Predicting Protein Stability Scores using Large Language Models

CS 7650 Final Project

Zoey (Li-Yen) Yang, Reshma Anugundanahalli Ramachandra, Aryan Vats, Nemath Ahmed Shaik

Introduction

Purpose:

Protein engineering is an active research area that aims in producing synthetic proteins with desired functions such as catalyzing chemical reactions or treating diseases. One challenge in engineering proteins is that a lot of the proteins are not stable enough[1]. An unstable protein will be easily denatured and will lose the functions we are originally interested in. Therefore, if we can predict the protein stability given a protein sequence before synthesizing it, we will be able to design a protein more efficiently by eliminating the time and cost for synthesizing proteins that are not stable. In this project, our goal is to use large language protein models to make predictions of protein stability using the amino acid sequences of the proteins.

Dataset:

In 2017, Rocklin et al published a research article[1] where the authors synthesized and measured the stability of more than 50,000 proteins. The dataset from this research was curated by Rao et al in 2019 as a benchmarking dataset[2], Tasks Assessing Protein Embeddings (TAPE), for protein property predictions. The TAPE dataset contains 68,977 protein sequences with their corresponding stability values. A protein sequence of length L is composed of amino acids (X_1, X_2, …., X_L), where X can be any of the 25 character alphabets (with 20 standard amino acids, 2 non-standard amino acids, 2 for ambiguous amino acids, and 1 if the amino acid is unknown).

Contribution:

Rao et al[2] have experimented a few deep learning methods, including Transformer, LSTM, and ResNet, to predict the stability scores of the protein sequences in the TAPE dataset. To our knowledge, there are no published results of stability predictions on this dataset using recently published protein language models(UniRep[3], ESM2[4], ProtBert[5], ProtTrans[6]) that are trained on large protein sequence databases. Therefore, in this project, we aim to explore the performance of protein stability prediction of the TAPE dataset of those pre-trained models. We hope to understand whether these models can help improve the downstream task of predicting protein stability scores.

Approach

The TAPE dataset was split into three subsets by the authors, with 53,614 samples in the training dataset, 2,512 samples in the validation dataset, and 12,851 samples in the test dataset. We trained our models with the training dataset, fine tuned the hyperparameters with the validation dataset, and evaluated their performance on the held-out test dataset. Finally, we evaluated the performance of the three subsets using MSE, RMSE, R2, and Spearman ρ of the predicted stability scores and the true stability scores. We also compared the results of Spearman ρ with the results from Rao et al.[2] on the test dataset. They implemented an one-hot

encoding model, a LSTM model, a transformer model, and a ResNet model that were pre-trained on the TAPE dataset using masked-token prediction or next-token prediction.

The details of fine-tuning the four pre-trained protein language models are as follow:

ESM2[4]

ESM2, Evolutionary Scale Modeling, is a deep contextual language model that was trained on 250 million protein sequences using unsupervised learning. The pretrained ESM2 is composed of 34-layer Transformer models and has ~113 M parameters. The embedding from ESM2 contains information of the biological representation of the given proteins. We installed the pretrained model from the hugging face repository, and we built two additional linear layers that map the last hidden state of the model to a single stability value. To fine-tune the model, we used Adam optimize with a learning rate of 0.0001 and a batch size of 64. We trained our model for 10 epochs with Mean Square Error as the loss function.

UniRep[3]

We also explored UniRep, a multilayer long short-term memory (mLSTM) "babbler" deep representation learner for protein engineering informatics. UniRep is capable of training, inferencing representations, generative modeling, and data management. We experimented three architecture of 64 units along with the trained architectures and begin evotuning the parameters. To fine-tune UniRep on the TAPE dataset, we used the Adam optimizer with a learning rate of 0.001 and a batch size of 512. We used early stopping based on the validation loss and trained UniRep for up to 10 epochs. They also used dropout regularization with a rate of 0.1 to prevent overfitting. UniRep is highly scalable and can be fine-tuned on large datasets with minimal modifications to the architecture or training procedure. The future work around UniRep with TAPE can be with more units (256, 1900).

ProtBert[5]

ProteinBERT is a protein language model pre-trained on the UniRef90 dataset on approximately 106M proteins. This model was then fine tuned for our use case with the TAPE dataset to predict the stability of a protein. The model takes protein sequences as inputs but it can also take protein GO annotations as additional inputs. The original ProteinBERT model was trained to accomplish State of the Art results on a wide variety of benchmarks such as Signal Peptide, Fluorescence, Secondary Structure and so on. The architecture of ProteinBERT is inspired by BERT but contains several protein specific innovations such as a global attention layer. This results in the model being able to process protein sequences of almost any length, especially long amino acids. We fine-tuned the model by encoding the TAPE dataset into ProteinBERT compatible data through the ProteinBERT library, and running the model on the TAPE train dataset by freezing all but the last two layers and adding a different output head that yields a single score for the stability of the protein. We use the Adam optimizer with a learning rate of 0.0001 and a reduction of the Learning Rate on plateaus with early stopping as well. This ensured that we did not overfit on the model and the global attention layers ensured stable performance across proteins of varied length.

ProtTrans[6]

Another pretrained model that we explored was ProtTrans. Similar to UniRep and ProtBERT, ProtTrans is a deep learning architecture designed to model protein sequences which was trained to predict the next amino acid for a given protein sequence. However, unlike UniRep and ProtBERT, ProtTrans was pre-trained on a larger dataset (consisting exclusively of protein sequences), but is also computationally more demanding while training. To fine-tune the ProtTrans model for the stability score prediction task using the TAPE dataset, Adam optimizer was used with an Mean Squared Error (MSE) loss function. The training was done with a learning rate of 0.0001, batch size of 64 and for one epoch. ProtTrans seemed to be highly computationally demanding, taking almost 5 hours to complete one epoch on a colab GPU. Thus, with better resources, the model's performance could increase with an increased number of epochs.

Performance

This section provides a comparative analysis of the performance of all the four pretrained models on the stability score prediction task for the train (Table 1), validation (Table 2) and test (Table 3) data sets. The metrics chosen are R-squared, Root Mean Square Error (RMSE), Mean Square Error (MSE) and Spearman ρ to measure how close the predicted stability scores are to the ground truth values.

Table 1: Model performance on the train dataset

	R^2	RMSE	MSE	Spearman ρ
UniRep	0.2	0.39	0.15	0.42
ESM	0.63	0.35	0.12	0.86
ProtBert	0.28	0.48	0.23	0.43
ProtTrans	0.47	0.34	0.12	0.77

Table 2: Model performance on the validation dataset

	R^2	RMSE	MSE	Spearman ρ
UniRep	0.12	0.4	0.16	0.39
ESM	0.53	0.45	0.20	0.75
ProtBert	0.38	0.51	0.26	0.60
ProtTrans	0.26	0.40	0.16	0.73

Table 3: Model performance on the test dataset

	R^2	RMSE	MSE	Spearman ρ

One-hot from Rao et al[2]	N/A	N/A	N/A	0.19
Transformer from Rao et al[2]	N/A	N/A	N/A	0.73
LSTM from Rao et al[2]	N/A	N/A	N/A	0.69
ResNet from Rao et al[2]	N/A	N/A	N/A	0.73
UniRep	-0.37	0.4	0.16	0.32
ESM	-0.50	0.50	0.25	0.70
ProtBert	0.04	0.43	0.18	0.60
ProtTrans	-0.35	0.44	0.19	0.71

From the results in the test dataset, we observed that among the four pre-trained models, ESM and ProtTrans performed better than the other two models. Those two models were able to achieve ~0.7 Spearman ρ in the test set. However, these models did not achieve better results compared to the Transformer and ResNet models that achieved ~0.73 Spearman ρ in Rao's et al publication. The reason may be coming from the fact that those models in Rao's et al were pre-trained directly on the TAPE dataset, whereas the four large protein language models we tested were trained on protein datasets that include all possible proteins in general. Future work on a more extensive hyperparameter tuning and longer training epochs of those protein language models may be helpful to provide a better comparison of the models' performance.

Summary of Contribution

Nemath: Worked on a mLSTM "babbler" (UniRep) and used it for TAPE dataset implementing custom models and inference.
Reshma: Implemented ProtTrans - Fine-tuned it for the stability score prediction task on the TAPE dataset.
Aryan: Implemented ProtBert - Handled the downstream transformation of ProtBert for stability scores, trained on the TAPE data.
Zoey: Conceived the project idea. Implemented ESM and fine-tuned it for the stability prediction task on the TAPE dataset.
All members contributed to writing the report.

Code Availability:

Our code is available at: https://github.gatech.edu/lyang417/ProteinStability_LLM

References

[1] G. J. Rocklin *et al.*, “Global analysis of protein folding using massively parallel design, synthesis, and testing,” *Science*, vol. 357, no. 6347, pp. 168–175, Jul. 2017, doi: 10.1126/science.aan0693.

[2] R. Rao *et al.*, “Evaluating Protein Transfer Learning with TAPE,” *Adv. Neural Inf. Process. Syst.*, vol. 32, pp. 9689–9701, Dec. 2019.

[3] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, “Unified rational protein engineering with sequence-based deep representation learning,” *Nat. Methods*, vol. 16, no. 12, Art. no. 12, Dec. 2019, doi: 10.1038/s41592-019-0598-1.

[4] A. Rives *et al.*, “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences,” *Proc. Natl. Acad. Sci.*, vol. 118, no. 15, p. e2016239118, Apr. 2021, doi: 10.1073/pnas.2016239118.

[5] N. Brandes, D. Ofer, Y. Peleg, N. Rappoport, and M. Linial, “ProteinBERT: a universal deep-learning model of protein sequence and function,” *Bioinformatics*, vol. 38, no. 8, pp. 2102–2110, Apr. 2022, doi: 10.1093/bioinformatics/btac020.

[6] A. Elnaggar *et al.*, “ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022, doi: 10.1109/TPAMI.2021.3095381.