nature biotechnology

Article

https://doi.org/10.1038/s41587-024-02353-6

Fast, sensitive detection of protein homologs using deep dense retrieval

Received: 15 September 2023

Accepted: 12 July 2024

Published online: 9 August 2024

Check for updates

Liang Hong[1,15], Zhihang Hu[1,15], Siqi Sun[2,3,15,16] ✉, Xiangru Tang[4,15], Jiuming Wang[1,5], Qingxiong Tan[1], Liangzhen Zheng[6,7], Sheng Wang[6,7], Sheng Xu[2,3], Irwin King[1], Mark Gerstein[4,8,9,10,16] ✉ & Yu Li[1,3,11,12,13,14,15,16] ✉

The identification of protein homologs in large databases using conventional methods, such as protein sequence comparison, often misses remote homologs. Here, we offer an ultrafast, highly sensitive method, dense homolog retriever (DHR), for detecting homologs on the basis of a protein language model and dense retrieval techniques. Its dual-encoder architecture generates different embeddings for the same protein sequence and easily locates homologs by comparing these representations. Its alignment-free nature improves speed and the protein language model incorporates rich evolutionary and structural information within DHR embeddings. DHR achieves a >10% increase in sensitivity compared to previous methods and a >56% increase in sensitivity at the superfamily level for samples that are challenging to identify using alignment-based approaches. It is up to 22 times faster than traditional methods such as PSI-BLAST and DIAMOND and up to 28,700 times faster than HMMER. The new remote homologs exclusively found by DHR are useful for revealing connections between well-characterized proteins and improving our knowledge of protein evolution, structure and function.

Protein homolog detection is a critical and fundamental component in computational biology, essential for almost all biological sequence-related research, such as protein structure predictions, biomolecular functional analysis, transcription regulation study, novel enzyme discovery and phylogenetic reconstruction[1–5]. In the biological sequence database, homologs represent evolution-related protein sequences with similar structures and functions. Thus, detecting homologs has been widely used as a primary step in evolutionary analysis and benefits drug discovery, disease diagnosis, biomarker prediction[6,7] and protein structure prediction[1]. Before the publication of AlphaFold2 (AF2)[1], protein homolog identification[8] and threading were considered efficient ways to predict protein three-dimensional (3D) structures[9,10]. Despite the impressive power of deep learning in AF2, people have realized the importance of protein homologs and multiple-sequence alignments (MSAs) in the AF2 framework from the CASP15 (Critical Assessment of Structure Prediction)[11] competition;

[1]Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong SAR, China. [2]Research Institute of Intelligent Complex Systems, Fudan University, Shanghai, China. [3]Shanghai AI Laboratory, Shanghai, China. [4]Department of Computer Science, Yale University, New Haven, CT, USA. [5]OneAIM Ltd., Hong Kong SAR, China. [6]Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China. [7]Shanghai Zelixir Biotech Company Ltd., Shanghai, China. [8]Computational Biology and Bioinformatics Program, Yale University, New Haven, CT, USA. [9]Department of Molecular Biophysics and Biochemistry, Yale University, New Haven, CT, USA. [10]Department of Statistics and Data Science, Yale University, New Haven, CT, USA. [11]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, MA, USA. [12]Institute for Medical Engineering and Science, Massachusetts Institute of Technology, Cambridge, MA, USA. [13]Broad Institute of MIT and Harvard, Cambridge, MA, USA. [14]The Chinese University of Hong Kong Shenzhen Research Institute, Shenzhen, China. [15]These authors contributed equally: Liang Hong, Zhihang Hu, Siqi Sun, Xiangru Tang, Yu Li. [16]These authors jointly supervised this work: Siqi Sun, Mark Gerstein, Yu Li. ✉e-mail: siqisun@fudan.edu.cn; mark@gersteinlab.org; liyu@cse.cuhk.edu.hk

AF2-based models with a mixture of MSAs remain top rankers compared to single-sequence-based approaches. Identifying homologs in a sensitive and fast way remains critical. Homolog detection has been extensively studied throughout the past decades with several methods proposed in three main categories. The most popular category constitutes traditional alignment-based methods, such as PSI-BLAST[8], MMseqs2 (ref. 12) and HMMER[13]. The classical algorithm BLAST (basic local alignment search tool) features high speed but fails to take the overall dependency into consideration, missing remote homologs with low sequence similarity. HMMER[13] can implicitly learn complex position-specific rules indicating evolution but is highly dependent on the quality of the profile, which can only be obtained from MSAs that are not always available. Other areas for improvement include the inability to detect structural homology and an insufficient account of protein domain information. The second category constitutes the use of the ranking method for homolog detection. For example, RankProp[14] was inspired by PageRank[15] to use BLASTp to build an all-versus-all sequence similarity network. Although it outperforms BLASTp, such a method cannot be scaled to large databases because of the quadratic growth of link data. Additionally, DeepRank[16] applies convolutional neural networks to represent protein sequences as embedding vectors in feature space and predicts homologous proteins by applying ranking algorithms. However, it is still limited to detecting homologs on the basis of sequence similarity while ignoring the structural information. The third category constitutes the detection of protein homologs by exploiting structural information[1,17–20]. This route has also encountered drawbacks such as a lack of speed and scalability because of structure searching accuracy or the structures themselves needing to be more trustworthy. Finding perfect matches among multiple sequences remains an immensely complex computational task[18,21]; traditional methods are often inadequate in their results and any flaws that may arise remain untouched, making for a less promising alignment of data[18,21].

Further research is needed to develop more advanced models to better tackle these challenges and capture the structural and functional information of proteins while also being computationally efficient. Recently, the approach of pretraining on extensive datasets of protein sequences and subsequently fine-tuning for specific downstream tasks related to sequencing, structure or function has shown promise. Bepler and Berger[22] proposed using a bidirectional long short-term memory (LSTM) model and a multitask framework to encode structural information enabling the transfer of knowledge between structurally related proteins. Rao et al. introduced a transformer-based language model, evolutionary scale modeling (ESM)[23–25], which demonstrated that information learned from protein sequences alone can greatly benefit various downstream tasks, such as secondary-structure prediction and contact prediction, outperforming LSTM-based models[26] by a large margin. Rao et al. also found that integrating MSAs into the model, referred to as the MSA transformer, led to state-of-the-art results on multiple structure-related benchmarks. A similar language model objective was also applied in AF2 (ref. 1), further improving structure prediction accuracy. In the domain of sequencing homology, Bileschi[27] leveraged embeddings from protein language models to concentrate on classification tasks within specific families, facilitating the annotation of novel proteins. Heinzinger[28] and Hamamsy[29] used protein language models to calculate embedding similarity and fit structural similarity measures such as the template modeling score (TM-score). As a result, they inherently exhibit the limitation of performing optimally only when comparing sequences of similar lengths. Although these approaches are innovative in their respective fields, they also manifest certain limitations in their application.

We propose the dense homolog retriever (DHR) framework, based on an advanced protein language model and dense retrieval, to detect remote protein homologs with speed and sensitivity, considering protein structural information implicitly during the process. The system leverages advanced protein language models to encode query sequences and databases for homology comparison through ranking simple similarity metrics on embedded representations. Furthermore, we also incorporate a contrastive learning strategy to aid in training performance enhancement. As an alignment-free method, DHR is orthogonal to most of the aforementioned traditional methods. DHR is not limited to known families nor does it require sequences to be of similar length, making it versatile and broadly applicable for sequence searching and MSA construction. Furthermore, it is efficient enough to retrieve a query from a dataset of 70 million entries in a few seconds on a single graphics processing unit (GPU) and scales up linearly with increased database size. It is 22 times faster than BLAST[30]. In comparison to profile-based methods such as JackHMMER[13], it is up to 28,700 times faster. DHR's trained encoders empowered with protein language models[23,24] capture rich structural and coevolution information[24,25,31,32], enabling us to achieve a >10% increase in sensitivity compared to previous traditional methods. To further showcase the potency of DHR and its usage, we conducted a comprehensive benchmark for homology detection downstream application. When concatenating our alignment-free DHR with JackHMMER, it proved to be 93 times faster than the aforementioned methods while constructing highly consistent MSAs with the AF2 default MSAs on the CASP13 and CASP14 datasets. In addition, DHR together with JackHMMER created a more diverse and comprehensive MSA when considering the number of MSAs and their associated effective number of sequences (Meff; Eq. (2)). Additionally, to demonstrate the efficacy of DHR, we thoroughly assessed the application of DHR-constructed MSAs to the AF2 pipeline, further boosting the prediction accuracy with a 0.4-Å root-mean-square deviation (r.m.s.d.) on average when merged with AF2 default MSA. In our final exploration of DHR's potential, we expanded its application to substantially larger datasets (specifically, BFD/MGnify) and benchmarked our results against an integrative and well-regarded tool, ColabFold-MMseqs2. Notably, DHR shared a 75% sequence overlap with MMseqs2—a profile-based method leveraging BFD/MGnify. Structure predictions using DHR not only compared favorably but also slightly outperformed ColabFold.

Results

An ultrafast and sensitive protein homolog searching pipeline

The main idea underlying the proposed pipeline to detect homologs is to encode the protein sequences into dense embedding vectors and the similarity among sequences can be computed in a very effective manner. To be more specific, we empowered our method with a protein language model by initializing with ESM[23,24] and integrating the contrastive learning technique as a way to effectively train our sequence encoder (database encoder and query encoder). This allows DHR to incorporate rich coevolutionary and structural information[24,25,31,32] with great efficacy in retrieving homologs. Our dual-encoder architecture then offers greater flexibility in our model, allowing the same protein sequence to generate different embeddings depending on its role as a query or candidate sequence (Fig. 1a). Contrastive learning then seeks to learn encoder transformations that embed positive input pairs nearby while pushing negative pairs far apart (Fig. 1b). Following the completion of the training phase in our dual-encoder framework, we are able to generate offline protein sequence embeddings of high quality.

We then leverage these embeddings and similarity search algorithms to retrieve homologs for each query protein (Fig. 1a). By designating similarity as the retrieval metric, we are able to find similar proteins in a much more accurate fashion than with traditional methods and use the similarity between two proteins for further analysis. Then, JackHMMER is incorporated to construct an MSA for our retrieved homologs. Our rapid and effective technique of discovering homologs, named DHR, harnesses dense retrieval to simplify searching within

https://doi.org/10.1038/s41587-024-02353-6

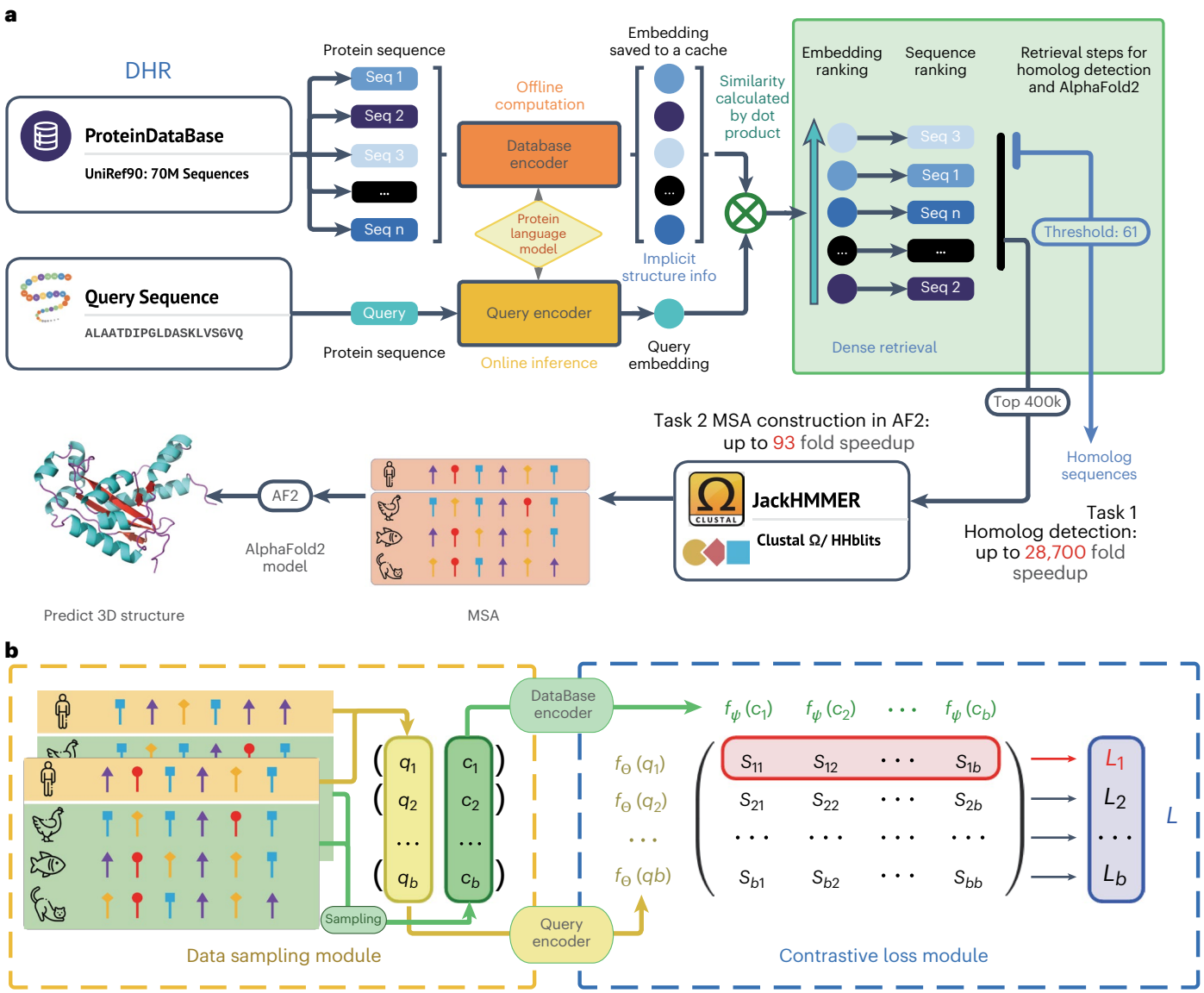


Fig. 1 | Overview of the proposed DHR framework. a, We leveraged a three-level hierarchical structure for our new framework—DHR. The main focus during training is to obtain the protein database encoder and query encoder. Through offline inference on the protein database and caching of embeddings thereafter, we are able to carry out vector-level similarity calculations on the basis of our set query representation, allowing us to generate an ordering for these sequences. For two consecutive downstream tasks, the inference pipeline uses the dot product to acquire the *K* most related sequences. Subsequently, JackHMMER is used on this small retrieved dataset to construct an MSA for further tasks such as 3D structure prediction or protein function forecasting. Before running retrieval, it is possible to encode the UniRef90 into vectors offline without compromising the speed of building MSA. **b**, Training pipeline for the proposed method. In the data sampling module, a batch of positive pairs, $(q_1, c_1), \ldots, (q_b, c_b)$, are sampled from MSAs. Then, a similarity matrix S is computed using the contrastive loss, in which each element $s_{ij} = f_\Theta(q_i)^t f_\Psi(c_j)$. Intuitively, the diagonal of the similarity matrix is trained to be larger than the corresponding off-diagonal elements in the same row because they represent all of the positive pairs.

the protein sequence embeddings without spending excessive time on progressive alignment or dynamic programming that is used in conventional methods. Our alignment-free approach is faster compared to alignment-based methods by avoiding the need to compute pairwise alignment scores over large datasets, with higher accuracy in comparison (Fig. 1a). Moreover, with great efficiency compared to AF2 default MSA construction, DHR can produce diverse MSAs that contain more information when considering their Meff (Eq. (2)). Such information has been proven to be beneficial in protein 3D structure modeling: AF2 with DHR MSA inputs may achieve comparable results to AF2 with default MSA settings. These results suggest that this method can act as an effective and efficient replacement for existing MSA pipelines. Furthermore, diverse MSAs from DHR can complement existing pipelines. We developed a hybrid model, DHR-meta, combining DHR and AF2 default, which outperformed both individual pipelines on CASP13DM (domain sequence) and CASP14DM targets. Additionally, DHR can scale to search large databases such as BFD/MGnify, producing high-quality MSAs[33–36]. These MSAs enhance protein structure prediction, rivaling ColabFold's performance in the CASP15 contest and the New-Protein dataset.

Highly sensitive homolog extraction with structure-aware information

Having obtained the generated protein embeddings, we assessed DHR by comparing it to methods on the standard SCOPe (structural classification of proteins) dataset[37,38]. SCOPe is a database of protein

structural interactions that include the carefully curated ordering of domains from most known proteins in a hierarchy based on structural and evolutionary ties. We conducted studies to investigate how existing homolog retrieval methods and DHR aligned with these curated domain labels. While based solely on sequences, DHR can be compared to other previous sequence-level matching homolog extraction methods such as PSI-BLAST[39], MMseqs2 (ref. 12) and DIAMOND[40] (configurations are detailed in the Methods). We also included a profile-based method, HMMER[41], and a structure-based method, Foldseek[42], to better illustrate DHR's effectiveness. Because the SCOPe dataset groups sequences into families on the basis of structural similarities, the sensitivity for SCOPe families reveals how much structural information these homolog extraction methods will capture. To begin, we used a representative sequence from each family as the query when searching the entirety of SCOPe. To locate all homologs belonging to the same family, we searched the database using representative sequences of each family. Hits (true positives, TPs) are sequences in the retrieved sequences that belong to the same family as the query. False positives (FPs) are defined as hits that are not within the family. Thus, a general comparison can be made between our approach and the baseline methods using their sensitivity rate ($\frac{TP}{TP+FN}$, where FN is the number of false negatives; Methods).

To ensure a fair and comprehensive comparison, we limited our analysis to the top 200 sequences retrieved by each method. DHR demonstrated exceptional performance, detecting notably more homologs (with a sensitivity of 93%) than any other method tested. Another widely used and sensitive method, HMMER, achieved the second-best result with a sensitivity of 91%. In contrast, Foldseek, despite relying solely on structural information, performed similarly to the sequence-based PSI-BLAST (both at ~88% sensitivity). This finding suggests that sequence-based methods, including DHR, are capable of incorporating rich structural information. When using MMseqs2 as a baseline for all sequences and categorizing cases with MMseqs2 sensitivity below 0.34 as hard cases for further investigation, DHR still showed a high average sensitivity over 80% with a perfect sensitivity of 100% in many cases (Fig. 2b). We show that, on several sequences among these hard cases (Fig. 2b), such as d1w0ha, only DHR was able to identify the proper homologs while other methods could not (Fig. 2b and Supplementary Fig. 1). In addition to being ahead in terms of the average results, DHR outperformed the baseline methods in sensitivity for most SCOPe families (Fig. 2c). Moreover, we observed that all methods struggled as SCOPe family sizes increased while DHR maintained a more robust performance (Fig. 2e).

To strengthen our results, we also incorporated another standard metric, the area under the curve before the first FP (AUC-1FP; Methods), on the SCOPe dataset, which is independent of the sequences retrieved by DHR. Both DHR and HMMER showcased the best results, achieving scores of 89% and 88%, respectively (Fig. 2d). Interestingly, the performance of PSI-BLAST and Foldseek was also comparable to that of DHR, a deviation from what we observed with the sensitivity metric. Meanwhile, MMseqs2 showed a slight improvement, reaching 82%, yet still trailing behind DHR by approximately 7%. Although different metrics may yield slightly varied performances, DHR consistently maintained a high level of performance. Furthermore, when considering execution time, DHR greatly outperformed its counterparts. Despite Foldseek, PSI-BLAST and HMMER showing comparable results in terms of AUC-1FP, their execution times were markedly longer. DHR was twice as fast as Foldseek, over 20 times faster than PSI-BLAST and substantially quicker than the profile-based method HMMER, which requires a time-consuming process to build profiles (Supplementary Table 1). On the other hand, MMseqs2, while comparable to DHR in terms of execution time when limited to two iterations, greatly lagged in performance (Fig. 2b,d and Supplementary Table 1). Importantly, the superiority of DHR was not merely a result of replicating our training and searching data. When using BLAST to explore the SCOPe database against our training set UniRef90, we found that most samples yielded fewer than 100 hits (Fig. 2f and Supplementary Table 3). Interestingly, approximately 500 samples returned no hits at all, indicating that they are 'unseen' structures with respect to our dataset. While these structures might pose challenges to DHR, they did not impact other alignment-based methods that are not based on deep learning. Nevertheless, DHR continued to produce high-quality predictions, achieving an AUC-1FP score of 89% (Fig. 2f). This highlights DHR's robustness and efficiency, proving its ability to handle data to which it has not been previously exposed. When extending our analysis to the superfamily level with more challenging remote homologs, all methods (DHR, HMMER, Foldseek and PSI-BLAST) experienced a notable performance decline, with an approximately 10% decrease across the board (Fig. 2d). Despite this drop, DHR managed to maintain a leading performance, achieving an AUC-1FP score of 80%, with HMMER, Foldseek and PSI-BLAST showing slightly lower results (Fig. 2d). We did not include HHblits[43], another mainstream method for homology searching, in previous comparisons because it requires a database of hidden Markov models (HMMs) provided by HH-suite. We additionally benchmarked HHblits on its curated SCOP95 dataset intersected with the SCOPe superfamily dataset and DHR still achieved highly accurate performance and faster computing speed on this dataset (Supplementary Fig. 8).

During homolog retrieval, it was shown that DHR sequence embeddings include massive structural information (Fig. 2a) and that DHR's accuracy on these retrieved homologs even surpasses that of structure-based alignment methods. This intriguing finding prompted us to explore further and revealed a correlation between DHR's sequence similarity rankings and structural similarity. Structural similarity between two distinct proteins is computed by DeepAlign[44], a protein structure analysis toolkit. DeepAlign superposes two protein structures and calculates the TM-score[45] according to a specified residue mapping. Studies based on the average across all SCOPe samples suggested that the TM-score decreases and r.m.s.d. increases monotonously along with DHR rank, determined by the embedding similarity between queries (Fig. 2g). A concrete example on query d1w0ha shows the reliability of DHR and the potential cause of failure of *k*-mer-based methods (PSI-BLAST and MMseqs2; Supplementary Fig. 1). Because PSI-BLAST and MMseqs2 did not yield any hits for d1w0ha, we looked at the top five retrieved homologs of DHR, which all belonged to the same family as d1w0ha in SCOPe. When we use PSI-BLAST to align the mapping, the structure similarity only yielded about 0.3 on these top five homologs compared to the structure alignment (Fig. 2h). As a result, the top five homologs that achieved higher scores were classified into the same family by DHR yet undetected by PSI-BLAST. This implies that DHR can capture more structural information (>0.5) than sequence-similarity-based methods (<0.5)[46], providing further evidence of its effectiveness in homolog retrieval.

Lastly, it is interesting to observe that DHR can offer structural information beyond the SCOPe families. We discovered two homologs scoring highly in DHR that mismatched with SCOPe labels, which were ranked third and fourth among the top five (Fig. 2i). When we look at the homologs retrieved from query d1w0ha, the TM-align score suggests that their similarity was strong enough to be classified into the same family and they were indeed clustered with the query in the 'δ-endotoxin C-terminal domain' family in the updated SCOP2 dataset[47] (Supplementary Table 2). This suggests a potential misclassification in SCOPe and that DHR-learned embeddings could contain supplementary and diverse structural information. We also performed a thorough scan on all 10,000 SCOPe families and calculated the TM-score of the top 20 target structures retrieved by DHR against the query target structures. The results showed several cases of inconsistent classification at the superfamily level between SCOPe and other structure classification datasets (Supplementary Table 2).

https://doi.org/10.1038/s41587-024-02353-6

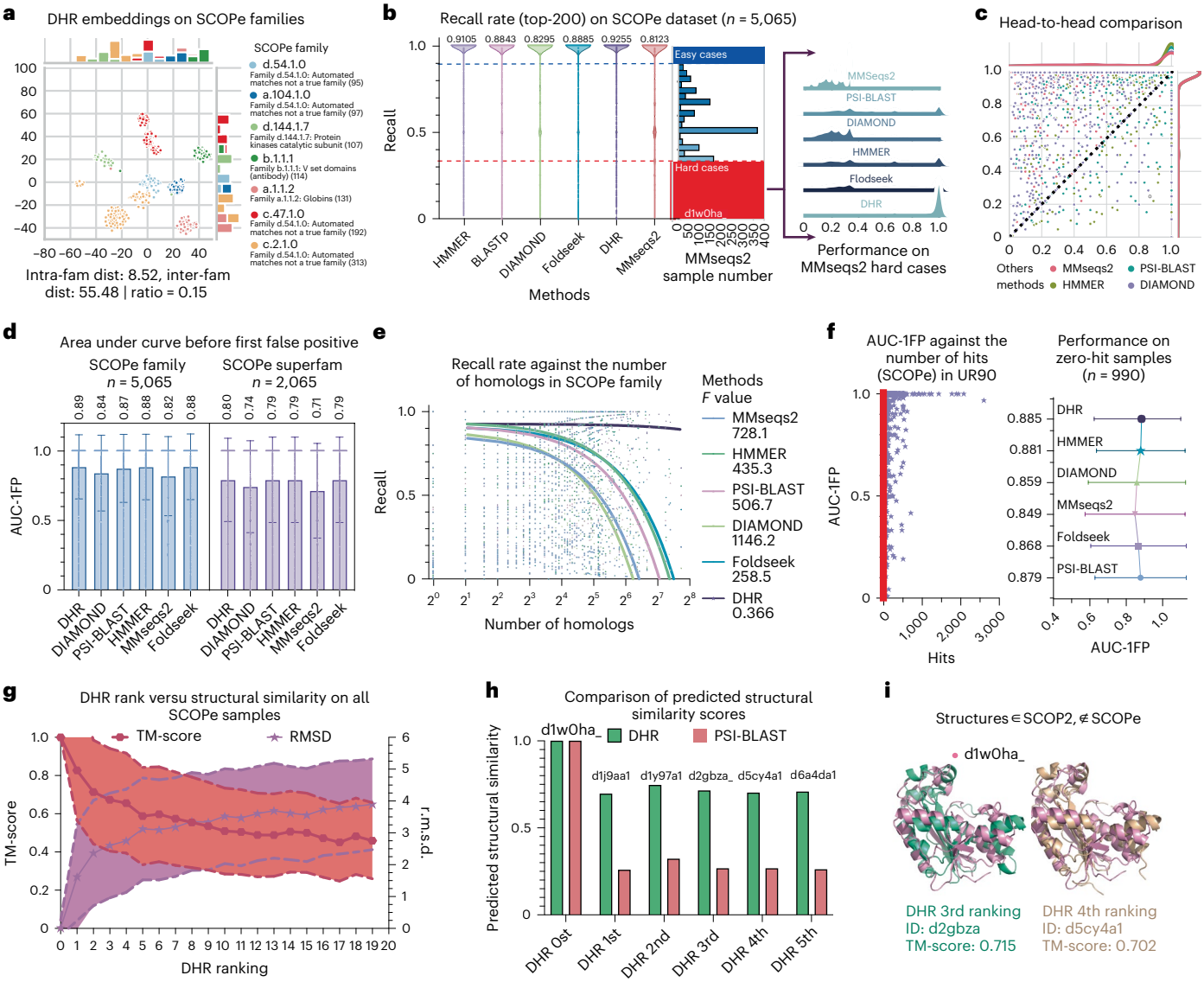


Fig. 2 | DHR outperforms previous methods on the SCOPe dataset regarding both speed and sensitivity. All methods used a single sequence as input unless especially mentioned. **a**, A *t*-distributed stochastic neighbor embedding visualization of the seven largest SCOPe family embeddings from DHR. **b**, Violin plot of recall (sensitivity) rate comparison of DHR to other methods on the SCOPe dataset of 5,065 queries. Right, the detailed performance of these methods within the hard cases (MMseqs2 recall rate below 0.34, colored in red). **c**, Head-to-head comparison of sensitivity between DHR and other methods. **d**, AUC-1FP metric comparison of different methods on the SCOPe family level (n = 5,065 queries) and the superfamily level (n = 2,065 queries). Data are presented as bar plots where the height of each bar represents the mean and error bars indicate the s.d. **e**, Regression plot of the recall rate against the SCOPe family size. **f**, Scatter plot of AUC-1FP against the number of sequence hits with the SCOPe dataset on UniRef90 (training set). Further visualization of the AUC-1FP of the zero-hit samples (n = 990) is provided as the mean value, with error bars indicating the s.d. **g**, The r.m.s.d. and TM-score relationship between the structure retrieved by DHR and the query structure according to the rank. We averaged the score on each rank with a central curve that connects the mean values of the data points, with two additional curves surrounding the central curve representing the s.d. A higher TM-score or lower r.m.s.d. indicates better structural similarity. **h**, Failure of *k*-mer-based methods on the c.55.3.5 family. The predicted structure similarity using DHR is much higher than that using *k*-mer-based methods, indicating that sequence-based methods using *k*-mers do not consider structural information. **i**, Two samples ranked highly by DHR for the c.55.3.5 family. SCOPe did not include them in the family but SCOP2 did. More potential problematic classification examples are shown in Supplementary Figs. 1–3 and Supplementary Table 2.

Rapid construction of rich and diverse MSAs

As a direct and critical downstream application of retrieving homologs, we could create an MSA with JackHMMER from homologs given by DHR and then compare it to the conventional AF2 default pipeline. Here, DHR served as a prefiltering method while JackHMMER used its multiple ungapped segment Viterbi (MSV) as a filtering method. This allowed us to assess both the quantity and the quality of MSAs constructed by DHR + JackHMMER in comparison to AF2 default (vanilla JackHMMER). Here, all MSA constructions were conducted using the same UniRef90 dataset. To accurately evaluate our performance, we also took into account different DHR or JackHMMER configurations, including DHR top {100k, 200k, 400k, 1M} and JackHMMER iteration {1, 2, 3}. We also included a DHR version without contrastive learning, 'DHR-w/o con', to demonstrate the necessity of our learning module. As JackHMMER's search time is mainly dedicated to scanning and aligning, by leveraging the alignment-free nature of DHR, this process can be greatly expedited for a remarkable increase in overall efficiency. All configurations of DHR + JackHMMER ran faster than vanilla JackHMMER, AF2

https://doi.org/10.1038/s41587-024-02353-6

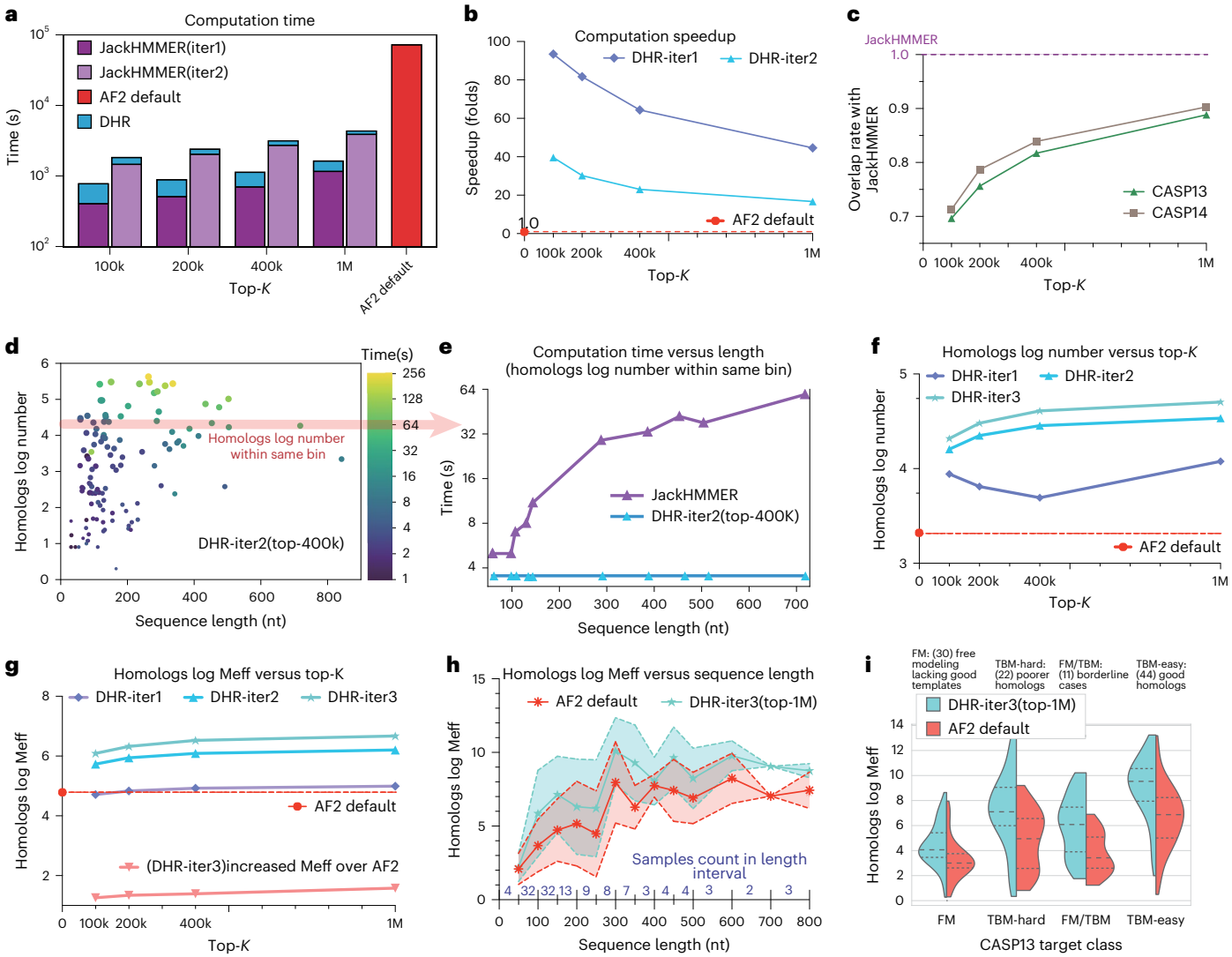


Fig. 3 | Performance of DHR on MSA construction and analysis of speed and quality on CASP13 dataset with n = 106 queries. a, Time expense at each step. Note that the *y* axis is set to a logarithmic scale. **b**, Overall improvement in speed of our method under different sensitivity settings compared to JackHMMER. The *x* axis controls the number of sequences retrieved by DHR and different colors denote the searching iterations using the subsequent JackHMMER. In the fastest setting, we achieved a 93-fold acceleration. **c**, Overlap ratio between our retrieved sequences and JackHMMER results. We conducted experiments on the CASP13DM and CASP14DM datasets with the highest overlap ratio above 90%, clearly demonstrating that our method builds MSAs rather similar to JackHMMER. **d**. Time consumption of searching two iterations with JackHMMER on CASP13DM samples relative to query sequence length and final homolog number. **e**, Selected samples (red arrow in **d**) with APPROXIMATELY 10,000 homologs in the MSA showing the effect of sequence length on searching time consumption. **f**, Logarithmic number of sequences in MSAs using DHR under different settings. The default MSAs built by the AF2 pipeline served as the baseline. **g**, The log Meff in the MSAs produced by DHR under different settings. **h**, The log Meff comparison for different sequence lengths using a central curve that connects the mean values of the data points, with two additional curves surrounding the central curve representing the s.d. **i**, Comparison of the Meff values of DHR and default AF2 in different CASP13 classes. Samples included queries with a good template (right) or with a poor or no template (left).

default, on average (Fig. 3a). Our projections on UniRef90 at a 70M rate brought a 93-fold acceleration in the lightest configuration (DHR-iter1 (100k)) on average (Fig. 3b). We anticipate even greater improvements if our proposed strategy is applied to broader sequence pools.

Surprisingly, despite being trained on a notably smaller dataset and using HHblits, DHR demonstrated an approximately 80% overlap with vanilla JackHMMER in constructing MSAs on UniRef90. This suggests that many MSA-related downstream tasks can be performed using DHR, yielding similar outcomes but with greater speed. We conducted this study on two test datasets, CASP13 and CASP14 domain sequences[48]. To ensure accuracy and reliability, our method only considers sequences with at least ten matching MSA homologs as valid and eliminates the remaining sequences. When using the top-1M configuration, DHR found around 90% of the same sequences as AF2 default JackHMMER at a greater speed. On the other hand, without contrastive learning and relying solely on ESM protein embedding, 'DHR-w/o con' achieved an overlap rate of only ~35% (Fig. 3c). As the quality of the MSAs is not directly associated with its overlap rate with JackHMMER, we continued to evaluate the quality of its MSAs using subsequent structure prediction applications. We discovered that DHR, when configured with 1M configurations and having the highest overlap rate, not only aligned closely with JackHMMER but also produced a variety of distinct MSAs, yielding the best outcomes (Fig. 6).

Not only did DHR obtain impressive computational acceleration and considerable overlapping rates but it was also capable of delivering increased homologs and MSAs for a query when compared to AF2 default (Fig. 3f). Because more and diverse MSAs could lead to a potential improvement of protein structure prediction[1], such an

increase is desired in MSA construction. Depending on the configuration, various levels of growth can be achieved. Among them, DHR-iter (1M) would experience the highest increase and a larger DHR top-*K* with more JackHMMER iterations would build more homologs. Furthermore, acquiring coevolutionary data from shorter sequences is more challenging because of fewer homologs in the database, which leads to a lower number of homologs collected with DHR (Fig. 3d). Nevertheless, another benefit of DHR is that it constructs the same number of homologs with varying lengths in a constant time, whereas JackHMMER scales linearly (Fig. 3e). Given that homologs or MSA may include redundant information, we used Meff introduced by Morcos et al.[49] to further provide preliminary data where we believed that more nonredundant MSAs may lead to better quality and diversity. Our research indicated that DHR-iter1 and AF2 default produced a similar Meff, while the highest Meff was yielded by DHR-iter3, superior to the performance of AF2 default by a margin up to 1.5-fold (Fig. 3g). While fewer homologs were retrieved for shorter sequences, a similar trend could be also observed in MSA log Meff; for sequences under 50 amino acids, their computed MSA log Meff values were below 2, while the maximum log Meff was achieved on sequences between 250 and 300 amino acids (Fig. 3h). We also assessed DHR using several hand-crafted categories from CASP13 and CASP14 (Fig. 3i). The results were unequivocal; all categories experienced substantial benefits from using DHR, with notably greater increases in TBM-hard cases (hard-to-find from template-based modeling) than in other areas. This reinforces our conclusion that DHR is a promising approach for constructing MSAs across all categories.

DHR-constructed MSA boosts AF2-based structure prediction

We previously revealed some statistical results of MSA log Meff to evaluate the quality of our MSAs built on retrieved homologs (Fig. 3g). Nevertheless, these evaluations mainly address diversity rather than quality. While there is no established standard for directly measuring MSA quality, we can indirectly estimate it by examining associated downstream applications, such as protein structure prediction. Here, we compared our results to those obtained by vanilla JackHMMER for predicting 3D structures using AF2 on the CASP13DM dataset. Consistent CASP target dataset results are detailed in Supplementary Table 6. We skipped the template searching phase and solely used UniRef90 to construct MSAs to prevent the feature stacking oriented from many databases and other potential impacts. An input query was given to obtain the top-*K* sequences (100k–1M) filtered by DHR and MSAs were constructed on these top-*K* sequences. To evaluate the accuracy of AF2's prediction, five models were produced and only that with the highest predicted local distance difference test (lDDT), which denotes confidence ranging from 0 to 100, was taken into consideration. To further assess their quality, we calculated the TM-score and global lDDT score[50] between native structures and modeled the results to measure how well our retrieved MSAs could aid in reconstructing a 3D structure. Then, we carried out a thorough analysis comparing all DHR setups to AF2 default. Standalone DHR performed similarly to AF2 default (TM-score ±0.02) and top-1M usually performed the best among all iterations (Fig. 4a,b and Supplementary Tables 5 and 7). While DHR produced a diverse MSA, we investigated whether it could be a supplement to AF2 default MSA and further boost the prediction results. We discovered that the merged MSA resulted in a performance boost (TM-score ±0.03) for DHR-merged-iter2 (top-400k), which merges MSAs from AF2 default with DHR-iter2 (top-400k). Moreover, the best performance was achieved by merging all MSAs from different DHR settings with AF2 default (DHR-merged-meta; Fig. 4a,b). These results suggest that, when used alone, DHR can be a fast and accurate replacement for the AF2 default MSA pipeline. When used in combination with the AF2 default MSA pipeline, DHR's improvement over vanilla JackHMMER on MSA log Meff offers AF2 default with diverse and useful 'structural' information that aids in the structure reconstruction process.

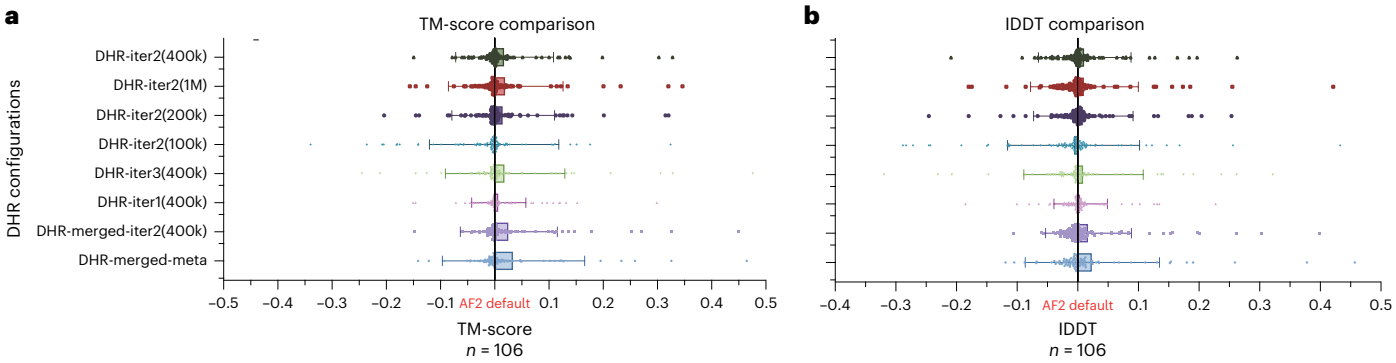
In addition, we analyzed the performance of standalone DHR and DHR-meta in comparison to AF2 default on several types of CASP13DM structures. Because all models displayed nearly perfect accuracy (<3-Å r.m.s.d.) on the TBM-easy class, we investigated the two remaining challenging classes. T1022s1-D2 demonstrated that DHR-meta is superior to DHR-iter2 (400k) and AF2 default by 5-Å and 14-Å r.m.s.d., respectively (Fig. 4d). As a consequence, DHR-meta showed the highest MSA log Meff; such an accurate prediction may have been because of its diversified MSA (Fig. 4e). This conclusion may not apply to all protein cases but it is generally supported by average statistical evidence. This investigation was undertaken by comparing the average MSA log Meff to the average TM-score generated by various DHR configurations and merged models for all CASP13 targets (Fig. 4g). The regression line indicated a monotonously increasing trend between these two variables. We also analyzed a potential failure case T0990-D2, in which DHR-meta folded a helix in the incorrect position, resulting in an 11-Å increase in r.m.s.d. compared to AF2 default (Fig. 4f). After careful examination, we observed that the DHR-retrieved MSA was not necessarily the reason for this result. We tested an MSA-encoding pipeline with an identical MSA and found notably better performance, which we then dubbed DHR-distill. DHR-distill achieved 0.03-Å lower r.m.s.d. and 0.01 higher TM-score compared to AF2 default, which we did not investigate in detail. We believe that the AF2-like and JackHMMER black-box effect was the cause of this difference and further research could be conducted to gain a better understanding of this observation. Our trials revealed that, when MSA is combined with customized protein structure prediction techniques, it can boost accuracy in models. This approach allows us to quickly identify MSAs and use them for precise protein construct prediction from sequences.

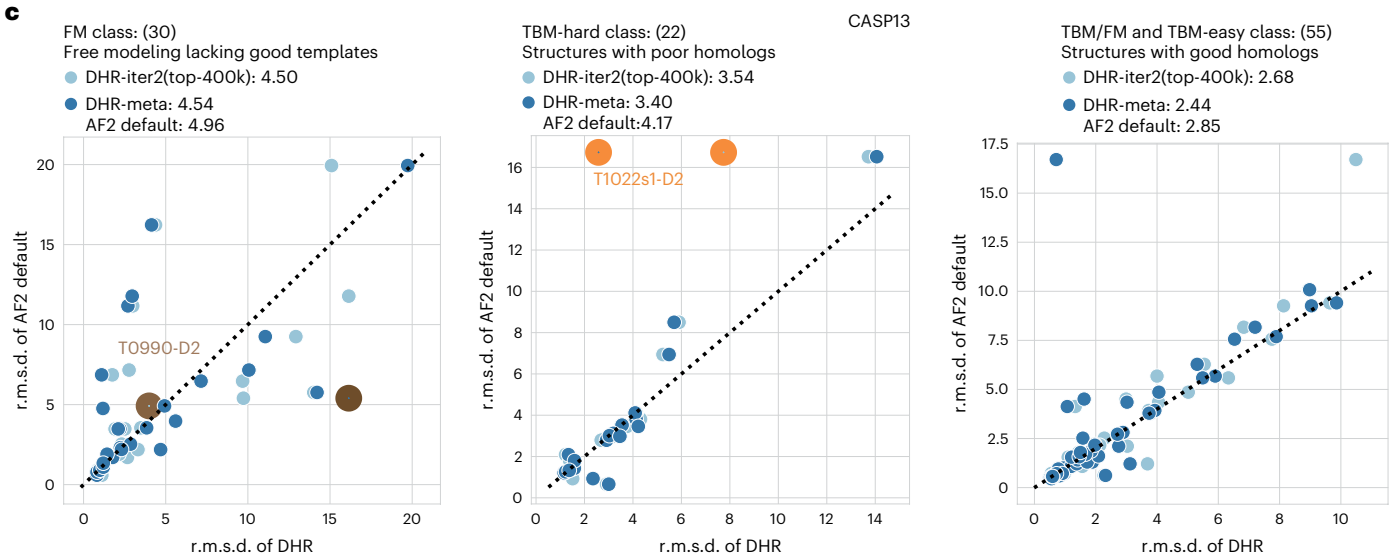
MSA is an essential component for accurate protein structure prediction

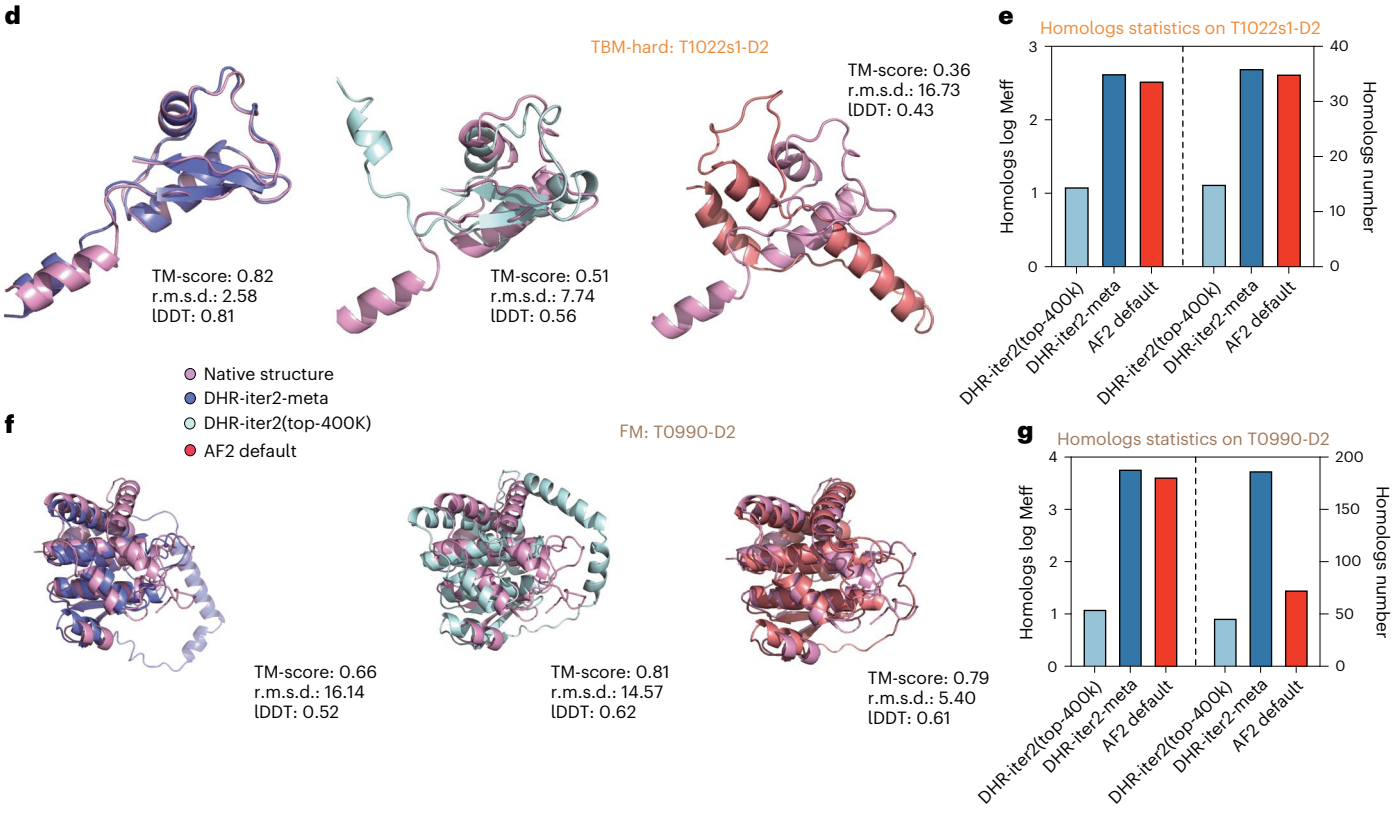
Recent advancements, such as RGN2 (ref. 36), ESMFold[25] and OmegaFold[32], use language models to bypass the process of producing MSAs. By doing so, they claim that MSAs are not necessarily needed for protein structures to be predicted successfully. These approaches also greatly reduce the processing time by eliminating the step of manually constructing MSAs. To examine the potential benefits that these language models could bring to protein structure prediction, we evaluated whether substituting MSA construction with language models on all CASP14DM targets would generate better results. To that end, we chose ESMFold and OmegaFold as our primary validation methods because they are open source with no data leakage related to any of the CASP datasets. AF2 Single-Seq was also included to conduct a model-based

Fig. 4 | Comparison of 3D structural modeling precision for each protein between the AF2 default MSAs and MSAs obtained from our method on CASP13DM with $n = 106$ queries. a,**b**, Bar plots where the height of each bar represents the mean and error bars indicate the s.d. Plots show the all-atom TM-score (**a**) and lDDT (**b**) evaluation of different DHR configurations through their downstream structure prediction result. The default AF2 pipeline is set as a baseline to show the gain in performance. **c**, The DHR-produced MSA in structure prediction compared to AF2 default MSA. We show three head-to-head comparisons of different classes using the r.m.s.d. as an evaluation metric. **d**, Case study on the outlier case T1022s1-D2 identified as good by DHR. We show that the DHR-built MSA can produce a better structure than the default and that, by merging them, we can get even better predictions. **e**., The log Meff of the three MSAs in **d**. A large number of sequences may lack a quality guarantee. **f**, Case study on the outlier case T0990-D2 identified as bad by DHR. The merged result has a chain lying in the wrong position compared to the default prediction, leading to overall failure. We distilled each MSA and merged them again to get the correct prediction. **g**, The overall relationship between Meff and TM-score. Each blue point corresponds to one configuration of DHR.

https://doi.org/10.1038/s41587-024-02353-6







https://doi.org/10.1038/s41587-024-02353-6

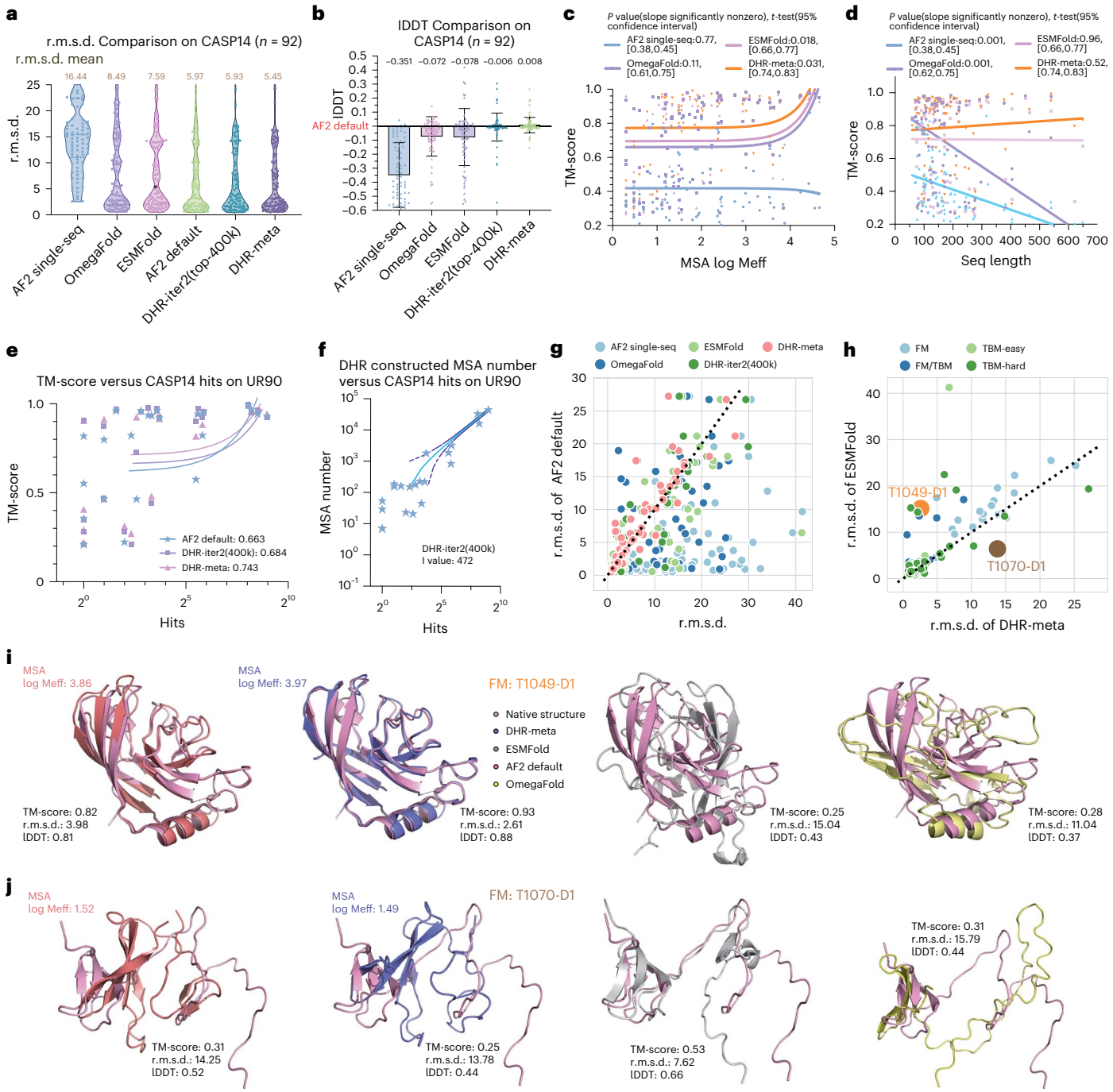


Fig. 5 | Comparison between MSA-based methods and MSA-free methods on CASP14DM with $n = 92$ queries. a, Violin plot of r.m.s.d. comparison using different methods. **b**, Bar plot of lDDT score comparison, setting the AF2 result on CASP14 as the baseline, where the height of each bar represents the mean improvement of each method over AF2 default and error bars indicate the s.d. **c**, Relationship between MSA log Meff and TM-score on the predicted structures. **d**, Relationship between query sequence length and TM-score on the predicted structure. **e**, Relationship between the CASP14 sequence hits on UniRef90 (training data) and TM-score. **f**, Regression plot of the relationship between the CASP14 sequence hits on UniRef90 (training data) and DHR-constructed MSA log Meff, where the central curve shows the fitted mode and the two additional curves surrounding the central curve indicate the 95% intervals. **g**, Head-to-head comparison between AF2 default and other methods on CASP14 using r.m.s.d. as the evaluation metric. Each dot represents a sample, with the *x* axis showing the custom method result and the *y* axis showing the AF2 default distance. Dots above the reference line indicate that the method performed better than AF2 default. **h**, Head-to-head comparison between AF2 default and our method with merged MSAs on different classes using r.m.s.d. as the metric. Dots above the reference line indicate that our method achieved higher precision. Two outliers are shown in **g**,**h**. **i**, Case study of an outlier case T1049-D1 identified as good by DHR. **j**. Case study of an outlier case T1070-D1 identified as bad by DHR.

ablation study. They were compared to MSA-guided AF2 models, such as AF2 default, AF2-DHR-iter2 (400k) and DHR-meta. As anticipated, we achieved an r.m.s.d. improvement of 2.1 Å for DHR-meta over ESMFold and OmegaFold, where ESMFold performed slightly better. Moreover, the superior performance of DHR-meta did not arise from the AF2 default-constructed MSA as AF2-DHR-iter2 (400k) achieved a 0.03-Å lower r.m.s.d. compared to AF2 default (Fig. 5a,b and Supplementary Tables 6, 8 and 9).

Furthermore, we investigated each case in CASP14DM to determine the causes for such disparity. In the more challenging cases where only a few MSAs were detectable, DHR-meta was observed to surpass language model-based methods (Fig. 5g), indicating that MSAs created by DHR can provide more critical structural information for 3D modeling. On simple cases with a large number of available MSAs, a language model could convey as much information as MSAs, resulting in the same performance (Fig. 5c). Examining yet another element impacting the reconstruction standard (sequence length), DHR-meta benefited from a slight TM-score boost as the sequence size increased. On the other hand, ESMFold remained stable while OmegaFold suffered from dramatic declines (Fig. 5d). We also noticed that algorithms such as OmegaFold exhaust the 32-GB memory of V100 GPUs, preventing them from completing the prediction pipeline for sequences longer than 650 amino acids, whereas our algorithm processed the sequence accurately. This indicates that a DHR-constructed MSA is indeed essential for high-resolution models, especially in cases with limited training data or long sequences.

To strengthen our claim, we conducted similarity tests using BLAST to search CASP14 targets against UniRef90 (our search database) and analyzed the number of hits. We observed that, on average, DHR achieved a higher TM-score than AF2 default and all methods showed increasing performance with more hits in the search database (Fig. 5e). For CASP14 samples that either had no hits or very few hits against UniRef90, DHR still managed to generate effective MSAs that enhanced structure prediction (Fig. 5e,f and Supplementary Tables 10 and 11). This resulted in an approximate increase of 0.01 in TM-score for CASP14 samples with zero hits. Furthermore, a detailed head-to-head comparison between DHR-meta and the most efficient single-sequence approach, ESMFold, was conducted for all CASP14 targets. Our analysis showed that DHR-meta outperformed ESMFold in nearly every case, especially in the most challenging category (free modeling (FM); no hit targets) (Fig. 5h). In particular, we demonstrate a specific sample where DHR-meta could identify more MSAs for such a challenging class, thus resulting in a more accurate prediction (Fig. 5i and Supplementary Table 11). Despite the few exceptions where DHR-meta lagged behind ESMFold, our study revealed that, in cases without any MSA, DHR-meta and AF2 default would underperform to the level of AF2 Single-Seq (Fig. 5j). Overall, our results demonstrated that MSA-based models can greatly improve prediction accuracy and efficacy when compared to language model-based approaches. We show that the use of MSAs is beneficial and should not be overlooked in protein structure prediction pipelines. Furthermore, we highlight the greater performance potential of our method in challenging cases with limited MSA data.

Scalability in large datasets and high-quality MSA generation

DHR proved its efficacy and ability in assisting structural prediction using the million-scale dataset UniRef90. We further present here that DHR is scalable to other datasets that are 100 times larger than UniRef90 and can still generate high-quality homologs or MSAs. Here, we performed our studies on BFM/MGnify, a combination of BFD and MGnify (2019_05), and included hundreds of millions of sequences in line with another standardized pipeline, ColabFold, which is a reproduction of AF2 using MMseqs2 to generate MSAs on BFM/MGnify by default. Specifically, we assessed the quality of DHR-generated MSAs through ColabFold on protein structure prediction against its default setting using CASP15 dataset. It is important to note that we did not retrain DHR on the large-scale dataset. Instead, the UniRef90-trained version was directly applied for MSA search inference. In contrast, within ColabFold, MMseqs2-BFD/MGnify was established with a profile database of BFD/MGnify, which is expected to offer natural advantages over DHR.

In evaluating the 50 CASP15 targets made available up to October 31, 2023 in terms of TM-score, DHR-ColabFold demonstrated a performance comparable to its default setting, MMseqs2, with a marginal improvement of approximately 0.01 (Fig. 6a, Supplementary Fig. 8 and Supplementary Table 9). Additionally, for most of the targets (36 of 50), DHR-ColabFold slightly outperformed ColabFold-MMseqs2. Despite this subtle enhancement, a review of the distinct MSAs generated by both methods (overlapping rate = 0.72) revealed varying rankings for each target. The observed variance suggests that DHR and existing homolog detection methodologies may serve similar roles. Notably, we included ESMFold, featured on the CASP15 leaderboard, to underscore the assertion that MSA is an indispensable component for precise structure prediction (Fig. 6a and Supplementary Table 12). While ESMFold showed an enhancement relative to its previous open version, it still lagged behind the MSA-based ColabFold (AF2) by approximately 0.09 in TM-score.

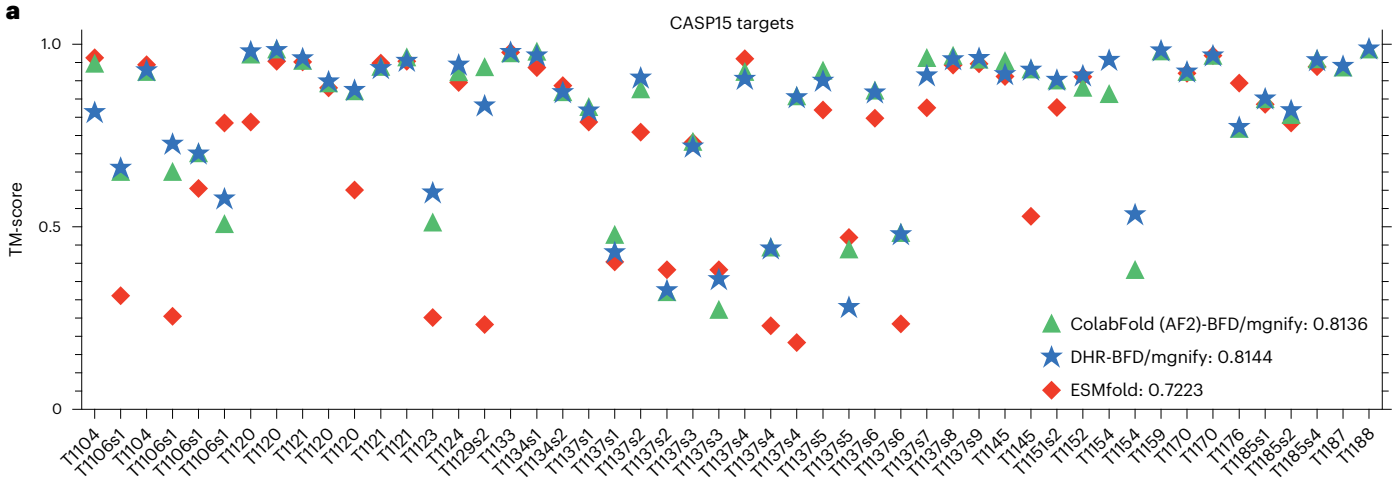
Predicting the structure of FM targets, especially those without structural templates in the Protein Data Bank (PDB) or with only limited MSAs, poses a great challenge in the field of structure prediction. In these complex scenarios, DHR stands out by generating meaningful MSAs, which improve structural predictions. DHR exceeded the performance of ColabFold, which uses MSAs constructed by MMseqs2, by 0.007 in terms of TM-score (Fig. 6b). Across all targets, DHR showed a slight increase in performance, evidenced by approximately 1.2-fold higher Meff compared to MMseqs2, translating to improved accuracy with a reduction of 0.15 Å in r.m.s.d. (Fig. 6c). Notably, for target T1129s2, DHR provided a 1.1 increase in Meff and an enhancement of 1.5 Å relative to ColabFold-MMseqs2 (Fig. 6c). Following the similarity tests in CASP14 and SCOPe, we also emphasize that the proficiency of DHR is not a consequence of simply memorizing query or hit pairs from its training data. We conducted a comprehensive similarity assessment of all targets against our training dataset with BLAST (Fig. 6d). Although it is anticipated that targets with greater similarity (larger hits) would enhance structure prediction, it is noteworthy that DHR's performance remained relatively consistent even for the least similar targets (with zero hits), with an average r.m.s.d. of 4.8 Å (Fig. 6d).

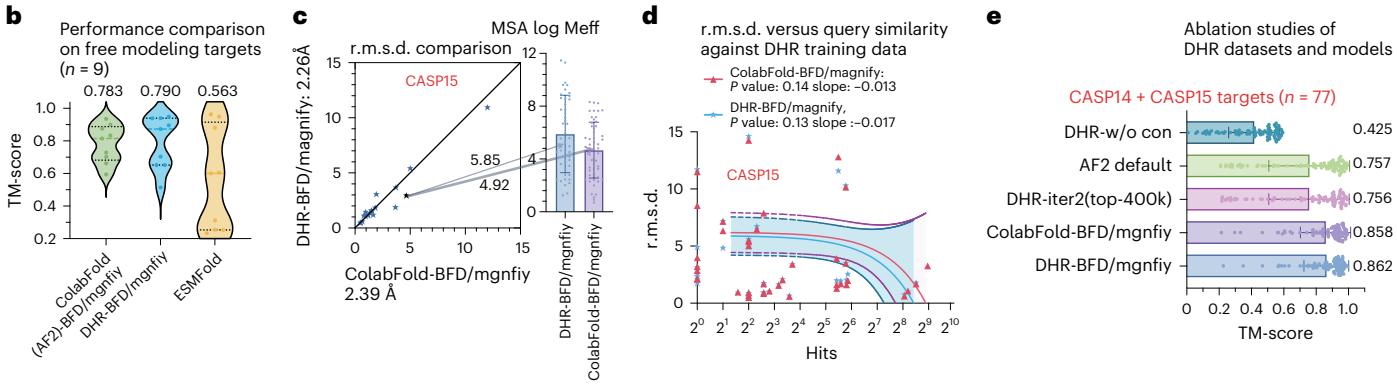
While DHR scaled well on the large-scale BFD/MGnify datasets, we further show that DHR is 'interpretable', as a larger dataset improved its

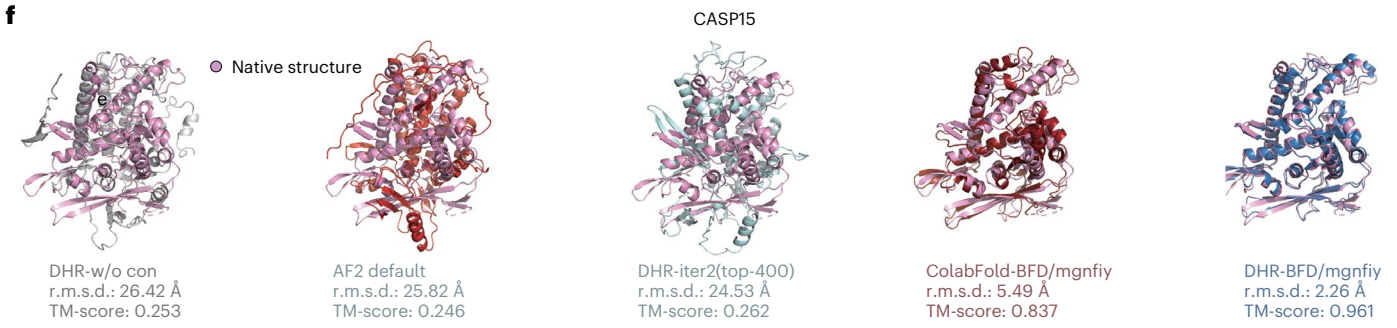
Fig. 6 | Scaling up DHR and assessing it on CASP15 with $n = 50$ queries. **a**, Comparative analysis of TM-score performance across 50 targets in CASP15. We used DHR-iter2 (top-400k) as the primary procedure for the MSA search, followed by ColabFold for downstream protein structure prediction. In contrast, ColabFold features its default pipeline using MMseqs2 for MSA search and ColabFold for structure prediction. Additionally, we included the results of ESMFold, an MSA-independent method based on single-sequence information, as extracted from the official results. **b**, Performance on nine FM targets in CASP15. **c**, Head-to-head r.m.s.d. comparison between ColabFold-MMseqs2 and ColabFold-DHR and their MSA log Meff results (represented as bar plots, where the height of each bar represents the mean and error bars indicate the s.d.). **d**, Regression plot of relationship between BLAST hits of a query (in CASP15) on our training set, UniRef90, against prediction r.m.s.d. values. More hits suggest a more similar query to our training set. The plots are presented as a central curve showing the fitted mode, with two additional curves surrounding the central curve showing the 95% intervals. **e**, Ablation studies of different DHR datasets and models on CASP14 + CASP15 ($n = 77$) targets, where DHR-iter2 (top-400k) was used. The results are presented as bar plots where the height of each bar represents the mean and error bars indicate the s.d. **f**. Visualization of a case study on CASP15 targets. This particular case was chosen on the basis of the notable increase in Meff observed when transitioning from the use of DHR-UniRef90 to DHR-BFD/MGnify. **g**–**i**, The results on our collected NewProtein dataset, which consists of 650 targets: head-to-head comparison of ColabFold-MMseqs2, ColabFold-DHR and DHR-MMseqs2 (merged) (**g**); MSA log Meff comparison (**h**); regression plot of the relationship between TM-score performance and MSA log Meff (**i**). The plots are presented as a central curve showing the fitted mode, with two additional curves surrounding the central curve showing the 95% intervals.

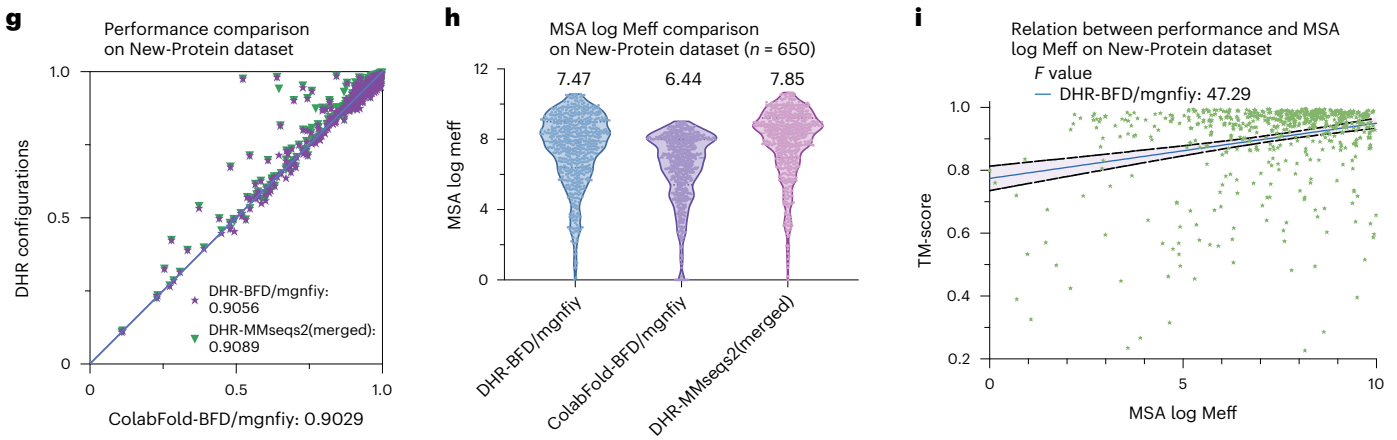
https://doi.org/10.1038/s41587-024-02353-6

MSA quality, indicating that contrastive training is a necessary component within DHR (Fig. 6e). When benchmarked against different data scales, DHR combined with ColabFold (AF2) offered performance on par with (subtle improvement) JackHMMER or MMseqs2. The TM-score disparity across the two datasets reached up to 0.12 points, implying DHR's scalability potential upon a subsequent increase in size of our training set. Notably, a DHR variant without contrastive training and relying solely on embedding similarity for homolog extraction had









the most inferior performance (Fig. 6e). This underscores the importance of our designed learning mechanisms for optimal performance. Beyond statistical results, the interesting case of T1037-D1 actually showed that the variations between datasets and DHR models were sometimes large (Fig. 6f). When leveraging the UniRef90 dataset, DHR alongside AF2 default exhibited diminished efficacy, equating to MSAs constructed solely using protein embeddings, labeled 'DHR-w/o con'. However, upon transitioning to the BFD/MGnify database, both DHR and MMseqs2 yielded notably higher MSA Meff scores, culminating in a 5.49-Å r.m.s.d. for ColabFold-MMseqs2 and an even lower r.m.s.d. for DHR accompanied by an increased extracted Meff (Fig. 6f and Supplementary Table 11).

While the CASP15 targets served as a blind test for DHR, the dataset comprised only 50 structures, which is relatively limited for extensive analysis. To address this, we introduced a new benchmark dataset, the New-Protein dataset, assembled from recently disclosed PDB structures with a cutoff date of January 1, 2022 to January 1, 2023, as well as a 70% similarity filter, ensuring that the 650 selected proteins were absent from our search and training set. On these targets, DHR was also able to generate more comprehensive and meaningful MSAs, achieving a log Meff approximately 1.0 higher than those created by MMseqs2 (Fig. 6h). Additionally, the integration of MSAs generated by DHR with those from JackHMMER in AF2 demonstrated similar effects; this integration approach was also successfully applied between DHR and MMseqs2, where their combined MSAs produced the highest log Meff and prediction accuracy across our dataset of 650 new protein targets, achieving a TM-score of 0.909 (Fig. 6g and Supplementary Table 13). Our statistical analysis on the New-Protein dataset confirmed that an increased MSA log Meff is indicative of better prediction performance (Fig. 6i). The above experiments demonstrated DHR's scalability in constructing high-quality MSAs on large-scale datasets for normal proteins (CASP targets and PDB data), which include resolved protein structures. Additionally, we further investigated the challenges of disordered proteins, which are involved in various essential biological processes[51]. DHR exhibited robust capabilities by constructing MSAs for disordered proteins on a large-scale search database with high diversity (Supplementary Fig. 9).

Discussion and Conclusions

Here, we presented DHR, a framework for efficient and accurate homolog detection. We used a contrastive learning strategy to train dual encoders to create fixed-dimensional embeddings. This method of embedding proteins allowed us to accurately and quickly identify related sequences by taking into account long-range dependency information across various protein families. Through the use of dot products, we were able to retrieve pertinent relationships with high precision—a capability that surpassed existing methods in terms of speed, with an over 10% increase in sensitivity and AUC-1FP rate and a 28,700-fold acceleration compared to profile-based methods. In addition, the homologs we obtained could improve efficiency and performance for downstream tasks. Specifically, MSAs constructed purely by DHR enabled AF2/ColabFold to match the performance of their default settings. Furthermore, integrating DHR with other MSA pipelines enhanced the performance, resulting in an approximate improvement of 0.4 Å on CASP14 over the standard AF2/ColabFold pipeline.

Despite its merits, DHR encountered challenges with sequences exceeding 1,000 nt, leading to a decline in the quality of retrieved homologs. This could be attributed to the constraints of the ESM embedding. Moreover, while DHR aims to replicate the behavior of JackHMMER using contrastive learning in a teacher–student paradigm, merely emulating JackHMMER might fall short in capturing profound biological insights. For future endeavors, limitations associated with longer sequences will be addressed. We also aim to refine protein embedding dimensions, balancing efficacy and memory optimization. Furthermore, a broader array of methods could be integrated into our learning framework.

Ultimately, DHR provides a solution to an important computational challenge in the area of computational biology and bioinformatics. It has the potential to unlock a wide range of applications in protein analysis, such as homology detection and sequence alignment.

Online content

Any methods, additional references, Nature Portfolio reporting summaries, source data, extended data, supplementary information, acknowledgements, peer review information; details of author contributions and competing interests; and statements of data and code availability are available at https://doi.org/10.1038/s41587-024-02353-6.

References

1. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
2. Helbert, W. et al. Discovery of novel carbohydrate-active enzymes through the rational exploration of the protein sequences space. *Proc. Natl Acad. Sci. USA* **116**, 6063–6068 (2019).
3. Penny, D., Foulds, L. R. & Hendy, M. D. Testing the theory of evolution by comparing phylogenetic trees constructed from five different protein sequences. *Nature* **297**, 197–200 (1982).
4. Cao, Y., Geddes, T. A., Yang, J. Y. H. & Yang, P. Ensemble deep learning in bioinformatics. *Nat. Mach. Intell.* **2**, 500–508 (2020).
5. Mitchell, P. J. & Tjian, R. Transcriptional regulation in mammalian cells by sequence-specific DNA binding proteins. *Science* **245**, 371–378 (1989).
6. Wang, Q. et al. Mutant proteins as cancer-specific biomarkers. *Proc. Natl Acad. Sci. USA* **108**, 2444–2449 (2011).
7. Gillette, M. A. & Carr, S. A. Quantitative analysis of peptides and proteins in biomedicine by targeted mass spectrometry. *Nat. Methods* **10**, 28–34 (2013).
8. Altschul, S. F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997).
9. Baker, D. & Sali, A. Protein structure prediction and structural genomics. *Science* **294**, 93–96 (2001).
10. Yang, J. et al. The I-TASSER suite: protein structure and function prediction. *Nat. Methods* **12**, 7–8 (2015).
11. The 15th community-wide experiment on the critical assessment of techniques for protein structure prediction. Available from https://predictioncenter.org/casp15/zscores_final.cgi
12. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).
13. Eddy, S. R. Accelerated profile HMM searches. *PLoS Comput. Biol.* **7**, e1002195 (2011).
14. Weston, J., Elisseeff, A., Zhou, D., Leslie, C. S. & Noble, W. S. Protein ranking: from local to global structure in the protein similarity network. *Proc. Natl Acad. Sci. USA* **101**, 6559–6563 (2004).
15. Page, L., Brin, S., Motwani, R. & Winograd, T. The PageRank citation ranking: bringing order to the web. Stanford InfoLab (1999).
16. Pang, L. et al. DeepRank: a new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management* (eds Lim, E.P. & Winslett, M.) (Association for Computing Machinery, 2017).
17. Lesk, A. M. & Chothia, C. How different amino acid sequences determine similar protein structures: the structure and evolutionary dynamics of the globins. *J. Mol. Biol.* **136**, 225–270 (1980).
18. Chatzou, M. et al. Multiple sequence alignment modeling: methods and applications. *Brief. Bioinform.* **17**, 1009–1023 (2016).

19. Xia, X., Zhang, S., Su, Y. & Sun, Z. MICAlign: a sequence-to-structure alignment tool integrating multiple sources of information in conditional random fields. *Bioinformatics* **25**, 1433–1434 (2009).
20. Armougom, F. et al. Expresso: automatic incorporation of structural information in multiple sequence alignments using 3D-Coffee. *Nucleic Acids Res.* **34**, W604–W608 (2006).
21. Aniba, M. R., Poch, O. & Thompson, J. D. Issues in bioinformatics benchmarking: the case study of multiple sequence alignment. *Nucleic Acids Res.* **38**, 7353–7363 (2010).
22. Bepler, T. & Berger, B. Learning protein sequence embeddings using information from structure. In *International Conference on Learning Representations* https://cap.csail.mit.edu/sites/default/files/research-pdfs/Learning%20Protein%20Sequence%20Embeddings%20Using%20Information%20from%20Structure-%20Bonnie%20Berger.pdf (ICLR, 2019).
23. Rao, R., Meier, J., Sercu, T., Ovchinnikov, S. & Rives, A. Transformer protein language models are unsupervised structure learners. In *International Conference on Learning Representations* https://openreview.net/pdf?id=fylclEqgvgd (ICLR, 2021).
24. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc. Natl Acad. Sci. USA* **118**, e2016239118 (2021).
25. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
26. Alley, E. C., Khimulya, G., Biswas, S., AlQuraishi, M. & Church, G. M.Unified rational protein engineering with sequence-only deep representation learning. *Nat. Methods* **16**, 1315–1322 (2019).
27. Bileschi, M. L. et al. Using deep learning to annotate the protein universe. *Nat. Biotechnol.* **40**, 932–937 (2022).
28. Heinzinger, M. et al. Contrastive learning on protein embeddings enlightens midnight zone. *NAR Genom. Bioinform.* **4**, lqac043 (2022).
29. Hamamsy, T. et al. Protein remote homology detection and structural alignment using deep learning. *Nat. Biotechnol.* **42**, 975–985 (2024).
30. Camacho, C. et al. BLAST+: architecture and applications. *BMC Bioinformatics* **10**, 1–9 (2009).
31. Rao, R. et al. MSA transformer. *Proc. Mach. Learn. Res.* **139**, 8844–8856 (2021).
32. Wu, R. et al. High-resolution de novo structure prediction from primary sequence. Preprint at *bioRxiv* https://doi.org/10.1101/2022.07.21.500999 (2022).
33. UniProt ConsortiumUniProt: a worldwide hub of protein knowledge. *Nucleic Acids Res.* **47**, D506–D515 (2019).
34. Steinegger, M., Mirdita, M. & Söding, J. Protein-level assembly increases protein sequence recovery from metagenomic samples manyfold. *Nat. Methods* **16**, 603–606 (2019).
35. Steinegger, M. & Söding, J. Clustering huge protein sequence sets in linear time. *Nature Commun.* **9**, 1–8 (2018).
36. Chowdhury, R. et al. Single-sequence protein structure prediction using a language model and deep learning. *Nat. Biotechnol.* **40**, 1617–1623 (2022).
37. Chandonia, J.-M., Fox, N. K. & Brenner, S. E. SCOPe: manual curation and artifact removal in the structural classification of proteins—extended database. *J. Mol. Biol.* **429**, 348–355 (2017).
38. Chandonia, J.-M., Fox, N. K. & Brenner, S. E. SCOPe: classification of large macromolecular structures in the structural classification of proteins—extended database. *Nucleic Acids Res.* **47**, D475–D481 (2019).
39. Altschul, S. F. & Koonin, E. V. Iterated profile searches with PSI-BLAST—a tool for discovery in protein databases. *Trends Biochem. Sci.* **23**, 444–447 (1998).
40. Buchfink, B., Xie, C. & Huson, D. H. Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* **12**, 59–60 (2015).
41. Johnson, L. S., Eddy, S. R. & Portugaly, E. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics* **11**, 1–8 (2010).
42. van Kempen, M. et al. Foldseek: fast and accurate protein structure search. *Nat. Biotechnol.* **42**, 243–246 (2024).
43. Remmert, M., Biegert, A., Hauser, A. & Söding, J. HHblits: lightning-fast iterative protein sequence searching by HMM–HMM alignment. *Nat. Methods* **9**, 173–175 (2012).
44. Wang, S., Ma, J., Peng, J. & Xu, J. Protein structure alignment beyond spatial proximity. *Sci. Rep.* **3**, 1–7 (2013).
45. Zhang, Y. & Skolnick, J. Scoring function for automated assessment of protein structure template quality. *Proteins* **57**, 702–710 (2004).
46. Xu, J. & Zhang, Y. How significant is a protein structure similarity with TM-score = 0.5? *Bioinformatics* **26**, 889–895 (2010).
47. Andreeva, A., Howorth, D., Chothia, C., Kulesha, E. & Murzin, A. G. SCOP2 prototype: a new approach to protein structure mining. *Nucleic Acids Res.* **42**, D310–D314 (2014).
48. Kryshtafovych, A., Schwede, T., Topf, M., Fidelis, K. & Moult, J. Critical assessment of methods of protein structure prediction (CASP)—round XIII. *Proteins* **87**, 1011–1020 (2019).
49. Morcos, F. et al. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proc. Natl Acad. Sci. USA* **108**, E1293–E1301 (2011).
50. Mariani, V., Biasini, M., Barbato, A. & Schwede, T. lDDT: a local superposition-free score for comparing protein structures and models using distance difference tests. *Bioinformatics* **29**, 2722–2728 (2013).
51. Necci, M., Piovesan, D. & Tosatto, S. C. E. Critical assessment of protein intrinsic disorder prediction. *Nat. Methods* **18**, 472–481 (2021).

Methods

DHR

DHR is built around three key elements. Firstly, it leverages a large protein language model to create rich embeddings. Secondly, a bi-encoder architecture is used to project the embeddings of queries and candidates into distinct spaces. Thirdly, through contrastive learning, the model differentiates between nonhomologous pairs by distancing them, while bringing closer the sequences that are homologous.

Unsupervised protein language model. The recent success of large-scale language model pretraining techniques, such as BERT (bidirectional encoder representations from transformers)[52], has led to notable breakthroughs in a variety of natural language processing tasks[53,54], The process of using such a model consists of two stages: pretraining and fine-tuning. During pretraining, the language model is trained in an unsupervised manner on a large dataset. In the fine-tuning stage, the pretrained model is further trained on a smaller dataset with human-labeled data, which can be used for tasks such as dialog response generation[55] and sentiment analysis[56–59]. Pretrained models generally perform better on downstream tasks than models that are randomly initialized, as they have already learned about the underlying data distribution before fine-tuning.

Pretraining on large protein sequence datasets followed by fine-tuning for downstream tasks related to protein structure or function has emerged as a promising strategy. Rao et al. introduced a transformer-based language model, ESM-1b[23,24], which demonstrated that information learned from protein sequences alone can greatly benefit various downstream tasks, outperforming LSTM-based models[26,60] by a large margin. Rao et al. also found that integrating MSAs into the model, referred to as the MSA transformer, led to state-of-the-art results on multiple structure-related benchmarks. The same language model objective was also applied in AF2[1], further improving the accuracy of structure prediction.

Dense retrieval, a method that uses a pretrained language model to efficiently retrieve documents with a relatively high recall rate, has gained recent attention in the text domain. In this approach, the query or document is transformed into a low-dimensional vector space by averaging the embeddings of each word, preserving its semantic meaning. The bi-encoder architecture, which includes a query encoder and a document encoder, maps both the query and the documents into the same low-dimensional space in the language processing area[56]. The similarity score between a query and a document is then defined as the dot product of the encoded vectors. Documents with high similarity scores to the query are retrieved. ANCE[57] suggests that using hard negatives can improve retrieval performance.

Pretraining. In the training stage, the model is initialized with parameters from ESM and iterates for ten epochs. ESM is based on the transformer and each outputs a series of vectors matching the number of tokens in the input sequence. We select the first vector as the fixed-dimension representation of the whole sequence. We train DHR with eight Nvidia V100 32-GB GPUs. Each GPU hosts 136 paired data in one training batch. We also implemented a cross-batch negative strategy to extend the batch size to 544. To build the training dataset for the proposed method, we first used the query sequence (denoted as q) to search from UniClust30 using HHblits[43] and build the MSA as the ground truth. The resulting MSA is denoted as $M^q_{(n+1)\times l} = [q, C^q] = \left[q, c^q_1, c^q_2, \ldots, c^q_n\right]$, where l is the sequence length, the first sequence q is the query and the remaining n sequences $C^q = c^q_{1:n}$ are searched homologs (1:n denotes set {1, 2, …, n}). For simplicity, we drop the superscript q if the context is clear. Because q and $c_{1:n}$ are from the same MSA, (q, c) are treated as a positive pair for any $c \in \{c_1, \ldots, c_n\}$. Note that the amount of training data could be enormous considering that the data construction process only depends on JackHMMER to build the MSA.

Bi-encoder model and contrastive learning. We denote the transformer-based query sequence and candidate encoder as f_Θ and f_Ψ, respectively, where Θ and Ψ are learnable model parameters. Given paired data, say (q, c) from the previous section, the query encoder f_Θ maps q into a d-dimensional vector, $f_\Theta(q) = h_q \in R^d$, and the candidate encoder maps c into a vector with same dimension, that is, $f_\Psi(c) = z_c \in R^d$. In fact, when using the transformer-based encoder, targets with a series of n residuals are embedded into a series of vectors with the dimension of $(n+1) \times 768$. Then, we fetch the vector embedding with dimension 1×768 as the sequence representation. We rely on the training procedure to learn the sequence embedding, reducing all information from the long sequence to a fixed-length vector. Finally, the inner product between the embeddings

$$s_{qc} = h_q^T \times z_c$$

is computed as the similarity score between query q and candidate c. In addition, both the query encoder and candidate encoder are initialized with ESM-1b because the model is already well pretrained with billions of protein sequences.

Self-supervised representation learning has made notable leaps fueled by progress in contrastive learning, which seeks to learn transformations that embed positive input pairs nearby while pushing negative pairs far apart[61–64]. Regarding our contrastive learning strategy, we further use in-batch data as negative examples. Compared to the actual sampling of negatives, this can greatly improve the training efficiency. More specifically, for a batch of paired data $(q_1, c_1), \ldots, (q_b, c_b)$, where b is the batch size, the negative samples for query q_i are all other candidates within the same batch, that is, all c_j where $i \neq j$. We demonstrate this in Fig. 1 with more detail. We get query–context pairs when a random homolog is drawn from the MSA with a query. The examples in diagonal lines are all positive examples and the others are negative examples. Suppose that the query and candidate encoder map these queries and their corresponding candidates into matrices $[h_1, h_2, \ldots, h_b] = [f_\Theta(q_1), f_\Theta(q_2), \ldots, f_\Theta(q_b)] \in R^{b\times d}$ and $[z_1, z_2, \ldots, z_b] = [f_\Psi(c_1), f_\Psi(c_2), \ldots, f_\Psi(c_b)] \in R^{b\times d}$; then, the similarity between all queries and homologs is denoted by $S_{b\times b} = [s_{ij}]_{i,j\in 1:b} = \left[h_i^T \cdot z_j\right]_{i,j\in 1:b}$, which can be computed efficiently by a simple matrix multiplication. Then, for each query, say for q_i, the task is to identify the homolog c_i among all the other negative candidates c_j, where $i \neq j$, which is a b-way classification problem. The objective function for query q_i can, therefore, be computed as $L_i = -\log\left(e^{s_{ii}}/\sum_{j=1}^{b} e^{s_{ij}}\right)$. The final objective function for the batch is the average of all queries, which is $L = \frac{1}{b}\sum_{i=1}^{b} L_i$. An illustration is presented for the contrastive loss module in Fig. 1.

Online inference. During the online inference stage, we use the candidate encoder to transform all the sequences from UniRef90 into vectors and store them in memory. Although this process is time consuming for millions of sequences, it only needs to be done once and can be completed offline. We then compute the similarity scores between the encoded query and all the sequences in the database using Eq. (1). Finally, the sequences are ranked by the similarity score and the top-K most similar sequences are retrieved using FAISS (Facebook artificial intelligence similarity search)[64]. It is worth noting that, to use the fast search algorithm in FAISS, the similarity score in Eq. (1) must be decomposable, which is why we use the dot product as the metric instead of a more complex neural network. The retrieved sequences are then used as inputs to JackHMMER for the final MSA of the input query, as shown in Fig. 1.

MSA construction. We use JackHMMER to construct MSAs on the retrieved dataset. Search parameters are fixed to the same as AF2 default settings. We only change the search iterations to get various outputs. The JackHMMER iterative search pipeline can be summarized

in a few steps. Firstly, it takes a single sequence as the query and a sequence database as a library. The probability of each amino acid appearing at a specific position of the sequence is inferred from the query and the distribution in the library and is processed as a position-specific scoring matrix (PSSM). Secondly, the library is searched with PSSM and output hits and alignments. Thirdly, the overall MSA is counted to build the new PSSM before repeating the second step.

Two parts of the output are stored for further usage, the MSA and hit sequence set. The MSAs can be directly used as input for the AF2 E2E module. For further exploration, we repeat the previous JackHMMER pipeline on the output sequence set to get new MSAs. This forces the JackHMMER program to initialize and construct on the hits given by the previous building iterations. We label them as 'distilled' MSAs because this repetitive approach forces the JackHMMER pipeline to converge.

Hyperparameters. In our study, to ensure a fair comparison, we adopted the highest-performing default parameters for MMseqs2, PSI-BLAST and DIAMOND. Specifically, we used the default settings for PSI-BLAST, the easy-search (all default) for MMseqs2 and the very-sensitive mode for DIAMOND; the search parameters inc$E$ and E for JackHMMER were set to 1×10^{-3}, $F_1 = 5 \times 10^{-4}$, $F_2 = 5 \times 10^{-5}$ and $F_3 = 5 \times 10^{-7}$.

Dataset selection

The train set we constructed included 2 million sequences selected from UR90 as queries[65] (version 2021, March). The JackHMMER algorithm was used to iteratively search for candidate sequences in UniClust30 and align these candidate sequences to the MSA. Each MSA was then truncated to 1,000 homologs. These sequences were referred back from the top-1k similar subsequences of each MSA. After conducting filtering, JackHMMER was then adopted to process the obtained distinct sequences[13]. In our implementation, the hyperparameters were set to the same as for AF2 except for the number iterations for a fair comparison. For the larger dataset, we used the BFD/MGnify dataset (version 2019, May) that contained around 300 million proteins as the candidate database.

Redundancy examination

To investigate potential redundancy between our training and testing datasets, we used PSI-BLAST for conducting searches. Specifically, the test dataset (referred to as the query dataset) was searched against our candidate datasets (UR90 or BFD/MGnify) and the resulting hits were meticulously recorded. A higher number of hits would suggest greater redundancy between the datasets, while fewer hits would indicate less overlap. Our analysis included the test sets CASP15 and a New-Protein set, both of which were chronologically posterior to the candidate datasets, thereby minimizing the chance of redundancy. Other datasets might share timeframes with our candidate sets, potentially increasing the likelihood of overlap. However, across our investigation, there was a general trend of receiving very few hits in SCOPe, CASP13 and CASP14, suggesting little redundancy.

Structural similarity

We use TM-score as the backbone and a pass in specific residue alignment for a customized structural similarity calculation. The TM-score is defined as follows:

$$\mathrm{TM-score} = \max\left[\frac{1}{L}\sum_{i=1}^{L_{ali}}\frac{1}{1+d_i^2/d_0^2}\right]$$

where L is the length of the amino acid sequence of the target protein and L_{ali} is the number of aligned residues that appear in both structures. d_i is the distance between the ith pair of aligned residues, which is dependent on the superposition of two structures. Here, 'max' refers to finding the optimal superposition to maximize this score. In the case of passing in a specified residue alignment, the superposition is performed accordingly and outputs a structural similarity score based on a sequence alignment.

MSA log Meff

Meff has a crucial role in quantifying the diversity and informational content of the alignment. Meff adjusts for redundancy by downweighting similar sequences, thus providing a more accurate representation of the unique information within the MSA. To address the broad range of Meff values and facilitate their comparison, the log Meff is commonly used. This normalization technique is especially useful in the analysis of large datasets, allowing for an effective comparison of the diversity present across different MSAs. The calculation of Meff is based on the following formula:

$$Meff = \sum_{i=1}^{N}\frac{1}{\sum_{j=1}^{N}S(i,j)}$$

where N represents the total number of sequences in the MSA and $S(i, j)$ denotes a similarity score between sequences i and j. Here, $S(i, j) = 1$ if $i = j$ and $0 \le S(i, j) \le 1$ for sequences that are not identical, typically reflecting the percentage of identical residues or using a more complex scoring function for conservative substitutions. The log Meff is calculated as follows:

$$\log \mathrm{Meff} = \log(\mathrm{Meff})$$

In the domain of protein structure prediction and bioinformatics, the log Meff indicator is invaluable for assessing MSA quality. High log Meff values indicate a diverse sequence set, conducive to accurate predictions of protein structures and functions. In contrast, low log Meff values may suggest a lack of diversity, potentially undermining the effectiveness of the computational predictions.

Execution of comparative baseline methods

In our study, we included both sequence-based methods and a structural-based alignment method, Foldseek. Moreover, HHblits could not be directly applied because it requires a database of HMMs such as those provided by HH-suite (for example, Uniclust or HHblits database derived from UniProt), where the sequences have already been converted into profile HMMs. In contrast, UniRef90 is a collection of individual protein sequences clustered at 90% sequence identity, which cannot be directly used in HHblits. The configurations for each method in our comparison were meticulously selected to ensure a fair and meaningful analysis. The specific parameters used were as follows:

- MMseqs2, with the parameters '-e 1.0 --e-profile 1.0 --num-iterations 3 -s 7.5', optimizing the balance between sensitivity and specificity for detecting sequence similarities.
- FoldSeek, with the parameter '-e 1×10^{-3}', allowing for high-precision structural alignment by leveraging its efficient search algorithm.
- HMMER, with the options '--noali -E 10,000 --F_1 2×10^{-2} --F_2 1×10^{-2} --F_3 7×10^{-3} --cpu 4 --inc$E$ 10,000', designed to maximize the detection of microbial sequences under stringent criteria.
- PSI-BLAST, with an E value of 1,000, facilitating broad, explorative sequence similarity searches across a wide array of protein databases.
- DIAMOND, with '--very-sensitive' and an E value of 1,000, combining high sensitivity with the practicality of an elevated E value to encompass a comprehensive range of potential matches.

Evaluation metrics

Sensitivity can be computed as $\frac{TP}{TP+FN}$. TP here refers to the correctly retrieved homologs. AUC-1FP refers to the area under the receiver operating characteristic (ROC) curve up to the point where the first FP is encountered. Focusing on AUC-1FP emphasizes the classifier's

precision in the early threshold or ranking settings, where the goal is to maximize TP identifications without incurring any FPs.

Memory usage

To tackle the substantial memory demands involved in storing embeddings, particularly when representing multiple domains within a single embedding framework, we integrated an efficient strategy leveraging the FAISS library. FAISS is specifically designed for scalability and efficiency, enabling the handling of large vector databases containing billions of vectors. It is optimized for both central processing unit and GPU usage, greatly reducing the memory footprint while maintaining high performance. Specifically, for the UR100 database with 277 million sequences and the BFD/MGnify database with 513 million sequences, we successfully used FAISS to achieve notable reductions in memory requirements. The reduced memory usage is shown in Supplementary Table 17.

Selection of protein language models

In the process of selecting an appropriate protein language model as our foundation, we considered both ESM-1 and ESM-2. To assess their performance in the context of our training, we randomly sampled a small subset from the SCOPe dataset for validation. For this validation, we used ESM-1 and ESM-2 as the embedding sources and calculated the recall for the top 200,000 iterations (iter1-top200k). The recall rates were 0.87 for ESM-1 and 0.86 for ESM-2, indicating no notable difference between the two models. Consequently, in this article, we opted for ESM-1 because of its slightly better performance in our tests. However, we also provide a version that uses ESM-2 for completeness.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

All data used in our work were obtained from related public datasets. We obtained all the protein targets and domain sequences from CASP (https://predictioncenter.org/) and structures from the PDB (https://www.rcsb.org/). For DHR pretraining, we downloaded UniRef50 from UniProt (https://www.uniprot.org/) and UniClust30 from UniClust (https://uniclust.mmseqs.com/). For homolog searching, we benchmarked on the SCOPe dataset (https://scop.berkeley.edu/astral/subsets/ver=2.08). For domain sequence MSA construction, we used UniRef90 from UniProt (https://www.uniprot.or g/). For the target sequence MSA construction benchmark, we downloaded bfd_mgy_colabfold from ColabFold (https://colabfold.mmseqs.com/).

Code availability

Source code for the DHR model, trained weights, inference scripts and server are available under an open-source license from GitHub (https://github.com/ml4bio/Dense-Homolog-Retrieval). The MSA construction pipeline used HMMER (https://hmmer.org/). We also used AF2 (https://github.com/deepmind/alphafold) and ColabFold (https://github.com/sokrypton/ColabFold) as our protein structure prediction modules. In addition, we used the software with the indicated version numbers in this work: Python 3.8.12, pytorch 1.11.0, numpy 1.21.2, Pandas 1.3.1, fair-esm 1.0.0, FAISS 1.7.2, Phylopandas commit cbb4a58, ColabFold 1.3.0, MMseqs2 13.45111, BLAST 2.5.0, DIAMOND 2.1.8.162 and Foldseek 7.04e0ec8.

References

52. Devlin, J., Chang, M.-W., Lee, K. & Toutanova, K. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proc. of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Volume 1, 4171–4186 (2019).
53. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* **37**, 2112–2120 (2021).
54. Mo, S. et al. Multi-modal self-supervised pre-training for regulatory genome across cell types. In *International Conference on Learning Representations* https://paperswithcode.com/paper/multi-modal-self-supervised-pre-training-for-2 (2022).
55. Tang, X. & Hu, P. Knowledge-aware self-attention networks for document grounded dialogue generation. In *International Conference on Knowledge Science, Engineering and Management* (eds Douligeris, C., Karagiannis, D. & Apostolou, D.) (Springer, 2019).
56. Karpukhin, V. et al. Dense passage retrieval for open-domain question answering. In *Proc. of the 2020 Conference on Empirical Methods in Natural Language Processing* 6769–6781 (2020).
57. Xiong, L. et al. Approximate nearest neighbor negative contrastive learning for dense text retrieval. In *International Conference on Learning Representations*. https://openreview.net/pdf?id=zeFrfgyZln (ICLR, 2021).
58. Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems* https://proceedings.neurips.cc/paper/2020/file/6b493230205f780e1bc26945df7481e5-Paper.pdf 9459-9474 (2020).
59. Zhang, Y. et al. RetGen: a joint framework for retrieval and grounded text generation modeling. In *Proc. of the AAAI Conference on Artificial Intelligence* https://cdn.aaai.org/ojs/21429/21429-13-25442-1-2-20220628.pdf (AAAI, 2022).
60. Heinzinger, M. et al. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics* **20**, 723 (2019).
61. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning* (eds Daumé, H. & Singh, A.) (JMLR, 2020).
62. Robinson, J., Chuang, C.-Y., Sra, S. & Jegelka, S. Contrastive learning with hard negative samples. In *International Conference on Learning Representations* https://openreview.net/pdf?id=CR1XOQ0UTh- (ICLR, 2021).
63. Huynh, T., Kornblith, S., Walter, M. R., Maire, M. & Khademi, M. Boosting contrastive self-supervised learning with false negative cancellation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (eds Bowyer, K., Medioni, G. & Scheirer, W.) (IEEE, 2022).
64. Johnson, J., Douze, M. & Jégou, H.Billion-scale similarity search with GPUs. *IEEE Trans. Big Data* **7**, 535–547 (2019).
65. Suzek, B. E., Wang, Y., Huang, H., McGarvey, P. B., Wu, C. H. & UniProt Consortium UniRef clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics* **31**, 926–932 (2015).

Acknowledgements

This study was supported by the Chinese University of Hong Kong (CUHK; award numbers 4937025, 4937026, 5501517 and 5501329 to Y.L.) and the IdeaBooster Fund (IDBF24ENG06 to Y.L.) and partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region (Hong Kong SAR), China (project no. CUHK 24204023 to Y.L.) and a grant from the Innovation and Technology Commission of the Hong Kong SAR, China (project no. GHP/065/21SZ to Y.L.). This research was also funded by the Research Matching Grant Scheme at CUHK (award numbers 8601603 and 8601663 to Y.L.) and a Hong Kong PhD fellowship (PF22-73180 to J.W.) from the Research Grants Council, Hong Kong SAR, China. This project was partially supported by funds from the

Focus Project of AI for Science of Comprehensive Prosperity Plan for Disciplines of Fudan University (to S.S.) and Shanghai Artificial Intelligence Laboratory (to S.S.). This work was partly supported by the Shenzhen-HongKong Joint Funding Project (Category A) under Grant No. SGDX20230116092056010 (to S.W.). This project was partially supported by funds from Schmidt Futures (M.G. and X.T.). The research presented in this paper was partially supported by the Research Grants Council of the Hong Kong Special Administrative Region, China (CUHK 14222922, RGC GRF 2151185 to I.K.). We thank Q. Li for the artwork.

Author contributions

L.H., Z.H., S.S., S.W. and Y.L. designed the research. L.H., Q.T., L.Z. and S.W. collected the dataset. L.H., Z.H. and Y.L. performed the data analysis. Z.H., S.S., X.T., M.G. and Y.L. wrote the manuscript. L.H., J.W., S.W., S.X. and I.K. set up the code base and server. S.S., M.G. and Y.L. supervised the research.

Competing interests

S.W. and L.Z. are the cofounders of Zelixir Biotech Co., Ltd. The other authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-024-02353-6.

Correspondence and requests for materials should be addressed to Siqi Sun, Mark Gerstein or Yu Li.

Peer review information *Nature Biotechnology* thanks Dong Xu and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

nature portfolio

Corresponding author(s): Yu Li

Last updated by author(s): Jul 9, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☒	☐	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	HMMER 3.3, HH-suite 3.3
Data analysis	Python=3.8.12. pytorch=1.11.0, a deep learning framework. numpy=1.21.2, a Python framework for linear algebra. Pandas=1.3.1, a framework for processing charts. fair-esm=1.0.0, pre-trained evolutionary scale models for protein. Faiss=1.7.2, a library for efficient vector similarity search. Phylopandas commit cbb4a58, https://github.com/heathcliff233/phylopandas, an optional library for efficient fasta file processing. DHR is publicly available at https://github.com/ml4bio/Dense-Homolog-Retrieval. It is the repository of the source code for this paper. Colabfold=1.3.0, a tool for efficiently running AlphaFold. MMseqs2=13.45111, a tool for fast sequence comparison. BLAST=2.5.0. DIAMOND=2.1.8.162. foldseek=7.04e0ec8.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

All data used in our work were obtained from related public datasets. We obtained all the protein targets and domain sequences from CASP (https://predictioncenter.org/) structures from Protein Data Bank (https://www.rcsb.org/).
For DHR pre-training, we downloaded UniRef50 from UniProt (https://www.uniprot.org/) and UniClust30 from UniClust (https://uniclust.mmseqs.com/). For homolog searching, we benchmarked on SCOPe dataset (https://scop.berkeley.edu/astral/subsets/ver=2.08). For domain sequence MSA construction, we use UniRef90 from UniProt (https://www.uniprot.org/). For the target sequence MSA construction benchmark, we downloaded bfd_mgy_colabfold from ColabFold (https://colabfold.mmseqs.com/).

Research involving human participants, their data, or biological material

Policy information about studies with human participants or human data. See also policy information about sex, gender (identity/presentation), and sexual orientation and race, ethnicity and racism.

Reporting on sex and gender	N/A
Reporting on race, ethnicity, or other socially relevant groupings	N/A
Population characteristics	N/A
Recruitment	N/A
Ethics oversight	N/A

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	We sampled 2M sequences from UniRef50 dataset and searched them against UniClust30 dataset. We determined the size according to the amount of data use in ESM training.
Data exclusions	No data exclusion is done prior to analysis.
Replication	Replication steps have been detailed in the Github repository. It would take about ten to twenty minutes to setup the environment. Another few minutes would be needed to download the embedding and model and a few seconds to run the experiments. Users can use their own dataset and may take more hours to build the embedding.
Randomization	We did train test split randomly following the standard machine learning practice.
Blinding	No blinding was performed as it was not relevant to this study.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern
☒	☐ Plants

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging

Plants

Seed stocks	N/A
Novel plant genotypes	N/A
Authentication	N/A