Protein language model powers accurate and fast sequence search for remote homology

Wei Liu[1], Ziye Wang[1], Ronghui You[1], Chenghan Xie[1], Hong Wei[2], Yi Xiong[3], Jianyi Yang[4*] and Shanfeng Zhu[1*]

[1]Institute of Science and Technology for Brain-Inspired Intelligence and MOE Frontiers Center for Brain Sciences, Fudan University, Shanghai, 200433, China.
[2]School of Mathematical Sciences, Nankai University, Tianjin, 300071, China.
[3]Department of Bioinformatics and Biostatistics, Shanghai Jiao Tong University, Shanghai, 200240, China.
[4]Ministry of Education Frontiers Science Center for Nonlinear Expectations, Research Center for Mathematics and Interdisciplinary Science, Shandong University, Qingdao, 200240, China.

*Corresponding author(s). E-mail(s): yangjy@sdu.edu.cn; zhusf@fudan.edu.cn;

Abstract

Homologous protein search is one of the most commonly used methods for protein annotation and analysis. Compared to structure search, detecting distant evolutionary relationships from sequences alone remains challenging. Here we propose PLMSearch (**P**rotein **L**anguage **M**odel), a homologous protein search method with only sequences as input. With deep representations from a pre-trained protein language model to predict similarity, PLMSearch can capture the remote homology information hidden behind the sequences. Extensive experiment results show that PLMSearch improves the sensitivity of other sequence search methods by more than threefold, and is comparable to state-of-the-art structure search methods. In particular, unlike traditional sequence search methods, PLMSearch can recall most remote homology pairs with low sequence similarity but sharing similar structures. PLMSearch is freely available at https://issubmission.sjtu.edu.cn/PLMSearch.

1 Introduction

Homologous protein search is a key component of bioinformatics methods used in protein function prediction [1–3], protein-protein interaction prediction [4], and protein-phenotype association prediction [5]. The goal of homologous protein search is to find the homologous protein from the target dataset (generally a large-scale standard dataset) for each query protein. And the target protein with a higher degree of homology should be ranked higher. According to the type of input

data, homologous protein search can be divided into sequence search and structure search.

Due to the low cost and large scale of sequence data, the most widely used homologous protein search methods are based on sequence similarity, such as BLAST [6], MMseqs2 [7], Hh-suite3 [8], and Diamond [9]. These methods are based on the idea that homologous protein pairs with high sequence similarity often share similar molecular and cellular functions and structures. Despite the success of homology inference based on sequence similarity, it remains challenging to detect distant evolutionary relationships from sequences only [10].

Detecting similarity between protein structures by 3D superposition provides higher sensitivity for identifying homologous proteins [11]. Protein structure search methods can be divided into: (1) structural alignment based, such as CE [12], Dali [13], and TM-align [14, 15]; (2) structural alphabet based, such as 3D-BLAST-SW [16], CLE-SW [17], and Foldseek [18]. Protein structure prediction methods like AlphaFold2 and AlphaFold Protein Structure Database (AFDB) have greatly reduced the cost of obtaining protein structures [19–21], which expands the usage scenarios of the structure search methods. However, in the vast majority of cases, the sequence search method is still more convenient. Especially in scenarios involving a large number of new sequences, such as metagenomic sequences [22], sequences generated by protein engineering [23] and antibody variant sequences [24].

At the same time, protein language models (PLMs) such as ESM [25] only take protein sequences as input and perform well in various downstream tasks [26], especially in structure-related tasks like secondary structure prediction and contact prediction [27]. These findings suggest that protein language models learn about protein structure, which can help downstream networks capture remote homology with only sequences as input. The latest research DEDAL learns a continuous representation of protein sequences that, combined with the SW algorithm, leads to a more accurate pairwise sequence alignment and homology detection method [28].

To improve the sensitivity of homologous protein search while maintaining the universality of sequence search, we propose PLMSearch (Fig. 1). PLMSearch mainly consists of the following three steps: (1) Filter out protein pairs that share the same Pfam clan domain [29]. (2) Predict the similarity between pre-filtered protein pairs with the protein language model and rank them. PLMSearch will not lose much sensitivity without structures as input, because it uses the protein language model to capture remote homology information from deep sequence embeddings. In addition, the SS-predictor used in this

step uses the structural similarity (TM-score) as the ground truth for training. This allows PLMSearch to acquire reliable similarity even without structures. (3) Sort the protein pairs based on their similarity, and output the search results for each query protein accordingly. Experiments show that PLMSearch improves the sensitivity of MMseqs2 by more than threefold, which is one of the best sequence search methods, and is close to the state-of-the-art structure search method Foldseek. This improvement is particularly apparent in remote homology pairs. In addition, PLMSearch is more than four orders of magnitude faster than structural alignment methods like TM-align.

2 Results

2.1 PLMsearch reaches similar sensitivities as structure search methods.

We benchmarked the sensitivity of PLMSearch and the state-of-the-art homologous protein search methods (see "Baselines") in the SCOPe40-test and the Swiss-Prot datasets (see "Datasets").

In the SCOPe40-test dataset, we performed an all-versus-all search test (see "Evaluation based on the SCOPe benchmark"). As shown in Fig. 2 and Extended Data Table A1, PLMSearch achieves a slightly lower sensitivity than those of TM-align and Dali, higher than that of the state-of-the-art structure search method Foldseek, and much higher than those of the sequence search method MMseqs2 and other structure search methods 3D-BLAST-SW and CLE-SW. The improvement is particularly apparent in superfamily and fold, which are shallower and have less significant similarity between proteins (3, 14, and 92 times exceeding MMseqs2 in family, superfamily, and fold, respectively). Supplementary Table 1 indicates that the primary reason for the improvement is that PLMSearch is more robust and makes the rank of the first FP lower, increasing the number of total TPs up to the first FP (21 times exceeding MMseqs2). When using the TM-score benchmark (see "Evaluation based on the TM-score benchmark") to evaluate methods other than TM-align (Supplementary Fig. 1, Supplementary Table 2), PLMSearch performs close to Foldseek, especially at P@10, which is more intuitive to users (P@10 means how many correct query results the user can get on average among the top 10 results).

In the Swiss-Prot dataset, we designed another two search tests with the same TM-score benchmark, both with Swiss-Prot as the target dataset. The two sets of search tests are Swiss-Prot to Swiss-Prot (Extended Data Table A2) and SCOPe40 to Swiss-Prot (Extended Data Table A3). PLMSearch achieves the best performance in both sets of search tests against Swiss-Prot.

2.2 PLMsearch always maintains high speed, even with limited computing resources

We compared the running times on the three search tests (Extended Data Table A4). By using PfamClan to pre-filter the protein pairs and SS-predictor to predict the similarity, instead of calculating the structural similarity (TM-score) of all protein pairs from scratch, PLMSearch is more than four orders of magnitude faster than TM-align in large-scale searches.

It is worth noting that PLMSearch can achieve similar efficiency even with CPU ONLY (Supplementary Table 3). This is due to the fact that PLMSearch calculates and preloads the deep embedding of all proteins in the target dataset in advance. This strategy helps to save much time by avoiding repeated forward propagations of a pre-trained model with a large number of parameters. The efficiency of PLMSearch also benefits from the fact that it only needs to calculate the protein pairs pre-filtered by PfamClan instead of calculating all pairs from scratch like SS-predictor, which makes PLMSearch more than 8 times faster than SS-predictor in large-scale scenarios (Supplementary Table 3).

Due to the small parameters of the downstream model, the time required for searching is actually very short. PLMSearch spends most of its time preprocessing the query sequence (Supplementary Table 4). It is worth noting that the time required for preprocessing grows linearly (time complexity $O(n)$). It takes only 0.03 s to generate the embedding and 0.26 s to generate the Pfam result of each protein, if tested in Swiss-Prot. However, the number of protein pairs to be searched grows quadratically (time complexity degree $O(n * m)$). Therefore, in large-scale protein searches, reducing search time is more critical. Especially in the common scenario where of a limited number of proteins are locally searched against several large datasets, which means that few query proteins need to be preprocessed, but many pairs need to be searched. PLMSearch excels in this common scenario.

In addition, when we evaluate structure search methods, we assume that the protein structure has been prepared in advance. In fact, although the predicted structures of most of the proteins in the UniProt dataset can be downloaded directly from https://alphafold.ebi.ac.uk/download [19, 21], there are still exceptions. In scenarios containing a large number of new sequences, the time to prepare structural data needs to be considered when using structure search methods. Using sequence search methods such as PLMSearch does not require this part of the time.

2.3 PLMSearch accurately detects remote homology pairs

We conducted a specific analysis of recalled pairs and missed pairs of different search methods in the Swiss-Prot to Swiss-Prot search test (Fig. 3). We calculated the TM-score and the sequence identity of protein pairs (see "Sequence identity calculation" Supplementary Section). In this paper, pairs with similar sequences and similar structures are defined as sequence identity > 0.3 [30] and TM-score > 0.5 [31] and are called "easy pairs"; pairs with dissimilar sequences but similar structures are defined as sequence identity < 0.3 but TM-score > 0.5 and are called "remote homology pairs" (Fig. 4).

Compared with easy pairs, remote homology pairs in the "twilight zone" of protein sequence homology are more difficult to find [30]. Among the four methods, even the least sensitive method MMseqs2 recalls all the easy pairs (579/579), but performs poorly on remote homology pairs (276/2138). In contrast, powered by the protein language model, SS-predictor and PLMSearch find most of the remote homology pairs (SS-predictor: 2024/2138, PLMSearch: 2097/2138, more than 7 times exceeding MMseqs2), and the recall rate exceeds Foldseek, which directly uses structural data as input (Foldseek: 1721/2138, see Supplementary Table 5).

2.4 PfamClan and SS-predictor make PLMSearch more robust

We first performed an ablation experiment based on the SCOPe benchmark on the SCOPe40-test search test (Supplementary Fig. 2, Supplementary Table 6). We compared the method based on TM-score (the first block) with the method based on the predicted similarity (the second block). Among them, TM-align (w/ PfamClan) has a similar pipeline to PLMSearch, but uses TM-score as similarity. Compared with TM-align and SS-predictor, methods with PfamClan (TM-align (w/ PfamClan) and PLMSearch) perform significantly better, especially on superfamily and fold. This shows that PfamClan always yields better results, whether using TM-score or the predicted similarity. Supplementary Table 7 indicates the primary reason is that PfamClan can avoid a large number of FPs, leading to the rank of the first FP appears significantly lower, so that the method can recall more TPs before the first FP.

We also performed an ablation experiment based on the TM-score benchmark on the Swiss-Prot to Swiss-Prot search test (Supplementary Fig. 3, Supplementary Table 8). The baselines contain PfamClan, which will filter out pairs containing the same Pfam clan domain. As a comparison, Euclidean, COS, SS-predictor (w/o COS), SS-predictor, and PLMSearch take the first 10,000 pairs with the highest similarity as the results.

In the ablation experiment: (1) With the help of PfamClan, PLMSearch performs better than SS-predictor (boosting Recall, Precision and F1-score by 2.15%, 1.88% and 2.15%, respectively). (2) Based on the same embeddings, SS-predictor performs better than other similarity-based search methods (boosting F1-score by 160% and 156% compared with Euclidean and COS, respectively).

We further compared the correlation between the predicted similarity and TM-score (Extended Data Fig. A1). The correlation between the similarity predicted by Euclidean (COS) and TM-score is not high, resulting in a large number of actually dissimilar protein pairs ranking first. The similarity predicted by SS-predictor is more correlated with TM-score (with a higher Pearson correlation coefficient and a higher Spearman correlation coefficient, see Supplementary Table 9). Moreover, SS-predictor increases the robustness and reliability in extreme cases by multiplying the predicted TM-score by the COS distance between protein embeddings as the final similarity (Supplementary Table 10).

We then analyzed "Missed pairs" and "Wrong pairs" (see Fig. 6 for definition) in the ablation experiment (Fig. 5). The "Missed pair" ("Wrong pairs") intersected by SS-predictor and PfamClan only account for a small part of all. They complement each other in PLMSearch and avoid a large number of missed pairs and wrong pairs that appeared in SS-predictor (green part in the figure), making PLMSearch more sensitive and accurate. In addition, PLMSearch further selects 10,000 pairs from the 19,238 protein pairs filtered by PfamClan according to similarity ranking. In this process, only one "Missed pair" (light blue part in Fig. 5a, 25.0% missed pairs of PfamClan) was added, while the wrong pair was reduced by 6805 (red part in Fig. 5b, 95.9% wrong pairs of PfamClan). This shows that PLMSearch's ranking according to similarity is very efficient.

2.5 PLMsearch, like TM-align, pays more attention to the global similarity

We analyzed the wrong pairs of different search methods in the Swiss-Prot to Swiss-Prot search test (Supplementary Table 11). Specifically, we counted the number of wrong pairs in the top 5,506 pairs (because among these four methods, MMseqs2 returned the fewest 5506 pairs as the search result). Among them, PLMSearch has the fewest "Wrong pairs" (1.4%, 36.8%, and 59.0% of MMseqs2, Foldseek, and SS-predictor).

We also performed a manual inspection on the protein pairs filtered by Foldseek but with a TM-score lower than 0.15 (Supplementary Table 12). As reported in Foldseek's paper, TM-align does not consider these pairs to share similar structures, as TM-align looks for global structural

superpositions. So the TM-scores of these pairs are low (TM-score(Default)<0.5, TM-score(Avg. length)< 0.15). Nevertheless, Foldseek filters them out because it pays more attention to local similarity. Like TM-align, the similarity calculated by PLMSearch is all less than the reference similarity 0.3, so PLMSearch also considers their structures not similar (see "Reference similarity of PLMSearch" Supplementary Section and Supplementary Table 13).

3 Discussion

In this work, we investigate the use of protein language models for homologous protein search. We propose PLMSearch, which takes only sequences as input and searches for homologous proteins using the protein language model and Pfam sequence analysis, allowing PLMSearch to extract remote homology information hidden behind sequences. Experiments show that PLMSearch achieves higher sensitivity than MMseqs2, another sequence search method, and is comparable to the state-of-the-art structure search method Foldseek. The improvement is especially noticeable in remote homology pairs.

It is worth noting that in the SCOPe40-test search test, not only are the sequences unseen during training, but also the fold to which the protein belongs. This means that PLMSearch can also search for domains belonging to new folds effectively, implying that PLMSearch learns universal biological properties that are not easily captured by other methods based on sequence alignment [10].

In the future, we will probe removing PfamScan [32] from the pre-filtering process, transforming the entire PLMSearch into a pure deep-learning model. Specifically, the current PLMSearch must find protein pairs that belong to the same Pfam clan based on PfamScan results. We hope to use the machine learning model to predict the fold the protein belonging to and form protein pairs between proteins belonging to the same fold, thereby saving the time required for PfamScan. Furthermore, rather than using a general-purpose large-parameter protein language model like ESM-1b, we will try to use the pre-training method to fine-tune a small-parameter protein model with fewer parameters but better for protein search, making PLMSearch lighter and faster.

In summary, we believe that PLMSearch has removed the low sensitivity limitations of sequence search methods. Since the sequence is more applicable and easier to obtain than structure, PLMSearch is expected to become a more convenient homologous protein search method.

4 Methods

4.1 PLMSearch pipeline

PLMSearch consists of three steps (Fig. 1). (1) PfamClan. We use PfamScan [32] to identify Pfam clan domains in query protein sequences and search a target dataset for proteins sharing the same Pfam clan domain. In addition, a limited number of query proteins lack any Pfam clan domain, or their Pfam clans differ from any target protein. To prevent such queries from yielding no results, all pairs between such query protein and target proteins will be retained. (2) Similarity prediction. We generate embeddings containing remote homology information using a protein language model. Subsequently, we use the SS-predictor to predict the similarity of each pair. (3) Search result. Finally, we sort the protein pairs based on their similarity, and output the search results for each query protein accordingly.

For the filtered protein pairs, if TM-align structure alignment is required, users can use the parallel architecture provided by us (Supplementary Fig. 4) to quickly calculate TM-scores.

4.2 PfamClan

As mentioned above, PfamClan filters out protein pairs that share the same Pfam clan domain(Fig. 1, step (1)). Compared to calculating the similarity of all protein pairs from scratch (as SS-predictor does), using PfamClan to pre-filter greatly reduces the calculation time (Supplementary Table 3) and avoids many "Wrong Pairs" (Fig. 5).

It is worth noting that the recall rate is more important in the initial pre-filter. PfamClan is based on a more relaxed standard of sharing the same Pfam clan domain, instead of sharing the same Pfam family domain. This feature helps PfamClan achieve a higher recall than PfamFamily (see "Pfam based pre-filter method" Supplementary Section).

4.3 Similarity prediction

Based on the protein language model and SS-predictor, PLMSearch performs further similarity prediction based on the pre-filter results of PfamClan (Fig. 1, step (2)). The motivation is that the clustering results based on PfamClan show a significant long-tailed distribution. One or two largest "Big clusters" contain the vast majority of proteins and protein pairs (>13% proteins and >59% protein pairs, see Extended Data Fig. A3). As the size of the dataset increases, the number of proteins contained in the "Big clusters" will greatly expand, further leading to a rapid increase in the number of pre-filter protein pairs (Supplementary Table 16). The required computing resources are excessive with TM-align used for all the filtered pairs. PLMSearch uses the predicted similarity instead of the TM-score calculated by

TM-align, which helps to greatly increase speed and avoids dependence on structures.

Protein language model [25, 33–35] learns only from sequence data and generates deep representations (embeddings) that contain multiple biological properties of proteins. Such deep representations have been widely used in various downstream tasks [36], especially the secondary structure prediction and contact prediction tasks that are related to structure [27]. Good performance on these tasks demonstrates that the deep representation (embedding) already contains the protein language model's understanding of protein structure.

As shown in Extended Data Fig. A4, the input protein sequences are first sent to the protein language model (ESM-1b here) to obtain the embedding of each amino acid, and finally the protein embedding is obtained through the average pooling layer. Subsequently, SS-predictor predicts the structural similarity (TM-score) between proteins through a bilinear projection network. SS-predictor multiplies the predicted TM-score by the COS distance between protein embeddings as the final similarity.

4.4 Datasets

4.4.1 SCOPe40

The SCOPe40 dataset consists of single domains with real structures. Clustering of SCOPe 2.01 [37, 38] at 40% sequence identity yielded 11,211 non-redundant protein domain structures ("SCOPe40"). As done in Foldseek, domains from SCOPe40 were split 8:2 by fold into SCOPe40-train and SCOPe40-test sets, and then domains with a single chain were reserved. We trained SS-predictor on SCOPe40-train and performed tests on SCOPe40-test as a benchmark. This means that the sequences and folds in the test set are both unseen during training time. In addition, in order to make a more objective comparison, the settings and metrics used in the all-versus-all SCOPe40-test search test are exactly the same as those used in Foldseek.

4.4.2 Swiss-Prot

Unlike SCOPe, the Swiss-Prot dataset consists of full-length, multi-domain proteins with predicted structures, which is closer to real-world scenarios. Because the throughput of experimentally observing the structure of proteins is very low and requires a lot of human and financial resources. The number of real structures in datasets like PDB [39–41] tends to be low. AlphaFold protein structure database (AFDB) obtains protein structure through deep learning prediction, so it can contain the entire protein universe and gradually become the mainstream protein structure dataset. Therefore, in this set of tests, we used Swiss-Prot with predicted structures from AFDB as the target dataset.

Specifically, we downloaded the protein sequence from UniProt (https://www.uniprot.org/downloads) [42] and the predicted structure from the AlphaFold Protein Structure Database (https://alphafold.ebi.ac.uk/download) [21]. A total of 542,317 proteins with both sequences and predicted structures were obtained. For these proteins (Supplementary Fig. 5a), we dropped low-quality proteins with an avg. pLDDT lower than 70, and used the remaining 498,654 proteins as target proteins (Supplementary Fig. 5b). Based on these target proteins, the following two sets of search tests are set up:

- Swiss-Prot to Swiss-Prot: With the above target proteins, 5 proteins with an avg. pLDDT higher than 95 were randomly selected as query proteins (Supplementary Fig. 5c, Supplementary Table 17).
- SCOPe40 to Swiss-Prot: To make the test more comprehensive, we randomly selected 50 proteins from SCOPe40-test (with experimental structures) as query proteins, and used the 498,654 predicted protein structures from Swiss-Prot as target proteins.

4.5 Evaluation benchmarks

4.5.1 Evaluation based on the SCOPe benchmark

We first compared various homologous protein search methods using the same metric as Foldseek [18]. We performed an all-versus-all search on the SCOPe40-test search test. All-versus-all search means both the query dataset and the target dataset were SCOPe40-test. We compared the performance of different methods in finding the same SCOPe family, superfamily, and fold (Supplementary Fig. 6). Specifically, for every query, we measured "the fraction of recalled TPs until the first FP" as sensitivity. For the three levels of family, superfamily, and fold, TPs are defined as the same family, the same superfamily but different families, and the same fold but different superfamilies, respectively, all excluding self-hits. Furthermore, protein pairs from different folds are FPs. We used a cumulative ROC curve and quantitatively measured the sensitivity by comparing the AUC for family-, superfamily-, and fold-level classifications.

It is worth mentioning that this evaluation metric is only applicable to the SCOPe dataset consisting of single domains, and it is problematic to evaluate the SCOPe/CATH domain annotations of multi-domain proteins [18]. Because without the Astral SCOPe single-domain dataset, domain annotation requires a reference annotation

method rather than human manual classification as the gold standard. The evaluation would then be uncontrollably biased towards methods that optimize for similar metrics. However, "good methods" under such metrics would make similar mistakes as reference methods. Also, false negatives of gold-standard annotation methods can make methods with different optimization metrics (maybe more sensitive) produce a large number of high-scoring FPs.

4.5.2 Evaluation based on the TM-score benchmark

To objectively evaluate different methods on various datasets, we computed the widely used TM-score as the structural similarity between protein pairs for the gold standard. TM-align first finds the best equivalent residues of two proteins, and then outputs a TM-score, which is used to measure the structural similarity between two proteins. Since the TM-score normalized by the length of the query protein is not symmetrical, we use the average length of the protein for normalizing to obtain a symmetric TM-score (TM-align(avg. length)). Compared to calculating TM-score for both comparison directions and averaging them together for each protein pair (TM-align(avg. score)) [43], TM-align(avg. length) has achieved better results in the evaluation based on the SCOPe benchmark (Supplementary Table 18), which is why we use the TM-align(avg. length) setting.

The metrics used for evaluation are described in detail in "Evaluation metric details based on TM-score" Supplementary Section. The various cases in the evaluation are described below (also see Fig. 6 for an example to explain the definitions a-f).

(a) High TM-score protein pairs: Protein pairs with a TM-score higher than 0.5, usually assumed to have the same fold [31, 44].

(b) Middle TM-score protein pairs: Protein pairs with a TM-score between 0.3 and 0.5.

(c) Low TM-score protein pairs: Protein pairs with a TM-score lower than 0.3, usually assumed as randomly selected irrelevant proteins.

(d) Filtered protein pairs: Protein pairs filtered by the search method.

(e) Missed pair: Protein pairs that are missed by search methods, but with a TM-score higher than 0.5.

(f) Wrong pair: Protein pairs that are filtered by search methods, but with a TM-score lower than 0.3.

4.6 Baselines

4.6.1 Previously proposed methods

(1) Sequence search: MMseqs2 [6]. (2) Structure search — structural alphabet: 3D-BLAST-SW

[16], CLE-SW [17], and Foldseek [18]. (3) Structure search — structural alignment, reviewed in [45]: CE [12], Dali [13], and TM-align [14, 15]. For the specific introduction and settings of these proposed methods, see "Baseline details" Supplementary Section.

4.6.2 Similarity-based search methods

Predict and sort the similarity between all protein pairs from scratch (Extended Data Fig. A2). Different search methods are distinguished according to the way they predicts similarity.

- Euclidean: Use the reciprocal of the Euclidean distance between embeddings as the similarity.

$$similarity(p, q) = \frac{1}{\sqrt{\sum_{i=1}^{n} (p_i - q_i)^2}} \quad (1)$$

- COS: Use the COS distance between embeddings as the similarity, ϵ is a small value to avoid division by zero.

$$similarity(p, q) = \frac{p \cdot q}{\max\left(\|p\|_2 \cdot \|q\|_2, \epsilon\right)} \quad (2)$$

- SS-predictor (w/o COS): Use the TM-score predicted by SS-Predictor as the similarity, but without multiplying the predicted TM-score with the COS distance between protein embeddings.
- SS-predictor: Use (the predicted TM-score * COS distance between embeddings) as the similarity of protein pairs. The difference from PLMSearch is that SS-predictor predicts all protein pairs from scratch instead of pairs from PfamClan.

4.7 Experiment settings

4.7.1 Pfam result generation

We obtained the Pfam family domains of proteins by PfamScan (version 1.6) [32]. For PfamClan, we query the comparison table Pfam-A.clans.tsv (https://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.clans.tsv.gz), and replace the family domain with the clan domain it belongs to. For the family domain that has no corresponding clan domain, we treat it as a clan domain itself.

4.7.2 Protein language model

ESMs are a set of protein language models that have been widely used in recent years. We use ESM-1b [25], a SOTA general-purpose protein language model, to generate protein embeddings.

4.7.3 SS-predictor training

We use the deep learning framework PyTorch (version 1.7.1), ADAM optimizer, with MSE as the loss function to train SS-predictor. The batch size is 100, and the learning rate is 1e-5 on 20 epochs.

The training label is the TM-score calculated by TM-align(avg. length). The dataset used for training is all protein pairs in SCOPe40-train. To speed up the training and reduce the computing resources required for training, we fix the parameters in ESM-1b and only train the parameters in the bilinear projection network.

4.7.4 Experimental environment

We conducted the environment on a server with a 72-core Intel(R) Xeon(R) Gold 6240 CPU @ 2.60 GHz and 503 GB RAM memory. The GPU environment of the server is 1×Tesla V100 PCIe 32 GB. The environment of the Webserver is CPU ONLY, with 8 * Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz.

References

[1] Yao, S. *et al.* NetGO 2.0: improving large-scale protein function prediction with massive sequence, text, domain, family and network information. *Nucleic Acids Research* **49** (W1), W469–W475 (2021) .

[2] You, R. *et al.* Netgo: improving large-scale protein function prediction with massive network information. *Nucleic acids research* **47** (W1), W379–W387 (2019) .

[3] You, R., Yao, S., Mamitsuka, H. & Zhu, S. DeepGraphGO: graph neural network for large-scale, multispecies protein function prediction. *Bioinformatics* **37**, i262–i271 (2021) .

[4] Hu, L., Wang, X., Huang, Y.-A., Hu, P. & You, Z.-H. A survey on computational models for predicting protein–protein interactions. *Briefings in Bioinformatics* **22** (5) (2021). Bbab036 .

[5] Liu, L., Huang, X., Mamitsuka, H. & Zhu, S. Hpolabeler: improving prediction of human protein–phenotype associations by learning to rank. *Bioinformatics* **36** (14), 4180–4188 (2020) .

[6] Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *Journal of Molecular Biology* **215** (3), 403–410 (1990) .

[7] Steinegger, M. & Söding, J. Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature biotechnology* **35** (11), 1026–1028 (2017) .

[8] Steinegger, M. *et al.* Hh-suite3 for fast remote homology detection and deep protein annotation. *BMC bioinformatics* **20** (1), 1–15 (2019) .

[9] Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale

using diamond. *Nature methods* **18** (4), 366–368 (2021) .

[10] Mahlich, Y., Steinegger, M., Rost, B. & Bromberg, Y. HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics* **34** (13), i304–i312 (2018) .

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

[12] Shindyalov, I. N. & Bourne, P. E. Protein structure alignment by incremental combinatorial extension (ce) of the optimal path. *Protein engineering* **11** (9), 739–747 (1998) .

[13] Holm, L. *Using Dali for Protein Structure Comparison*, 29–42 (Springer US, New York, NY, 2020).

[14] Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins: Structure, Function, and Bioinformatics* **57** (4), 702–710 (2004) .

[15] Zhang, Y. & Skolnick, J. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research* **33** (7), 2302–2309 (2005) .

[16] Yang, J.-M. & Tung, C.-H. Protein structure database search and evolutionary classification. *Nucleic acids research* **34** (13), 3646–3659 (2006) .

[17] Wang, S. & Zheng, W.-M. Clepaps: fast pair alignment of protein structures based on conformational letters. *Journal of bioinformatics and computational biology* **6** (02), 347–366 (2008) .

[18] van Kempen, M. *et al.* Foldseek: fast and accurate protein structure search. *bioRxiv* (2022) .

[19] Jumper, J. *et al.* Highly accurate protein structure prediction with alphafold. *Nature* **596** (7873), 583–589 (2021) .

[20] Baek, M. *et al.* Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373** (6557), 871–876 (2021) .

[21] Varadi, M. *et al.* Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research* **50** (D1), D439–D444 (2022) .

[22] Shan, S. *et al.* Deep learning guided optimization of human antibody against sars-cov-2

variants with broad neutralization. *Proceedings of the National Academy of Sciences* **119** (11), e2122954119 (2022) .

[23] Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N. & Madani, A. ProGen2: Exploring the Boundaries of Protein Language Models. *arXiv e-prints* arXiv:2206.13517 (2022) [cs.LG].

[24] Mitchell, A. L. *et al.* MGnify: the microbiome analysis resource in 2020. *Nucleic Acids Research* **48** (D1), D570–D578 (2019) .

[25] Rives, A. *et al.* Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences* **118** (15), e2016239118 (2021) .

[26] Hu, M. *et al.* Exploring evolution-based &-free protein language models as protein function predictors. *arXiv preprint arXiv:2206.06583* (2022) .

[27] Rao, R. *et al.* Evaluating protein transfer learning with tape. *Advances in neural information processing systems* **32** (2019) .

[28] Llinares-Lopez, F., Berthet, Q., Blondel, M., Teboul, O. & Vert, J.-P. Deep embedding and alignment of protein sequences. *Nature Methods* 1–8 (2022) .

[29] Mistry, J. *et al.* Pfam: The protein families database in 2021. *Nucleic acids research* **49** (D1), D412–D419 (2021) .

[30] Rost, B. Twilight zone of protein sequence alignments. *Protein engineering* **12** (2), 85–94 (1999) .

[31] Xu, J. & Zhang, Y. How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26** (7), 889–895 (2010) .

[32] Mistry, J., Bateman, A. & Finn, R. D. Predicting active site residue annotations in the pfam database. *BMC bioinformatics* **8**, 298 (2007) .

[33] Nijkamp, E., Ruffolo, J., Weinstein, E. N., Naik, N. & Madani, A. Progen2: exploring the boundaries of protein language models. *arXiv preprint arXiv:2206.13517* (2022) .

[34] Meier, J. *et al.* Language models enable zero-shot prediction of the effects of mutations on protein function. *Advances in Neural Information Processing Systems* **34**, 29287–29303 (2021) .

[35] Lin, Z. *et al.* Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv* (2022) .

[36] Hu, M. *et al.* Exploring evolution-based &-free protein language models as protein function predictors. *arXiv preprint arXiv:2206.06583* (2022) .

[37] Fox, N. K., Brenner, S. E. & Chandonia, J.-M. SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures. *Nucleic Acids Research* **42** (D1), D304–D309 (2013) .

[38] Chandonia, J.-M. *et al.* SCOPe: improvements to the structural classification of proteins – extended database to facilitate variant interpretation and machine learning. *Nucleic Acids Research* **50** (D1), D553–D559 (2021) .

[39] Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Research* **28** (1), 235–242 (2000) .

[40] Burley, S. K. *et al.* RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Research* **49** (D1), D437–D451 (2020) .

[41] Sehnal, D. *et al.* Mol* Viewer: modern web app for 3D visualization and analysis of large biomolecular structures. *Nucleic Acids Research* **49** (W1), W431–W437 (2021) .

[42] Consortium, T. U. UniProt: the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research* **51** (D1), D523–D531 (2022) .

[43] Bepler, T. & Berger, B. Learning the protein language: Evolution, structure, and function. *Cell systems* **12** (6), 654–669 (2021) .

[44] Zhang, Y., Hubner, I. A., Arakaki, A. K., Shakhnovich, E. & Skolnick, J. On the origin and highly likely completeness of single-domain protein structures. *Proceedings of the National Academy of Sciences* **103** (8), 2605–2610 (2006) .

[45] Hasegawa, H. & Holm, L. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology* **19** (3), 341–348 (2009) .

5 Author contributions

S.Z. conceived the project. S.Z. and J.Y. supervised the project. W.L., S.Z. and J.Y. designed the research and performed the analyses. W.L. wrote the software. H.W. sorted out the structure data. W.L. wrote the first draft of the manuscript. All authors contributed to the revision of the manuscript prior to submission and all authors read and approved the final version.

6 Competing interests

All authors declare no competing interests.

7 Figures and Tables

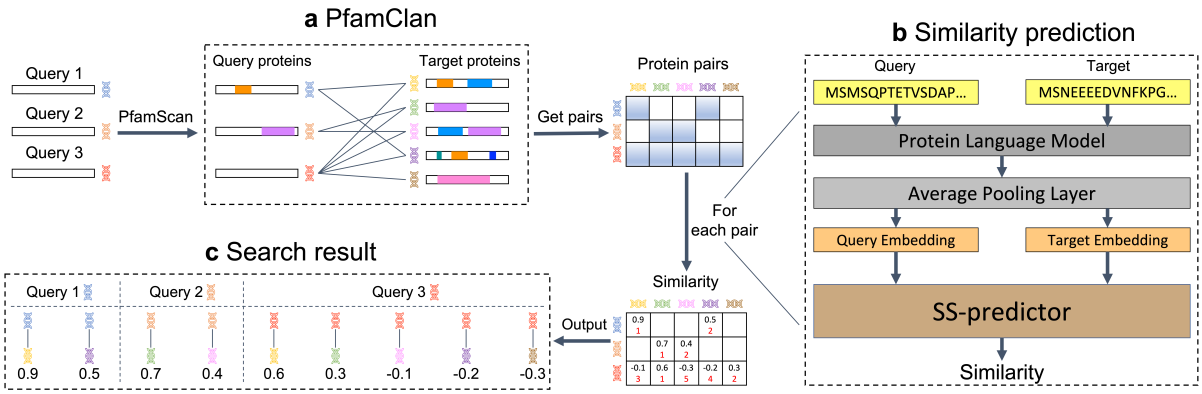


Fig. 1 Overview of the PLMSearch pipeline. a, PfamClan. Initially, PfamScan [32] identifies the Pfam clan domains of the query protein sequences (one), which are depicted in different color blocks. Subsequently, PfamClan searches the target dataset for proteins sharing the same Pfam clan domain with the query proteins. Notably, the last query protein lacks any Pfam clan domain, and therefore, its all pairs with target proteins are retained. **b,** Similarity prediction. The protein language model generates deep sequence embeddings for each protein pair. Subsequently, SS-predictor predicts the similarity of each protein pair. **c,** Search result. Finally, PLMSearch sorts the protein pairs based on their similarity and outputs the search results for each query protein separately.

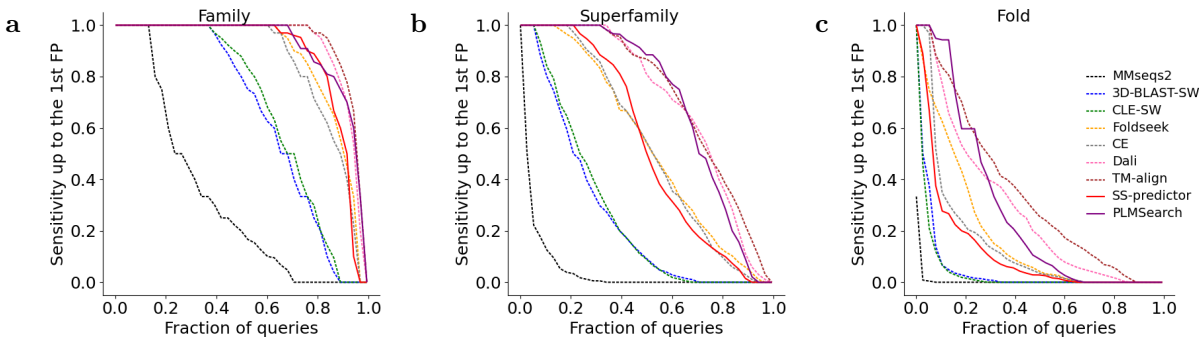


Fig. 2 PLMsearch reaches similar sensitivities as structure search methods. We performed an all-versus-all search test and compared the methods' performance for finding protein pairs of the same SCOPe family, superfamily, and fold by measuring for each query the fraction of true positive matches (TPs) out of all possible correct matches until the first false positive match (FP). For the three levels of family, superfamily, and fold, TPs are defined as the same family, the same superfamily but different families, and the same fold but different superfamilies, respectively. FPs are pairs from different folds. The sensitivity of different methods is measured by the area under the curve (AUC) of the cumulative ROC curve up to the first FP. Extended Data Table A1 statistics AUROC.

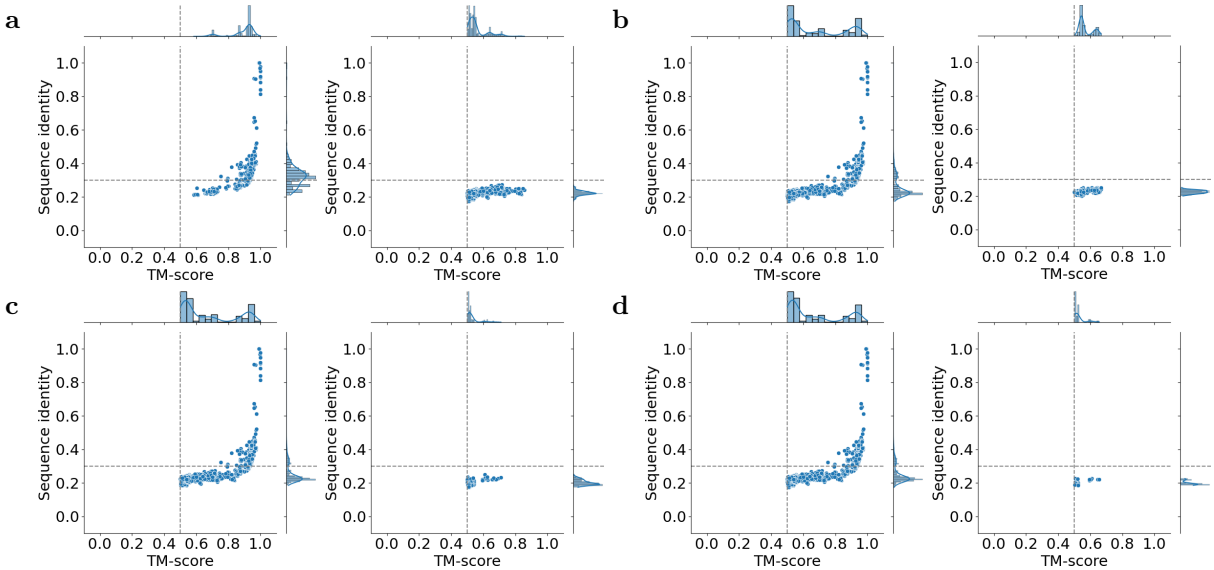


Fig. 3 PLMSearch accurately detects remote homologs. a, MMseqs2. **b,** Foldseek. **c,** SS-predictor. **d,** PLMSearch. For recalled pairs (left column) and missed pairs (right column) of different methods in the Swiss-Prot to Swiss-Prot search test, the TM-score (x-axis) and sequence identity (y-axis) between protein pairs are shown on the 2D scatter plot. The thresholds, sequence identity > 0.3 [30] and TM-score > 0.5 [31], are shown by dashed lines.

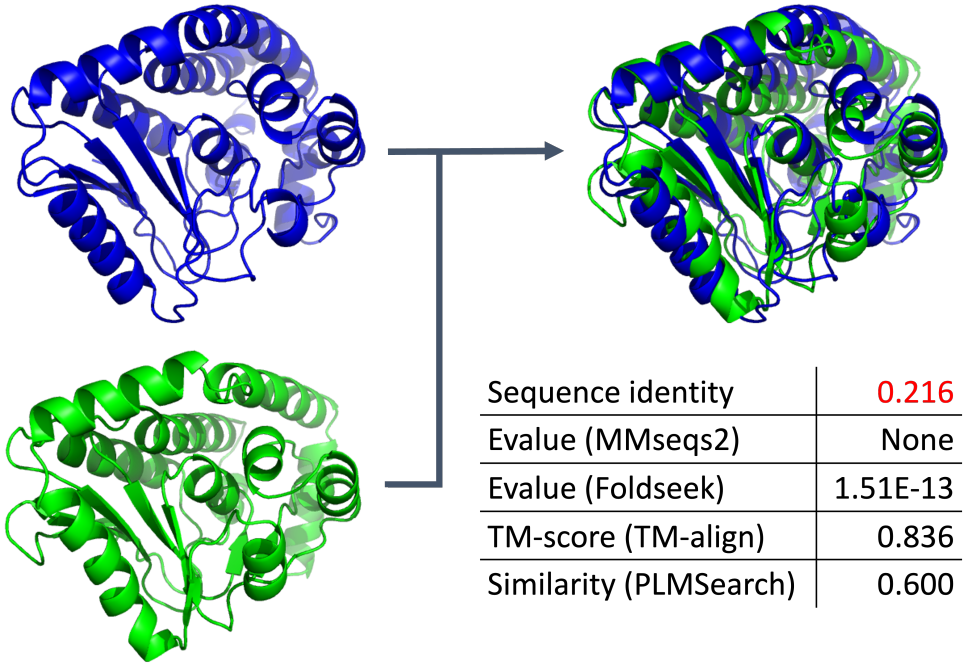


Fig. 4 Case study of remote homology pairs. Protein pairs with dissimilar sequences but similar structures are defined as sequence identity < 0.3 but TM-score > 0.5 and are called remote homology pairs here. The sequence identity between Q08558 (the first sequence, blue structure) and I6Y3U6 (the second sequence, green structure) is low. Thus, it is difficult to find this remote homology pair only through the sequence alignment (For the convenience of presentation, only the sequence alignment results of the first 41 amino acids are shown). Like structure search methods (Foldseek and TM-align), PLMSearch, powered by the protein language model, captures the deep protein homology pair that is missed by MMseqs2.

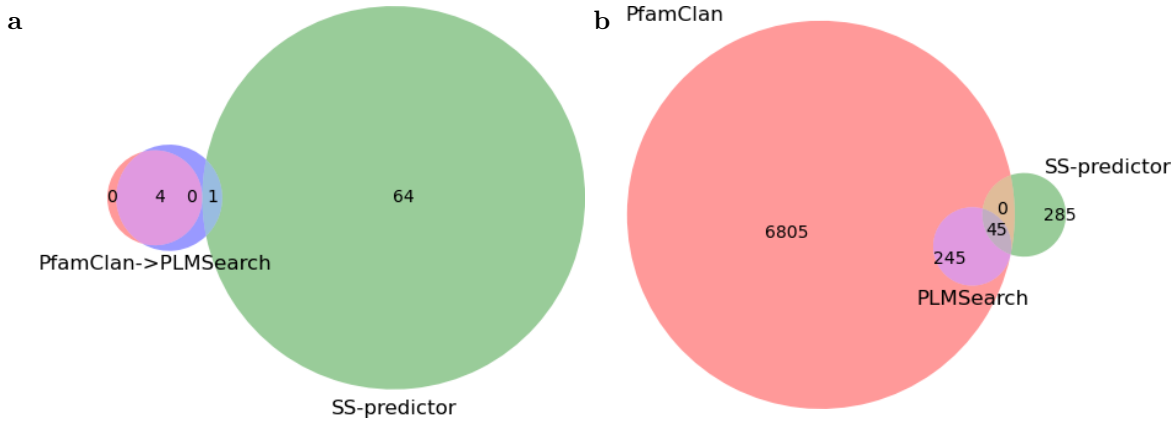


Fig. 5 PfamClan and other components make PLMSearch more robust. "Missed pairs" and "Wrong pairs" analysis of PfamClan, SS-predictor, and PLMSearch in ablation experiment. **a,** The venn diagram of "Missed pair". **b,** The venn diagram of "Wrong pair".

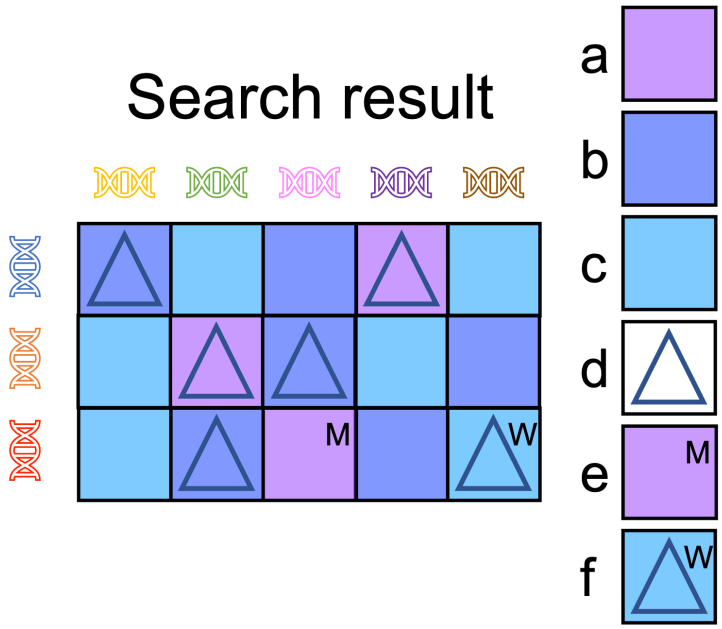
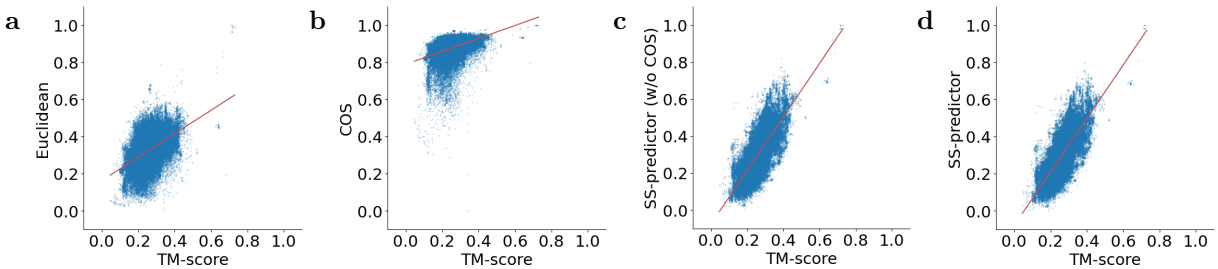
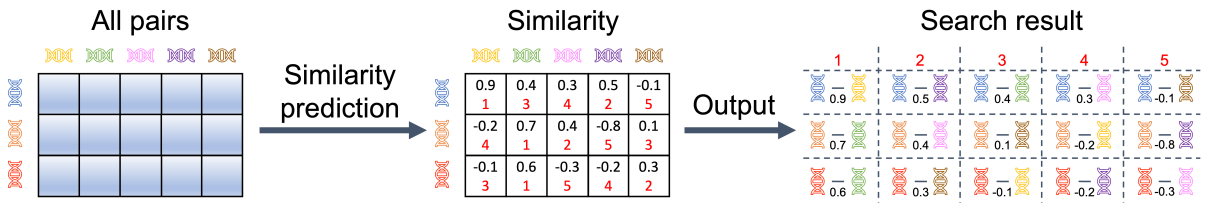


Fig. 6 Definition diagram. An example composed of three query proteins and five target proteins to explain various cases in the evaluation based on the TM-score benchmark. The **a-f** cases are represented by the corresponding legend **a-f**. Among the 15 protein pairs, three "High TM-score protein pairs" are marked with **a**; six "Medium TM-score protein pairs" are marked with **b**; six "Low TM-score protein pairs" are marked with **c**; six "Filtered protein pairs" are marked with **d**. The protein pair at (3,3) has a TM-score higher than 0.5 but is not filtered out, which is a "Missed pair" marked as **e**. The protein pair at (3,5) has a TM-score lower than 0.3 but is filtered out, which is a "Wrong pair" marked as **f**.

Appendix A Extended Data



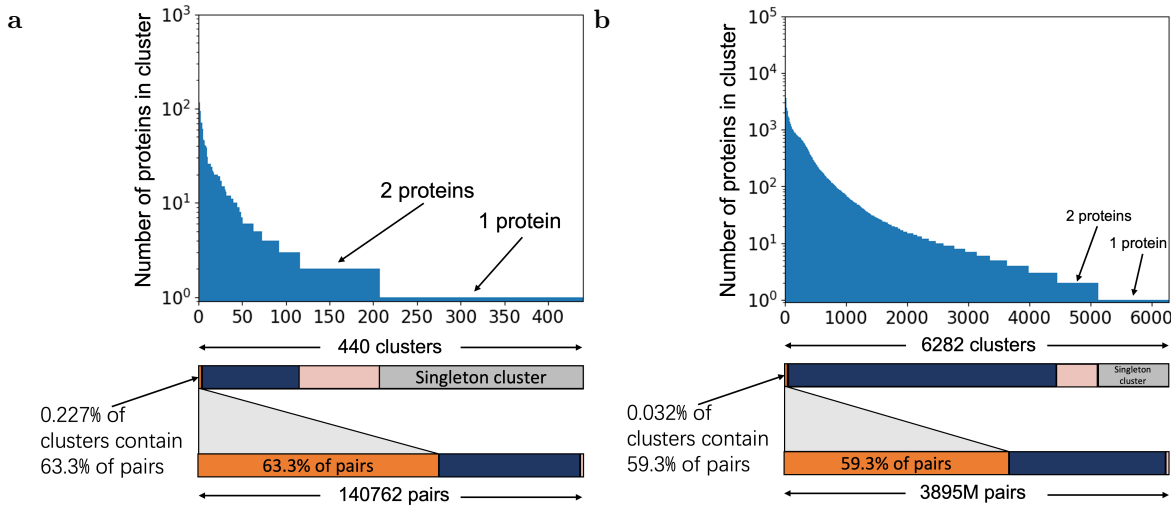
Extended Data Fig. A1 Two-dimensional scatter plot of the predicted similarity and TM-score. a, Euclidean. **b,** COS. **c,** SS-predictor (w/o COS). **d,** SS-predictor. We randomly selected 100,000 protein pairs in the Swiss-Prot to Swiss-Prot search test and used Euclidean, COS, SS-predictor (w/o COS), and SS-predictor as the predicted similarity. We normalized the predicted similarity to 0-1 as the Y-axis, and used TM-align to calculate their TM-score (between 0-1) as the x-axis, thereby plotting the 100,000 protein pairs as points on a 2D plane.



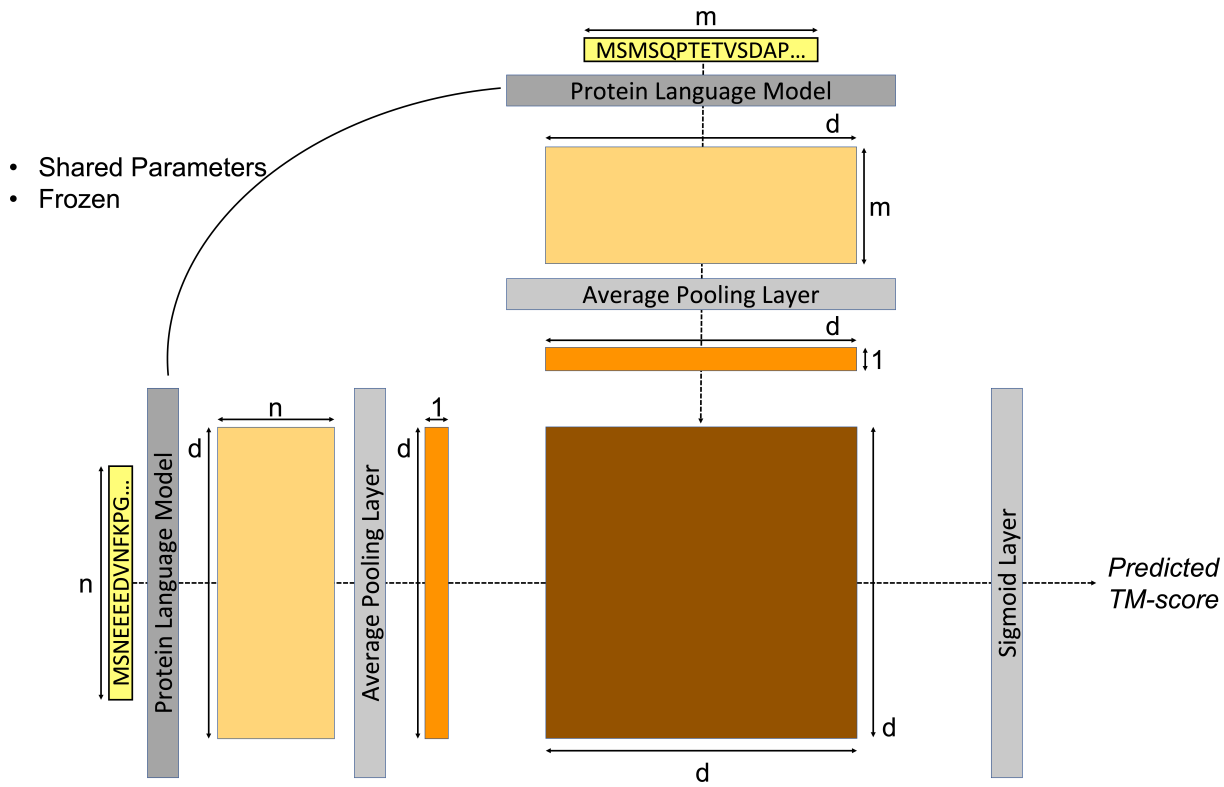
Extended Data Fig. A2 Similarity-based search methods. The similarity of all protein pairs is predicted and sorted, and then outputted as a search result.

Methods	Family	Superfamily	Fold
Sequence search			
MMseqs2	0.318	0.051	0.003
Structure search — structural alphabet			
3D-BLAST-SW	0.653	0.255	0.045
CLE-SW	0.673	0.266	0.034
Foldseek	0.869	0.532	0.178
Structure search — structural alignment			
CE	0.847	0.527	0.148
Dali	0.923	0.702	0.282
TM-align	**0.936**	**0.721**	**0.346**
Our methods			
Euclidean	0.700	0.310	0.039
COS	0.705	0.316	0.040
SS-predictor (w/o COS)	0.864	0.516	0.114
SS-predictor	0.876	0.530	0.118
PLMSearch	0.911	0.702	0.276

Extended Data Table A1 The area under the curve (AUC) of the cumulative ROC curve up to the first FP on the SCOPe40-test search test. It is the same with the average sensitivity up to the first FP.



Extended Data Fig. A3 Clustering results based on PfamClan on datasets of different sizes. a, SCOPe40-test. **b,** Swiss-Prot. The clustering results based on PfamClan show a significant long-tailed distribution. More than 50% of the pre-filtered protein pairs (orange rectangles in the figure) are from the largest 1 or 2 clusters (Big clusters), which actually only accounts for a very small part of the entire clusters (for SCOPe40-test, 0.227%; for Swiss-Prot, 0.032%). See Supplementary Table 16 for specific statistical data.



Extended Data Fig. A4 SS-predictor. Two proteins in a protein pair are composed of m and n amino acids respectively. First, the protein language model converts the sequences into $m * d$ and $n * d$ amino acid embeddings, where d is the dimension of the embedding. Then a average pooling layer converts them into the corresponding $1 * d$ protein embedding z. Finally, a bilinear projection network predicts the TM-score of a protein embedding pair $(z1, z2)$ by $z1 * W * z2 + b$, where the matrices W and the scalar b are the learned parameters. SS-predictor multiplies the predicted TM-score by the COS distance between protein embeddings as the final similarity.

Methods	AUPR	MAP	P@100
MMseqs2	0.314	0.430	0.562
Foldseek	0.631	0.708	0.810
Euclidean	0.329	0.478	0.698
COS	0.333	0.483	0.708
SS-predictor (w/o COS)	0.796	0.675	0.732
SS-predictor	0.893	0.783	**0.868**
PLMSearch	**0.974**	**0.873**	**0.868**

Extended Data Table A2 Evaluation based on the TM-score benchmark on the Swiss-Prot to Swiss-Prot search test.

Methods	AUPR	MAP	P@100
MMseqs2	0.085	0.190	0.355
Foldseek	0.301	0.422	0.493
Euclidean	0.061	0.305	0.426
COS	0.077	0.309	0.428
SS-predictor (w/o COS)	0.242	0.380	0.444
SS-predictor	0.302	0.463	0.515
PLMSearch	**0.434**	**0.501**	**0.522**

Extended Data Table A3 Evaluation based on the TM-score benchmark on the SCOPe40 to Swiss-Prot search test.

Methods	Run time (s)		
	SCOPe40	Swiss-Prot to Swiss-Prot	SCOPe40 to Swiss-Prot
Methods based on structure			
Foldseek	5	16	17
TM-align	8,831	32,980	242,651
Methods based on sequence			
MMseqs2	2	6	6
SS-predictor	31	34	176
PLMSearch	18	25	48

Extended Data Table A4 Time comparison. The scale of SCOPe40, Swiss-Prot to Swiss-Prot, and SCOPe40 to Swiss-Prot are 2,207 * 2,207, 5 * 498,654, and 50 * 498,654, respectively.