Bioinformatics
Advance Access Publication Date: 1 May 2023

Embbedings to Phylogenies

Adel Schmucklermann [1,*], Alexander Fastner [1,*], Tobias Olenyi [1], Ivan Koludarov [1], Kyra Erckert[1], Tobias Senoner[1], Burkhard Rost [1]

[1]Department of Informatics, Bioinformatics and Computational Biology - i12, TUM-Technical University of Munich, Boltzmannstr. 3, Garching, 85748, Munich, Germany.

Abstract

Motivation:
This paper explores the use of protein sequence embeddings as a means of constructing phylogenetic trees, with the aim of reducing the time required compared to traditional methods. By plotting the correlation between sequence similarity and embeddings, we found that certain dimensions of the embeddings correlated more than others, confirming that the embeddings do contain evolutionary information. To filter out noise and extract more information, we employed PCA and t-SNE techniques, as well as a Variational Autoencoder (VAE) to reduce embedding dimensions.
Results: Our analysis showed that while the embeddings do contain some evolutionary information, they also contain a significant amount of noise. We found that a VAE did not improve the quality of constructed trees, and that the presence of a proper outgroup is crucial for constructing a solid tree.
Availability: Github: https://github.com/AlexanderFastner/PyTorch_Lightning_VAE
Contact: ge39row@tum.de , ge34vot@tum.de
Supplementary information: Supplementary data are available in the Appendix.

1 Introduction

Phylogenetic trees are a vital tool in evolutionary biology used to visualize the relationships between different organisms based on their genetic sequences. The process of constructing phylogenetic trees involves comparing the genetic sequences of different organisms and identifying the similarities and differences between them. Traditionally, multiple sequence alignments (MSAs) are used to align the protein sequences to find conserved regions. The proteins are then clustered by these regions to build phylogenetic trees. The similarity of sequences in evolutionary relationships is generally derived through methods such as maximum likelihood or Bayesian inference. Although MSAs are currently the most reliable way to create trees, it is a very time consuming method. In order to reduce time, we used protein sequence embeddings to cluster them without doing MSAs first.
Protein embeddings are a numerical representation of a protein sequence that capture important features of the sequence itself. These embeddings can be compared at a granular level to create more accurate phylogenetic trees faster than with traditional methods. We verified that the embeddings contain evolutionary information by plotting correlation against sequence similarity, the measure used by previous methods constructing trees.
Looking at the correlation between sequence similarity and the dimensions of the embeddings in our Uniclust30 and Uniclust90 data sets, we found that certain dimensions correlated more than others. To extract more information and filter out noise, we utilized principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE). We reduced the embedding dimensions as in [11] with a Variational Autoencoder (VAE) and applied neighbor joining to construst unrooted trees and UPGMA for rooted trees.

2 Methods

2.1 Data Set

To verify that the information contained in the embeddings correlates with sequence similarity we ran an analysis on Uniclust30 and Uniclust90 datasets [7]. We used a random selection of 1000 samples from each and calculated sequence and embedding similarities from them.
We used the KLK dataset [3], and worked with both the esm-2 [12] and prott5 [2] embeddings of it. We found that the esm2 embeddings performed better and primarily focused our analysis on these.
We also ultilized both the kinase and phosphatase datasets containing headers and sequences from [11]. We generated our own esm-2 embeddings from these to use as a comparison against the KLK esm-2 embeddings.

2.2 Sequence Similarity

We build embeddings for 1000 random sequences of Uniclust30 and Uniclust90 [7] and calculated their correlation with Needleman Wunsch sequence similarity scores. Existing MSA based tree construction algorihts like NJ and UPGMA rely on sequence similarity to construct trees and we verified that some information contained in the embedding correlates to sequence similarity.
We calculated the Pearson's, Kendall and Spearman's correlation of the euclidean, cosine and chebyshev distance of the embeddings and sequence similarity scores of Uniclust30 and Uniclust90. To verify that our results were not skewed by sequence length we tested against sequence identity.

2.3 Correlation of Sequence Similarity in Embedding Dimensions

We looked at each dimension of the embedding vectors individually. A single dimension of two embeddings got tested against the sequence similarity score of the two corresponding protein sequences with Spearman's correlation, since it performed the best on the previous analysis. In this part of the study, the cosine distance of two FASTA sequences of the proteins was used as a pairwise similarity measure. This pairwise comparison was performed on every single dimension of the 1024 dimensional embedding vector for every protein sequence pair.
We also created vectors of dimension pairs with different sizes and set them against sequence similarity. One dimension pair vector consisted of ranging from two to nine dimensions. All individual dimensions got permutated without repetition with each other. The same calculations were carried out as previously mentioned.
In order to find the best fitting linear relationship between the similarity of the embeddings, we applied linear regression on the 1000 Uniclust90 embeddings. All dimension pair vectors of size two were used as x-input and the corresponding sequence similarity of the pairs as labels. A second linear regression was applied on all dimensions at once for comparison.

2.4 Dimension Reduction

2.4.1 Dimension Reduction with VAE

We utilized a VAE as a way to reduce the dimensions of the embeddings and hopefully eliminate noise and thus improve signal. We executed the VAE and the additional branch support analysis from [11] on our data set, which did not perform well on the our data and hence implemented a new VAE and branch support ourselves.
Our model was trained on the KLK-esm2 dataset which we split into training and validation sets of 314 and 79 respectively. During hyperparameter optimization the model was trained with one, three, 100 and 500 epochs, with one epoch delivering the best result. The architecture utilized three hidden layers and we tested with various sizes for each. The sizes are relative and the decoder uses mirrored sizes: 1/2/8/16, 1/4/8/16, 1/4/16/32, 1/8/32/64.
Moreover, the following non-linear activation functions from the torch.nn package were tested as well: ReLU, Sigmoid, Tanh and Mish. The best parameters we found were; one epoch, hidden layer sizes of 1/4/16/32, and a Sigmoid activation function.

2.4.2 Dimension Reduction with t-SNE and PCA

To compare the results of the VAE to more widely used dimension reduction techniques, t-SNE and PCA were performed on the KLK-esm2 data set. t-SNE with perplexity = 22, learning_rate = 100, n_iter = 1000, random_state = 42 and PCA with n_components = 2, random_state = 42.

2.5 Tree Construction

After training the VAE on 314 embeddings, the encoder was called to encode all 437 embeddings of the KLK-esm2 data set into the latent space. To build the distance matrix as well as the clustering, we used the code from [11].
Before applying a clustering algorithm to produce phygenetic trees a distance matrix was created with four different metrics: cosine, euclidean, manhattan and TS-SS. TS-SS stands for triangle area similarity (TS) and sector area similarity (SS) and is a vector similarity measure [4], which is set to perform better on larger data sets than euclidean or cosine distance. On each distance matrix neighbor joining (NJ) [8] as well as UPGMA [9] were carried out separately to construct an unrooted and rooted tree with the corresponding newick file.

2.6 Comparison of Trees

We used the data, which was created with ExaBayes [1] after building the MSA on the KLK protein sequences as the reference tree. The tree contructed on the KLK esm2 embeddings without any dimension reduction technique, neither VAE, t-SNE nor PCA, will be referred to as the non-trained tree. Every tree was compared to the two trees.
The comparison was based on the robinson-foulds (RF) distance, euclidean distance and symmetric difference[10] as well as RF distance and partitions of the branches that were found in one tree but not the other [5]. Furthermore, we inspected the trees also visually with iTOL [6] and the ETE3 visualization package [5].

3 Results and Discussion

3.1 Correlation with Sequence Similarity

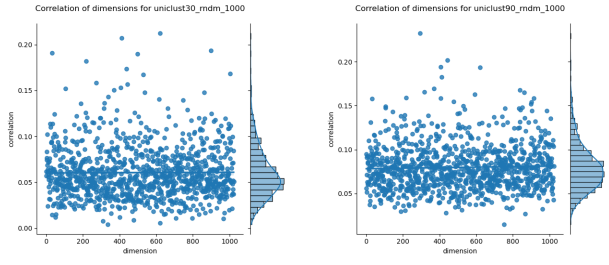
Overall, the non-parametric Spearman's correlation had more significant results than the parametric Pearson's and the non-parametric Kendall rank coefficient, which evaluates ordinal associations. The cosine distance of the Uniclust90 embeddings had the highest Pearson's and Spearman's correlation score of 0.31 with a p-value of 2.2e-16 (Table 1). Uniclust90 had in general higher correlation scores than Uniclust30, which is expected, because the protein sequences in Uniclust90 have a pairwise sequence identity up to 90% and in Uniclust30 only up to 30%. The euclidean distance had a slightly higher correlation with sequence similarity on Uniclust30 than cosine or chebyshev. Regarding Uniclust90, the cosine distance resulted in a more appropriate measure. We decided to use Spearman's correlation and cosine distance for further analysis.

	Pearson's		**Kendall**		**Spearman's**	
Dataset	Uniclust30	Uniclust90	Uniclust30	Uniclust90	Uniclust30	Uniclust90
euclidean	0.22	0.28	0.17	0.20	0.25	0.29
cosine	0.21	0.31	0.14	0.21	0.21	0.31
chebyshev	0.19	0.25	0.15	0.18	0.22	0.27

Table 1. Correlation of Embedding Distance and Sequence Similarity

3.2 Dimension Importance

Taking a closer look at the individual dimensions of the embeddings and comparing them to similarity of the FASTA sequences showed that certain dimensions are more correlated with sequence similarity than others (Fig. 1). Setting a cut-off at 0.15, the number of significant dimensions with a correlation of 0.15 or above was 14 for Uniclust30 and 17 for Uniclust90. Both data sets share the following eight significant dimensions: 31, 216, 369, 406, 410, 422, 621 and 897. Moreover, Uniclust30 has more dimensions around 0.2. Nonetheless, there is one outlier with the highest correlation of 0.23, which is dimension 294 for Uniclust90.
Taking into account only significant dimensions with a correlation threshold of 0.15 and building all permutations of dimension pair vectors of size two to nine, the correlation did not increase with larger dimension vectors. Instead the scores averaged more out.
Furthermore, the linear regression on the 1000 protein embeddings of Uniclust90 found a fit with a consistent Root Mean Squared Error (RMSE) of around 0.07 over all dimension pair vectors of size two. Giving all dimensions at once to the second linear regression model resulted also in an RMSE of 0.07. The established intercept was 0.1 and the slope 0.01 of the second model. As there is evidence for a linear relationship, we decided to we implemented a VAE as in [11] to train the model on the distribution of the embeddings over all dimensions with a chance to increase the correlation between sequence similarity and embedding distance.



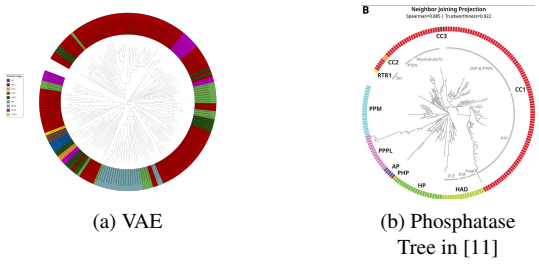
(a) Uniclust30 (b) Uniclust90

Fig. 1: **Correlation of Embedding Dimensions and Sequence Similarity.** Spearman's correlation of 1000 random sequences of Uniclust30 and Uniclust90 and sequence similarity.

3.3 Dimension Reduction to Construct Trees

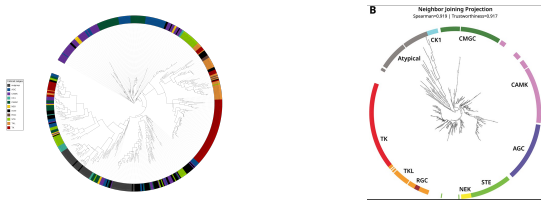
3.3.1 Variational Autoencoder

The paper [11] presented promising results in constructing phylogenetic trees using their VAE approach on the kinase and phosphatase datasets. In an effort to validate their findings and test the generalizability of our VAE approach, we attempted to replicate their results on these same datasets. The circular visualization for our trees is done with [6]. We were unable to recreate the successful tree construction of [11] in our attempt at a VAE. While our trained tree does seem to cluster somewhat, it does not give the perfectly polished output seen in [11].



(a) VAE (b) Phosphatase Tree in [11]

Fig. 2: **Comparison of Phosphatase Trees.** On the left hand side is the output from our VAE training and on the right is the NJ tree generated by [11].



(a) VAE (b) Kinase Tree in [11]

Fig. 3: **Comparison of Kinase Trees.** On the left hand side is the output from our VAE training and on the right is the NJ tree generated by [11].

In Fig. 2 our VAE tree is rather scattered and does not cluster nearly as nicely as b) from [11]. When testing on the kinase data set (Fig. 3) we were again unable to replicate the results reported in [11]. In their paper they use branch support metrics to help improve the trees. They sample many trees from the VAE and use the already known reference tree to only include branches which appear in the newly sampled trees and the reference tree above a threshold many times. This shows as the small number of blank branches decreases, leading to a more polished looking tree on the kinase data set. When we re-implemented branch support ourselves that does not rely on the reference tree, but uses only the newly created trees. It did not significantly improve our resulting trees. Because of this, we must conclude that using the reference tree in the training and validation process as in [11] improves tree construction, but is not useful for any data set without a known reference tree.

3.3.2 VAE Results on KLK Data Set

The KLK dataset was our dataset for testing our VAE approach due to several factors. First as we discovered, evaluating trees based solely on numeric metrics is difficult. Thus, having prior knowledge of this dataset and an intuition about how the final tree should look, made comparing various trees easier. It having a substantial outgroup also likely helped in tree construction.

The encoder of the trained VAE was used to map all 437 KLK embeddings into the latent space. The evaluation of the number of training iterations of one, three, 100 and 500 epochs showed that the outcome trees of the model trained on one epoch were the most similar to the reference tree. Out of the four tested activation functions (ReLU, Sigmoid, Tanh and Mish), Sigmoid captured the distribution of the embeddings the best and came the closest to the dispersion of distances of the not VAE reduced embeddings (Fig. 4 and Appendix).

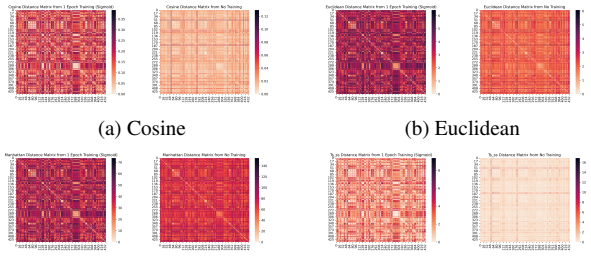
Since neighbor joining clusters the nodes by the smallest distance to each other, we looked at how the pairwise distances are spread across the trained and non-trained (no dimension reduction, embeddings for the four metrics (cosine, euclidean, manhattan and TS-SS) (Fig. 4). Although the total range of values for manhattan differs the most between trained and non-trained, the distribution of the values holds the closest. Moreover, the non-trained embeddings have stronger outliers that are more distant to other embeddings, which are essential to building the correct tree, but are discarded by the VAE.

Regarding the choice of a distance metric for neighbor joining and UPGMA, we found that on the VAE embeddings with Sigmoid activation function euclidean and manhattan distance provided the best tree construction with neighbor joining (Fig. 5). On other activation functions and UPGMA, cosine and euclidean gave the most promising results. Triangle Area Similarity – Sector Area Similarity (TS-SS) was, in our testing, the farthest away from the reference tree on any metric.

We tested using various loss functions to improve training performance; Kullback–Leibler divergence (KL), binary cross entroy (BCE), and mean squared error (MSE). We found that changing the loss function did little to change the output trees but MSE had a slight edge and was used for the generated trees in this report.

3.3.3 Reduced Data Set

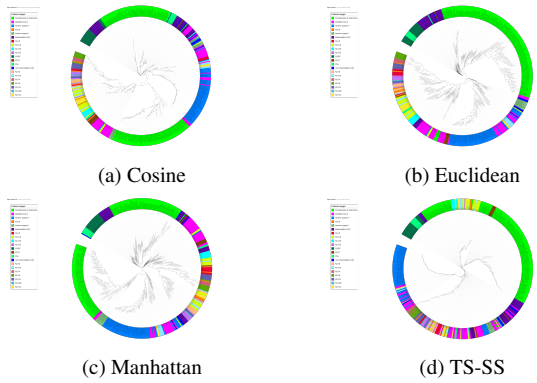
To test the significance of how not having an outgroup would affect tree construction we removed the outgroup from the KLK dataset. The removal of the outgroup noticeably worsens the generated reference tree (Fig. 6)). In this case the VAE training results in an improvement over the reference. All distance metrics were an improvement over the untrained tree with euclidean performing best.



(a) Cosine (b) Euclidean

(c) Manhattan (d) TS-SS

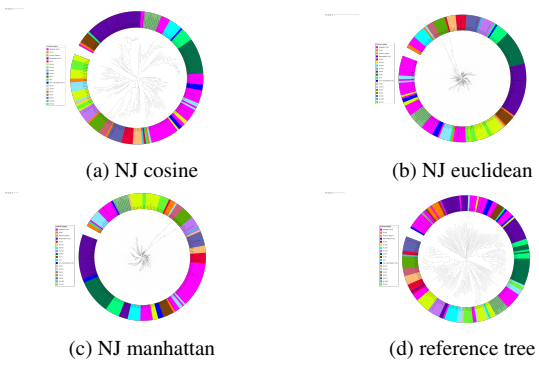
Fig. 4: **Distance Matrix of VAE-Trained and Non-Trained KLK Embeddings.** Heatmap of the distance matrix calculated with cosine, euclidean, manhattan and TS_SS distance of KLK embeddings trained on VAE with one epoch and Sigmoid activation function compared to the distance matrix of the KLK embeddings without any dimension reduction.



(a) Cosine (b) Euclidean

(c) Manhattan (d) TS-SS

Fig. 5: **Phylogenetic trees of VAE Trained KLK Embeddings.** Phylogenetic trees of VAE trained with sigmoid activation function on the KLK data set. Four different distance metrices (cosine, euclidean, manhattan and TS_SS) were used to calculate the distance matrix and apply NeighborJoining.



(a) NJ cosine (b) NJ euclidean

(c) NJ manhattan (d) reference tree

Fig. 6: **Comparison of Reduced KLK Trees.** NJ trees with various distance metrics with training and the NJ cosine reference tree.

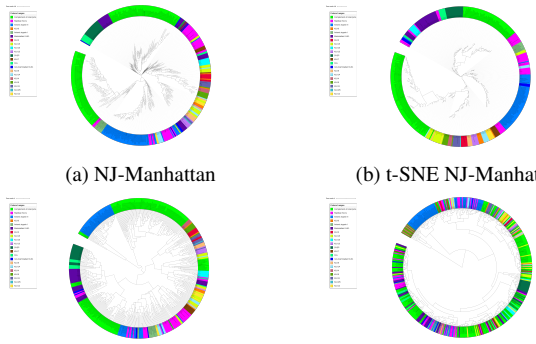
3.3.4 T-SNE and PCA

To explore the effectiveness of dimensionality reduction techniques for our embedding vectors, we applied two commonly used methods, t-SNE and PCA, to our dataset. Both t-SNE and PCA are widely used techniques for reducing high-dimensional data to a lower-dimensional space while preserving important structure and relationships between data points. A summary of the results can be seen in Fig. 7.

The NJ plots are much closer between the VAE and t-SNE with Manhattan distance working well. The UPGMA tree is rather scattered on the VAE but the PCA UPGMA construction did not perform well on this dataset and is not close to the reference tree. We can conclude that choosing between NJ and UPGMA is dependent on the dimension reduction technique. For VAE and t-SNE, neighbor joining reveals the best results and when using PCA, UPGMA would be the better option.

Although the numeric comparison of a newly constructed tree to the reference tree varied greatly even on similar looking trees, it gave a first overview. Out of the the four ways to construct a tree (NJ-metric, UPGMA-metric, t-SNE-NJ-metric, PCA-UPGMA-metric), each paired with all of the four metrics (cosine, euclidean, manhattan, TS-SS), t-SNE-manhattan had the lowest RF distance with 672 of 850.

Shown in Fig. 12 are the outputs of our VAE, t-SNE, non-trained and the reference tree in a circular [6] and non-circular visualization. Both our VAE and t-SNE seperate the outgroup (light green) into two clusters whereas the reference and non-trained tree keep it together. t-SNE keeps the proteins of the individual families more together in their group than VAE does. Nonetheless, VAE comes close to t-SNE in tree construction but, ultimately we conclude that the non-trained tree is the closest to the reference.



(a) NJ-Manhattan (b) t-SNE NJ-Manhattan

(c) UPGMA-Euclidean (d) PCA UPGMA-Euclidean

Fig. 7: **Phylogenetic Trees of KLK Embeddings Trained with VAE, t-SNE and PCA.** Phylogenetic trees of KLK embeddings, VAE-trained with 1 epoch and sigmoid activation function, were constructed with either NJ and manhattan distance or UPMGA and euclidean distance and compared to t-SNE and PCA trained. a) Dimension reduction by VAE and build by NJ and manhattan distance. b) Dimension reduction was performed with t-SNE and the tree build by NJ and manhattan distance. c) Dimension reduction by VAE and build on UPGMA and euclidean distance. d) Dimension reduction by PCA and build on UPGMA and euclidean distance.

4 Conclusion

Based on the analysis of the correlation between embeddings and sequence similarity, it can be concluded that the embeddings capture some evolutionary information but also contain a significant amount of noise. The use of a VAE did not lead to an improvement in the quality of constructed trees compared to embeddings without any dimension reduction technique. PCA did not effectively capture more information for constructing trees. t-SNE, on the other hand, performed better than PCA and was about equal or sometimes even better to the VAE.

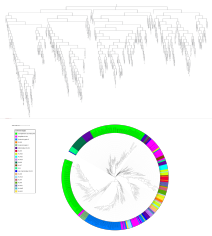


Fig. 8: VAE

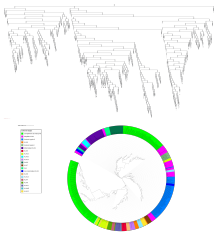


Fig. 9: t-SNE

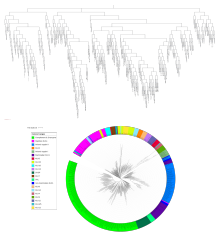


Fig. 10: Non-Trained

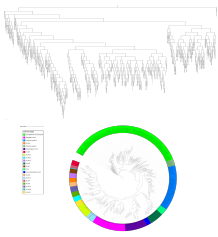


Fig. 11: Reference

Fig. 12: **Phylogenetic Trees of KLK Embeddings.** Phylogenetic trees of KLK embeddings with two different dimension reduction techniques: t-SNE and VAE. They are compared to the reference tree. Fig. 8) Dimension reduction by VAE and build by NJ and manhattan distance. Fig. 9) Dimension reduction was performed with t-SNE and the tree build by NJ and manhattan distance. Fig. 10) Non-Trained tree, without any dimension reduction and build by NJ and manhattan distance. Fig. 11) Reference Tree.

The reference tree created without any VAE training was found to have definite room for improvement, but the conclusion was drawn that the VAE did not find the right distribution to effectively reduce dimensions to improve the non-trained tree. The exception to this is when no outgroup is present in a data set. It was found that in order to have a solid tree, an outgroup is necessary with a cluster distance not too near or far away from the data points. Without an outgroup, the VAE-trained tree was an improvement on the reference, but with a proper outgroup, the training did not seem to improve the tree.

Acknowledgements

We would like to thank the Rostlab and especially, Tobias Olenyi, Ivan Koludarov, Kyra Erckert and Tobias Senoner for their support and supervision.

Funding

This work did not receive any funding.

References

[1]Andre J Aberer, Kassian Kobert, and Alexandros Stamatakis. Exabayes: Massively parallel bayesian tree inference for the whole-genome era. *Molecular Biology and Evolution*, page msu236, 2014.

[2]Christian Dallago Ghalia Rehawi Yu Wang Llion Jones Tom Gibbs Tamas Feher Christoph Angerer Martin Steinegger Debsindhu Bhowmik Burkhard Rost Ahmed Elnaggar, Michael Heinzinger. Prottrans: Towards cracking the language of life's code through self-supervised learning. *BioRxiv*, 2021.

[3]Mikheyev AS Barua A, Koludarov I. Co-option of the same ancestral gene family gave rise to mammalian and reptilian toxins. *BMC Biol*, 19(1):268, 2021.

[4]Arash Heidarian and Michael J. Dinneen. A hybrid geometric approach for measuring similarity level among documents and document clustering. pages 142–151, 2016.

[5]Francois Serra Jaime Huerta-Cepas and Peer Bork. Ete 3: Reconstruction, analysis and visualization of phylogenomic data. *Mol Biol Evol*, 2016.

[6]Ivica Letunic and Peer Bork. Interactive Tree Of Life (iTOL) v5: an online tool for phylogenetic tree display and annotation. *Nucleic Acids Research*, 49(W1):W293–W296, 04 2021.

[7]Milot Mirdita, Lars von den Driesch, Clovis Galiez, Maria J. Martin, Johannes Söding, and Martin Steinegger. Uniclust databases of clustered and deeply annotated protein sequences and alignments. *Nucleic Acids Research*, 45(D1):D170–D176, 11 2016.

[8]N Saitou and M Nei. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425, 07 1987.

[9]Robert R. Sokal and Charles Duncan Michener. A statistical method for evaluating systematic relationships. *University of Kansas science bulletin*, 38:1409–1438, 1958.

[10]J. Sukumaran and Mark T. Holder. Dendropy: A python library for phylogenetic computing. *Bioinformatics 26*, pages 1569–1571, 2010.

[11]Wayland Yeung, Zhongliang Zhou, Liju Mathew, Nathan Gravel, Rahil Taujale, Brady O'Boyle, Mariah Salcedo, Aarya Venkat, William Lanzilotta, Sheng Li, and Natarajan Kannan. Tree visualizations of protein sequence embedding space enable improved functional clustering of diverse protein superfamilies. *Briefings in Bioinformatics*, 24(1), 01 2023. bbac619.

[12]Roshan Rao Brian Hie Zhongkai Zhu Wenting Lu Nikita Smetanin Robert Verkuil Ori Kabeli Yaniv Shmueli Allan dos Santos Costa Maryam Fazel-Zarandi Tom Sercu Salvatore Candido Alexander Rives Zeming Lin, Halil Akin. Evolutionary-scale prediction of atomic level protein structure with a language model. *BioRxiv*, 2022.

Appendix

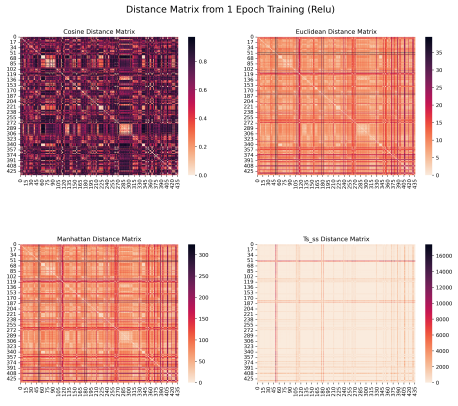


Fig. 13: **Distance Matrix of ReLU VAE-Trained KLK Embeddings.** Heatmap of the distance matrix calculated with cosine, euclidean, manhattan and TS_SS distance of KLK embeddings trained on VAE with one epoch and ReLU activation function.

Fig. 14: **Distance Matrix of Mish VAE-Trained KLK Embeddings.** Heatmap of the distance matrix calculated with cosine, euclidean, manhattan and TS_SS distance of KLK embeddings trained on VAE with one epoch and Mish activation function.

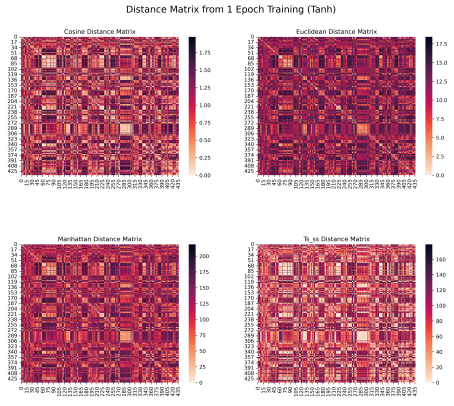


Fig. 15: **Distance Matrix of Tanh VAE-Trained KLK Embeddings.** Heatmap of the distance matrix calculated with cosine, euclidean, manhattan and TS_SS distance of KLK embeddings trained on VAE with one epoch and Tanh activation function.