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, 266237, 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 experimental results show that PLMSearch can search millions of query-target protein pairs in seconds like MMseqs2 while increasing the sensitivity 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://dmiip.sjtu.edu.cn/PLMSearch/.

1 Introduction

Homologous protein search is a key component of bioinformatics methods used in protein function prediction [1–5], protein-protein interaction prediction [6], and protein-phenotype association prediction [7]. The goal of homologous protein search is to find the homologous proteins from the target dataset (generally a large-scale standard dataset like Swiss-Prot [8]) for each query protein. The target protein with a higher probability 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 MMseqs2 [9], BLASTp [10], and Diamond [11]. Despite the success of homology inference based on sequence similarity, it remains challenging to detect distant evolutionary relationships from sequences only [12]. Sequence profiles and profile hidden Markov models (HMMs) are condensed representations of multiple sequence alignment (MSAs) that specify for each position the probability of observing each of the 20 amino acids in evolutionarily related proteins. When the sequence identity is lower than 0.3, methods based on profile HMMs such as HMMER [13], HHsearch [14] and HHblits [15, 16], are currently the best tools for homologous protein search.

In cases of highly distant evolutionary relationships, sequences may have diverged to such an extent that we can no longer detect their relatedness. Since structures diverge much more slowly than sequences, detecting similarity between protein structures by 3D superposition provides higher sensitivity [17]. Protein structure search methods can be divided into: (1) contact/distance map-based, such as Map_align [18], EigenTHREADER [19] and DiscoVER [20]; (2) structural alphabet-based, such as 3D-BLAST-SW [21], CLE-SW [22], Foldseek, and Foldseek-TM [23]; (3) structural alignment-based, such as CE [24], Dali [25], and TM-align [26, 27]. Protein structure prediction methods like AlphaFold2 and AlphaFold Protein Structure Database (AFDB) have greatly reduced the cost of obtaining protein structures [28–30], 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 [31], sequences generated by protein engineering [32], and antibody variant sequences [33].

At the same time, protein language models (PLMs) such as ESMs [34–36] and ProtTrans [37] only take protein sequences as input, trained on hundreds of millions of unlabeled protein sequences using self-supervised tasks such as masked amino acid prediction. PLMs perform well in various downstream tasks [38], especially in structure-related tasks like secondary structure prediction and contact prediction [39]. More recently, ProtENN [40] uses an ensemble deep learning framework that generated protein sequence embeddings to classify protein domains

into Pfam families [41]; CATHe [42] trains an ANN on embeddings from the PLM ProtT5 [37] to detect remote homologs for CATH [43] superfamilies; Embedding-based annotation transfer (EAT) [44] uses Euclidean distance between vector representations (ProtT5 embeddings) of proteins to transfer annotations from a set of labeled lookup protein embeddings to query protein embeddings; DEDAL [45] and latest pLM-BLAST [46] obtain a continuous representation of protein sequences that, combined with the Smith-Waterman algorithm, leads to a more accurate pairwise sequence alignment and homology detection method. These methods apply representations generated by deep learning models to protein domain classification, protein annotation, and pairwise sequence alignment, fully demonstrating the advantage of deep learning models in identifying remote homology. However, protein language models are not fully utilized for large-scale protein sequence search.

To improve the sensitivity while maintaining the universality and efficiency of sequence search, we propose PLMSearch (Fig. 1). PLMSearch mainly consists of the following three steps: (1) PfamClan filters out protein pairs that share the same Pfam clan domain [41]. (2) SS-predictor (**S**tructural **S**imilarity predictor) predicts the similarity between all query-target pairs with embeddings generated by the protein language model. 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 as input. (3) PLMSearch sorts the pairs pre-filtered by PfamClan based on their predicted similarity and outputs the search results for each query protein accordingly. Search tests on SCOPe40-test and Swiss-Prot reveal that PLMSearch is always one of the best methods and provides the best tradeoff between accuracy and speed. Specifically, PLMSearch can search millions of query-target protein pairs in seconds like MMseqs2, but increases the sensitivity by more than threefold, and approaches the state-of-the-art structure search methods. The improvement in sensitivity is particularly apparent in remote homology pairs.

2 Results

2.1 PLMsearch reaches similar sensitivities as structure search methods

We benchmarked the sensitivity of PLMSearch and SS-predictor, five other sequence search methods (MMseqs2, Blastp, HHblits, EAT, and pLM-BLAST), four structural alphabet-based search

methods (3D-BLAST-SW, CLE-SW, Foldseek, and Foldseek-TM), three structural alignment-based search methods (CE, Dali, and TM-align) and performed search tests on SCOPe40-test and Swiss-Prot after filtering of homologs from the training set (see "Datasets", "Metrics", and "Baselines" Section). Fig. 2 shows the results of the ten most competitive methods in sensitivity and speed. In the SCOPe40-test dataset (2,207 proteins), we performed an all-versus-all search test. Therefore, a total of 4,870,849 query-target pairs were tested for all the methods (Fig. 2a-c). In the Swiss-Prot search test, we randomly selected 50 queries from Swiss-Prot and 50 queries from SCOPe40-test (a total of 100 query proteins) and searched for 430,140 target proteins in Swiss-Prot. Therefore, a total of 43,014,000 query-target pairs were tested for the six most efficient methods for searching large-scale datasets (Fig. 2d). Supplementary Fig. 1 shows the results of the other two structural alphabet-based and two structural alignment-based search methods.

Extended Data Table A1 and Extended Data Table A2 records the specific values of each metric. PLMSearch performs well on most metrics, especially at the superfamily-level and fold-level, which are shallower and have less significant similarity between protein sequences. Taking AUROC of the fold-level as an example, PLMSearch's sensitivity (0.298) is below TM-align (0.346), higher than pLM-BLAST (0.176), Foldseek (0.216), Dali (0.281), and Foldseek-TM (0.296), and much higher than other homologous protein search methods. PLMSearch is 3, 14, and 149 times exceeding MMseqs2 in AUROC of the family-level (from 0.318 to 0.917), superfamily-level (from 0.050 to 0.733), and fold-level (from 0.002 to 0.298), respectively. Extended Data Table A3 indicates that the primary reason for the improvement is that PLMSearch is more robust and makes the rank of the first FP (**F**alse **P**ositive) lower, increasing the number of total TPs (**T**rue **P**ositives) up to the first FP (24 times exceeding MMseqs2, from 2.74 to 66.07).

2.2 PLMSearch searches millions of query-target pairs in seconds

We first compared the total search time of different methods for the all-versus-all search test on SCOPe40-test (2,207 proteins, 4,870,849 query-target pairs). To ensure fairness, we used the same computing resources (a 56-core Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz and 256 GB RAM server) when implementing different methods in the evaluation. For HHblits, pLM-BLAST, TM-align, and various other parallelizable methods, we utilized all 56 cores by default. Moreover, almost all methods need to preprocess the target dataset in advance. This part of the time is not required

while searching, so we did not include the preprocessing time in the search time statistics. As shown in the legend of Fig. 2a, by using SS-predictor to predict the similarity instead of calculating the structural similarity (TM-score) of all protein pairs, SS-predictor (10 s) and PLMSearch (4 s) is one of the fastest methods, and more than four orders of magnitude faster than TM-align (11,303 s).

PLMSearch can achieve similar efficiency on our publicly available web server with CPU only (64 * Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz and 512 GB RAM). Searching a query against Swiss-Prot (568K proteins) and UniRef50 (53.6M proteins) [8] takes approximately 0.03 min and 0.5 min respectively (Extended Data Table A4). This is because PLMSearch generates and preloads the embeddings of all target proteins in advance. This strategy helps to save much time by avoiding repeated forward propagations of the protein language model with a large number of parameters and saves the time for loading embeddings from disk to RAM. As a result, just one forward pass through the SS-predictor network is required to predict the similarity of millions of query-target pairs and complete the search in seconds (Fig. 5b).

PLMSearch, in reality, spends the majority of its time preprocessing query sequences (Extended Data Fig. A1, Supplementary Table 1), and this portion of time does not rise as the target dataset grows larger. In practice, users typically have a limited number of proteins to search across several massive datasets (few queries to preprocess but many pairs to search), and PLMSearch excels in this common scenario.

2.3 PLMSearch accurately detects remote homology pairs

Remote homology pairs generally refer to homologous protein pairs with dissimilar sequences but similar structures [47]. Such protein pairs have low sequence similarity, so their homology is difficult to be detected by sequence alignment-based methods (MMseqs2, Blastp), but can be detected by structure-based search methods (Foldseek, Foldseek-TM) (Fig. 3a). In this study, pairs with similar sequences and similar structures are defined as sequence identity > 0.3 [47] and TM-score > 0.5 [48, 49] 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". We conducted a specific analysis of recalled pairs and missed pairs (defined in Fig. 3b). We calculated the TM-score and the sequence identity (see "Sequence alignment" Supplementary Section) of the recalled/missed pairs and projected them onto a 2D scatter plot (Fig. 3c-h).

Compared with easy pairs (the first quadrant in Fig. 3c-h), remote homology pairs (the

fourth quadrant in Fig. 3c-h) in the "twilight zone" of protein sequence homology are more difficult to find [47]. Among the six methods (Extended Data Table A5), even the least sensitive methods MMseqs2 and Blastp recall all the easy pairs (574/574), but perform poorly on remote homology pairs (MMseqs2: 183/1105, Blastp: 203/1105). In contrast, powered by the protein language model, SS-predictor and PLMSearch search out most of the remote homology pairs (SS-predictor: 1051/1105, PLMSearch: 1098/1105, six times exceeding MMseqs2), and the recall rate exceeds Foldseek, which directly uses structural data as input (Foldseek: 934/1105, Foldseek-TM: 940/1105).

2.4 Ablation experiments: PfamClan and SS-predictor make PLMSearch more robust

To evaluate PLMSearch without the PfamClan component, we added an additional search test that only included query proteins that failed to scan any Pfam domain. This corresponds to the new and unclassified proteins in the real scene, and these sequences are often very important for some innovative research. Specifically, we screened a total of 123 proteins that failed to scan any Pfam domain from 2,207 queries in the SCOPe40-test. We used MAP and P@K to evaluate the search results of these 123 proteins (Supplementary Fig. 2, Supplementary Table 2). As expected, the performance of PLMSearch is exactly the same as that of SS-predictor, because PfamClan does not filter out any protein pairs, whereas PLMSearch still produces relatively sensitive search results (MAP = 0.577, P@1 = 0.813, P@10 = 0.726).

To evaluate PLMSearch without the SS-predictor component, we first clustered the SCOPe40-test and Swiss-Prot datasets based on PfamClan (Extended Data Fig. A2, Supplementary Table 3). Specifically, proteins belonging to the same Pfam clan are clustered. The clustering results show a significant long-tailed distribution. After pre-filtering with PfamClan, more than 50% of the pre-filtered protein pairs (orange rectangles in the figure) are from the largest 1-2 clusters (big clusters), which only accounts for a very small part of the entire clusters (SCOPe40-test: 0.227%; Swiss-Prot: 0.032%). Therefore, big clusters will result in a significant number of irrelevant protein pairs in the pre-filtering results, reducing accuracy, and must be further sorted and filtered based on similarity, which is what SS-predictor does.

Furthermore, among all similarity-based search methods (See "Baselines"), we further compared the correlation between the predicted similarity and TM-score (Extended Data Fig. A3a). 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-/Spearman correlation coefficient). This is why SS-predictor outperforms other similarity-based search methods with the same embeddings as input (Extended Data Fig. A3b-e, Extended Data Table A1, Extended Data Table A2).

3 Discussion

We study the use of protein language models for large-scale homologous protein search in this work. 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 buried behind sequences. Experiments reveal that PLMSearch outperforms MMseqs2 in terms of sensitivity and is comparable to the state-of-the-art structural search approach Foldseek. The improvement is especially noticeable in remote homology pairs. PLMSearch, on the other hand, is one of the fastest search methods in comparison to other baselines, capable of searching millions of query-target protein pairs in seconds. We also summarized the ten most competitive approaches based on their input and performance in Supplementary Table 4. It is worth mentioning that pLM-BLAST [46] achieves respectable sensitivity. This is mainly because pLM-BLAST extends the concept of BLAST by replacing invariant substitution matrices with per-residue similarities between protein embeddings. However, as described in the original paper of pLM-BLAST, the primary limitation of pLM-BLAST is the maximum size of the target dataset (as searching enormous, redundant datasets containing millions of sequences would be computationally costly). This is mainly reflected in the following two aspects: (1) pLM-BLAST needs to use per-residue embeddings for global and local alignments, which means that it needs to retain the embeddings of all amino acids for each protein in the target dataset ($N * L_i * D$, N represents the number of proteins, L_i represents the length of the protein, and D represents the embedding dimension). In contrast, PLMSearch only needs to retain per-protein embeddings as the target dataset ($N * 1 * D$). This results in a size difference of 2-3 orders of magnitude and a significant challenge when implementing a dataset with the size of UniRef50 containing 53.6M proteins [8]. (2) pLM-BLAST needs to calculate the similarity between protein pairs through specific global and local alignments, based on COS similarity prefiltering but still slower than most search methods. Overall, PLMSearch searches faster than pLM-BLAST and only requires per-protein embeddings rather than larger per-residue embeddings. However, pLM-BLAST is able to generate specific sequence alignments based on residue-level information, especially local alignments.

In the future, we intend to develop a deep learning method PLMAlign to improve sequence alignment. In the first step of large-scale searching, PLMSearch only considers the per-protein embedding averaged by per-residue embedding to improve efficiency and save RAM space. Subsequently, for potential homologous query-target pairs searched by PLMSearch, PLMAlign calculates the mutual attention between query-target per-residue embeddings based on the transformer network in deep learning, which is expected to explore better global and local sequence alignment results.

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 large-scale homologous protein search method.

4 Methods

4.1 PLMSearch pipeline

PLMSearch consists of three steps (Fig. 1). (1) PfamClan. Initially, PfamScan [50] identifies the Pfam clan domains of the query protein sequences. Subsequently, PfamClan searches the target dataset for proteins sharing the same Pfam clan domain with the query proteins. 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 proteins and target proteins will be retained. (2) Similarity prediction. The protein language model generates deep sequence embeddings for query and target proteins. Subsequently, SS-predictor predicts the similarity of all query-target pairs. (3) Search result. Finally, PLMSearch selects the similarity of the protein pairs prefiltered by PfamClan, sorts the protein pairs based on their similarity, and outputs the search results for each query protein separately.

For the searched query-target pairs, we added parallel sequence-based Smith-Waterman alignment and structure-based TM-align at the end of the pipeline, so that users can easily generate their interested alignments and obtain corresponding sequence identity and TM-score. We admit that the sequence-based Smith-Waterman alignment may be inconsistent with the predicted similarity by PLMsearch, especially for remote homologs. In the future, the newly developed deep learning-based approach PLMAlign will use per-residue embeddings to generate more precise sequence alignments.

4.2 PfamClan

PfamClan filters out protein pairs that share the same Pfam clan domain (Fig. 1a). It is worth noting that the recall rate is more important in the

initial pre-filtering. PfamClan is based on a more relaxed standard of sharing the same Pfam clan domain, instead of sharing the same Pfam family domain (what PfamFamily does). This feature allows PfamClan to outperform PfamFamily in recall rate (Fig. 4) and successfully recalls high TM-score protein pairs that PfamFamily misses (Extended Data Table A6).

4.3 Similarity prediction

Based on the protein language model and SS-predictor, PLMSearch performs further similarity prediction based on the pre-filtering results of PfamClan (Fig. 1b). The motivation is that the clustering results based on PfamClan show a significant long-tailed distribution (Extended Data Fig. A2). As the size of the dataset increases, the number of proteins contained in big clusters will greatly expand, further leading to a rapid increase in the number of pre-filtered protein pairs (Supplementary Table 3). The required computing resources are excessive with TM-align used for all the filtered pairs. PLMSearch uses the predicted similarity instead, which helps to greatly increase speed and avoids dependence on structures.

As shown in Fig. 5a, the input protein sequences are first sent to the protein language model (ESM-1b here) to generate per-residue embeddings ($m * d$, where m is the sequence length and d is the dimension of the vector), and the per-protein embedding ($1 * d$) is obtained through the average pooling layer. Subsequently, SS-predictor predicts the structural similarity (TM-score) between proteins through a bilinear projection network (Fig. 5b). However, it is difficult for the predicted TM-score to directly rank protein pairs with extremely high sequence identity (often differing by only a few amino acids). This is because the predicted TM-score was trained on SCOPe40-train, where protein pairs share sequence identity < 0.4, so cases with extremely high sequence identity were not included. At the same time, COS similarity performs well in cases with extremely high sequence identity (see P@1 in Extended Data Fig. A3d-e) but becomes increasingly insensitive to targets after Top-10. Therefore, the final similarity predicted by SS-predictor is composed of the predicted TM-scores and the COS similarity to complement each other. We studied the reference similarity of COS similarity in the search test of SCOPe40-train (Supplementary Fig. 3) and found that when COS similarity > 0.997, the probability that the query-target pair belongs to the same fold is 100%. Therefore, we assemble the predicted TM-score with the top COS similarity as follows. When COS similarity > 0.997, SS-predictor similarity equals COS similarity. Otherwise, SS-predictor similarity equals the predicted TM-score multiplied by COS similarity. Ablation experiments prove that combining COS similarity improves the sensitivity of SS-predictor

(Extended Data Fig. A3b-e, Extended Data Table A1, Extended Data Table A2).

We also studied the reference similarity of SS-predictor similarity in the search test of SCOPe40-train. Fig. 6a shows that when similarity < 0.6, there are almost no protein pairs in the same SCOP Fold. On the other hand, when similarity > 0.8, the probability of the two proteins in the same SCOP Fold rapidly increases to 99.4% (See Supplementary Table 5). There is a clear phase transition occurring around the similarity of $0.6-0.8$. In the ridge plot Fig. 6b, as expected, the protein pairs in the same fold and different folds are well grouped in two different similarity ranges, i.e. the protein pairs in the same fold have a higher similarity and the protein pairs in different folds have a lower one. However, since similarity and SCOP fold do not have a one-to-one correspondence, there is a small overlap. Furthermore, unlike Foldseek, which focuses on local similarity, the similarity of SS-predictor, like TM-score, focuses on global similarity (Extended Data Table A7).

4.4 Datasets

4.4.1 SCOPe40

The SCOPe40 dataset consists of single domains with real structures. Clustering of SCOPe 2.01 [51, 52] at 0.4 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, 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. It is worth mentioning that each domain in SCOPe40-test belongs to a different fold from all domains in SCOPe40-train, so the difference between training and testing data is much larger than that of pure random division. We also studied the max sequence identity of each protein in SCOPe40-test relative to SCOPe40-train (Supplementary Fig. 4). From the figure, we can draw a similar conclusion that the sequences in SCOPe40-test and SCOPe40-train are quite different, and most of the max sequence identity is between 0.2 and 0.3. PLMSearch performs well in SCOPe40-test (Fig. 2a-c, Extended Data Table A1), implying that PLMSearch learns universal biological properties that are not easily captured by other methods based on sequence alignment [12].

4.4.2 Swiss-Prot

Unlike SCOPe, the Swiss-Prot dataset consists of full-length, multi-domain proteins with predicted structures, which are closer to real-world scenarios. Because the throughput of experimentally observed structures is very low and requires a lot of human and financial resources. The number of real structures in datasets like PDB [53–55] 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 [8] and the predicted structure from the AlphaFold Protein Structure Database [30]. A total of 542,317 proteins with both sequences and predicted structures were obtained. For these proteins, we dropped low-quality proteins with an avg. pLDDT lower than 70, and left 498,659 proteins. In order to avoid possible data leakage issues, like SCOPe40, we used 0.4 sequence identity as the threshold to filter homologs in Swiss-Prot from the training set. Specifically, we used the previously screened 498,659 proteins as query proteins and SCOPe40-train as the target dataset. We first pre-filtered potential homologous protein pairs with MMseqs2 and calculated the sequence identity between all these pairs. The query protein from Swiss-Prot will be discarded if the sequence identity between the query protein and any target protein is greater than or equal to 0.4. Finally, 68,519 proteins were deleted via homology filtering, leaving 430,140 proteins in Swiss-Prot, which we employed in our experiments. We also studied the max sequence identity of each protein in Swiss-Prot relative to SCOPe40-train (Supplementary Fig. 4) and found that the vast majority of them were below 0.3.

Subsequently, we randomly selected 50 queries from Swiss-Prot and 50 queries from SCOPe40-test as query proteins (a total of 100 query proteins) and searched for 430,140 target proteins in Swiss-Prot. Therefore, a total of 43,014,000 query-target pairs were tested. The search test for 50 query proteins from Swiss-Prot and SCOPe40-test are called "Swiss-Prot to Swiss-Prot" and "SCOPe40-test to Swiss-Prot" respectively.

4.4.3 Target datasets on web server

We currently have the following four target datasets for users to search: (1) Swiss-Port (568K proteins) [8], the original dataset without de-redundancy; (2) PDB (680K proteins) [53–55]; (3) UniRef50 (53.6M proteins) [8]. UniRef50 is built by clustering UniRef90 seed sequences that have at least 50% sequence identity to and 80% overlap with the longest sequence in the cluster; (4) Self (the query dataset itself). Searching a query on Swiss-Prot and UniRef50 takes approximately 0.03 min and 0.5 min respectively (Extended Data Table A4).

4.5 Metrics

We evaluated different homologous protein search methods using the following four metrics: AUROC, AUPR, MAP, and P@K.

- AUROC [23]: The mean sensitivity over all queries, where the sensitivity is the fraction of TPs in the sorted list up to the first FP, all excluding self-hits.
- AUPR [23]: Area under the precision-recall curve.
- MAP [56]: Mean average precision (MAP) for a set of queries is the mean of the average precision scores for all query proteins.

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \quad (1)$$

where Q is the number of queries.

- P@K [19]: For homologous protein search, as many queries have thousands of relevant targets, and few users will be interested in getting all of them. Precision at k (P@k) is then a useful metric (e.g., P@10 corresponds to the number of relevant results among the top 10 retrieved targets). P@K here is the mean value for each query.

On SCOPe40-test, we performed an all-versus-all search test, which means both the query and the target dataset were SCOPe40-test. To make a more objective comparison, the settings used in the all-versus-all search test are exactly the same as those used in Foldseek [23]. Specifically, for family-level, superfamily-level, and fold-level recognition (Supplementary Fig. 5), TPs were defined as the same family, same superfamily but different family, and same fold but different superfamily, respectively. Hits from different folds are FPs. After sorting the alignment result of each query according to similarity (described in Supplementary Table 6), we calculated the sensitivity as the fraction of TPs in the sorted list up to the first FP to better reflect the requirements for low false discovery rates in automatic searches. We then took the mean sensitivity over all queries as AUROC. Additionally, we plotted weighted precision-recall curves (precision = TP/(TP+FP) and recall = TP/(TP+FN)). All counts (TP, FP and FN) were weighted by the reciprocal of their family, superfamily or fold size. In this way, families, superfamilies and folds contribute linearly with their size instead of quadratically [13]. MAP and P@K were also calculated according to the TPs and FPs defined by fold-level.

On search tests against Swiss-Prot, TPs were defined as pairs with a TM-score > 0.5, otherwise FPs. MAP, P@K is then calculated. The reason is that it is problematic to evaluate the SCOPe/-CATH domain annotations of multi-domain proteins on search tests against Swiss-Prot [23]. 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.

4.6 Baselines

4.6.1 Previously proposed methods

(1) Sequence search: MMseqs2 [9], BLASTp [10], HHblits [15, 16], EAT [44], and pLM-BLAST [46]. (2) Structure search — structural alphabet: 3D-BLAST-SW [21], CLE-SW [22], Foldseek, and Foldseek-TM [23]. (3) Structure search — structural alignment, reviewed in [57]: CE [24], Dali [25], and TM-align [26, 27]. 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 query-target pairs (Fig. 5c). Different search methods are distinguished according to the way they predict similarity.

- Euclidean: Use the reciprocal of the Euclidean distance between embeddings. ϵ is a small value to avoid division by zero.

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

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

$$similarity(p, q) = \frac{p \cdot q}{\max\left(\|p\|_2 \cdot \|q\|_2, \epsilon\right)} \quad (3)$$

- SS-predictor (w/o COS): Use the predicted TM-score without combining COS similarity.
- SS-predictor: Combine the predicted TM-score with the top COS similarity. The difference from PLMSearch is that SS-predictor predicts all protein pairs instead of pairs pre-filtered by PfamClan.

4.7 Experiment settings

4.7.1 Pfam result generation

We obtained the Pfam family domains of proteins by PfamScan (version 1.6) [50]. For PfamClan, we query the comparison table Pfam-A.clans.tsv 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 [34], a SOTA general-purpose protein language model, to generate protein embeddings.

4.7.3 SS-predictor training

We used the deep learning framework PyTorch (version 1.7.1), ADAM optimizer, with MSE as the loss function to train SS-predictor. The batch size was 100, and the learning rate was 1e-5 on 20 epochs. The training ground truth was the TM-score calculated by TM-align. The dataset used for training was all protein pairs in SCOPe40-train (8,953 proteins). TM-scores for all protein pairs between them were calculated. Therefore, a total of 80,156,209 query-target pairs were used to train SS-predictor. The parameters in ESM-1b were frozen and only the parameters in the bilinear projection network were trained.

4.7.4 Experimental environment

We conducted the experiments on a server with a 56-core Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz and 256 GB RAM. The environment of our publicly available web server is 64 * Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz and 512 GB RAM.

5 Data availability

The sequences and structures of the SCOPe40 dataset are available at https://scop.berkeley.edu. The sequences of the Swiss-Prot dataset are freely available under the Creative Commons Attribution (CC BY 4.0) License from https://www.uniprot.org. The predicted structures are freely available from the AlphaFold Protein Structure Database at https://alphafold.ebi.ac.uk/download. Pfam is freely available under the Creative Commons Zero ('CC0') license from https://pfam.xfam.org. For PfamClan, we query the comparison table Pfam-A.clans.tsv at https://ftp.ebi.ac.uk/pub/databases/Pfam/current_release/Pfam-A.clans.tsv.gz. ESM-1b protein language model is available at https://github.com/facebookresearch/esm. Source data of our work is provided at https://dmiip.sjtu.edu.cn/PLMSearch/static/download/plmsearch_data.tar.gz.

6 Code availability

PLMSearch and related tutorials are freely available to the public at GitHub https://github.com/maovshao/PLMSearch/blob/main/pipeline.ipynb. Reproducing our results and regenerating the main and supplementary figures requires only one file at GitHub https://github.com/maovshao/PLMSearch/blob/main/main.ipynb.

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] Torres, M., Yang, H., Romero, A. E. & Paccanaro, A. Protein function prediction for newly sequenced organisms. *Nature Machine Intelligence* **3**, 1–11 (2021) .

[5] Unsal, S. *et al.* Learning functional properties of proteins with language models. *Nature Machine Intelligence* **4**, 227–245 (2022) .

[6] 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 .

[7] 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) .

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

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

[10] 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) .

[11] Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale using diamond. *Nature methods* **18** (4), 366–368 (2021) .

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

[13] Eddy, S. R. Accelerated profile hmm searches. *PLoS computational biology* **7** (10), e1002195 (2011) .

[14] Söding, J. Protein homology detection by HMM–HMM comparison. *Bioinformatics* **21** (7), 951–960 (2004) .

[15] Remmert, M., Biegert, A., Hauser, A. & Söding, J. Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment. *Nature methods* **9** (2), 173–175 (2012) .

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

[17] 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) .

[18] Ovchinnikov, S. *et al.* Protein structure determination using metagenome sequence data. *Science* **355** (6322), 294–298 (2017) .

[19] Buchan, D. W. A. & Jones, D. T. EigenTHREADER: analogous protein fold recognition by efficient contact map threading. *Bioinformatics* **33** (17), 2684–2690 (2017) .

[20] Bhattacharya, S., Roche, R., Moussad, B. & Bhattacharya, D. Discover: distance- and orientation-based covariational threading for weakly homologous proteins. *Proteins: Structure, Function, and Bioinformatics* **90** (2), 579–588 (2022) .

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

[22] 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) .

[23] van Kempen, M. *et al.* Fast and accurate protein structure search with foldseek. *Nature Biotechnology* 1–4 (2023) .

[24] 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) .

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

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

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

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

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

[30] 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) .

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

[32] 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].

[33] 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) .

[34] 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) .

[35] 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) .

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

[37] Elnaggar, A. *et al.* Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence* **44** (10), 7112–7127 (2021) .

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

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

[40] Bileschi, M. L. *et al.* Using deep learning to annotate the protein universe. *Nature Biotechnology* **40** (6), 932–937 (2022) .

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

[42] Nallapareddy, V. *et al.* CATHe: detection of remote homologues for CATH superfamilies using embeddings from protein language models. *Bioinformatics* **39** (1) (2023). Btad029 .

[43] Sillitoe, I. *et al.* Cath: increased structural coverage of functional space. *Nucleic acids research* **49** (D1), D266–D273 (2021) .

[44] Heinzinger, M. *et al.* Contrastive learning on protein embeddings enlightens midnight zone. *NAR Genomics and Bioinformatics* **4** (2) (2022). Lqac043 .

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

[46] Kaminski, K., Ludwiczak, J., Pawlicki, K., Alva, V. & Dunin-Horkawicz, S. plm-blast – distant homology detection based on direct comparison of sequence representations from protein language models. *bioRxiv* (2023) .

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

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

[49] 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) .

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

[51] 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) .

[52] 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) .

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

[54] 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) .

[55] 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) .

[56] Wikipedia contributors. Evaluation measures (information retrieval) — Wikipedia, the free encyclopedia (2023).

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

7 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.

8 Competing interests

All authors declare no competing interests.

9 Figures and Tables

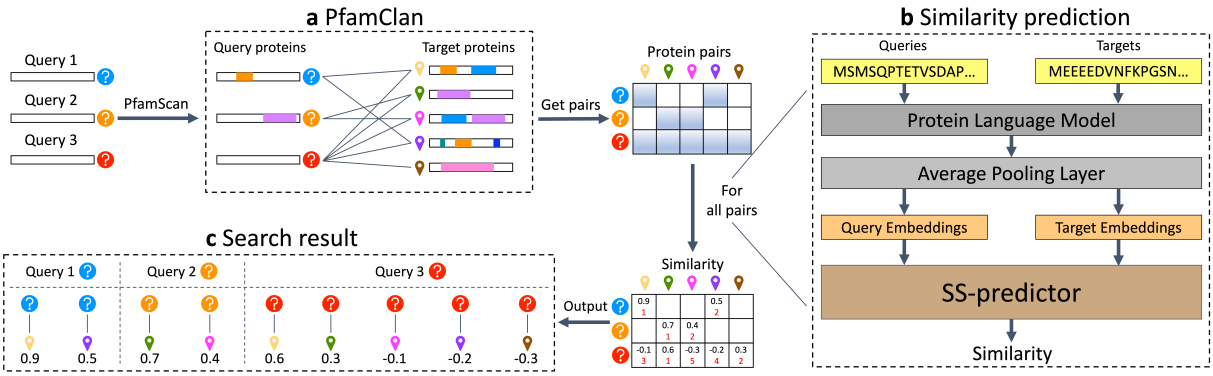


Fig. 1 Overview of the PLMSearch pipeline. a, PfamClan. Initially, PfamScan [50] identifies the Pfam clan domains of the query protein sequences, which are depicted in different color blocks. Subsequently, PfamClan searches the target dataset for proteins sharing the same Pfam clan domain with the query 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 query and target proteins. Subsequently, SS-predictor predicts the similarity of all query-target pairs. **c,** Search result. Finally, PLMSearch selects the similarity of the protein pairs pre-filtered by PfamClan, sorts these protein pairs based on their predicted similarity, and outputs the search results for each query protein separately.

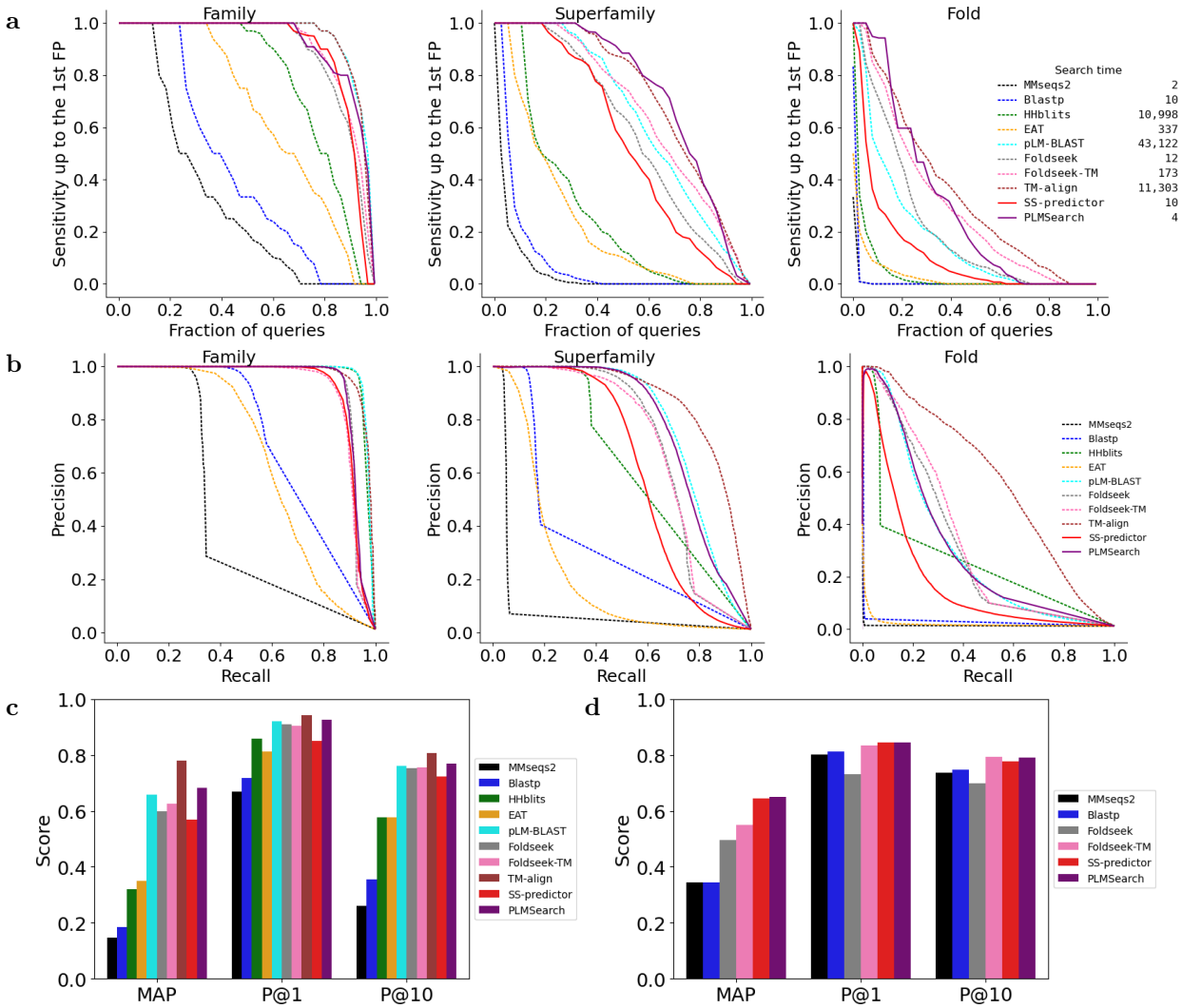


Fig. 2 PLMsearch reaches similar sensitivities as structure search methods. a-c, The all-versus-all search test on SCOPe40-test. For family-level, superfamily-level, and fold-level recognition, TPs were defined as same family, same superfamily but different family, and same fold but different superfamily, respectively. Hits from different folds are FPs. After sorting the alignment result of each query according to similarity, we calculated the sensitivity as the fraction of TPs in the sorted list up to the first FP to better reflect the requirements for low false discovery rates in automatic searches. **a,** We took the mean sensitivity over all queries as AUROC. In addition, the total search time for the all-versus-all search test with a 56-core Intel(R) Xeon(R) CPU E5-2680 v4 @ 2.40 GHz and 256 GB RAM server is shown on the legend. **b,** Precision-recall curve. **c,** MAP, P@1, and P@10. **d,** MAP, P@1, and P@10 on a search test with Swiss-Prot as the target dataset. Extended Data Table A1, Extended Data Table A2 records the specific values of each metric.

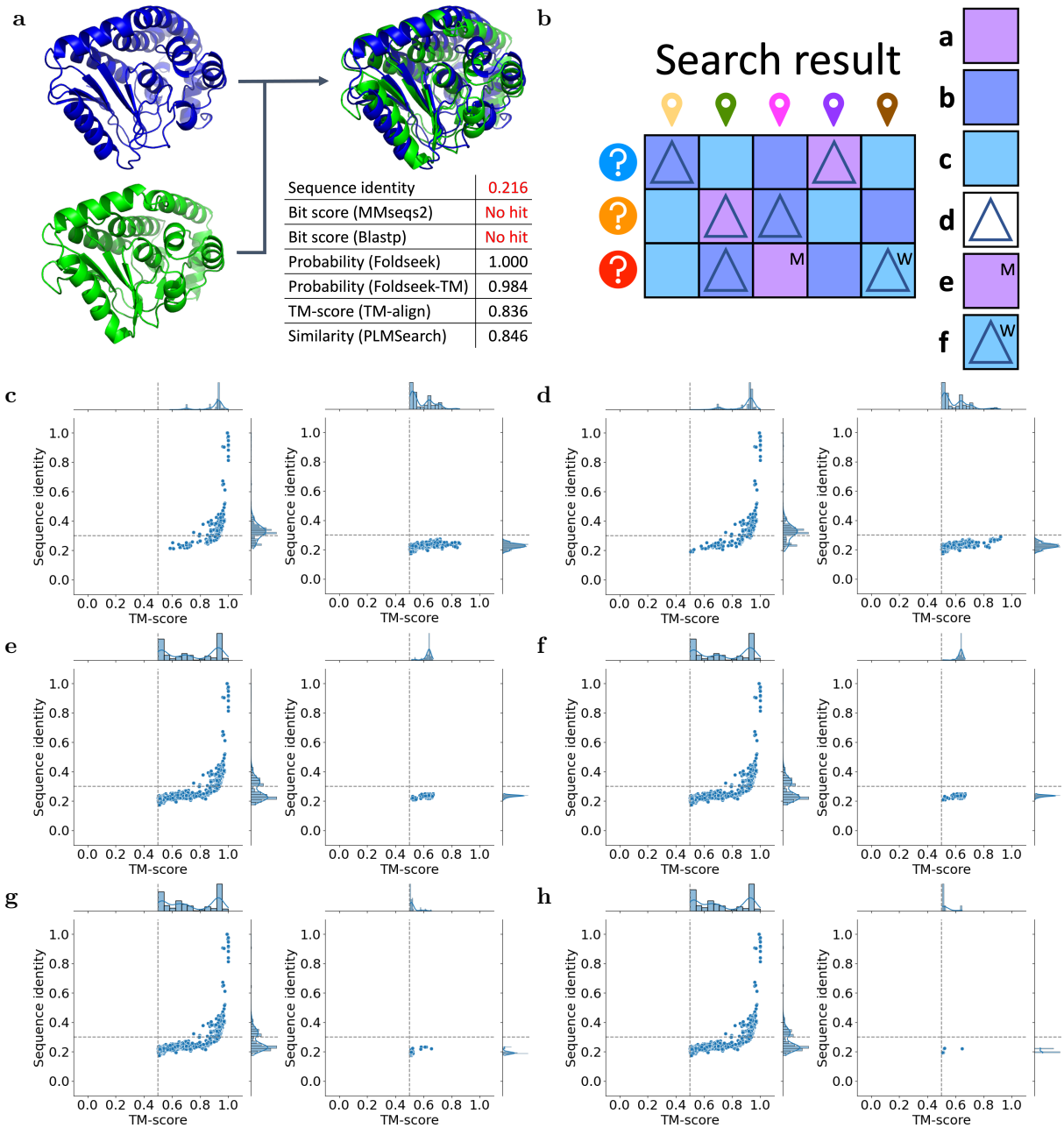


Fig. 3 PLMSearch accurately detects remote homology pairs. a, Case study. Protein pairs with dissimilar sequences but similar structures are called remote homology pairs here. The sequence identity between the Q08558 (the blue structure) and I6Y3U6 (the green structure) is low (0.216<0.3). Thus, it is difficult to find this remote homology pair with sequence alignment only. However, Foldseek, Foldseek-TM, TM-align, and PLMSearch capture the remote homology pair that is missed by MMseqs2 and Blastp. **b,** Definition diagram. This search result example is composed of three queries and five targets. Among the 15 protein pairs, **Legend a** marks three pairs with a TM-score > 0.5, usually assumed to have the same fold [48, 49]. **Legend b** marks six pairs with a TM-score between 0.2-0.5. **Legend c** marks six pairs with a TM-score < 0.2, usually assumed as randomly selected irrelevant pairs [48, 49]. **Legend d** marks six filtered pairs. **Legend e** marks the pair at (3,3) with a TM-score > 0.5 but is not filtered out, which is a "Missed" pair. Correspondingly, protein pairs in (1,4) and (2,2) are "Recalled" pairs. **Legend f** marks the pair at (3,5) with a TM-score < 0.2 but is filtered out, which is a "Wrong" pair. **c-h,** From the search results of five randomly selected queries to avoid oversampling (with Swiss-Prot as the target dataset, a total of 2,150,700 query-target pairs), we selected the 5000 pairs with the highest similarity for different search methods and counted the recalled and missed pairs: **c,** MMseqs2. **d,** Blastp. **e,** Foldseek. **f,** Foldseek-TM. **g,** SS-predictor. **h,** PLMSearch. For recalled pairs (left) and missed pairs (right) in each subplot, 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 [47] and TM-score = 0.5 [48, 49], are shown by dashed lines. All methods successfully recall the easy pairs in the first quadrant. But for remote homology pairs in the fourth quadrant, SS-predictor & PLMSearch did the best, followed by Foldseek & Foldseek-TM, and MMseqs2 & Blastp were the worst. Extended Data Table A5 records the specific values of each metric.

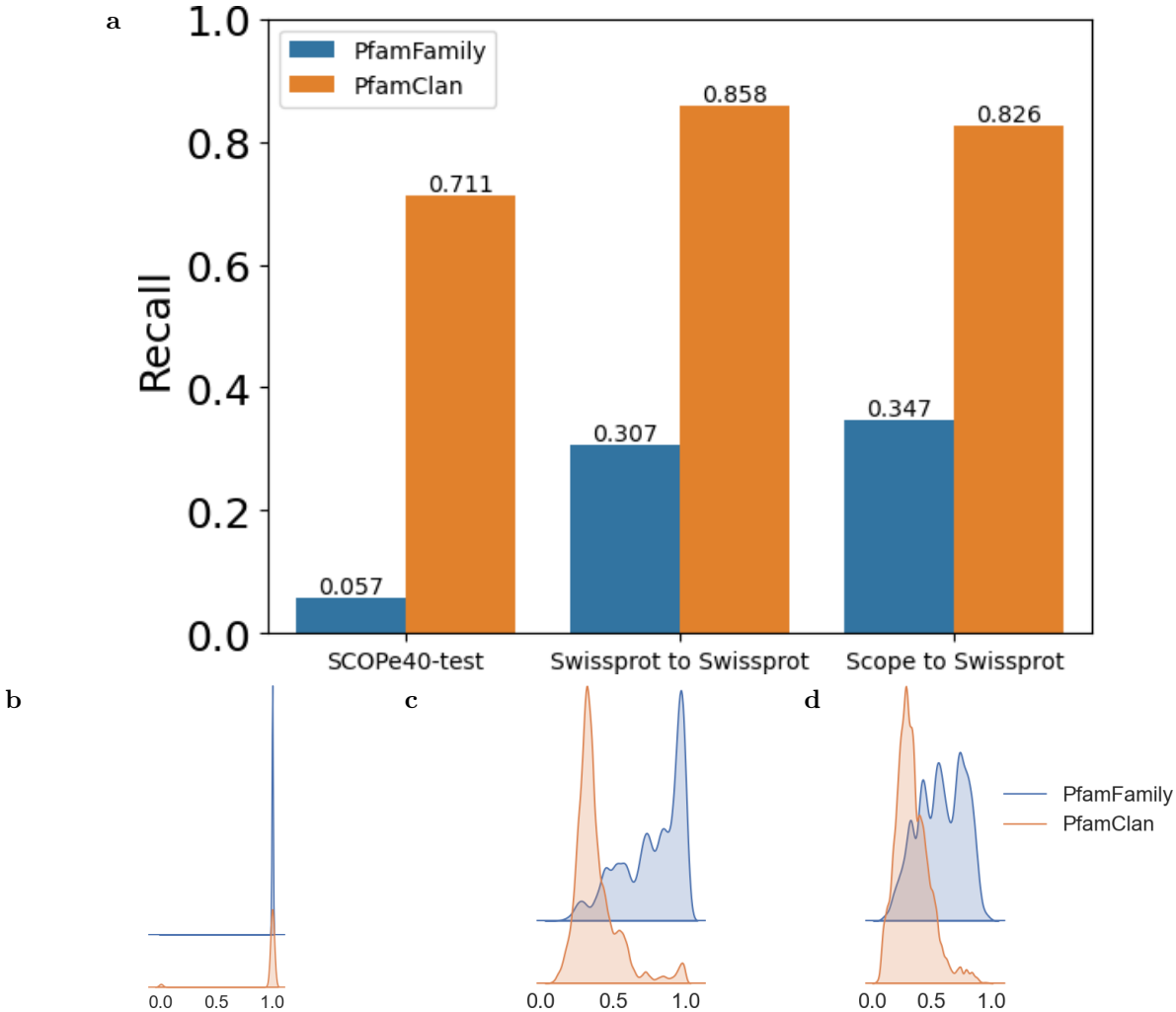


Fig. 4 The pre-filtering results of PfamFamily & PfamClan. a, We evaluated the pre-filtering results of PfamFamily & PfamClan on the SCOPe40-test, Swiss-Prot to Swiss-Prot, and SCOPe40-test to Swiss-Prot search tests (see "Datasets"). PfamClan achieves a higher recall rate. **b,** Same(1) or Different(0) fold on SCOPe40-test, **c-d,** TM-score distributions using kernel density estimation (smoothed histogram using a Gaussian kernel with the width automatically determined). **c,** Swiss-Prot to Swiss-Prot; **d,** SCOPe40-test to Swiss-Prot. The distribution of PfamFamily is to the right as a whole, because the requirements of PfamFamily are stricter than PfamClan, so the protein pair it recalls has a higher probability of being in the same fold and sharing a higher TM-score. However, this also leads to PfamFamily having a lower recall rate and missing some homologous protein pairs as shown in Extended Data Table A6. It is worth noting that the recall rate is more important in the initial pre-filtering.

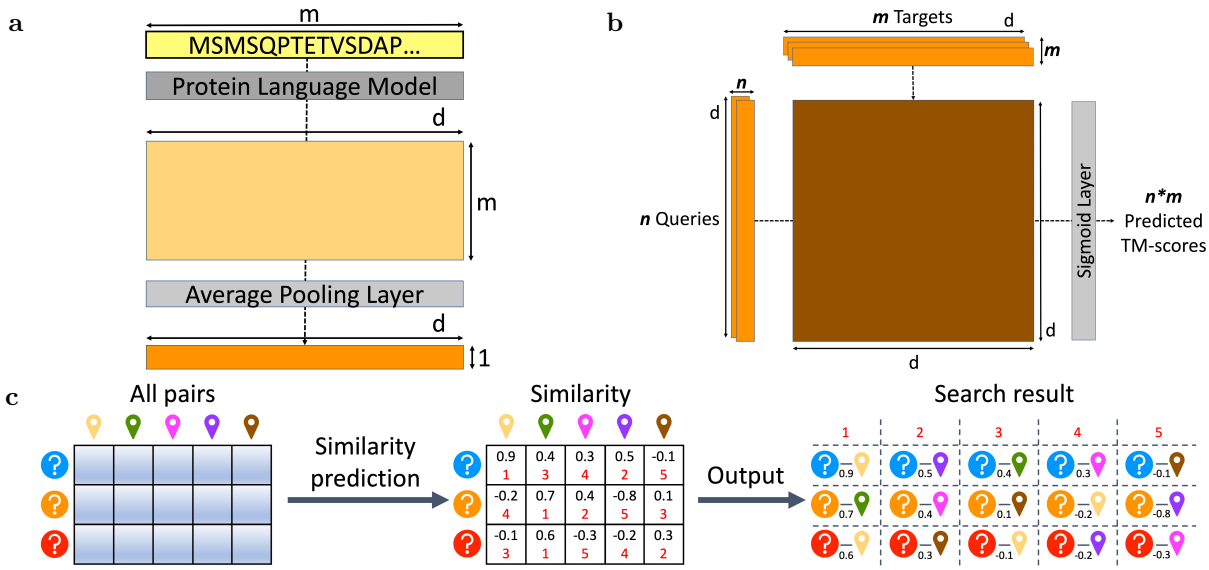


Fig. 5 SS-predictor. a, Protein embedding generation. The protein language model converts the protein sequence composed of m amino acids into $m * d$ per-residue embeddings, where d is the dimension of the embedding. Then the average pooling layer converts them into the corresponding $1 * d$ per-protein embedding z. **b,** SS-predictor. A bilinear projection network predicts all the TM-scores between n query embeddings $z1$ and m target embeddings $z2$ by $z1 * W * z2$, where the matrices W are the learned parameters. **c,** Similarity-based search methods. The similarity of all protein pairs is predicted and sorted and then outputted as a search result. According to different similarity, four corresponding search methods are: Euclidean, COS, SS-predictor (w/o COS), and SS-predictor.

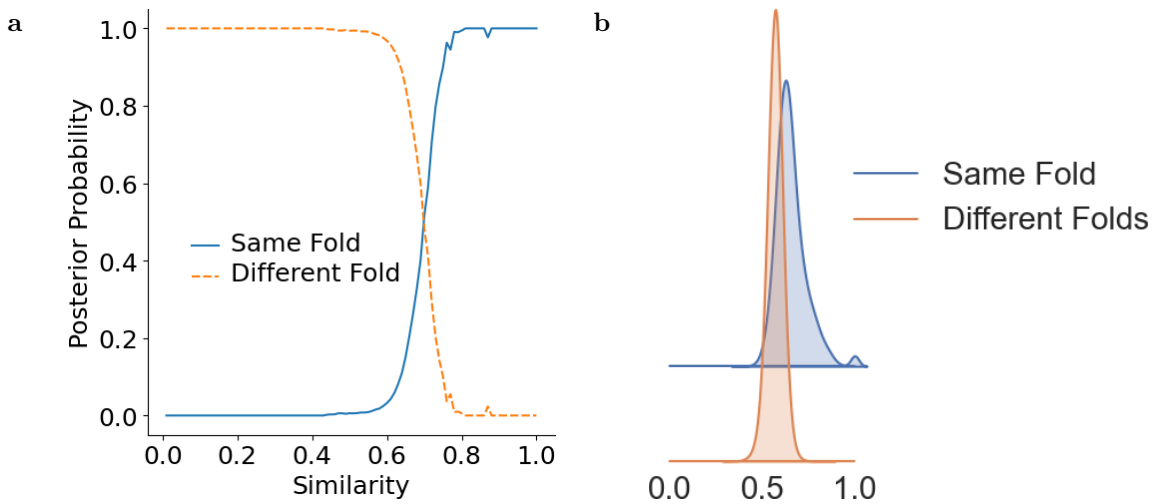
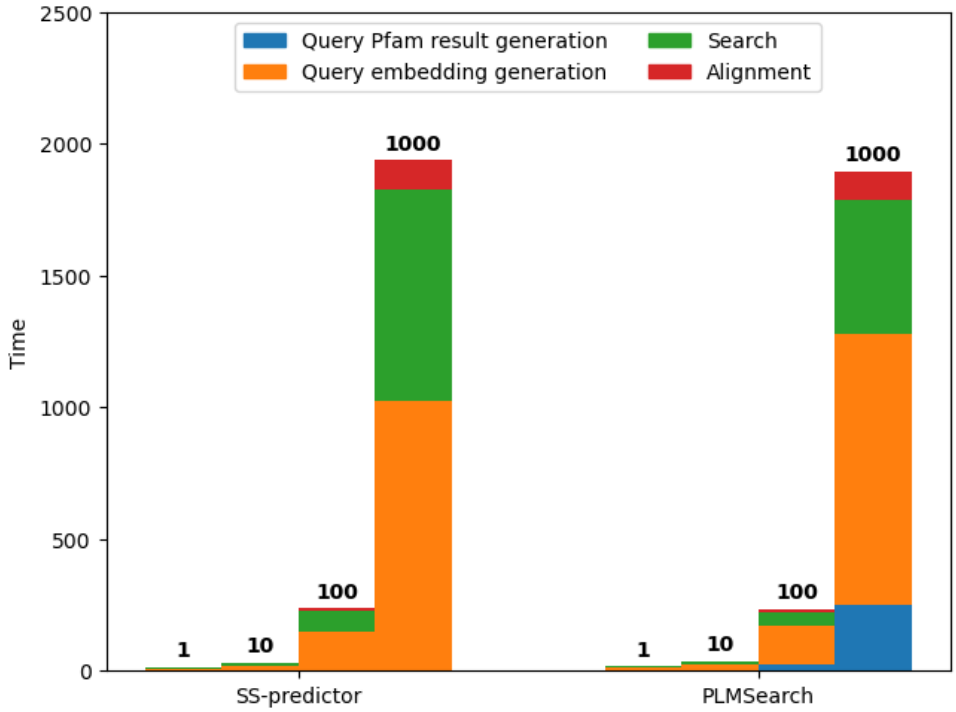
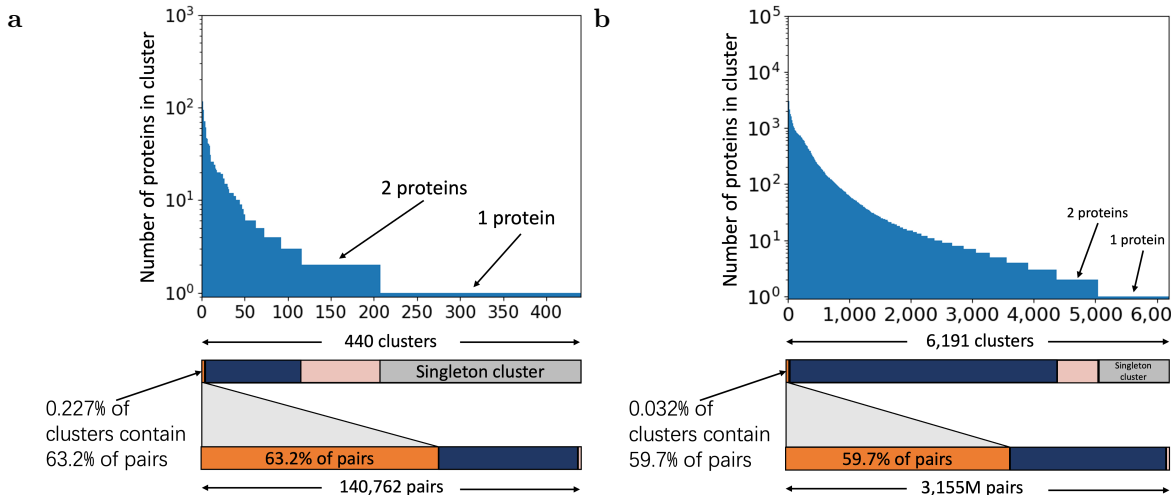


Fig. 6 Reference similarity. a, The posterior probability of proteins with a given similarity being in the same fold or different folds in SCOPe40-train. The posterior probability corresponding to the similarity is shown in Supplementary Table 5. Protein pairs with a similarity lower than 0.6 is usually assumed as randomly selected irrelevant protein pair. Protein pairs with a similarity higher than 0.8 is usually assumed to share the same fold. **b,** Similarity distribution of the same and different folds protein pairs using kernel density estimation (smoothed histogram using a Gaussian kernel with the width automatically determined). The similarity of protein pairs belonging to the same protein pair is significantly higher than that of protein pairs belonging to different folds.

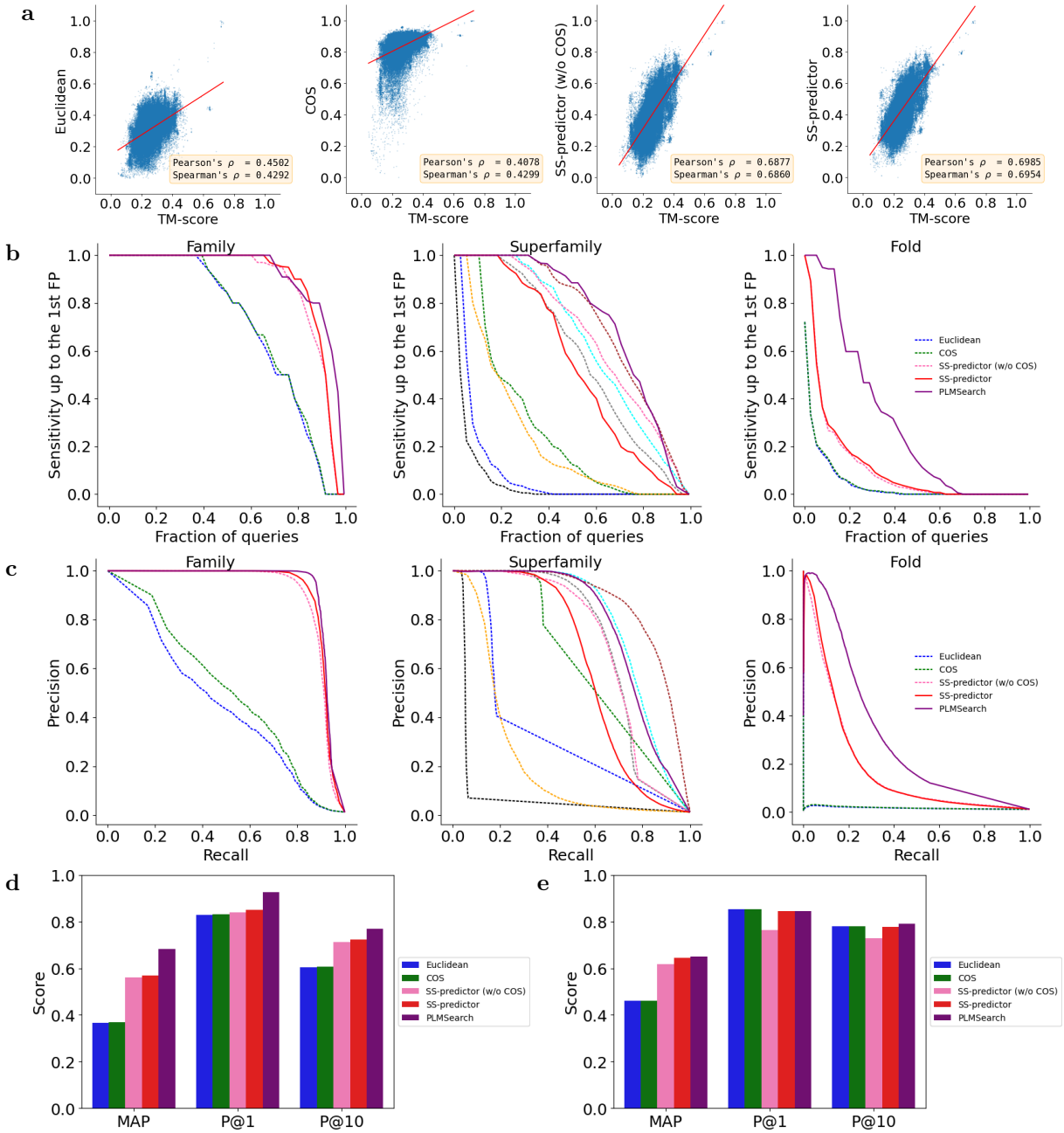
Appendix A Extended Data



Extended Data Fig. A1 Running time of the web server. The environment of the web server is CPU ONLY, with 64 * Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz and 512 GB RAM. The time required to search 1, 10, 100, and 1,000 query proteins for SS-predictor and PLMSearch with Swiss-Port as the target dataset were counted respectively. The time required for each step is divided by blocks of different colors, of which the preprocessing time (Query Pfam result generation and Query embedding generation) occupies the vast majority. The total run time and the detailed step run time are shown in Extended Data Table A4 and Supplementary Table 1 respectively.



Extended Data Fig. A2 Clustering results based on Pfam clan on SCOPe40-test and Swiss-Prot. a, SCOPe40-test. **b,** Swiss-Prot. Proteins belonging to the same Pfam clan are clustered. The clustering results show a significant long-tailed distribution. After pre-filtering with PfamClan, more than 50% of the pre-filtered protein pairs (orange rectangles in the figure) are from the largest 1-2 clusters (big clusters), which only accounts for a very small part of the entire clusters (SCOPe40-test: 0.227%; Swiss-Prot: 0.032%). Therefore, big clusters will result in a significant number of irrelevant protein pairs in the pre-filtering results, reducing accuracy, and must be further sorted and filtered based on similarity, which is what SS-predictor does. See Supplementary Table 3 for specific statistical data.



Extended Data Fig. A3 Ablation experiments: PfamClan and SS-predictor make PLMSearch more robust. **a,** Two-dimensional scatter plot of the predicted similarity and TM-score. From left to right are Euclidean, COS, SS-predictor (w/o COS), and SS-predictor. We randomly selected 100,000 protein pairs from the search results of five queries (with Swiss-Prot as the target dataset, 100,000 among a total of 2,150,700 query-target pairs) 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. SS-predictor obtained the highest correlation coefficient with TM-score. **b-e,** Ablation experiments, with the same metrics used in Fig. 2. Extended Data Table A1, Extended Data Table A2 records the specific values of each metric. SS-predictor and PLMSearch perform better than other similarity-based search methods. The exception is P@1 in **e**, where Euclidean and COS perform better. In all search tests, we generally observe that Euclidean and COS perform well at P@1 and P@10, but the accuracy drops rapidly after Top-10. SS-predictor achieves higher accuracy by fusing the top COS in the final similarity.

	AUROC			AUPR			MAP	P@K	
Methods	Family	Superfamily	Fold	Family	Superfamily	Fold	MAP	P@1	P@10
Sequence search									
MMseqs2	0.318	0.050	0.002	0.430	0.091	0.014	0.147	0.668	0.260
Blastp	0.527	0.161	0.004	0.717	0.342	0.029	0.183	0.717	0.354
HHblits	0.920	0.363	0.064	0.969	0.623	0.256	0.320	0.858	0.577
EAT	0.648	0.230	0.025	0.646	0.225	0.020	0.350	0.813	0.575
pLM-BLAST	**0.940**	0.642	0.176	**0.973**	0.779	0.305	0.659	0.921	0.760
Structure search — structural alphabet									
3D-BLAST-SW	0.653	0.255	0.045	0.621	0.264	0.047	0.446	0.825	0.604
CLE-SW	0.672	0.265	0.033	0.432	0.171	0.035	0.440	0.814	0.592
Foldseek	0.883	0.584	0.214	0.921	0.703	0.320	0.598	0.908	0.751
Foldseek-TM	0.898	0.664	0.296	0.906	0.695	0.337	0.626	0.905	0.756
Structure search — structural alignment									
CE	0.847	0.527	0.148	0.882	0.627	0.245	0.618	0.897	0.734
Dali	0.923	0.702	0.281	0.948	0.814	0.454	0.702	0.927	0.790
TM-align	0.935	0.721	**0.346**	0.971	**0.866**	**0.569**	**0.781**	**0.941**	**0.806**
Our methods									
Euclidean	0.699	0.309	0.039	0.456	0.107	0.016	0.365	0.829	0.603
COS	0.705	0.316	0.040	0.514	0.130	0.017	0.367	0.830	0.606
SS-predictor (w/o COS)	0.874	0.515	0.108	0.901	0.601	0.179	0.559	0.840	0.711
SS-predictor	0.889	0.542	0.113	0.913	0.613	0.183	0.569	0.850	0.723
PLMSearch	0.917	**0.733**	0.298	0.925	0.768	0.312	0.683	0.926	0.770

Extended Data Table A1 All-versus-all search test on the SCOPe40-test dataset of single-domain structures. The definition of AUROC, AUPR, MAP, and P@K is detailed in "Metrics" Section. The highest value achieved for each metric is highlighted in bold.

	MAP	P@K	
Methods	MAP	P@1	P@10
Baselines			
MMseqs2	0.345	0.802	0.737
Blastp	0.343	0.812	0.748
Foldseek	0.497	0.732	0.697
Foldseek-TM	0.551	0.833	**0.794**
Our methods			
Euclidean	0.459	**0.854**	0.780
COS	0.461	**0.854**	0.778
SS-predictor (w/o COS)	0.617	0.762	0.729
SS-predictor	0.643	0.843	0.777
PLMSearch	**0.651**	0.843	0.789

Extended Data Table A2 Search test with Swiss-Prot as the target dataset. TPs are protein pairs with TM-scores higher than 0.5. The definition of MAP and P@K is detailed in "Metrics" Section. The highest value achieved for each metric is highlighted in bold.

Methods	Family	Superfamily	Fold	Total
Sequence search				
MMseqs2	2.20	0.51	0.02	2.74
Blastp	4.39	1.21	0.05	5.65
HHblits	9.10	11.25	3.92	24.29
EAT	6.17	6.12	1.65	13.96
pLM-BLAST	8.96	30.57	13.95	53.50
Structure search — structural alphabet				
3D-BLAST-SW	6.95	8.92	1.44	17.32
CLE-SW	6.97	9.65	1.10	17.74
Foldseek	8.57	28.33	14.19	51.10
Foldseek-TM	8.68	34.62	20.79	64.11
Structure search — structural alignment				
CE	8.70	23.82	8.72	41.25
Dali	9.23	33.62	16.74	59.60
TM-align	**9.29**	**38.57**	**24.83**	**72.70**
Our methods				
Euclidean	6.88	8.52	1.70	17.12
COS	6.94	8.79	1.77	17.52
SS-predictor (w/o COS)	8.79	19.89	5.40	34.08
SS-predictor	8.92	21.81	6.01	36.76
PLMSearch	9.24	34.31	22.51	66.07

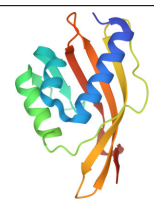
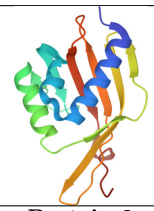
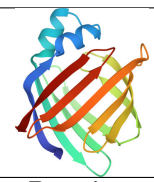
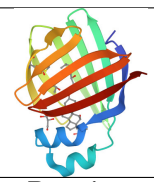
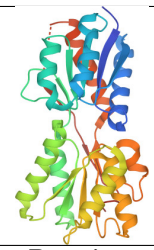
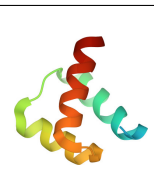
Extended Data Table A3 The average number of family TPs, superfamily TPs, fold TPs, and total TPs up to the first FP on the SCOPe40-test search test. The average number of the total TPs also means the average rank of the first FP. The highest value achieved for each metric is highlighted in bold.

Query num	1	10	100	1,000
Swiss-Prot (568K proteins)				
SS-predictor	0.2 min	0.5 min	3.9 min	32.3 min
PLMSearch	0.2 min	0.6 min	3.9 min	31.5 min
UniRef50 (53.6M proteins)				
SS-predictor	1.8 min	6.1 min	54.8 min	523.5 min
PLMSearch	1.8 min	6.3 min	54.7 min	516.4 min

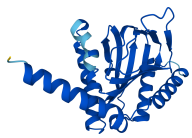
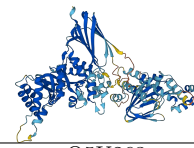
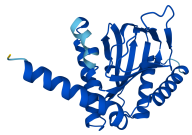
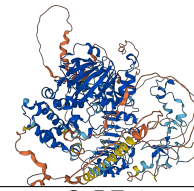
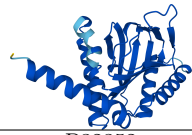
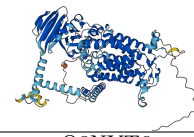
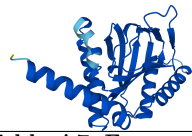
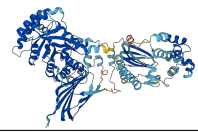
Extended Data Table A4 Total running time of the web server. The environment of the web server is CPU ONLY, with 64 * Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz and 512 GB RAM. The time required to search 1, 10, 100, and 1,000 query proteins with Swiss-Prot (568K proteins, the original dataset without de-redundancy) and UniRef50 (53.6M proteins) as the target dataset were counted respectively. Searching a query against Swiss-Prot and UniRef50 takes approximately 0.03 min and 0.5 min respectively.

	All pairs		Easy pairs		Remote homology pairs	
Methods	Recall	Miss	Recall	Miss	Recall	Miss
MMseqs2	0.450	0.549	**1.000**	**0.000**	0.165	0.834
Blastp	0.462	0.537	**1.000**	**0.000**	0.183	0.816
Foldseek	0.898	0.101	**1.000**	**0.000**	0.845	0.154
Foldseek-TM	0.901	0.098	**1.000**	**0.000**	0.850	0.149
SS-predictor	0.967	0.032	**1.000**	**0.000**	0.951	0.048
PLMSearch	**0.995**	**0.004**	1.000	0.000	**0.993**	**0.006**

Extended Data Table A5 The recall rate of different methods for easy pairs and remote homology pairs. We selected the 5000 pairs with the highest similarity for different search methods and counted the recalled and missed pairs. As shown in Fig. 3 **c-h**, easy pairs refer to the protein pairs with similar sequences and similar structures in the first quadrant. Remote homology pairs refer to the protein pairs with dissimilar sequences but similar structures in the fourth quadrant. All pairs refer to all protein pairs with TM-score > 0.5 in the first and fourth quadrants.

	Protein 1	**Protein 2**
SCOPe ID	d1oh0a_	d1ohpa_
PfamFamily	PF12680 (SnoaL_2)	PF02136 (NTF2)
PfamClan	CL0051 (NTF2)	CL0051 (NTF2)
TM-score	0.926	
Structure		
	Protein 1	**Protein 2**
SCOPe ID	d1ifca_	d2qo4a_
PfamFamily	PF00061 (Lipocalin)	PF14651 (Lipocalin_7)
PfamClan	CL0116 (Calycin)	CL0116 (Calycin)
TM-score	0.884	
Structure		
	Protein 1	**Protein 2**
SCOPe ID	d3c3ka_	d2gx6a_
PfamFamily	PF13377 (Peripla_BP_3)	PF13407 (Peripla_BP_4)
PfamClan	CL0144 (Periplas_BP)	CL0144 (Periplas_BP)
TM-score	0.880	
Structure		
	Protein 1	**Protein 2**
SCOPe ID	d3k2aa_	d1akha_
PfamFamily	PF05920 (Homeobox_KN)	PF00046 (Homeodomain)
PfamClan	CL0123 (HTH)	CL0123 (HTH)
TM-score	0.876	
Structure		

Extended Data Table A6 Case study for the pre-filtering results of PfamFamily & PfamClan. We investigated several protein pairs with TM-score>0.5 but missed by PfamFamily, and found that although the protein pairs do not share the same family, they contain similar families belonging to the same clan. Therefore, pre-filtering with PfamClan instead of Pfamfamily can help recall these protein pairs.

	Query	Target	TM-score		Foldseek	SS-predictor
			Default	Avg. length	Probability	Similarity
UniProt ID	P32352	Q5HJR8	0.343	0.173	1.000	0.607
Length	222	745				
Structure						
UniProt ID	P32352	Q5U263	0.456	0.189	0.795	0.604
Length	222	1,146				
Structure						
UniProt ID	P32352	Q5RF50	0.375	0.190	0.975	0.629
Length	222	758				
Structure						
UniProt ID	P32352	Q8NYT6	0.334	0.168	0.996	0.609
Length	222	745				
Structure						

Extended Data Table A7 Four protein pairs selected for the manual inspection of protein pairs. They are filtered by Foldseek but with a TM-score<0.2 (Wrong pairs, defined in Fig. 3b). TM-align(Default) uses the query protein length as the normalized length. TM-align(Avg. length) uses the average length of protein pairs as the normalized length. As reported in Foldseek's paper, Foldseek searches out these pairs because it focuses on local similarity. However, TM-align and PLMSearch focus on global similarity, so these pairs have TM-score<0.5 and similarity of SS-predictor close to 0.6.