Unsupervised Feature Learning for Multimodal Protein Structure Encoding

Peter Howell [1 2] **Kai Davidson** [1 2] **Alexsandra Antoski** [2 3]

Abstract

Protein data is large and unruly to work with so a solution of an embedding of protein data would allow more efficient downstream training would help decrease the computational load on tasks relating to proteins. We expand on the work of Nguyen & Hy on multimodal protein autoencoding. We employed an experimentally verified dataset, changed their VGAE implementation, and employed attention and a concrete autoencoder, all as potential improvements to their approach. Our approach of VGAE led to improvements in the graph embedding by 0.4% on the enzyme identification downstream based on our data. Our best performing AutoEncoder for the fused data had a loss of 0.25 MSE.

1. Introduction

Biological systems are complex to understand. Figuring out how they work requires a deep understanding of the many organic molecules within an organism, but also experimental data to verify these results. This data can end up being massive in scale, so the shift to using computational methods for analyzing it has led to the development of fields like Bioinformatics. In the last decade, there has been frequent development of novel methodologies that take advantage of the vast amounts of biological data, and there has been much success using techniques in Deep Learning like AlphaFold, which recently won a Nobel Prize (Jumper et al., 2021). In particular, proteins are an essential building block of biology, and there are numerous databases devoted to storing all of the data associated with the expression of these proteins. We wanted to leverage deep learning approaches to try to learn meaningful representations from protein data while minimizing storage costs. In addition, creating an embedding that maintains the important features of proteins may increase the accuracy with downstream tasks.

1.1. Research contributions

We leveraged existing work done by Nguyen & Hy (2024) who also tried to tackle this problem. We modified some of their approaches and changed their dataset to purely experimentally verified protein structures to address an issue we saw with their experimental method. We are verifying their results with a smaller dataset that is based on the biology and research that has been experimentally done up to this point. We tested the efficacy of this task on the data that is available through traditional methods. We also changed the graph embedding method which we believe improved our results in this modality, though due to the dataset difference we can't be entirely sure. We tested different autoencoders for our fused data as well to measure the difference between the efficacy of encoding models. As well we varied the number of features to embed the data which varied our results further. All of the results we derived lead to helping point out some inconsistencies with replicating their work on experimental proteins, as well as future approaches that could be applied to increase performance.

2. Related Work

The paper "Multimodal pretraining for unsupervised protein representation learning" by Nguyen & Hy (2024) delves directly into our topic of interest. The issue we found with their approach is that they use solely generated structures from AlphaFold. Their dataset consisted of approximately five hundred thousand generated proteins by AlphaFold. This raises concern with the overall validity of their results. AlphaFold, by Google DeepMind, is an AI model that predicts protein structures with accuracy comparable to experimental results (Jumper et al., 2021). Although accurate, these structures are still uncertain and not all reviewed. Additionally, AlphaFold's performance "declines rapidly" for proteins with over two chains (Bryant et al., 2022). We propose that using purely generated structures could cause uniformity as well as embed any AlphaFold model folding bias into their embeddings. Another consideration of using generated structures is the confidence level for certain structures. When processed in

[1]Program in Bioinformatics & Computational Biology, Worcester Polytechnic Institute [2]Department of Computer Science, Worcester Polytechnic Institute [3]Program in Data Science, Worcester Polytechnic Institute. Correspondence to: Peter Howell <pjhowell@wpi.edu>.

CS/DS 541 Deep Learning Final Project, Worcester Polytechnic Institute.

their code, the embeddings don't have any consideration for the confidence level of the AlphaFold structures leading to training on these structures just producing noise from the error.

Details of their implementation, also brought up possible improvements in their methodology. Their VGAE embedding wasn't based on the biology of distance between residues, so we modified their implementation to be and some inconsistent error rates made this an interesting problem to do further research on.

The other work that we drew inspiration from during our project was the paper preprint from (Abid et al., 2019). We wanted to see how the use of a temperature value could improve embedding down to a specified dimension for our protein encodings.

3. Proposed Method

Our pipeline was very similar to that of Nyugen and Hy, though we had a few developments we hoped would offer improvements (Nguyen & Hy, 2024). The overall process consisted of preprocessing to transform the raw protein structure into a graph, point cloud, and tokenized sequence, then using an autoencoder on each of these three modalities, then combining these representations and using a final autoencoder on this combination. Firstly, we altered the construction of the graphs to be more grounded in biochemistry, connecting two nodes with an edge when they are within 6 Å of each other (Shervashidze et al., 2011). We also experimented with several different autoencoders at the last step of the pipeline. We replace the AutoFusion method with a variation of Concrete Autoencoders, which use temperature to introduce noise during the training process to gradually reduce the input to the latent space (Abid et al., 2019). We added a layer of multi-headed self-attention prior to the concrete autoencoder to hopefully contextualize the concatenated representations and enable a more meaningful final representation to be learned.

In addition to altering the computational architecture, we also elected to use a different data source. Due to the reasons mentioned in the previous section, we stuck with experimentally verified and annotated structures. We started with roughly 36,000 mmCIF files from UniProtKB, all of which had been reviewed and had an experimentally verified 3D protein structure.

Experimental structures are much more expensive and difficult to verify. The leading method is using X-Ray Crystallography which researchers use to find the structure of proteins down to a few angstroms of precision. The issue is the arduous process involved with finding the 3D structure of any protein can take months, which is why at the current moment there are so few experimentally validated protein structures.

Nyugen and Hy processed modalities ignoring non-standard amino acids, as in they skipped adding a value when processing graphs and sequences. Here we ignored proteins that contained any non-standard amino acids, giving us a more honest representation of the protein's structure.

Finally, we swapped out their autoencoder implementation, and tried various replacements: a concrete autoencoder with and without self attention, and a modified autoencoder architecture also with and without self attention.

4. Experiment

There are three parts where we evaluated our model's success; comparing success encoding and decoding; implementing two of the downstream tasks proposed in the original project, and seeing the performance of the original project given our dataset.

We compared the performance of encoding and decoding using MSE. We did not modify the PAE or ESM-2 models, but since they are trained on a different set of data this gives us some insight on the performance of these models given reviewed protein structures.

We also trained Nguyen & Hy's exact VGAE with their method for graph construction on our data. We saw that in this case, our model outperformed theirs, achieving lower reconstruction loss on pretraining and higher accuracy in the protein-ligand affinity downstream task.

Since the bulk of the computational load came from the initial processing and encoding stage, we were able to train several versions of the last layer encoding. This included concrete and not concrete autoencoders with and without attention. We were not able to experiment with other hyperparameters like the initial temperature of the concrete autoencoders. We used each of these four models in our downstream tasks.

For the two downstream tasks, we have enzyme identification and protein ligand-binding affinity. Given the time to process and train our models and lack of data availability, we were unable to fully implement all four downstream tasks. To give the best comparison we used the same hyperparameters and evaluation metrics.

The final evaluation we took was to compare the results of our pipeline versus the original using our dataset. We only were able to compare their VGAE on our model but due to the computational cost of encoding all proteins with the models we were unable to run their entire codebase on our protein data.

5. Results

For both pretraining and downstream tasks, Nyugen and Hy's work performed better than ours, other than pretraining on VGAE. The results from VGAE pretraining on the peer-reviewed structures indicated that the residue based graph construction is potentially a better approach, as less data was needed to train the model to the performance of Nyugen and Hy's.

Data	Metric	Nguyen & Hy	Our Results
Train	MSE	0.951	0.9896
Validation	MSE	0.952	0.9951
Test	AUC	0.95	0.9551
MSE	Precision	0.97	0.9716

Table 1. VGAE pretraining results

For the PAE, we used the same architecture and process but trained on only peer-reviewed protein structures. This PAE model had a high MSE, nearly double that obtained by Nguyen and Hy.

Data	Metric	Nguyen & Hy	Our Results
Train	MSE	6.51	11.7226
Validation	MSE	7.84	12.97
Test	Chamfer	7.89	12.8688

Table 2. PAE pretraining results

The results from pretraining our multiple configurations of the CAE did not perform better than the MPRL model. Our best performing configuration was a standard autoencoder with the following architecture:

Nguyen & Hy : $640 * 3 \rightarrow \frac{(640*3)}{2} \rightarrow 1024$

Ours: $640 * 3 \rightarrow 640 \rightarrow 64$

The results of the pretraining of the multimodal models can be seen in the table below.

Model	Train	Validation	Test
Nguyen & Hy	0.03	0.299	0.03
Our CAE	0.4123	0.4109	0.4121
Our CAE + Attention	0.3964	0.3933	0.3945
Our AE	0.2513	0.2540	0.2506
Our AE + Attention	0.2613	0.2671	0.2636

Table 3. Multimodal pretraining reconstruction MSE results

The inclusion of the concrete layer in our final encoder did not benefit the model's performance, nor did attention. Our simple autoencoder with and without attention outperformed the validation loss of Nguyen and Hy, however not the train or test loss. This was a cause for skepticism that we elaborate on in the discussion.

In the downstream tasks, models trained on our data did not perform as well. In the first task of enzyme identification, we achieved relatively lower accuracy than Nguyen and Hy's models. However, we were able to duplicate their graph approach but train it on our data, and found that our graph approach with radial edge connections outperformed it 4.

Model	Accuracy
Single-Mode Encoding	
VGAE (Nguyen & Hy)	**81.4**
VGAE (Nguyen & Hy Our Data)	72.35
VGAE (Ours)	***72.75***
PAE (Nguyen & Hy)	**80.4**
PAE (Ours)	62.44
ESM-2 (Nguyen & Hy)	**82.7?**
ESM-2 (Ours)	75.4
Multimodal Encoding	
MPRL (Nguyen & Hy)	**83.9**
AE	67.45
AE + Attention	65.75
CAE	65.16
CAE + Attention	59.19

Table 4. Enzyme identification downstream task results

Our PLA results didn't show any improvement on that of Nguyen & Hy. As seen in the table below. This shows that the reduction in performance could be linked to relative size of the datasets.

Model	MSE ↓	CI ↑	r_m^2
Single-Mode Encoding			
VGAE (Nguyen & Hy)	0.508	0.631	0.032
VGAE (Ours)	0.818	0.529	-0.152
Multimodal Encoding			
MPRL (Nguyen & Hy)	0.248	0.699	0.414
AE	0.733	0.578	-0.55
AE + Attention	0.740	0.5777	-0.454
CAE	0.740	0.5779	-0.453
CAE + Attention	0.733	0.578	-0.55

Table 5. Protein ligand binding affinity downstream task results

6. Discussion

The models trained on our smaller dataset of experimental structures performed worse in every metric on the downstream tasks. However, we evaluated their VGAE implementation on our data set and the result aligned with our findings from pretraining, indicating that radius-based edge creation may be more effective at encoding the graph representation of a protein.

As mentioned previously, the PAE model was not modified in architecture, but it was trained on a different dataset. Here there was not as significant a difference in error as the autoencoder implementations. We believe that the higher error rate is indicative of the variability of the experimental structures compared to what might be more uniform structures generated by AlphaFold. In addition, a larger dataset may aid in the encoding of point cloud structures as it may be able to find uniform patterns among all of the data leading to a lower overall loss during reconstruction.

It is worth mentioning that the same pretrained model ESM-2 performed worse when we evaluated this task. This confuses us as the pretrained model has not been updated for over 2 years and we are evaluating off of the same dataset proposed in their paper. Similarly, our modifications did not compare to the results given by Nyugen and Hy. We are therefore somewhat skeptical of the results presented by Nguyen and Hy.

In terms of the performance of the multimodal representation, we believe its poor evaluations were in part due to the decrease in latent dimensionality, which was far less than Nyugen and Hy. The embeddings may have been reduced in dimensionality too much to preserve enough information to learn from.

Downstream tasks stuff here.

We wanted to see if only experimental structures would be enough. The variety, heterogeneity, and noise present in experimental structures makes training autoencoders difficult with this relatively small amount of data. The success that came from this was that we were able to derive a VGAE model that minutely improved the work done by Nguyen & Hy (Table 1).

7. Conclusions and Future Work

This paper highlights the issues that may be involved with training downstream models on generated data from other networks.

Our graph model performed remarkably well, and outperformed the previous graph model when compared on the same data. This reinforces the importance for computational approaches in biomedical research to be grounded in biophysical and biochemical principles.

An evaluation that would be valuable would be to compare across all of our evaluations to see how the original project architecture and pipeline performed using only peer-reviewed experimental protein structures. Another way to further evaluate would be to explore datasets similar in size on both proposed methodologies. This would uncover the impact of training on more peer-reviewed protein structure data or the performance of less predicted protein structures from AlphaFold.

With the rapid growth in machine learning and artificial intelligence, this project serves as a reminder to stay grounded in fundamental principles in physics, chemistry, and biology, and let those inform the development of novel analytical techniques that will transform our biomedical knowledge.

References

Abid, Abubakar, Balin, Muhammad Fatih, and Zou, James. Concrete Autoencoders for Differentiable Feature Selection and Reconstruction, January 2019. URL http://arxiv.org/abs/1901.09346. arXiv:1901.09346 [cs].

Bryant, Patrick, Pozzati, Gabriele, Zhu, Wensi, Shenoy, Aditi, Kundrotas, Petras, and Elofsson, Arne. Predicting the structure of large protein complexes using AlphaFold and Monte Carlo tree search. *Nature Communications*, 13(1):6028, October 2022. ISSN 2041-1723. doi: 10.1038/s41467-022-33729-4. URL https://www.nature.com/articles/s41467-022-33729-4.

Jumper, John, Evans, Richard, Pritzel, Alexander, Green, Tim, Figurnov, Michael, Ronneberger, Olaf, Tunyasuvunakool, Kathryn, Bates, Russ, Žídek, Augustin, Potapenko, Anna, Bridgland, Alex, Meyer, Clemens, Kohl, Simon A. A., Ballard, Andrew J., Cowie, Andrew, Romera-Paredes, Bernardino, Nikolov, Stanislav, Jain, Rishub, Adler, Jonas, Back, Trevor, Petersen, Stig, Reiman, David, Clancy, Ellen, Zielinski, Michal, Steinegger, Martin, Pacholska, Michalina, Berghammer, Tamas, Bodenstein, Sebastian, Silver, David, Vinyals, Oriol, Senior, Andrew W., Kavukcuoglu, Koray, Kohli, Pushmeet, and Hassabis, Demis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, August 2021. ISSN 1476-4687. doi: 10.1038/s41586-021-03819-2. URL https://www.nature.com/articles/s41586-021-03819-2. Publisher: Nature Publishing Group.

Nguyen, Viet Thanh Duy and Hy, Truong Son. Multimodal pretraining for unsupervised protein representation learning. *Biology Methods and Protocols*, 9(1): bpae043, January 2024. ISSN 2396-8923. doi: 10.1093/biomethods/bpae043. URL https://doi.org/10.1093/biomethods/bpae043.

Shervashidze, Nino, Schweitzer, Pascal, Leeuwen, Erik Jan van, Mehlhorn, Kurt, and Borgwardt, Karsten M. Weisfeiler-Lehman Graph Kernels. *Journal of Machine Learning Research*, 12(77):2539–2561,

2011. URL http://jmlr.org/papers/v12/shervashidze11a.html.