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. Fast and accurate methods to search with structures will be essential to make use of the vast databases that have recently become available, 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 method, called Progres, is available at `https://github.com/greener-group/progres`. It has accuracy comparable to the best current methods and can search the AlphaFold database TED domains in a tenth of a second per query on CPU.

Introduction

A variety of methods have been developed to compare, align and search with protein structures [1] including comparing residue-residue distances [2, 3, 4], considering local geometry [5], coordinate alignment [6] and 3D Zernike descriptors [7, 8]. Since protein structure is more conserved than sequence [9] these methods have proved useful in remote homology detection [10], protein classification [11], inferring function from structure [12], clustering large databases [13, 14] 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 such as the AlphaFold Protein Structure Database [15, 16] or the ESM Metagenomic Atlas [17] with these methods is slow. Recently Foldseek [18] 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 [19], 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 [20, 11, 21]; we use supervised contrastive learning [22] 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.

A number of recent methods have used protein structure graph embeddings [23, 24] and contrastive learning [25, 26]. Embedding protein folds has also been done using residue-level features [27, 14], and GNNs acting on protein structure have been used for function prediction [28]. 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 [29, 30, 31]. Contrastive learning using protein classifications has also improved language models for protein sequences, showing clustering that better preserves protein structure space [32]. Protein structure has been incorporated into language models more broadly, often with the intention of searching for remote homology [33, 34, 35, 36, 37, 38].

Results

We trained a simple GNN, called Progres (PROtein GRaph Embedding Search), to embed a protein structure independent of its sequence (see Figure 1A). Since we use distance and torsion angle features based on coordinates the embedding is SE(3)-invariant, i.e. it doesn't change with translation or rotation of the input structure. As shown in Figure 1B, supervised contrastive learning [22] on SCOPe domains [20, 39] 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 [40] 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. 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. The impact of changing model hyperparameters is shown in Table S1. The distribution of values across the embedding dimensions is shown in Figure S1.

In order to assess the accuracy of the model for structure searching, we follow a similar procedure to Foldseek [18]. 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 may 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] and Foldseek-TM [18] for recovering domains in SCOPe from the same fold, superfamily and family. Its strong performance at the fold level indicates an ability to find remote homologs. Progres is more sensitive than the EAT [32] and ESM-2 [17] protein language model embeddings, and also the baseline sequence searching method of MMseqs2 [41]. This indicates the benefits of comparing structures rather than just sequences for detecting homology. Figure 2A-C shows the performance across different SCOPe classes, protein sizes and contact orders. Progres does particularly well on all β domains, smaller domains and domains with higher contact order. It has lower performance on membrane proteins and larger domains. As shown in Figure 2D, performance drops of significantly when the number of embedding dimensions is below 32.

For searching a single structure against SCOPe on CPU the model is faster than Foldseek with most run time in Python module loading. For example, going from 1 to 100 query structures increases run time from 1.3 s to 2.4 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

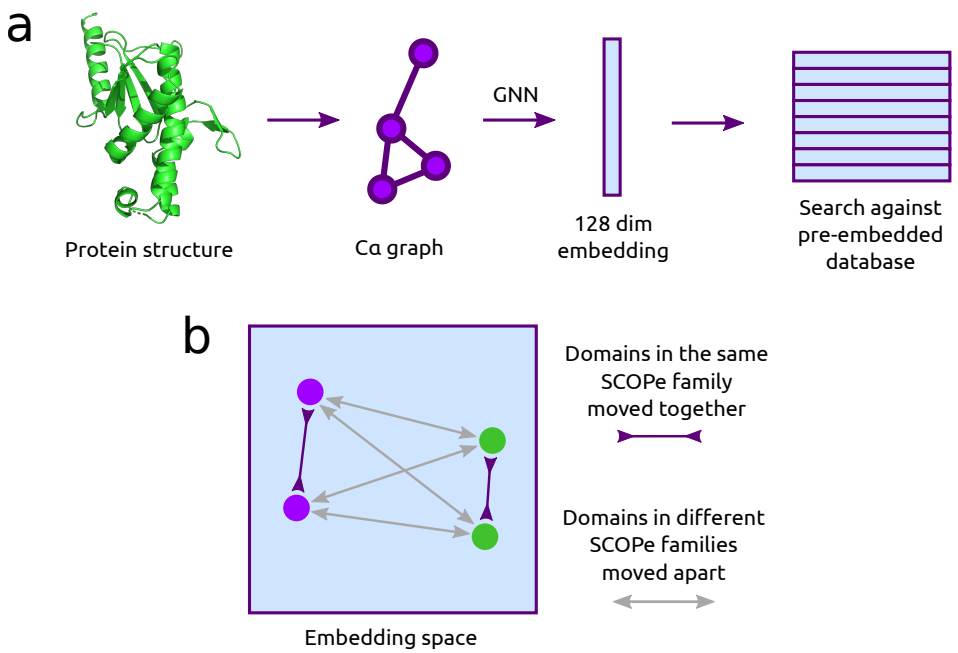

Figure 1 Protein structure embedding. (A) Protein domains are treated as a graph with Cα atoms as nodes and edges between Cα atoms within 10 Å. A GNN embeds the graph into a 128-dimensional representation. This can be compared quickly to a pre-embedded search database. (B) Supervised contrastive learning [22] is used to train the model, with embeddings for domains in the same SCOPe family pushed together and embeddings for domains in different SCOPe families pushed apart.

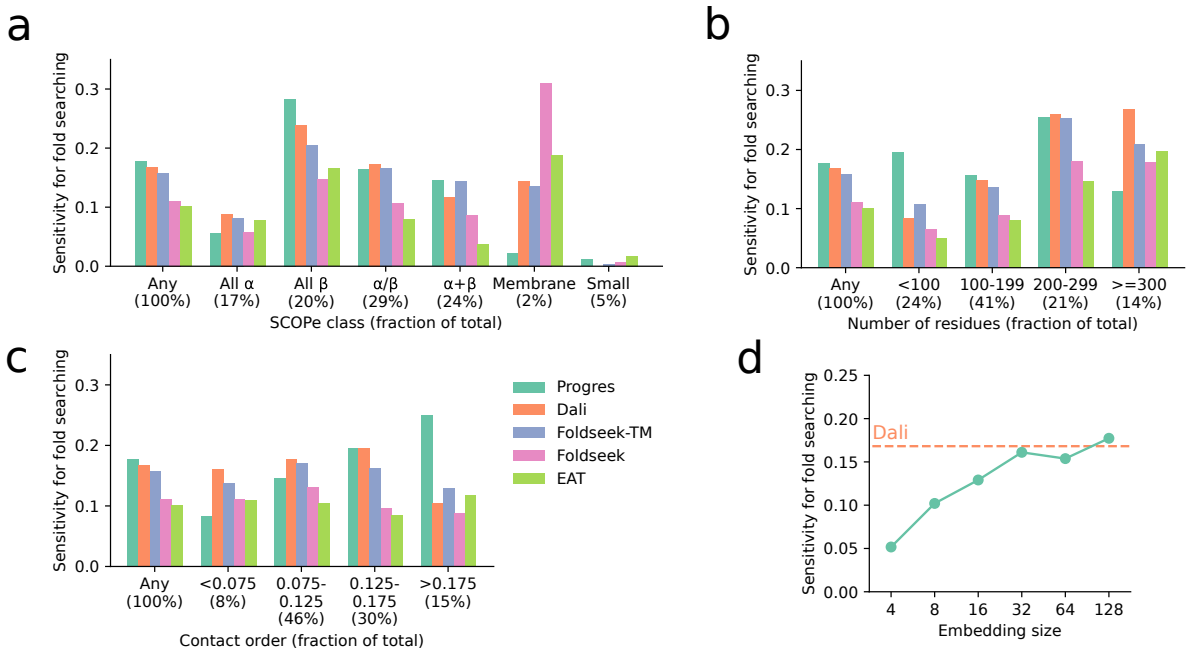

Figure 2 Model performance on different protein types. In each case the "Any" category is the same as in Table 1. (A) Sensitivity for fold searching by SCOPe class. (B) Sensitivity for fold searching by protein sequence length. (C) Sensitivity for fold searching by contact order, a measure of the sequence separation of contacting residues. (D) Sensitivity for fold searching across different embedding sizes. A model was trained from scratch for each embedding size. See Table S1 for further ablations.

Software	Fold	Superfamily	Family	Run time (single)	Run time (all-v-all)
Progres (this work)	0.177	0.706	0.877	1.3 s (CPU)	163 s (CPU), 127 s (GPU)
Dali [3]	0.168	0.709	0.885	508 s	> 1 month
Foldseek-TM [18]	0.158	0.666	0.859	4.8 s	2 h 47 m
Foldseek [18]	0.111	0.644	0.850	2.3 s	250 s
TM-align fast [6]	0.100	0.594	0.806	390 s	∼23 days
3D-SURFER [42]	0.046	0.140	0.349	7.2 s	24 h
EAT [32]	0.101	0.615	0.843	34 s (GPU)	4 h 37 m (GPU)
ESM-2 [17]	0.014	0.221	0.477	28 s (GPU)	590 s (GPU)
MMseqs2 [41]	0.001	0.165	0.433	0.9 s	17.1 s

Table 1 Comparison of ability to retrieve homologous proteins from SCOPe. A similar procedure to Foldseek [18] is followed with a set of 400 domains. For each domain the fraction of true positives (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-prepared. Run time (all-v-all) is the time taken to calculate all pairwise distances between the 15,177 domains from structure. EAT, ESM-2 and MMseqs2 use sequence not structure for searching. 3D-SURFER and EAT are trained with structural information and may have seen proteins in the test set during training.

structure or structures against the pre-computed embeddings of a huge database such as the AlphaFold database [8, 13, 14]. The recent TED study split the whole AlphaFold database into domains using a consensus-based approach [43]. We embed the TED domains clustered at 50% sequence identity and use FAISS [44] to considerably speed up the search time against the resultant database of 53 million structures. This allows a search time of a tenth of a second per query on CPU, after an initial data loading time of around a minute. Since we search exhaustively with FAISS, the results are not changed, though the approximate score calculation means the similarity score does vary slightly from the exact value. For the SCOPe test set used above, the mean difference between FAISS and exact similarity scores for the top hit is 0.006. As shown in Figure S2, the best TM-align score to the query among the top 5 hits has a mean of 0.80 across the SCOPe test set, with 94% being over 0.5. This indicates that searching is accurate even when using a large database.

Figure 3 shows 2D t-SNE embeddings [45] of the 128 dimensions of our model embedding. This shows the lower-dimensional protein fold space [46, 47, 48] 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. There is a clear protein size gradient across the t-SNE embedding. A t-SNE embedding for the AlphaFold database TED domains compared to ECOD [21] and the AlphaFold 21 organisms set [16, 49] shows the volume of new structural information available in the AlphaFold database. The Progres score between two embeddings is the cosine similarity score normalised to run between 0 and 1, with 1 indicating identical embeddings. As shown in Figure 3E a Progres score of 0.8 indicates that two proteins share the same fold, analogous to a TM-align score of 0.5.

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

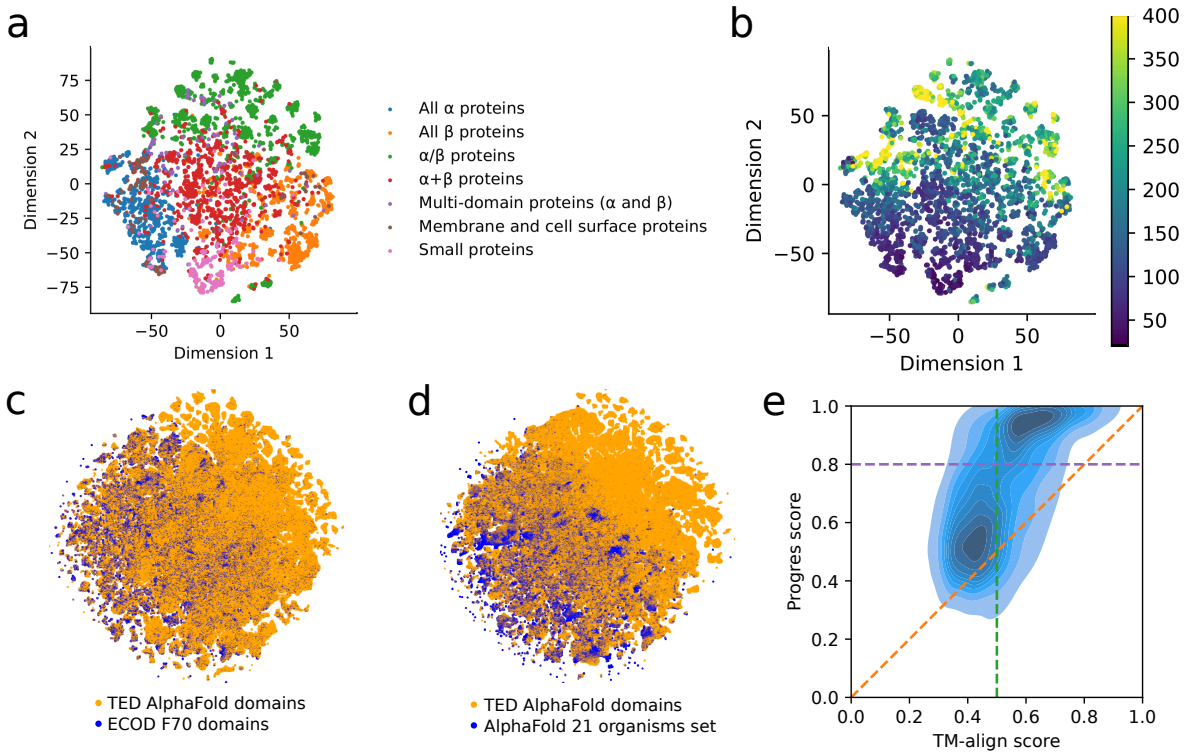

Figure 3 Exploring Progres embeddings. (A) 2D t-SNE embedding of the 128 dimensions of our model embedding for the Astral set of SCOPe domains clustered at 40% sequence identity (15,177 domains). The domains are coloured by SCOPe class. t-SNE was carried out using a perplexity value of 30. (B) The same data 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. (C) 2D t-SNE of the AlphaFold database TED domains [43] clustered at 50% sequence identity and the ECOD F70 set of domains in the PDB [21]. 5m (9%) of the TED domains are chosen randomly for the t-SNE for computational reasons. (D) A similar comparison of the TED domains to the AlphaFold 21 model organisms set [16, 49]. (E) Comparison of Progres score to TM-align score. For each of the 400 domains in the test set the top 200 matches in the Astral 40% sequence identity set according to TM-align are considered. The Pearson correlation coefficient is 0.60. The green line shows the TM-align score threshold of 0.5 indicating the same fold. The purple line shows the Progres score threshold of 0.8 indicating the same fold.

multiple domains, long disordered regions or complexes. Fortunately, there are a number of tools such as Merizo [50], SWORD2 [51] and Chainsaw [52] that can split query structures into domains. 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. As shown in Figure S3 the Progres embeddings are fairly robust to truncating residues from the termini, with truncations of 20 residues giving an embedding with a similarity of 0.8 to the full length domain embedding for 89% of domains with 200-299 residues. This means that minor inaccuracies in predicting domain boundaries are unlikely to cause a problem.

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 fold, superfamily and family sensitivities analogous to Table 1 are 0.190, 0.383 and 0.546 respectively. This indicates similar performance at finding distantly related folds, the main use of structure searching over sequence searching, though there is a drop in performance at finding closely-related domains.

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 [13, 14, 43]. 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 [53, 54], and a diffusion model to move through the embedding space. Properties of proteins such as evolution [55], topological classification [56], the completeness of protein fold space [57], the continuity of fold space [58], function [59] and dynamics could also be explored in the context of the low-dimensional fold space. Structure embeddings could also be used to identify regions of unknown density in cryo-electron tomography studies. 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.

Methods

Training

Structures in the Astral 2.08 95% sequence identity set including discontinuous domains were used for training [60]. 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 [41], 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 [61]. Some processing was also carried out with BioStructures.jl [62]. 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 [40].

PyTorch was used for training [63]. The neural network architecture was similar to the E(n)-equivariant GNN in Satorras et al. 2021 [19]. 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 analogous to a standard GNN with relative squared norms inputted to the edge operation [19]. Edges are sparse and are between Cα atoms within 10 Å of each other. 6 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 [64], 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 [22] 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 present 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 [22]. 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 [65] 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 [18]. The 15,177 Astral 2.08 40% sequence identity set domains were embedded with the model. The embeddings are stored as Float16 to reduce the size of large databases on disk, but this has no effect on search performance as shown in Table S1. 400 of these domains were chosen randomly and held out of the training data as described previously. Like Foldseek, we only chose domains with at least one other family, superfamily and fold member. For each of these 400 domains, the cosine similarity of embeddings to each of the 15,177 domains was calculated and the domains ranked by similarity with the query domain included. For each domain, we measured the fraction of 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.

All CPU methods were run on an Intel i9-10980XE CPU and with 256 GB RAM. Progres, Foldseek and MMseqs2 were run on 16 threads. The GPU methods were run on a RTX A6000 GPU. Progres was run with PyTorch 1.11. Foldseek version 8.ef4e960 was used. For TM-align we used the fast mode, which has similar performance to the normal mode [18]. For 3D-SURFER the neural network model and mainchain atoms were used. EAT was run with the "–use_tucker 1" flag. 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. For MMseqs2, easy-search with a sensitivity of 7.5 was used.

For contact order, all residue pairs with Cβ atoms (Cα for glycine) within 8 Å are considered. The contact order of a structure is then defined as

$$\frac{\sum_{i}^{N} S_i}{LN}$$

where S_i is the sequence separation of the residues in contacting pair i, N is the number of contacting pairs and L is the sequence length of the protein.

Databases

The AlphaFold database domain embeddings were prepared from the TED set of domains [43] using cluster representatives from clustering at 50% sequence identity. Clustering was carried out with MMseqs2 using the command "mmseqs easy-cluster ted_100.fasta clusterRes tmp –min-seq-id 0.5 -c

0.9 –cov-mode 5 -s 7.5". This gave 53,344,209 clusters, fewer than the TED analysis due to the use of easy-cluster over easy-linclust. The FAISS [44] index was prepared using "IndexFlatIP(128)", which carries out exhaustive searching using the same cosine similarity as Progres.

Availability

A Python package allowing structure searching and generation of pre-embedded databases is available along with datasets, training scripts and benchmarking scripts under a permissive license at `https://github.com/greener-group/progres`. The trained model and pre-embedded databases are available at `https://zenodo.org/record/7782088`.

Acknowledgements

We thank the UCL Bioinformatics Group for useful discussions and Jake Grimmett, Toby Darling and Ivan Clayson for help with high-performance computing. KJ is a PhD student in the group of Sjors Scheres. This work was supported by the Medical Research Council, as part of United Kingdom Research and Innovation (also known as UK Research and Innovation) [MC_UP_1201/33 to JGG and MC_UP_A025-1013 to SS]. For the purpose of open access, the MRC Laboratory of Molecular Biology has applied a CC BY public copyright licence to any Author Accepted Manuscript version arising.

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] Z Guo, Y Wang, and G Ou. Utilizing the Scale-Invariant feature transform algorithm to align distance matrices facilitates systematic protein structure comparison. *Bioinformatics*, 40(3):btae064, 2024.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] A Al-Fatlawi, M Menzel, and M Schroeder. Is Protein BLAST a thing of the past? *Nat Commun*, 14:8195, 2023.

[11] 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.

[12] 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.

[13] I Barrio-Hernandez, J Yeo, J Jänes, M Mirdita, C L M Gilchrist, T Wein, M Varadi, S Velankar, P Beltrao, and M Steinegger. Clustering predicted structures at the scale of the known protein universe. *Nature*, 622:637–645, 2023.

[14] J Durairaj, A M Waterhouse, T Mets, T Brodiazhenko, M Abdullah, G Studer, G Tauriello, M Akdel, A Andreeva, A Bateman, T Tenson, V Hauryliuk, T Schwede, and J Pereira. Uncovering new families and folds in the natural protein universe. *Nature*, 622:646–653, 2023.

[15] 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.

[16] 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.

[17] 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`.

[18] M van Kempen, S S Kim, C Tumescheit, M Mirdita, J Lee, C L M Gilchrist, J Söding, and M Steinegger. Fast and accurate protein structure search with Foldseek. *Nature Biotechnology*, 42:243–246, 2023.

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

[20] 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.

[21] 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.

[22] 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`.

[23] S M Kandathil, A M Lau, and D T Jones. Foldclass and Merizo-search: embedding-based deep learning tools for protein domain segmentation, fold recognition and comparison. *bioRxiv*, 2024. URL `https://www.biorxiv.org/content/early/2024/03/29/2024.03.25.586696`.

[24] 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.

[25] J Luo and Y Luo. Contrastive learning of protein representations with graph neural networks for structural and functional annotations. *Pac Symp Biocomput*, 28:109–120, 2023.

[26] T Hamamsy, M Barot, J T Morton, M Steinegger, R Bonneau, and K Cho. Learning sequence, structure, and function representations of proteins with language models. *bioRxiv*, 2023. URL `https://www.biorxiv.org/content/early/2023/11/26/2023.11.26.568742`.

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

[28] 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.

[29] 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.

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

[31] 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`.

[32] 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):lqac043, 2022.

[33] T Hamamsy, J T Morton, R Blackwell, D Berenberg, N Carriero, V Gligorijevic, C E M Strauss, J K Leman, K Cho, and R Bonneau. Protein remote homology detection and structural alignment using deep learning. *Nat Biotechnol*, 2023.

[34] W Liu, Z Wang, R You, C Xie, H Wei, Y Xiong, J Yang, and S Zhu. PLMSearch: Protein language model powers accurate and fast sequence search for remote homology. *Nat Commun*, 15:2775, 2024.

[35] Y Liu and H-B Shen. Foldexplorer: Fast and accurate protein structure search with sequence-enhanced graph embedding, 2023.

[36] J Zheng, G Wang, Y Huang, B Hu, S Li, C Tan, X Fan, and S Z Li. Lightweight contrastive protein structure-sequence transformation, 2023. URL https://arxiv.org/abs/2303.11783.

[37] B G Iovino, H Tang, and Y Ye. Protein domain embeddings for fast and accurate similarity search. *bioRxiv*, 2023. URL https://www.biorxiv.org/content/early/2023/11/27/2023.11.27.567555.

[38] M Heinzinger, K Weissenow, J G Sanchez, A Henkel, M Steinegger, and B Rost. Prostt5: Bilingual language model for protein sequence and structure. *bioRxiv*, 2023. URL https://www.biorxiv.org/content/early/2023/07/25/2023.07.23.550085.

[39] 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.

[40] 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.

[41] 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.

[42] Y Xiong, J Esquivel-Rodriguez, L Sael, and D Kihara. 3D-SURFER 2.0: Web Platform for Real-Time Search and Characterization of Protein Surfaces. *Methods Mol Biol*, 1137:105–117, 2014.

[43] A M Lau, N Bordin, S M Kandathil, I Sillitoe, V P Waman, J Wells, C A Orengo, and D T Jones. Exploring structural diversity across the protein universe with the encyclopedia of domains. *bioRxiv*, 2024. URL https://www.biorxiv.org/content/early/2024/03/27/2024.03.18.585509.

[44] M Douze, A Guzhva, C Deng, J Johnson, G Szilvasy, P-E Mazaré, M Lomeli, L Hosseini, and H Jégou. The Faiss library, 2024. URL https://arxiv.org/abs/2401.08281.

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

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

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

[48] 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.

[49] 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. *Commun Biol*, 6(1):160, 2023.

[50] A M Lau, S M Kandathil, and D T Jones. Merizo: a rapid and accurate protein domain segmentation method using invariant point attention. *Nat Commun*, 14:8445, 2023.

[51] G Cretin, T Galochkina, Y Vander Meersche, A G de Brevern, G Postic, and J-C Gelly. SWORD2: hierarchical analysis of protein 3D structures. *Nucleic Acids Research*, 50(W1):W732–W738, 2022.

[52] J Wells, A Hawkins-Hooker, N Bordin, B Paige, and C Orengo. Chainsaw: protein domain segmentation with fully convolutional neural networks. *bioRxiv*, 2023. URL https://www.biorxiv.org/content/early/2023/07/19/2023.07.19.549732.

[53] 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.

[54] 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.

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

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

[57] 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.

[58] 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.

[59] 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`.

[60] 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.

[61] 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.

[62] 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.

[63] 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.

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

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

Fast protein structure searching using structure graph embeddings

Joe G Greener and Kiarash Jamali

Supplementary Data

Ablation	**Fold**	**Superfamily**	**Family**
No ablation	0.177	0.706	0.877
64 embedding dimensions	0.154	0.673	0.864
32 embedding dimensions	0.161	0.699	0.884
16 embedding dimensions	0.129	0.654	0.850
8 embedding dimensions	0.102	0.491	0.696
4 embedding dimensions	0.052	0.181	0.353
12 layers	0.150	0.709	0.902
4 layers	0.148	0.645	0.841
No τ angle node feature	0.159	0.690	0.869
No Gaussian noise	0.072	0.516	0.738
No sinusoidal positional encoding	0.052	0.402	0.642
Float32 database for searching	0.177	0.706	0.877
Superfamily holdout validation	0.190	0.383	0.546

Table S1 Model ablations. In each case a model is trained from scratch and the same testing procedure is used as in Table 1. No ablation is the model presented in the results with 128 embedding dimensions and 6 layers. The embedding dimension ablation is shown visually in Figure 2D. Removing the τ angle node feature means the network is E(3)-invariant not SE(3)-invariant. Superfamily holdout validation is described in the discussion.

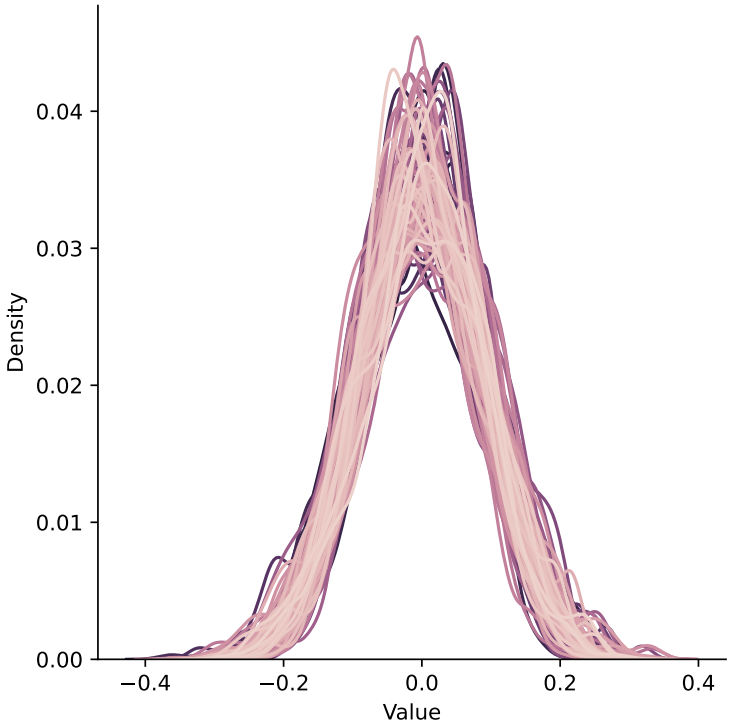

Figure S1 Values of embedding dimensions. Each protein in the Astral 40% sequence identity set is embedded with Progres and the distribution of values in each of the 128 dimensions is shown.

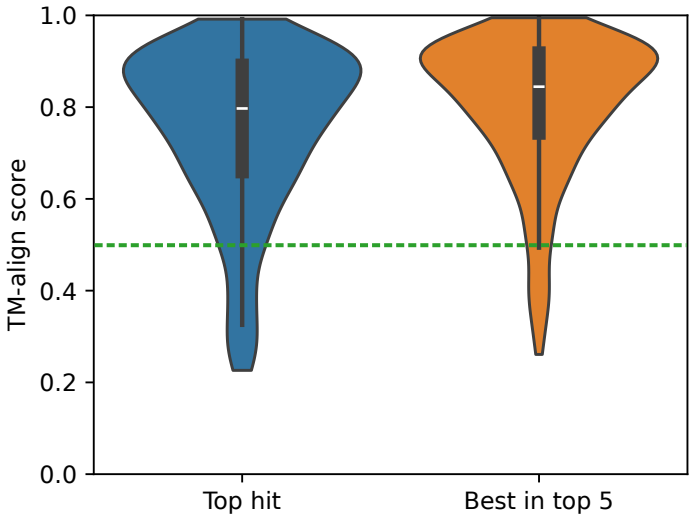

Figure S2 Searching SCOPe domains against the AlphaFold database. Each of the 400 domains in the test set is searched against the AlphaFold database TED domains clustered at 50% sequence identity (53 million domains) and the TM-align score between the top hit and the query is calculated. This gives a mean TM-align score of 0.75 with 90% above 0.5. The highest TM-align score among the top 5 hits is also calculated and gives a mean of 0.80 with 94% above 0.5.

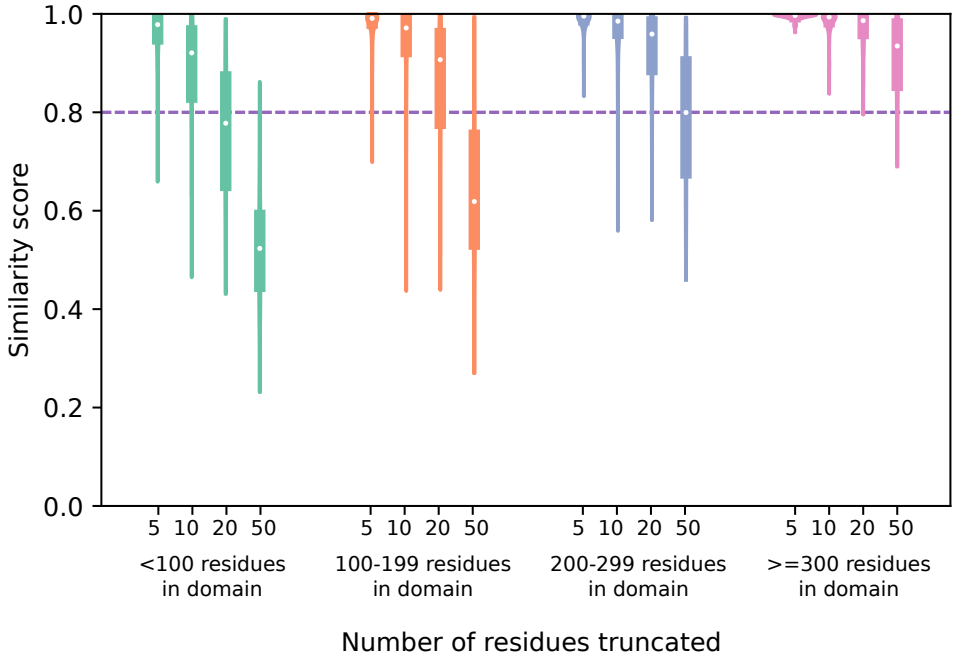

Figure S3 The effect of truncating domains on Progres embeddings. For each of the 400 domains in the SCOPe test set a number of residues were removed from the N-terminus or the C-terminus and the truncated domain was embedded. The Progres similarity score to the full length domain was then computed. The results are categorised by the number of residues in the full length domain. The line shows a similarity score of 0.8, indicating the same fold (see Figure 3E).