Contrastive learning on protein embeddings enlightens midnight zone at lightning speed

Michael Heinzinger[1,2], Maria Littmann[1], Ian Sillitoe[3], Nicola Bordin[3], Christine Orengo[3] & Burkhard Rost[1, 4]

1 TUM (Technical University of Munich) Dept Informatics, Bioinformatics & Computational Biology - i12, Boltzmannstr. 3, 85748 Garching/Munich, Germany
2 TUM Graduate School, Center of Doctoral Studies in Informatics and its Applications (CeDoSIA), Boltzmannstr. 11, 85748 Garching, Germany
3 Institute of Structural and Molecular Biology, University College London, London WC1E 6BT, UK
4 Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching, Germany & TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany
* Corresponding author: mheinzinger@rostlab.org, http://www.rostlab.org/ Tel: +49-289-17-811 (email rost: assistant@rostlab.org)

Abstract

Thanks to the recent advances in protein three-dimensional (3D) structure prediction, in particular through *AlphaFold 2 and RoseTTAFold*, the abundance of protein 3D information will explode over the next year(s). Expert resources based on 3D structures such as SCOP and CATH have been organizing the complex sequence-structure-function relations into a hierarchical classification schema. Experimental structures are leveraged through multiple sequence alignments, or more generally through homology-based inference (HBI) transferring annotations from a protein with experimentally known annotation to a query without annotation. Here, we presented a novel approach that expands the concept of HBI from a low-dimensional sequence-distance lookup to the level of a high-dimensional embedding-based annotation transfer (EAT). Secondly, we introduced a novel solution using single protein sequence representations from protein Language Models (pLMs), so called embeddings (Prose, ESM-1b, ProtBERT, and ProtT5), as input to contrastive learning, by which a new set of embeddings was created that optimized constraints captured by hierarchical classifications of protein 3D structures. These new embeddings (dubbed ProtTucker) clearly improved what was historically referred to as threading or fold recognition. Thereby, the new embeddings enabled the intrusion into the midnight zone of protein comparisons, i.e., the region in which the level of pairwise sequence similarity is akin of random relations and therefore is hard to navigate by HBI methods. Cautious benchmarking showed that ProtTucker reached much further than advanced sequence comparisons without the need to compute alignments allowing it to be orders of magnitude faster. Code is available at https://github.com/Rostlab/EAT .

Key words: protein structure classification, protein language model, self-supervised learning, contrastive learning, CATH.

Abbreviations used: **3D**, three-dimensional; **BFD**, Big Fantastic Database (11); **CATH**, hierarchical classification of protein 3D structures in Class, Architecture, Topology and Homologous superfamily (4,12); **EAT**, Embedding-based Annotation Transfer; **ESM-1b**, pLM from Facebook dubbed Evolutionary Scale Modeling (7); **FNN**, Feed-forward Neural Network; **FunFams**, functional families as sub-classification of the most fine-grained H level in CATH (13); **HBI**, Homology Based Inference; **HMM**, Hidden Markov Model; **HMMer**, particular method for HMM-profile alignments (6); **LM**, Language Model; **MMseqs2**, fast database search and multiple sequence alignment method (10); **MSA**, Multiple Sequence Alignment; **NLP**, Natural Language Processing; **PDB**, Protein Data Bank; **pLM**, protein Language Model; **ProSE**, pLM based on long short-term memory (LSTM) cells dubbed Protein Sequence Embeddings (1); **ProtBERT**, pLM based on the LM BERT (14); **ProtT5**, pLM based on the LM T5 (15).

Introduction

Phase-transition from daylight through twilight into midnight zone. Protein sequence determines structure determines function. This simple chain underlies the success of grouping proteins into families from sequence (3,16-22). Information from experimental high-resolution three-dimensional (3D) structures expands the perspective from families to super-families (12,23-25) that often reveal evolutionary and functional connections not recognizable from sequence alone (26,27). Thus, 3D helps penetrating through the twilight zone of sequence alignments (28,29) into the midnight zone (30).

The transition from daylight, through twilight to midnight zone is characterized by a phase-transition, i.e., a sigmoid function describing an order of magnitude increase in recall (relations identified/identifiable) and decrease in precision (relations correctly identified/all identified) over a narrow range of sequence similarity. For instance, for the HSSP-value (HVAL) (29,31): in the daylight zone for HVAL>5 (corresponding to >25% pairwise sequence identity – PIDE - for >250 aligned residues), over 90% of all pairs of proteins have largely similar 3D structures, while at the beginning of the midnight zone for HVAL<-5 (<15% PIDE for >250 aligned residues), over 90% have different 3D structures. Thus, the transition from daylight to midnight zone is described by a phase-transition in which over about ten percentage points in PIDE precision drops from 90% to 10%, i.e., from almost *all correct* to almost *all incorrect* within ±5 points PIDE.

The particular details at which point of sequence similarity the twilight zone begins and how extreme the transition, depend on the phenotype: more dramatic at lower PIDE for structure (29) and less dramatic at higher PIDE for function (32 ,33,34).

Most pairs of proteins with similar structure populate the midnight zone, i.e. they have very little sequence similarity (30). Therefore, the potential gain from pushing the threshold for comparisons is very high: since most discoveries (here: *inferring from sequence that two proteins have similar structure*) are in the midnight zone which is reached by a phase-transition, moving the threshold T from what users perceive as best to a slightly smaller number T-ε could have big impact (e.g., 10-fold for ε=5 for structure). In fact, any push a little lower WILL bring discoveries, mostly at the expense of even more false positives (here: proteins with dissimilar structure).

This simple reality has been driving the advance of methods using sequence similarity to establish relations: from advanced pairwise comparisons (35-38) over sequence-profile (39-42) to profile-profile comparisons (27,43-48) or efficient shortcuts to the latter (10,49,50). All those methods share one simple idea, namely, to use evolutionary information (EI) proxied by varying degrees of detectable sequence similarity to create families of related proteins. These are summarized in multiple sequence alignments (MSAs). Using such information as input to machine learning methods has been generating essentially all state-of-the-art (SOTA) prediction methods for almost three decades (51-54), and has also been one major key behind the breakthrough in protein structure prediction through *AlphaFold 2* (55) (recently this success was rudimentarily re-engineered in RoseTTAFold (56)). More recently, transfer- or representation-learning offer a novel route toward single sequence comparisons.

Embeddings capture language of life written in proteins. Every year algorithms improve natural language processing (NLP), particularly by feeding large text corpora into Deep Learning (DL) based Language Models (LMs) (14,15,57,58). These advances have been transferred to proteins through protein Language Models (pLMs) equating amino acids with words in NLP and the sequence of entire proteins with sentences. Such pLMs learn to predict masked or missing amino acids using large databases of raw protein sequences as input (5,7,59-63), or by refining the pLM through another supervised task (1,64). Processing the information learned by the pLM, e.g., by constructing vector representations from the last hidden layers of the networks forming the pLMs, yields a representation of protein sequences referred to as embeddings (Fig. 1 in (5)). Embeddings have been used successfully as exclusive input to predicting secondary structure and subcellular localization at performance levels almost reaching (7,59,60,65,66) or even exceeding (5,67,68) the SOTA using evolutionary information from MSAs as input. Embeddings can even substitute sequence similarity for homology-based annotation transfer (69,70). The power of such embeddings has been increasing with the advance of algorithms and the growth of data (5). Naturally, there will be some limit to such improvements. However, the recent advances prove that this limit has not nearly been reached when writing this (as of October 2021).

Embeddings from pLMs capture a diversity of higher-level features of proteins, including various aspects of protein function and structure (5,7,60,69-74). In fact, pLMs such as ProtT5 (5) or ESM-1b (7) capture aspects about protein structure so impressively that inter-residue distances – and consequently 3D structure – can be predicted without using MSAs, even with shallow (few free parameters) Deep Learning (DL) architectures (66).

Supervised learning directly maps the input to the class output. Instead, contrastive learning (75-77), optimizes a new embedding space in which similar samples are pushed closer, dissimilar samples farther apart. Consequently, contrastive learning relies solely on the similarity between pairs (or triplets) of samples instead of on class label. This orthogonal perspective – defining similarity in embedding rather than sequence space combined with contrastive learning - led us to hypothesize that we might find structurally and functionally consistent sub-groups within protein families from raw sequences. As a proof-of-principle, we established that clustering FunFams (4,78) benefits from using contrastive learning on embeddings (70). The benefit of optimizing embeddings specifically for protein fold detection as defined by SCOPe (79) has recently been shown (73). Other approaches for fold prediction aim at learning fold-specific motifs (80) or pairwise similarity scores (81) via DL. However, most of the top-performing approaches for fold detection rely on information extracted from MSAs (82) and do not utilize the transfer-learning capabilities offered by recent pLMs.

Here, we expanded on the hypothesis that replacing supervised learning by contrastive learning ideally fits the hierarchical structure of CATH. We propose an approach that marries both, self-supervised pretraining and contrastive learning, by representing protein sequences as embeddings (fixed-size vectors derived from pre-trained pLMs), and using increasing overlap in the CATH hierarchy as a notion of increasing structural similarity to contrastively learn a new embedding space. We used the pLM ProtT5 (5) as static feature encoder (no fine-tuning of the pLM) to retrieve initial embeddings that were then mapped by a feed-forward neural network (FNN) to a new, learned embedding space optimized on CATH through contrastive learning. More specifically, the Soft Margin Loss was used with triplets of proteins (anchor, positive, and negative) to optimize the new embedding space toward maximizing the distance between proteins from different CATH classes (anchor-negative pairs) while minimizing the distance between proteins in the same CATH class (anchor-positive pairs). Triplets of varying structural similarity were used simultaneously to optimize a single, shared network: all four CATH-levels were simultaneously learned by one FNN. The resulting embeddings were dubbed ProtTucker and were established to identify more distant relations than is possible from sequence alone. One important objective of ProtTucker is to contribute toward studying entire functional modules through the identification of more distant relations as has been established to be crucial for capturing mimicry and hijacking of SARS-CoV-2 (83).

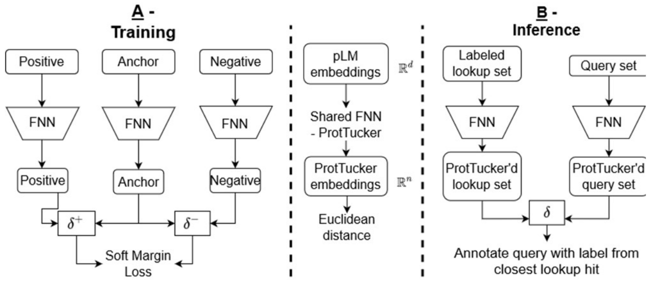

Fig. 1: Sketch of ProtTucker. Panel A illustrates how protein triplets were used to contrastively learn the CATH hierarchy. First, embeddings from protein Language Models (pLMs; here: ProSE (1), ESM-1b (7), ProtBERT (5), and ProtT5 (5)) were used as static feature encoders for protein triplets (anchor, positive, negative). The embedding of each protein was processed separately by the same, shared feed-forward neural network (FNN with hard parameter sharing; this FNNs was dubbed ProtTucker) resulting in a newly, learned CATH-optimized embedding for each protein. During optimization, the Soft Margin Loss was used to maximize the distance between proteins from different CATH classes (anchor-negative pairs) while minimizing the distance between proteins in the same CATH class (anchor-positive pairs). Triplets of varying structural similarity were used simultaneously to optimize a single, shared network, i.e., all four CATH-levels were simultaneously learned by the same FNN. Panel B sketches how the contrastive learning FNN is used for inference, i.e., making predictions for new proteins. For all proteins in a lookup set with experimental annotations (labeled proteins; here the CATH lookup set), as well as for a query protein without experimental annotations (unlabeled proteins) all embeddings are extracted through the following two steps: (1) extract the per-residue embeddings from the original pLM (ProSE, ESM-1b, ProtBERT, ProtT5) and create per-protein embeddings by averaging over protein length, and (2) use those embeddings as input to the pre-trained FNNs, i.e., ProtTucker. Similar to homology-based inference (HBI), predictions are generated by transferring the annotation of the closest hit from the lookup set to the query protein. While HBI defines the closest hit usually as the hit with the lowest E-value, EAT, the embedding-based annotation transfer exploited here, transferred to the hit with the smallest Euclidean distance in ProtTucker embedding space.

Methods

CATH hierarchy. The CATH (4,25) hierarchy (v4.3) classifies three-dimensional (3D) protein structures obtained from the PDB (Protein Data Bank (84)) at the Class-, Architecture-, Topology- and Homologous superfamily-level. On average, higher levels (further away from root, e.g., H>T>A>C) are more similar in their 3D structure or have more residues for which the same level of 3D similarity is reached. We used increasing overlap in this hierarchical classification as a proxy to define increasing structural similarity between pairs of proteins. For example, we assumed that any two proteins with the same topology (T) are structurally more similar than any two proteins with identical architecture (A) but different topology (T). In more formal terms: SIM3D(P1,P2)>SIM3D(P3,P4), where T(P1)=T(P2) & T(P3)≠T(P3) & A(P3)=A(P3). This notion of similarity was applied on all four levels of CATH.

Data set. The sequence-unique datasets provided by CATH v4.3 (123k proteins, CATH-S100) provided training and evaluation data for ProtTucker. A test set (300 proteins, dubbed test300 in the following) for final evaluation and a validation set (200 proteins, dubbed ***val200***) for early stopping were randomly split off from CATH-S100 while ensuring that (1) every homologous superfamily appeared maximally once in test300 ∩ val200 and (2) each protein in test300 & val200 has a so called Structural Sub-group (SSG) annotation, i.e., clusters of domain structure relatives that superpose within 5Å (0.5nm), in CATH. To create the training set, we removed any protein from CATH-S100 that shared more than 20% pairwise sequence identity (PIDE) to any validation or test protein according to MMSeqs2 (10) applying its iterative profile-search (*--num-iterations 3*) with highest sensitivity (*-s 7.5*) and bidirectional coverage (*--cov-mode 0*). Additionally, large families (>100 members) within CATH-S100 were clustered at 95% PIDE and length coverage of 95% of both proteins using MMSeqs2 (bidirectional coverage; *--cov-mode 0*). The cluster representatives were used for training (74k proteins, dubbed ***train74k***) and as lookup set during early stopping. We needed a lookup set because contrastive learning outputs directly embeddings instead of class predictions. For the final evaluation on *test300*, we created another lookup set but ignored val200 proteins during redundancy reduction (77k proteins, dubbed ***lookup77k***). This allowed to access validation proteins and proteins sequence-similar to those during the final evaluation while "hiding" them during training nor using them for any other optimization. To ensure strict non-redundancy between lookup77k and test300, we further removed any protein from *test300* with HVAL>0 (29) to any protein in *lookup77* (219 proteins, dubbed ***test219*** in the following). All performance measures were computed using *test219*.

Data augmentation has been shown to be crucial for contrastive learning to reach performance in other fields (85). However, no straightforward way exists to augment protein sequences as randomly changing sequences very likely alters or even destroys protein structure and function. Therefore, we decided to use homology-based inference (HBI) for data augmentation during training, i.e., we created a new training set based on Gene3D (v21.0.1) which uses Hidden Markov Models (HMMs) derived from CATH domain structures to transfer annotations from labeled CATH to unlabeled UniProt. Towards this end, we first clustered the 61M protein sequences in Gene3D at 50% PIDE and 80% coverage of both proteins (bidirectional coverage; *--cov-mode 0*) and then applied the same MMSeqs2 profile-search (*--num-iterations 3 –s 7.5*) as outlined above to remove cluster representatives with ≥20% PIDE to any protein in test300 or val200 (PIDE(P_{train},$P_{test300|val200}$)≤20%). This filtering yielded 11M sequences for an alternative training set (dubbed ***train11M***).

The remote homology detection capability of ProtTucker was further analyzed using a strictly non-redundant, high-quality dataset. Toward this end, we first clustered CATH v4.3 at 30% using HMM profiles from HMMER and additionally discarded all proteins that did not have an equivalent entry in SCOPe, i.e., the domain boundaries and the domain-superfamily assignment had to be nearly identical (3186 proteins, CATH-S30). We used the highly sensitive structural alignment scoring tool SSAP (8,9) to compute the structural similarity between all protein pairs in this set.

We probed whether or not ProtTucker embeddings might also help in solving tasks unrelated to protein structure/CATH, using as proxy a dataset assessing subcellular location prediction in ten states (67,86). Toward this end, we used Embedding-based Annotation Transfer (EAT) to transfer annotations from 9.5k proteins used to develop other methods (dubbed *DeepLocTrain)* to 490 proteins in a recently proposed test set (dubbed *setHard*) that was strictly non-redundant to *DeepLocTrain*. Datasets described elsewhere in more detail (67,86).

Data representation. Protein sequences were encoded through distributed vector representations (embeddings) derived from four different pre-trained protein language models (pLM): (1) **ProtBERT** (5) based on the NLP (Natural Language Processing) algorithm BERT (14). ProtBERT has been trained on BFD (Big Fantastic Database) with over 2.3 billion protein sequences (87). (2) **ESM-1b** (7) is conceptually similar to (Prot)BERT (both use a stack of Transformer encoder modules) but trained on UniRef50 (88). (3) **ProtT5-XL-U50** (5) (dubbed ProtT5 for simplicity) based on the NLP sequence-to-sequence model T5 (Transformer encoder-decoder architecture) (15) trained on BFD and fine-tuned on Uniref50. (4) **ProSE** (1) trained long short-term memory cells (LSTMs (89)) either solely on 76M unlabeled protein sequences in UniRef90 (**ProSE-DLM**) or on additionally predicting intra-residue contacts and structural similarity from 28k SCOPe proteins (79)(multi-task: **ProSE-MT**). While ProSE, ProtBert and ESM-1b were trained on reconstructing corrupted tokens/amino acids from non-corrupted (protein) sequence context (masked language modeling), ProtT5-XL-U50 was trained by *teacher forcing*, i.e., input and targets were fed to the model with inputs being corrupted protein sequences and targets being identical to inputs but shifted to the right (span generation with span size of 1 for ProtT5). All pLMs, except for ProSE-MT, were optimized only through self-supervised learning exclusively using unlabeled sequences for pre-training.

PLMs output a single vector for each residue yielding an *LxN*-dimensional matrix (L: protein length, N: embedding dimension; N=1024 for ProtBERT/ProtT5; N=1280 for ESM-1b; N=6165 for ProSE). From this *L x N* embedding matrix, we derived a fixed-size N-dimensional vector representation for each protein by averaging over protein length (Fig. 1 (5)). The pLMs were used as static feature encoder only, i.e., no gradient was backpropagated for fine-tuning. As recommended in the original publication (5), for ProtT5, we only used the encoder part of ProtT5 in half-precision to embed protein sequences. Similarly, ProtBERT embeddings were derived in half-precision.

Contrastive learning: architecture. A two-layer feedforward neural network (FNN) was used to project fixed-size per-protein/sentence-level embeddings from 1024-d (or 1280-d/6165-d for ESM-1b or ProSE respectively) to 256 and further to 128 dimensions with the standard hyperbolic tangent (*tanh*) as non-linearity function between layers. We also experimented with deeper/more sophisticated networks without apparent gain from blowing up the number of free parameters (data not shown). This confirmed previous findings that simple networks suffice when inputting advanced embeddings (5,7,65,66). As the network was trained using contrastive learning, no final classification layer was needed. Instead, the 128-dimensional output space was optimized directly.

Contrastive learning: training. During training, the new embedding space spanned by the output of the FNN was optimized to capture structural similarity using triplets of protein embeddings. Every triplet consisted of three proteins: an anchor, a positive and a negative. In each epoch, all proteins in *train74k* were used once as anchor, while positives and negatives were sampled randomly from *train74k*. Toward this end, we applied a *hierarchy-sampling* filter that proceeded as follows. First, a random level α (α=[1,2,3,4]) describing the increasing structural overlap between triplets was picked to define a positive label (sharing the same CATH-label as the anchor up to α) as well as a negative label (sharing the same CATH-label as the anchor up to but not including α). For instance, if the anchor's CATH-label were 1.25.10.60 (Rad61, Wapl domain) and we randomly picked α=3 (topology-level), only proteins with the anchor's topology (1.25.10.x;

Leucine-rich Repeat Variant) would qualify as positive while the negative would have to share the anchor's architecture (1.25.y.z; Alpha Horseshoe) but would have to have a different topology (y!=10). Self-hits of the anchor were excluded. From the subset of training proteins that complied with those constraints, one positive and one negative protein were picked at random. If no triplets could be formed that met this criterion (for example, α=4 with a single-member homologous superfamily could not produce a valid positive because no positives were available for this anchor/α combination), a different α was sampled at random until a valid triplet could be formed (eventually, all proteins will have a partner at the class level).

This hierarchical sampling zooms into semi-hard triplets (anchor, positive and negative) by ensuring a certain overlap of anchor and negative compared to picking a negative at random. Thereby, trivial triplets are under-sampled (avoided), i.e., those with 3D structure so different that the separation becomes trivial (*daylight zone*). However, triplets were still selected at random, as long as they complied to the above constraints. It has been shown that sampling neither too trivial (little signal), nor too hard triplets (much noise/outliers) is crucial for the success of contrastive learning (90). Besides the hierarchy-sampling outlined above, one technical solution toward this end is referred to as *batch-hard sampling* (90). This increased the chance of picking semi-hard samples and avoids extreme outliers in a computationally efficient manner by enforcing to sample hard triplets only within each mini-batch but not within the entire data set. We combined *batch-hard sampling* with the triplets created using *hierarchy-sampling* by re-wiring all proteins, irrespective of anchor, positive or negative, within one mini-batch such that they satisfied the hierarchy-sampling criterion and had maximum/minimal Euclidean distance for anchor-positive/anchor-negative pairs. Assuming that multiple proteins with the topology Leucine-rich Repeat Variant from the example above were within one mini-batch, the hardest positive for each anchor would be picked by choosing the anchor-positive pair with the largest Euclidean distance. This sampling was applied to all four levels of the CATH-hierarchy, so triplets were re-wired on all four CATH levels resulting in a total batch-size of about: batch_size * 3 * 4. This was an "about" instead of "equal" because for some mini-batches, not all proteins had valid triplets for all four levels.

Finally, the same two-layer FNN was used (hard parameter sharing) to project the 1024-d (or 1280-d/6165-d for ESM-1b or ProSE respectively) embeddings of all proteins, irrespective of anchor, positive or negative, to a new 128-d vector space. The *Soft Margin Loss* was used to optimize this new embedding space such that anchor-positive pairs were pulled together (reduction of Euclidean distance) while pushing apart anchor-negative pairs (increase of Euclidean distance). As a consequence, triplets of varying structural similarity were used simultaneously to optimize a single, shared network, i.e., all four CATH-levels were learned by the same network at the same time (Fig. 1A). We used the *Adam optimizer* (91) with a learning rate of 0.001, and a batch-size of 256 to optimize the network. The effective batch-size increased due to batch-hard sampling to a maximum of 3072, depending on the number of valid triplets that could be formed within the current mini-batch. Training terminated (*early stopping*) at the highest accuracy in predicting the correct homologous superfamily for set *val200.*

Evaluation and prediction (inference). Supervised training directly outputs class predictions. Contrastive learning, instead, outputs a new embedding space. Consequently, predictions were generated similar to homology-based inference (HBI), i.e. if we had a protein with known annotations (CATH assignment) X and a query protein without annotations Q, then HBI describes the inference of transferring the annotation from X to Q when SIM(X,Q)>threshold. For contrastive learning instead of transferring the annotation of the lookup protein (X) with smallest E-value (translating to highest SIM) to a query protein (Q), we used the shortest Euclidean distance in embedding space to transfer the annotation (Fig. 1B). In previous studies (5,69,70), we found the Euclidean distance to perform better than cosine distances that appear more popular in AI/NLP communities. The Euclidean distance was also used to optimize ProtTucker embeddings. The final evaluation was performed on the set *test300* with the *lookup77k* as lookup set (set of all X). If no protein in the lookup set shared the annotation of the query protein at a certain level of the CATH-hierarchy (more likely for H than for C), the sample was excluded from the evaluation of this CATH-level as no correct prediction was possible (Table S1).

During evaluation, we compared the performance of our embedding-based annotation transfer (EAT) to HBI using the sequence comparisons from MMSeqs2 (10). While transferring only the HBI hit with the lowest E-value, we searched for hits up to an E-value of 10 to ensure that most proteins had at least one hit while using the highest sensitivity setting (-s 7.5). Additionally, we used publicly available CATH-Gene3D (4,92) Hidden Markov Models (HMMs) along with HMMER (6) to detect remote homologs up to an E-value of 10.

For both approaches, EAT and HBI, we computed the accuracy as the fraction of correct hits for each CATH-level. A hit at lower CATH-levels could be correct if and only if all previous levels were correctly predicted. Due to varying number of samples at different CATH-levels (Table S1), performance measures not normalizing by the background numbers could still be higher at lower levels. If a query protein did not have a hit in the lookup set though a lookup protein of the same CATH annotation exists, it was considered as wrong prediction when computing accuracy. A random baseline was computed by transferring annotations from a randomly picked protein in *lookup77k* to *test219.*

Performance measures. The four coarse-grained classes at the top CATH level ("C") are defined by their secondary structure content. These four branch into 5481 different superfamilies with distinct structural and functional aspects (CATH v4.30). However, most standard metrics are defined for binary cases which requires some grouping of predictions into four cases: 1) TP (true positives): correctly predicted to be in the *positive* class, 2) TN (true negatives): correctly predicted to be in the *negative* class, 3) FP (false positives): incorrectly predicted to be positives, and 4) FN (false negatives): incorrectly predicted to be in in the negative class. Here, we focused on performance measures applicable for multiclass problems and are implemented in scikit (93). These were in particular: **accuracy** (Acc, Eqn. 1) as the fraction of correct predictions

$$Accuracy(y,\hat{y}) = \frac{1}{n_samples}\sum_{i=0}^{n_samples-1} 1(\hat{y}_i = y_i) \qquad \text{(Eqn. 1)}$$

with y_i being the ground truth (experimental annotation) and $\hat{y}_i$ the prediction for protein i. In analogy, we defined **coverage** as the proportion of the *test219* proteins for which a classifier made a prediction at a given prediction reliability $\widehat{y_i^r}$ and reliability threshold θ:

$$Coverage(y,\hat{y}) = \frac{1}{n\_samples}\sum_{i=0}^{n\_samples-1} 1(\hat{y}_i^r < \theta) \qquad \text{(Eqn. 2)}$$

In these definitions accuracy corresponds to precision, and coverage to recall binarizing a multiclass problem through micro-averaging, i.e., by counting the total TPs, FPs and FNs globally, irrespective of the class. The multi-class extension of **Matthew's correlation coefficient (**MCC, (94)) was defined as:

$$MCC = \frac{c \times s - \sum_k^K p_k \times t_k}{\sqrt{(s^2 - \sum_k^K p_k^2) \times (s^2 - \sum_k^K t_k^2)}} \qquad \text{(Eqn. 3)}$$

with $t_k = \sum_i^K C_{ik}$ as the number of times class k truly occurred, $p_k = \sum_i^K C_{ki}$ as the number of times class k was predicted, $c = \sum_k^K C_{kk}$, the total number of samples correctly predicted, and $s = \sum_i^K \sum_j^K C_{ij}$, the total number of samples.

95% confidence intervals for accuracy and MCC were estimated over n=1000 bootstrap sets; for each bootstrap set we randomly sampled predictions from the original test set with replacement. Standard deviation was calculated as the difference of each test set (x_i) from the average performance $\langle X \rangle$ (Eqn. 4). The standard error was calculated by dividing σ by the square root of sample size (Eqn. 5) and 95% confidence intervals were estimated by multiplying the standard error by 1.96.

$$\text{Standard deviation (StdDev)} = \sqrt{\frac{x_i - \langle X \rangle^2}{n}} \qquad \text{(Eqn. 4)}$$

$$\text{Standard error (StdErr)} = \frac{SD}{\sqrt{(n-1)}} \qquad \text{(Eqn. 5)}$$

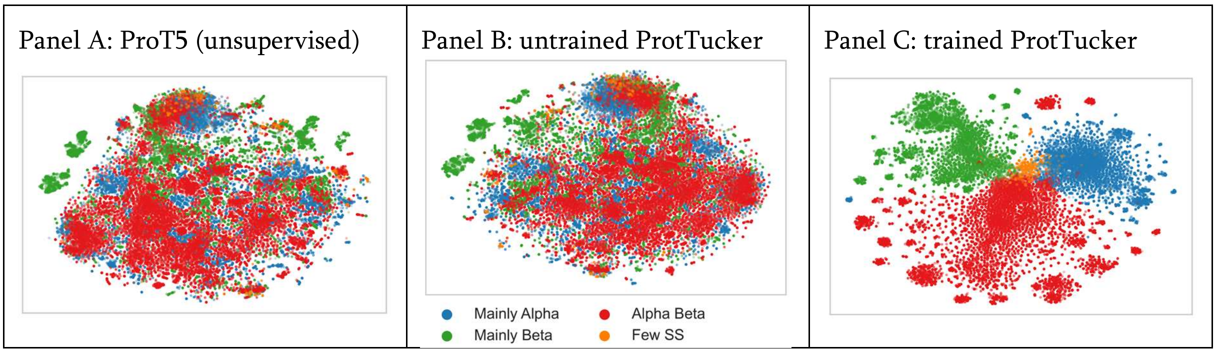

Fig. 3: Contrastive learning improved CATH class-level clustering. Using t-SNE (2), we projected the high-dimensional ProtTucker(ProtT5) embedding space onto 2D before (Panel A; ProtT5) and after (Panel C; ProtTucker(ProtT5)) contrastive learning. To assess the impact of different embedding dimensions (ProtT5: 1024-d vs ProtTucker(ProtT5): 128-d), Panel B visualized the same data embedded with an untrained version of ProtTucker. For all plots, the embedding dimensionality was first reduced by Principal Component Analysis (PCA) to 50 dimensions and parameters of the subsequent t-SNE were identical (perplexity=150, learning_rate=400, n_iter=1000, seed=42). The colors mark the major class level of CATH (C), distinguishing proteins according to their major distinction in secondary structure content.

Results

Generalization from HBI to EAT. Homology-based inference (HBI) uses sequence similarity to transfer annotations from labeled (experimentally characterized) to unlabeled proteins. More specifically, an unlabeled query protein Q is aligned against a set of proteins X with experimental annotations (dubbed *lookup set*) and the annotation of the best hit, e.g. measured as lowest E-value, is transferred IF it is below a certain threshold (e.g. E-value(Q,X)<10^{-3}). Here, we generalized HBI through embedding-based annotation transfer (EAT) by removing the implicit evolutionary connection (homology-based) and expanding from distance in sequence to distance in embedding space (Fig. 1B). Towards this end, we first compared embeddings from five different protein language models (pLMs), namely ProSE-DLM & ProSE-MT (1), ProtBERT & ProtT5 (5), and ESM-1b (7). Next, we used triplets of proteins (anchor, positive, negative) to learn a new embedding space by pulling protein pairs from the same CATH class (anchor-positive) closer together while pushing apart pairs from different CATH classes (anchor-negative; Fig. 1A). In the following, we referred to this method as to ProtTucker. To develop ProtTucker, we did not fine-tune the pre-trained pLMs. Instead, we created a new embedding space using a two-layer feed-forward neural network (FNN).

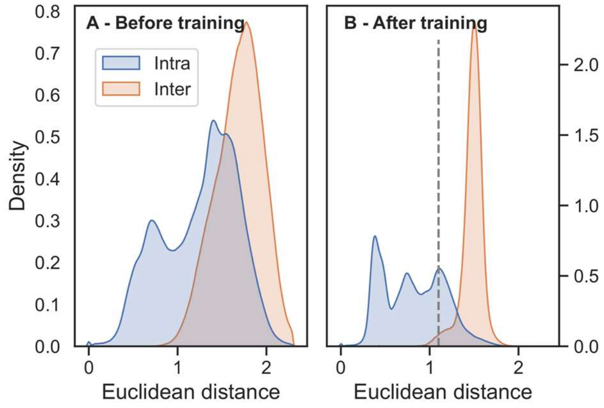

Fig. 2: Contrastive learning separated positives from negatives. The structural similarity defined by increasing overlap in CATH drove the contrastive learning of a new embedding space. The new embeddings distanced protein pairs with different CATH classifications (red; same topology but different superfamily) while focusing pairs with the same CATH classification (blue; same superfamily). These graphs compared the Euclidean distance for all such pairs from the set *train74k* before (Panel A) and after (Panel B) contrastive training. Input to the FNN were the raw embeddings from ProtT5 (5), output were the new ProtTucker(ProtT5) embeddings. The dashed line at Euclidean distance of 1.1 in B marked the threshold at which EAT performances started to decrease (Fig. 5).

EAT with raw embeddings level with HBI. First, we tested EAT using embeddings from pre-trained pLMs to transfer annotations from all proteins in set *lookup77k* to any protein in set *test300*. All pLMs significantly (at 95% CI - confidence interval) outperformed random annotation transfer (Table 1). Performance differed between pLMs (Table 1), with ProtBERT (5) being consistently worse than LSTM-based ProSE-DLM or more advanced transformers realized by ESM-1b (7) and ProtT5 (5). The latter two also numerically outperformed ProSE-DLM and HBI using MMseqs2 (10), especially on the hardest level of superfamilies. However, MMseqs2 had been used for redundancy-reduction, i.e., the data set had been optimized for minimal performance of MMseqs2. HBI using publicly available HMM-profiles from CATH-Gene3D (4) together with the HMM-profile based advanced sequence alignment method HMMER (6) designed for more remote homology detection, outperformed all raw embeddings on the level of homologous superfamilies, while embeddings from ESM-1b and ProtT5 appeared superior on the class- and architecture-level (Table 1). In fact, all HBI values, for MMseqs2 and HMMER, were highest for the H-level, and 2nd highest for the C-level. In contrast, raw pLM embeddings mirrored the random baseline trend, with numbers being inversely proportional to the rank in C, A, T, H (highest for C, lowest for H, Table 1).

EAT improved through optimized embeddings. ProSE-MT expands ProSE-DLM by additionally training on intra-residue contacts and structural similarity using labeled data from SCOPe (1). This additional effort was reflected in the increased performance for all CATH levels (Table 1, ProSE-MT>ProSE-DLM). The supervision pushed the LSTM-based ProSE-MT to reach performance levels close to the unsupervised, raw embeddings from transformer-based ProtT5. The performance gap increased with classification difficulty (Table 1, ProtT5>ProSE-MT, especially at H-level).

Table 1: Accuracy for annotation transfer to queries in test219 *,*

	Method/Input	*C*	*A*	*T*	*H*	*Mean*
Baseline	*Random*	29 ±6	9 ±4	1 ±2	0 ±0	10 ±3
HBI	*MMSeqs2 (sequence)*	52 ±7	36 ±6	29 ±6	35 ±6	38 ±6
	HMMER (profile)	70 ±6	60 ±6	59 ±7	77 ±7	67 ±6
EAT - Unsupervised	*ProSE-DLM*	74 ±6	48 ±7	28 ±6	25 ±7	44 ±6
	ESM-1b	79 ±5	61 ±6	50 ±7	57 ±8	62 ±7
	ProtBERT	67 ±6	38 ±6	22 ±6	18 ±6	36 ±6
	ProtT5	84 ±5	67 ±6	57 ±6	64 ±8	68 ±6
EAT - Supervised	*ProSE-MT*	82 ±5	65 ±6	52 ±7	56 ±8	64 ±7
EAT - Contrastive learning – ProtTucker	*ProSE-DLM*	78 ±4	53 ±6	32 ±6	29 ±7	48 ±6
	ProSE-MT	87 ±4	68 ±6	53 ±7	55 ±8	66 ±6
	ESM-11b	87 ±4	68 ±6	59 ±7	70 ±7	71 ±6
	ProtBERT	81 ±5	52 ±7	37 ±6	39 ±8	52 ±7
	ProtT5	**89 ±4**	75 ±6	64 ±6	76 ±6	76 ±6
	ProtT5 (train11M)	88 ±4	**77 ±5**	**68 ±5**	**79 ±7**	**78 ±6**

* Accuracy (Eqn. 1) for predicting CATH (3,4) levels by transferring annotations from *Lookup77k* (lookup set) to *test219* (queries); shown for each of the four levels of the CATH database from the most coarse-grained level class C to the most fine-grained level of homology H. The column *Mean* marked the average over the four CATH class performance for each method. Queries with at least one lookup protein of the same CATH classification but without any hit at E-value<10 for MMSeqs/HMMER were counted as incorrect predictions. Errors indicate bootstrapped 95% confidence intervals, i.e., 1.96 standard errors (Eqn. 5). Queries with at least one lookup protein of the same CATH annotation but without any hit (no hit with E-value<10 for MMSeqs/HMMER; irrelevant for EAT) were counted as wrong predictions. **Bold** letters mark the numerically highest values (averages over all *test219* proteins) in each column irrespective of the confidence interval.

Methods: Baseline: Random transferred the label of a randomly picked protein; HBI: MMSeqs2 (10) used single sequence search to transfer the annotation of the hit with the lowest E-value; HBI: HMMER used HMM-profiles (6); EAT-unsupervised: embedding-based transfer of annotations using the smallest Euclidean distance measured in embedding space of unsupervised pLMs ProSE-DLM , ESM-1b (7), ProtBERT and ProtT5 (5); EAT-supervised: annotation transfer using ProSE-MT trained on structural data in SCOPe; EAT: contrastive learning ProtTucker: contrastive learning trained on CATH classifications in *train74k* using as input embeddings from ProSE-DLM, ProSE-MT, ESM-1b, ProtBERT, and ProtT5; ProtTucker-ProtT5 (train*11M)* trained on additional data from Gene3D (*train11M*).

EAT improved by contrastively learning embeddings. The idea underlying contrastive learning is to bring members from the same class closer while pushing those from different classes further apart. One success of contrastive learning is the degree to which these two distributions (same vs. different) were separated due to training. In fact, the distribution of all pairwise Euclidean distances within (intra/same) and between (inter/different) homologous superfamilies in the data set *train74k* changed substantially through contrastive learning (Fig. 2: before and after for ProtT5). Before contrastive learning the distributions between/inter (Fig. 2: red) and within/intra (Fig. 2: blue) overlapped much more than after (Fig. 2: more overlap left panel than right).

Displaying the information learned by the embeddings, we compared t-SNE projections colored by the four main CATH classes before (Fig. 3A) and after (Fig. 3C) contrastive learning. These two projections compared 1024 dimensions from ProtT5 (Fig. 3A) with 128 dimensions from ProtTucker (Fig. 3C). To rule out visual effects from higher dimensionality, we also compared the untrained, randomly initialized version of ProtTucker using pre-trained ProtT5 embeddings as inputs (Fig. 3B). In all cases, the set of proteins (train74k) and the parameters for dimensionality reduction were identical. The t-SNE projection of the raw ProtT5 embeddings qualitatively suggested some class separation/clustering. The information underlying this rudimentary separation was preserved when projecting the ProtT5 embeddings through an untrained ProtTucker (Fig. 3B). However, embeddings from ProtTucker(ProtT5), i.e., those available after refining ProtT5

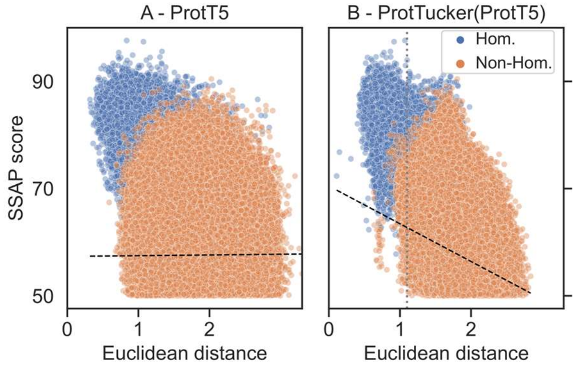

Fig. 4: ProtTucker captured fine-grained structural similarity. A set of non-redundant proteins (CATH-S30) probed the remote homology detection embeddings before (Panel A, ProtT5) and after contrastive learning (Panel B, ProtTucker(ProtT5)). The Euclidean distance between the ProtTucker embeddings (Panel B) correlated better with structural similarity computed via SSAP than their unsupervised counterparts (Panel A): Spearman rho=0.22 and rho=0.05 (black dashed lines), respectively. This correlation increased to 0.37 and 0.26 respectively when considering protein pairs with higher structural similarity (SSAP-score > 70 (8,9)). Of all structurally similar protein pairs, only 1.8% (53k) were in the same homologous superfamily (blue). The unsupervised ProtT5 already separated homologous pairs from others, but ProtTucker(ProtT5) improved, especially, for hard cases with low structural similarity. The gray dashed line at Euclidean distance of 1.1 in *Panel B* marked the threshold at which EAT performances started to decrease (Fig. 5).

through contrastive learning, separated the classes much more clearly.

To further probe to which extent contrastive learning helped in capturing remote homologs, we compared the Euclidean distance between all protein pairs in a 30% non-redundant dataset (CATH-S30) with the structural similarity of those pairs computed via SSAP (8,9) (Fig. 4). From the ~10M possible pairs between the 3,186 proteins in CATH-S30 (problem not fully symmetric, therefore N*(N-1): 10.1M), 7.1M had to be discarded due to low quality (SSAP-score <50), leaving 2.9M pairs of which only 1.8% (53k pairs) had the same homologous superfamily (Fig. 4: blue). Despite this imbalance, unsupervised ProtT5 (Fig. 4A) already separated those to a certain extent from protein pairs with different homologous superfamily (Fig. 4: orange). Still, ProtTucker(ProtT5) improved this separation, especially, for pairs with low structural similarity (Fig. 4B). This is also reflected in the Spearman correlation coefficient between the structural similarity and the Euclidean distance which increased from 0.05 to 0.22 after applying contrastive learning. When considering only the subset of pairs that are likely to have a similar fold (SSAP-score>70), this correlation increases to 0.26 and 0.37 for ProtT5 and ProtTucker(ProtT5), respectively.

The trend captured by the better separation of distributions (Fig. 2) and structural features (Fig. 3, Fig. 4) translated directly into performance increases: all embeddings optimized on the CATH hierarchy through contrastive learning yielded a better EAT classification than the raw embeddings from pre-trained pLMs (Table 1). As ProtTucker described the process of refining raw embeddings through contrastive learning, we used the annotation ProtTucker(X) – in this section also shortened to PT(X) - to refer to the embeddings output by using the pre-trained pLM X as input for the contrastive learning. The improvements were larger for more fine-grained CATH levels, i.e., all models improved significantly for the H-level while only PT(ProtBERT) and PT(ESM-1b) improved 4-14 or 0-21 percentage points for the C-, and the H-level, respectively. PT(ProtT5) consistently outperformed all other pLMs on all four CATH-levels, with an increasing performance gap toward the more fine-grained H-level at which all pLMs except for PT(ESM-1b) performed significantly worse. The improvements from contrastive learning for PT(ProSE-DLM) and PT(ProSE-MT) were mostly consistent but largely insignificant. Especially, the model already optimized using labeled data (ProSE-MT) hardly improved through another round of supervision by contrastive learning and even worsened slightly at the H-level.

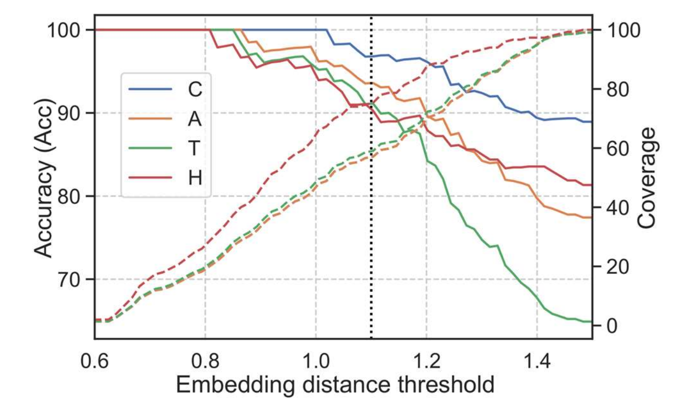

Fig. 5: Embedding distance correlated with reliability. Similar to varying E-value cut-offs for HBI, we examined whether the fraction of correct predictions (accuracy; left axis; Eqn. 1) depended on embedding distance (x-axis) for EAT (embedding-based annotation transfer). Toward this end, we transferred annotations for all four levels of CATH (Class: blue; Architecture: orange; Topology: green; Homologous superfamily: red) from proteins in the set lookup77k to the queries in set test219 (Panel B in Fig. 1) using the hit with smallest Euclidean distance. The fraction of test219 proteins having a hit below a certain distance threshold (coverage, right axis, dashed lines; Eqn. 2) was evaluated separately for each CATH level. For example, at a Euclidean distance of 1.1 (marked by black vertical dots), 78% of the test219 proteins found a hit at the H-level (Cov(H)=78%) and of 89% were correctly predicted (Acc(H)=89%). Similar to decreasing E-values for HBI, decreasing embedding distance correlated with EAT performance. This correlation importantly enables users to select only the, e.g., 10% top hits, or all hits with an accuracy above a certain threshold, or as many hits to a certain CATH level as possible, depending on the objectives.

We augmented the training set for PT(ProtT5) by adding HBI-hits from HMM-profiles provided by CATH-Gene3D (if sequence dissimilar to test300). This increased the training set size from 74k ($74*10^3$) to 11m ($11*10^6$) proteins (15-fold increase) and raised performance, although the higher values were neither statistically significant nor consistent (Table 1: values in last row not always higher than those in 2nd to last row).

Embedding distance correlated with accuracy. The performance (here measured as MCC, Eqn. 3) of HBI inversely correlated with E-value (Fig. 6, HBI-methods): more significant hits (lower E-values) more often shared the same CATH level than less significant hits (higher E-values). In analogy, we explored the corresponding relation for EAT, namely the correlation between accuracy (Eqn. 1) and embedding distance for ProtTucker(ProtT5). Indeed, accuracy correlated with embedding distance (Fig. 5: solid lines) while recall inversely correlated (Fig. 5: dashed lines) for all four classes. For instance, when transferring only annotations for closest hits with Euclidean distances of 1.1 or less, predictions were made for 59%, 59%, 61% or 78% of the test set (coverage, Eqn. 2) of these 97%, 93%, 90% or 89% were correct for levels C, A, T, H, respectively.

ProtTucker reached into the midnight zone. The annotation transfer by HBI depends on the sequence similarity between the query (unknown annotation) and template (experimental annotation) are. Usually, this significance is measured as the chance of finding a hit at random for a given database size (E-value; the lower the better). Here, we compared the effect of gradually removing hits depending on their E-values. Essentially, this approach measured how sensitive performance was to the degree of redundancy reduction between query and lookup set. For instance, at a value of 10^{-3} (dashed vertical lines in Fig. 6), all pairs with E-values≤10^{-3} were removed (note: this was the standard for all results reported above). Traditional HBI based on sequence similarity alone clearly performed much better when redundancy remained (Fig. 6: HMMer and MMseqs much lower toward right for all CATH levels). While we observed a similar tendency for EAT, this was much less pronounced, i.e., EAT succeeded for pairs of proteins with very different sequences (Fig. 6 toward right) almost as well as for more pairs for which a more sequence similar pair could have been found (Fig. 6 toward left: EAT almost as high as toward right).

ProtTucker not a generalist. We evaluated the generality of ProtTucker embeddings by using them as exclusive input to predict subcellular location in ten states. Toward this end, we used EAT to transfer annotations from an established training set (*DeepLocTrain*, Table S2) to a strictly non-redundant test set (*setHard*, Table S2). While the ProtTucker(ProtT5) embeddings outperformed the raw ProtT5 embeddings in the CATH classification for which they were optimized (structural similarity; Table 1), there appeared no performance gain in predicting location. Conversely, performance also did not decrease significantly, indicating that the new embeddings retained some of the information available in ProtT5 embeddings.

Family size mattered. By clustering very large protein families (>100 members after redundancy reduction) at 95% PIDE, we constrained the redundancy in set *train74k*. Nevertheless, when splitting *test219* into three bins of varying family sizes, we still observed a trend towards higher accuracy (Eqn. 1) for larger families at the H-level (Fig. S1). We chose the three bins such that they contained about the same number of samples (small families: ≤10 members, medium: 11-70, and large: ≥70 members). Especially, unsupervised EAT using the raw ProtT5 embeddings exhibited a clear trend towards higher accuracy with increasing family size. In contrast, the two HBI-methods (MMseqs2, HMMER), as well as EAT using the optimized ProtTucker(ProtT5) embeddings performed similarly for small and medium-sized families and much better for large families.

EAT complements HBI. As previously shown (70), ProtTucker embeddings can improve the clustering of functional families, e.g., FunFams, (78). Here, we showed how EAT can be used to detect outliers. Toward this end, we computed pairwise Euclidean distances between the embeddings of all protein pairs in set *train74k* and analyzed the five pairs (10 proteins) with the largest Euclidean distance despite sharing the same homologous superfamily (Table S3). To find potential alternative annotations, we further computed the nearest neighbors of those ten proteins. For instance, the nearest neighbor (3skjF00, the human *Galactose-binding domain-like* (95)) of one of the outliers (3heiA00, the human *phosphorylase kinase* (96)) links to the same UniProt entry (*EPHA2_HUMAN* (88)) with the same enzymatic activity (EC number 2.7.10.1 (97)). In contrast to this, the protein that is also part of the *Phosphorylase Kinase* superfamily but has large embedding distance (4pdyA01, the bacterial aminoglycoside phosphotransferase C Chang et al., unpublished PDB identifier 4PDY) links to a different UniProt entry (*C8WS74_ALIAD*).

EAT blazingly fast. The time required to generate optimized ProtTucker(ProtT5) embeddings using *train74k* took in total 26m, with 11m spent on creating ProtT5 embeddings and 15m needed for training. All times were measured on a single Nvidia RTX A6000 with 48GB of vRAM and an AMD EPYC ROME 7352 (same machine used to compute all time estimates). Embeddings were generated by running ProtT5 in half-precision with batch processing.

When predicting for new queries (referred to as *inference* in the NLP machine learning community), the proposed ProtTucker models require *labeled lookup* proteins from which annotations can be transferred to unlabeled query proteins. This lookup set needs to be pre-computed only once for the first query and can be re-used for all subsequent queries at any future time. The time required to generate ProtTucker embeddings from the embeddings of pLMs was negligible as its generation required only a single forward pass through a two-layer FNN. This implied that the total time for EAT with ProtTucker was largely determined by the speed with which the embeddings can be extracted from each pLM. For instance, creating per-protein embeddings from ProtT5 for the 123k proteins in CATH-S100 required 1080 seconds (s). The total time for creating ProtTucker(ProtT5) embeddings for the same set on the same machine was 1098s (18.3m), i.e., ProtTucker added about 1.6% more demands on the resources. Creating HMM profiles for the same set using either MSAs from MMSeqs2 (*--num-iterations 3, -s 7.5*) or jackhmmer would have taken 15m or 30h, respectively.

To predict using EAT, users have to embed only single query proteins requiring, on average, 0.009s per protein for the CATH-S100 set. Using either single protein sequence search (MMSeqs2), pre-computed HMM profiles (HMMER) or pre-computed embeddings (ProtTucker) to transfer annotations from CATH-S100 to a single query protein took on average 0.025s, 1.5s or 0.0008s, respectively. Proteins in PDB and CATH are, on average, roughly half as short (173 residues) as those from UniProt (343 residues). This is relevant for estimating runtime, because embedding generation scales quadratically with sequence length as highlighted in SOM Fig. 13 of (5).

Building on top of bio_embeddings package (98) we have made a script available that simplifies EAT https://github.com/Rostlab/EAT.

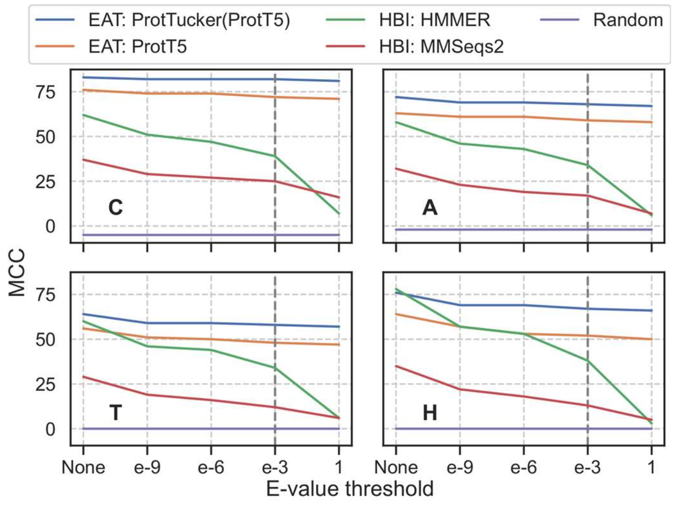

Fig. 6: Performance decreasing with lower residual sequence similarity. We analyzed the effect (upon MCC, Eqn. 3) of removing proteins from *lookup77k* based on their E-value with respect to *test219* for four methods, two HBI-based (green: HMMER (6) and red: MMSeqs2 (10)) and two EAT-based (orange: raw ProtT5 (5) and blue: contrastive learning optimized ProtTucker(ProtT5)). Toward this end, E-values were derived by searching protein sequences in *test219* against *lookup77k* using (i) HMM-profiles from CATH-Gene3D (4) through HMMer and (ii) MMSeqs2 sequence search with highest sensitivity (*-s 7.5, -cov 0*). The E-value cut-off "None" referred to the performance without applying any threshold, i.e., all proteins in lookup77k were used for annotation transfer; all other thresholds referred to removing proteins below this E-value from *lookup77k*. Predictions were considered as false positives when no hit was found, and pairs without remaining CATH class matches were ignored. While the performance of EAT using raw ProtT5 and refined ProtTucker(ProtT5) embeddings decreased when removing more sequence similar pairs (toward right), HBI-based methods dropped significantly more. The default threshold for most sequence searches (E-value<1e-3) was highlighted by vertical, gray, dashed lines.

Discussion

Prototype for supervised learning of hierarchies. We presented a new solution for combining the information implicitly contained in the embeddings from protein Language Models (pLMs) and contrastive learning to learn directly from hierarchically sorted data. As proof-of-concept, we applied the concept to the CATH hierarchy of protein structures (3,4,25,99). Hierarchies are difficult to handle by traditional supervised learning solutions. One typical "hack" is to learn each level in the hierarchy independently (100-102) at the price of both having less information for higher levels (or, depending on the implementation on all levels) and/or of not explicitly benefiting from the hierarchy. Instead, our solution of using contrastive learning on protein triplets (anchor, positive, negative) to learn a new embedding space by pulling together protein pairs from the same CATH classification (anchor-positive) while pushing apart pairs from different CATH classifications (anchor-negative) benefits from CATH's hierarchical structure. Simultaneously training a single, shared feed-forward neural network (FNN) on triplets from all four CATH classification levels allowed the supervised network to capture the hierarchical sorting directly. Encoding protein sequences using previously trained pLMs allowed us to transfer information readily from large but unlabeled protein sequence databases such as BFD (87) to four orders of magnitude smaller but experimentally annotated (labeled) proteins of known 3D structure classified by CATH. In turn, this allowed us to readily leverage aspects of protein structure captured by pLMs that are informative enough to predict structure from embeddings alone (66). Although the raw, pre-trained, unoptimized embeddings clearly captured aspects of the classification (Fig. 2A, Fig. 3A, Fig. 4A, Table 1), contrastive learning boosted this signal significantly (Fig. 2B, Fig. 3C, Fig. 4B, Table 1).

Raw embedding EAT competitive with advanced profile alignments in hit detection. In analogy to the technical solution of homology-based inference (HBI), we used embedding based annotation transfer (EAT, Fig. 1B) to transfer annotations from a labeled set of lookup proteins (here all proteins with a known CATH classification) to an unlabeled set of query proteins (any protein of known sequence but without known structure). However, instead of transferring annotations from the closest hit in sequence space (spanned by PIDE - percentage pairwise sequence identity, E-values, or scores resulting from matching position-specific scoring metrices - PSSMs), EAT transferred annotations to the hit with shortest Euclidean distance in embedding space. On the one hand, the embedding space has a higher dimensionality than sequence space, making the computation of distances more complex. On the other hand, we constrained our approach to the simplistic Euclidean distance as ProtTucker was optimized using this metric. We previously found this relatively simple measure to predict protein function as defined by Gene Ontology (GO) at levels competitive to much more complex approaches (69). Most importantly, we did not yet account for statistics, i.e., we simply computed a score for distance rather than an expectation value assessing the difference between query and background score (all possible queries) (36,103).

The very concept of EAT proved so successful that the raw embeddings from two different pre-trained pLMs (ESM-1b (7), and ProtT5 (5)) already set the bar high for the prediction of CATH levels. The raw, unoptimized values from ESM-1b and ProtT5 outperformed HBI based on advanced HMM-profiles from HMMER (6) on the C- and A-level while falling short on the H-level (Table 1). Furthermore, we showed that ProtT5 already separated protein pairs with the same from those with different homologous superfamilies even when using a lookup set that consisted only of proteins with maximally 30% pairwise sequence identity (Fig 4A). Importantly, this competitive performance was achieved at a much smaller cost in terms of runtime: transferring an annotation from 123k lookup proteins to a single query using a Nvidia RTX A6000 with 48GB of vRAM and an AMD EPYC ROME 7352 took EAT only around 0.0008s using pre-computed embeddings compared to 1.5s for HMMer using pre-computed HMM profiles.

As the lookup embeddings or HMM profiles are computed only once, we neglected this additional step. Obviously, the costs for such pre-computation by far exceeded that for single queries: pre-computing HMM profiles using MMseqs2 took 15m, pre-computing embeddings about 18m using the same set and hardware but utilizing CPU and GPU, respectively.

As for HBI, the accuracy of EAT also increased accuracy for larger families (Fig. S1). One explanation could be that the larger the family, the higher the random hit rate, simply because there are more possible hits. Another, more subtle (and given the enormous compute time needed to train ProtT5 embeddings, more difficult to test) explanation is that the largest CATH families represent most of the largest protein families (4). In fact, a few hundred of the largest superfamilies cover half of the entire sequence space (4,104). Simply due to their immense size, these large families have been sampled more during the pre-training of ProtT5. Yet another possible explanation is that large families tend to be so large because they constitute what physicists refer to as attractors (105), i.e. large families are larger because they are more likely to exist.

EAT on embeddings from contrastive learning intruding into midnight zone. The embedding space resulting from contrastive learning, introduced here, improved performance consistently for all four pLMs (Table 1). This was revealed through several ways of looking at the results from embeddings with and without contrastive learning: (1) the increased separation of protein pairs within the same protein superfamily and between different superfamilies (Fig. 2), (2) the qualitative improvement in the clustering of t-SNE projections (Fig. 3), the better correlation of embedding distance and structural similarity (Fig. 4) and (3) the quantitative improvement in the EAT benchmark (Table 1). On top, the Euclidean distance between the query protein and the lookup hit correlated with accuracy (Fig. 5).

While the best performing pLM (ProtTucker(ProtT5)) was similar to HBI using HMM-profiles on the most fine-grained level of homologous superfamilies (CATH level H, Table 1), the power of EAT became increasingly dominant, the more diverged the level of inference, i.e., EAT outperformed HBI for more distant relations from the midnight zone (CATH level C, Table 1). When making the data set even more non-redundant, i.e., removing more similar sequences, this trend became clearer (Fig. 6). Despite increasing difficulty, the performance of EAT decreased almost only insignificantly while the performance of HBI approached random values for high E-value cut-offs. This trend was supported by the correlation of structural similarity as defined by SSAP (8,9) and the Euclidean distance between protein pairs in a 30% non-redundant data set (Fig. 4). This confirmed that pLMs can detect remote structural homologs with low sequence similarity, and that contrastive learning enabled embeddings to capture subtle differences in structural similarity.

Taken together, these results indicated that contrastive learning captured structural hierarchies in a high-dimensional clustering providing users a novel powerful tool to uncover structural similarities clearly beyond what has been achievable with 50 years of optimizing sequence-based alignment techniques. Using EAT to complement HBI could become crucial for a variety of applications, ranging from finding remote structural templates for protein 3D structure predictions over prioritizing new proteins without any similarity to an existing structure to filtering potentially wrong annotations. One particular example has recently been shown for the proteome of SARS-CoV-2 to unravel entire functional components possibly relevant for fighting COVID-19 (83).

ProtTucker embeddings improved FunFams clustering. In a previous study (70), we showed that a simplistic predecessor of ProtTucker embeddings helped to refine the clustering of proteins sharing one function within CATH superfamilies, the so called FunFams (78). Toward this end, functional consistency was proxied through the enzymatic activity as defined by the EC (Enzyme Commission (97)) number. Applying a preliminary version of ProtTucker embeddings optimized on CATH, improved the annotation transfer of EC-numbers and of ligand binding residues (70). Towards this, we used pairwise Euclidean distance in ProtTucker(ProtBERT) embedding space together with the clustering algorithm DBSCAN (106) to remove outliers from existing FunFams and to create new, more functionally coherent FunFams. As for CATH, the contrastively trained ProtTucker(ProtBERT) also improved for FunFams over its unsupervised counterpart, the raw pLM ProtBERT. Here, we expanded this analysis by showing how EAT offered a complementary perspective on existing HBI-based annotations. Using the proposed method, we could spot potential outliers, i.e., samples with the same annotation but large embedding distance. This might become essential to clean up databases. On top, we could also obtain alternative labels for such outliers from the nearest neighbors (Table S3). This could help in database cleaning. Although we could not reproduce the same level of success when applying EAT to the task of inferring subcellular location in ten states (Table 3), the CATH-optimized ProtTucker embeddings also did not perform worse.

Generic advantages of *contrastive learning*. Contrastive learning simplifies learning on hierarchies as compared to supervised training which usually requires flattening the hierarchy thereby loosing information inherent to the hierarchical structure. Other possible advantages of contrastive learning include the following three. (1) Dynamic data update (*online learning*): While supervised networks require re-training from the beginning to benefit from new data, contrastively trained networks can benefit from new data by simply updating the lookup set. This could even add completely new classes, such as proteins for which the classification will become available only in the future. HBI, of course, has the same advantage, i.e., strictly speaking this advantage originates from the difference between learning to cluster proteins into existing families independently of other family members versus clustering by identifying the most similar proteins in that family. (2) Learn the access, not the data: Instead of forcing the supervised network to memorize the training data, contrastive learning learns how to access the data stored in an external lookup set. (3) Compression: Contrastive learning can act as a compression technique. For instance, for this project we reduced the disk space required to store the embeddings of proteins threefold by projecting 1024-dimensional vectors from ProtT5 onto 128 dimensions while improving performance (Table 1). This renders new queries (inference) more efficient and enables scaling up to very large lookup sets. (4) Interpretability: knowing from which protein an annotation was transferred might help users to benefit more from a certain prediction than just the prediction itself. For instance, knowing that an unnamed query protein shares all CATH levels with a particular glucocorticoid receptor might suggest some functional implications helping to design future experiments.

Conclusions

Embeddings from protein Language Models (pLMs) extract the information learned by these models from unlabeled protein sequences. Embedding-based Annotation Transfer (EAT) replacing the proximity in sequence space used by homolog-based inference (HBI) by that in embedding space is already competitive with traditional alignment methods in the task to transfer CATH annotations from a template protein with experimental CATH annotations to an unlabeled query protein. Although not quite reaching the performance of advanced profile-profile searches by HMMer for all four CATH levels, the best embeddings surpassed HMMer for two of the four levels (C and A). When optimizing embeddings through contrastive learning for the goal of transferring CATH annotations, EAT using these new embeddings consistently outperformed all sequence comparison techniques tested. Importantly, this higher performance was reached at a fraction (EAT=0.0008s vs HBI=1.5s) of the costs in terms of computing time. Although the new embeddings optimized through contrastive learning for CATH did not improve performance for a completely different task, namely the prediction of subcellular location in ten classes, the CATH-optimized solution did also not perform significantly worse. Remarkably, just like HBI, the performance of EAT using the optimized ProtTucker embeddings was proportional to family size with increased accuracy for larger families.

Acknowledgements

Thanks primarily to Tim Karl, but also to Guy Yachdav & Laszlo Kajan (all TUM) for invaluable help with hardware and software; to Inga Weise (TUM) for support with many other aspects of this work. This work was supported by the Bavarian Ministry of Education through funding to the TUM and by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium für Bildung und Forschung), by two grants from BMBF (031L0168 and program "Software Campus 2.0 (TUM) 2.0" 01IS17049) as well as by a grant from Deutsche Forschungsgemeinschaft (DFG-GZ: RO1320/4-1). Last, but not least, thanks to all those who maintain public databases in particular Steven Burley (PDB, Rutgers), Alan Bridge (Swiss-Prot, SIB, Lausanne), Alex Bateman (UniProt, EBI Hinxton) and their crews, and to all experimentalists who enabled this analysis by making their data publicly available.

References

1. Bepler, T. and Berger, B. (2021) Learning the protein language: Evolution, structure, and function. *Cell Syst*, **12**, 654-669 e653.
2. Van der Maaten, L. and Hinton, G. (2008) Visualizing data using t-SNE. *Journal of machine learning research*, **9**.
3. Das, S., Sillitoe, I., Lee, D., Lees, J.G., Dawson, N.L., Ward, J. and Orengo, C.A. (2015) CATH FunFHMMer web server: protein functional annotations using functional family assignments. *Nucleic Acids Res*, **43**, W148-153.
4. Sillitoe, I., Bordin, N., Dawson, N., Waman, V.P., Ashford, P., Scholes, H.M., Pang, C.S.M., Woodridge, L., Rauer, C., Sen, N. *et al.* (2021) CATH: increased structural coverage of functional space. *Nucleic Acids Research*, **49**, D266—D273.
5. Elnaggar, A., Heinzinger, M., Dallago, C., Rehawi, G., Wang, Y., Jones, L., Gibbs, T., Feher, T., Angerer, C., Steinegger, M. *et al.* (2021) ProtTrans: Towards Cracking the Language of Life's Code Through Self-Supervised Learning. *MACHINE INTELLIGENCE*, **14**, 30.
6. Finn, R.D., Clements, J. and Eddy, S.R. (2011) HMMER web server: interactive sequence similarity searching. *Nucleic Acids Res*, **39**, W29-37.
7. Rives, A., Meier, J., Sercu, T., Goyal, S., Lin, Z., Liu, J., Guo, D., Ott, M., Zitnick, C.L., Ma, J. *et al.* (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, **118**.
8. Orengo, C.A. and Taylor, W.R. (1996) SSAP: Sequential structure alignment program for protein structure comparison. *Methods in Enzymology*, **266**, 617-635.
9. Taylor, W.R. and Orengo, C.A. (1989) A holistic approach to protein structure alignment. *Protein Eng.*, **2**, 505-519.
10. Steinegger, M. and Soding, J. (2017) MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat Biotechnol*, **35**, 1026-1028.
11. Steinegger, M., Mirdita, M. and Söding, J. (2019) Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat Methods*, **16**, 603-606.
12. Orengo, C.A., Flores, T.P., Taylor, W.R. and Thornton, J.M. (1993) Identification and classification of protein fold families. *Protein Engineering*, **6**, 485-500.
13. Das, S., Lee, D., Sillitoe, I., Dawson, N.L., Lees, J.G. and Orengo, C.A. (2015) Functional classification of CATH superfamilies: a domain-based approach for protein function annotation. *Bioinformatics*, **31**, 3460-3467.
14. Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, pp. 4171—4186.
15. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W. and Liu, P.J. (2020) Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, **21**, 1-67.
16. Sonnhammer, E.L. and Kahn, D. (1994) Modular arrangement of proteins as inferred from analysis of homology. *Protein Sci*, **3**, 482-492.
17. Sonnhammer, E.L., Eddy, S.R. and Durbin, R. (1997) Pfam: a comprehensive database of protein domain families based on seed alignments. *Proteins: Structure, Function, and Genetics*, **28**, 405-420.
18. Yona, G., Linial, N. and Linial, M. (1999) ProtoMap: automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins: Structure, Function, and Genetics*, **37**, 360-378.
19. Bateman, A., Birney, E., Durbin, R., Eddy, S.R., Finn, R.D. and Sonnhammer, E.L. (1999) Pfam 3.1: 1313 multiple alignments and profile HMMs match the majority of proteins. *Nucleic Acids Research*, **27**, 260-262.
20. Gough, J. and Chothia, C. (2002) SUPERFAMILY: HMMs representing all proteins of known structure. SCOP sequence searches, alignments and genome assignments. *Nucleic Acids Research*, **30**, 268-272.
21. Wu, C.H., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z.Z., Ledley, R.S., Lewis, K.C., Mewes, H.W., Orcutt, B.C. *et al.* (2002) The Protein Information Resource: an integrated public resource of functional annotation of proteins. *Nucleic Acids Research*, **30**, 35-37.
22. Pandit, S.B., Bhadra, R., Gowri, V.S., Balaji, S., Anand, B. and Srinivasan, N. (2004) SUPFAM: a database of sequence superfamilies of protein domains. *BMC Bioinformatics*, **5**, 28.
23. Holm, L. and Sander, C. (1994) The FSSP database of structurally aligned protein fold families. *Nucleic Acids Research*, **22**, 3600-3609.
24. Murzin, A.G. (1996) Structural classification of proteins: new superfamilies. *Current Opinion in Structural Biology*, **6**, 386-394.
25. Orengo, C.A., Michie, A.D., Jones, S., Jones, D.T., Swindells, M.B. and Thornton, J.M. (1997) CATH - a hierarchic classification of protein domain structures. *Structures*, **5**, 1093-1108.
26. Todd, A.E., Orengo, C.A. and Thornton, J.M. (2001) Evolution of function in protein superfamilies, from a structural perspective. *Journal of Molecular Biology*, **307**, 1113-1143.
27. Yona, G. and Levitt, M. (2002) Within the twilight zone: a sensitive profile-profile comparison tool based on information theory. *Journal of Molecular Biology*, **315**, 1257-1275.
28. Doolittle, R.F., Feng, D.-F., Johnson, M.S. and McClure, M.A. (1986) Origins and evolutionary relationships of retroviruses. *Q. Rev. Biol.*, **64**, 1-30.
29. Rost, B. (1999) Twilight zone of protein sequence alignments. *Protein Engineering*, **12**, 85-94.
30. Rost, B. (1997) Protein structures sustain evolutionary drift. *Folding & Design*, **2**, S19-S24.
31. Mika, S. and Rost, B. (2003) UniqueProt: creating representative protein sequence sets. *Nucleic Acids Research*, **31**, 3789-3791.
32. Rost, B. (2002) Enzyme function less conserved than anticipated. *Journal of Molecular Biology*, **318**, 595-608.
33. Mika, S. and Rost, B. (2006) Protein–protein interactions more conserved within species than across species. *PLoS Computational Biology*, **2**, e79.
34. Nehrt, N.L., Clark, W.T., Radivojac, P. and Hahn, M.W. (2011) Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS Comput Biol*, **7**, e1002073.
35. Smith, T.F. and Waterman, M.S. (1981) Identification of common molecular subsequences. *Journal of Molecular Biology*, **147**, 195-197.
36. Pearson, W.R. and Lipman, D.J. (1988) Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, **85**, 2444-2448.
37. Sander, C. and Schneider, R. (1991) Database of homology-derived structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Genetics*, **9**, 56-68.
38. Higgins, D.G., Bleasby, A.J. and Fuchs, R. (1992) CLUSTAL V: improved sofware for multiple sequence alignment. *Computer Applications in Biological Science*, **8**, 189-191.
39. Thompson, J., Higgins, D. and Gibson, T. (1994) CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, **22**, 4673-4690.
40. Sjölander, K., Karplus, K., Brown, M.P., Hughey, R., Krogh, A., Mian, I.S. and Haussler, D. (1996) Dirichlet mixtures: a method for improving detection of weak but significant protein sequence homology. *Computer Applications in Biological Science*, **12**, 327-345.
41. Altschul, S.F., Madden, T.L., Schaeffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped Blast and PSI-Blast: a new generation of protein database search programs. *Nucleic Acids Research*, **25**, 3389-3402.
42. Eddy, S.R. (1998) Profile hidden Markov models. *Bioinformatics*, **14**, 755-763.

43. Jaroszewski, L., Rychlewski, L. and Godzik, A. (2000) Improving the quality of twilight-zone alignments. *Protein Science*, **9**, 1487-1496.
44. Sadreyev, R. and Grishin, N. (2003) COMPASS: a tool for comparison of multiple protein alignments with assessment of statistical significance. *Journal of Molecular Biology*, **326**, 317-336.
45. Edgar, R.C. and Sjolander, K. (2004) COACH: profile-profile alignment of protein families using hidden Markov models. *Bioinformatics*, **20**, 1309-1318.
46. Wang, G. and Dunbrack, R.L., Jr. (2004) Scoring profile-to-profile sequence alignments. *Protein Sci*, **13**, 1612-1626.
47. Soding, J. (2005) Protein homology detection by HMM-HMM comparison. *Bioinformatics*, **21**, 951-960.
48. Sievers, F., Wilm, A., Dineen, D., Gibson, T.J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Soding, J. *et al.* (2011) Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol Syst Biol*, **7**, 539.
49. Remmert, M., Biegert, A., Hauser, A. and Soding, J. (2012) HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment. *Nat Methods*, **9**, 173-175.
50. Przybylski, D. and Rost, B. (2007) Consensus sequences improve PSI-BLAST through mimicking profile-profile alignments. *Nucleic Acids Research*, **35**, 2238-2246.
51. Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O. and Ofran, Y. (2003) Automatic prediction of protein function. *Cellular and Molecular Life Sciences*, **60**, 2637-2650.
52. Rost, B. and Sander, C. (1992) Jury returns on structure prediction. *Nature*, **360**, 540.
53. Rost, B. (1996) PHD: predicting one-dimensional protein structure by profile based neural networks. *Methods in Enzymology*, **266**, 525-539.
54. Rost, B. and Sander, C. (1993) Prediction of protein secondary structure at better than 70% accuracy. *Journal of Molecular Biology*, **232**, 584-599.
55. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zidek, A., Potapenko, A. *et al.* (2021) Highly accurate protein structure prediction with AlphaFold. *Nature*.
56. Baek, M., Dimaio, F., Anishchenko, I., Dauparas, J., Ovchinnikov, S., Lee, G.R., Wang, J., Cong, Q., Kinch, L.N., Schaeffer, R.D. *et al.* (2021) Accurate prediction of protein structures and interactions using a three-track neural network. *Science*, **373**, 871-876.
57. Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018) Deep contextualized word representations. *arXiv*, **arXiv:1802.05365**.
58. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A. *et al.* (2020) Language Models are Few-Shot Learners. *arXiv*.
59. Alley, E.C., Khimulya, G., Biswas, S., AlQuraishi, M. and Church, G.M. (2019) Unified rational protein engineering with sequence-based deep representation learning. *Nature Methods*, **16**, 1315-1322.
60. Heinzinger, M., Elnaggar, A., Wang, Y., Dallago, C., Nechaev, D., Matthes, F. and Rost, B. (2019) Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, **20**, 723.
61. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. and Rives, A. (2020) Transformer protein language models are unsupervised structure learners. *bioRxiv*, 2020.2012.2015.422761.
62. Madani, A., McCann, B., Naik, N., Shirish Keskar, N., Anand, N., Eguchi, R.R., Huang, P. and Socher, R. (2020) ProGen: Language Modeling for Protein Generation. *arXiv*.
63. Ofer, D., Brandes, N. and Linial, M. (2021) The language of proteins: NLP, machine learning & protein sequences. *Computational and Structural Biotechnology Journal*, **19**, 1750-1758.
64. Bepler, T. and Berger, B. (2019), *Seventh International Conference on Learning Representations*. 2019/02/22 ed.
65. Marquet, C., Heinzinger, M., Olenyi, T., Dallago, C., Erckert, K., Bernhofer, M., Nechaeev, D. and Rost, B. (2021) Embeddings from protein language models predict conservation and variant effects. *Human Genetics*.
66. Weißenow, K., Heinzinger, M. and Rost, B. (2021) Protein language model embeddings for fast, accurate, alignment-free protein structure prediction. *bioRxiv*, 2021.2007.2031.454572.
67. Stärk, H., Dallago, C., Heinzinger, M. and Rost, B. (2021) Light Attention Predicts Protein Location from the Language of Life. *bioRxiv*, 2021.2004.2025.441334.
68. Littmann, M., Heinzinger, M., Dallago, C., Weissenow, K. and Rost, B. (2021) Protein embeddings and deep learning predict binding residues for various ligand classes. *bioRxiv*.
69. Littmann, M., Heinzinger, M., Dallago, C., Olenyi, T. and Rost, B. (2021) Embeddings from deep learning transfer GO annotations beyond homology. *Sci Rep*, **11**, 1160.
70. Littmann, M., Bordin, N., Heinzinger, M., Schütze, K., Dallago, C., Orengo, C. and Rost, B. (2021) Clustering FunFams using sequence embeddings improves EC purity *Bioinformatics*.
71. Asgari, E. and Mofrad, M.R. (2015) Continuous Distributed Representation of Biological Sequences for Deep Proteomics and Genomics. *PLoS One*, **10**, e0141287.
72. Vig, J., Madani, A., Varshney, L.R., Xiong, C., Socher, R. and Rajani, N.F. (2020) BERTology Meets Biology: Interpreting Attention in Protein Language Models. *arXiv*.
73. Villegas-Morcillo, A., Makrodimitris, S., van Ham, R.C.H.J., Gomez, A.M., Sanchez, V. and Reinders, M.J.T. (2021) Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function. *Bioinformatics*, **37**, 162-170.
74. Hamid, M.-N. and Friedberg, I. (2019) Identifying antimicrobial peptides using word embedding with deep recurrent neural networks. *Bioinformatics*, **35**, 2009-2016.
75. Becker, S. and Hinton, G.E. (1992) Self-organizing neural network that discovers surfaces in random-dot stereograms. *Nature*, **355**, 161-163.
76. Bromley, J., Bentz, J.W., Bottou, L., Guyon, I., LeCun, Y., Moore, C., Säckinger, E. and Shah, R. (1993) Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, **7**, 669-688.
77. Le-Khac, P.H., Healy, G. and Smeaton, A.F. (2020) Contrastive representation learning: A framework and review. *IEEE Access*.
78. Sillitoe, I., Cuff, A.L., Dessailly, B.H., Dawson, N.L., Furnham, N., Lee, D., Lees, J.G., Lewis, T.E., Studer, R.A., Rentzsch, R. *et al.* (2013) New functional families (FunFams) in CATH to improve the mapping of conserved functional sites to 3D structures. *Nucleic Acids Res*, **41**, D490-498.
79. Fox, N.K., Brenner, S.E. and Chandonia, J.-M. (2014) SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic acids research*, **42**, D304-D309.
80. Li, C.-C. and Liu, B. (2019) MotifCNN-fold: protein fold recognition based on fold-specific features extracted by motif-based convolutional neural networks. *Briefings in Bioinformatics* **21**, 2133-2141.
81. Liu, B., Li, C.-C. and Yan, K. (2020) DeepSVM-fold: protein fold recognition by combining support vector machines and pairwise sequence similarity scores generated by deep learning networks. *Briefings in bioinformatics*, **21**, 1733-1741.
82. Chen, J., Guo, M., Wang, X. and Liu, B. (2018) A comprehensive review and comparison of different computational methods for protein remote homology detection. *Brief Bioinform*, **19**, 231-244.
83. O'Donoghue, S.I., Schafferhans, A., Sikta, N., Stolte, C., Kaur, S., Ho, B.K., Anderson, S., Procter, J., Dallago, C., Bordin, N. *et al.* (2021) SARS-CoV-2 structural coverage map reveals viral protein assembly, mimicry, and hijacking mechanisms. *Molecular Systems Biology* **12**, in press.
84. Burley, S.K., Berman, H.M., Bhikadiya, C., Bi, C., Chen, L., Di Costanzo, L., Christie, C., Dalenberg, K., Duarte, J.M., Dutta, S. *et al.* (2019) RCSB Protein Data Bank: biological macromolecular structures enabling research and education in fundamental biology, biomedicine, biotechnology and energy. *Nucleic Acids Research*, **47**, D464-D474.

85. Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020), *International conference on machine learning*. PMLR, pp. 1597-1607.
86. Almagro Armenteros, J.J., Sonderby, C.K., Sonderby, S.K., Nielsen, H. and Winther, O. (2017) DeepLoc: prediction of protein subcellular localization using deep learning. *Bioinformatics*, **33**, 3387-3395.
87. Steinegger, M., Mirdita, M. and Soding, J. (2019) Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat Methods*, **16**, 603-606.
88. The UniProt Consortium. (2021) UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res*, **49**, D480-D489.
89. Hochreiter, S. and Schmidhuber, J. (1997) Long short-term memory. *Neural Computation*, **9**, 1735-1780.
90. Hermans, A., Beyer, L. and Leibe, B. (2017) In defense of the triplet loss for person re-identification. *arXiv preprint arXiv:1703.07737*.
91. Kingma, D.P. and Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
92. Buchan, D.W., Shepherd, A.J., Lee, D., Pearl, F.M., Rison, S.C., Thornton, J.M. and Orengo, C.A. (2002) Gene3D: structural assignment for whole genes and genomes using the CATH domain structure database. *Genome Res*, **12**, 503-514.
93. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. and Dubourg, V. (2011) Scikit-learn: Machine learning in Python. *the Journal of machine Learning research*, **12**, 2825-2830.
94. Gorodkin, J. (2004) Comparing two K-category assignments by a K-category correlation coefficient. *Computational biology and chemistry*, **28**, 367-374.
95. Peng, L., Oganesyan, V., Damschroder, M.M., Wu, H. and Dall'Acqua, W.F. (2011) Structural and functional characterization of an agonistic anti-human EphA2 monoclonal antibody. *J Mol Biol*, **413**, 390-405.
96. Himanen, J.P., Goldgur, Y., Miao, H., Myshkin, E., Guo, H., Buck, M., Nguyen, M., Rajashankar, K.R., Wang, B. and Nikolov, D.B. (2009) Ligand recognition by A-class Eph receptors: crystal structures of the EphA2 ligand-binding domain and the EphA2/ephrin-A1 complex. *EMBO Rep*, **10**, 722-728.
97. Webb, E.C. (1992) *Enzyme Nomenclature 1992. Recommendations of the Nomenclature committee of the International Union of Biochemistry and Molecular Biology*. 1992 ed. Academic Press, New York.
98. Dallago, C., Schuetze, K., Heinzinger, M., Olenyi, T., Littmann, M., Lu, A.X., Yang, K.K., Min, S., Yoon, S., Morton, J.T. *et al.* (2021) Learned embeddings from deep learning to visualize and predict protein sets. *Curr Protoc*, **1**, e113.
99. Sillitoe, I., Dawson, N., Lewis, T.E., Das, S., Lees, J.G., Ashford, P., Tolulope, A., Scholes, H.M., Senatorov, I., Bujan, A. *et al.* (2019) CATH: expanding the horizons of structure-based functional annotations for genome sequences. *Nucleic Acids Res*, **47**, D280-D284.
100. Jensen, L.J., Gupta, R., Blom, N., Devos, D., Tamames, J., Kesmir, C., Nielsen, H., Staerfeldt, H.H., Rapacki, K., Workman, C. *et al.* (2002) Prediction of human protein function from post-translational modifications and localization features. *Journal of Molecular Biology*, **319**, 1257-1265.
101. Nair, R. and Rost, B. (2005) Mimicking cellular sorting improves prediction of subcellular localization. *Journal of Molecular Biology*, **348**, 85-100.
102. Kernytsky, A. and Rost, B. (2009) Using genetic algorithms to select most predictive protein features. *Proteins: Structure, Function, and Bionformatics*, **75**, 75-88.
103. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W. and Lipman, D.J. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, **25**, 3389-3402.
104. Dessailly, B.H., Nair, R., Jaroszewski, L., Fajardo, J.E., Kouranov, A., Lee, D., Fiser, A., Godzik, A., Rost, B. and Orengo, C. (2009) PSI-2: Structural genomics to cover protein domain family space. *Structure*, **17**, 869-881.
105. Finkelstein, A.V. and Reva, B.A. (1991) A search for the most stable folds of protein chains. *Nature*, **351**, 497-499.
106. Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996), *Kdd*, Vol. 96, pp. 226-231.