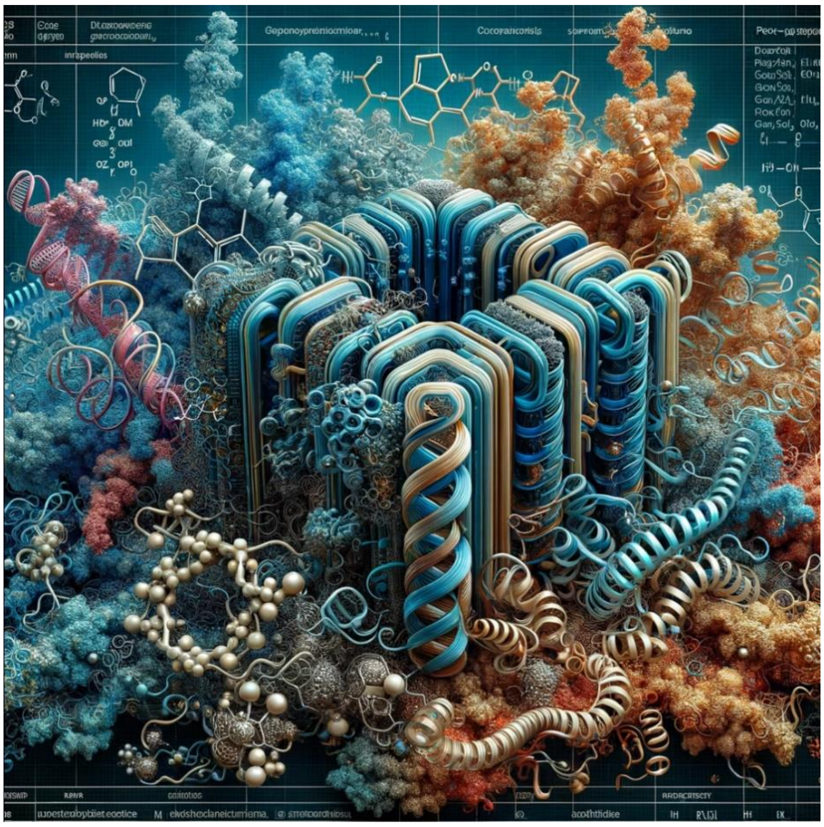
Representing protein structures with deep learning models based on geometricus



Joe Zhang
1206052
Supervised by:
Aalt-Jan van Dijk

1 Abstract

With millions of protein structures from large protein language models, leveraging protein structure information for protein property predictions and protein design requires an embedding method for structures. Self-supervised learning can be used to denoise the data and augement the information. To represent protein structure, geometricus is a novel method for protein structure representation using moment invariants, which can capture the main components of an 3D object irrelevant to rotation. In this project, we trained various autoencoder models for protein structure embedding and customized model structures to explore combining sequence information with protein structures. The protein embedding space was used for SCOP class prediction and protein function prediction. We explored the viability of models for protein space representation based on geometricus, providing a possibility to use self-supervised models to augement the structure information from geometricus for downstream tasks and combine sequence information into geometricus.

2 Introduction

Proteins are a group of versatile molecules; they perform a vast number of functions in cells. Protein functions are mostly determined by their structures. Hence, compared to protein sequences, protein structures provide us with more insight into protein function and interaction. To investigate protein structures, many experimental methods have been developed, such as mass spectrometry, X-ray crystallography, and nuclear magnetic resonance. With the development of deep learning methods, computationally predicting protein 3D structures is much less time-consuming and expensive than determining them experimentally (Wei, 2019). For example, Alphafold uses information from Multiple Sequence Alignment (MSA), residue-pair matrix, and structure templates to predict structures based on the input sequence (Jumper et al., 2021). Recently, Meta Fundamental AI Research Protein Team (FAIR) trained a group of transformer protein models among which ESM2 can achieve comparable accuracy as AlphaFold in cases when no MSA is available. They represent protein sequences in a higher dimensional space to make multi-task predictions, including protein functions, residue-residue contacts, and mutational effects. (Lin et al., 2023a). These deep-learning models provide accurate-predicted protein structures based on protein sequences, which can pave the way for experimental methods and enable high-throughput structure-based analysis (Varadi & Velankar, 2022). Only sequence-based models work fine, with structure information, the performance can be improved. The functions of proteins mostly depend on structure domains. With those structure data, training large deep-learning models to have better performance for predicting protein properties would be possible. However, to enable this, a way of representing those structures as input for deep learning is required.

Geometricus, a novel protein structure embedding method, splits protein 3D structures into shapemers, and represents them as vectors of 3D moment invariants (MIs) irrelevant to structure rotation, which can be the input of deep learning models (Durairaj et al., 2020). Count matrices can be generated by counting unique moment invariants. This method has been used to compare structural differences and functionally unknown protein annotations between structures in AlphaFold database (AFDB) and in Protein Data Bank (PDB) (Durairaj et al., n.d.). It provides a method to represent structural information as input for deep learning models. It can potentially improve the performance of downstream tasks.

Only a small fraction of proteins are annotated with functions. Self-supervised learning is widely used for information extraction when the labels for supervised learning tasks are sparse. Using self-supervised learning can extract information from unlabeled data and achieve better performance. Autoencoder and Variational Autoencoder are widely used to generate embeddings for downstream prediction or generative tasks (Kingma & Welling, 2013; Michelucci, 2022). Self-supervised methods are also commonly used for protein structure embeddings, some models are developed for some 3D structure tasks e.g. predicting residue distances and dihedral angles (Chen et al., 2022; Zhang et al., 2022). Vector Quantised-Variational AutoEncoder (VQ-VAE) was used for protein structure search model—Foldseek (van Kempen et al., 2023).The self-supervised learning methods have partially replaced hand-crafted features as the study focus (Villegas-Morcillo et al., 2021). In most cases, neuron networks can capture much more complex relationships of vast data, which makes them dominant.

Natural language processing is the study of making machines understand human languages. In most biological sequence analyses, these studies shared the same rationale with natural language processing, among which, the transformer is the zenith of the era (Vaswani et al., n.d.). Those large language models are based on the transformer architecture. Bidirectional Encoder Representations from Transformers (BERT) deep learning architectures have also been applied for protein embedding. ProteinBERT, a type of denoising autoencoder, takes protein sequences for local embedding, using cross-attention to combine sequence information with function annotations. It achieved comparable accuracy on diverse protein benchmarks yet has fewer parameters to train than other models (Brandes et al., 2022a). Einagger et al. trained multiple transformer-based protein Language models to embed millions of protein sequences, achieving competitive performance compared with state-of-the-art methods which are based on Multiple sequence alignment (MSA) for per-residue and protein-level classification tasks, which shows that using the information of the protein itself instead of comparison of MSA can work well yet efficient. The attention-based model was also applied for protein structure embedding; Contactlib-Att abstracts protein structures as hydrogen bond vectors and then uses an attention-based model to make SCOP superfamily classification (Chen et al., n.d.). Combining structure information with sequences is another possibility for a more powerful protein model.

To fuel the development of protein modeling, several benchmarks and datasets have been introduced. CAFA is a protein function prediction task that is continuously updated. The current CAFA5 competition aims to predict terms from the three different gene ontologies terms (molecular function, biological process, cellular component) based on different popular protein databases. We will use the experimentally determined protein database from CAFA5 for training and testing our models.

Based on all the information, we aim to develop embedding models for protein structure representation and finetune the model for downstream tasks. Finetune means the existing model with an extra layer would be trained for other prediction tasks.

3 Method

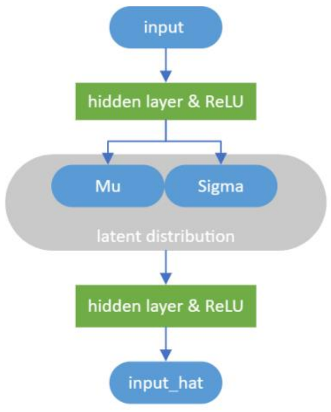
3.1 Variational Autoencoder (VAE)

3.1.1 Count Matrix Datasets

In the previous master thesis project from Ricardo, 30,307,021 proteins were retrieved from Pfam (Mistry et al., 2021). Three datasets with 10000, 100000, and 1000000 proteins were generated from random subsets of the Pfam protein dataset. In these three datasets, protein structures are represented as shapemers; each shapemer is a sub-structure represented by characteristic numbers invariant to structure rotation (Moment Invariants); then, by counting each unique shapemer, count matrices are generated for each protein. Train, validation, and test sets have been split already with percentages: 94%, 5%, and 1%.

3.1.2 Model Structure

VAE is used to condense the count matrix for protein structure embedding. These models were trained on count matrix datasets. Model input embedding sizes, the numbers of unique shapemers (count matrix length), are 2539, 3849, and 5451 for the 10000, 100000, and 1000000 datasets, respectively. A single hidden layer was



applied in the encoder and decoder, followed by a ReLU function. The hidden size was set to 50, and the latent variable (z) was set with a dimension of 3 (Fig 1).

Fig 1. VAE model structure

3.1.3 Embedding Training

For training and validation steps, KL divergence and reconstruction loss were combined:

$$L = 0.2 * KL\big(q(z|x)||p(z|x)\big) + 0.8 * (\hat{y} - y)^2$$

$$q(z|x)\ is\ the\ approximate\ distribution\ generated\ by\ VAE\ model$$
$$p(z|x)\ is\ the\ true\ posterior\ distribution$$

KL divergence is used to force the model to approximate the hypothetical true posterior distribution. To compare with the Denoised autoencoder built in the previous project, only reconstruction loss was considered in the test set. For the embedding plot, KL divergence was also put into the loss function for regularization. To push the model more focus on regenerating inputs instead of approximating the normal distribution, the weight for reconstruction is set to 0.8 and the KL divergence is set to 0.2. The batch size was set to 64, and the Adam optimizer was used for training with a learning rate of 10^{-3}.

3.2 Transformer

3.2.1 Moment Invariant Datasets

Two datasets were retrieved for the models.

The first dataset from the CAFA5 train dataset was used for pretraining and one of the downstream tasks: GO prediction. After filtering proteins without structure information from the alphafold database, 132491 proteins were retained. The length of the corresponding sequences ranged from 16 to 2699 amino acids. During pretraining, only protein structural information of moment invariants and sequences was visible to models. For the finetuning part, the predictions were only based on the Molecular function (MF) part. The MF terms are selected which have more than 100 annotated proteins (appendix).

The second dataset is 24908 proteins from the Scope database, with four different structural classes, i.e., 4319 proteins of all-alpha (all alpha-helix), 5495 proteins of all-beta (all beta-sheet), 8212 proteins of alpha+beta (alpha helix and beta sheet occur separately), and 6882 proteins of alpha/beta (alpha helix and beta sheet alternates). This dataset provides input for the task of protein structural classification and latent space plot.

3.2.2 Sequence and Structure Encoding

Sequences were first represented by 28 unique tokens. In addition to the standard 20 amino acids, this included the amino acids, Selenocysteine (U) and Pyrrolysine (O); four ambiguous amino acids unknown (X), Asparagine or aspartate (B), Glutamine or glutamate (Z), and Leucine or isoleucine (J); and two extra training tokens, padding token and mask token. The padding token was used to pad the batches of protein sequences into the same length. Mask token was used to mask several tokens of sequences to make the model to predict. Structure embedding was implemented using geometricus with default settings: kmer of length 8, resolution 1, and 4 types of MIs. The scales of numbers of MIs are from 1 to 17, plus an extra padding token.

3.2.3 Model Structure

Two models were applied based on Transformer architecture. After representing sequences using tokens, an embedding layer was used to represent each amino acid as a vector of 16. The first one (gBERT) only takes MIs as input, the multi-head self-attention layer of 4 16-length heads was used for analyzing the interaction between each substructure. The transformer block was repeated three times. For the second model (sgBERT), a convolution layer was used for extracting information from protein sequences with kernel size 8, padding 3. The 8-mers generated would correspond to the shapemer from the same position in proteins. Multi-head cross-attention was used for combining sequence and structure information with the same number of heads and head length as gBERT, where sequences were inputted as Query, MIs were inputted as Key and Value. The computational result of Query and Key would be the weight for each shapemers, so, making sequences as key and MIs as Key would potentially extract the interaction between sequences and structures. Then moment invariants go through fully connected layers (Fig2).

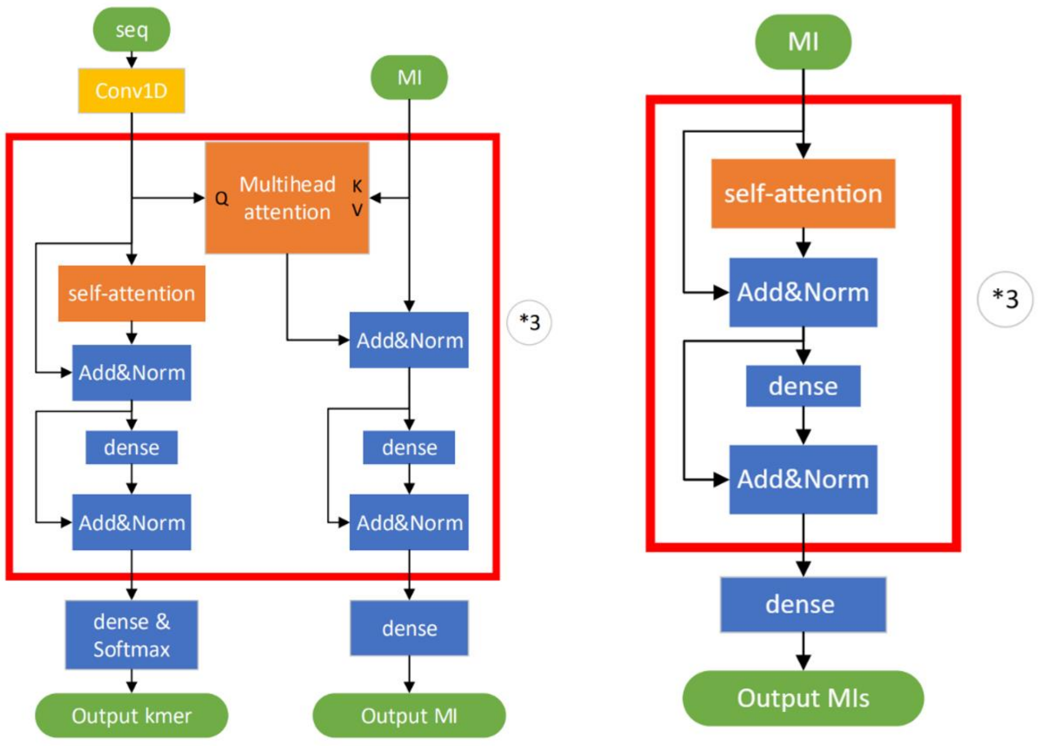


Fig 2 sgBERT (left) and gBERT (right) model structure. The transformer block was repeated three times. Multihead attention: attention with two different inputs. Self-attention:

attention with only one input. Dense: fully connected layer. Add&Norm: ideas from ResNet and layer normalization.

3.2,4 Pretraining

CAFA5 dataset was used for the pretraining task (132491 proteins, and 490 MF function terms); it is split into train and validation sets (80%, 20%). Xavier initialization was implemented for all BERT-like models.

In the sgBERT pretraining, 5% amino acids of sequences and 5% shapemers of structures were randomly masked with mask tokens 27 and 18, respectively. Each shapemer contains four MIs. When one shapemer was chosen to be masked, four MIs were all masked. The loss function was customized:

$$L = \frac{1}{2}\sum_{c=1}^{26} log\frac{e^{x_c}}{\sum_{i=1}^{26} e^{X_i}}y_c + \frac{1}{2}\sum_{mi=1}^{4}(\hat{y}_{mi} - y_{mi})^2$$

$$where\ c\ is\ the\ number\ of\ amino\ acid\ tokens\ (26\ in\ total),$$
$$mi\ is\ the\ number\ of\ different\ moment\ invariant\ type\ (4\ in\ total)$$

The cross-entropy loss for sequences (red part) and MSEloss for MIs (blue part) were added together. They are set with the same weight since both are equally important. In cross-entropy loss, x is the amino acid token probability value generated by the model, $\frac{e^{x_c}}{\sum_{i=1}^{26} e^{X_i}}$ is the SoftMax function, averaged along 26 tokens' probability. y is the label to predict. $\hat{y}_{mi}$ is the MI value the mode generated. They were set with the same weight (1/2). The batch size was set to 16. 16-mixed precision was used for acceleration. Early stopping with patience 3 was applied to reduce overfitting. The range of epochs was set to (2,10000). In the beginning, the learning rate was set to $2 * 10^{-5}$ training for 19 epochs, Unexpectedly, training loss was abruptly turned into NaN in the 20th epoch. To deal with the problem, the learning rate was adjusted to $1*10^{-5}$ and continued training for 120 epochs.

In the gBERT pretraining, only MSE loss function was used. The batch size was set to 4. Other configurations were the same as above, except the learning rate is 1*10-5 from the beginning. The model was trained for 20 epochs.

3.2.5 Downstream Task

The embedding output will be averaged along the length dimension to make sure every protein has the same embedding size. An extra linear layer and SoftMax function were added to make four SCOP class predictions. The model with a frozen pretraining part and the model with an unfrozen pretraining part were compared. Frozen means the parameters will not be updated during training, whereas unfrozen means the parameters will be. The learning rate was set to $1 * 10^{-3}$. The batch size was set to 4. Early stop was implemented with a patience of 3, monitored based on validation loss. Different settings of training were implemented:

1) Frozen gBERT pretrainied model with an extra fully connected layer. 2) Unfrozen gBERT pretrained model with an extra fully connected layer (the pretaining model parameters would be adjusted during training). 3) Frozen sgBERT

pertrained model outputs MIs embeddings to an extra fully connected layer. 4) Unfrozen sgBERT pertrained model outputs MIs embeddings to an extra fully connected layer. 5) Frozen sgBERT pretrained model outputs MIs and sequences. They are concatenated and put into an extra fully connected layer. 6) Unfrozen sgBERT pretrained model outputs MIs and sequences. They are concatenated and put into an extra fully connected layer.

For protein function prediction, an extra layer was added to make MF function predictions. Frozen and unfrozen settings are compared. The learning rate was set to $1*10^{-4}$, and the batch size was set to 2. Early stop was implemented with patience of 3, monitored based on validation loss.

3.2 Principle Component Analysis (PCA)

PCA is a method for dimension reduction. It is used for embedding space plots since all the embedding spaces of models are larger than two.

4 Results

Overall, we want to investigate the embedding space of models and benchmark the performance of models on different tasks. VAE models reconstruction loss and embedding spaces, and prediction performances of transformer models are shown below. We also investigate what happened in the attention heads to know which part of the protein the model focuses on.

4.1 VAE

VAE achieved lower reconstruction loss than the denoising autoencoder model (DAE_shallow) built in previous project. The results of the denoising autoencoder were copied from the log files from the previous project (Table 1). The loss values indicate improvements of reconstruction, but the embedding plots shows that in the latent spaces, there are no clear patterns (appendix). Further investigation shows that posterior collapse happened in the latent spaces, which means latent spaces are just normal distributions without any feature from inputs (appendix).

Validation loss	10000	100000	1000000
DAE_shallow	1.809	0.823	0.296
VAE	0.205	0.135	0.105

Table 1. MSE validation reconstruction loss comparison between the previous model and the VAE model. Both models have the same number of layers. VAE models only consider reconstruction loss. Lower loss means the model can reconstruct input value with higher accuracy. 10000, 100000, and 1000000 are three datasets with corresponding numbers of proteins in previous project.

4.2 Transformer

During pretraining, the masked predictions were plotted.

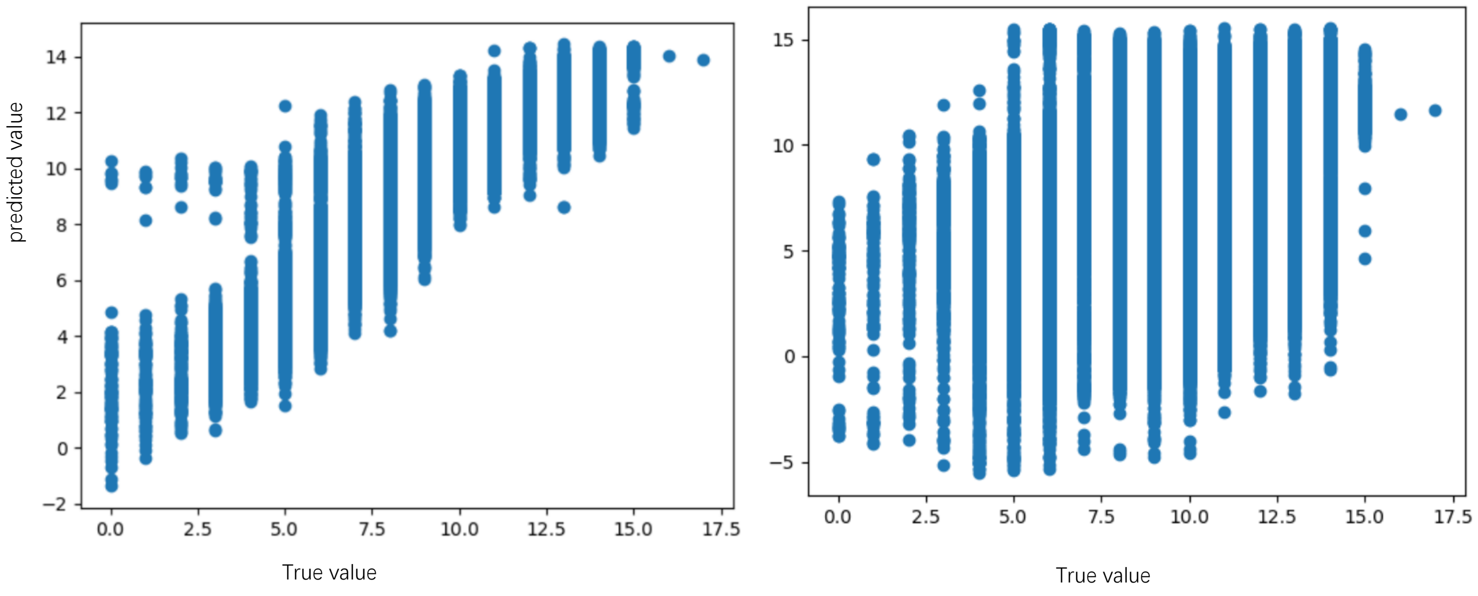


Fig 4. Pretraining plot for gBERT (left) and sgBERT (right) based on CAFA validation set, X-axis is the true value of MIs, Y-axis is the predicted MI values.

Fig 4 shows that gBERT, which only takes MIs as input, has better performance on masked MI prediction. The sgBERT takes sequences and structures as inputs; the performance on single-masked MI prediction is worse. More input types require more parameters and larger model size. The sgBERT only has embedding size 4, which limits the ability of the model to reconstruct both sequences and MIs. Combining sequence with structure is for general protein embeddings. The embeddings can contain both protein sequence and structure information.

The embedding result of SCOP database was plotted with four classes as labels.

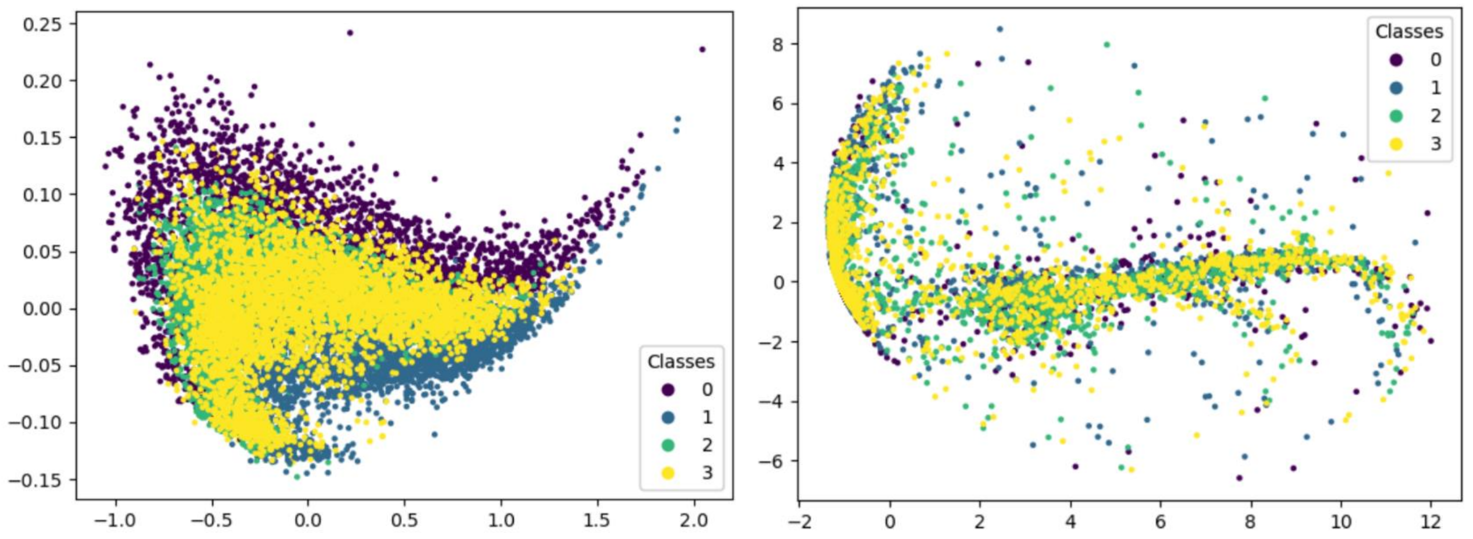


Fig 5. PCA plot for SCOP dataset embedding using gBERT (left) and sgBERT (right). 0: all-alpha. 1: all-beta. 2: alpha+beta. 3: alpha/beta

Based on Fig 5, without sequence information, protein structures embedded by gBERT have a clearer separation. But alpha+beta and alpha/beta still seem more mixed together than all-alpha (0) and all-beta (1) classes, which may indicate that the model is not position-aware: when two proteins contain the same elements of secondary structures but have different orders, the model may not separate them in latent space. When sequences and MIs are combined in sgBERT, proteins in latent space are not well-separated based on structure classes. In this case, combining sequence information into structure embeddings is not helpful. Fig 4 also shows that MI reconstruction of sgBERT is not good as gBERT.

Since in Fig5, the four classes are not clearly separated, an extra fully connected layer was added for the classification task. We tried different settings of the models:

	gBERT	**sgBERT** MI	**sgBERT** MI+seq	**geometricus**
Embedding Frozen	74.08%	76.81%	**78.18%**	65.99%
Embedding Unfrozen	75.58%	76.26%	75.62%	

Table 2. Scope class prediction with different embedding configurations.

In Table 2, The result showed that different settings achieve comparable accuracy. The best one is using frozen pretrained sgBERT model, the sequence embeddings and MI embeddings are concatenated as input for the last classification layer. The model with frozen pretrain part outperforms the unfrozen one except gBERT. The validation loss for the frozen model is lower than the unfrozen model. So there is no overfitting happened. In this less complex task, simple embeddings without finetuning can make better predictions. Here, we conclude that even though in latent space, combining sequences information into structures embeddings cannot make better separation of structure classes but it does improve the class prediction performance. Overall, the model performances are better than geometricus.

The attention part of the model is handy to investigate which part of the sequence contributes most to the prediction. To check whether the attention head can detect the function domain information related to the structure class, proteins with successful predictions were randomly chosen to investigate the weights of attention in sgBERT (appendix). This part is only exploring results.

The attention weight plots indicated that the whole pattern of attentions is changed. After finetuning, several spots in attention are highlighted. But there is no biological domain highlighted, for example, the protein domain of 3MAY is arranged from position 4 to 81 on InterPro (Paysan-Lafosse et al., 2023)(appendix). No clear corresponding patterns found in the attention head plot. To further investigate it, other proteins' attention plots were made (appendix).

We investigate attention changes for more proteins. These pairs of heatmaps of different proteins show that there is no clear pattern for the changing of attention weights (appendix). Generally, after finetuning, several positions of attention heads are highlighted with more weight. Compared with pretrained plots, attention seems condensed on fewer positions (appendix).

To check the model performance on tasks with more labels, CAFA benchmark is used. gBERT and sgBERT are finetuned with an extra fully connected layer to predict 490 MFs. The data of function term for each protein is very sparse. Even the protein with the most annotated function terms only has 61 terms among 490. In this case, the accuracy is not representative; most of the models can achieve 98% accuracy by predicting 0 for most of the function terms. F-score, precision, and recall were used for further comparison.

	Precision	**recall**	**F-score**
geometricus	0.267	0.0817	0.125
gBERT	0.256	0.0968	0.140
gBERT frozen	0.259	0.091	0.134
sgBERT(MIs)	**0.271**	0.0812	0.125
sgBERT(MIs) frozen	0.267	0.105	0.150
sgBERT(MIs+seqs)	0.270	**0.132**	**0.177**
sgBERT(MIs+seqs) frozen	**0.271**	0.131	0.177

Table 3. MF prediction evaluation results. For unfrozen model, during training for prediction, all parameters are backpropagated. sgBER(MIs) means the model only use MIs for MF prediction. sgBERT(MIs+seqs) means MIs and sequences are concatenated for prediction, which shows the best performance.

In table3, sgBERTs generally have higher precision, which means there are less false positive generated comparing with gBERT. Recall is generally less than precision because the imbalance of data. The most abundant MF function is annotated in 57380 proteins whereas the least MF function is only annotated in 100 proteins. After embeddings of sgBERT, the F score is slightly better than basic geometricus. There is no significant difference shown between frozen and unfrozen models based on three evaluation scores (table 3).

5 Discussion

VAE model:

VAE introduced variational Bayesian. Since the latent space is continuous, allowing easy random sampling and make the model more generative (Cinelli et al., 2021). The underlying idea for VAE is to approximate the probability of latent space z given input x (P(z|x)). One thing that needs to be considered carefully is posterior collapse (Wang et al., n.d.). The posterior probability P(z|x) may nearly become the same as P(z), which means the latent space may almost forget all information of input data. P(z) is the probability of z without input given as the condition. The model would only depend on the decoder to regenerate the inputs. In VAE model on the 10000 protein dataset, means (μs) of latent distribution for different x is differentiable in the latent space, which indicates that there is no posterior collapse. However, for VAE models on 100000 protein and 1000000 protein datasets, μs in latent space are all the same, which indicates posterior collapse. As the count matrix inputs become larger, the models can hardly distinguish differences among inputs. VQ-VAE was introduced to overcome the problem (Oord et al., 2017) and could potentially be applied on our data as well.

NaN loss:

Initially, masking was implemented by using the minimum negative value ($-1 * 10^{-9}$ for float32 or $-1 * 10^{-4}$ for float16). This potentially causes NaN loss during pretraining since the mask was implemented on some randomly selected shapemers and amino acids. During the feedforward pass, whole rows or columns are masked with minimum negative values. Assumingly, the reason would be that in the SoftMax function of multi-head attention, an extremely negative value would cause zero output, which may cause intractable gradients during backpropagation since the mask is for all MIs of a shapemer, which means a whole row or column of matrix in attention are masked (supplementary materials). To avoid the problem, special tokens were used for masking. These tokens were not in the vocabulary of sequences and MIs. In BERT's original paper, they proposed that the extra token introduced in pretraining may cause misleading in the finetuning part since the mask token does not exist in finetuning (Devlin et al., n.d.). To avoid this problem, they also replace the mask position with random tokens 10% of the time. However, our models don't implement the same method due to the limitation of time. Another potential solution is introducing temperature (T) in the SoftMax function (Agarwala et al., 2020). As a hyperparameter, T needs to be optimized. But its viability still needs to be tested.

BERT-like architecture:

Our two BERT-like model embeddings seem not sufficient for the complex classification problem, GO prediction. Those state-of-the-art models can achieve more than 60% F-score. After all, the embedding size of our model is only 4. Besides the small embedding size, another obstacle is the imbalanced dataset. The models even make no function prediction for some proteins. Considering original BERT, its sequence-length agnostic architecture makes it perform well for question answering. For classification tasks, two different methods were introduced for classification. The first one is introducing a classification (CLS) token for embedding, Chang et al. proposed a multi-CLS embedding proved to perform better than classic BERT CLS method in NLP task (Chang et al., 2022). The second one is taking the mean alone the various-length dimension. When the classification task becomes complex as protein function prediction, more CLS tokens or a larger embedding size need to be used. Considering protein function information is based on domains or vital residues, taking the mean for the downstream task does not seem representative. It is not suggested to use the embedding directly for complex tasks unless the model size is large enough, e.g. ESM models can achieve high performance for function prediction. whereas the embeddings can be used as a part of the inputs for an ensemble model instead of as a single input.

Positional encoding:

For the proteinBERT model, it was indicated that convolutional layers provide the model with position location information (Brandes et al., 2022b). In our model, the pretraining embedding plot of gBERT suggested that the model cannot distinguish substructure positions, even though we used convolutional layers for the sequence. Introducing positional encoding can solve it (Vaswani et al., n.d.).

Sequence length agnostic:

proteinBERT periodically changes the sequence length and introduces START, END tokens to make the model cope with various length proteins. However, when the sequence length is longer than the chosen sequence length, the model would randomly select a subsequence. Our BERT-like model does not restrict the sequence length to a certain range. Instead, batches of protein sequences and structures would be padded into the same length, which is agnostic to sequence length. But the maximum protein length is 2699 during training, the model performance on those proteins larger than this size remains questionable.

Function prediction:

The imbalance of function data is a problem. In DeepFRI model, they customized the loss function by adding weight for each function term. The weight is the fraction of the average number of annotations for a function term divided by the number of annotations for the specific function term (Gligorijević et al., 2021). So, if the function term has less protein annotated, the corresponding weight would be higher. Here we did not implement any weight modification, which should be improved.

6 Future work

The sgBERT and gBERT both are trained on experimentally determined protein sequences and structures. With more in-silico-generated protein structures and larger model sizes, the models should be generalized with these proteins. Also, there are now 17 different moment invariant types available in geometricus, which could increase the structure embeddings size. In addition, MIs representation can distinguish unique substructures. However, it cannot identify more detailed information related to protein function, e.g., discrete residues. Recently, Graph neuron networks (GNN) have been widely used for protein-related tasks, which can better capture interactions between protein atoms. There were many studies implementing GNN for protein representation. DeepFRI represented amino acid interactions as graphs, combining sequence embedding from an LSTM model to make protein function predictions. This model was trained on experimentally validated protein structures or homology-based structures (Gligorijević et al., 2021). Zuobai Zhang et al. implemented Graph representation for proteins with the same idea of k-mer and radius similar as geometricus. By contrastive learning, the model performs well on multiple protein function-related tasks and shows the highest sensitivity for protein search (Zhang et al., 2022). All those already showed the power of GNNs. The model performance with geometricus inputs should be compared with graph neural network approach later. Generally, MIs are easier to generate comparing with Graph representation.

Recently, the emergence of large language models (LLMs) made it possible that multiple tasks can be automated, and the performance is unparalleled (Lin et al., 2023b; OpenAI et al., 2023; Touvron et al., 2023). It shows that the huge size of the model could make a significant better performance on various tasks. These models laid the foundation for multiple tasks by providing embeddings. In protein-related embeddings, a larger model would also generate more meaningful representations. If we want to generate embeddings with much more information, a large model size is inevitably needed. Large computational resources are required, especially for resource-hungry models like transformers. To better utilize computational resources, some optimization methods can be introduced, e.g. Megatron-LM (Shoeybi et al., 2019). In addition, there are some new model structures aiming to compensate for resource-hungry deficiency of transformers, e.g. RetNet, RWKV (Peng et al., 2023; Sun et al., 2023). Though these structures are not widely implemented, the performance of the model still needs to be validated. Potentially, they can be used for resource-efficient sequence-like protein embedding.

Another important trend is leveraging the power of large protein language models for sequence embedding and then combining these with structure information. This approach is quite efficient for downstream supervised tasks. GAT-GO uses graph attention networks to embed inter-residue contact graphs, ESM-1b embedding was used for node-level features (Boadu et al., 2023; Lai & Xu, 2022). Specifically for protein function prediction, HEAL, innovating a hierarchical graph transformer structure (HGT), combines multiple sequence embeddings and structure distance maps. It introduced super nodes generated by graph convolutional networks for the HGT module (Gu et al., 2023).

7 Conclusion

Here we develop gBERT model to augement geometricus structure representations. sgBERT can combine sequence information with geometricus structure representations. Combining multiple inputs can improve the performance of deep learning model.

Overall, the performance of the models on small tasks is significantly better than the original geometricus. But it is still weak for complex tasks as function prediction. The next step would be to increase the hidden size of the models and leverage existing large protein language models for sequence information to make better embeddings.

8 Reference

Agarwala, A., Pennington, J., Dauphin, Y., & Schoenholz, S. (2020). *Temperature check: theory and practice for training models with softmax-cross-entropy losses*. http://arxiv.org/abs/2010.07344

Boadu, F., Cao, H., & Cheng, J. (2023). Combining protein sequences and structures with transformers and equivariant graph neural networks to predict protein function. *Bioinformatics*, *39*, I318–I325. https://doi.org/10.1093/bioinformatics/btad208

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2022a). ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, *38*(8), 2102–2110. https://doi.org/10.1093/bioinformatics/btac020

Brandes, N., Ofer, D., Peleg, Y., Rappoport, N., & Linial, M. (2022b). ProteinBERT: a universal deep-learning model of protein sequence and function. *Bioinformatics*, *38*(8), 2102–2110. https://doi.org/10.1093/bioinformatics/btac020

Chang, H.-S., Sun, R.-Y., Ricci, K., & McCallum, A. (2022). *Multi-CLS BERT: An Efficient Alternative to Traditional Ensembling*. http://arxiv.org/abs/2210.05043

Chen, C., Zha, Y., Zhu, D., Ning, K., & Cui, X. (n.d.). *Hydrogen bonds meet self-attention: all you need for general-purpose protein structure embedding*. https://doi.org/10.1101/2021.01.31.428935

Chen, C., Zhou, J., Wang, F., Liu, X., & Dou, D. (2022). *Structure-aware Protein Self-supervised Learning*. http://arxiv.org/abs/2204.04213

Cinelli, L. P., Marins, M. A., da Silva, E. A. B., & Netto, S. L. (2021). Variational methods for machine learning with applications to deep networks. In *Variational Methods for Machine Learning with Applications to Deep Networks*. Springer International Publishing. https://doi.org/10.1007/978-3-030-70679-1

Devlin, J., Chang, M.-W., Lee, K., Google, K. T., & Language, A. I. (n.d.). *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. https://github.com/tensorflow/tensor2tensor

Durairaj, J., Akdel, M., de Ridder, D., & van Dijk, A. D. J. (2020). Geometricus represents protein structures as shape-mers derived from moment invariants. *Bioinformatics*, *36*, I718–I725. https://doi.org/10.1093/bioinformatics/btaa839

Durairaj, J., Pereira, J., Akdel, M., New, V., Ai, Y. M., & Schwede, T. (n.d.). *What is hidden in the darkness? Characterization of AlphaFold structural space*. https://doi.org/10.1101/2022.10.11.511548

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

Gu, Z., Luo, X., Chen, J., Deng, M., & Lai, L. (2023). Hierarchical graph transformer with contrastive learning for protein function prediction. *Bioinformatics*, *39*(7). https://doi.org/10.1093/bioinformatics/btad410

Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A. J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., … Hassabis, D. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2

Kingma, D. P., & Welling, M. (2013). *Auto-Encoding Variational Bayes*. http://arxiv.org/abs/1312.6114

Lai, B., & Xu, J. (2022). Accurate protein function prediction via graph attention networks with predicted structure information. *Briefings in Bioinformatics*, *23*(1). https://doi.org/10.1093/bib/bbab502

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023a). Evolutionary-scale prediction of atomic-level protein structure with a language model. In *Science* (Vol. 379). https://www.science.org

Lin, Z., Akin, H., Rao, R., Hie, B., Zhu, Z., Lu, W., Smetanin, N., Verkuil, R., Kabeli, O., Shmueli, Y., dos Santos Costa, A., Fazel-Zarandi, M., Sercu, T., Candido, S., & Rives, A. (2023b). Evolutionary-scale prediction of atomic-level protein structure with a language model. In *Science* (Vol. 379). https://www.science.org

Michelucci, U. (2022). *An Introduction to Autoencoders*. http://arxiv.org/abs/2201.03898

Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G. A., Sonnhammer, E. L. L., Tosatto, S. C. E., Paladin, L., Raj, S., Richardson, L. J., Finn, R. D., & Bateman, A. (2021). Pfam: The protein families database in 2021. *Nucleic Acids Research*, *49*(D1), D412–D419. https://doi.org/10.1093/nar/gkaa913

Oord, A. van den, Vinyals, O., & Kavukcuoglu, K. (2017). *Neural Discrete Representation Learning*. http://arxiv.org/abs/1711.00937

OpenAI, :, Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., Almeida, D., Altenschmidt, J., Altman, S., Anadkat, S., Avila, R., Babuschkin, I., Balaji, S., Balcom, V., Baltescu, P., Bao, H., Bavarian, M., … Zoph, B. (2023). *GPT-4 Technical Report*. http://arxiv.org/abs/2303.08774

Paysan-Lafosse, T., Blum, M., Chuguransky, S., Grego, T., Pinto, B. L., Salazar, G. A., Bileschi, M. L., Bork, P., Bridge, A., Colwell, L., Gough, J., Haft, D. H., Letunić, I., Marchler-Bauer, A., Mi, H., Natale, D. A., Orengo, C. A., Pandurangan, A. P., Rivoire, C., … Bateman, A. (2023). InterPro in 2022. *Nucleic Acids Research*, *51*(D1), D418–D427. https://doi.org/10.1093/nar/gkac993

Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Biderman, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K. K., He, X., Hou, H., Lin, J., Kazienko, P., Kocon, J., Kong, J., Koptyra, B., Lau, H., … Zhu, R.-J. (2023). *RWKV: Reinventing RNNs for the Transformer Era*. http://arxiv.org/abs/2305.13048

Shoeybi, M., Patwary, M., Puri, R., LeGresley, P., Casper, J., & Catanzaro, B. (2019). *Megatron-LM: Training Multi-Billion Parameter Language Models Using Model Parallelism*. http://arxiv.org/abs/1909.08053

Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., & Wei, F. (2023). *Retentive Network: A Successor to Transformer for Large Language Models "*. https://aka.ms/GeneralAI

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., Rodriguez, A., Joulin, A., Grave, E., & Lample, G. (2023). *LLaMA: Open and Efficient Foundation Language Models*. http://arxiv.org/abs/2302.13971

van Kempen, M., Kim, S. S., Tumescheit, C., Mirdita, M., Lee, J., Gilchrist, C. L. M., Söding, J., & Steinegger, M. (2023). Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*. https://doi.org/10.1038/s41587-023-01773-0

Varadi, M., & Velankar, S. (2022). The impact of AlphaFold Protein Structure Database on the fields of life sciences. In *Proteomics*. John Wiley and Sons Inc. https://doi.org/10.1002/pmic.202200128

Vaswani, A., Brain, G., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (n.d.). *Attention Is All You Need*.

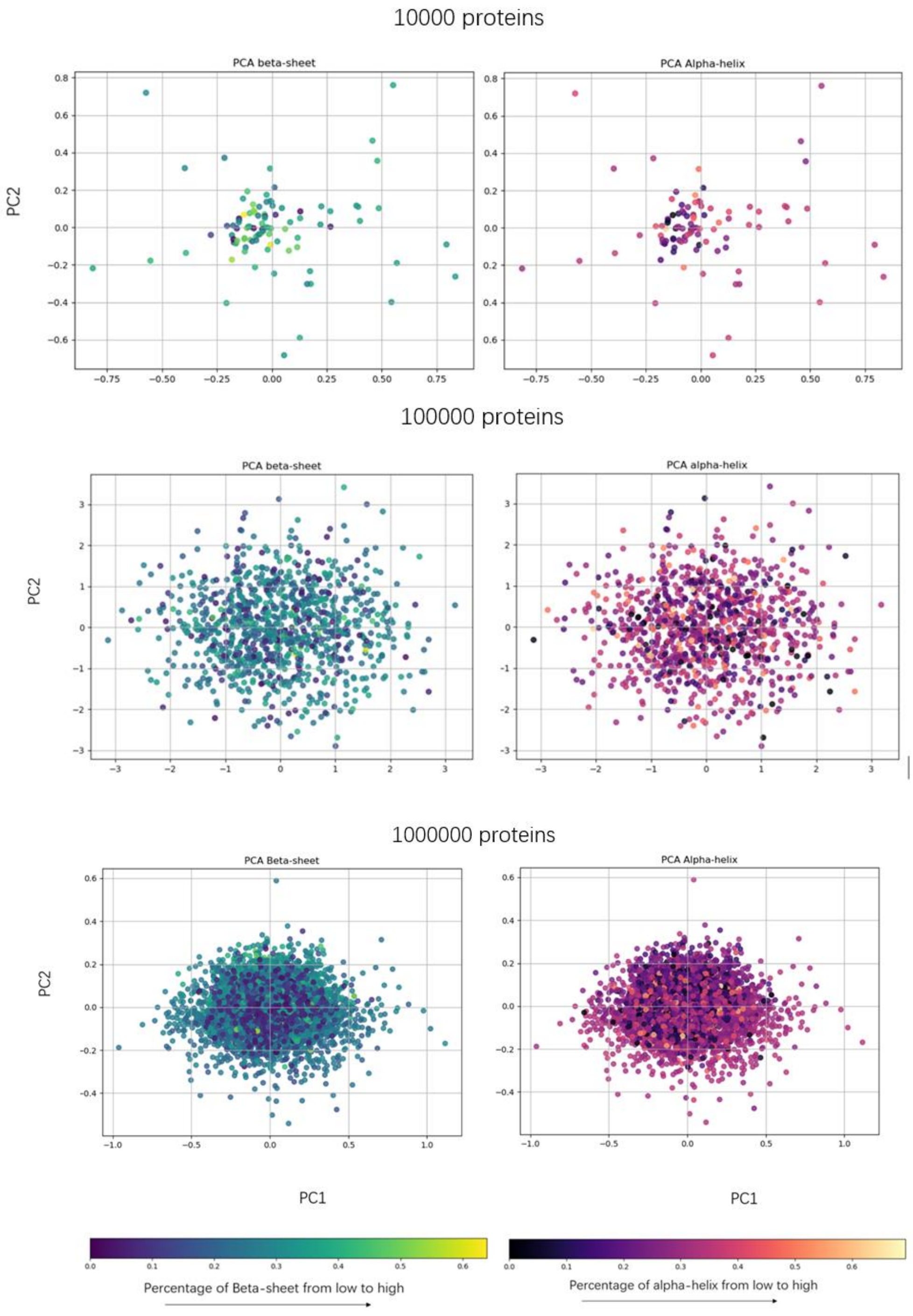
Villegas-Morcillo, A., Makrodimitris, S., Van Ham, R. C. H. J., Gomez, A. M., Sanchez, V., & Reinders, M. J. T. (2021). Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, *37*(2), 162–170. https://doi.org/10.1093/bioinformatics/btaa701

Wang, Y., Blei, D. M., & Cunningham, J. P. (n.d.). *Posterior Collapse and Latent Variable Non-identifiability*.

Wei, G.-W. (2019). Protein structure prediction beyond AlphaFold. *Nature Machine Intelligence*, *1*(8), 336–337. https://doi.org/10.1038/s42256-019-0086-4

Zhang, Z., Xu, M., Jamasb, A., Chenthamarakshan, V., Lozano, A., Das, P., & Tang, J. (2022). *Protein Representation Learning by Geometric Structure Pretraining*. http://arxiv.org/abs/2203.06125

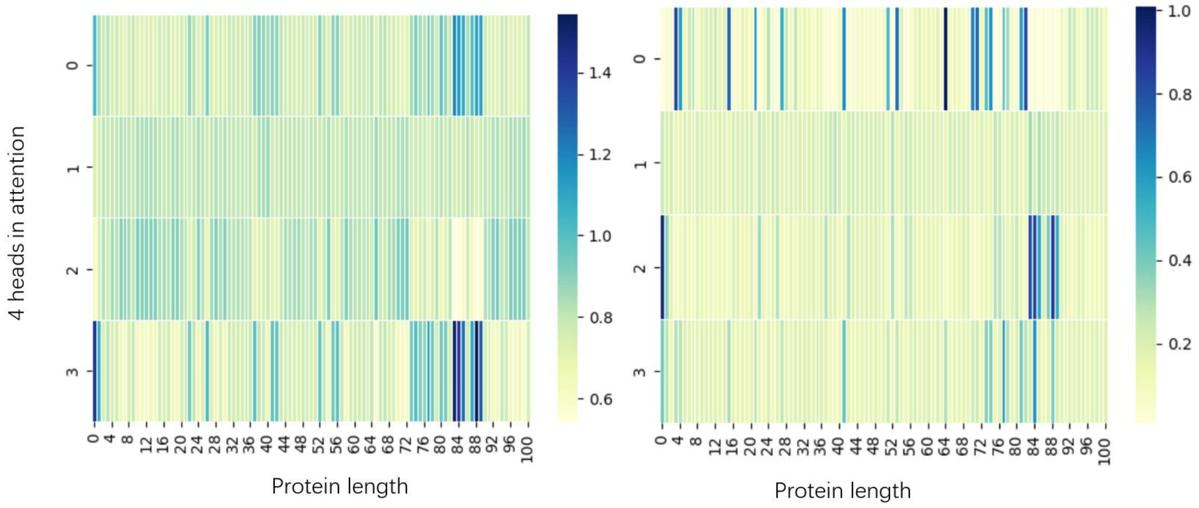
8 Appendix



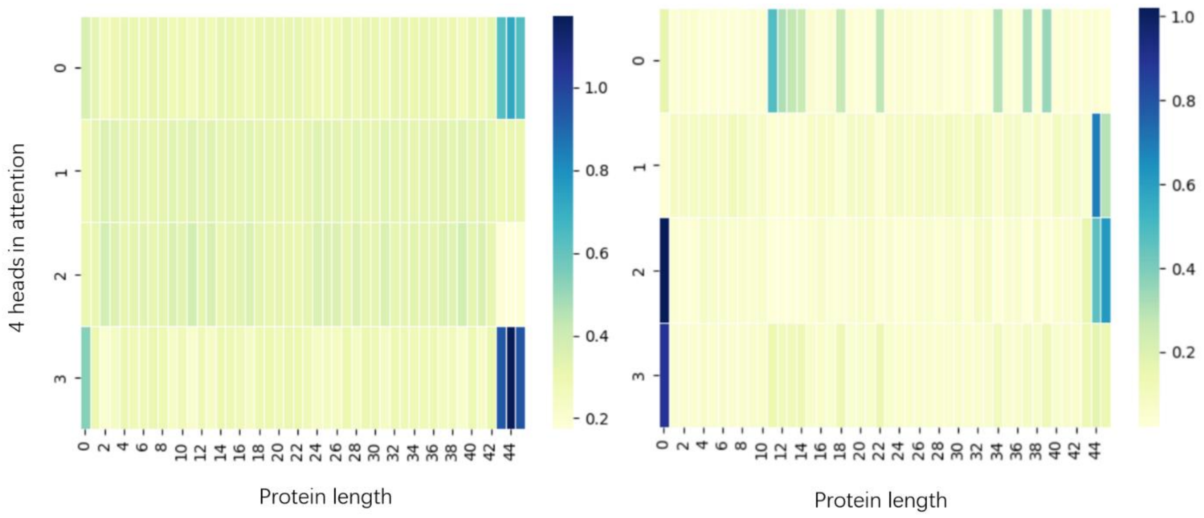
Latent spaces were plotted using PCA for the VAE model based on the 10000-protein,100000-protein, and 1000000-protein dataset. each dataset was plotted with different labels, percentage of Beta-sheet for the left plots and percentage of Alpha-helix on the right. There is no clear pattern in the graphes.

Latent space	μ_1	μ_2	μ_3
100000	$-3.1101 * 10^{-5}$	$-8.0154 * 10^{-5}$	$-5.9693 * 10^{-5}$
1000000	$-1.9668 * 10^{-5}$	$-3.5610 * 10^{-5}$	$1.0340 * 10^{-5}$

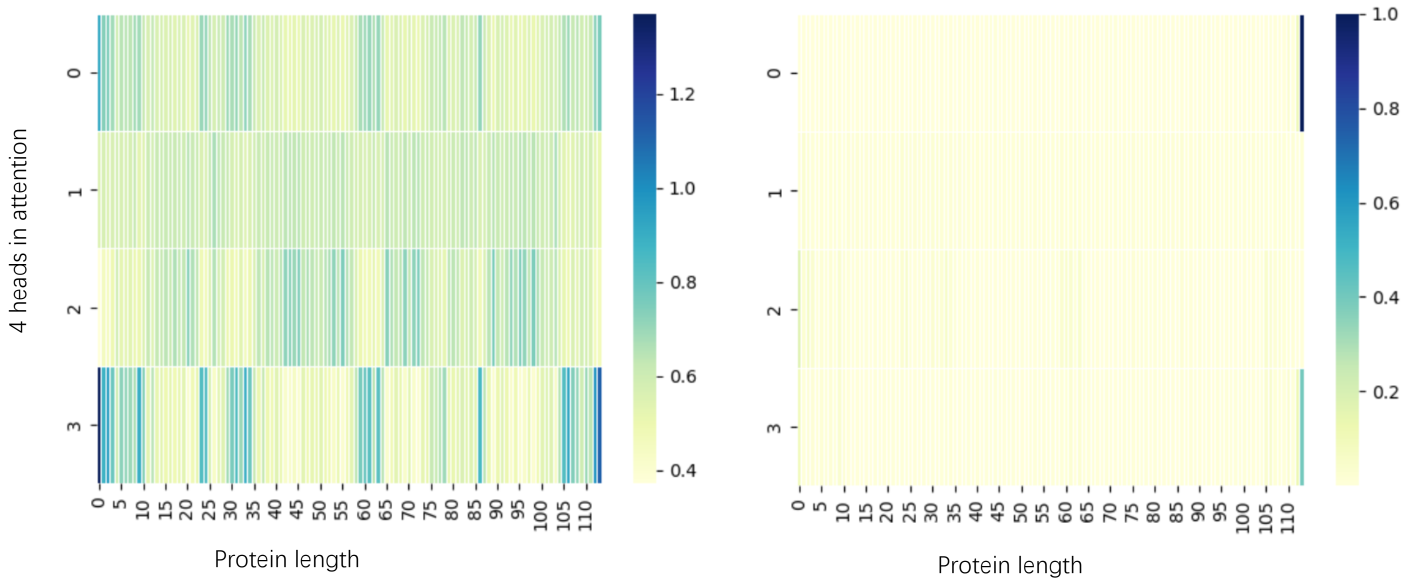
The same μs in VAE models on 100000 and 1000000 datasets indicate that the posterior collapse happened.



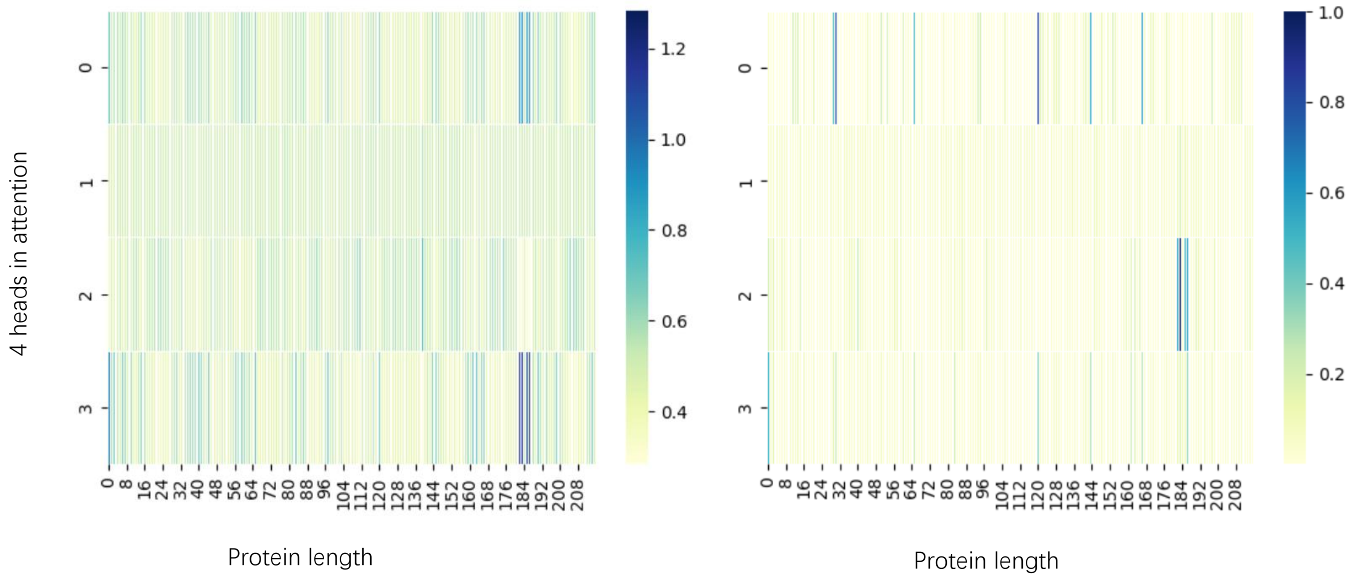
Attention weights after normalization for protein 3MAY. Left one is the plot for pretrained sgBERT, right one is the plot for finetuned sgBERT.



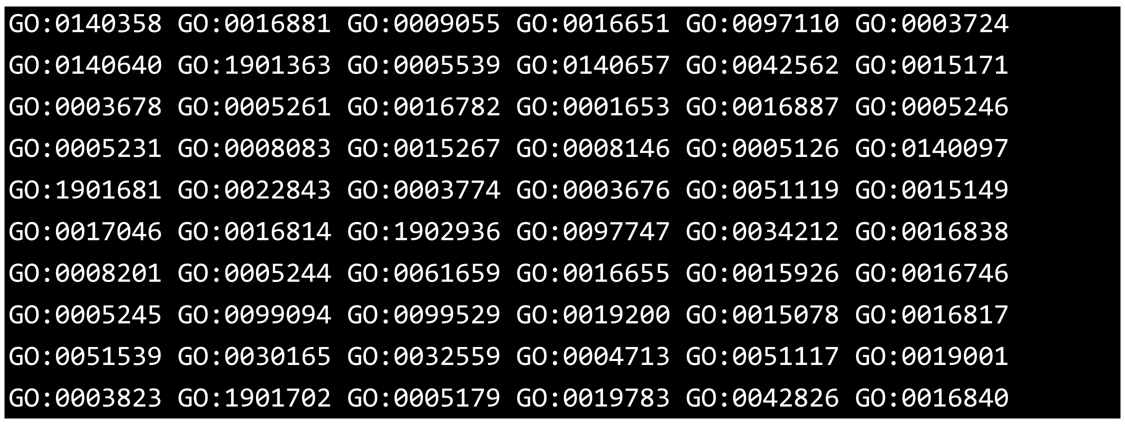
Attention weights after normalization for protein 3SWY. The left one is the plot for pretrained sgBERT, and the right one is the plot for finetuned sgBERT.



Attention weights after normalization for protein 1V9V. Left one is the plot for pretrained sgBERT, right one is the plot for finetuned sgBERT.



Attention weights after normalization for protein 1PBW. Left one is the plot for pretrained sgBERT, right one is the plot for finetuned sgBERT.

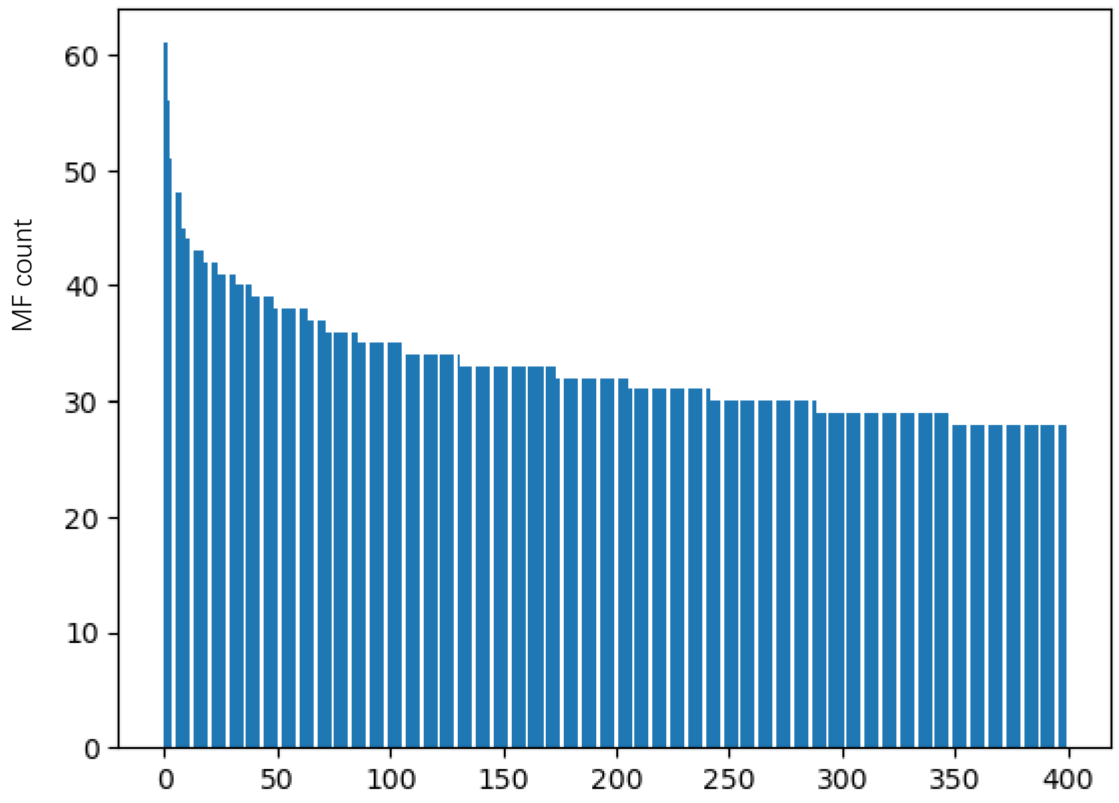


```
GO:0140358 GO:0016881 GO:0009055 GO:0016651 GO:0097110 GO:0003724
GO:0140640 GO:1901363 GO:0005539 GO:0140657 GO:0042562 GO:0015171
GO:0003678 GO:0005261 GO:0016782 GO:0001653 GO:0016887 GO:0005246
GO:0005231 GO:0008083 GO:0015267 GO:0008146 GO:0005126 GO:0140097
GO:1901681 GO:0022843 GO:0003774 GO:0003676 GO:0051119 GO:0015149
GO:0017046 GO:0016814 GO:1902936 GO:0097747 GO:0034212 GO:0016838
GO:0008201 GO:0005244 GO:0061659 GO:0016655 GO:0015926 GO:0016746
GO:0005245 GO:0099094 GO:0099529 GO:0019200 GO:0015078 GO:0016817
GO:0051539 GO:0030165 GO:0032559 GO:0004713 GO:0051117 GO:0019001
GO:0003823 GO:1901702 GO:0005179 GO:0019783 GO:0042826 GO:0016840
```

```
GO:0008047 GO:0031072 GO:0042803 GO:0043021 GO:0019887 GO:0004984
GO:0000030 GO:0008378 GO:0016903 GO:0016229 GO:0005488 GO:0008092
GO:0003677 GO:0008238 GO:0019902 GO:0044877 GO:0048029 GO:0016772
GO:0016538 GO:0046906 GO:0019213 GO:0098531 GO:0005506 GO:0015605
GO:0051213 GO:0017076 GO:0001227 GO:0015020 GO:0005524 GO:0008289
GO:1901618 GO:0019888 GO:0004518 GO:0003899 GO:0004197 GO:0019955
GO:0030545 GO:0016788 GO:0000287 GO:0015036 GO:0008081 GO:0043539
GO:0005543 GO:0015662 GO:0016893 GO:0005230 GO:0052689 GO:0002020
GO:0016922 GO:0140101 GO:0022804 GO:0016646 GO:0017171 GO:0005516
GO:0017124 GO:0005178 GO:0008270 GO:0004175 GO:0022884 GO:0015932
GO:0015101 GO:0016705 GO:0003779 GO:0015103 GO:0003714 GO:0004532
GO:0032451 GO:0009678 GO:0015085 GO:0016769 GO:0043565 GO:0022857
GO:0016462 GO:0048306 GO:0001221 GO:0015075 GO:0008324 GO:0008194
GO:0017022 GO:0042379 GO:0140678 GO:0000981 GO:0030546 GO:0097367
GO:0000149 GO:0000977 GO:0016620 GO:0016776 GO:0015291 GO:0005216
GO:0016879 GO:0016830 GO:0000978 GO:0015399 GO:0016757 GO:0050661
GO:0030276 GO:0061980 GO:0015459 GO:0003700 GO:0019239 GO:0051087
GO:0051219 GO:0008157 GO:0016248 GO:0031369 GO:0016875 GO:0045182
GO:0015145 GO:0019904 GO:0003777 GO:0003713 GO:0005125 GO:0101005
GO:0008276 GO:0042887 GO:0061733 GO:0019787 GO:0004521 GO:0008186
GO:0003725 GO:0001664 GO:0038023 GO:0019199 GO:0005372 GO:0016278
GO:0003824 GO:0050660 GO:0140104 GO:0004180 GO:0140098 GO:0016866
GO:0005319 GO:0140297 GO:0008374 GO:0008234 GO:0019894 GO:0004867
GO:0019901 GO:1904315 GO:0042054 GO:0140318 GO:0004725 GO:0140223
GO:0031267 GO:0016638 GO:0022890 GO:0003743 GO:0052866 GO:0032934
GO:0016836 GO:0015631 GO:0003684 GO:0061135 GO:0004177 GO:0016831
GO:0032561 GO:0015179 GO:0008528 GO:0008170 GO:0001216 GO:0061783
GO:0009975 GO:0016829 GO:0060589 GO:0004869 GO:0008233 GO:0016790
GO:0140359 GO:0015081 GO:0019840 GO:0003690 GO:0032182 GO:0003727
GO:0001067 GO:0016616 GO:0005343 GO:1901265 GO:0019208 GO:0030145
GO:0019207 GO:0004659 GO:0015453 GO:0017111 GO:0005496 GO:0004842
GO:0016787 GO:0046982 GO:0046983 GO:0015297 GO:0061134 GO:0016791
GO:0016874 GO:0005198 GO:0022836 GO:0051540 GO:0046332 GO:0003729
GO:0070566 GO:0038024 GO:0019903 GO:0030554 GO:0004386 GO:0016247
GO:0004683 GO:0016645 GO:0046872 GO:0016796 GO:0004896 GO:0051015
GO:0005546 GO:0016684 GO:0043167 GO:0042393 GO:0019838 GO:0004712
GO:0030971 GO:0120013 GO:0030170 GO:0033218 GO:0090079 GO:0005525
GO:0030414 GO:0140828 GO:0016706 GO:0008375 GO:0008236 GO:0003735
GO:0004312 GO:0008135 GO:0019205 GO:0004527 GO:0050839 GO:0051536
GO:0061629 GO:0004888 GO:0005200 GO:1901682 GO:0016810 GO:0003730
GO:1990837 GO:0022832 GO:0015035 GO:0098772 GO:0032553 GO:0072349
GO:0016410 GO:0008408 GO:0004407 GO:0016614 GO:0140030 GO:0010333
GO:0035091 GO:0000987 GO:0043177 GO:0004364 GO:0070063 GO:0016849
GO:0046914 GO:0140938 GO:0045296 GO:0031490 GO:0042162 GO:0022853
```

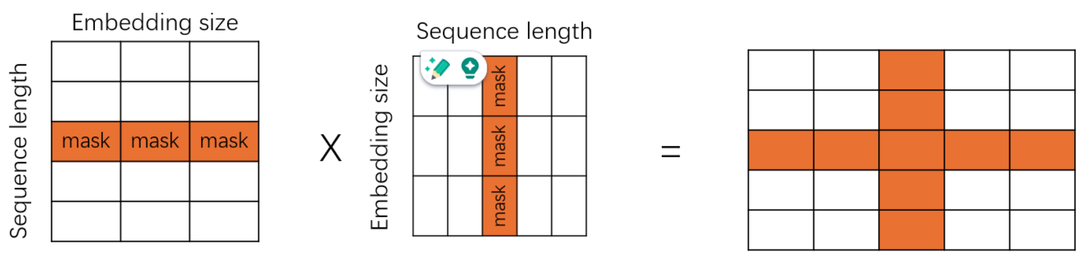
```
GO:0022834 GO:0004674 GO:0008094 GO:0060090 GO:0015295 GO:0019900
GO:0042578 GO:0140110 GO:0004553 GO:0003723 GO:0043169 GO:0004623
GO:0030246 GO:0032266 GO:0051082 GO:0016701 GO:0030594 GO:0046527
GO:0034061 GO:0060089 GO:0005507 GO:0008395 GO:0140296 GO:0016818
GO:0070851 GO:0019843 GO:0008017 GO:0043178 GO:0002039 GO:0046915
GO:0032555 GO:0005548 GO:0004519 GO:0019899 GO:0004520 GO:0004930
GO:0016835 GO:0048018 GO:0001046 GO:0016628 GO:0016763 GO:0008168
GO:0004620 GO:0008483 GO:0020037 GO:0031625 GO:0051020 GO:0004722
GO:0019209 GO:1990841 GO:0005085 GO:0030674 GO:0008013 GO:0022803
GO:0008171 GO:0016854 GO:0008757 GO:0005509 GO:0005515 GO:0008173
GO:0016878 GO:0001099 GO:0140677 GO:0042277 GO:0008237 GO:0016811
GO:0060590 GO:0016740 GO:0016799 GO:0043022 GO:0015318 GO:0004252
GO:0015108 GO:0097159 GO:0008188 GO:0005102 GO:0004857 GO:0015175
GO:0072341 GO:0030234 GO:0016597 GO:0015079 GO:0016779 GO:0051287
GO:0044325 GO:0005201 GO:0008227 GO:0016405 GO:0008509 GO:0140375
GO:0016758 GO:0030247 GO:0016298 GO:0001228 GO:0004497 GO:0140034
GO:0046873 GO:0031406 GO:0016407 GO:1990782 GO:0004536 GO:0008514
GO:0016773 GO:0005215 GO:0042802 GO:0016798 GO:0016765 GO:0035591
GO:0090729 GO:0003924 GO:0016860 GO:0035251 GO:0022829 GO:0016780
GO:0008134 GO:0050840 GO:0008028 GO:0004112 GO:0001098 GO:0019003
GO:0051018 GO:0005096 GO:0016747 GO:0016853 GO:0036094 GO:0016741
GO:0140299 GO:0001217 GO:0015144 GO:0071949 GO:0019842 GO:0016859
GO:0015293 GO:0003682 GO:0016209 GO:0005342 GO:0140096 GO:0016667
GO:0035639 GO:1901505 GO:0016301 GO:0016709 GO:0098960 GO:0004222
GO:0004402 GO:0033293 GO:0046943 GO:0090482 GO:0005518 GO:0000049
GO:0034062 GO:0016491 GO:0003697 GO:0004843 GO:0004540 GO:1901981
GO:0004721 GO:0070279 GO:0043130 GO:0043621 GO:0016877 GO:0044389
GO:0043168 GO:0016627 GO:0003712 GO:0099106
```

All the Molecular Function terms used for prediction



Top 400 proteins with maximum number of MF function annotated.

This plot is the number of Molecular functions annotated for each protein. Only 400 proteins are plotted. The protein with the greatest number of MF annotations only have 61 function terms.



Mask strategy causes NaN loss. Initially mask with extreme negative value would generate a matrx with a whole row or/and column of extreme negative value. When SoftMax implemented, the gradient would be intractable.