-trained weights and injects a trainable rank decomposition matrix into each layer of the architecture, can minimize compute intensity of fine-tuning procedures that this research can lead to.

# Research Objectives

- O1: Understand the reasoning behind the observed distributions of different embedding types. Explaining both individual and relative results for *E*-score models.
- O2: Understand what properties of embeddings help produce better cosine similarity and alignment results.
- O3: Understand why cosine similarity results primarily fall within a positive range.
- O4: Determine how models can be fine-tuned to improve *E*-score method results.

# Methodology

## 4.1 Protein composition

Proteins are not completely random in nature. By determining the frequencies of amino acids in our dataset of protein sequences, we showcase that there is not an equal distribution of amino acids present in nature. We also use these frequencies to perform a simulation on completely random proteins for a given length *n* of a polypeptide. By simulating every combination and calculating the cosine similarity for a given length of proteins using only the frequency of amino acids as a constraint, we are able to outline one factor contributing to observed cosine similarity results (O1).

## 4.2 Embeddings and cosine similarity

By applying the above analysis and further supporting it with more properties of proteins such as their secondary structures, we analyze and explain why cosine similarity results are mostly positive (O2, O3). Similar to how in NLP we would observe documents having similar sentences (ex: e-mails always contain a selection of entry and closing statements such as ”Good morning” and ”Warm regards”), the rules that proteins follow would result in similarities between sequences.

To support findings from embedding vector and cosine similarity analysis, background knowledge about the properties of different models is used to explain the performance differences (O1). Table 5.3 highlights some key properties about the models available in the *E*-score method.

Results from the papers proposing each model are used to support findings in Chapter 5 (O4). Details regarding ProtTrans models, ESM-1b, and ESM2 are found in their respective papers (Elnaggar et al., 2021; Elnaggar et al., 2022; Lin et al., 2022; Rao et al., 2020; Rives et al., 2019).

Empirical procedures involve obtaining and collecting data for natural and random protein sequences, using them as input for each model, and collecting information about embedding vector distributions and cosine similarity between embedding vectors for every generated embedding. Findings are validated through t-tests to determine statistical significance of results for embeddings and cosine similarity.

Source code for these empirical procedures used to generate results is located on GitHub. Empirical procedures use the following: embedding generation for selected sequences; normalization of embedding values for comparison; averaging embedding values for different models for both random and non-random sequences; and averaging cosine similarity between embeddings for both random and non-random sequences.

# Results

## 5.1 Data

Data was obtained through the Conserved Domain Database (CDD) for different Multiple Sequence Alignmentss (MSAs) from the list of 49 selected in the $E$-score paper (Ashrafzadeh et al., 2023; Marchler-Bauer et al., 2015). Selected MSAs are found in Table 5.1 and in the code repository.

Procedure for obtaining CDD MSA data:

1. Select a source from Table 5.1.
2. Search for the source on the CDD website.
3. Click 'Representatives' under 'Links', send to FASTA format.
4. For reference alignments: click 'Download Alignment' instead of going to 'Representatives'

Alignment pairs $i, j$ were enumerated by iterating through each FASTA file: $\forall i\, \forall j,\ i \neq j$. These pairs were used to determine embedding value distributions (O1) and cosine similarity distributions (O3) for natural proteins. Reference alignments serve as necessary data for future fine-tuning efforts based on drawn conclusions.

Random sequence data was generated by randomly selecting residues of equal probability with replacement for sequences of random lengths between 100 and 400. This data was used to compare random embeddings and cosine similarity to naturally-observed results (O1).

## 5.2 Protein composition

Sequence similarity is essential in sequence analysis within bioinformatics (Ofer et al., 2021). 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 5.1 by taking the n-ary Cartesian product.

$$Theoretical\,Limit = \prod_{k=1}^{\infty} |A| = \prod_{k=1}^{\infty} 20 = 20 \times 20 \times \ldots \tag{5.1}$$

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 (Nevers et al., 2023).

Table 5.1: 10 MSAs with the most proteins from CDD used in the $E$-score comparison procedure (Ashrafzadeh et al., 2023; Marchler-Bauer et al., 2015).

| MSAs | | | |
|---|---|---|---|
| Conserved Domain | Source | Proteins | Length |
| *CS_CSD* | cd00024 | 522 | 98 |
| *7tm_classA_rhodopsinlike* | cd00637 | 405 | 808 |
| *FYVE_like_SF* | cd00065 | 392 | 266 |
| *Mblike* | cd01040 | 384 | 239 |
| *SH*2 | cd00173 | 352 | 214 |
| *C*1 | cd00029 | 281 | 99 |
| *KAZAL_FS* | cd00104 | 273 | 74 |
| *Globin_sensor* | cd01068 | 193 | 223 |
| *Bbox*2 | cd19756 | 127 | 65 |
| *NBD_sugarkinase_HSP*70*_actin* | cd00012 | 125 | 1154 |

Databases such as UniProt (Consortium, 2022) and PeptideAtlas (Desiere et al., 2006) are repositories filled with peptide sequences. UniProt contains over 250 million unique peptide sequences and counting (Consortium, 2022).

Peptide sequences are not completely random because of the constraints imposed on them. Similar to letters or words in a given language within natural language, the frequency of each amino acid observed in nature is not equally distributed (Beals et al., 1999).

Proteins form secondary structures as part of larger tertiary and quaternary structures. The most common of these secondary structures are $\alpha$ helices and $\beta$ pleated sheets (Ma et al., 2018). Because of this, algorithms such as an Evolutionary Neural Network Algorithm (ENNA) are able to distinguish natural proteins from randomly generated proteins with an accuracy of over 94% (De Lucrezia et al., 2012).

The distribution of the observed amino acids in all of the protein sequences from the 10 MSAs in Table 5.1 is shown in Table 5.2. Counts were acquired by reading FASTA file sequences for each MSA and generating a LaTeX table containing names, frequencies, and percentages for the 20 most common amino acids.

Table 5.2: Distribution of amino acids found in the 10 selected MSAs. A few occurrences of 'B' (nondeterministically either N or D) and some occurrences of 'X' (undetermined or atypical amino acid) were left out for simplicity.

| Amino Acid | Symbol | Frequency | Percent | Diff From Equal | P-value |
|---|---|---|---|---|---|
| Leucine | L | 152859 | 9.099 | 4.099 | 0.0e+00 |
| Serine | S | 141844 | 8.443 | 3.443 | 0.0e+00 |
| Alanine | A | 127926 | 7.614 | 2.614 | 0.0e+00 |
| Glutamic Acid | E | 108476 | 6.457 | 1.457 | 0.0e+00 |
| Valine | V | 105408 | 6.274 | 1.274 | 0.0e+00 |
| Arginine | R | 99687 | 5.934 | 0.934 | 3.2e-293 |
| Glycine | G | 96906 | 5.768 | 0.768 | 3.6e-202 |
| Threonine | T | 96702 | 5.756 | 0.756 | 4.1e-196 |
| Lysine | K | 94251 | 5.610 | 0.610 | 3.6e-130 |
| Aspartic Acid | D | 88980 | 5.296 | 0.296 | 5.2e-33 |
| Isoleucine | I | 87579 | 5.213 | 0.213 | 5.9e-18 |
| Proline | P | 86463 | 5.146 | 0.146 | 2.5e-09 |
| Glutamine | Q | 74206 | 4.417 | 0.583 | 6.3e-134 |
| Asparagine | N | 73490 | 4.374 | 0.626 | 1.3e-154 |
| Phenylalanine | F | 64495 | 3.839 | 1.161 | 0.0e+00 |
| Tyrosine | Y | 46324 | 2.757 | 2.243 | 0.0e+00 |
| Histidine | H | 43163 | 2.569 | 2.431 | 0.0e+00 |
| Cysteine | C | 36749 | 2.187 | 2.813 | 0.0e+00 |
| Methionine | M | 35289 | 2.100 | 2.900 | 0.0e+00 |
| Tryptophan | W | 19243 | 1.145 | 3.855 | 0.0e+00 |

## 5.3 *E*-score model differences

The transformer models used in the *E*-score method (see Table **??**) vary in performance (O1). ProtT5 outperformed the 5 other models available when computing end-gap-free alignments for six different conserved domain MSAs. ProtT5 and ESM2, the second best model, were compared and it was evident that ProtT5 outperformed ESM2 with statistically significant results (Ashrafzadeh et al., 2023).

*E*-score's protein transformers models have significantly different pre-training configurations (Elnaggar et al., 2021; Rives et al., 2019), some of which are highlighted in Table 5.3 (O1).

Protein transformer model pre-training configurations significantly impact model performance. For example, ProtT5 has 3 billion parameters compared to ProtAlbert having 224 million. Model performance and number of parameters are highly correlated, which is supported by the Chinchilla paper's findings for training compute-optimal LLMs (Hoffmann et al., 2022). Through the results from the comparison between models in the *E*-score paper (Ashrafzadeh et al., 2023), it was evident that the encoder-decoder model ProtT5 outperformed both the encoder-only models (ESM1b, ESM2, ProtBert, ProtAlbert) and the decoder-only model (XLNet).

Table 5.3: Pre-training configuration for protein language models (Elnaggar et al., 2021; Rives et al., 2019). UR = UniRef.

| Hyperparam | ProtT5 | ProtBert | ProtXLNet | ProtAlbert | ESM1b | ESM2 |
|---|---|---|---|---|---|---|
| Dataset | UR50 | UR100 | UR100 | UR100 | UR50 | UR50 |
| # of Layers | 24 | 30 | 30 | 12 | 33 | 33 |
| Embedding Dim | 1024 | 1024 | 1024 | 4096 | 1280 | 1280 |
| # of Params | 3B | 420M | 409M | 224M | 650M | 650M |
| Learning Rate | 0.01 | 0.002 | 0.00001 | 0.002 | 0.0004 | 0.0004 |

## 5.4 Embeddings

Understanding embedding distributions is crucial in understanding cosine similarity results and how they can be improved (O2). Embedding distributions were compared for all ProtTrans models in the *E*-score method for both randomly selected natural protein sequences and randomly generated sequences. Embedding value distributions are visualized in Figure 5.1. The procedure for obtaining average embedding values is described below:

- Obtain n sequences to provide as input to a model
- Produce and store the embedding values for all n sequences
- Normalize the embedding values, then obtain the average and standard deviation of all n embeddings

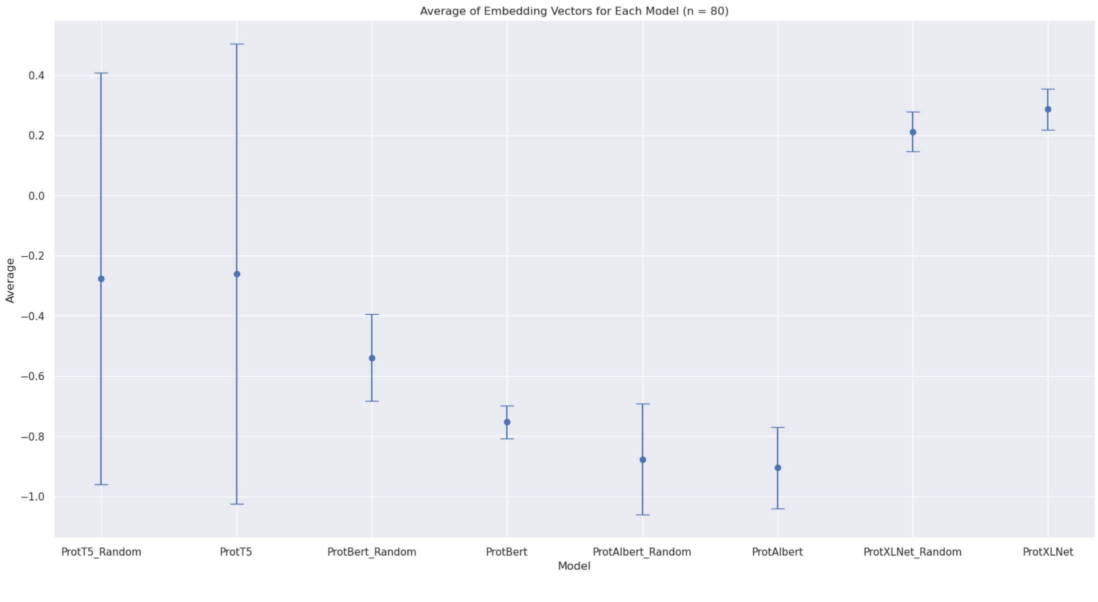

Figure 5.1: Average embedding values for 80 random and non-random (randomly chosen from CDD) embeddings for all ProtTrans models. Values scaled to -1...1.

## 5.5 Cosine Similarity

Cosine similarity distributions were compared for all ProtTrans models in the *E*-score method for randomly selected natural protein sequences and randomly generated sequences. Cosine similarity distributions are visualized in Figure 5.2. The procedure for determining cosine similarity distributions is described below:

- Get embeddings for n sequences from a selected model.
- Calculate the cosine similarity between every pair $i, j$ of embeddings, for a total of $n^2$ cosine similarity calculations.

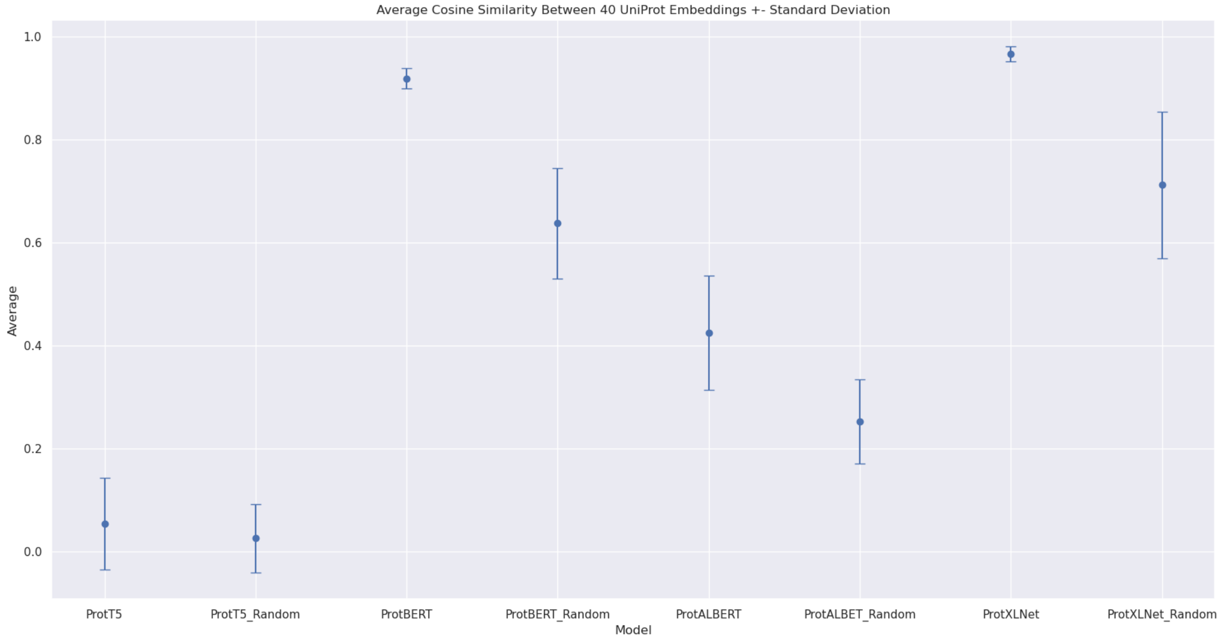

Figure 5.2: Average cosine similarity between sample and random embeddings for all ProtTrans models. P-Values are all 0.000 between any column and overall average of 0.59 (and between any chosen comparison).

# Discussion

## 6.1 Implications

The results derive implications that can be used to enhance the *E*-score method with LoRA fine-tuning. This is primarily applicable to ProtT5, as it is the best-performing method with pre-made fine-tuning notebooks that can expanded upon, but is also applicable to other models. We can fine-tune the other models to catch up to ProtT5's performance by generating embeddings with more variance in their average values. Additionally, we can create a custom penalty function to punish these models for producing mostly similar cosine similarity results, bringing the average cosine similarity result closer to 0 and closer to ProtT5's average.

The connection between observations in nature and embedding results from models is highly evident from this research. Most models fail to capture variance because of these laws governing nature, which ProtT5 managed to overcome (likely only because of its size) as results in Section 5.4 outlined. AlphaFold (Jumper et al., 2021) is a protein structure prediction method developed by Google DeepMind that uses protein transformers as the E-score method does. Because we are able to predict protein structure with transformers, it is evident that primary structures, secondary structures, structural motifs, and other properties of proteins are heavily correlated. Results from Section 5.1 and 5.2 further support this claim and outline the importance of adapting models to account for these rules. More efficient training strategies can be researched to improve the performance of models despite identical size and training time (more compute-optimal).

## 6.2 Limitations and Generalizations

Limited compute power (GPU: RTX 4070 Super) impacted the scale of embedding distribution and cosine similarity distribution procedures. With more compute power, these procedures could be conducted for all 49 MSAs used in the *E*-score method and an equivalent number of random sequences. This would greatly improve the validity of the results.

Results are generalizable to other systems utilizing ProtTrans, ESM-1b, and ESM2 models (Elnaggar et al., 2021; Rives et al., 2019). Novel conclusions can be used to support fine-tuning models for their respective use-cases. NLP use cases may repeat experimental procedures in future research to determine word frequency (ex: the word ”what” is much more common in English than ”myriad”), embedding distribution (O1), and their correlation with the results of a respective method (such as *E*-score's O3) to find parallel conclusions.

# Conclusions

This study aimed to address the limited insight into factors that contribute to better model performance in the *E*-score alignment method (Ashrafzadeh et al., 2023). Objectives included understanding embedding distributions for models and for both random and non-random sequences (O1, O2); understanding cosine similarity between these embeddings(O3); and determining how we can improve models in task performance (O4).

Key results and conclusions:

1. Proteins are not random in nature. Amino acid frequencies are not equal and vary to form particular secondary, tertiary, and quaternary structures. The reference MSAs contain significantly different frequencies for all 20 common amino acids (Section 5.2).

2. *E*-score model performance is correlated heavily with the size of a model. This is supported by ProtT5 (3 billion parameters) outperforming every other model, with ESM2 performing second best (650 million parameters) (Section 5.3).

3. Embedding value distributions with a higher variance perform better in the *E*-score method. This is supported by ProtT5 significantly outperforming all other models with a much higher variance. For worst performing models, variance is constrained by the non-random nature of proteins with random sequences having a significantly higher variance (Section 5.4).

4. Cosine similarity distributions are heavily correlated with model performance. ProtT5, the best method, has an average cosine similarity close to 0. All other models over-represent positive cosine similarity results, implying that they fail to capture variation as well as ProtT5 (Section 5.5).

# Future Work and Lessons Learned

Using the ProtTrans per-protein fine-tuning notebook as a basis to fine-tune ProtT5 for the *E*-score method may lead to significant performance benefits, especially if modified for other models. This requires significant modifications to the fine-tuning process:

- Fine-tune the model with the ProtT5 per-protein notebook as a basis, creating a LoRA adapter for the *E*-score method.
- Modify the fine-tuning notebook to work on pairs of inputs as opposed to a singular input, with penalties being assigned based on how far the *E*-score alignment score for the pair of embeddings is from the true reference alignment.

Significant lessons learned from this research:

1. Higher variance in produced embeddings is highly correlated to improved performance, meaning highly flexible models may be the key to improved *E*-score performance.
2. Average cosine similarity results closer to 0 are highly correlated with better *E*-score performance. Models that make use of the full -1...1 cosine similarity range with better-produced embeddings perform better than those with mostly positive results. Fine-tuning models to reach a mean of 0 is likely to lead to better performance.
3. The rules governing protein sequences observed in the world lead to higher cosine similarity results in all cases. Fine-tuning models to better capture variation while accounting for these properties (i.e. amino acid frequency) may lead to stronger results.

## 8.1 Acknowledgements

Dr. Lucian Ilie provided feedback on the thesis proposal and guidance throughout the process to conduct research following the original *E*-score paper (Ashrafzadeh et al., 2023).

# References

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., & Lipman, D. J. (1990). Basic local alignment search tool. *Journal of molecular biology*. https://doi.org/10.1016/S0022-2836(05)80360-2

Ashrafzadeh, S., Golding, G. B., Ilie, S., & Ilie, L. (2023). Scoring alignments by embedding vector similarity. https://doi.org/10.1101/2023.08.30.555602

Beals, M., Gross, L., & Harrell, S. (1999). Amino acid frequency. https://www.nimbios.org/~gross/bioed/webmodules/aminoacid.htm

Consortium, T. U. (2022). UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, *51*(D1), D523–D531. https://doi.org/10.1093/nar/gkac1052

De Lucrezia, D., Slanzi, D., Poli, I., Polticelli, F., & Minervini, G. (2012). Do natural proteins differ from random sequences polypeptides? natural vs. random proteins classification using an evolutionary neural network. *PLOS ONE*, *7*(5), 1–10. https://doi.org/10.1371/journal.pone.0036634

Desiere, F., Deutsch, E. W., King, N. L., Nesvizhskii, A. I., Mallick, P., Eng, J., Chen, S., Eddes, J., Loevenich, S. N., & Aebersold, R. (2006). The PeptideAtlas project. *Nucleic Acids Research*, *34*(suppl$_1$), D655–D658. https://doi.org/10.1093/nar/gkj040

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/arXiv.1810.04805

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Yu, W., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2021). Prottrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. https://doi.org/10.1109/TPAMI.2021.3095381

Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M., Bhowmik, D., & Rost, B. (2022). Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, *44*(10), 7112–7127. https://doi.org/10.1109/TPAMI.2021.3095381

Gligorijević, V., Renfrew, P. D., Kosciolek, T., Leman, J. K., Berenberg, D., Vatanen, T., Chandler, C., Taylor, B. C., Fisk, I. M., Vlamakis, H., Xavier, R. J., Knight, R., Cho, K., & Bonneau, R. (2021). Structure-based protein function prediction using graph convolutional networks. *Nature Communications*, *12*(1), 3168. https://doi.org/10.1038/s41467-021-23303-9

Henikoff, S., & Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*. https://doi.org/10.1073/pnas.89.22.10915

Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., de Las Casas, D., Hendricks, L. A., Welbl, J., Clark, A., Hennigan, T., Noland, E., Millican, K., van den Driessche, G., Damoc, B., Guy, A., Osindero, S., Simonyan, K., Elsen, E., … Sifre, L. (2022). Training compute-optimal large language models.

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard,

A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., . . . Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Khurana, D., Koli, A., Khatter, K., & Singh, S. (2023). Natural language processing: State of the art, current trends and challenges. *Multimedia Tools and Applications*, *82*(3), 3713–3744. https://doi.org/10.1007/s11042-022-13428-4

Kulmanov, M., & Hoehndorf, R. (2019). DeepGOPlus: improved protein function prediction from sequence. *Bioinformatics*, *36*(2), 422–429. https://doi.org/10.1093/bioinformatics/btz595

Lai, B., & Xu, J. (2021). Accurate protein function prediction via graph attention networks with predicted structure information. *bioRxiv*. https://doi.org/10.1101/2021.06.16.448727

Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., & Soricut, R. (2020). Albert: A lite bert for self-supervised learning of language representations.

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., et al. (2022). Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. https://doi.org/10.48550/arXiv.1907.11692

Ma, Y., Liu, Y., & Cheng, J. (2018). Protein secondary structure prediction based on data partition and semi-random subspace method. *Scientific Reports*, *8*(1), 9856. https://doi.org/10.1038/s41598-018-28084-8

Marchler-Bauer, A., Derbyshire, M. K., Gonzales, N. R., Lu, S., Chitsaz, F., Geer, L. Y., Geer, R. ., He, J., Gwadz, M., Hurwitz, D. I., Lanczycki, C. J., Lu, F., Marchler, G. ., Song, H. S., Thanki, N., Wang, Z., Yamashita, R. A., Zhang, D., Zheng, C., & Bryant, S. H. (2015). Cdd: Ncbi's conserved domain database. https://doi.org/10.1093/nar/gku1221

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space.

Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning* (2nd ed.). MIT Press. (Original work published 2012)

Nevers, Y., Glover, N. M., Dessimoz, C., & Lecompte, O. (2023). Protein length distribution is remarkably uniform across the tree of life. *Genome Biology*, *24*(1), 135. https://doi.org/10.1186/s13059-023-02973-2

Ofer, D., Brandes, N., & Linial, M. (2021). The language of proteins: Nlp, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, *19*, 1750–1758. https://doi.org/https://doi.org/10.1016/j.csbj.2021.03.022

Pennington, J., Socher, R., & Manning, C. (2014). GloVe: Global vectors for word representation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532–1543). Association for Computational Linguistics. https://doi.org/10.3115/v1/D14-1162

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. https://doi.org/10.48550/arXiv.1910.10683

Rao, R., Meier, J., Sercu, T., Ovchinnikov, S., & Rives, A. (2020). Transformer protein language models are unsupervised structure learners. *bioRxiv*. https://doi.org/10.1101/2020.12.15.422761

Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C. L., Ma, J., & Fergus, R. (2019). Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *PNAS*. https://doi.org/10.1101/622803

Senior, A. W., Evans, R., Jumper, J., Kirkpatrick, J., Sifre, L., Green, T., Qin, C., Žídek, A., Nelson, A. W. R., Bridgland, A., Penedones, H., Petersen, S., Simonyan, K., Crossan, S., Kohli, P., Jones, D. T., Silver, D., Kavukcuoglu, K., & Hassabis, D. (2020). Improved protein structure prediction using potentials from deep learning. *Nature*, *577*(7792), 706–710. https://doi.org/10.1038/s41586-019-1923-7

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction* (Second). The MIT Press. https://inst.eecs.berkeley.edu/~cs188/sp20/assets/files/SuttonBartoIPRLBook2ndEd.pdf

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). Attention is all you need. https://doi.org/10.48550/arXiv.1706.03762

Yang, J., Anishchenko, I., Park, H., Peng, Z., Ovchinnikov, S., & Baker, D. (2019). Improved protein structure prediction using predicted inter-residue orientations. *bioRxiv*. https://doi.org/10.1101/846279

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., & Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*. https://doi.org/10.48550/arXiv.1906.08237