Explaining embedding results for scoring alignments

Riley Gavigan

Department of Computer Science, University of Western Ontario

Supervisor: L. Ilie
Instructor: N. Madhavji
CS 4490Z

Introduction

- Proteins are one of the essential molecules of life. By computing alignments between protein sequences, we can find similarities among protein sequences. This is essential in identifying protein structure and function.
- *E*-score (Ashrafzadeh et al., 2023) is a method to compute alignments using contextual embeddings produced by protein language models. It outperforms state-of-the-art protein scoring methods.
- This study investigates embedding vector distributions produced by these protein language models, as well as their cosine similarity.
- This investigation leads to implications that can improve these protein language models for the *E*-score method, resulting in improved performance.

Table of Contents

1. Background
2. Research Methodology
3. Results
4. Novelty and Analysis
5. Limitations of Results
6. Impact on Theory and Practice
7. Validation
8. Conclusions
9. Future Work
10. Lessons Learned

Background: Protein Language Models

- Protein Language Models are based off of Natural Language Processing models. These are deep-learning models trained on large amounts of protein data.

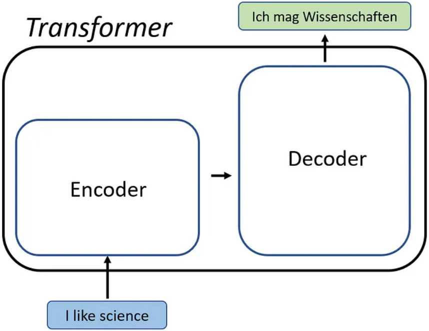

Figure: Simple example of the transformer architecture with an NLP example. Author: Diego Unzueta

Background: *E*-score

- The *E*-score method makes use of the following Protein Language Models: ProtT5, ProtBERT, ProtALBERT, ProtXLNet, ESM1b, and ESM2 (Elnaggar et al., 2021; Rives et al., 2019).
- *E*-score works by computing the cosine similarity between two embeddings generated by one of these models.
- This cosine similarity identifies the similarity between two provided sequences.

$$\mathrm{CosSim}(A,B) = \cos(\theta) = \frac{A \cdot B}{\|A\|\|B\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2}\sqrt{\sum\limits_{i=1}^{n} B_i^2}} .$$

Figure: Cosine similarity calculation for *E*-score (Ashrafzadeh et al., 2023).

Research Methodology

- Tools & Concepts Used
  - Python: PyTorch (CUDA), SciPy, Seaborn, NumPy, Transformers
  - Statistical Analysis: Distributions, T-Tests, Error Bar Visualization
  - Data: Conserved Domain Database (CDD) reference alignments and sequences
- 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.

Results

- Proteins are not random in nature (Ofer et al., 2021). Amino acid frequencies are not equal, some are more common than others.
- Model performance for the E-score method is highly correlated with model size. Larger model = better performance.
- Embedding value distributions with a higher variance perform better in the E-score method (Ashrafzadeh et al., 2023).
- Cosine similarity distributions are highly correlated with model performance. Models with cosine similarity averages closer to 0 perform better.

Results: Amino Acid Frequencies

- Amino acid frequencies vary to form particular secondary, tertiary, and quaternary structures (Ofer et al., 2021).

Table: Distribution of amino acids found in the 10 selected MSAs.

Amino Acid	Symbol	Frequency	Percent	Diff From Equal
Leucine	L	152859	9.099	4.099
Serine	S	141844	8.443	3.443
Alanine	A	127926	7.614	2.614
Glutamic Acid	E	108476	6.457	1.457
Valine	V	105408	6.274	1.274
Arginine	R	99687	5.934	0.934
...	...	...	...	...
Tryptophan	W	19243	1.145	3.855

Results: Model Size

- Larger models perform better than smaller models. ProtT5, the best performing model, has 3 billion parameters (Elnaggar et al., 2021). Similarly, ESM2, the second best performing model, has 650 million parameters (Rives et al., 2019).

Table: Pre-training configuration for protein language models (Elnaggar et al., 2021).

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

Results: Embedding Value Distributions

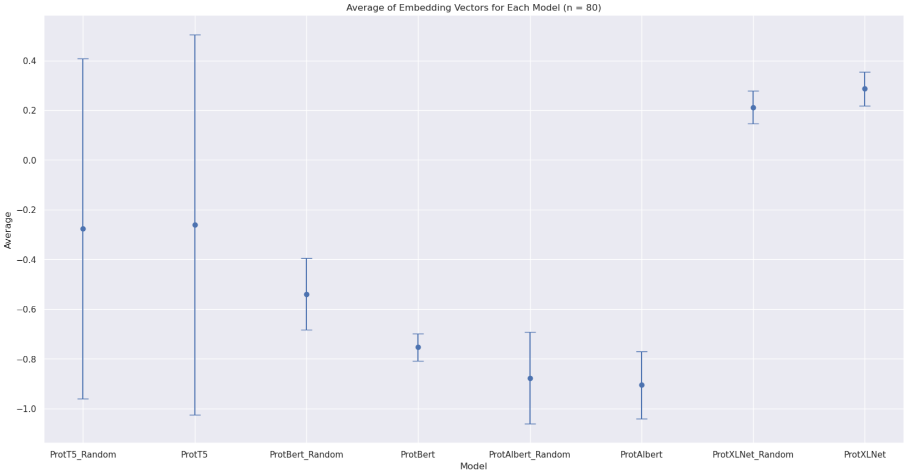

Figure: Average embedding values for 80 random and non-random (randomly chosen from CDD) embeddings for all ProtTrans models. Values scaled to -1...1.

Results: Cosine Similarity

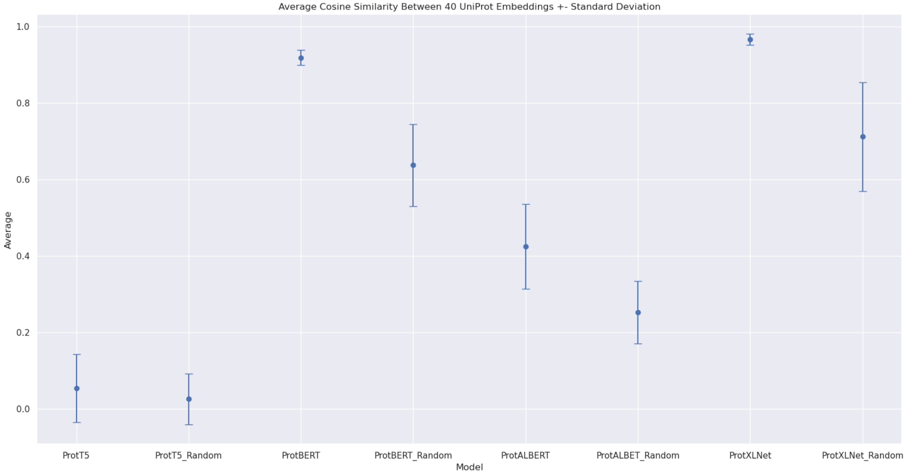

Figure: Average cosine similarity for all ProtTrans models. P-Values are all 0.000 between any column.

Novelty & Analysis: Embedding Value Distributions

- Developing highly flexible models will result in better E-score performance. Specifically, models that capture more variation will perform better.
- Models should be trained with this in mind, remembering that results are impacted by amino acid distribution and model property results as well.
- **Data**: ProtT5's distributions are significantly greater than other ProtTrans models and performs significantly better, supporting this claim.

Novelty & Analysis: Cosine Similarity

- A high-performing model has average cosine similarity distributions close to 0. This indicates that embedding value distributions are properly capturing similarities and differences.
- Penalizing models during training according to labeled alignment scores vs. calculated cosine similarity will result in stronger models.
- Our results provide insight into how we can better adapt and improve protein language models for the *E*-score method (Ashrafzadeh et al., 2023), both fine-tuning or custom model creation with unique labels for UniRef datasets (Consortium, 2022).
- **Data**: ProtT5's cosine similarity distribution is the closest to an average of 0 with a low standard deviation, supporting this claim.

Limitations of Results

- **Limitation**: Limited compute power impacted scale for analysis of embedding value and cosine similarity distributions.
- **Solution**: More compute power to allow computation for all *E*-score method MSAs, greatly improving validity and generalizability. Ex: NVIDIA H100/A100 TPU.

Impact on Theory and Practice

- We have insight into how we can improve protein language models for the *E*-score method. These insights can be generalized to other protein language model tasks, such as structure prediction (Z. Yang et al., 2019).
    - Ex: Instead of cosine similarity, perform the same study on the calculation specific to that task.
- When training or fine-tuning protein language models in the future, modifying rewards/penalties to account for variance in embeddings and cosine similarity/scoring tasks may lead to improved model performance.
    - Ex: Either increase model size or penalize small variations more when training on UniRef datasets (Consortium, 2022).
- The non-random nature of protein composition and patterns should be accounted for when working with protein language models.

Conclusions: Objectives 1 & 2

- **O1**: The nature of non-random protein composition, combined with models that better capture variance performing better, explains the observed distributions of embeddings (Results: Embedding Value Distributions).
    - Higher variance is correlated with greater performance, and random sequences have a higher variance than naturally observed proteins.
- **O2**: Embeddings with greater variance result in better cosine similarity and alignment results in the *E*-score method.
    - Comparing both distribution charts (Results: Cosine Similarity and Results: Embedding Value Distributions) displays the high correlation between greater variance and cosine similarity averages approaching 0.

Conclusions: Objectives 3 & 4

- **O3**: Cosine similarity results are generally higher because of the non-random nature of proteins (Results: Amino Acid Frequencies).
    - As shown in our results (Results: Cosine Similarity), random average cosine similarities are always closer to 0 than non-random cosine similarities.
- **O4**: With a novel implementation to fine-tune our models 2 sequences at a time, we can penalize embedding vectors based on how close alignments are to human-labeled alignment scores.

Future Work

- 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.
  - **Procedure**: Fine-tune the model with the ProtT5 per-protein notebook as a basis, creating a LoRA adapter for the *E*-score method.
  - **Note**: 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.
- Repeating this study on Natural Language Processing models, since protein models are based on NLP equiavlents and we can further generalize results.

Lessons Learned

- Higher variance in produced embeddings is highly correlated to improved performance, meaning highly flexible models may be the key to improved E-score performance.
- 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.
- 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.

References

- 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
- 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
- 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

References

- 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
- 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
- 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