Comparing Latent Space Representations of Protein Sequencing Models

By

Isabelle Fox, Jonathan Jwa, Brian Lee, Joseph Wu

For

Dr. Peter Tonner of National Institute of Standards and Technology (NIST)

Computational Modeling & Data Analytics at Virginia Tech
7 December 2022

EXECUTIVE SUMMARY

Team Amino Amigos spent the semester working on Comparative AI Identification of Protein Sequences under the supervision of Dr. Peter Tonner of the National Institute of Standards and Technology (NIST). Proteins influence and control many of the processes occurring inside every living cell. Because of their ubiquity, understanding how a protein's sequence (an ordered collection of twenty different amino acids) influences downstream function plays a critical role across the life sciences. NIST is interested in exploring whether latent protein representations learned by different machine learning (ML) models – UniRep, XLNet, and ESM – are related or not. To compare model spaces, the team used exploratory data analysis to discover early trends in the model spaces. Then, we reduced the initial outputs of the models, via Principal Component Analysis, and employed Gaussian Mixture Models to cluster the dimension-reduced output for optimal cluster and co-clustering analysis. The team found for ProteinGym inputs, and particularly for inputs relating to the p53 protein, that ESM, UniRep, and XLNet produce different representations which reveal the model spaces are not the same. Investigations conducted by this team are not foolproof – depending on input into the model, the resulting model spaces change. It is difficult to look inwards towards what exactly about the proteins leads to these changes because of the nature of the analysis completed by the team, but also the intricacy of the initial data.

1. PROBLEM STATEMENT

Thorough research has designated proteins as one of the most important biomolecules in the cells of living organisms. Built as sequences of amino acids, proteins are considered the main "working molecules" in an organism because of their support for a diverse collection of crucial functions, even down to the cellular level. These functions range from those structural – the physical building blocks to bodily matter – to enzymatic – facilitators of chemical reactions in the body. It is commonly thought that these unique functionalities result from their distinct structure set by the sequence of amino acids from which they are constructed.

Proteins fold into various structures based on their amino acid sequence, which likely plays a significant role in their functions. However, the relationship between a protein's amino acid sequence and its downstream functionality is largely unknown. Although well-known protein sequences and structures have been associated with an obvious function, finding this information is often a painstaking effort in the lab. Furthermore, establishing and identifying the association remains a challenge. But naturally, discovering it is invaluable. Bioengineers who have this capability are closer to designing proteins that will fold into structures that they hope will produce specific biological outcomes in an organism.

In the most recent years, bioengineers have turned to AI-driven approaches to their protein research efforts to rival those of traditional experimental methods. Large strides have been made in training models that predict protein structure given its sequence. AlphaFold2, developed by Google's sister company DeepMind, achieved a median score of 92.4 in 2020 when predicting the structures of proteins, a jump from a median score of 80 [4]. In November of 2022, Mark Zuckerberg boasted that Meta's AI research body built a model that achieves protein folding predictions sixty times faster than current technologies [5]. Similarly, the Unified rational protein engineering with sequence-based deep representation learning (UniRep) model, developed in 2019, saw promising success in learning fundamental features of proteins, retaining structural and biophysical semantics in a statistical representation [6].

These are exciting frontiers of research. However, although such models retain impressive predictive power of aspects of proteins based on a given sequence, it is often difficult to characterize semantically, what these models are learning as well. Under the sponsorship of the National Institutes of Standards and Technology, this project is aimed to assess whether latent space representations of protein amino acid sequences learned by different machine learning models are similar or different. This project serves to further the understanding of how we think about how machine learning models are applied to protein research tasks.

2. ETHICAL CONSIDERATIONS

Analyzing and leveraging data with strong relationships to biology and biomedical engineering naturally raises serious ethical considerations to be made. The value of living things is threatened by any ability to manufacture and select desired biological outcomes, even if done with moral intent. The misinterpretation of inaccurate data will cause pharmacists, nutritionists, doctors, etc. to mistreat or misguide their clients causing such ethical issues in those fields outside our control.

As previously mentioned, this data supposedly contains information formulated by machine learning methods, a technology that is considered a new frontier in protein sequencing applications. Such data requires caution when working with it. A failure to construct an appropriate data model that could determine which specific protein can be either beneficial or harmful to people with certain conditions and failure to correctly scrutinize it, then it will lead to the mistreatment of people which could then lead to many serious health issues such as death. However, the hope is that new discoveries made using this data and about the machine learning models that output them build further understanding of such relationships.

3. LITERATURE REVIEW

3.1 Comparing High-Dimensional Neural Recordings by Aligning Their Low-Dimensional Latent Representations [1]

The aim of our project is to compare the latent spaces created by a collection of protein sequencing models. Analyzing the results of dimensional scaling (reducing very large, high-dimensional data down to only a few dimensions) is something we are well-versed in from CMDA 3654. However, for this project, we need to compare different latent spaces, generated by different models, and this proves to be a more abstract, confusing adversary. This paper discusses the alignment methods by which researchers can compare low-dimensional latent spaces for neural networks. Because of the nature of neural networks, dimension reduction results are not directly comparable to each other, so alignment methods are needed to create a common space by which model latent space can be compared. [1] Canonical Correlation Analysis (CCA) is similar to PCA, but instead of picking an axis that describes the most variance within one dataset at a time, CCA picks an axis that describes the most variance for all of the data provided. Then the dimension reduction can be performed the same as PCA with the found axes. [1] The content of this paper played a role in choosing our dimension reduction technique.

3.2 A General Framework for Manifold Alignment [2]

Many of the possible solutions discussed for a mathematical model included a form of dimensionality reduction, the process of transforming a high-dimensional space into a low-dimensional space. We worked with different-sized datasets as the result of the two different models initially provided. The first step towards comparing these datasets is making sure that these dimensions are the same. One possible solution is implementing manifold learning, an approach to non-linear dimensionality reduction. This paper provides a general introduction to manifold alignment and explains how this method approaches situations dealing with high-dimensional data. The author acknowledges that comparing and connecting two high-dimensional datasets is difficult given the differing nature of their features, but proposes the solution of manifold alignment. Manifold alignment "builds connections between two or more disparate data sets by aligning their underlying manifolds and provides knowledge transfer across the data sets." [2] In this case, manifold refers to the topological space represented by each dataset in the n-dimensional Euclidean plane. Manifold alignment takes each dataset and maps each feature/column into a common space where the elements of each can be directly compared. The author reveals that the manifold alignment approach can be distinguished into two types. A first type is an unsupervised approach that includes a diffusion map-based alignment and a Procrustes alignment which maps the datasets to a low-dimensional space while some rotational and scaling components are removed in order for the alignment of the two sets to match. [2] A second type is a semi-supervised approach which first creates a joint manifold that represents the union of the given manifolds and maps it to a low-dimensional latent space. [2] While the paper gives a good overview of these two different approaches, the team focused most heavily on implementing Principle Component Analysis instead. This paper provided valuable information on the method of manifold alignment and the value of dimension reduction in our project.

3.3 Learned protein embedding for machine learning [3]

Our main unit of observation comes from proteins, which possess unique sequences of amino acids. To make this sequence data more interpretable by machines, pretrained machine learning networks were used to map the amino acid sequence to numerical data. However, this data is high-dimensional and intractable. Considering that the goal of our project is to learn the relationships between these representations, this work by Yang and colleagues details interesting research done to learn the embeddings of protein sequences using machine learning methods. This paper introduces the main usage of machine learning in protein

sequences, to predict protein properties for protein designing and engineering. It has also brought attention to how each protein sequence is encoded and could determine what can be learned through the models and how even the most powerful model could produce a poor result, allowing us to find if the models given to us have any correlativity amongst each other. Such work is highly relevant to our project as a clear demonstration that insights on relationships between protein sequence latent space representations can be discovered. When starting on this project with NIST, it was a very real possibility that there might not exist any relationship with how models represent protein sequences. Additionally, Yang makes use of unsupervised learning, which is how much of the pretrained models we are trying to evaluate were trained on. This article provided valuable information regarding the utilization of machine learning and neural networks on protein sequences.

4. PROJECT COMPONENTS & CRITERIA

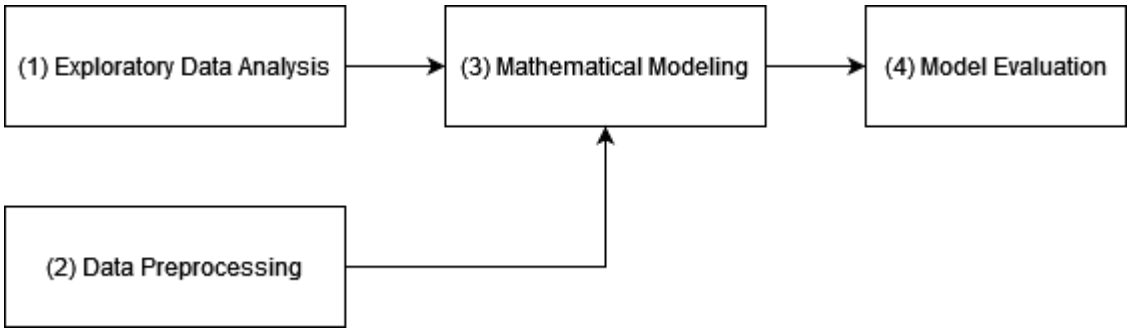

Figure 1: Diagram of Components

4.1 Preliminary Components

Exploratory Data Analysis. Any project that works with data containing underlying relationships requires some exploration of the datasets' contents. This includes extracting key statistics and observing the distributions of various aspects of the data. In the case of these datasets, machine learning outputs do not have any clear semantic meaning attached to the features in the data to the best of our knowledge, so exploration had to venture further than what was in the data matrix. For example, while the features of the data do not contain any semantic meaning, getting at the research question can be done by finding relationships between sequence length and feature vector magnitude.

Data Preprocessing. This smallest component of the project, data preprocessing is the component in which data is cleaned and organized to be ready to be used to fit the data. It is here where we would decide if or how we should concatenate data for specific analyses or if any normalization needs to take place on the datasets.

4.2 Substantive Components and Criteria

Mathematical Modeling. The data provided gives the latent space representation output of three different machine learning models for comparison. Every single dataset can have over 1900 features and thousands of rows. It directly follows that processing data of this size is not feasible in a reasonable amount of time with the computing resources readily available to the group. As such, dimension reduction techniques need to be employed in order to transform the provided data down to a size in which insightful results can be gathered. Mathematical Modeling is the component responsible for this necessary transformation.

This component of the project involves constructing models from the data that reduce the number of dimensions to a common number that is not only easier to analyze, but also able to be visualized by

our analyses. This is done with the intention of having the most sensible data and preserving the most information possible.

- *Interpretability* – Mathematical modeling techniques and algorithms should build dimension-reducing models whose outputs are in one dimension as a numerical output, or two and three dimensions numerical output, which can be viewed as graphical plots and used for downstream data analysis.
- *Ease of coding* – Mathematical model functions should be able to run and fit the data without impedance from significant bugs or significant modifications to the data to appease hyperparameters. No more than 15 minutes of runtime per run.
- *Data Format* – The data fed into our model should be in .csv format and/or converted into a Pandas DataFrame, and should be able to handle input from high-dimensional data with up to 1900 features.

Model Evaluation. After the provided data is dimension-reduced in the Mathematical Modeling component, it is possible to then compare the machine learning model spaces to determine whether or not these spaces are similar or different. The Model Evaluation component provides a means by which to compare these dimension-reduced outputs and answer the question at the heart of the problem statement.

- *Interpretability* – The evaluation should be understandable by the client and comprehensible to the untrained eye.
- *Reliability* – The model evaluation should show consistent output throughout the different protein sequences to form the reliability of the model.
- *Output* – The output of the evaluation must provide some numerical metric from our mathematical models interpreting the similarity between protein sequencing models.

5. SELECTED SOLUTIONS

5.1 Mathematical Modeling

For the Mathematical Modeling solution, the team used Principal Component Analysis (PCA) to transform our high-dimensional datasets, consisting of outputs from protein sequencing models, to a low-dimensional space allowing us to create interesting visualizations and conduct interpretable analyses. PCA is a well-studied method of dimensionality reduction that allows us to fit a model to the data and transform it to lie along two or three axes. With the use of singular value decomposition, PCA preserves the most important information while minimizing information loss. Our datasets consisting two features, which allows our protein sequence to be representable on a two-dimensional plot. The Python module Scikit-learn contains a PCA class with functions that can perform these mathematical operations to fit a model of a dataset and conduct dimensional reduction to the desired number of components with a single function.

5.2 Model Evaluation

For our approach to model evaluation, the team decided to use Gaussian Mixture Models (GMMs) as the clustering method to compare the latent spaces of the protein sequencing models. A Python implementation of Gaussian Mixture Models in Scikit-learn provided the means to cluster a given dataset. As was the method suggested by our sponsor Dr. Tonner, this was the preferred choice because GMMs also employ maximum likelihood estimation (MLE) under the hood of their computations. The GMM was used to fit training data and then cluster the remaining test data. Clustered data of protein sequences can be then used into two subcomponents of model evaluation, an analysis of optimal clusters, and co-clustering analysis.

5.2.1 Optimal Cluster Analysis. Because GMMs use MLE, the Bayesian Information Criterion could be found for each cluster model generated. Thus the first step of our clustering was to perform optimal clustering analysis. Clustering models were built for the ML models' PCA-reduced outputs for $k = 1, 2, ..., 5$. The clustering model with the lowest BIC for each dataset pair was determined and defined as the optimal model. The optimal number of clusters for that dataset pair were determined to be the number of clusters in the previously defined optimal model. The optimal number of clusters was then compared between the same inputs across the different models. The idea behind the analysis is that if these model spaces are similar, the optimal number of clusters for the outputs between models on the same inputs will also be similar.

5.2.2 Co-Clustering Behavior Analysis. The clustering results from the GMM are certainly interesting on their own and can be used to help reveal what kinds of protein representations are considered similar by the GMM for one ML model. However, they do not reveal much about the similarity of these representations as expressed by the other ML models. Instead, the model evaluation would require looking at the co-cluster assignments, and assessing whether the GMM clusters the same proteins together across different ML model outputs.

After creating Gaussian Mixture Models for each dataset we aimed to observe, we then performed a pairwise comparison of the proportion of set differences for each cluster $i = 0...k$ belonging to one model output and clusters $j = 0...k$ belonging to another model output simultaneously. This exhaustive comparison is necessary because clustering assignment and ordering are arbitrary. In other words, for $i = 0...k$, cluster i for one model might not necessarily be the same intended cluster for another model if such a relationship between the two ML model outputs were to exist.

The intuition behind this analysis, then, is to determine the amount of protein co-cluster assignment overlap in the ones predicted for each ML model output. A higher difference percentage indicates less co-cluster overlap and therefore, less similar representation of protein sequences between two ML models.

6. RESULTS

Explanation of Datasets

Before delving into the intricacies of mathematical modeling and metric evaluation portions of the project, we first need to take a closer look at the data. Our data is complicated and very confusing to understand with many different files, subfolders, and datasets to keep track of. The datasets we are working with typically come in the form of around one or two thousand columns by some thousands of rows. The initial data we received focused on two models: UniRep and XLNet. We later take a closer look at the p53 protein which came with more datasets and an additional model, ESM, to go along with the two aforementioned models.

In terms of the shape of the data, the number of columns is determined by the type of model (UniRep, XLNet, or ESM) that the protein sequence is fed into while the number of rows is simply the number of protein sequences that were inputted. Each row represents a latent vector representing a given protein sequence that can be found in the last column of the dataset. We utilized a series of different exploratory data analysis approaches in order to obtain a better grasp of the data and unearth any underlying trends for both the initial data as well as the p53 case study.

The initial data we received was in the form of two types: ProteinGym and SeqDesign. The ProteinGym files contained six datasets. Each dataset contained a real protein amino acid sequence and the rest of the set was made of "fake" protein sequences – generated by substitutions and deletions of the original real sequence. On the other hand, the SeqDesign consisted of four robust datasets that each contained a subset of different typed proteins. SeqDesign sequences were all real amino acid sequences. The source of these SeqDesign files was explicitly given to us, so the origin of these datasets is unclear.

A short time after working with the initial data, we shifted our focus to a specific type of protein, p53, as suggested by our sponsor. p53 is a type of tumor protein and acts as a tumor suppressor by preventing cells from dividing uncontrollably [7]. Aside from the important nature of this protein, the selection to analyze p53 may seem random. However, it acts as an important case that provides us insight into whether or not the models agree when it comes down to specific types of proteins. While there are countless proteins to analyze, the p53 case study should provide enough results to fulfill the scope our team has set.

There are four datasets classified as p53 input. Three of the four datasets are very similar to one another. So much so that the team does not always use all four datasets in analysis. For more computationally expensive analysis, the team chose to use two of the four datasets instead of all four. For less expensive analysis, the entire population of p53 data was used. Hence the p53 analyses have discrepancies in the amount of data used.

6.1 Data Preprocessing

First, to get a higher level understanding of the initial data, we decided to concatenate the data together and separate it by the type of data (ProteinGym/SeqDesign) and by the type of model (UniRep/XLNet). In short, all UniRep files in ProteinGym were concatenated together while the same was done for XLNet; the same process was done with the SeqDesign excerpts. By consolidating all relevant datasets down to a few files, we are able to get a better understanding of the shape of the remaining datasets while limiting the number of extraneous variables. While we initially did an analysis on both the ProteinGym and SeqDesign sections, the scope of this project shifted to prioritize our findings on solely ProteinGym for the sake of time. Consequently, the preliminary results discussed in this section will mainly focus on the results obtained from ProteinGym.

Additional, smaller preprocessing steps were taken, including finding and removing missing values in the datasets as well as standardizing the values in each dataset by removing the mean and scaling to unit variance. In order to transform our data in this manner, we leveraged preprocessing functions from Scikit-learn such as `StandardScaler()`. Scaling our data makes it easy for our model to learn and understand the data which makes this step an important part of the preprocessing process.

6.2 Exploratory Data Analysis: ProteinGym Data

Once we had some sort of understanding of the initial data, we were able to take the initial steps into exploring the data. One of the first things that came to mind was to see if the length of each of the protein's amino acid sequences was a factor in the latent vector output produced by UniRep and XLNet. One issue is that each row output is a thousand by one vector which cannot be easily represented. In order to get a standardized value for each row vector, we decided to compute the magnitude of each vector. While not a perfect solution, it works well to give us a standard, representable value for each entry in each dataset for both models. Another issue is the usage of sequence length when solely analyzing the ProteinGym files as all of the sequence lengths in their respective files are the same resulting in less datapoints to observe. As a result, this may not provide as much insight into whether or not amino acid sequence length plays a role in the latent vector output. Regardless, the datasets at hand do have a wide range of sequence lengths that changes from dataset to dataset which may give some understanding into whether the models interpret the protein sequences similarly or differently.

By plotting the average magnitude of the proteins in each model dataset and sorting them by the protein's amino acid sequence length, we can see if there is a similar trend between the two. We are not so much as comparing the magnitude values between each protein in both models, but rather observing if there is a similar trend in terms of magnitude when the sequence length of the proteins change.

Using Matplotlib, we were able to plot the average magnitude of the latent vectors of each dataset sorted by sequence length for both UniRep and XLNet of ProteinGym as seen below in figure 2.

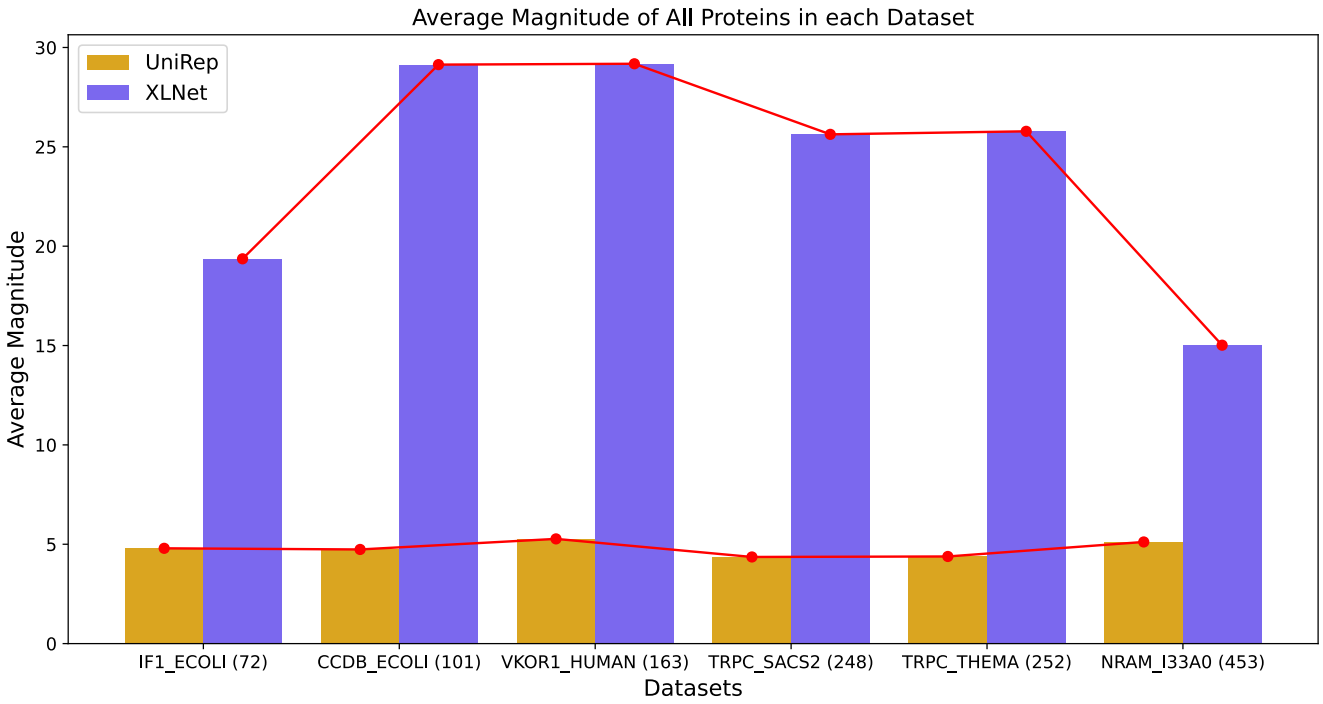

Figure 2: Average Magnitude of all ProteinGym Datasets

As seen in figure 1, the average magnitude of the proteins for both model versions of each dataset are plotted; the two models are differentiated by color. With the help of a trend line, we can compare how the magnitude changes when the sequence length increases. While there does not seem like there is a clear trend between how long a protein's amino acid sequence is and the magnitude value of the protein. However, we do see an major difference in how the magnitude changes between the UniRep and XLNet datasets. XLNet's magnitude increases substantially from the first to the second dataset while decreasing significantly from the second last to the last dataset. On the other hand, it seems that UniRep's magnitude mostly stays stagnant with some minor shifts seen in the second to the third dataset as well as from the third to the fourth dataset. A key difference is in the second last to the last dataset where UniRep's magnitude increases instead of decreases. While this plot does not tell us much about the affect that amino acid sequence length has on the latent vector output, it does demonstrate some signs that the two models UniRep and XLNet may interpret proteins differently. While this analysis provides some insight into each model's behavior, we must leverage principal component analysis in addition to our clustering technique in order to obtain a more concrete answer to our research question.

6.3 Exploratory Data Analysis: p53 Case Study

Our exploratory data analysis on the ProteinGym data gave us some insight on the data, but raised some questions about the behavior of the models. The initial ProteinGym data was made up of a variety of different types of proteins including ECOLI, HUMAN, SACS2, THEMA, and I33A0. We hypothesize that these models may approach distinct types of proteins in a more similar or different manner. Perhaps, the models are more similar when comparing a single dataset of HUMAN proteins rather than in a dataset of various mixed types of proteins. By limiting our data scope down to one specific type of protein, we should be able to extract information on whether these models behave dissimilarly when focused on one

protein or if there is no distinct different with the proteinGym datasets. As mentioned in section 6.1, we plan on conducting a case study on the p53 human protein which comes with an additional model and four new datasets to analyze. Our next steps include doing a similar sort of analysis on the p53 protein and its datasets.

When beginning to analyze the data, we realized that our approach to p53 would be somewhat different than the previous analysis on the proteinGym data. To start, all four datasets are centered on the same protein, but are differentiated by the patterns of substitution among them. This means that we cannot use sequence length or the shift from different typed datasets to pull out any trends or patterns to compare the models to. As a result, we found it difficult to do a substantial amount of basic analysis on these datasets. Additionally, because the four datasets analyze the same p53 protein, it seemed redundant to do the same analysis on all of them. The resulting analysis centers around the comparison of the average magnitude between the three models.

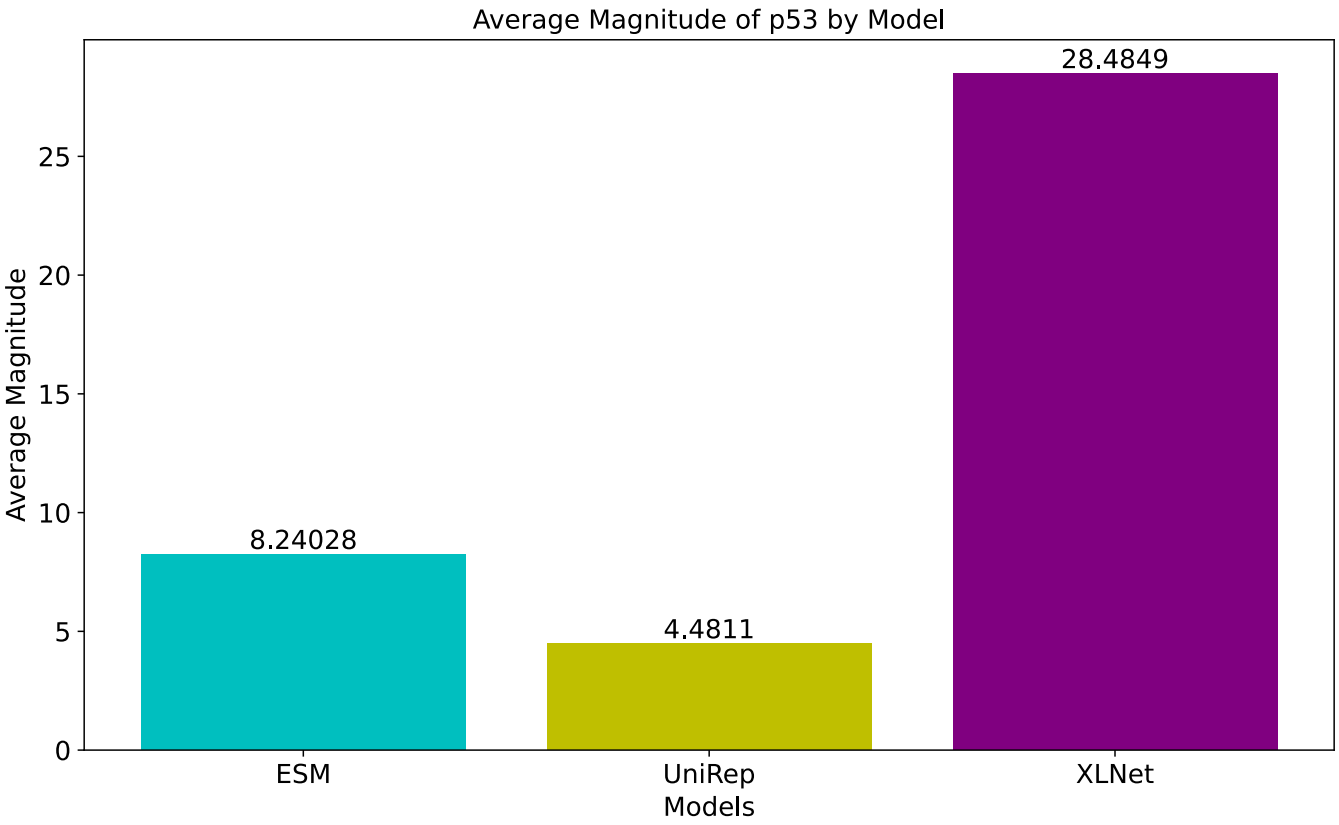

Figure 3: Average Magnitude of all ProteinGym Datasets

Figure 3 reveals the average magnitude of the protein p53 sorted by the ESM, UniRep, and XLNet versions of the p53 dataset. While this plot does not tell us much about how the models are similar or dissimilar through trend lines of other proteins, it does provide some insight into how the average magnitude of the models differs from model to model. This magnitude value does not inherently tell us whether or not the models interpret protein sequencing as all of the models have different dimensions. Instead, it should give us some more understanding of the values that are held within each latent vector. The ESM, UniRep,

and XLNet modes have 1281, 1901, and 1025 dimensions respectively. The fact that the smallest model by dimension has by far the largest magnitude tells us that the values in the vector space should be much larger than the other two models. Meanwhile, ESM has around 600 fewer dimensions than UniRep but still has about twice its magnitude. Perhaps, the more dimensions a vector spans, the more spread out the values have to be for each dimension which would result in a much lower magnitude. While this information may not be the deciding factor that answers our research question, it is definitely an element to keep in mind going forward.

6.4 Optimal Number of Clusters: ProteinGym Data

The first step in our clustering-related results was to perform optimal clustering analysis (as described in section 5.2.1) For this first round of optimal clustering, the investigation was performed on the Principal Component Analysis reduced outputs from XLNet and UniRep on ProteinGym inputs.

As described previously, our team opted to use Gaussian Mixture Model for our clustering technique. In Scikit-learn, the Gaussian Mixture Model packages allow one to calculate the Bayesian Information Criterion (BIC) for the resulting clustering model. Recall that the lower the BIC, the better the model in this analysis. The following tables show the optimal number of clusters alongside the BIC of the optimal clustering model for XLNet and UniRep outputs on each of the 6 input datasets in ProteinGym. Our initial data contained XLNet and UniRep outputs from ProteinGym inputs, so those are the only two models being compared below.

Dataset	Optimal k for XL-Net	BIC
CCDB_ECOLI_Adkar_2012	1	296.97
IF1_ECOLI_Kelsic_2016	2	1022.46
NRAM_I33A0_Jiang_standard_2016	1	-51.87
TRPC_SACS2_Chan_2017	3	-814.72
TRPC_THEMA_Chan_2017	1	-944.70
VKOR1_HUMAN_Chiasson_activity_2020	1	5.75

Table 1: Optimal Number of Clusters for XLNet output on ProteinGym inputs

Dataset	Optimal k for UniRep	BIC
CCDB_ECOLI_Adkar_2012	3	-699.22
IF1_ECOLI_Kelsic_2016	3	-650.92
NRAM_I33A0_Jiang_standard_2016	2	-248.41
TRPC_SACS2_Chan_2017	4	-1352.13
TRPC_THEMA_Chan_2017	5	-1387.38
VKOR1_HUMAN_Chiasson_activity_2020	2	-385.33

Table 2: Optimal Number of Clusters for UniRep output on ProteinGym inputs

Observing the optimal k for XLNet and UniRep between the 6 different datasets in table 1 & 2, it is apparent that *none* of the optimal number of clusters for each dataset match these two models. UniRep outputs always have a higher number of optimal clusters than XLNet outputs, which is an interesting trend

to note and emphasizes the difference between the two model spaces. Further, comparing the BIC values generated between the same inputs for UniRep and XLNet, we see that they are very different. XLNet BIC scores range from 1022.46 to -944.70, while all BIC scores for UniRep clustering models are well below 0. If the resulting model spaces created by ProteinGym data were the same for XLNet and UniRep, there should be the same optimal number of clusters for each output and similar BIC scores among the optimal models. This is clearly violated and provides significant evidence that there are differences in the resulting model spaces between XLNet and UniRep.

6.5 Optimal Number of Clusters: ProteinGym Data, p53 Case Study

After the overview of the models given by the optimal cluster analysis of ProteinGym inputs, the team redirected our efforts to a more specific case study. Our client thoughtfully suggested analyzing the differences in the model space outputs among a set of specific protein inputs – inputs generated from protein p53.

For the p53-specific data, there were 4 datasets provided as input to 3 different machine learning models. XLNet and UniRep were models evaluated in the previous section, but our sponsor also included another model for analysis, ESM. Thus the optimal clustering analysis was completed just as it had been for the overall ProteinGym inputs, but now for p53 inputs on three different model outputs (XLNet, UniRep, ESM). The following tables show the optimal number of clusters alongside the BIC of the optimal model for each of the 4 datasets generated by p53 data.

Dataset	Optimal k for XL-Net	BIC
p53_HUMAN_Giacomelli_NULL_Etoposide_2018-protein	5	-6039.18
p53_HUMAN_Giacomelli_NULL_Nutlin_2018-protein	4	-6068.57
p53_HUMAN_Giacomelli_WT_Nutlin_2018	4	-6020.18
p53_HUMAN_Kotler_2018-protein	2	-766.24

Table 3: Optimal Number of Clusters for XLNet output on p53 inputs

Dataset	Optimal k for UniRep	BIC
p53_HUMAN_Giacomelli_NULL_Etoposide_2018-protein	4	-6915.58
p53_HUMAN_Giacomelli_NULL_Nutlin_2018-protein	4	-7083.60
p53_HUMAN_Giacomelli_WT_Nutlin_2018	4	-6906.65
p53_HUMAN_Kotler_2018-protein	4	-546.61

Table 4: Optimal Number of Clusters for UniRep output on p53 inputs

Dataset	Optimal k for ESM	BIC
p53_HUMAN_Giacomelli_NULL_Etoposide_2018-protein	2	-14976.67
p53_HUMAN_Giacomelli_NULL_Nutlin_2018-protein	4	-15068.14
p53_HUMAN_Giacomelli_WT_Nutlin_2018	2	-15192.31
p53_HUMAN_Kotler_2018-protein	1	-2034.44

Table 5: Optimal Number of Clusters for ESM output on p53 inputs

Among all three models shown in table 3, 4 & 5, the only dataset in which the outputs share the same number of optimal clusters is the second dataset, p53_HUMAN_Giacomelli_NULL_Nutlin_2018-protein. Otherwise, no other dataset has an agreement across all three models. Comparing the optimal number of clusters alongside the BIC score for each optimal model, ESM varies significantly from the other two models. With the exception of the one dataset all models agree on, ESM has noticeably fewer optimal clusters and has much smaller BIC scores than the other models. Thus ESM outputs differ from UniRep and XLNet outputs on p53 inputs.

While there is little agreement among all of the models simultaneously, UniRep and XLNet agree on the optimal number of clusters for the second and third datasets (p53_HUMAN_Giacomelli_NULL_Nutlin_2018-protein, p53_HUMAN_Giacomelli_WT_Nutlin_2018) respectively. They also have pretty similar BIC scores for each of their optimal models, which may also point to some similarity between the model spaces of UniRep and XLNet. Thus we cannot conclude that XLNet and UniRep model spaces are significantly different on p53 inputs through this optimal cluster analysis.

When comparing the latent space of all three models, this optimal clustering analysis indicates that there are significant differences between ESM and the other two models. However, since there is evidence both for and against UniRep and XLNet models having similar latent space representations, the conclusion on the similarity between these two models' spaces was deferred to an analysis completed in the following section.

6.6 Co-Clustering Behavior: ProteinGym Data, p53 case study

Recalling the model evaluation component of the project, we assessed the similarity of protein representations across two given models by analyzing the co-clustering assignment behavior learned by a GMM. An 80%-20% training-test split was applied to each dataset when used in fitting the GMM model for clustering. After the GMM is fit to the training data, it predicts a cluster assignment to each sample protein in the test data.

The combinations of dataset (ProteinGym or SeqDesign), protein type (P53 or otherwise), specific source, and the number of clusters are innumerable. To observe co-clustering analysis results, we follow one such example performed on the ProteinGym p53 protein dataset from the Giacomelli dataset using an arbitrarily chosen cluster number of $k = 5$. Figure 4 depicts a clustering operation on the Giacomelli trial processed by XLNet.

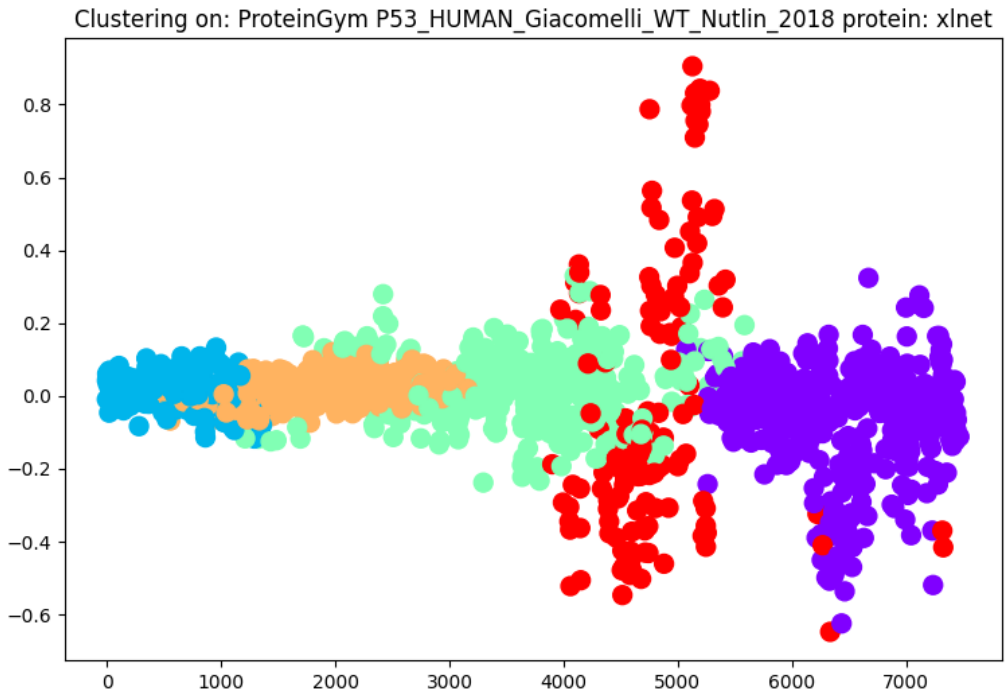

Figure 4: Clustering assignments predicted by a Gaussian Mixture Model for $k = 5$ clusters.

When performing clustering on this dataset of the XLNet output, the clusters appear to be clear and generally well-defined. One could identify to which cluster a 2D protein representation at a given coordinate plausibly belongs. This is an indication that there is something semantically meaningful about the latent features of the protein representation, even when reduced to two dimensions by PCA. There is reason to believe that the vector mapping from sequence to latent vector preserves unseen semantics about the biology in a protein sequence, learning something meaningful about the biology. This reflects positively on the viability of these machine learning models in representing protein sequences in their latent spaces.

As previously mentioned, however, evaluating these models for similarity with each other entails examining whether their output's clusters cluster the same protein together. Performing a pairwise comparison of set differences of the protein representations in clusters $i = 0...k$ on one model and clusters $j = 0...k$ from another assesses all possible comparisons which is necessary because of the arbitrary clustering order. Figure 5 illustrates two heatmaps comparing XLNet output clusters against UniRep and ESM clustering similarity of the Giacomelli dataset when using five clusters.

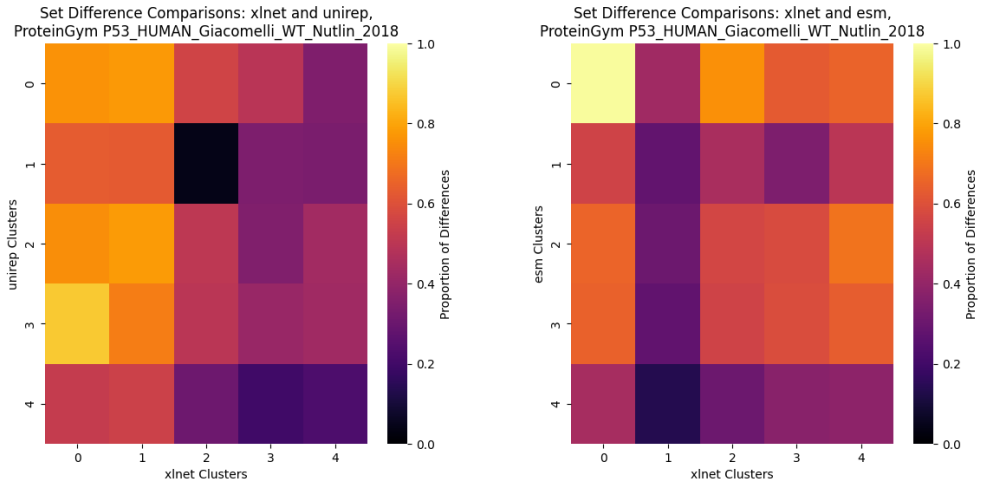

Figure 5: A heatmap of set difference proportions across all clusters made on the XLNet output against those of UniRep and ESM outputs. Warmer colors indicate a higher difference for the comparisons of models' clusters at a row, $i = 0...k$, and column, $j = 0...k$.

As can be seen from the heatmaps of these set difference comparisons on the Giacomelli dataset, the clustering behavior is considerably different across model outputs. The greater number of warmer-colored squares indicates more cluster comparisons between the two models that do not share the same unique protein sequences. The protein sequences that were clustered together from one ML model output are clustered differently for assignments in another ML model output. In other words, clusters do not have the same meaning for different ML model outputs; there is no obvious equivalent of a cluster in one model that exists in another. We can observe the heatmaps now comparing UniRep output clusters against XLNet and ESM clustering similarity of the Giacomelli dataset using five clusters again.

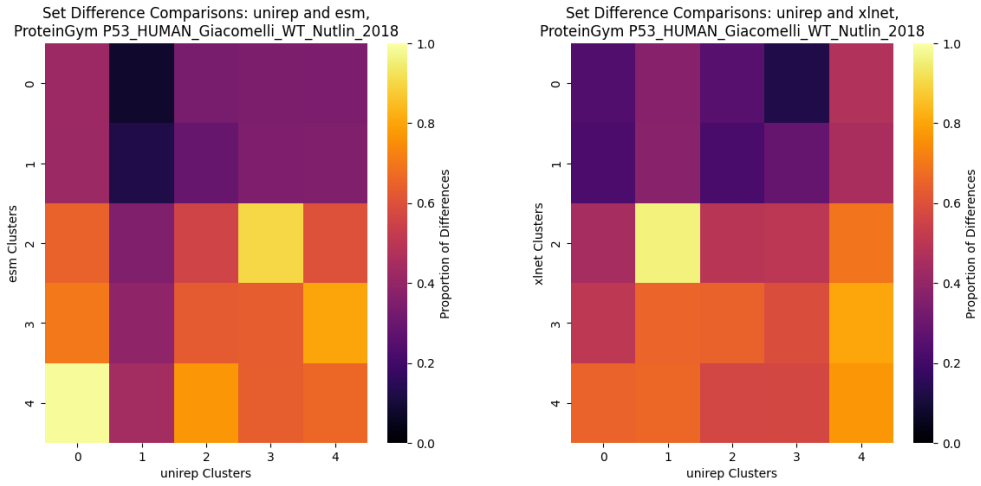

Figure 6: A heatmap of set difference proportions across all clusters made on the UniRep output against those of XLNet and ESM outputs. Warmer colors indicate a higher difference for the comparisons of models' clusters at a row, $i = 0...k$, and column, $j = 0...k$.

The frequency of moderately high set difference comparisons once again indicates that the three models

are quite different in co-clustering behavior. This furthers earlier results that claim that the ML models ESM, UniRep, and XLNet have different latent space representations. Because the clustering behavior appears inconsistent, there is reason to believe that they are representing protein sequences differently even when learning from the same ones. Models that were more similar in latent space representation, would intuitively have consistent clustering behavior as indicated by a number of definitively dark squares rather than middling colors as in Figure 5. Because of the arbitrary clustering, not all squares will be dark in such a scenario, but the intensity of coloration should be much darker on the heatmap.

Although these analyses are of the same Giacomelli dataset, such co-clustering patterns appeared to be quite similar regardless of the dataset or models used. Taking a sneak peek at such a phenomenon, another analysis can be condcuted, this time on the ProteinGym p53 dataset from the Kotler experiment.

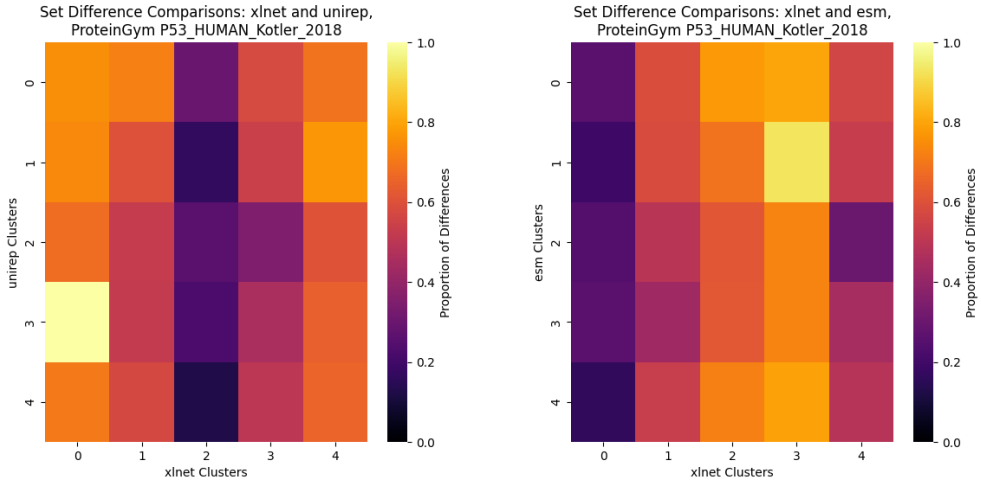

Figure 7: A heatmap of set difference proportions across all clusters made on the XLNet output against those of UniRep and ESM outputs. Warmer colors indicate a higher difference for the comparisons of models' clusters at a row, $i = 0...k$, and column, $j = 0...k$.

From the heatmaps of set difference comparisons in Figure 7, now on the Kotler dataset, the cluster comparisons across two given models appear even more dissimilar, although the ones that are similar are similar to a more intense degree. Based on these results, although the ML model latent representations of proteins appear to be different the dataset used appears to play a role in how similar or different the learned representations appear to be.

7. LIMITATIONS

There are some limitations that may restrict the use and applicability of our model. Our model seeks to determine whether or not these protein engineered models' output are similar or different through the use of dimensionality reduction techniques. The input for our model would have to be at least two latent vector output .csv files that have been run through some protein engineered models (ESM/UniRep/XLNet) for our script to produce some interpretable result. Our model may discover that for particular proteins, the models different rather than similar, but will not go into the explanation of the results, as that was outside of our scope. Our model is limited in this sense as it does not provide a reasoning behind this phenomena, but instead gives a yes or no answer to a yes or no question. This model would certainly be limiting to scientists or engineers that would like to know the understanding behind the results.

There were some additional limitations that came in the form of data legibility or lack thereof. The data that we received was complicated and at times faulty. The initial data received was convoluted as there were many folders and subfolders to keep track of. These subfolders were the source datasets that contained each model .csv files. All of these .csv files were named after the model and there was no differentiating between two `unirep.csv` files from different subfolders. A lot of time was spent deciphering which datasets we were working with which caused a lot of confusion in the early stages of the project. There were also instances where our scripts would stall on some dataset citing 'missing values', but no missing values were found after some examination. Working with this data was rocky at times which definitely limited the time spent discovering results.

8. RESULTS, FOR CLIENT

Analyses done in the Results sections find that XLNet, UniRep and ESM have different latent space representations for ProteinGym inputs and specifically, p53 inputs. The results contribute to a greater understanding of these "black box" protein sequence machine learning models. Conclusions made here also highlight the need for further research into each one of these models individually. While we are sure there are some similarities among the models, our findings that these model spaces are different on our tested inputs indicates that further investigation into the properties of these models will be more successful if those investigations are done model by model. Furthermore, various protein sequencing models cannot be assumed to act similarly, and it might be worthwhile to investigate what kinds of data or what applications these models are best suited for.

9. TEAM ROLES

- **Isabelle Fox**
  - Technical contribution: Isabelle created the clustering methods that were later pipelined by Brian. She also wrote functions for and performed the optimal clustering analysis
  - Nontechnical contribution: Scheduled weekly meetings with Dr. Tonner, presented a portion of the Tools & Techniques presentation, and created and delivered the Elevator Pitch
- **Jonathan Jwa**
  - Technical contribution: Jonathan worked on the mathematical modeling script to conduct dimensionality reduction on our high-dimensional data set allowing us more flexibility with our further analyses and visualizations.
  - Nontechnical contribution: Jonathan presented the Tools and Techniques Presentation.
- **Brian Lee**
  - Technical contribution: Brian programmed pipelines to obtain or process results in a streamlined fashion, particularly for mathematical modeling tasks and obtaining results. Brian ran these analyses as well.
  - Nontechnical contribution: In addition to delivering the Midterm Presentation alongside Joseph, Brian took special care to document aspects of the group's progress, including to-do lists and sponsor-team meeting notes.
- **Joseph Wu**

- Technical contribution: Joseph primarily worked on the exploratory data analysis section and produced humble visualizations on the findings.
- Nontechnical contribution: Joseph contributed to the Tech Memos and delivered parts of the Midterm Presentation alongside Brian.

10. CONCLUSIONS AND FUTURE WORK

In this project, our team, the Amino Amigos, has attempted to answer the stunning question of whether the latent representations of protein sequences by machine learning models are similar or different. Analyses used to answer this problem statement were primarily centered around clustering methods and observing the clustering behavior to examine anything hinting at the underlying semantics of the data indicating similarity.

Based on the results that were obtained, the team was able to conclude with confidence that the expression of protein sequences by these machine learning models is different in their representation. At the very least, there is not enough evidence to suggest that these learned representations of proteins are similar. Not only is the algorithm to find an optimal number of clusters to which to fit the data cannot be agreed upon, but also, pairwise comparisons of clusters revealed no consistent similarities in clustering behavior.

There are many directions in which to further the work on this project. Most simply, we welcome the confirmation or challenge of the conclusions made in this project by replicating it with both a wider range of dataset sources as well as additional ML models to output this new data in its own latent representation to further the problem statement. In addition, future work can also examine the role of different methods used to answer the problem statement, something our team could have done given more time. For example, although we employed principal component analysis to reduce the dimensionality of the datasets, a variety of other mathematical modeling techniques exist, such as manifold learning methods, or canonical correlation analysis, both of which might preserve the original semantics of the data differently. Lastly, future work lies in reconnecting such results back to the biological application of this technology. This aspect of future work is highly interdisciplinary and remains beyond our expertise, but it is exciting to think about what these results mean for the field of computational biology. Assuming our conclusions are correct, bioengineers may be able to study if there is any biophysical reason proteins are being represented in these different models. Identifying such a relationship may be able to sharpen the efforts of machine learning engineers to construct a model architecture that is robust for a protein of interest, like the p53 protein, which is involved with cell growth regulation. By connecting future work on this project with its biological origins, perhaps it can help create treatments for those affected by biological ailments much later down the line.

Special thanks to Dr. Peter Tonner of NIST, who graciously allotted his time to help us complete this project to the best of our ability, even when finding new employment during the semester. He consistently gave helpful input to maximize our effectiveness with solution approaches and acted as a translator of biological background to make a challenging project highly interesting and relevant.

REFERENCES

[1] Dabagia, Max, Konrad P., Kording, and Eva L. Dyer. "Comparing high-dimensional neural recordings by aligning their low-dimensional latent representations." arXiv preprint arXiv:2205.08413 (2022).

[2] Wang, Chang, and Sridhar Mahadevan. "A general framework for manifold alignment." 2009 AAAI Fall Symposium Series. 2009.

[3] Yang K. Kevin, and Wu, Zachary, and Bedbrook N, Claire and Arnold H., Frances. "Learned protein embeddings for machine learning." Bioinformatics, vol. 34, no. 15, pp. 2642-2648, Aug. 2018, doi:10.1093.

[4] Service, Robert. "Protein structures for all." Science, 16 Dec, 2021 [Online], doi:10.1126/science.acx9810

[5] Meta. "New AI Research Could Drive Progress in Medicine and Clean Energy." Meta Newsroom, 1 Nov, 2022 [Online].

[6] Alley, E.C., Khimulya, G., Biswas, S. et al. Unified rational protein engineering with sequence-based deep representation learning. Nat Methods 16, 1315–1322 (2019). https://doi.org/10.1038/s41592-019-0598-1

[7] MedlinePlus [Internet], "TP53 gene: Medlineplus genetics," MedlinePlus. [Online]. Available: https://medlineplus.gov/genetics/gene/tp53/. [Accessed: 07-Dec-2022].

“We have neither given nor received unauthorized assistance on this assignment.”