Fast protein structure searching using structure graph embeddings

Joe G Greener
Medical Research Council Laboratory of Molecular Biology
Cambridge, UK
jgreener@mrc-lmb.cam.ac.uk

Kiarash Jamali
Medical Research Council Laboratory of Molecular Biology
Cambridge, UK

Abstract

Comparing and searching protein structures independent of primary sequence has proved useful for remote homology detection, function annotation and protein classification. With the recent leap in accuracy of protein structure prediction methods and increased availability of protein models, attention is turning to how to best make use of this data. Fast and accurate methods to search databases of millions of structures will be essential to this endeavour, in the same way that fast protein sequence searching underpins much of bioinformatics. We train a simple graph neural network using supervised contrastive learning to learn a low-dimensional embedding of protein structure. The embedding can be used to search structures against large structural databases with accuracy comparable to current methods. The speed of the method and ability to scale to millions of structures makes it suitable for this structure-rich era. The method, called Progres, is available at https://github.com/jgreener64/progres.

Introduction

A variety of methods have been developed to compare, align and search with protein structures [1] including comparing residue-residue distances [2, 3], considering local geometry [4], coordinate alignment [5] and 3D Zernike descriptors [6, 7]. Since protein structure is more conserved than sequence [8] these methods have proved useful in remote homology detection, protein classification [9], inferring function from structure [10] and assessing the accuracy of structure predictions. The highest accuracy methods tend to be careful comparisons based on coordinates like Dali [3], but searching large structural databases [11, 12] such as the AlphaFold Protein Structure Database [13, 14] or the ESM Metagenomic Atlas [15] with these methods is slow. Recently Foldseek [16] has addressed this problem by converting primary sequence into a sequence of learned local tertiary motifs. It then uses the rich history of fast sequence searching in bioinformatics to dramatically reduce the pairwise comparison time of the query with each member of the database. It follows that to further reduce search time, the pairwise comparison step should be made even faster.

Inspired by the impressive performance of simple graph neural networks (GNNs) using coordinate information for a variety of molecular tasks [17], we decided to train a model to embed protein structures into a low-dimensional representation. Two embeddings can be compared very quickly by cosine similarity and a query can be compared to each member of a pre-embedded database in a vectorised manner on CPU or GPU. It makes sense to use expertly-curated classifications of protein structures when training such an embedding [18, 9, 19]; we use supervised contrastive learning [20]

Machine Learning for Structural Biology Workshop, NeurIPS 2022.

to allow the embedding to be learned in a manner that reflects such an understanding of protein structure space and returns search results consistent with it.

Our method is similar to another approach that embeds protein structure [21], though our embedding is based on coordinates rather than hydrogen bonds. Embedding protein folds has also been done using residue-level features [22, 23], and GNNs acting on protein structure have been used for function prediction [24]. Other studies have used unsupervised contrastive learning on protein structures and show that the representations are useful for downstream prediction tasks including protein structural similarity [25, 26]. However, their contrastive learning is based on sub-structures in a protein rather than a protein classification. Another approach [27] adopts a similar strategy but makes use of residue-residue distances and dihedral angles. Contrastive learning using protein classifications has also improved language models for protein sequences, showing clustering that better preserves protein structure space [28].

Methods

Training

Structures in the Astral 2.08 95% sequence identity set including discontinuous domains were used for training [29]. We chose 400 domains randomly from the Astral 2.08 40% sequence identity set to use as a test set (see below) and another 200 domains to use as a validation set to monitor training. We removed domains with 30% or greater sequence identity to these 600 domains using MMseqs2 [30], and also removed domains with fewer than 20 or more than 500 residues. This left 30,549 domains in 4,862 families for training.

mmCIF files were downloaded and processed with Biopython [31]. Some processing was also carried out with BioStructures.jl [32]. Cα atoms were extracted for the residues corresponding to the domain. Each Cα atom is treated as a node with the following features: number of Cα atoms within 10 Å divided by the largest such number in the protein, whether the Cα atom is at the N-terminus, whether the Cα atom is at the C-terminus, and a 64-dimensional sinusoidal positional encoding for the residue number in the domain [33].

PyTorch was used for training [34]. The neural network architecture was similar to the E(n)-equivariant GNN in Satorras et al. 2021 [17]. We used a PyTorch implementation (`https://github.com/lucidrains/egnn-pytorch`) and a configuration similar to the molecular data prediction task, i.e. not updating the particle position. In this case the model is E(3)-invariant and is analogous to a standard GNN with relative squared norms inputted to the edge operation [17]. Edges are sparse and are between Cα atoms within 10 Å of each other. 12 such layers with residual connections are preceded by a one-layer multilayer perceptron (MLP) acting on node features and followed by a two-layer MLP acting on node features. Node features are then sum-pooled and a two-layer MLP generates the output embedding, which is normalised. Each hidden layer has 128 dimensions and uses the Swish/SiLU activation function [35], apart from the edge MLP in the GNN which has a hidden layer with 256 dimensions and 64-dimensional output. The final embedding has 128 dimensions.

Supervised contrastive learning [20] is used for training. Each epoch cycles over the 4,862 training families. For each family, 5 other families are chosen randomly. For each of these 6 families, 6 domains from the family in the training set are chosen randomly. If there are fewer than 6 domains in the family, duplicates are added to give 6. This set of 36 domains with 6 unique labels is embedded with the model and the embeddings are used to calculate the supervised contrastive loss with a temperature of 0.1 [20]. During training only, Gaussian noise with variance 1.0 Å is added to the x, y and z coordinates of each Cα atom. Training was carried out with the Adam optimiser [36] with learning rate 5×10^{-5} and weight decay 1×10^{-16}. Each set of 36 domains was treated as one batch. Training was stopped after 500 epochs and the epoch with the best family sensitivity on the validation set was used as the final model. Training took around a week on one RTX A6000 GPU.

Testing

For testing a similar approach to Foldseek was adopted [16]. The 15,177 Astral 2.08 40% sequence identity set domains were embedded with the model. 400 of these domains were chosen randomly and held out of the training data as described previously. Like Foldseek, we only chose domains

Table 1: Comparison of ability to retrieve homologous proteins from SCOPe. A similar procedure is followed as for Foldseek [16] with a set of 400 domains. For each domain the fraction of TPs detected up to the first incorrect fold is calculated (higher is better). TPs are same family in the case of family-level recognition, same superfamily and not same family in the case of superfamily-level recognition, and same fold and not same superfamily in the case of fold-level recognition. The mean of this fraction over all 400 domains is reported. Run time (single) is the time taken to search a structure of 150 residues (d1a6ja_ in PDB format) against all the 15,177 Astral 2.08 40% sequence identity set domains, with the database pre-embedded in the case of Progres and ESM-2. Run time (all-v-all) is the time taken to calculate all pairwise distances between the 15,177 domains from structure, not from embedding in the case of Progres and ESM-2. ESM-2 and MMseqs2 use sequence not structure for searching.

Software	**Family**	**Superfamily**	**Fold**	**Run time (single)**	**Run time (all-v-all)**
Progres (this work)	0.878	0.680	0.144	1.49 s (CPU)	309 s (CPU), 205 s (GPU)
Dali [3]	0.885	0.709	0.168	508 s	> 1 month
Foldseek [16]	0.821	0.578	0.070	2.11 s	154 s
TM-align fast [5]	0.806	0.594	0.100	390 s	∼23 days
ESM-2 [15]	0.477	0.221	0.014	28 s (GPU)	590 s (GPU)
MMseqs2 [30]	0.433	0.165	0.001	0.90 s	17.1 s

with at least one other family, superfamily and fold member. For each of these 400 domains, the cosine embedding distance to each of the 15,177 domains was calculated and the domains ranked by similarity. For each domain, we measured the fraction of true positives (TPs) detected up to the first incorrect fold detected. TPs are same family in the case of family-level recognition, same superfamily and not same family in the case of superfamily-level recognition, and same fold and not same superfamily in the case of fold-level recognition. Progres, Dali, Foldseek, TM-align and MMseqs2 were all run on an Intel i9-10980XE CPU and with 256 GB RAM. Progres was run with PyTorch 1.11. Progres, Foldseek and MMseqs2 were run on 16 threads. For TM-align we used the fast mode. ESM-2 embeddings used the esm2_t36_3B_UR50D model which has a 2560-dimensional embedding. The mean of the per-residue representations was normalised and comparison between sequences was carried out with cosine similarity. ESM-2 and Progres in GPU mode were run on a RTX A6000 GPU.

Results

We trained a simple GNN, called Progres (PROtein GRaph Embedding Search), to embed a protein structure independent of its sequence. Since we use distance features based on coordinates the embedding is E(3)-invariant, i.e. it doesn't change with translation, rotation or reflection of the input structure. Supervised contrastive learning [20] on SCOPe domains [18, 37] is used to train the model, moving domains closer or further apart in the embedding space depending on whether they are in the same SCOPe family or not. Sinusoidal position encoding [33] is also used to allow the model to effectively use information on the sequence separation of residues. The main intended use of such an embedding is fast searching for similar structures by comparing the embedding of a query structure to the pre-computed embeddings of a database of structures.

In order to assess the accuracy of the model for structure searching, we follow a similar procedure to Foldseek [16]. Since our model is trained on SCOPe domains it is important not to use domains for training that appear in the test set. We select a random set of 400 domains from the Astral 2.08 40% sequence identity set for testing. No domains in the training set have a sequence identity of 30% or more to these 400 domains. This represents the realistic use case that the query structure has not been seen during training - for example it is a predicted or new experimental structure - but other domains in the family have been seen during training. The easier case of searching with the exact domains used for training gives superior results that are not reported here, and the harder case of searching with completely unseen folds is discussed later.

As shown in Table 1 our model has sensitivity comparable to Dali [3], Foldseek and TM-align [5] for recovering domains in SCOPe from the same family, superfamily and fold. It appears that for some complex β and α/β folds Progres is able to perform better than Foldseek, and the higher sensitivity at

the fold level suggests that it may be better at detecting remote homology. It is considerably more sensitive than the ESM-2 protein language model embeddings [15], which here are used directly for searching without downstream training, and the baseline sequence searching method of MMseqs2 [30]. This indicates the benefits of comparing structures rather than just sequences for detecting homology. Our model does not give structural alignments but if these are required they can be computed with tools like Dali after fast initial filtering with Progres.

For searching a single structure against SCOPe on CPU the model is marginally faster than Foldseek with most run time in Python module loading. For example, going from 1 to 100 query structures increases run time from 1.49 s to 3.13 s. When searching with multiple structures, most run time is in generating the query structure embeddings. Consequently, the speed benefits of the method arise when searching a structure or structures against the pre-computed embeddings of a huge database such as the AlphaFold database [7, 23]. By artificially copying the SCOPe search set to 10 million or 100 million structures we find that our method takes 2.7 s or 15.7 s to search this set with a single query on CPU. Such a search would take much longer with Foldseek. This is an indication that very fast and accurate searching of large databases is possible.

Figure 1 shows a 2D t-SNE embedding [38] of the 128 dimensions of our model embedding for SCOPe. This shows the lower-dimensional protein fold space [39–41] created by our embedding. SCOPe classes tend to cluster together, with α+β folds appearing between the all α and all β folds which show little overlap. Folds tend to cluster together and there is a clear protein size gradient across the t-SNE embedding. The two orthogonal properties that vary across the t-SNE embedding seem to be α to β (top right to bottom left in Figure 1) and small to large domain (top left to bottom right).

Discussion

The model presented here is trained and validated on protein domains; due to the domain-specific nature of the training it is not expected to work without modification on protein chains containing multiple domains, long disordered regions or complexes. It is not immediately clear how the contrastive training would transfer to the multi-domain case. For the model to be useful for searching the AlphaFold database, the database structures would first have to be split into domains before the embeddings are pre-computed. There has been progress on this front from CATH-Assign [43], which gives around 370,000 models for the initial AlphaFold models for 21 model organisms and should scale up to the larger recent batch of models. Splitting database structures into domains could also be done at lower accuracy using automated splitting based on residue-residue contacts [44] and the pLDDT or predicted aligned error of the AlphaFold models. The query should also be single domain, though larger structures can be split manually. Searching with domains separately can overcome issues that arise from searching with multiple domains at the same time, such as missing related proteins due to differing orientations of the domains.

One issue with supervised learning on domains is whether performance drops when searching with domains that the model has not seen anything similar to during training. We trained an identical model on a different dataset where 200 domains were used for testing and domains were removed from the training set if they were from the same SCOPe superfamily as any of the testing domains. The family, superfamily and fold sensitivities analogous to Table 1 are 0.545, 0.390 and 0.183 respectively. This indicates a drop in performance for retrieving domains from the same family and superfamily but similar performance for retrieving domains from the same fold. When searching against SCOPe with Progres this drop in performance is not relevant since no superfamilies have been excluded from training. It could present an issue when searching for novel folds in large databases such as the AlphaFold database or the ESM Metagenomic Atlas [15]. Training on the larger set of CATH-Assign domains would go some way to addressing this.

Aside from searching for similar structures, an accurate protein structure embedding has a number of uses. Fast protein comparison is useful for clustering large sets of structures, for example to identify novel folds in the AlphaFold database [43], and Progres could be used to speed up such methods in combination with purpose-built libraries for similarity searching such as Faiss [45]. The embedding of a structure is just a set of numbers, and therefore can be targeted by differentiable approaches for applications like protein design. A decoder could be trained to generate structures from the embedding space [46, 47]. Properties of proteins such as evolution [48], topological classification

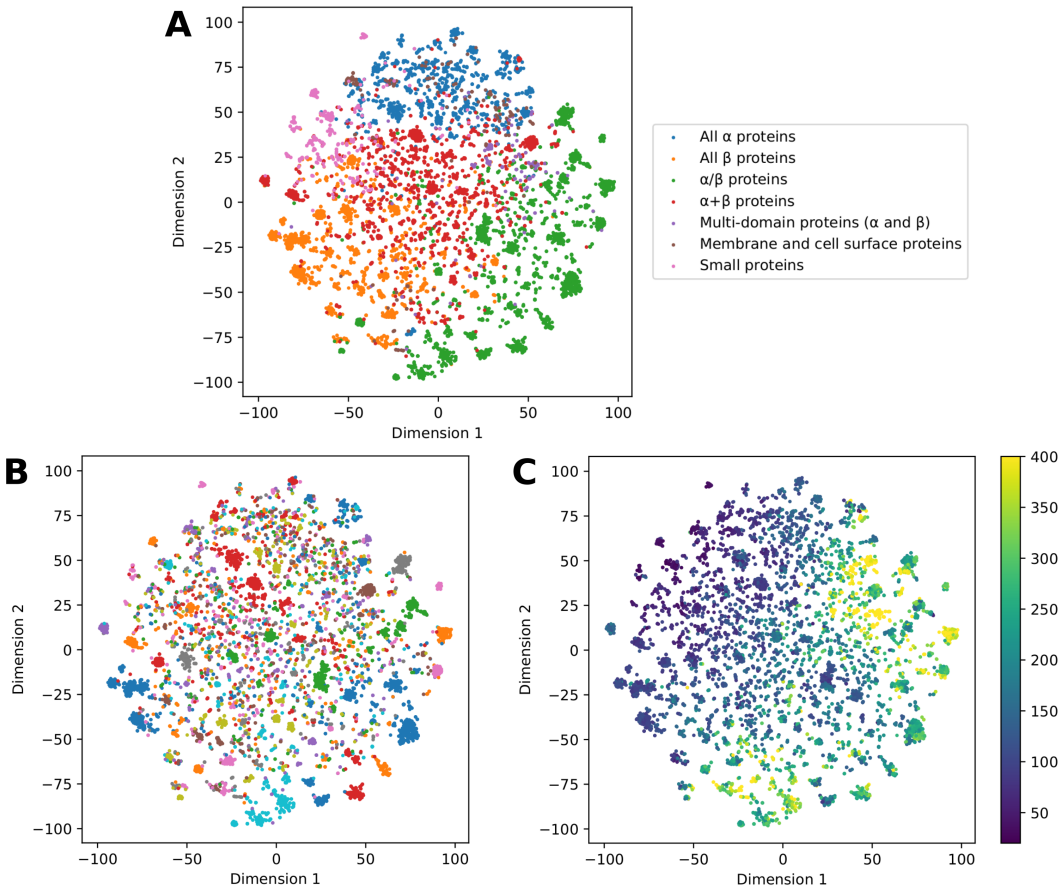

Figure 1: 2D t-SNE embedding of the 128 dimensions of our model embedding for the Astral set of SCOPe domains clustered at 40% sequence identity (15,177 domains). t-SNE was carried out with scikit-learn [42] using a perplexity value of 30. (A) Coloured by SCOPe class. (B) Coloured by SCOPe fold; there are 1,253 folds shown and colours are used multiple times. (C) Coloured by number of residues in the domain. The median length of domains is 149 residues. For colouring, the maximum number of residues in a domain is treated as 400.

[49], the completeness of protein fold space [50], the continuity of fold space [51], function [12] and dynamics could also be explored in the context of the low-dimensional fold space.

There are drawbacks to the method presented here: only domains can be searched, no residue alignment is returned, the model would ideally be retrained periodically to take new families into account, and exact mirror images of proteins give the same embedding. However, we believe that the extremely fast pairwise comparison allowed by structural embeddings is an effective way to take advantage of the opportunities provided by the million structure era.

Availability

A Python package allowing structure searching (against SCOPe, CATH and ECOD) and generation of pre-embedded databases is available under a permissive license at `https://github.com/jgreener64/progres`. Datasets and training scripts will be made available on publication.

Acknowledgements

We thank the UCL Bioinformatics Group for useful discussions and Jake Grimmett, Toby Darling and Ivan Clayson for help with high-performance computing. This work was supported by the Medical Research Council.

References

[1] H Hasegawa and L Holm. Advances and pitfalls of protein structural alignment. *Curr Opin Struct Biol*, 19(3):341–348, 2009.

[2] W R Taylor and C A Orengo. Protein structure alignment. *J Mol Biol*, 208(1):1–22, 1989.

[3] L Holm. Using Dali for Protein Structure Comparison. *Methods Mol Biol*, 2112:29–42, 2020.

[4] I N Shindyalov and P E Bourne. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng*, 11(9):739–747, 1998.

[5] Y Zhang and J Skolnick. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res*, 33(7):2302–2309, 2005.

[6] D Guzenko, S K Burley, and J M Duarte. Real time structural search of the Protein Data Bank. *PLoS Comput Biol*, 16(7):e1007970, 2020.

[7] T Aderinwale, V Bharadwaj, C Christoffer, G Terashi, Z Zhang, R Jahandideh, Y Kagaya, and D Kihara. Real-time structure search and structure classification for AlphaFold protein models. *Commun Biol*, 5(1):316, 2022.

[8] K Illergård, D H Ardell, and A Elofsson. Structure is three to ten times more conserved than sequence–a study of structural response in protein cores. *Proteins*, 77(3):499–508, 2009.

[9] I Sillitoe, N Bordin, N Dawson, V P Waman, P Ashford, H M Scholes, C S M Pang, L Woodridge, C Rauer, N Sen, M Abbasian, S Le Cornu, S D Lam, K Berka, I H Varekova, R Svobodova, J Lees, and C A Orengo. CATH: increased structural coverage of functional space. *Nucleic Acids Res*, 49(D1):D266–D273, 2021.

[10] C Zhang, P L Freddolino, and Y Zhang. COFACTOR: improved protein function prediction by combining structure, sequence and protein-protein interaction information. *Nucleic Acids Res*, 45(W1):W291–W299, 2017.

[11] S M Kandathil, J G Greener, A M Lau, and D T Jones. Ultrafast end-to-end protein structure prediction enables high-throughput exploration of uncharacterized proteins. *Proc Natl Acad Sci U S A*, 119(4), 2022.

[12] J K Leman, P Szczerbiak, P D Renfrew, V Gligorijevic, D Berenberg, T Vatanen, B C Taylor, C Chandler, S Janssen, A Pataki, N Carriero, I Fisk, R J Xavier, R Knight, R Bonneau, and T Kosciolek. Sequence-structure-function relationships in the microbial protein universe. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/04/27/2022.03.18.484903`.

[13] J Jumper, R Evans, A Pritzel, T Green, M Figurnov, O Ronneberger, K Tunyasuvunakool, R Bates, A Žídek, A Potapenko, A Bridgland, C Meyer, S A A Kohl, A J Ballard, A Cowie, B Romera-Paredes, S Nikolov, R Jain, J Adler, T Back, S Petersen, D Reiman, E Clancy, M Zielinski, M Steinegger, M Pacholska, T Berghammer, S Bodenstein, D Silver, O Vinyals, A W Senior, K Kavukcuoglu, P Kohli, and D Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature*, 596(7873):583–589, 2021.

[14] M Varadi, S Anyango, M Deshpande, S Nair, C Natassia, G Yordanova, D Yuan, O Stroe, G Wood, A Laydon, A Žídek, T Green, K Tunyasuvunakool, S Petersen, J Jumper, E Clancy, R Green, A Vora, M Lutfi, M Figurnov, A Cowie, N Hobbs, P Kohli, G Kleywegt, E Birney, D Hassabis, and S Velankar. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic Acids Res*, 50(D1):D439–D444, 2022.

[15] Z Lin, H Akin, R Rao, B Hie, Z Zhu, W Lu, A dos Santos Costa, M Fazel-Zarandi, T Sercu, S Candido, and A Rives. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/07/21/2022.07.20.500902`.

[16] M van Kempen, S S Kim, C Tumescheit, M Mirdita, J Söding, and M Steinegger. Foldseek: fast and accurate protein structure search. *bioRxiv*, 2022. URL https://www.biorxiv.org/content/early/2022/02/09/2022.02.07.479398.

[17] V G Satorras, E Hoogeboom, and M Welling. E(n) equivariant graph neural networks, 2021. URL https://arxiv.org/abs/2102.09844.

[18] J M Chandonia, N K Fox, and S E Brenner. SCOPe: classification of large macromolecular structures in the structural classification of proteins-extended database. *Nucleic Acids Res*, 47 (D1):D475–D481, 2019.

[19] H Cheng, R D Schaeffer, Y Liao, L N Kinch, J Pei, S Shi, B H Kim, and N V Grishin. ECOD: an evolutionary classification of protein domains. *PLoS Comput Biol*, 10(12):e1003926, 2014.

[20] P Khosla, P Teterwak, C Wang, A Sarna, Y Tian, P Isola, A Maschinot, C Liu, and D Krishnan. Supervised contrastive learning, 2020. URL https://arxiv.org/abs/2004.11362.

[21] C Chen, Y Zha, D Zhu, K Ning, and X Cui. Hydrogen bonds meet self-attention: all you need for protein structure embedding. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 12–17, 2021.

[22] A Villegas-Morcillo, V Sanchez, and A M Gomez. FoldHSphere: deep hyperspherical embeddings for protein fold recognition. *BMC Bioinformatics*, 22(490), 2021.

[23] J Durairaj, J Pereira, M Akdel, and T Schwede. What is hidden in the darkness? characterization of AlphaFold structural space. *bioRxiv*, 2022. URL https://www.biorxiv.org/content/early/2022/10/13/2022.10.11.511548.

[24] V Gligorijević, P D Renfrew, T Kosciolek, J K Leman, D Berenberg, T Vatanen, C Chandler, B C Taylor, I M Fisk, H Vlamakis, R J Xavier, R Knight, K Cho, and R Bonneau. Structure-based protein function prediction using graph convolutional networks. *Nat Commun*, 12(3168), 2021.

[25] Z Zhang, M Xu, A R Jamasb, V Chenthamarakshan, A Lozano, P Das, and J Tang. Protein representation learning by geometric structure pretraining. In *First Workshop on Pre-training: Perspectives, Pitfalls, and Paths Forward at ICML 2022*, 2022.

[26] P Hermosilla and T Ropinski. Contrastive representation learning for 3D protein structures, 2022. URL https://arxiv.org/abs/2205.15675.

[27] C Chen, J Zhou, F Wang, X Liu, and D Dou. Structure-aware protein self-supervised learning, 2022. URL https://arxiv.org/abs/2204.04213.

[28] M Heinzinger, M Littmann, I Sillitoe, N Bordin, C Orengo, and B Rost. Contrastive learning on protein embeddings enlightens midnight zone. *NAR Genomics and Bioinformatics*, 4(2), 2022. ISSN 2631-9268.

[29] J M Chandonia, G Hon, N S Walker, L Lo Conte, P Koehl, M Levitt, and S E Brenner. The ASTRAL Compendium in 2004. *Nucleic Acids Res*, 32:D189–D192, 2004.

[30] M Steinegger and J Söding. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol*, 35(11):1026–1028, 2017.

[31] P J Cock, T Antao, J T Chang, B A Chapman, C J Cox, A Dalke, I Friedberg, T Hamelryck, F Kauff, B Wilczynski, and M J de Hoon. Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, 25(11):1422–1423, 2009.

[32] J G Greener, J Selvaraj, and B J Ward. BioStructures.jl: read, write and manipulate macromolecular structures in Julia. *Bioinformatics*, 36(14):4206–4207, 2020.

[33] A Vaswani, N Shazeer, N Parmar, J Uszkoreit, L Jones, A N Gomez, L Kaiser, and I Polosukhin. Attention is all you need, 2017. URL https://arxiv.org/abs/1706.03762.

[34] A Paszke, S Gross, F Massa, A Lerer, J Bradbury, G Chanan, T Killeen, Z Lin, N Gimelshein, L Antiga, A Desmaison, A Kopf, E Yang, Z DeVito, M Raison, A Tejani, S Chilamkurthy, B Steiner, L Fang, J Bai, and S Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. 2019.

[35] D Hendrycks and K Gimpel. Gaussian error linear units (GELUs), 2016. URL `https://arxiv.org/abs/1606.08415`.

[36] D P Kingma and J Ba. Adam: A method for stochastic optimization, 2014. URL `https://arxiv.org/abs/1412.6980`.

[37] A Andreeva, E Kulesha, J Gough, and A G Murzin. The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Res*, 48(D1):D376–D382, 2020.

[38] L van der Maaten and G Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008.

[39] L Holm and C Sander. Touring protein fold space with Dali/FSSP. *Nucleic Acids Res*, 26(1): 316–319, 1998.

[40] R Kolodny, L Pereyaslavets, A O Samson, and M Levitt. On the universe of protein folds. *Annu Rev Biophys*, 42:559–582, 2013.

[41] S Nepomnyachiy, N Ben-Tal, and R Kolodny. Global view of the protein universe. *Proc Natl Acad Sci U S A*, 111(32):11691–11696, 2014.

[42] F Pedregosa, G Varoquaux, A Gramfort, V. Michel, B Thirion, O Grisel, M Blondel, P. Prettenhofer, R Weiss, V Dubourg, J Vanderplas, A Passos, D Cournapeau, M Brucher, M Perrot, and E Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[43] N Bordin, I Sillitoe, V Nallapareddy, C Rauer, S D Lam, V P Waman, N Sen, M Heinzinger, M Littmann, S Kim, S Velankar, M Steinegger, B Rost, and C Orengo. Alphafold2 reveals commonalities and novelties in protein structure space for 21 model organisms. *bioRxiv*, 2022. URL `https://www.biorxiv.org/content/early/2022/06/03/2022.06.02.494367`.

[44] W Zheng, X Zhou, Q Wuyun, R Pearce, Y Li, and Y Zhang. FUpred: detecting protein domains through deep-learning-based contact map prediction. *Bioinformatics*, 36(12):3749–3757, 2020.

[45] J Johnson, M Douze, and H Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019.

[46] X Guo, Y Du, S Tadepalli, L Zhao, and A Shehu. Generating tertiary protein structures via an interpretative variational autoencoder, 2020. URL `https://arxiv.org/abs/2004.07119`.

[47] J Ingraham, V K Garg, R Barzilay, and T Jaakkola. Generative models for graph-based protein design. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2019.

[48] X Ding, Z Zou, and C L Brooks III. Deciphering protein evolution and fitness landscapes with latent space models. *Nat Commun*, 10(1):5644, 2019.

[49] W R Taylor. A 'periodic table' for protein structures. *Nature*, 416(6881):657–660, 2002.

[50] P Cossio, A Trovato, F Pietrucci, F Seno, A Maritan, and A Laio. Exploring the universe of protein structures beyond the Protein Data Bank. *PLoS Comput Biol*, 6(11):e1000957, 2010.

[51] J Skolnick, A K Arakaki, S Y Lee, and M Brylinski. The continuity of protein structure space is an intrinsic property of proteins. *Proc Natl Acad Sci U S A*, 106(37):15690–15695, 2009.