Foldseek: fast and accurate protein structure search

Michel van Kempen,[1, *] Stephanie S. Kim,[2, *] Charlotte Tumescheit,[2]
Milot Mirdita,[1] Johannes Söding,[1, 3, †] and Martin Steinegger[2, 4, †]

Highly accurate structure prediction methods are generating an avalanche of publicly available protein structures. Searching through these structures is becoming the main bottleneck in their analysis. Foldseek enables fast and sensitive comparisons of large structure sets. It reaches sensitivities similar to state-of-the-art structural aligners while being four orders of magnitude faster. Foldseek is free open-source software available at foldseek.com and as a webserver at search.foldseek.com.

Contact: soeding@mpinat.mpg.de, martin.steinegger@snu.ac.kr

The recent breakthrough in *in-silico* protein structure prediction at near-experimental quality by AlphaFold2 [1] and then RoseTTAFold [2] is revolutionizing structural biology and bioinformatics. The European Bioinformatics Institute (EBI) in collaboration with AlphaFold2/DeepMind has already made 1 106 829 protein structures publicly available and plans to extend this library to hundreds of millions of structures this year [3]. With these novel computational approaches, it will not be long before billions of high quality protein structures become available [4]. The scale of this treasure trove poses challenges to state-of-the-art analysis methods.

Currently, the most widely used approach to protein annotation and analysis is based on sequence similarity search [5–8]. The goal is to find homologous sequences from which properties of the query sequence can be interferred, such as molecular and cellular functions and structure. Despite the success of sequence-based homology inference, many proteins cannot be annotated because detecting distant evolutionary relationships from sequences alone remains challenging [9].

Detecting similarity between protein structures by 3D superposition offers higher sensitivity for identifying homologous proteins [10]. The imminent availability of high-quality structure models for any protein of interest could allow us to use structure comparison to improve homology-based inference and structural, functional and evolutionary analyses. However, despite decades of effort to improve speed and sensitivity of structural aligners, current tools are much too slow to cope with the expected scale of structure databases.

For example, searching with a single query structure through a database with 100 million protein structures would take the popular TMalign [11] tool around a month on one CPU core, and an all-versus-all comparison would take around 10 millennia on a 1 000 core cluster. In comparison, sequence searching is five orders of magnitude faster: An all-versus-all comparison of 100 M sequences would take MMseqs2 [6] at high search sensitivity only around a week on the same cluster.

Structural alignment tools are slower for two reasons. First, whereas sequence search tools employ fast and sensitive prefilter algorithms to gain several orders of magnitude in speed, no comparable prefilters exist for structure searches. Second, structural similarity scores are non-local: changing the alignment in one part affects the similarity in all other parts. For example in TMalign, two highly interdependent optimizations are performed: The pairing up of residues that are to be aligned with each other, and the superposition of the 3D structures by minimizing some distance measure between aligned residues. Most structural aligners, such as the popular TMalign, DALI, and CE [11–13], solve the alignment optimization problem by iterative or stochastic optimization.

To increase speed, a crucial idea is to describe the amino acid backbone of proteins as sequences over a structural alphabet and compare structures using sequence alignments [14]. In this way, structural alphabets reduce structure comparisons to much faster sequence alignments. Many ways to discretize the local amino acid backbone have been proposed [15]. Most, such as CLE, 3D-BLAST, and Protein Blocks, discretize the conformations of short stretches of usually 3 to 5 C_α atoms [16–18]. 3D-BLAST and CLE trained a substitution matrix for their structural alphabet and rely on an aligner like BLAST [5] to perform the sequence searches.

For Foldseek, we developed a novel type of structural alphabet that does not describe the backbone but rather tertiary interactions. The 20 states of the 3D-interactions (3Di) alphabet describe for each residue i the geometric conformation with its spatially closest residue j. Compared to the various backbone structural alphabets, 3Di has three key advantages: First, the dependency of consecutive 3Di letters on each other is much weaker than for backbone structural alphabets, where for instance a helix state is followed by another helix state with high probability. The dependency decreases information density and results in high-scoring false alignments. Second, the frequencies of the 3Di states are more evenly distributed than for backbone states, for which 60 % describe generic secondary structure states. This further increases information density in 3Di sequences (**Supplementary Table 1**) and decreases false positives. Third, in backbone structural alphabets, less information is contained in the highly conserved protein cores (consisting mostly of regular secondary structure elements) and more in the predominantly non-conserved coil/loop regions. In contrast, 3Di sequences have the highest information density in conserved cores and the lowest in loop regions.

Foldseek (**Fig. 1a**) (1) discretizes the query structures into sequences over the 3Di alphabet and then searches through the 3Di sequences of the target structures using the double-diagonal k-mer-based prefilter and gapless alignment prefilter modules from MMseqs2, our highly optimized and parallelized open-source sequence search software [6]. (2) High scoring hits

[*] These two authors contributed equally

[†] Authors to whom correspondence should be addressed

[1] Quant. & Comput. Biology, Max Planck Institute for Multidisciplinary Sciences, Göttingen, Germany. [2] School of Biological Sciences, Seoul National University, Seoul, South Korea. [3]Campus-Institute Data Science (CIDAS), Goldschmidtstrasse 1, 37077 Göttingen, Germany. [4] Artificial Intelligence Institute, Seoul National University, Seoul, South Korea

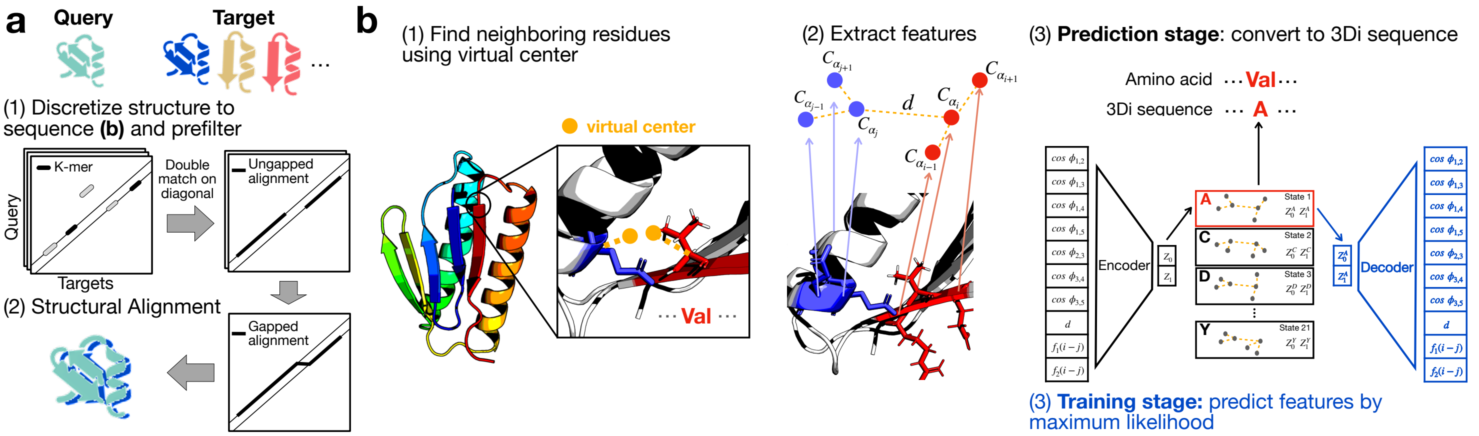

FIG. 1. **Foldseek workflow.** (**a**) Foldseek searches a set of query structures through a set of target structures. (1) Query and target structures are discretized into 3Di sequences (see **b**). To detect candidate structures, we apply the fast and sensitive k-mer and ungapped alignment prefilter from MMseqs2 on the 3Di sequences. (2) Followed by a local alignment using a vectorized Smith-Waterman algorithm combining both 3Di and amino acid substitution scores. Alternatively, a global alignment is computed with an accelerated TMalign version. (**b**) Learning the 3Di alphabet: (1) 3Di states describe tertiary interaction between a residue i and its nearest neighbor j. Nearest neighbors have the closest virtual center distance (yellow). Virtual center (**Supplementary Fig. 1**) positions were optimized for maximum search sensitivity. (2) To describe the interaction geometry of residues i and j, we extract seven angles, the euclidean C_α distance, and two sequence distance features from the six C_α coordinates of the two backbone fragments (blue, red). (3) These 10 features are used to define 20 3Di states by training a vector-quantized variational autoencoder [19] modified to learn states that are maximally evolutionarily conserved. For structure searches, the encoder predicts the best-matching 3Di state for each residue.

are aligned locally (default) or aligned globally with TMalign. The novel local alignment stage combines structural and amino acid substitution scores for improved sensitivity without sacrificing speed. The construction of the 3Di alphabet is summarized in **Fig. 1b** and **Supplemental Figs 1–3**.

To minimize high-scoring false positives and provide reliable E-values, for each match the score of the reversed query sequence is subtracted from the original score. Furthermore, a compositional bias correction is applied that lowers the substitution scores of 3Di states enriched within a local 40 residue sequence window (see "Pairwise local structural alignments"). E-values are calculated based on an extreme-value score distribution whose parameters are predicted by a neural network from 3Di sequence composition and length (see "E-Values").

We measured the sensitivity and speed of Foldseek and six structure alignment tools with single-domain structures (**Fig. 2a-b**) on the SCOPe40 dataset [20]. This dataset contains 11 211 protein domains clustered at 40 % sequence identity. We performed an all-versus-all search and compared the performance for finding members of the same SCOPe family, superfamily, and fold (true positive matches, TPs) by measuring the fraction of TPs out of all possible correct matches for the query until the fifth false positive (FP). FPs are matches to a different fold (see "SCOPe Benchmark"). The sensitivity was measured by the area under the curve (AUC) of the cumulative ROC curve up to the fifth FP.

Foldseek reaches sensitivities at family and superfamily level below Dali, higher than the structural aligner CE, and performs similarly to TMalign and TMalign-fast. Foldseek is much more sensitive than structural alphabet-based search tools 3D-BLAST and CLE-SW (**Fig. 2a-b**). Even on the fold level, where most TPs are between non-homologous superfamilies, it is more sensitive than CE and similar to TMalign. Yet on this small, single-domain benchmark it is more than 3,000 times faster than TMalign, DALI, and CE (**Fig. 2b**). On the much larger AlphaFoldDB, where Foldseek approaches its full speed, it is around 184,600 and 23,000 faster than DALI and TMalign, respectively (**Fig. 2d**). Its E-values are accurate, which is critical for homology searching (**Fig. 2c**)

To assess the reliability and speed of Foldseek with full-length protein chains, we performed an all-versus-all Foldseek search on the AlphaFoldDB. For each query structure we computed the TMalign score of Foldseek's second best match (the best match is the self-match). We ignored matches for which the average of the predicted Local Distance Difference Test (pLDDT [1]) from query and target is below 80 or which are fragmented. All but 1,675 out of 133,813 second-best matches with high alignment confidence (Foldseek score per aligned column ≥ 1.0) had a good TM-score (≥ 0.5), indicating that the fold was correctly recognized (**Supplementary Fig. 4**). Manual inspection of outliers with high Foldseek score per column and low TMscores (< 0.5) revealed Foldseek matches with multiple smaller, correctly aligned regions (**Supplementary Table 2**). Even though their average pLDDT is above 80, the relative orientation of correctly folded segments is often not correctly predicted by AlphaFold2. TM-align does not identify these as homologs, as it searches for global structural superpositions, thus overlooking significant local similarities.

We investigated the sensitivity for detecting very remote homologs by counting the number of cross-kingdom hits within AlphaFoldDB. Foldseek and MMseqs2 found cross-kingdom hits for 34.5% and 27.4% of the 364 357 queries, respectively. Overall, Foldseek finds 3.4 times as many cross-kingdom hits as MMseqs2 (see **Supplementary Fig. 5**).

To facilitate access to Foldseek, we developed a user-friendly webserver optimized to quickly return results for sin-

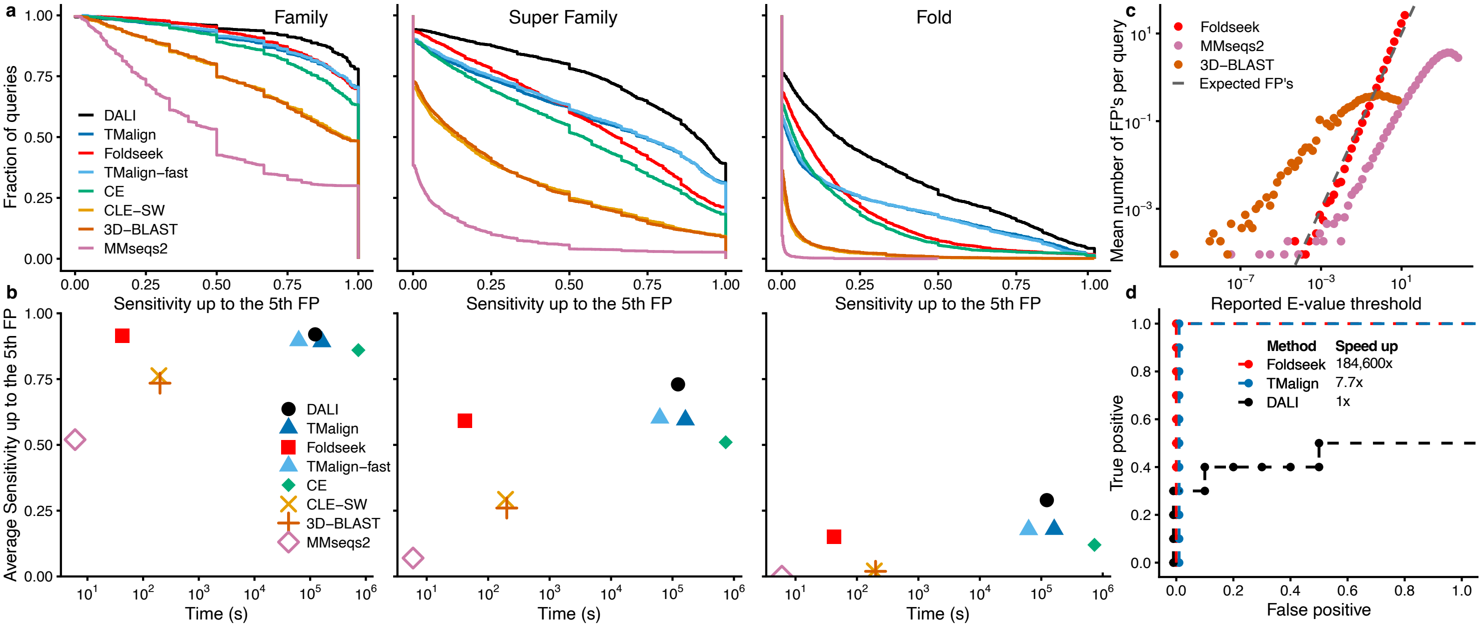

FIG. 2. **Foldseek reaches similar sensitivities as structural aligners at thousands of times their speed** (**a**) Cumulative distributions of sensitivity for homology detection. Sensitivity is the area under the ROC curve up to the fifth false positive, for all-versus-all searches with the 11 211 single-domain structures of the SCOPe40 database). True positives are matches within the same family, superfamily or fold (see main text). (**b**) Sensitivity versus total runtime on an AMD EPYC 7702P 64-core CPU for the all-versus-all searches. (**c**) Accuracy of reported E-values: Mean number of FP hits versus reported E-value threshold. (**d**) Top10 hits of search with RdRp (6M71_A) through the AlphaFold/Proteome with Foldseek, TMalign and DALI.

gle queries. It performs searches through three structure databases, AlphaFoldDB/Proteome, AlphaFoldDB/Swiss-Prot, and the PDB100, using one of three alignment methods: standard Foldseek (default), Foldseek without amino acid scoring, and TMalign. The server takes PDB files as input and returns a list of matched structures, query-target sequence alignments, similarity scores, and E-values or TMscores.

We compared the Foldseek webserver with TMalign and DALI by searching with the SARS-CoV-2 RNA-dependent RNA polymerase (RdRp, PDB: 6M71_A [21]; 942 residues) through the AlphaFoldDB (Proteome + Swiss-Prot) containing 804 872 protein structures. The searches took TMalign 33h and DALI 10 days to complete on a single core. Foldseek took 5 seconds, which is about 23 000, 180 600 times faster than TMalign and DALI respectively. We compared the top 10 hits of the AlphaFoldDB/Proteome database (**Fig. 2d**). Foldseek as well as TMalign contain only reverse transcriptase (RT) domains, which are structurally similar to RdRps. DALI finds three RdRp and two RT hits, and five FPs hits to kinases (**Supplementary Table 3**). Foldseek finds significant hits with E-values between 10^{-7} to 10^{-6}, while TMalign reports low TM-scores between 0.419 and 0.42. This illustrates a key difference between structural aligners, which depend on finding a global 3D superposition, and Foldseek's local alignment. Foldseek is independent of the relative orientation of domains and therefore excels at detecting homologous multi-domain structures.

The availability of high-quality protein structures for nearly every structured protein is going to be transformative for structural biology and bioinformatics. What could until recently only be done by analyzing sequences can now be done with structures. The main limitation in our view, the four orders of magnitude slower speed of structure comparisons, is removed by Foldseek.

REFERENCES

[1] Jumper, J. *et al.* *Nature* **596**, 583–589 (2021).
[2] Baek, M. *et al.* *Science* **373**, 871–876 (2021).
[3] Varadi, M. *et al.* *Nucleic Acids Res* **50**, D439–D444 (2022).
[4] Burley, S. K. *et al.* *Nucleic Acids Res* **49**, D437–D451 (2021).
[5] Altschul, S. F. *et al.* *J Mol Biol* **215**, 403–410 (1990).
[6] Steinegger, M. & Söding, J. *Nat Biotechnol* **35**, 1026–1028 (2017).
[7] Steinegger, M. *et al.* *BMC Bioinform* **20**, 473 (2019).
[8] Buchfink, B. *et al.* *Nat Methods* **18**, 366–368 (2021).
[9] Rost, B. *Protein Eng Des Sel* **12**, 85–94 (1999).
[10] Illergård, K. *et al.* *Proteins* **77**, 499–508 (2009).
[11] Zhang, Y. & Skolnick, J. *Nucleic Acids Res* **33**, 2302–2309 (2005).
[12] Holm, L. *Methods Mol Biol* **2112**, 29–42 (2020).
[13] Shindyalov, I. N. & Bourne, P. E. *Protein Eng Des Sel* **11**, 739–747 (1998).
[14] Guyon, F. *et al.* *Nucleic Acids Res* **32**, W545–W548 (2004).
[15] Ma, J. & Wang, S. *Adv Protein Chem Struct Biol* **94**, 121–175 (2014).
[16] Wang, S. & Zheng, W.-M. *J Bioinform Comput Biol* **6**, 347–366 (2008).
[17] Yang, J.-M. & Tung, C.-H. *Nucleic Acids Res* **34**, 3646–3659 (2006).
[18] de Brevern, A. G. *et al.* *Proteins* **41**, 271–287 (2000).
[19] Van den Oord, A. *et al.* *Adv Neur Inf Proc Syst (NIPS)* **30** (2017).
[20] Chandonia, J.-M. *et al.* *Nucleic Acids Res* **47**, D475–D481 (2019).
[21] Gao, Y. *et al.* *Science* **368**, 779–782 (2020).

Acknowledgements

We thank Nicola Bordin, Ian Sillitoe and Christine Orengo for reporting issues and providing valuable feedback, and Yang Zhang and Marcin Wojdyr for making TMalign and the Gemmi library freely accessible, and Do-Yoon Kim for creating the Foldseek logo.

M.S. acknowledges support from the National Research Foundation of Korea grants [2019R1A6A1A10073437, 2020M3A9G7103933, 2021R1C1C102065, 2021M3A9-I4021220]; and the Creative-Pioneering Researchers Program through Seoul National University. S.K. acknowledges support by the National Research Foundation of Korea (NRF) grant No. 2019R1A6A1A10073437. M.M. and J.S. acknowledge support by the German ministry for education and research (BMBF) via grant horizontal4meta.

This work used the Scientific Compute Cluster at GWDG, the joint data center of Max Planck Society for the Advancement of Science (MPG) and University of Göttingen.

Author contributions

M.K., S.K., J.S. & M.S. designed research. M.K., S.K., C.T., & M.S. developed code and performed analyses. M.K. and J.S. developed the 3Di alphabet. M.M. developed the webserver. M.K., S.K., C.T., M.M., J.S. & M.S. wrote the manuscript.

Competing financial interests

The authors declare no competing financial interests.

METHODS

Overview Foldseek enables fast and sensitive comparison of large structure sets. It encodes structures as sequences over the 20-state 3Di alphabet and thereby reduces structural alignments to 3Di sequence alignments. The 3Di alphabet developed for Foldseek describes tertiary residue-residue interactions instead of backbone conformations and proved critical for reaching high sensitivities. Foldseek's prefilter finds two *similar*, spaced 3Di k-mer matches in the same diagonal of the dynamic programming matrix. By not restricting itself to exact matches, the prefilter achieves high sensitivity while reducing the number of sequences for which full alignments are computed by several orders of magnitude. Further speed-ups are achieved by multi-threading and utilizing single instruction multiple data (SIMD) vector units. Owing to the SIMDe library (github.com/simd-everywhere/simde), Foldseek runs on a wide range of CPU architectures (x86_64, arm64, ppc64le) and operating systems (Linux, macOS). The core modules of Foldseek, which build on the MMseqs2 framework [22], are described in the following paragraphs.

Create database The `createdb` module converts a set of Protein Data Bank (PDB; [23]) or macromolecular Crystallographic Information File (mmCIF) formatted files into an internal Foldseek database format using the gemmi package (project-gemmi.github.io). The format is compatible with the MMseqs2 database format, which is optimized for parallel access. We store each chain as a separate entry in the database. The module follows the MMseqs2 `createdb` module logic, however, in addition to the amino acid sequence it computes the 3Di sequence from the 3D atom coordinates of the C_α and C_β, C_{backbone} and N coordinates (see "Descriptors for 3Di structural alphabet"). The 3Di and amino acid sequence, and the C_α floating-point coordinates are stored in the database.

Prefilter The `prefilter` module generates similar k-mers and detects double, consecutive, similar k-mer matches that occur on the same diagonal. In contrast to the MMseqs2 prefilter, the Foldseek prefilter utilizes the 3Di information instead of the amino acid sequence information to generate similar k-mers using a 3Di substitution matrix (see "3Di substitution score matrix"). This criterion suppresses hits to non-homologous structures effectively, as they are less likely to have consecutive k-mer matches on the same diagonal by chance. To counteract the effect of regions with 3Di compositions that differ from the database average, a compositional bias correction is applied in a way analogous to MMseqs2 [24]. For each hit we perform an ungapped alignment over the diagonals with double, consecutive, similar k-mer matches and sort those by the maximum ungapped diagonal score. Alignments with a score of at least 15 bits are passed onto the next stage.

Pairwise local structural alignments After the prefilter has removed the vast majority of non-homologous sequences, pairwise alignments are performed on the remaining sequences in the `structurealign` module. Sequences are aligned using a SIMD accelerated Smith-Waterman algorithm [25, 26]. We extended this implementation to support amino acid and 3Di scoring, compositional bias correction, and 256-bit-wide vectorization. The score linearly combines amino acid and 3Di substitution scores with weights 1.4 and 2.1, respectively. A compositional bias correction is applied to the amino acid and 3Di scores. To further suppress high-scoring false positive matches, for each match we align the reversed query sequence against the target and subtract the reverse score from the forward score.

E-Values To estimate E-values for each match, we trained a neural network to predict the mean μ and scale parameter λ of the extreme value distribution for each query. We built a module in Foldseek called `computemulambda`, which takes a query and database structures as input and aligns the query against a randomly shuffled version of the database sequences. For each query sequence the module produces N random alignments and fits to their scores an extreme-value (Gumbel) distribution. The maximum likelihood fitting is done using the Gumbel fitting function taken from HMMER3 (`hmmcalibrate`) [27]. To train the network, we predicted μ and λ for 100 000 sequences sampled from the AlphaFoldDB. We trained the network to predict μ and λ from the mono-residue composition of the query and its length. The network has 22 input nodes, 2 fully-connected layers with 32 nodes each (ReLU activation) and two linear output nodes. The optimizer ADAM with learning rate 0.001 was used for training. When testing the resulting E-values on searches with scrambled sequences, the log of the mean number of false positives per query turned out to have an accurately linear dependence on the log of the reported E-values, albeit with a slope of 0.32 instead of 1. We therefore correct the E-values from the neural network by taking them to the power of 0.32. We compared how well the mean number of FPs at a given E-value agreed with the E-values reported by Foldseek, MMseqs2, and 3D-Blast, (**Fig. 2c** for SCOPe and **Supplementary Fig. 6** for AlphaFoldDB). We considered a hit as FP if it was in a different fold and had a TM-score lower than 0.3. Furthermore, we ignored all cross-fold hits within the four- to eight-bladed β-propeller superfamilies (SCOPe b.66-b.70) and within the Rossman-like folds (c.2-c.5, c.27, c.28, c.30, and c.31) because of the extensive cross-fold homologies within these groups [28].

Pairwise global structural alignments using TM-align We also offer the option to use TM-align for pairwise alignments. We implemented TM-align based on the C_α atom coordinates and made adjustments to improve the (1) speed and (2) memory usage. (1) TM-align performs multiple floating-point based Needleman-Wunsch (NW) alignment steps, while applying different scoring functions (e.g., score secondary structure, Euclidean distance of superposed structures or fragments, etc.) TM-align's NW code did not take advantage of SIMD instructions, therefore, we replaced it by parasail's [29] SIMD-based NW implementation and extended it to support the different scoring functions. We also replaced the TM-score computation using fast_protein_cluster's SIMD based implementation [30]. Our NW implementation does not compute exactly the same alignment since we apply affine gap costs while TM-align does not. (2) TMalign requires 17 bytes × query length × target length of memory, we reduce the constant overhead from 17 to 4 bytes. If Foldseek is used in TM-align mode (parameter `--alignment-type 1`), we replace the reported E-value column with TM-scores normalized by the query length. The results are ordered in descending order by TM-score.

Descriptors for 3Di structural alphabet The 3Di alphabet describes the tertiary contacts between residues and their nearest neighbors in 3D space. For each residue i the conformation of the local backbone around i together with the local backbone around its nearest neighbor j is approximated by 20 discrete states (see **Supplementary Fig. 3**). We chose the alphabet size $A = 20$ as a trade-off between encoding as much information as possible (large A) and limiting the number of similar 3Di k-mers that we need to generate in the k-mer based prefilter. This number scales with A^k, giving us an alphabet size similar to the size of the amino acid alphabet. The discrete single-letter states are formed from neighborhood descriptors containing ten features encoding the conformation of backbones around residues i and j represented by the C_α atoms $(\mathrm{C}_{\alpha,i-1}, \mathrm{C}_{\alpha,i}, \mathrm{C}_{\alpha,i+1})$ and $(\mathrm{C}_{\alpha,j-1}, \mathrm{C}_{\alpha,j}, \mathrm{C}_{\alpha,j+1})$. The descriptors use the five unit vectors along the following directions,

$$\begin{array}{ll} u_1 : \mathrm{C}_{\alpha,i-1} \rightarrow \mathrm{C}_{\alpha,i} & u_4 : \mathrm{C}_{\alpha,j} \rightarrow \mathrm{C}_{\alpha,j+1} \\ u_2 : \mathrm{C}_{\alpha,i} \quad\ \rightarrow \mathrm{C}_{\alpha,i+1} & u_5 : \mathrm{C}_{\alpha,i} \quad \rightarrow \mathrm{C}_{\alpha,j} \\ u_3 : \mathrm{C}_{\alpha,j-1} \rightarrow \mathrm{C}_{\alpha,j}. & \end{array}$$

We define the angle between u_k and u_l as ϕ_{kl}, so $\cos\phi_{kl} = u_k^T u_l$. The seven features $\cos\phi_{12}$, $\cos\phi_{34}$, $\cos\phi_{15}$, $\cos\phi_{35}$, $\cos\phi_{14}$, $\cos\phi_{23}$, $\cos\phi_{13}$, and the distance $|\mathrm{C}_{\alpha,i} - \mathrm{C}_{\alpha,j}|$ describe the conformation between the backbone fragments. In addition, we encode the sequence distance with the two features $\mathrm{sign}(i-j)\,\min(|i-j|, 4)$ and $\mathrm{sign}(i-j)\,\log(|i-j|+1)$.

Learning the 3Di states using a VQ-VAE The ten-dimensional descriptors were discretized into an alphabet of 20 states using a variational autoencoder with vector-quantized latent variables (VQ-VAE) [31]. In contrast to the standard VQ-VAE, we trained the VQ-VAE not as a simple generative model but rather to learn states that are maximally conserved in evolution. To that end, we trained it with pairs of descriptors $\mathbf{x}_n, \mathbf{y}_n \in \mathbb{R}^{10}$ from structurally aligned residues, to predict the distribution of $\mathbf{y}_n$ from $\mathbf{x}_n$. The VQ-VAE consists of an encoder and decoder network with the discrete latent 3Di state as a bottleneck in-between. The encoder network embeds the 10-dimensional descriptor $\mathbf{x}_n$ into a two-dimensional continuous latent space, where the embedding is then discretized by the nearest centroid, each centroid representing a 3Di state. Given the centroid, the decoder predicts the probability distribution of the descriptor $\mathbf{y}_n$ of the aligned residue. After training, only encoder and centroids are used to discretize descriptors. Encoder and decoder networks are both fully connected with two hidden layers of dimension 10, a batch normalization after each hidden layer and ReLU as activation functions. The encoder, centroids, and decoder have 242, 40, and 352 parameters, respectively. The output layer of the decoder consists of 20 units predicting μ and σ^2 of the descriptors x of the aligned residue, such that the decoder predicts $\mathcal{N}(x|\mu, I\sigma^2)$ (with diagonal covariance). We trained the VQ-VAE on the loss function defined in Equation (3) in [31] (with commitment loss $= 0.25$) using the deep-learning framework PyTorch (version 1.9.0), the ADAM optimizer, with a batch size of 512, and a learning rate of 10^{-3} over 4 epochs. Using Kerasify, we integrated the encoder network into Foldseek. The domains from the SCOPe database were split 80 %/20 % by fold into training and validation sets. For the training, we structurally aligned the structures with TMalign, removed all alignments with a TM-score below 0.6, and removed all aligned residue pairs with a distance between their C_α atoms of more than 5 Å. We trained the VQ-VAE with 100 different initial parameters and chose the model that was performing best in the benchmark on the validation dataset (the highest sum of ratios between 3Di AUC and TMalign AUC for family, superfamily and fold level).

3Di substitution score matrix We trained a BLOSUM-like substitution matrix for 3Di sequences from pairs of structurally aligned residues used for the “VAE-VQ training”. First, we determined the 3Di states of all residues. Next, the substitution frequencies between 3Di states were calculated by counting how often two 3Di states were structurally aligned. (Note that the substitution frequencies from state A to B and the opposite direction are equal.) Finally, the score $\mathrm{S}(x,y) = 2\log_2 \frac{p(x,y)}{p(x)\,p(y)}$ for substituting state x through state y is the log-ratio between the substitution frequency $p(x,y)$ and the probability that the two states occur independently, scaled by the factor 2.

Optimize nearest-neighbor selection To select nearest-neighbor residues that maximize the performance of the resulting 3Di alphabet in finding and aligning homologous structures, we introduced the virtual center V of a residue. The virtual center position is defined by the angle θ (V-C_α-C_β), the dihedral angle τ (V-C_α-C_β-N), and the length l ($|V - \mathrm{C}_\alpha|$). For each residue i we selected the residue j with the smallest distance between their virtual centers. The virtual center was optimized on the training and validation structure sets used for the VQ-VAE training by creating alphabets for positions with $\theta \in [0, 2\pi]$, $\tau \in [-\pi, \pi]$ in 45° steps, and $l \in \{1.53\text{Å}\, k : k \in \{1, 1.5, 2, 2.5, 3\}\}$ (1.53Å is the distance between C_α and C_β). The virtual center defined by $\theta = 270°$, $\tau = 0°$ and $l = 2$ performed best in the benchmark. For glycines, the C_β positions were approximated by forming a tetrahedral from C_α. This virtual center preferably selects long-range, tertiary interactions and only falls back to selecting interactions to $i+1$ or $i-1$ when no other residues are nearby. In that case, the interaction captures only the backbone conformation.

SCOPe Benchmark We downloaded SCOPe 2.07 [32] structures, clustered at 40 % sequence identity, containing 11 211 domains, for the generation of 3Di states and for the performance evaluation of Foldseek. The SCOPe benchmark set consists of single domains with an average length of 174 residues. In our benchmark, we compare the domains all-versus-all. Per domain, we measured the fraction of detected TPs up to the 5th false positive. For family-, superfamily- and fold-level recognition, TPs were defined as same family, same superfamily and not same family, and same fold and not same superfamily, respectively. Hits from different folds are FPs.

AlphaFold database used for all-versus-all search We downloaded the AlphaFoldDB [33] version 1 containing 365,198 protein models and searched it all-versus-all using Foldseek `-s 9.5 --max-seqs 2000`. For our second best hit analysis we consider only models with: (1) an average C_α's pLDDT greater than or equal to 80, and (2) models of non-fragmented domains. We also computed the structural similarity for each pair using TMalign (default options).

Performance evaluation: Sensitivity In order to evaluate the sensitivity of the structural alignment tools, we used a cumulative ROC curve analysis. After sorting the alignment result of each query, we calculated the fraction of TPs in the list up to the 5th false positives. We quantitatively measured the sensitivity by comparing the area under the curve (AUC) for family-, superfamily-, and fold-level classifications.

Performance evaluation: Runtime Using the SCOPe benchmark dataset, the runtime of the pairwise structural alignment was evaluated for all methods. Depending on the processing time of each tool, the runtimes of the structural alignment tools TM-align, DALI, and CE were estimated on 10 % of the benchmark set (1 121 proteins randomly selected from the SCOPe domains). Tools with multi-threading support (MMseqs2 and Foldseek) were executed with 64 threads, tools without were parallelized by breaking the query set into 64 equally sized chunks and executing them in parallel.

Tools and options for benchmark comparison Following are command lines used in the SCOPe benchmark.

Foldseek We used Foldseek commit `4de45` during this analysis. Foldseek was run with the following parameters:

```
--threads 64 -s 9.5 -e 10 --max-seqs 2000
```

MMseqs2 We used the default MMseqs2 (release `13-45111`) search algorithm to obtain the sequence-based alignment result. MMseqs2 sorts the results by e-value and score. We searched with: `--threads 64 -s 7.5 -e 10000 --max-seqs 2000`

CLE-Smith-Waterman We used PDB Tool v4.80 (`github.com/realbigws/PDB_Tool`) to convert the benchmark structure set to CLE sequences. After the conversion, we used SSW [26] (commit `ad452e`) to align CLE sequences all-versus-all. We sorted the results by alignment score. The following parameters were used to run SSW: (1) protein alignment mode (`-p`), (2) gap open penalty of 100 (`-o 100`), (3) gap extend penalty of 10 (`-e 10`), (4) CLE's optimized substitution matrix (`-a cle.shen.mat`), (5) returning alignment (`-c`). The gap open and extend values were inferred from DeepAlign [34]. The results are sorted by score in descending order.

```
ssw_test -p -o 100 -e 10 -a cle.shen.mat -c
```

3D-BLAST We used 3D-BLAST (beta102) with BLAST+ (2.2.26) and SSW [26] (version ad452e). We first converted the PDB structures to a 3D-BLAST database using `3d-blast -sq_write` and `3d-blast -sq_append`. We searched the structural sequences against the database using `blastp` with the following parameters: (1) we used 3D-BLAST's optimized substitution matrix (`-M 3DBLAST`), (2) number of hits and alignments shown of 12 000 (`-v 12000 -b 12000`), (3) E-value threshold of 1 000 (`-e 1000`) (4) disabling query sequence filter (`-F F`) (5) gap open of 8 (`-G 8`), and (6) gap extend of 2 (`-E 2`). 3D-BLAST's results are sorted by E-value in ascending order:

```
blastall -p blastp -M 3DBLAST -v 12000 -b 12000 -e
1000 -F F -G 8 -E 2
```

For Smith-Waterman we used (1) gap open of 8 (2) gap extend of 2 and (3) returning alignments (`-c`) (4) using the 3D-BLAST's optimized substitution matrix (`-a 3DBLAST`), (5) protein alignment mode (`-p`): `ssw_test -o 8 -e 2 -c -a 3DBLAST -p`. Presented in **Figure 2** are the Smith-Waterman results, since BLAST performed worse with an average AUC of 0.573, 0.127, 0.009 for family-, superfamily- and fold-classification, respectively.

TMalign We downloaded and compiled the `TMalign.cpp` source code (version 2019/08/22) from the Zhang group website. We ran the benchmark using default parameters and `-fast` for the fast version. We used the TM score normalized by the 1st chain (query) in all our analyses. Default: `TMalign query.pdb target.pdb`

Fast: `TMalign query.pdb target.pdb -fast`

DALI We installed the standalone DaliLite.v5. For the SCOPe benchmark set, input files were formatted in DAT files with DALI's `import.pl`. The conversion to DAT format produced 11 137 valid structures out of the 11 211 initial structures for the SCOPe benchmark. After formatting the input files, we calculated the protein alignment with DALI's structural alignment algorithm. The results were sorted by DALI's Z-score:

```
import.pl -pdbfile query.pdb -pdbid PDBid -dat DAT
dali.pl -cd1 queryDATid -db targetDB.list -TITLE
systematic -dat1 DAT -dat2 DAT -outfmt "summary"
-clean
```

CE We used BioJava's [35] (version 5.4.0) implementation of the combinatorial extension (CE) alignment algorithm. We modified one of the modules of BioJava under shape configuration to calculate the CE value. Our modified `CEalign.jar` file requires a list of query files, path to the target PDB files, and an output path as input parameters. This Java module runs an all-versus-all CE calculation. The Jar file of our implementation of CE calculation is provided.

```
java -jar CEalign.jar querylist.txt
TargetPDBDirectory OutputDirectory
```

Hardware specifications for benchmarks The runtime benchmarks were executed on a machine with an AMD EPYC 7702P 64-core CPU and 1024 GB RAM memory.

Webserver The Foldseek webserver is a continuation of the MMseqs2 webserver [36]. To allow for searches in seconds we implemented MMseqs2's pre-computed database indexing capabilities in Foldseek. Using these, the search databases can be held fully in system memory by the operating system and instantly accessed by each Foldseek process, thus avoiding expensive accesses to slow disk drives. A similar mechanism was used to store and read the associated taxonomic information. The AlphaFoldDB/Proteome (v1), AlphaFoldDB/Swiss-Prot (v2), and PDB100 require 3.9GB, 3.6GB, and 2.2GB RAM, respectively. The databases are kept in memory using `vmtouch` (`github.com/hoytech/vmtouch`).

Code availability Foldseek is GPLv3-licensed free open source software. The source code and binaries for Foldseek can be downloaded at `github.com/steineggerlab/foldseek`. The webserver code is available at `github.com/soedinglab/mmseqs2-app`. The analysis scripts are available at: `github.com/steineggerlab/foldseek-analysis`.

Data availability Benchmark data and Foldseek databases are available at: `wwwuser.gwdg.de/~compbiol/foldseek`.

REFERENCES

[22] Steinegger, M. & Söding, J. *Nat Biotechnol* **35**, 1026–1028 (2017).

[23] Burley, S. K. *et al. Nucleic Acids Res* **47**, D520–D528 (2019).

[24] Hauser, M. *et al. Bioinformatics* **32**, 1323–1330 (2016).

[25] Farrar, M. *Bioinformatics* **23**, 156–161 (2007).

[26] Zhao, M. *et al. PLOS One* **8**, e82138 (2013).

[27] Eddy, S. R. *PLOS Comput Biol* **7**, e1002195 (2011).

[28] Söding, J. & Remmert, M. *Curr Opin Struct Biol* **21**, 404–411 (2011).

[29] Daily, J. *BMC Bioinform* **17**, 81 (2016).

[30] Hung, L.-H. & Samudrala, R. *Bioinformatics* **30**, 1774–1776 (2014).

[31] Van den Oord, A. *et al. Adv Neur Inf Proc Syst (NIPS)* **30** (2017).

[32] Chandonia, J.-M. *et al. Nucleic Acids Res* **47**, D475–D481 (2019).

[33] Varadi, M. *et al. Nucleic Acids Res* **50**, D439–D444 (2022).

[34] Jiménez-Moreno, A. *et al. J Struct Biol* **213**, 107712 (2021).

[35] Lafita, A. *et al. PLOS Comput Biol* **15**, e1006791 (2019).

[36] Mirdita, M. *et al. Bioinformatics* **35**, 2856–2858 (2019).