ProstT5: Bilingual Language Model for Protein Sequence and Structure

Michael Heinzinger[1,*,‡], Konstantin Weissenow[1,*], Joaquin Gomez Sanchez[1], Adrian Henkel[1], Martin Steinegger[2,3] & Burkhard Rost[1,4]

1 School of Computation, Information, and Technology (CIT), Department of Informatics, Bioinformatics & Computational Biology, TUM (Technical University of Munich)
2 School of Biological Sciences & 3 Artificial Intelligence Institute, SNU (Seoul National University)
4 Institute for Advanced Study (TUM-IAS), Lichtenbergstr. 2a, 85748 Garching/Munich, Germany & TUM School of Life Sciences Weihenstephan (TUM-WZW), Alte Akademie 8, Freising, Germany
* contributed equally, ‡corresponding author: mheinzinger@rostlab.org

Abstract

Advanced Artificial Intelligence (AI) enabled large language models (LLMs) to revolutionize Natural Language Processing (NLP). Their adaptation to protein sequences spawned the development of powerful protein language models (pLMs). Concurrently, AlphaFold2 broke through in protein structure prediction. For the first time, we can now systematically and comprehensively explore the dual nature of proteins that act and exist as three-dimensional (3D) machines and evolve in linear strings of one-dimensional (1D) sequences. Here, we leverage pLMs to simultaneously model both modalities by combining 1D sequences with 3D structure in one generic model. For this, we encode protein structures as token sequences using the 3Di-alphabet introduced by Foldseek. The resulting "structure-sequence" representation is processed by a pLM to extract features and patterns. Toward this end, we constructed a non-redundant dataset from AlphaFoldDB and fine-tuned an existing pLM (ProtT5) to translate between 3Di and amino acid sequences. As a proof-of-concept for our novel approach, dubbed Protein structure-sequence T5 (ProstT5), we showed improved performance for subsequent prediction tasks, and for “inverse folding”, namely the generation of novel protein sequences adopting a given structural scaffold (“fold”). Our work showcased the potential of pLMs to tap into the information-rich protein structure revolution fueled by AlphaFold2. It paves the way for the development of tools optimizing the integration of this vast 3D structure data resource, opening new research avenues in the post AlphaFold2 era. We released our model at https://github.com/mheinzinger/ProstT5 .

Keywords: protein language model, protein design, transfer learning, artificial intelligence (AI)

Abbreviations & Glossary: **1D**, one-dimensional (string such as secondary structure); **3D**, three-dimensional (coordinates); **3Di**, 1D-strings representing protein 3D structure (taken from Foldseek [1]); **AFDB**, AlphaFold Protein Structure Database; **AI**, Artificial Intelligence; **CATH**, hierarchical classification of protein 3D structures in Class, Architecture, Topology and Homologous superfamily; **CNN**: convolutional neural network; **EAT**, Embedding-based Annotation Transfer; embeddings fixed-size vectors derived from pre-trained pLMs; **LLM**, large language model; **ML**, machine learning (here we drop the distinction between ML and AI considering existing delineations more or less arbitrary for our ends); **pLM**: protein Language Model; **SOTA**: state-of-the-art.

Introduction

Large language models (LLMs) such as Transformers [2] have revolutionized Natural Language Processing (NLP) and have culminated in ChatGPT and GPT4 affecting the daily life of millions. Adapting these techniques to protein sequences by equating words with amino acids and sentences with protein sequences started a wealth of new, powerful tools to model protein sequences [3]–[11]. The success of these protein Language Models (pLMs) builds heavily on the radical rethinking of how to best leverage evolutionary information from large but unlabeled data. Instead of searching for evolutionary related proteins in large sequence databases, pLMs extract meaningful features directly from protein sequences. The *knowledge* acquired by pLMs is readily transferable to subsequent protein prediction tasks. For general-purpose pLMs, this so-called transfer learning succeeds for many aspects of protein prediction, including, for function prediction: gene ontology [12], transport signals [13], binding residues [14], or subcellular location [15], [16], and for protein structure prediction: 2D [17] and 3D structure [6], fold classification [18], [19], or intrinsically disordered regions [20], [21]. The same knowledge extracted by the pLM, can also be queried for protein design [22]–[24], dynamic optimization [25]–[27] or the inference of drug-target interactions [28].

Concurrent with pLMs, *AlphaFold2* [29] largely solved the protein structure prediction problem. By July 2023 accurate structure predictions (AFDB [30]) are available for over 200 million protein sequences in UniProt [31]. This revolution opens up exciting possibilities to explore the dual nature of proteins - from their one-dimensional (1D) amino acid sequences to their unique three-dimensional (3D) structures.

Here, we propose leveraging pLMs to simultaneously model both modalities (1D and 3D). First, we encoded 3D structures as 1D strings (of tokens) to make them amenable to LLM techniques. Towards this end, we utilize the 3Di-alphabet introduced by the 3D comparison tool Foldseek [1]. Essentially, 3Di transliterates 3D coordinates into 1D strings of 20 letters, with one letter describing each residue (sequence position) in a protein (the number is deliberately the same as that of natural amino acids). This allows plugging-in highly optimized sequence search algorithms to compare 3D structures [32]. The same conversion allows feeding sequences of 3Di into a pLM (Fig. 1A). Besides learning to extract features from 3D structure (Fig. 1C), our solution invites switching between both modalities, i.e., translating from sequence to structure, and *vice versa* (Fig. 1B/D). This opens new scientific avenues toward, e.g., inverse folding, structure-guided mutation effect prediction or *in silico* alignment generation.

To quickly summarize our contributions:

- Available resources: adding large and diverse dataset consisting of proteins with high-quality 3D structure predictions from Alphafold2 which is publicly available[1,2]
- Approach and available resources: fine-tuning existing encoder-decoder pLM (ProtT5, [4]) to translate between protein structure (3Di) and sequence (amino acids) and making checkpoints publicly available[3].
- Result: coarse-grained structure representation from 3Di tokens suffices to create new protein sequences (*inverse folding*).

[1] PDB with 3D coordinates: zotero link coming soon
[2] CSV with 3Di sequences: https://huggingface.co/datasets/adrianhenkel/lucidprots_full_data
[3] Model checkpoint: https://huggingface.co/Rostlab/ProstT5

- Result: generating a structure-sequence (3Di) from its amino-acid-counterpart suffices to detect proteins with very diverged sequence and similar structure not reachable by traditional sequence-based alignment methods.
- Result: ProstT5 embeddings (hidden states) capture aspects of protein structures beyond the base model (ProtT5).

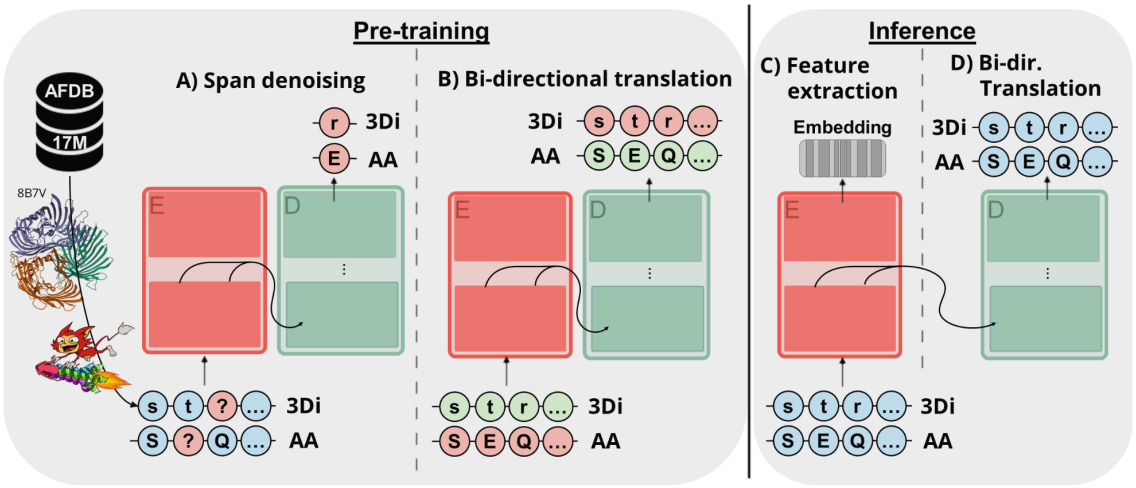

Fig. 1: Sketch of ProstT5. Pre-training: Foldseek [1] transferred protein 3D coordinates into 3Di tokens, i.e., 1D descriptions of 3D structure that assign each residue in a protein into one of twenty states. We used 17 million (17M) high-quality, non-redundant and diverse 3D predictions from AFDB [30]. An existing pLM with encoder-decoder architecture (here, ProtT5 [4]) was leveraged as an already pre-trained starting point for translating between 1D sequence (amino acids, AA) and 3D structure (3Di). Firstly, we applied the original pre-training objective of ProtT5 (span-based denoising) to both, AAs and 3Di, to teach the model the new 3Di tokens while avoiding catastrophic forgetting of AAs (Panel A). Secondly, the resulting model was further trained to translate between AAs and 3Di and vice versa (Panel B). The final model, ProstT5 (Protein structure T5) extracts the information in its internal embeddings that can be input into downstream applications. This includes established feature extraction using only the encoder [4] (Panel C), or, bi-directional translation (Panel D) either from AAs to 3Di ("folding") or from 3Di to AAs ("inverse folding").

Results

ProstT5 pre-training. The proteins in our ProstT5 data splits that we derived from [33] were on average shorter than proteins in the PDB (Fig. 2A; average in PDB [34]: 255 residues, compared to 206-238 (test-train). The amino acid (AA) distribution, however, was similar between our sets of predicted structures and the PDB with experimental structures (Fig. 2D). This was in stark contrast to the distribution of 3Di-tokens which exhibited a severe class imbalance towards few over-represented tokens, in particular for the three states *v*, *d*, and *p* (lower-case letter for distinction to AAs). These three tokens exceeded 50% of all residues in our dataset. This imbalance was also more pronounced for our data than for the PDB at large. The interpretation of this trend is made difficult by the data-driven way in which the 3Di tokens were learnt in the first place. To remediate this problem, we correlated 3Di tokens derived from experimental structures in the PDB with the three secondary structure elements (Fig. 2B). This revealed a clear preference of some 3Di-tokens towards helix (H: v,l), strand (E: e,i,k,t,w,y) or other (L: d,p). In fact, 60% (12 out of the 20) of the 3Di-tokens had a clear preference (>70%) for one of the three secondary structure states (H, E, L).

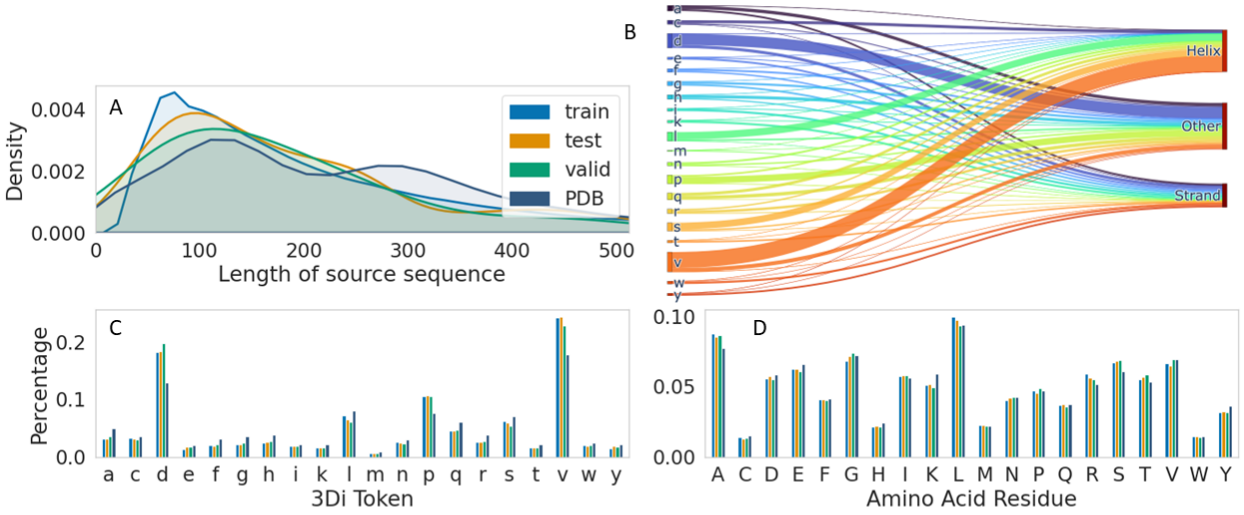

Fig. 2: Dataset analysis. We compared basic properties (A: sequence length, D: amino acid composition, and C: 3Di-distribution) of the proteins in our data sets (blue: training, green: validation, orange: test set) with experimentally resolved proteins (dark blue: PDB [34]). The PDB data also revealed the relation between 3Di-tokens (panel B: left: 3Di states in lower-case for distinction to amino acids; right: secondary structure states, i.e., helix, strand and other). While the amino acid distribution was similar between predicted (AFDB [30]) and experimental (PDB) structures (D), some 3Di-tokens (d, v, p) were clearly more over-represented in AlphaFold2 predictions than in the PDB. The Sankey diagram (B), revealed those tokens mostly related to helix (3Di: v) or other (*loop*; 3Di: d and p). In contrast, 3Di tokens accounting mostly for strands (a, w, y) appeared more frequently in the PDB than in our data. Proteins in our data also tended to be slightly shorter than proteins in PDB, with average lengths of 206-238 (test-train) and 255, respectively.

We used the set *train17M* to expand ProtT5's [4] original pre-training objective (span-based denoising [35]) to cover both, AA- and 3Di-sequences (Fig. 1), which effectively led to a set with 2*17M=34M samples (Fig. 1A). Once the loss on the validation set started to plateau, the actual training on translating between AAs and 3Di and *vice versa* was started (Fig. 1B). Without clear signs of convergence (loss still decreased albeit at decreasing speed), training was stopped due to the decreasing tradeoff between compute/energy required for any further improvement (potentially sacrificing improvements for saving resources).

Bi-linguality improves structure encoding. After finishing both pre-training phases (Fig. 1A/B) on set *train17M*, we benchmarked the resulting pLM ProstT5 on representative protein prediction tasks. As before [4], we began with secondary structure prediction as proxy. As usual for pLMs, we extracted the hidden states of the encoder's last layer (Fig. 1C), and used these vectors, dubbed ***embeddings***, as input for the subsequent, 2nd-step supervised prediction tasks. Thanks to ProstT5's bilinguality, we derived embeddings for both AA- and 3Di-sequences. The embeddings were input into a convolutional neural network (CNN) to classify each residue into helix (H), strand (E), or other (L, Fig. 3A). We used *biotrainer* [36] together with FLIP [37] to replicate the training setup of previous work [4],

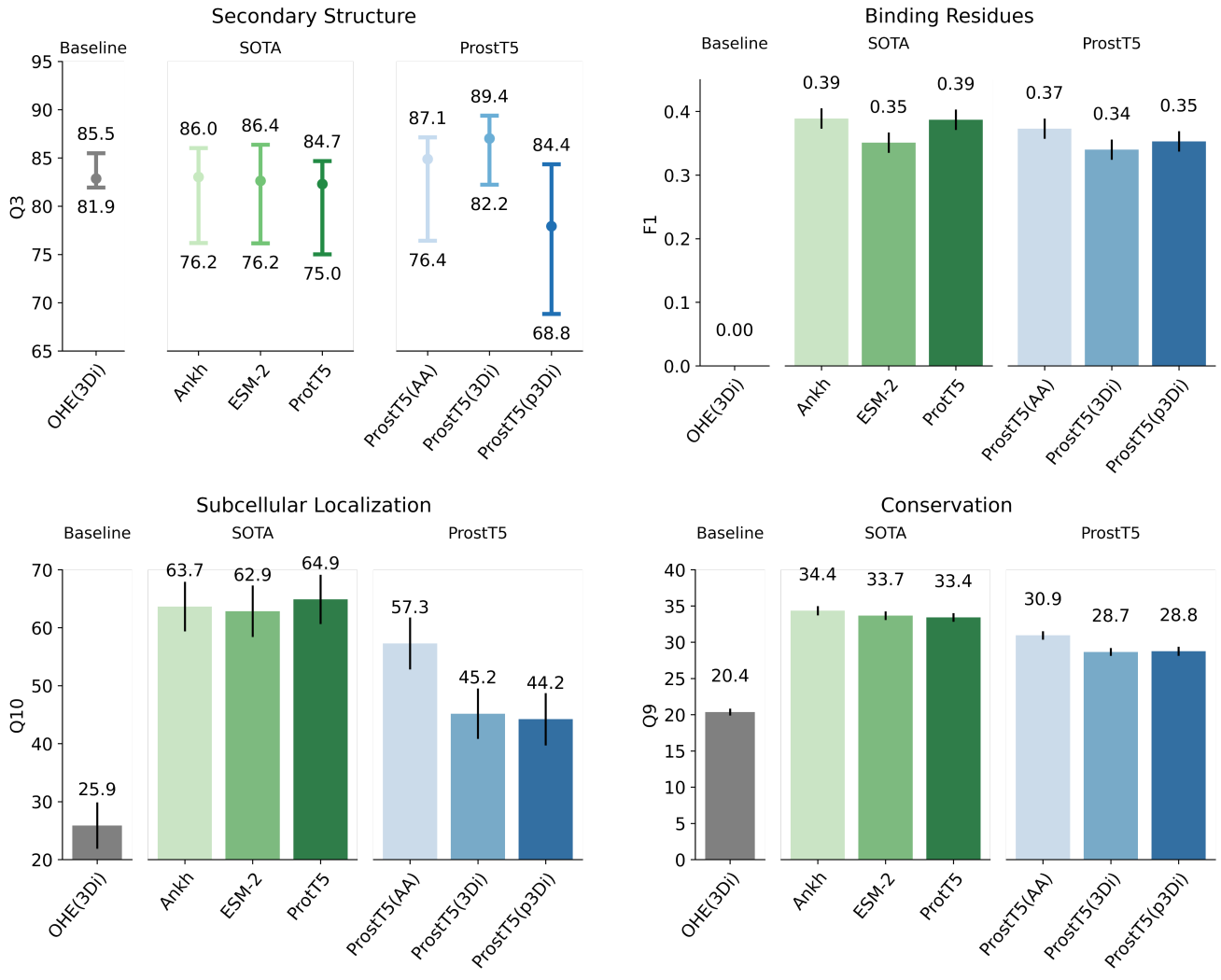

Fig. 3: Protein prediction tasks exclusively inputting pLM embeddings. We probed the relevance of the information learned by ProstT5 through using its embeddings as input to subsequent supervised prediction methods, as introduced before [3]. In particular, we compared ProstT5 to SOTA general purpose pLMs using only amino acid sequences as input (ProtT5 [4], Ankh [10], and ESM-2 (3B) [6]) on four different prediction tasks, namely the per-residue prediction of secondary structure (A; performance: Q3: three-state per-residue accuracy; data sets: middle: *CASP12* [38], lower bar: *CASP14*, upper bar: *NEW364* [4]; note: since each of those sets is supposed to measure performance, the difference between them proxied the error in the estimate), binding residues (B; performance: F1; data: *testSet300* [14]), conservation (D; performance: Q9: nine-state per-residue accuracy; data: [39]), and the per-protein prediction of subcellular location (C; performance: ten-state per-protein accuracy, Q10; data: *setHARD* [15]). As a baseline, we also probed the information content readily available from one-hot-encoded 3Di-sequences (OHE(3Di)). For panels B-D, the bars mark the 95% confidence interval, i.e., ±1.96 * standard errors, estimated via bootstrapping.

i.e., we optimized a two-layer CNN on the *NetSurfP-2.0* training data [40] and measured performance on three test sets (*CASP12* [38], *CASP14* [41] and *NEW364* [4]). The difference in three-state per-residue accuracy (Q3 [42]) between the three sets estimated the error (lower limit: CASP14, upper limit: NEW364; dot between: CASP12). The existing state-of-the-art (SOTA) general–purpose sequence-based pLMs (ProtT5, ESM-2 [6] and Ankh [10]) appeared to perform alike with ProtT5 being slightly worse. Even without leveraging 3Di-information (ProstT5(AA) used only AA input), ProstT5 improved over its base model ProtT5 (Fig. 3A). When inputting 3Di-sequences derived from experimental structures (ProstT5(3Di)), Q3 approached 90% on *NEW364*, getting close to the upper bound given by experimental agreement of identical sequences [43]. Obviously, this implies circularity: using experimental 3D structure to predict secondary structure. Indeed, when using predicted 3Di-sequences (ProstT5(p3Di)), Q3 dropped below the base model ProtT5 and using one-hot-encoded experimental 3Di-sequences (OHE(3Di)) reached performance competitive to SOTA. We also tested whether concatenations of sequence and structure embeddings would capture orthogonal information, and while performance improved in absolute terms, none of the improvement was significant (Fig. S1A).

ProstT5 is not a general-purpose pLM. We tried to avoid *catastrophic forgetting* [44] of what ProtT5 had extracted from pre-training on protein sequences during fine-tuning by continuing de-noising on amino acid sequences and bi-lingual translation. Nevertheless, certain information appeared to have been lost during this process as shown by a clear decrease in subcellular location prediction performance when using amino acid sequences as input (Fig 3C: ProtT5 vs ProstT5(AA)). Other tasks, such as prediction of binding residues (Fig 3B) or conservation (Fig 3D), show a similar trend, albeit with a less severe drop in performance. One-hot-encoding of 3Di (OHE(3Di)) also appeared to be less useful for those tasks. Concatenating amino acid embeddings from ProtT5 and ProstT5, can compensate for this, and can even lead to a numerical improvement as shown for binding residue prediction (Fig S1B).

Table 1: Classification of proteins into CATH hierarchy (folds)*

	Method/Input	*CATH-C*	*CATH-A*	*CATH-T*	*CATH-H*	*Mean*
Baseline	*Random*‡	29±6	9±4	1±2	0±0	10±3
HBI	*MMseqs2*‡	52±7	36±6	29±6	35±6	38±6
EAT unsupervised	ESM-1b‡	79±5	61±6	50±7	57±8	62±7
	Ankh	84±5	69±6	60±7	67±8	70±6
	ProtT5‡	84±5	67±6	57±6	64±8	68±6
	ProstT5(AA)	85±5	74±6	64±6	69±7	73±6
	ProstT5(p3Di)	85±5	71±6	60±7	73±7	72±6
	ProstT5(3Di)	**90±4**	**77±6**	**65±6**	**75±7**	**77±6**
	ProstT5(cat)	88±4	74±6	65±7	74±7	75±6

* Performance: accuracy for predicting CATH [45] levels (from coarse- to fine-grained: C, A, T, H) by transferring annotations from a lookup set to a strictly non-redundant set of queries taken from [18]. The column *Mean* marked the average over the four performance values. Values of methods marked by the symbol ‡ were taken from the literature [18]. Methods: *Baseline*: Random transfer of labels by randomly picking a protein; *HBI (homology-based inference)*: *MMSeqs2* [32] used single sequence search to transfer the annotation of the hit with the lowest *E*-value; *EAT-unsupervised*: embedding-based annotation transfer using the shortest Euclidean distance measured in embedding space of unsupervised pLMs ESM-1b [5], ProtT5 and ProstT5 with different inputs (AA=amino acid sequence input, p3Di=3Di predicted by ProstT5, 3Di=3Di from experimental structures, cat=concatenation of AA and p3Di). Error bars mark the 95% confidence interval with ±1.96 standard errors. Bold numbers mark the highest numerical values. Note that the error bars were so high that only statistically significant were only the differences between *random* and *HBI*, as well as, those between *EAT* and everything else (although the lowest end of pLMs and ProstT5(cat), just differed significantly within the 95% confidence interval.

CATH classification of proteins. We also assessed performance beyond single residues by benchmarking ProstT5-embeddings on the classification of proteins into structural classes (so-called *folds*) using embedding-based annotation transfer (EAT [18]; replacing sequence-similarity by the Euclidean distance in embedding space to transfer annotations from a lookup database to a query protein). To simplify comparability, we replicated existing benchmarks [18] on CATH [45] which uses structural similarity to capture evolutionary and functional relationships of proteins beyond sequence similarity. Given that ProstT5 can generate embeddings from sequence (ProstT5(AA)) or structure input, we benchmarked both (predicted 3Di: ProstT5(p3Di), experimental 3Di: ProstT5(3Di)). All improved over ProtT5, ESM-1b and Ankh (Table 1). Compared to ProtT5, embeddings from amino acids (ProstT5(AA)) mostly improved for the CATH levels of architecture (CATH-A, Table 1) and topology (CATH-T, Table 1), while embeddings from 3Di states (ProstT5(3Di) and ProstT5(p3Di)) improved most for the fine-grained classification of homologous superfamilies (CATH-H). This orthogonal information can be leveraged by concatenating the embeddings from amino acid- and predicted 3Di-sequences (ProstT5(cat)) improving over either method at all CATH levels. We also benchmarked the effect of optimizing ProstT5-embeddings using contrastive learning [18] and while overall performance improved through task-specific optimization, the general trends remained the same (SOM: Table S1).

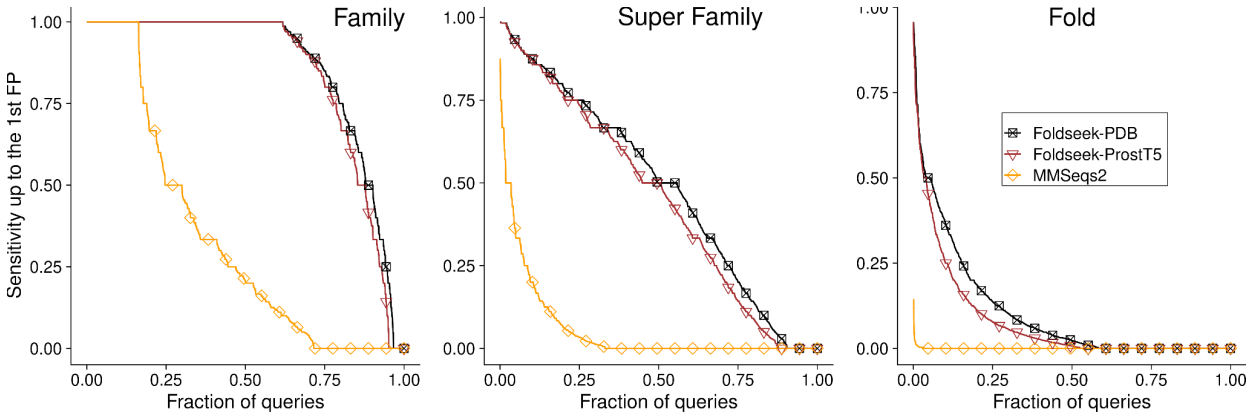

Fig. 4: Successful remote homology detection with predicted 3Di. We replicated the Foldseek benchmark [1] on SCOPe40 [46] using 3Di strings generated by ProstT5 (Foldseek-ProstT5) and compared the sensitivity up to the first false positive (protein with different classification) with the performance of Foldseek on experimental structures (Foldseek-PDB). For all three levels (from most coarse-grained family level on the left, over the superfamily level in the center, to the most fine-grained level of *fold* on the right), ProstT5-predicted 3Di strings sufficed to almost reach the performance of PDB structures while significantly outperforming traditional sequence alignment (*MMSeqs2* [32]).

Remote homology detection succeeded. To evaluate ProstT5 further, we tested its ability to distinguish between proteins with similar and dissimilar structures. First, we computed the sequence similarity of generated 3Di-sequences with 3Di-sequences derived from AFDB for all proteins in our test set. Despite reporting a pairwise similarity (using global sequence alignment [47] with a 3Di-specific substitution matrix [1]) reaching as high as 75%, we struggled to put this into perspective due to a lack of other methods predicting 3Di-tokens from amino acid sequences. We decided to probe the remote homology detection capability of ProstT5 generated 3Di sequences by replicating the Foldseek benchmark. We replaced structure-sequences (3Di) derived from experimental structures in the

Foldseek benchmark (40% non-redundant version of SCOPe [46]) by 3Di strings translated from corresponding amino acid sequences. We compared the sensitivity up to the first false-positive on the level of *family*, *superfamily* and *fold* for Foldseek applied on 3Di-sequences derived from experimental data (Fig. 4: Foldseek-PDB) against the ProstT5-generated 3Di strings (Fig. 4: Foldseek-ProstT5). For reference, we put this performance into perspective of traditional sequence alignment using *MMSeqs2* [32]. Using predicted 3Di states, ProstT5 (ROC-AUC on the level of super-family: 0.45) reached performance levels close to experimental structures (ROC-AUC=0.49) and vastly outperformed traditional sequence-based searches (ROC-AUC=0.06).

***Inverse Folding*: creating new protein sequences with similar structure.** The *bi-lingual* nature of ProstT5 (AA→3Di and 3Di→AA) suggested generating amino acid sequences for a given structure (as described by its 3Di-tokens). As pairs of proteins with diverged sequences may adopt similar 3D structures [48], [49], we measured success in creating new sequences through the similarity in predicted 3D structure between the *groundtruth* (3D predicted by AFDB [30]) and *predictions* (3D structure predicted by ESMFold [6] for ProstT5-generated sequences). As our model assigns probabilities to a sequence of amino acids given some conditioning context (3Di), we used the validation set to compare various free parameters (e.g., beam-search [50], nucleus sampling [51] or top-k sampling [52]) that affect the translation (3Di→AA) and its quality by modulating the probabilities assigned to a specific sequence during sequential decoding (SOM: Table S3). For the final evaluation on the test set, we chose a configuration providing a trade-off between similarity (to the native) in terms of structure and unsurprising sequence (proxied by Kullback-Leibler divergence between the amino acid distribution in UniProt and the sequences generated [53]). We measured structural similarity (ESMFold prediction of generated sequences vs. AlphaFold2 prediction of the native sequence) by three scores: lDDT [54],TM-score [55] and RMSD (as implemented in [55]).

First, we established an upper bound for our performance analysis by comparing ESMFold to AlphaFold2 predictions for the native sequences (Table 2: *Native/ESMFold*). Although ProstT5 generated sequences with, on average, as little as 21% PIDE to the native were predicted to adopt similar structures (average lDDT=72), the *de facto* standard for inverse folding, i.e., graph-based *ProteinMPNN* [56], succeeded in generating sequences close to this upper-bound (lDDT(ProteinMPNN)=77 vs. lDDT(Native/ESMFold)=78). However, the amino acid distribution of ProstT5-generated sequences was closer to the native distribution as measured by entropy (Table 2).

Motivated by the success of ProstT5-predicted 3Di-sequences for remote homology detection (Fig. 4), we also probed whether ProstT5-generated backtranslations (3Di→AA→3Di), provided any indication for the quality of the inverse folding (new sequences for given structure). Towards this end, we correlated the structural similarity (lDDT) of native sequences predicted by AlphaFold2 and ProstT5-generated sequences predicted by ESMFold against sequence similarity between the native 3Di-sequence and the 3Di-sequences generated from the translated AA sequence (Fig. 5A). We observed a high correlation (Spearman's R of 0.64) for what we referred to as the *roundtrip accuracy*, i.e., translating from native 3Di to AAs which were then translated back into 3Di for comparison with the starting point (native 3Di). We further applied this idea to constrain generated sequences, i.e., we generated AA sequences until a *roundtrip accuracy* ≥70% was reached and retained only the candidate with the maximal *roundtrip accuracy*. Even when limiting this to ten for saving resources, we already observed a minor, yet consistent improvement for all metrics (Table 2 - ProstT5(rTrip70), Fig. 5B). Sequences generated by *ProstT5* and *ProteinMPNN* agreed well in their predictions (Fig. 5C, Spearman's R of 0.52). For the proteins in our test set, there was no difference in the inverse folding

performance between those from clusters and those that remained singletons (Fig. 5). We cherry-picked cases for which both models (ProstT5 and ProteinMPNN) generated sequences resulting in high-quality structures (Fig. 6A and 6B) but we also zoomed into cases where either of the models failed (Fig. 6C and Fig. 6D, ProstT5>ProteinMPNN and ProstT5<ProteinMPNN, respectively).

Table 2: Inverse folding comparison*

	Native/ESMFold	*ProteinMPNN*	*ProstT5*	*ProstT5(rTrip70)*
lDDT ↑	0.78±0.01	**0.77±0.01**	0.72±0.01	0.73±0.01
RMSD ↓	2.55±0.01	**2.61±0.01**	2.90±0.01	2.81±0.01
TM-score ↑	0.62±0.02	**0.61±0.02**	0.58±0.02	0.60±0.02
PIDENT	100±0	**29.6±1**	21.9±0.9	22.4±0.9
Entropy ↓	0.13±0.01	0.39 +/- 0.03	0.20±0.01	**0.19±0.01**

* Performance: structural similarity of ESMFold [6] and AlphaFold2 [29] predictions for native (Natural/ESMFold) and generated sequences in our test set. Sequences were generated using ProteinMPNN, ProstT5 and a filtered version of ProstT5 (ProstT5(rTrip70)) which uses the model's own back translation capability to filter by sequence similarity between native 3Di sequences and their counterpart predicted from generated AA sequences (3Di→AA→3Di). We generated AA sequences until either a 3Di back translation sequence similarity of at least 70 or a maximum number of attempts (here: 10) was reached. Single-sequence based ESMFold predictions for generated sequences were compared against the native groundtruth predicted by AlphaFold2 using lDDT [54], RMSD, TM-score [55], percentage pairwise sequence identity (PIDE) and entropy (KL-divergence between the AA distribution in UniProt and the generated sequences). Error bars indicate 95% confidence intervals estimated from 1000 bootstrap samples. Arrows next to metrics indicate whether higher (↑) or lower (↓) values are better. For PIDENT applied to inverse folding, it is not clear whether higher is necessarily better.

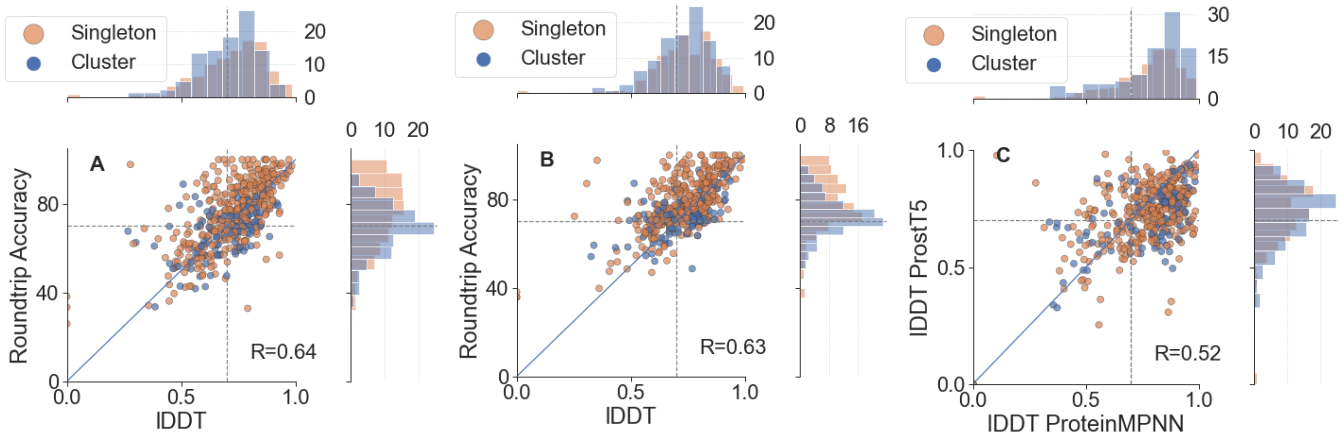

Fig. 5: ProstT5 back translation correlates with inverse folding quality. For each protein in our test set, we used ProstT5 to generate new AA sequences from its native counterpart described by its 3Di sequence (3Di→AA) with the constraint that newly generated sequences share similar structures (measured by lDDT [54]). For the native structures, we took *AlphaFold2* [29] predictions. For the generated sequences, we used single-sequence-based *ESMFold* [6] to avoid bias towards multiple-sequence-alignment. We dubbed the similarity between 3Di sequences derived from native structures and the 3Di sequences predicted by ProstT5 from the newly generated proteins "*Roundtrip accuracy*" (*3Di*→AA→3Di, y-axis in Panel A and B). The Spearman R in panel A reports the correlation of this roundtrip accuracy against the structural similarity (lDDT) between *ESMFold* predictions for mutant sequences and *AlphaFold2* for the native. Panel B: We leveraged this correlation (R=0.64) to let ProstT5 control its mutations by generating AA sequences until either a *Roundtrip accuracy*≥70 was reached or for maximally ten attempts. This constraint slightly decreased the Pearson R (0.64 vs. 0.63) but improved the structural similarity between the newly generated sequences (Table 2). Panel C compared the same ProstT5 generations (constrained by roundtrip accuracy, panel B) with ProteinMPNN [56] in terms of lDDT. There seems to be a common trend towards easy inverse folding targets as both methods showed good agreement in terms of lDDT (Spearman R of 0.52). In all cases we differentiate between test set proteins that are part of a cluster (blue) or not (singletons - orange), i.e., marginal plots show the distribution of both classes as percentage values. Dashed lines mark a lDDT or roundtrip accuracy of 70.

Speed. The time required to generate embeddings is identical for ProtT5 and ProstT5 because the network architecture did not change [4]. Generating embeddings for the human proteome from the Pro(s)tT5 encoder requires around 35m (minutes) or 0.1s (seconds) per protein using batch-processing and half-precision (fp16) on a single RTX A6000 GPU with 48 GB vRAM. The translation is comparatively slow (0.6-2.5s/protein at an average length of 135 and 406, respectively) due to the sequential nature of the decoding process which needs to generate left-to-right, token-by-token. We only used batch-processing with half-precision and left further speed-ups via LLM-specific optimizations [57] to future work.

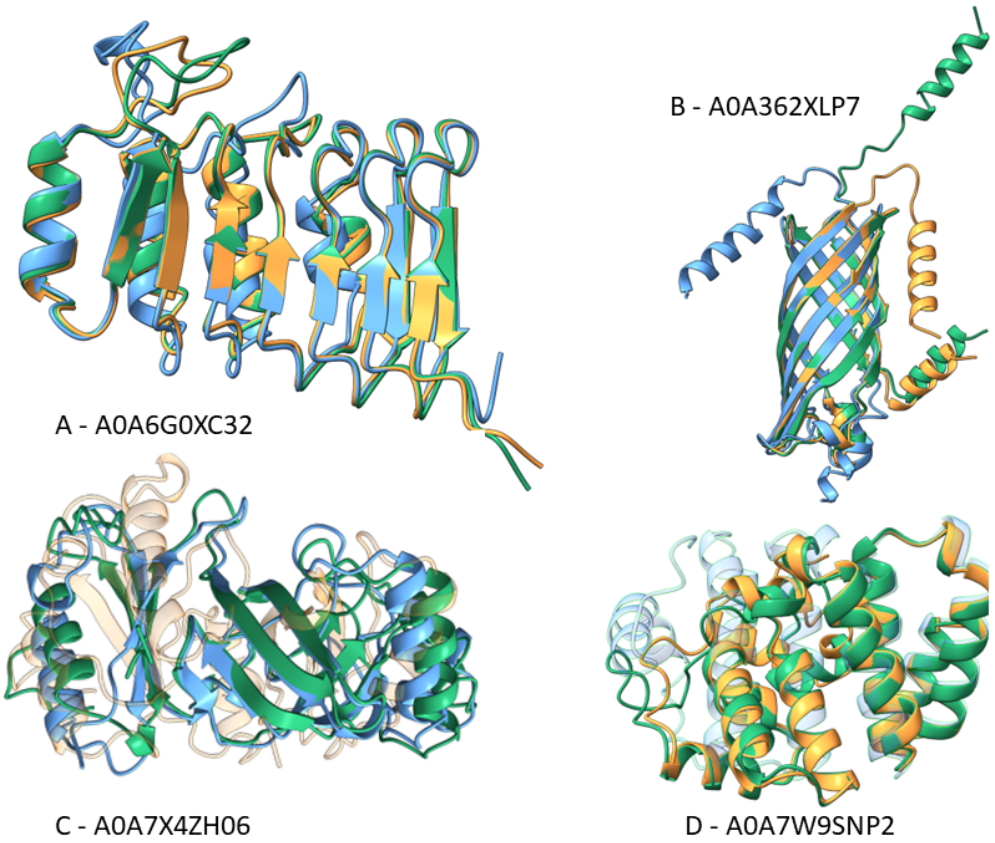

Fig. 6: Inverse folding examples. We manually picked four examples from our test set for which both (A and B), ProstT5 and ProteinMPNN, only ProstT5 (C) or only ProteinMPNN (D) generated sequence with high structure similarity to their natural counterparts. Structures colored in green show AlphaFold2 predictions/groundtruths, blue and orange depict ESMFold predictions of ProstT5- and ProteinMPNN-generated sequences, respectively. We picked examples such that they show diversity in their structural composition (beta-strands and alpha-helices) and their place of action (transmembrane (B) vs soluble (A,C,D)). Both methods can produce proteins that share only little sequence but high structural similarity to their native counterparts (A & B: lDDT of 76-95 or RMSD of 1.1-2.6 at 23-44% sequence similarity) but there are also cases where only either of them succeeds (C: ProstT5(lDDT)=68 vs ProteinMPNN(lDDT)=34; D: ProstT5(lDDT)=56 vs ProteinMPNN(lDDT)=75). For better visibility, we increased transparency for cases with poor structural superposition (C: ProteinMPNN, D: ProstT5).

Discussion

Standing on the shoulders of giants. The avalanche of recent progress in NLP was triggered by the introduction of attention [58] paired with excellently scaling Transformers [2]. Mining this breakthrough requires models with billions of free parameters trained on gigantic data sets. Thus, in computational biology Transformers have been limited to training large language models (LLMs) on amino (AA) and nucleic acid sequences [59]. AlphaFold2 [29] with its over 200 million predictions (AFDB, [30]) has changed this limitation fundamentally.

By enabling scalable structure search at the speed of sequence comparison, Foldseek [1] made a first big step towards leveraging this new information. Foldseek needed to harvest several solutions to make this possible, the arguably most important was to map the 3D coordinate for each residue in a protein structure to one of the 20 so-called 3Di tokens through a vector-quantized variational autoencoder (VQ-VAE [60]). The resulting conversion from structure (3D) to sequence (1D) allows Foldseek to leverage highly optimized sequence comparison tools [32] for structure search. Thanks to foldseek's sensitivity of detecting remote homologs using 3Di, we postulate that those structure-sequences contain enough information to train a language model on translating from structure to sequence.

Sampling from proteins' Janus face. Janus is the double-faced Roman god of amongst others duality and transitions. Here, we combined the two aspects of proteins (structure: AFDB→3Di and sequence) by fine-tuning the sequence-based protein language model (pLM) ProtT5 [4] on 17M proteins, each with L (protein length) doublets of 3Di- and AA-sequences. Similar to the combination of images or movies with text [61], we merged protein sequences (AA) and structures (3Di) into a single model, a new pLM, dubbed ProstT5. This increased the flexibility for querying knowledge stored in the weights of the model via human interaction going beyond embedding extraction. Despite notable exceptions [7], [22], established pLMs are mostly limited to feature extraction from encoder-style Transformers [4]–[6]. Instead, T5's encoder-decoder structure allows ProstT5 to additionally act as a conditional language model that assigns probabilities to a sequence of tokens given some conditioning context. In our case, we conditioned on structure (3Di) to generate sequences (AAs) and *vice versa*. This translation capability opens many possibilities. For instance, it can be used to replace 3Di sequences derived from 3D structures by 3Di sequences translated by ProstT5 from AA sequence. In fact, our predicted 3Di sequences reached levels of identifying the structural similarity between protein pairs with very diverse sequences (often coined *remote homology detection*) approaching that of using the experimental structures (Fig. 4). ProstT5 reached this impressively high level without any usage of structure, i.e., without even having to use structure predictions. Inverting the translation task (from 3Di→AA), ProstT5 successfully accomplished *inverse folding*, i.e., identified unknown proteins with similar structures and different sequences (Table 2, Fig. 5, Fig. 6). Albeit not reaching the *de facto* standard current method toward this end, ProteinMPNN [56], which uses a graph-based, message-passing network for this task, our proof-of-concept already reached an average lDDT of 72 and even outperformed ProteinMPNN for some cases (Fig. 5C, 6C). This, at least, suggested some complementary of the two approaches.

More interesting applications emerge when combining both translation directions in series. We showcased the usefulness of stringing together both directions by using ProstT5 to assess the quality of its own predictions generated during inverse folding (Fig. 5B). First ProstT5 generated novel AA sequences conditioned on adopting any desired structural template (here given by the 3Di sequence from AlphaFold2). Next, we used the same model, ProstT5, to translate the novel AA sequences back

into 3Di sequences that we matched to the starting point (native 3Di or structural template) using sequence similarity applied to 3Di (3Di→AA→3Di). If the generated AA sequence still adopts the same structure, this should yield high similarity between the source structure (3Di) and the structure (3Di) predicted by ProstT5 from the generated AA sequence. Indeed, this similarity that we dubbed *roundtrip accuracy* correlated well (Spearman R of 0.64) with the structural similarity (lDDT) of 3D predictions of generated and native sequences (Fig. 5A). When giving the model ten attempts to reach a minimal *roundtrip accuracy* (≥70), we observed a minor, yet consistent improvement on all metrics (Table 2).

Traditional embedding extraction. When merging protein sequences (AA) and structures (3Di) in a single model, we hypothesized such multi-modal pre-training to increase the usefulness of amino acid embeddings as input to subsequent structure-related prediction tasks. Indeed, for secondary structure prediction and the classification of proteins in similar structural classes, ProstT5 embeddings improved over other classification methods with (Fig. 3A, Table S1, Fig. S1A) and without supervision (Table 1). Yet, not all protein prediction tasks benefited directly from coupling 3Di and AA. In fact, more function-related tasks performed even slightly worse (Fig. 3B-D). For location prediction (Fig. 3C), this might partially be explained by decreased 3D structure prediction precision at the ends of the sequence including the N-terminal signal peptides that are crucial for subcellular localization. For other tasks such as binding or conservation prediction, the drop in prediction performance is less pronounced, i.e., insignificant for conservation, and might be explained by repurposing some of the model's capacity for structure-specific information. However, we could compensate for this drop by a simple concatenation of the embeddings from the original ProtT5 and the AA-based part of ProstT5 (ProstT5(AA)). In particular, in predicting binding or residues to other ligands (excluding other proteins, [14]), the simple concatenation performed best (Fig. S1B: ProtT5 + ProstT5(AA) vs. SOTA in Fig. 3B). Although this increase remained statistically insignificant.

Limitations. By building a highly non-redundant, and diverse data set consisting only of high-quality 3D structures, we managed to maximize the amount of sequence-structure space covered by a minimum of proteins. While this approach toward reducing redundancy avoided excessive bias towards large families that exists in both PDB [34] and sequence databases [31], our filter might have introduced other aspects of bias. If so, ProstT5 might have picked up such bias as common to any other LLM that may amplify bias in its training data [62]. Most important might be aspects of bias pertaining to structure predictions and how we filter and represent those (3Di). For instance, filtering the AFDB for predictions with high pLDDT removes most intrinsically disordered proteins [63] and enriches short [64], well-structured, preferably helical proteins [65] (Fig. 1). On the one hand, this is intended because we want to avoid training on non-structured proteins. On the other hand, the high class imbalance of 3Di-tokens (2 of 20 3Di tokens accounted for half of all residues, Fig. 1), impaired training as some proteins were represented by a single 3Di-token, e.g., most all-helical proteins contained only one 3Di-token (d). We tried to address this by removing extreme outliers with more than 95% of the residues being assigned to the same 3Di-token. Nevertheless, class imbalance remained high, so future work will most likely benefit from improved and potentially more balanced 3D→1D mappings.

Outlook. LLMs had been phenomenally successful when the release of GPT4 advanced another magnitude. Our proof-of-principle solution rendering protein structures directly amenable to LLMs will benefit from future NLP improvements, in particular, from better sampling from conditional language models [66]. Fine-grained control over this sampling might improve *in silico* creations of predicted

multiple sequence alignments (MSAs) when sampling from the ProstT5-improved cycle from single sequence to potentially multiple structure(s) to multiple sequences (AA→3Di**s**→AA**s**). Constant speed-ups of LLMs [57] also decrease translation latency which in turn renders the translation from AA to 3Di an attractive alternative to searching large metagenomic datasets at the sensitivity of structure comparison while avoiding the overhead of actually having to predict 3D structures. Deriving embeddings from structures will also expand the power of embedding-based alignments [67], [68], and retrieval Transformers [69]. Our proposed integration of 3D information into pLMs may only constitute the first step towards building truly multi-model pLMs that capture the multiple facets of protein structure, function and evolution. Expanding ProstT5 from being bilingual into becoming truly polyglot by adding other, potentially more function-centric, conditioning tags such as Gene Ontology terms [70] might be the first step toward future advanced pLMs. Recent developments on increasing the context length of LLMs [71] will allow future work to condition on full-length UniProt entries, or even all papers that mention a certain protein. This way, LLMs can integrate any knowledge existing for a certain protein today.

Conclusion

Over the last two years, we have been witnessing how the protein structure revolution ignited by *AlphaFold2* enables groundbreaking scientific discoveries. However, integrating the wealth of information arising from this new, gigantic data resource demands the development of novel tools to optimally leverage the potential. *Foldseek* is a first leap paving the way for new avenues in the era post *AlphaFold2*. ProstT5 describes a proof-of-concept that exemplifies how language modeling techniques and Transformers can be used for tapping into this information-rich goldmine of 3D structures.

Methods

ProstT5 data set

Our translation from 1D amino acid sequences to 1D structure sequences (3Di tokens) began with a recently published [33] clustered version of the AlphaFold Protein Structure Database (AFDB [30]). This dataset was created by two-step clustering and one step of quality filtering.

(i) MMseqs2 [32] clustered 214 million (M) UniprotKB [31] protein sequences from AFDB such that no pair had over 50% pairwise sequence identity (PIDE) at 90% sequence overlap. For each of the 52M resulting clusters, the protein with the highest predicted local distance difference test (pLDDT) score [29] was selected as the representative.

(ii) Foldseek [1] clustered the 52M representatives further into 18.8M clusters enforcing a pairwise minimal E-value of 10^-2 at 90% sequence overlap for each Foldseek (structural) alignment. From those 18.8M, 2.8M clusters contained two or more members (16M were singletons, i.e., no other protein could be aligned using the procedure above). To avoid bias towards exotic outliers and to increase the training set, we expanded each cluster, maximally, by its 20 most diverse members using HHBlits [72]. This expansion increased from 2.8M clusters to 18.6M proteins leading to a total set size of 34.6M proteins when combined with the singletons.

(iii) Finally, we added three filtering steps: remove (a) low-quality structure predictions (pLDDT<70), (b) short proteins (length<30), and (c) proteins with highly repetitive 3Di-strings (>95% of assigned to single 3Di token). The final training set contained 17M proteins (4.3M singletons with respect to the original 16M). As we are translating into both directions, i.e., from 3Di to amino acids (AA) and vice versa, this corresponded to 34M training samples. From those, we randomly split off 1.2k proteins for validation and 1.2k for final testing while ensuring that clusters were moved to either of the sets such that all members of one cluster always end up in the same split. After keeping only representatives to avoid bias towards clusters, we ended up with 474 proteins for validation and final testing each.

For comparison of the final dataset to PDB [34] (Fig. 2), we downloaded PDB's "ss_dis.txt" file (28.06.2023) which simplifies extraction of (un-)resolved residues and their secondary structure elements. Corresponding 3Di sequences were extracted from the PDB version provided by Foldseek. For the analysis, we removed any entry that a) could not be matched between both versions, b) had a length mismatch in the Foldseek 3Di string and the AA sequence length in PDB, c) removed all unresolved residues and d) transformed 8-state secondary structure as defined by DSSP [73] to 3-states by mapping {G,H,I}→*Helix*, {B,E}→*Strand*, {-,T,S}→*Other*.

ProstT5 training

ProstT5 pre-training. To learn translating between structure (3Di) and sequence (AA), we chose the already pre-trained protein language model (pLM) ProtT5 (marked *ProtT5-XL-U50* in the original publication [4]). We could have started from scratch but wanted to save resources by building on top of existing knowledge. ProtT5 is based on the sequence-to-sequence model T5 [35] trained on reconstructing corrupted tokens from 2.1B metagenomic protein sequences in the BFD (Big Fantastic Database, [74]) and 40M protein sequences from Uniref50 [75], a version of UniProt [31] clustered at 50% sequence similarity. We chose this pLM, because the original Transformer consisting of an encoder-decoder architecture that is used by (Prot)T5 lends itself to translation tasks. During training, the encoder learns to parse the source language while the decoder learns to produce meaningful output in the target language conditioned on the encoder output.

Learning new 3Di tokens. In a first step, we expanded the existing ProtT5 vocabulary consisting of the 20 standard amino acids and 128 special tokens introduced for span-corruption [35] by the 20 3Di tokens. To avoid token collision during tokenization of amino acids and 3Di strings (which use identical letters), we cast all 3Di sequences to lower-case before using them as input to the model. Additionally, we added two special tokens ("<fold2AA>", "<AA2fold>") which are prepended to the input to indicate the directionality of the translation. More specifically, *<fold2AA>* instructs the model to translate from the input of a 3Di structure sequence into an amino acid sequence, while *<AA2fold>* indicates the inverse direction, i.e., instructs the model to generate a 3Di structure sequence from an amino acid input. With this setup, we continued the ProtT5 pre-training on *train17M*, i.e., reconstructing corrupted tokens from non-corrupted context, but now simultaneously using protein sequences (amino acids) and structures (3Di). By training on both modalities simultaneously we tried to avoid catastrophic forgetting which will become important later when translating in both directions. The 3B (3*10^9) free parameters of ProtT5 were fine-tuned with a learning rate of 10^-3 for 10 days on 8 Nvidia A100 each with 80GB vRAM using a batch-size of 16. As pLMs benefit from training on many samples (or manyfold repetitions of the same samples [4]), we increased throughput by limiting sequence lengths to 256 (truncating longer sequences) and using DeepSpeed (stage-2) [76], gradient accumulation steps (5 steps), mixed half-precision (bf16, [77]) and PyTorch2.0's torchInductor compiler [78]. Thereby, the model trained on 102M (1.02*10^8) samples corresponding to about three epochs (1 epoch = presentation of each sample once) over the 34M protein (structure) sequences in *train17M*.

Learning bi-directional translation. In a last step, the resulting model, which can now "read" 3Di tokens, was trained to translate between sequences of amino acids and 3Di *structure states*. Both directions were trained simultaneously with the prefixes *<fold2AA>* and *<AA2fold>* indicating the directionality of the translation. The translation was trained on the same machine (8 A100 á 80GB vRAM) and setup (DeepSpeed stage-2, gradient accumulation, bf16, torchInductor) as before. However, we changed the learning rate to 10^-5 for the initial 100K (10^5) steps (6 days) on sequences with ≤256 residues (again truncating longer sequences), and increased to 512 for another 600K (6*10^5) steps (20 days). While increasing sequence length, we had to lower batch-size (from 16 to 6) which we compensated for by increasing the number of gradient accumulation steps (from 1 to 4). In total, we trained for around 700K (7*10^5) steps (about 4 epochs) on set *train17M*. We dubbed the final model *ProstT5* for *Protein structure sequence T5*.

Evaluation benchmarks

Transfer learning. One way to establish the value of pLMs is to use the vector representations they learned, referred to as the embeddings (Fig. 1C and Fig 1 in [79]), as input to subsequent supervised prediction tasks [3]. Ultimately, this is the concept of transfer learning which, in the first step, requires substantial computing resources to create general purpose pLMs. In the second step, the embeddings from these pLMs, i.e. the essence of what they learnt, are input to any supervised prediction task of interest. In this logic, the performance of some standardized, non-optimized set of 2nd step supervised prediction tasks becomes the best way to evaluate the validity of the pLM (here ProstT5) as a general-purpose model. As redundancy between training and testing is crucial even in relative comparisons [80], this aspect makes it so difficult to adequately evaluate protein prediction on known

data [38], [81], which is why we focused on a limited number of standard benchmarks that we tried to reproduce as closely as possible to existing work using biotrainer [36] and FLIP [37].

Supervised learning: per-residue secondary structure. Given its proven track record to benchmark pLMs [3]–[6], [10] and to ease comparison to other methods, we replicated previous benchmarks [4]. To predict properties of single tokens (here: single amino acids, dubbed residues when joined in proteins), we used the training set published with NetSurfP-2.0 [40] (3-state secondary structure using DSSP [73]: helix, strand, and other). We benchmarked using three public test data sets, namely CASP12 [38], CASP14 and NEW364 [4]. We report performance on CASP12 and NEW364 for comparability to existing methods but those sets allow for indirect information leakage as they overlap with AlphaFold2 training data (and thus with our set *train17M*). We used the same convolutional neural network and hyperparameters as described in detail elsewhere [4].

Supervised learning: per-residue: binding. For predicting whether a ligand (small molecule, metal ion, or DNA/RNA; essentially only excluding protein-protein interactions) is binding to a specific residue in a protein, we replicated training (*DevSet1014*) and testing (*TestSet300*) of a recent method of [14] (also using a two-layer CNN; with the same training parameters). For simplicity, we skipped the more fine-grained evaluation of different binding types focusing on the binary binding/not.

Supervised learning: per-residue: conservation. One surprising recent result established that pLMs can reliably predict the conservation of a residue in a protein family without using any multiple sequence alignment defining a family as input [39]. Here, we replicated the training and evaluation used before [39]. In brief, we used *ConSurf10k* [39], [82], a 25% non-redundant dataset derived from high-quality PDB [34] structures, to train a two-layer CNN to classify each residue into one of nine conservation classes (1=highly variable, 9=highly conserved) defined by ConSurf-DB [82].

Supervised learning: per-protein: subcellular location. For predicting features of entire proteins, we classified each protein into one of ten subcellular locations [15]. More specifically, we used the *DeepLoc* training data [83] to train a light-attention network [15] which we evaluated using a 20% non-redundant test set (*setHARD*). We copied setup and hyperparameters from the literature [15].

Supervised learning: per-protein: superfamily classification. We used CATH [45] to classify proteins into superfamilies replicating previous work [18]. In brief, we used the CATH hierarchy (v4.3) which classifies three-dimensional (3D) protein structures at the four levels Class (most distantly related pairs), Architecture, Topology and Homologous superfamily (most similar pairs). As described in [18], we used contrastive learning to train a two-layer feed-forward-neural network to learn a new embedding space in which proteins with increasing overlap in the hierarchy are pulled closer together while others get pushed further away. A hit at lower CATH-levels could be correct if all previous levels were correctly predicted. Due to the varying number of samples at different CATH-levels, performance measures not normalized by background numbers could be higher for lower levels.

Unsupervised classification: per-protein: superfamily prediction. Besides serving as input to 2nd-step supervised training, embeddings can also be used without further modifications for classifications directly. One solution is the so-called embedding-based annotation transfer (EAT - [12],

[18]) that proceeds as follows. Given a protein K of experimentally known annotation and another protein Q with missing annotation: if the Euclidean distance between the two is below some empirical threshold T (if D(embedding(Q),embedding(K))<T): transfer annotation of K to Q. Arguably, EAT generalizes what is used for most database annotations and is often referred to as homology-based inference (HBI) that copies annotations when Q is sufficiently sequence similar to K. In order to classify/predict, the annotation of the most similar protein in the lookup database is transferred to the query protein.

We used a previously published dataset [18] to probe how well ProstT5 per-protein embeddings (average-pooling over the residue-embeddings derived fixed-length vector for each protein irrespective of its length) alone distinguished between different levels of structural similarity. Instead of applying supervision and contrastive learning, we used EAT to transfer annotations from a lookup set to a 20% non-redundant test set. For protein sequences, this task corresponded to something as daring as using 20d-vectors with the amino acid composition of two proteins to establish whether or not those have similar structure. Again, we computed the accuracy as the fraction of correct hits for each CATH-level.

Besides EAT, ProstT5's translation capabilities open new possibilities for unsupervised benchmarking of the information stored in the model. For example, one can use ProstT5 to translate from sequence to structure and use predicted structure (3Di) for remote homology detection.

***Folding*: from sequence to structure.** An alternative way to extract information from ProstT5 is to predict 3Di sequences from amino acid sequences (Fig. 1D) and use the predicted 3Di sequences as input to *Foldseek* to search for (structurally) related proteins. Towards this end, we reproduced the *Foldseek* benchmark replacing 3Di strings derived from experimental data by ProstT5 predictions. In brief, Foldseek performs an all-against-all search of SCOPe40 (SCOPe 2.01 clustered at 40% pairwise sequence identity [46]) to measure the fraction of finding members of the same SCOPe family, superfamily and fold (true-positive (TP) matches) for each query out of all possible correct matches until the first false positive (FP: match to different fold). *Foldseek*, when used on PDB structures, uses C-alpha backbone information to rerank hits, which slightly improves performance in the SCOPe40 benchmark. Since no C-alpha information is available when using ProstT5 to generate 3Di strings, we disabled this feature to evaluate ProstT5 but activated it when running Foldseek for fair comparison.

***Inverse folding*: from structure to sequence.** The term *inverse folding* [84], [85] has been applied to the challenge of finding all the protein sequences that adopt a particular 3D structure (in the past loosely referred to as "the fold"). By design ProstT5 appears ideally suited to address this challenge by simply inverting the direction of the translation, i.e., by reconstructing amino acid sequences from 3Di-sequences. Toward this end, we considered sequence similarity to be a weak measure for success as there are many, potentially very dissimilar, sequences that still adopt the same structure. Instead, we used structural similarity, i.e., Local Distance Difference Test (lDDT, [54]), template-modeling score (TM-score [55]) and root-mean-square-deviation (RMSD). To obtain 3D coordinates, we predicted structures using ESMFold [6] for all protein sequences created by ProstT5 and compared these predictions to the AFDB *groundtruth* for the native sequence.

Sampling from translations. In contrast to traditional classification, so-called conditional language models assign probabilities to a sequence of words given some conditioning context. Here, we either generated amino acid sequences conditioned upon structure (3Di sequences) or vice versa. As there

are multiple techniques to sample from this process, each with individual hyperparameters, we compared different sampling strategies [50]–[52]. All comparisons and resulting decisions were based on the validation set while the final testing set was only used to report final performance of the hyperparameter combination that worked best on the validation set.

AA→3Di (folding): When translating from amino acid (AA) sequences to 3Di-sequences, we used the sequence similarity (below) between the groundtruth 3Di sequences from the AFDB and the ESMFold predictions from the generated 3Di sequences to compare different sampling strategies. We used global Needleman-Wunsch alignment [47] as implemented in biotite [86] together with the 3Di substitution matrix from Foldseek [1] to compute sequence similarities. We compared all combinations of the following parameters (SOM: Table S2): a) number of beams ∈ [0,3] [50], b) temperature ∈ [0.8, 1.0, 1.2], c) top-p ∈ [0.85, 0.9, 0.95] [51], d) top-k ∈ [3,6] [52], and, e) repetition penalty ∈ [1.0, 1.2, 1.4]. For all analysis presented here, we used the following huggingface generation configuration because it achieved the highest sequence similarity: "do_sample": True, "num_beams": 3, "top_p" : 0.95, "temperature" : 1.2, "top_k" : 6, "repetition_penalty" : 1.2.

3Di→AA (inverse folding): To reconstruct amino acid sequences (AA) from 3Di-sequences, we again compared all combinations of the following sampling parameters (SOM: Table S3): a) number of beams ∈ [0,3] [50], b) temperature ∈ [0.8, 0.9, 1.0, 1.1, 1.2], c) top-p ∈ [0.85, 0.9, 0.95] [51], d) top-k ∈ [3,6] [52], and, e) repetition penalty ∈ [1.0, 1.2, 1.4]. However, this time, we defined success in terms of a combination of the lDDT (comparing ESMFold predictions of our generated sequences against AFDB) and naturalness as proxied by relative entropy (or Kullback-Leibler divergence) between the amino acid distribution in UniProt and the generated sequences [53]. This resulted in the following configuration: "do_sample": True, "top_p" : 0.85, "temperature" : 1.0, "top_k" : 3, "repetition_penalty" : 1.2.

Acknowledgements

Thanks primarily to the team at the Leibniz Supercomputing Center (LRZ, Munich), especially to Juan Durillo Barrionuevo, for providing access and guidance which enabled large-scale GPU training. Thanks also to Chris Dallago (Nvidia), Adam Grzywaczewski (Nvidia) and Noelia Ferruz (IBMB) for helpful discussions. Thanks also to Tim Karl (TUM) for invaluable help with hardware and software and to Nicola Bordin (UCL) for providing access to the non-redundant PDB structures of CATH. MH and BR were supported by the Bavarian Ministry of Education through funding to the TUM, by a grant from the Alexander von Humboldt foundation through the German Ministry for Research and Education (BMBF: Bundesministerium für Bildung und Forschung), and 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, to the team at EMBL-EBI who teamed-up with Deepmind to make AlphaFold2 3D structure predictions for hundreds of millions of proteins in UniProt publicly available to everyone.

References

[1] M. van Kempen *et al.*, “Fast and accurate protein structure search with Foldseek,” *Nat. Biotechnol.*, pp. 1–4, May 2023, doi: 10.1038/s41587-023-01773-0.
[2] A. Vaswani *et al.*, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
[3] M. Heinzinger *et al.*, “Modeling aspects of the language of life through transfer-learning protein sequences,” *BMC Bioinformatics*, vol. 20, no. 1, p. 723, Dec. 2019, doi: 10.1186/s12859-019-3220-8.
[4] A. Elnaggar *et al.*, “ProtTrans: Toward Understanding the Language of Life Through Self-Supervised Learning,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7112–7127, Oct. 2022, doi: 10.1109/TPAMI.2021.3095381.
[5] A. Rives *et al.*, “Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences,” *Proc. Natl. Acad. Sci.*, vol. 118, no. 15, p. e2016239118, Apr. 2021, doi: 10.1073/pnas.2016239118.
[6] Z. Lin *et al.*, “Evolutionary-scale prediction of atomic-level protein structure with a language model,” *Science*, vol. 379, no. 6637, pp. 1123–1130, Mar. 2023, doi: 10.1126/science.ade2574.
[7] A. Madani *et al.*, “ProGen: Language Modeling for Protein Generation,” *bioRxiv*, p. 2020.03.07.982272, Mar. 2020, doi: 10.1101/2020.03.07.982272.
[8] N. Ferruz, S. Schmidt, and B. Höcker, “ProtGPT2 is a deep unsupervised language model for protein design,” *Nat. Commun.*, vol. 13, no. 1, Art. no. 1, Jul. 2022, doi: 10.1038/s41467-022-32007-7.
[9] K. K. Yang, N. Fusi, and A. X. Lu, “Convolutions are competitive with transformers for protein sequence pretraining.” bioRxiv, p. 2022.05.19.492714, Feb. 23, 2023. doi: 10.1101/2022.05.19.492714.
[10] A. Elnaggar *et al.*, “Ankh ☥ : Optimized Protein Language Model Unlocks General-Purpose Modelling.” bioRxiv, p. 2023.01.16.524265, Jan. 18, 2023. doi: 10.1101/2023.01.16.524265.
[11] B. Chen *et al.*, “xTrimoPGLM: Unified 100B-Scale Pre-trained Transformer for Deciphering the Language of Protein.” bioRxiv, p. 2023.07.05.547496, Jul. 06, 2023. doi: 10.1101/2023.07.05.547496.
[12] M. Littmann, M. Heinzinger, C. Dallago, T. Olenyi, and B. Rost, “Embeddings from deep learning transfer GO annotations beyond homology,” *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Jan. 2021, doi: 10.1038/s41598-020-80786-0.
[13] F. Teufel *et al.*, “SignalP 6.0 predicts all five types of signal peptides using protein language models,” *Nat. Biotechnol.*, vol. 40, no. 7, Art. no. 7, Jul. 2022, doi: 10.1038/s41587-021-01156-3.
[14] M. Littmann, M. Heinzinger, C. Dallago, K. Weissenow, and B. Rost, “Protein embeddings and deep learning predict binding residues for various ligand classes,” *Sci. Rep.*, vol. 11, no. 1, Art. no. 1, Dec. 2021, doi: 10.1038/s41598-021-03431-4.
[15] H. Stärk, C. Dallago, M. Heinzinger, and B. Rost, “Light attention predicts protein location from the language of life,” *Bioinforma. Adv.*, vol. 1, no. 1, p. vbab035, Jan. 2021, doi: 10.1093/bioadv/vbab035.
[16] V. Thumuluri, J. J. Almagro Armenteros, A. R. Johansen, H. Nielsen, and O. Winther, “DeepLoc 2.0: multi-label subcellular localization prediction using protein language models,” *Nucleic Acids Res.*, vol. 50, no. W1, pp. W228–W234, Jul. 2022, doi: 10.1093/nar/gkac278.
[17] K. Weissenow, M. Heinzinger, and B. Rost, “Protein language-model embeddings for fast, accurate, and alignment-free protein structure prediction,” *Structure*, vol. 30, no. 8, pp. 1169-1177.e4, Aug. 2022, doi: 10.1016/j.str.2022.05.001.
[18] “Contrastive learning on protein embeddings enlightens midnight zone | NAR Genomics and Bioinformatics | Oxford Academic.” https://academic.oup.com/nargab/article/4/2/lqac043/6605840 (accessed Jun. 22, 2023).
[19] V. Nallapareddy *et al.*, “CATHe: detection of remote homologues for CATH superfamilies using embeddings from protein language models,” *Bioinformatics*, vol. 39, no. 1, p. btad029, Jan. 2023, doi: 10.1093/bioinformatics/btad029.

[20] D. Ilzhöfer, M. Heinzinger, and B. Rost, “SETH predicts nuances of residue disorder from protein embeddings,” *Front. Bioinforma.*, vol. 2, 2022, Accessed: Jul. 10, 2023. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fbinf.2022.1019597
[21] I. Redl *et al.*, “ADOPT: intrinsic protein disorder prediction through deep bidirectional transformers,” *NAR Genomics Bioinforma.*, vol. 5, no. 2, p. lqad041, Jun. 2023, doi: 10.1093/nargab/lqad041.
[22] G. Munsamy, S. Lindner, P. Lorenz, and N. Ferruz, “ZymCTRL: a conditional language model for the controllable generation of artificial enzymes”.
[23] N. Ferruz, M. Heinzinger, M. Akdel, A. Goncearenco, L. Naef, and C. Dallago, “From sequence to function through structure: Deep learning for protein design,” *Comput. Struct. Biotechnol. J.*, vol. 21, pp. 238–250, Jan. 2023, doi: 10.1016/j.csbj.2022.11.014.
[24] R. Verkuil *et al.*, “Language models generalize beyond natural proteins.” bioRxiv, p. 2022.12.21.521521, Dec. 22, 2022. doi: 10.1101/2022.12.21.521521.
[25] V. Padmakumar, R. Y. Pang, H. He, and A. P. Parikh, “Extrapolative Controlled Sequence Generation via Iterative Refinement.” arXiv, Jun. 07, 2023. doi: 10.48550/arXiv.2303.04562.
[26] B. L. Hie *et al.*, “Efficient evolution of human antibodies from general protein language models,” *Nat. Biotechnol.*, pp. 1–9, Apr. 2023, doi: 10.1038/s41587-023-01763-2.
[27] “A high-level programming language for generative protein design | bioRxiv.” https://www.biorxiv.org/content/10.1101/2022.12.21.521526v1.abstract (accessed Jul. 21, 2023).
[28] R. Singh, S. Sledzieski, B. Bryson, L. Cowen, and B. Berger, “Contrastive learning in protein language space predicts interactions between drugs and protein targets,” *Proc. Natl. Acad. Sci.*, vol. 120, no. 24, p. e2220778120, Jun. 2023, doi: 10.1073/pnas.2220778120.
[29] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, Art. no. 7873, Aug. 2021, doi: 10.1038/s41586-021-03819-2.
[30] M. Varadi *et al.*, “AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models,” *Nucleic Acids Res.*, vol. 50, no. D1, pp. D439–D444, Jan. 2022, doi: 10.1093/nar/gkab1061.
[31] The UniProt Consortium, “UniProt: a worldwide hub of protein knowledge,” *Nucleic Acids Res.*, vol. 47, no. D1, pp. D506–D515, Jan. 2019, doi: 10.1093/nar/gky1049.
[32] M. Steinegger and J. Söding, “MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets,” *Nat. Biotechnol.*, vol. 35, no. 11, Art. no. 11, Nov. 2017, doi: 10.1038/nbt.3988.
[33] I. B. Hernandez *et al.*, “Clustering predicted structures at the scale of the known protein universe.” bioRxiv, p. 2023.03.09.531927, Mar. 10, 2023. doi: 10.1101/2023.03.09.531927.
[34] “RCSB PDB,” Nov. 20, 2020. http://www.rcsb.org/ (accessed Nov. 20, 2020).
[35] C. Raffel *et al.*, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer,” *ArXiv191010683 Cs Stat*, Oct. 2019, Accessed: Mar. 21, 2020. [Online]. Available: http://arxiv.org/abs/1910.10683
[36] J. G. Sanchez, S. Franz, M. Heinzinger, B. Rost, and C. Dallago, “Standards, tooling and benchmarks to probe representation learning on proteins,” presented at the NeurIPS 2022 Workshop on Learning Meaningful Representations of Life, Nov. 2022. Accessed: Jul. 17, 2023. [Online]. Available: https://openreview.net/forum?id=adODyN-eeJ8
[37] C. Dallago *et al.*, “FLIP: Benchmark tasks in fitness landscape inference for proteins,” presented at the Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2), Jan. 2022. Accessed: Aug. 07, 2022. [Online]. Available: https://openreview.net/forum?id=p2dMLEwL8tF
[38] L. A. Abriata, G. E. Tamò, B. Monastyrskyy, A. Kryshtafovych, and M. Dal Peraro, “Assessment of hard target modeling in CASP12 reveals an emerging role of alignment-based contact prediction methods,” *Proteins Struct. Funct. Bioinforma.*, vol. 86, pp. 97–112, 2018.
[39] C. Marquet *et al.*, “Embeddings from protein language models predict conservation and variant effects,” *Hum. Genet.*, vol. 141, no. 10, pp. 1629–1647, Oct. 2022, doi: 10.1007/s00439-021-02411-y.

[40] M. S. Klausen *et al.*, “NetSurfP-2.0: Improved prediction of protein structural features by integrated deep learning,” *Proteins Struct. Funct. Bioinforma.*, vol. 87, no. 6, pp. 520–527, 2019, doi: 10.1002/prot.25674.
[41] A. Kryshtafovych, T. Schwede, M. Topf, K. Fidelis, and J. Moult, “Critical assessment of methods of protein structure prediction (CASP)—Round XIV,” *Proteins Struct. Funct. Bioinforma.*, vol. 89, no. 12, pp. 1607–1617, 2021, doi: 10.1002/prot.26237.
[42] B. Rost and C. Sander, “Prediction of protein secondary structure at better than 70% accuracy,” *Journal of molecular biology*, vol. 232, no. 2. Elsevier Science, pp. 584–599, 1993.
[43] B. Rost, C. Sander, and R. Schneider, “Redefining the goals of protein secondary structure prediction,” *J. Mol. Biol.*, vol. 235, no. 1, pp. 13–26, Jan. 1994, doi: 10.1016/S0022-2836(05)80007-5.
[44] M. McCloskey and N. J. Cohen, “Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem,” in *Psychology of Learning and Motivation*, G. H. Bower, Ed., Academic Press, 1989, pp. 109–165. doi: 10.1016/S0079-7421(08)60536-8.
[45] I. Sillitoe *et al.*, “CATH: increased structural coverage of functional space,” *Nucleic Acids Res.*, vol. 49, no. D1, pp. D266–D273, Jan. 2021, doi: 10.1093/nar/gkaa1079.
[46] 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.*, vol. 47, no. D1, pp. D475–D481, Jan. 2019, doi: 10.1093/nar/gky1134.
[47] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *J. Mol. Biol.*, vol. 48, no. 3, pp. 443–453, Mar. 1970, doi: 10.1016/0022-2836(70)90057-4.
[48] A. M. Lesk and C. Chothia, “How different amino acid sequences determine similar protein structures: The structure and evolutionary dynamics of the globins,” *J. Mol. Biol.*, vol. 136, no. 3, pp. 225–270, Jan. 1980, doi: 10.1016/0022-2836(80)90373-3.
[49] B. Rost, “Protein structures sustain evolutionary drift,” *Fold. Des.*, vol. 2, pp. S19–S24, Jun. 1997, doi: 10.1016/S1359-0278(97)00059-X.
[50] A. K. Vijayakumar *et al.*, “Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models.” arXiv, Oct. 22, 2018. doi: 10.48550/arXiv.1610.02424.
[51] A. Holtzman, J. Buys, L. Du, M. Forbes, and Y. Choi, “The Curious Case of Neural Text Degeneration.” arXiv, Feb. 14, 2020. Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/1904.09751
[52] A. Fan, M. Lewis, and Y. Dauphin, “Hierarchical Neural Story Generation.” arXiv, May 13, 2018. Accessed: Jun. 22, 2023. [Online]. Available: http://arxiv.org/abs/1805.04833
[53] V. Vacic, V. N. Uversky, A. K. Dunker, and S. Lonardi, “Composition Profiler: a tool for discovery and visualization of amino acid composition differences,” *BMC Bioinformatics*, vol. 8, p. 211, Jun. 2007, doi: 10.1186/1471-2105-8-211.
[54] V. Mariani, M. Biasini, A. Barbato, and T. Schwede, “lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests,” *Bioinformatics*, vol. 29, no. 21, pp. 2722–2728, Nov. 2013, doi: 10.1093/bioinformatics/btt473.
[55] Y. Zhang and J. Skolnick, “Scoring function for automated assessment of protein structure template quality,” *Proteins Struct. Funct. Bioinforma.*, vol. 57, no. 4, pp. 702–710, 2004, doi: 10.1002/prot.20264.
[56] J. Dauparas *et al.*, “Robust deep learning based protein sequence design using ProteinMPNN.” bioRxiv, Jun. 04, 2022. doi: 10.1101/2022.06.03.494563.
[57] Benjamin Lefaudeux and Francisco Massa and Diana Liskovich and Wenhan Xiong and Vittorio Caggiano and Sean Naren and Min Xu and Jieru Hu and Marta Tintore and Susan Zhang and Patrick Labatut and Daniel Haziza, “facebookresearch/xformers.” Meta Research, Jul. 14, 2023. Accessed: Jul. 14, 2023. [Online]. Available: https://github.com/facebookresearch/xformers
[58] D. Bahdanau, K. Cho, and Y. Bengio, “Neural Machine Translation by Jointly Learning to Align and Translate,” *ArXiv14090473 Cs Stat*, May 2016, Accessed: Mar. 26, 2020. [Online]. Available:

http://arxiv.org/abs/1409.0473
[59] H. Dalla-Torre *et al.*, “The Nucleotide Transformer: Building and Evaluating Robust Foundation Models for Human Genomics.” bioRxiv, p. 2023.01.11.523679, Mar. 09, 2023. doi: 10.1101/2023.01.11.523679.
[60] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, “Neural Discrete Representation Learning.” arXiv, May 30, 2018. doi: 10.48550/arXiv.1711.00937.
[61] J.-B. Alayrac *et al.*, “Flamingo: a Visual Language Model for Few-Shot Learning.” arXiv, Nov. 15, 2022. doi: 10.48550/arXiv.2204.14198.
[62] N. Meade, E. Poole-Dayan, and S. Reddy, “An Empirical Survey of the Effectiveness of Debiasing Techniques for Pre-trained Language Models.” arXiv, Apr. 02, 2022. doi: 10.48550/arXiv.2110.08527.
[63] M. Akdel *et al.*, “A structural biology community assessment of AlphaFold2 applications,” *Nat. Struct. Mol. Biol.*, vol. 29, no. 11, Art. no. 11, Nov. 2022, doi: 10.1038/s41594-022-00849-w.
[64] V. Monzon, D. H. Haft, and A. Bateman, “Folding the unfoldable: using AlphaFold to explore spurious proteins,” *Bioinforma. Adv.*, vol. 2, no. 1, p. vbab043, Jan. 2022, doi: 10.1093/bioadv/vbab043.
[65] A. O. Stevens and Y. He, “Benchmarking the Accuracy of AlphaFold 2 in Loop Structure Prediction,” *Biomolecules*, vol. 12, no. 7, Art. no. 7, Jul. 2022, doi: 10.3390/biom12070985.
[66] S. Yao *et al.*, “Tree of Thoughts: Deliberate Problem Solving with Large Language Models.” arXiv, May 17, 2023. doi: 10.48550/arXiv.2305.10601.
[67] L. Pantolini, G. Studer, J. Pereira, J. Durairaj, and T. Schwede, “Embedding-based alignment: combining protein language models and alignment approaches to detect structural similarities in the twilight-zone,” Bioinformatics, preprint, Dec. 2022. doi: 10.1101/2022.12.13.520313.
[68] F. Llinares-López, Q. Berthet, M. Blondel, O. Teboul, and J.-P. Vert, “Deep embedding and alignment of protein sequences,” *Nat. Methods*, vol. 20, no. 1, Art. no. 1, Jan. 2023, doi: 10.1038/s41592-022-01700-2.
[69] C. Ma *et al.*, “Retrieved Sequence Augmentation for Protein Representation Learning.” bioRxiv, p. 2023.02.22.529597, May 23, 2023. doi: 10.1101/2023.02.22.529597.
[70] R. P. Huntley *et al.*, “The GOA database: gene Ontology annotation updates for 2015,” *Nucleic Acids Res.*, vol. 43, no. Database issue, pp. D1057-1063, Jan. 2015, doi: 10.1093/nar/gku1113.
[71] A. Bulatov, Y. Kuratov, and M. S. Burtsev, “Scaling Transformer to 1M tokens and beyond with RMT.” arXiv, Apr. 19, 2023. Accessed: Jul. 14, 2023. [Online]. Available: http://arxiv.org/abs/2304.11062
[72] “HHblits: lightning-fast iterative protein sequence searching by HMM-HMM alignment | Nature Methods.” https://www.nature.com/articles/nmeth.1818 (accessed Jan. 17, 2023).
[73] W. Kabsch and C. Sander, “Dictionary of protein secondary structure: Pattern recognition of hydrogen-bonded and geometrical features,” *Biopolymers*, vol. 22, no. 12, pp. 2577–2637, 1983, doi: 10.1002/bip.360221211.
[74] M. Steinegger, M. Mirdita, and J. Söding, “Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold,” *Nat. Methods*, vol. 16, no. 7, Art. no. 7, Jul. 2019, doi: 10.1038/s41592-019-0437-4.
[75] B. E. Suzek, H. Huang, P. McGarvey, R. Mazumder, and C. H. Wu, “UniRef: comprehensive and non-redundant UniProt reference clusters,” *Bioinformatics*, vol. 23, no. 10, pp. 1282–1288, May 2007, doi: 10.1093/bioinformatics/btm098.
[76] J. Rasley, S. Rajbhandari, O. Ruwase, and Y. He, “DeepSpeed: System Optimizations Enable Training Deep Learning Models with Over 100 Billion Parameters,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, in KDD ’20. New York, NY, USA: Association for Computing Machinery, Aug. 2020, pp. 3505–3506. doi: 10.1145/3394486.3406703.
[77] P. Micikevicius *et al.*, “Mixed Precision Training.” arXiv, Feb. 15, 2018. doi: 10.48550/arXiv.1710.03740.
[78] P. Wu, “PyTorch 2.0: The Journey to Bringing Compiler Technologies to the Core of PyTorch (Keynote),” in *Proceedings of the 21st ACM/IEEE International Symposium on Code Generation and*

Optimization, in CGO 2023. New York, NY, USA: Association for Computing Machinery, Feb. 2023, p. 1. doi: 10.1145/3579990.3583093.

[79] M. Heinzinger and B. Rost, “Opinion piece/Book chapter: Artificial Intelligence learns protein prediction.” OSF Preprints, Jul. 07, 2023. doi: 10.31219/osf.io/n7wfp.

[80] B. Rost, “Enzyme Function Less Conserved than Anticipated,” *J. Mol. Biol.*, vol. 318, no. 2, pp. 595–608, Apr. 2002, doi: 10.1016/S0022-2836(02)00016-5.

[81] P. Radivojac *et al.*, “A large-scale evaluation of computational protein function prediction,” *Nat. Methods*, vol. 10, no. 3, Art. no. 3, Mar. 2013, doi: 10.1038/nmeth.2340.

[82] A. Ben Chorin *et al.*, “ConSurf-DB: An accessible repository for the evolutionary conservation patterns of the majority of PDB proteins,” *Protein Sci.*, vol. 29, no. 1, pp. 258–267, 2020, doi: 10.1002/pro.3779.

[83] J. J. Almagro Armenteros, C. K. Sønderby, S. K. Sønderby, H. Nielsen, and O. Winther, “DeepLoc: prediction of protein subcellular localization using deep learning,” *Bioinformatics*, vol. 33, no. 21, pp. 3387–3395, Nov. 2017, doi: 10.1093/bioinformatics/btx431.

[84] M. J. Sippl, “Calculation of conformational ensembles from potentials of mena force: An approach to the knowledge-based prediction of local structures in globular proteins,” *J. Mol. Biol.*, vol. 213, no. 4, pp. 859–883, Jun. 1990, doi: 10.1016/S0022-2836(05)80269-4.

[85] D. T. Jones, W. R. Taylort, and J. M. Thornton, “A new approach to protein fold recognition,” *Nature*, vol. 358, no. 6381, Art. no. 6381, Jul. 1992, doi: 10.1038/358086a0.

[86] P. Kunzmann and K. Hamacher, “Biotite: a unifying open source computational biology framework in Python,” *BMC Bioinformatics*, vol. 19, no. 1, p. 346, Oct. 2018, doi: 10.1186/s12859-018-2367-z.