nature biotechnology

Brief Communication

https://doi.org/10.1038/s41587-023-01773-0

Fast and accurate protein structure search with Foldseek

Received: 17 February 2022

Accepted: 30 March 2023

Published online: 8 May 2023

Check for updates

Michel van Kempen[1,6], Stephanie S. Kim[2,6], Charlotte Tumescheit[2], Milot Mirdita[1,2], Jeongjae Lee[2], Cameron L. M. Gilchrist[2], Johannes Söding[1,3] ✉ & Martin Steinegger[2,4,5] ✉

As structure prediction methods are generating millions of publicly available protein structures, searching these databases is becoming a bottleneck. Foldseek aligns the structure of a query protein against a database by describing tertiary amino acid interactions within proteins as sequences over a structural alphabet. Foldseek decreases computation times by four to five orders of magnitude with 86%, 88% and 133% of the sensitivities of Dali, TM-align and CE, respectively.

The recent developments in in silico protein structure prediction at near-experimental quality[1,2] are advancing structural biology and bioinformatics. The European Bioinformatics Institute already holds over 214 million structures predicted by AlphaFold2 (ref. 3), and the ESM Atlas contains over 617 million metagenomic structures predicted by ESMFold[4]. The scale of these databases poses challenges to state-of-the-art analysis methods.

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 inferred, 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 three-dimensional (3D) superposition offers higher sensitivity for identifying homologous proteins[10]. The availability of high-quality structures for any protein of interest allows us to use structure comparison to improve homology 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 today's scale of structure databases.

Searching with a single query structure through a database with 100 million protein structures would take the popular TM-align[11] tool a month on one CPU core, and an all-versus-all comparison would take 10 millennia on a 1,000-core cluster. Sequence searching is four to five orders of magnitude faster: an all-versus-all comparison of 100 million sequences would take MMseqs2 (ref. 6) only around a week on the same cluster.

Structural alignment tools (reviewed in ref. 12) are slower for two reasons. First, whereas sequence search tools employ fast and sensitive prefilter algorithms to gain orders of magnitude in speed, no similar prefilters exist for structure alignment. Second, structural similarity scores are non-local: changing the alignment in one part affects the similarity in all other parts. Most structural aligners, such as the popular TM-align, Dali and CE[11,13,14], 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[15]. Structural alphabets thus reduce structure comparisons to much faster sequence alignments. Many ways to discretize the local amino acid backbone have been proposed[16]. Most, such as CLE, 3D-BLAST and Protein Blocks, discretize the conformations of short stretches of usually 3–5 C_α atoms[17–19].

For Foldseek, we developed a type of structural alphabet that does not describe the backbone but, rather, tertiary interactions. The 20 states of the 3D interaction (3Di) alphabet describe for each residue *i* the geometric conformation with its spatially closest residue *j*. 3Di has three key advantages over traditional backbone structural alphabets. (1) Weaker dependency between consecutive letters and (2) more evenly distributed state frequencies, both enhancing information density and reducing false positives (FPs) (Supplementary Table 1). (3) The highest information density is encoded in conserved protein

[1]Quantitative and Computational Biology Group, 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), Göttingen, Germany. [4]Artificial Intelligence Institute, Seoul National University, Seoul, South Korea. [5]Institute of Molecular Biology and Genetics, Seoul National University, Seoul, South Korea. [6]These authors contributed equally: Michel van Kempen, Stephanie S. Kim. ✉e-mail: soeding@mpinat.mpg.de; martin.steinegger@snu.ac.kr

https://doi.org/10.1038/s41587-023-01773-0

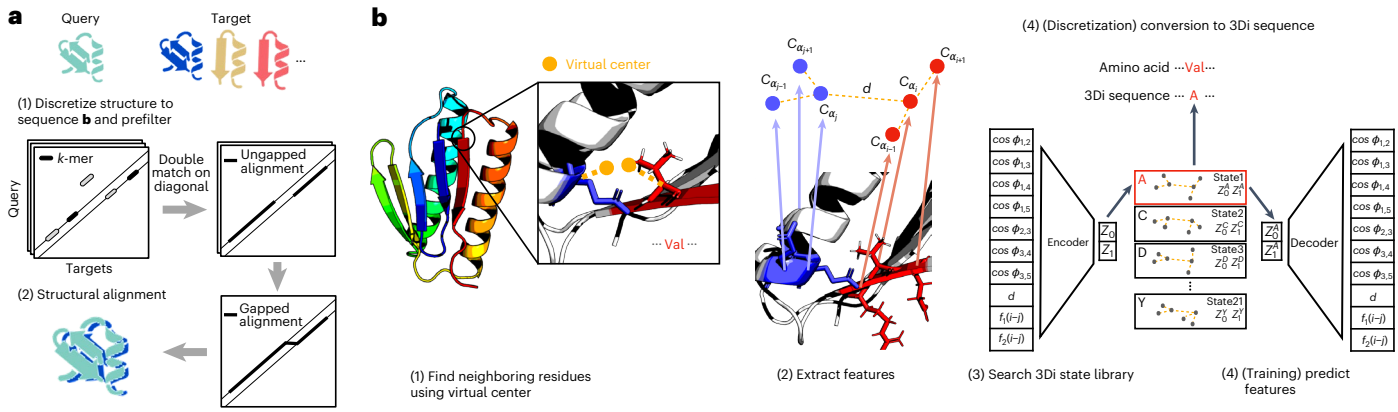

Fig. 1 | Foldseek workflow. a, Foldseek searches a set of query structures through a set of target structures. (1) Query and target structures are discretized into 3Di sequences (see **b**). To detect candidate structures, we apply the fast and sensitive *k*-mer and ungapped alignment prefilter of MMseqs2 to the 3Di sequences, (2) followed by vectorized Smith–Waterman local alignment combining 3Di and amino acid substitution scores. Alternatively, a global alignment is computed with a 1.7-times accelerated TM-align version (Supplementary Fig. 12). **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 positions (Supplementary Fig. 1) 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 and red). (3) These 10 features are used to define 20 3Di states by training a VQ-VAE[28] modified to learn states that are maximally evolutionary conserved. For structure searches, the encoder predicts the best-matching 3Di state for each residue.

cores and the lowest in non-conserved coil/loop regions, whereas the opposite is true for backbone structural alphabets.

Foldseek (https://foldseek.com/) (Fig. 1a) (1) discretizes the query structures into sequences over the 3Di alphabet and then uses a pre-trained 3Di substitution matrix (Supplementary Table 2) to search through the 3Di sequences of the target structures using the double-diagonal *k*-mer-based prefilter and gapless alignment prefilter modules from MMseqs2, our open-source sequence search software[6]. (2) High-scoring hits are aligned locally using 3Di (default) or globally with TM-align (Foldseek-TM). The local alignment stage combines 3Di and amino acid substitution scores. The construction of the 3Di alphabet is summarized in Fig. 1b and Supplementary Figs. 1–3.

To reduce high-scoring FPs and provide reliable *E* values, we subtracted the reversed query alignment score from the original score and applied a compositional bias correction within a local 40-residue sequence window (see the 'Pairwise local structural alignments' subsection). *E* values are calculated using an extreme-value score distribution, with parameters predicted by a neural network based on 3Di sequence composition and query length (see the '*E* values' subsection). Ranking of hits is determined by alignment bit score multiplied by the geometric mean of alignment TM-score and local distance difference test (LDDT). Foldseek also reports the probability for each match to be homologous, based on a fit of true and false matches on SCOPe.

We measured the sensitivity and speed of Foldseek, six protein structure alignment tools, an alignment-free structure search tool (Geometricus[20]) and a sequence search tool (MMseqs2 (ref. 6)) on the SCOPe dataset of manually classified single-domain structures[21]. Clustering SCOPe 2.01 at 40% sequence identity yielded 11,211 non-redundant protein sequences (SCOPe40). We performed an all-versus-all search and compared the tools' performance for finding members of the same SCOPe family, superfamily and fold (true-positive (TP) matches) by measuring for each query the fraction of TPs out of all possible correct matches until the first FP, a match to a different fold (see the 'SCOPe benchmark' subsection).

We first measured the sensitivity to detect relationships at family and superfamily level by the area under the curve (AUC) of the cumulative receiver operating characteristic (ROC) curve up to the first FP (Fig. 2a and Supplementary Fig. 4). Foldseek's sensitivity is below Dali and TM-align, higher than the structural aligner CE and much above the structural alphabet-based search tools 3D-BLAST and CLE-SW (Fig. 2a). In a precision-recall analysis, Foldseek-TM and Foldseek have the highest and third-highest area under the precision-recall curve on each of the three levels (Fig. 2b and Supplementary Fig. 4). Notably, Foldseek-TM improves over TM-align because its prefilter suppresses high-scoring FPs. Both sort hits by the average query and target length normalized TM-scores for best performance in the SCOPe benchmark.

Foldseek's performance is similar across all six secondary structure classes in SCOPe (Supplementary Fig. 5). On this small SCOPe40 benchmark set, Foldseek is more than 4,000 times faster than TM-align and Dali and over 21,000 times faster than CE (Fig. 2c). On the much larger AlphaFoldDB (version 1), where Foldseek approaches its full speed, it is around 184,600 and 23,000 times faster than Dali and TM-align, respectively (see below).

We devised a reference-free benchmark to assess search sensitivity and alignment quality of structural aligners (Fig. 2d) on a realistic set of full-length, multi-domain proteins. We clustered the AlphaFoldDB (version 1) to 34,270 structures using BLAST and SPICi[22]. We randomly selected 100 query structures from this set and aligned them against the remaining structures. TP matches are those with an LDDT score[23] of at least 0.6 and FPs below 0.25, ignoring matches in between. We set the LDDT thresholds according to the median inter-fold and intra-fold superfamily and family LDDT scores of SCOPe40 alignments (Supplementary Fig. 6). For other thresholds, see Supplementary Fig. 7. A domain-based sensitivity assessment would require a reference-based prediction of domains. To avoid it, we evaluated the sensitivity per residue. Figure 2d shows the distribution of the fraction of query residues that were part of alignments with at least *x* TP targets with better scores than the first FP match. Again, Foldseek has similar sensitivity as Dali, CE and TM-align and much higher sensitivity than CLE-SW and MMseqs2.

We analyzed the quality of alignments produced by the top five matches per query. We computed the alignment sensitivity as the number of TP residues divided by the query length and the precision as the number of TP residues divided by the alignment length. TP residues are those with residue-specific LDDT score above 0.6; FP residues are below 0.25; and residues with other scores are ignored. Figure 2e shows the average sensitivity versus precision of the 100 × 5 structure alignments.

https://doi.org/10.1038/s41587-023-01773-0

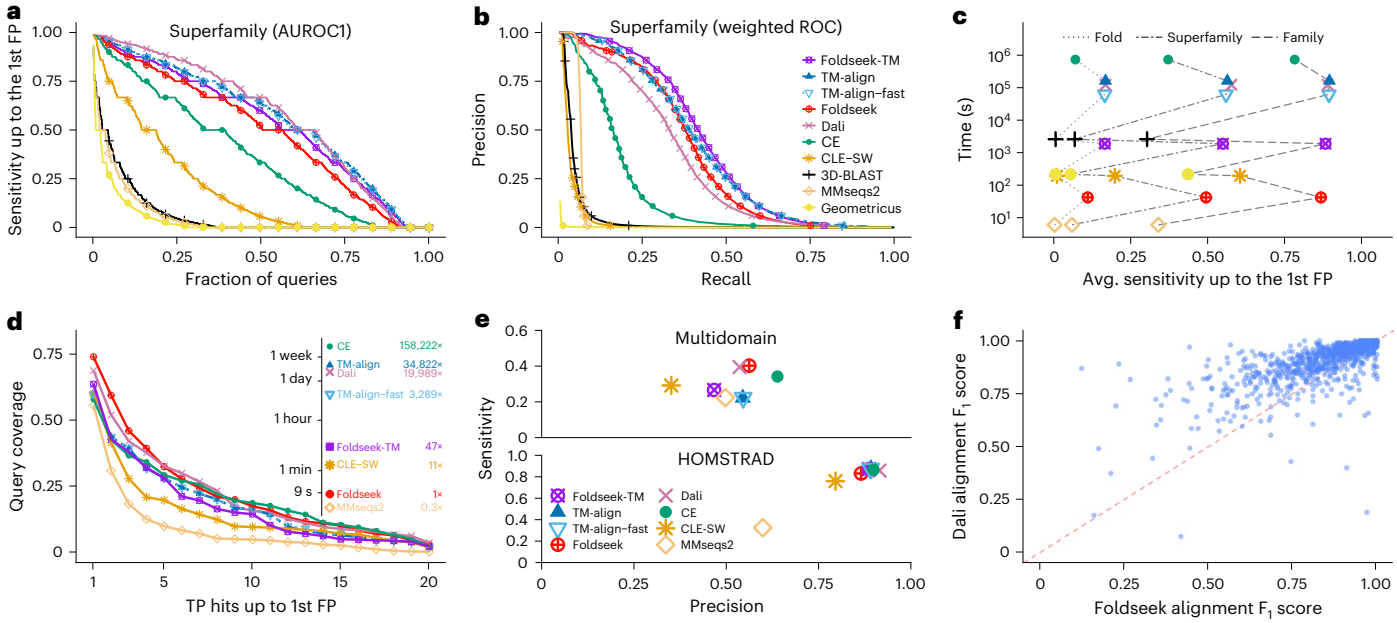

Fig. 2 | Foldseek reaches similar sensitivities as structural aligners at thousands of times their speed. a, Cumulative distributions of sensitivity for homology detection on the SCOPe40 database of single-domain structures. TPs are matches within the same superfamily; FPs are matches between different folds. Sensitivity is the area under the ROC (AUROC) curve up to the first FP (see Supplementary Fig. 4 for family and fold). **b**, Precision-recall curve of SCOPe40 superfamilies (see Supplementary Fig. 4 for family and fold). **c**, Average sensitivity up to the first FP for family, superfamily and fold versus total runtime on an AMD EPYC 7702P 64-core CPU for the all-versus-all searches of 11,211 structures of SCOPe40. **d**, Search sensitivity on multi-domain, full-length AlphaFold2 protein models. One hundred queries, randomly selected from AlphaFoldDB (version 1), were searched against this database. Per-residue query coverage (*y* axis) is the fraction of residues covered by at least *x* (*x* axis) TP matches ranked before the first FP match. **e**, Alignment quality for alignments of AlphaFoldDB (version 1) protein models (top panel), averaged over the top five matches of each of the 100 queries. Sensitivity = TP residues in alignment / query length; precision = TP residues / alignment length. Reference-based alignment quality benchmark on HOMSTRAD alignments. **f**, Alignment quality comparison between Foldseek and Dali for each HOMSTRAD family. The F_1 score is the harmonic mean between sensitivity and precision.

Foldseek alignments are more accurate and sensitive than MMseqs2, CLE-SW and TM-align, similarly accurate as Dali and 13% less precise but 15% more sensitive than CE. In the reference-based HOMSTRAD alignment quality benchmark[24], Foldseek performs slightly below CE, Dali and TM-align (Fig. 2e). Figure 2f shows the comparison between Foldseek and Dali in alignment quality for all HOMSTRAD families (see Supplementary Fig. 8 for example alignments).

To find potentially problematic high-scoring Foldseek FPs, we searched the set of unfragmented models in AlphaFoldDB (version 1) with average predicted LDDT[1]≥80 against itself. We inspected the 1,675 (of 133,813) high-scoring FPs (score per aligned column ≥ 1.0, TM-score < 0.5), revealing queries with multiple structured segments but with incorrect relative orientations (Supplementary Table 3 and Supplementary Fig. 9). The folded segments were correctly aligned by Foldseek. This illustrates that 3D aligners such as TM-align may overlook homologous structures that are not globally superposable, whereas Foldseek (as well as the two-dimensional (2D) aligner Dali) is independent of relative domain orientations and excels at detecting homologous multi-domain structures[12].

We developed a webserver (https://search.foldseek.com) for multi-database searches, including AlphaFoldDB (version 4: Proteomes and Swiss-Prot), AlphaFoldDB (version 4) and CATH[25] clustered at 50% sequence identity, ESM Atlas-HQ and Protein Data Bank (PDB)[26].

We compared Foldseek webserver, TM-align and Dali using SARS-CoV-2 RdRp (PDB: 6M71, chain A (ref. 27); 942 residues) in AlphaFoldDB (version 1). Search times were 10 d for Dali, 33 h for TM-align and 6 s for Foldseek, making it 180,000 and 23,000 times faster. All top 10 hits were known RdRp homologs (Supplementary Table 4).

The availability of high-quality structures for nearly every folded protein is transformative for biology and bioinformatics. Sequence-based analyses will soon be largely superseded by structure-based analyses. The main limitation in our view—the four orders of magnitude slower speed of structure comparisons—is removed by Foldseek.

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/s41587-023-01773-0.

References

1. Jumper, J. et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
2. Baek, M. et al. Accurate prediction of protein structures and interactions using a three-track neural network. *Science* **373**, 871–876 (2021).
3. Varadi, M. et al. AlphaFold Protein Structure Database: massively expanding the structural coverage of protein–sequence space with high-accuracy models. *Nucleic Acids Res.* **50**, D439–D444 (2022).
4. Lin, Z. et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
5. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).
6. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nat. Biotechnol.* **35**, 1026–1028 (2017).

7. Steinegger, M. et al. HH-suite3 for fast remote homology detection and deep protein annotation. *BMC Bioinformatics* **20**, 473 (2019).
8. Buchfink, B., Reuter, K. & Drost, H.-G. Sensitive protein alignments at tree-of-life scale using DIAMOND. *Nat. Methods* **18**, 366–368 (2021).
9. Mahlich, Y., Steinegger, M., Rost, B. & Bromberg, Y. HFSP: high speed homology-driven function annotation of proteins. *Bioinformatics* **34**, i304–i312 (2018).
10. Illergård, K., Ardell, D. H. & Elofsson, A. Structure is three to ten times more conserved than sequence—a study of structural response in protein cores. *Proteins* **77**, 499–508 (2009).
11. Zhang, Y. & Skolnick, J. TM-align: a protein structure alignment algorithm based on the TM-score. *Nucleic Acids Res.* **33**, 2302–2309 (2005).
12. Hasegawa, H. & Holm, L. Advances and pitfalls of protein structural alignment. *Curr. Opin. Struct. Biol.* **19**, 341–348 (2009).
13. Holm, L. Using Dali for protein structure comparison. *Methods Mol. Biol.* **2112**, 29–42 (2020).
14. Shindyalov, I. N. & Bourne, P. E. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. *Protein Eng.* **11**, 739–747 (1998).
15. Guyon, F., Camproux, A.-C., Hochez, J. & Tuffery, P. SA-Search: a web tool for protein structure mining based on a structural alphabet. *Nucleic Acids Res.* **32**, W545–W548 (2004).
16. Ma, J. & Wang, S. Algorithms, applications, and challenges of protein structure alignment. *Adv. Protein Chem. Struct. Biol.* **94**, 121–175 (2014).
17. Wang, S. & Zheng, W.-M. CLePAPS: fast pair alignment of protein structures based on conformational letters. *J. Bioinform. Comput. Biol.* **6**, 347–366 (2008).
18. Yang, J.-M. & Tung, C.-H. Protein structure database search and evolutionary classification. *Nucleic Acids Res.* **34**, 3646–3659 (2006).
19. de Brevern, A. G., Etchebest, C. & Hazout, S. Bayesian probabilistic approach for predicting backbone structures in terms of protein blocks. *Proteins* **41**, 271–287 (2000).
20. Durairaj, J., Akdel, M., de Ridder, D. & van Dijk, A. D. J. Geometricus represents protein structures as shape-mers derived from moment invariants. *Bioinformatics* **36**, i718–i725 (2020).
21. 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).
22. Jiang, P. & Singh, M. SPICi: a fast clustering algorithm for large biological networks. *Bioinformatics* **26**, 1105–1111 (2010).
23. 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).
24. Mizuguchi, K., Deane, C. M., Blundell, T. L. & Overington, J. P. HOMSTRAD: a database of protein structure alignments for homologous families. *Protein Sci.* **7**, 2469–2471 (1998).
25. Bordin, N. et al. AlphaFold2 reveals commonalities and novelties in protein structure space for 21 model organisms. *Commun. Biol.* **6**, 160 (2023).
26. Burley, S. K. et al. RCSB Protein Data Bank: powerful new tools for exploring 3D structures of biological macromolecules for basic and applied research and education in fundamental biology, biomedicine, biotechnology, bioengineering and energy sciences. *Nucleic Acids Res.* **49**, D437–D451 (2021).
27. Gao, Y. et al. Structure of the RNA-dependent RNA polymerase from COVID-19 virus. *Science* **368**, 779–782 (2020).
28. Van den Oord, A., Vinyals, O. & Kavukcuoglu, K. Neural discrete representation learning. *Proc. of the 31st Conference on Neural Information Processing Systems*. https://proceedings.neurips.cc/paper_files/paper/2017/file/7a98af17e63a0ac09ce2e96d03992fbc-Paper.pdf (NIPS, 2017).

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 using single instruction, multiple data (SIMD) vector units. Owing to the SIMDe library (https://github.com/simd-everywhere/simde), Foldseek runs on a wide range of CPU architectures (x86_64, arm64 and ppc64le) and operating systems (Linux and macOS). The core modules of Foldseek, which build on the MMseqs2 framework[6], are described in the following paragraphs.

Create database

The `createdb` module converts a set of PDB (ref. 29), macromolecular crystallographic information file (mmCIF) formatted files or Foldcomp compressed structure (FCZ (ref. 30)) files into an internal Foldseek database format using the Gemmi package (https://gemmi.readthedocs.io/en/latest/) or the Foldcomp library. 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 backbone atom and C_β coordinates (see the 'Descriptors for 3Di structural alphabet' and 'Optimize nearest-neighbor selection' subsections). Backbone atom and C_β coordinates are needed only for the nearest-neighbor selection. For C_α-only structures, Foldseek reconstructs backbone atom coordinates using PULCHRA[31]. Missing C_β coordinates (for example, in glycines) are defined such that the four groups attached to the C_α are arranged at the vertices of a regular tetrahedron. The 3Di and amino acid sequences and the C_α coordinates are stored in the Foldseek database. To save disk space, we optionally compress the C_α coordinates losslessly, beginning with three uncompressed 4-byte floating-point C_α coordinates and storing all subsequent coordinates as 2-byte signed integer differences[32]. If any difference is too large to be represented with a 2-byte signed integer, we fall back to 4-byte floats for all C_α coordinates.

Prefilter

The `prefilter` module detects double matches of similar, spaced words (*k*-mers) that occur on the same diagonal. The *k*-mer size is dynamically set to $k = 6$ or $k = 7$ depending on the size of the target database. Similar *k*-mers are those with a 3Di substitution matrix score above a certain threshold, whereas MMseqs2 uses an amino acid substitution matrix to compute the similarity (see the '3Di substitution score matrix' subsection). The gapless double-match 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 avoid FP matches due to regions with biased 3Di sequence composition, a compositional bias correction is applied in a way analogous to MMseqs2 (ref. 33). 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 on to the next stage. We implemented an optional taxonomy filter within the prefiltering step to help users search through taxonomic subsets of the target database. After the gapless double-diagonal matching stage and before the ungapped alignment stage, we reject all potential target hits that do not lie within a taxonomic clade specified by the user.

Pairwise local structural alignments

After the prefilter has removed the vast majority of non-homologous sequences, the `structurealign` module computes pairwise alignments for the remaining sequences using an SIMD-accelerated Smith–Waterman algorithm[34,35]. 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. We optimized these two weights and the ratio of gap extend to gap open penalty on ~1% of alignments (all-versus-all on 10% of randomly selected SCOPe40 domains). A compositional bias correction is applied to the amino acid and 3Di scores. To further suppress high-scoring FP matches, for each match we align the reversed query sequence against the target and subtract the reverse bit score from the forward bit score.

Structural bit score

We rank hits by a 'structural bit' score—that is, the product of the bit score produce by the Smith–Waterman algorithm and the geometric mean of average alignment LDDT and the alignment TM-score.

Fast alignment LDDT computation

To improve the LDDT score computation speed, we store the 3D coordinates of the query in a grid using spatial hashing. Each grid cell spans 15 Å, which is the default radius considered for the LDDT computation. For each aligned query residue *i*, we compute the distances to all C_α atoms within a 15 Å radius by searching all neighboring grid cells of the query residue's grid cell. For each residue *j*, we compute the distance between the C_α atoms of *i* and *j* and the distance of the corresponding aligned target residues. Query and target distances for the aligned pairs are subtracted, and the differences *d* are transformed into LDDT scores $s = 0.25 \times ((d < 0.5) + (d < 1.0) + (d < 2.0) + (d < 4.0))$. For each *i*, we obtain the means of the scores for all Cα atoms *j* within the 15 Å radius of *i*. The LDDT score is the mean of these means over all query residues *i*.

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. The module `computemulambda` 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`)[36]. To train the neural network, it is critical to use query and target proteins that include problematic regions, such as structurally biased, disordered or badly modeled regions that occur ubiquitously in full-length proteins or modeled structures. We, therefore, trained the network on 100,000 structures sampled from the AlphaFoldDB (version 1). We trained a neural network to predict μ and λ from the amino acid composition of the query and its length (so a scrambled version of the query sequence would produce the same μ and λ). The network has 22 input nodes, two 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 FPs 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 (Supplementary Fig. 10; see Supplementary Fig. 11 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-bladed 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[37].

Probability of TP match

Foldseek computes for each match a simple estimate for the probability that the match is a TP match given its structural bit score. Here, hits within the same superfamily are TP; hits to another fold are FP; and hits to the same family or to another superfamily are ignored. We estimate the structural bit score distributions of TP and FP hits ($p(score|TP)$ and $p(score|FP)$), which allow us to calculate the probability of a TP $p(TP|score) = \frac{p(\text{score}|TP)\,p(TP)}{p(\text{score}|TP)\,p(TP)+p(\text{score}|FP)\,p(FP)}$. Both score distributions were fitted on SCOPe40 with a mixture model consisting of two gamma distributions (resulting in five parameters for each function). For the fitting, the function `gammamixEM` from the R package `mixtools`[38] was used. We excluded cross-fold hits between certain folds as in the *E* value estimation. For example, Foldseek finds around the same number of FPs and TPs with a score of 51 in SCOPe40. The probability for a hit with score 51 is, therefore, 50%.

Pairwise global structural alignments using TM-align

We also offer the option to use TM-align for pairwise structure alignment instead of the 3Di-based alignment. 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 (for example, score secondary structure, Euclidean distance of superposed structures or fragments). TM-align's NW code did not take advantage of SIMD instructions; therefore, we replaced it by parasail's[39] 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[40]. Our NW implementation does not compute exactly the same alignment because we apply affine gap costs, whereas TM-align does not (Supplementary Fig. 12). (2) TM-align requires 17 bytes × query length × target length of memory, and we reduce the constant overhead from 17 bytes to 4 bytes. If Foldseek is used in TM-align mode (parameter `--alignment-type 1`), TM-align is used for the alignment stage after the prefilter step, where we replace the reported *E* value column with TM-scores normalized by the query length. The results are ordered in descending order by average TM-score by default.

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 (Supplementary Fig. 3). We chose the alphabet size $A = 20$ as a tradeoff between encoding as much information as possible (large *A*; Supplementary Fig. 13) and limiting the number of similar 3Di *k*-mers that we need to generate in the *k*-mer-based prefilter, which scales with A^k. The discrete single-letter states are formed from neighborhood descriptors containing 10 features encoding the conformation of backbones around residues *i* and *j* represented by the C_α atoms ($C_{\alpha,i-1}$, $C_{\alpha,i}$, $C_{\alpha,i+1}$) and ($C_{\alpha,j-1}$, $C_{\alpha,j}$, $C_{\alpha,j+1}$). The descriptors use the five unit vectors along the following directions:

$$\mathbf{u_1}: C_{\alpha,i-1} \to C_{\alpha,i} \quad \mathbf{u_4}: C_{\alpha,j} \to C_{\alpha,j+1}$$
$$\mathbf{u_2}: C_{\alpha,i} \to C_{\alpha,i+1} \quad \mathbf{u_5}: C_{\alpha,i} \to C_{\alpha,j}$$
$$\mathbf{u_3}: C_{\alpha,j-1} \to C_{\alpha,j}.$$

We define the angle between u_k and u_l as ϕ_{kl}, so $\cos\phi_{kl} = \mathbf{u_k}^T\mathbf{u_l}$. 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 $|C_{\alpha,i} - C_{\alpha,j}|$ describe the conformation between the backbone fragments. In addition, we encode the sequence distance with the two features $\text{sign}(i-j)\ \min(|i-j|, 4)$ and $\text{sign}(i-j)\ \log(|i-j|+1)$.

Learning the 3Di states using a vector quantized variational autoencoder

The 10-dimensional descriptors were discretized into an alphabet of 20 states using a vector quantized variational autoencoder (VQ-VAE)[28]. In contrast to standard clustering approaches such as *k*-means, VQ-VAE is a nonlinear approach that can optimize decision surfaces for each of its states. 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 y_n from 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 x_n into a 2D 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 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 $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}, \boldsymbol{I\sigma^2})$ (with diagonal covariance).

We trained the VQ-VAE on the loss function defined in Equation (3) in ref. 28 (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 four epochs. Using Kerasify (https://github.com/moof2k/kerasify), we integrated the encoder network into Foldseek. The domains from SCOPe40 were split 80%/20% by fold into training and validation sets. For the training, we aligned the structures with TM-align, 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 TM-align 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 'VQ-VAE training'. First, we determined the 3Di states of all residues. Next, the substitution frequencies among 3Di states were calculated by counting how often two 3Di states were structurally aligned. (Note that the substitution frequencies from state A to state B and the opposite direction are equal.) Finally, the score $S(x,y) = 2\log_2\frac{p(x,y)}{p(x)p(y)}$ 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.

3Di alphabet cross-validation

We trained the 3Di alphabet (the VQ-VAE weights) and the substitution matrix by four-fold cross-validation on SCOPe40. We split the SCOPe40 dataset into four parts, such that all domains of each fold ended up in the same part of the four parts. 3Di alphabets were trained on three parts and tested on the remaining part, selecting each of the four parts in turn as a test set. The 80:20 split between training and validation sets to select the best alphabet out of the 100 VQ-VAE runs happens within the 3/4 of the cross-validation training data. Supplementary Fig. 14 shows the mean sensitivity (black) and the standard deviation (gray

area) in comparison to the final 3Di alphabet, for which we trained the 3Di alphabet on the entire SCOPe40 (red). No overfitting was observed, despite training 492 parameters (282 neural network and 210 substitution matrix entries). In Fig. 2, we, therefore, show the benchmark results for the final 3Di alphabet, trained on the entire SCOPe40.

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 ($|V - C_\alpha|$) (Supplementary Fig. 1). 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 $l \in \{1.53\text{ Å } k\colon k \in \{1, 1.5, 2, 2.5, 3\}\}$ (1.53 Å 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 SCOPe benchmark.

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 the SCOPe40 structures (available at https://wwwuser.gwdg.de/~compbiol/foldseek/scop40pdb.tar.gz).

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 first FP. For family-level, superfamily-level 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.

Evaluation SCOPe benchmark

After sorting the alignment result of each query (described in the 'Tools and options for benchmark comparison' subsection), we calculated the sensitivity as the fraction of TPs in the sorted list up to the first FP, all excluding self-hits. For comparison, we took the mean sensitivity over all queries for family-level, superfamily-level and fold-level classifications. We evaluated only SCOPe members with at least one other family, superfamily and fold member. We measure the sensitivity up to the 1st FP (ROC1) instead, for example, up to the 5th FP, because ROC1 better reflects the requirements for low false discovery rates in automatic searches.

Additionally, we plotted precision-recall curves for each tool (Fig. 2b and Supplementary Fig. 4). After sorting the alignment results by the structural similarity scores (as described in the 'Tools and options for benchmark comparison' subsection) and excluding self-matches, we generated a weighted precision-recall curve for family-level, superfamily-level and fold-level classifications (precision = TP / (TP + FP) and recall = TP / (TP + FN)). All counts (TP, FP and FN) were weighted by the reciprocal of their family, superfamily or fold size. In this way, folds, superfamilies and families contribute linearly with their size instead of quadratically[36].

Runtime evaluations on SCOPe and AlphaFoldDB

We measured the speed of structural aligners on a server with an AMD EPYC 7702P 64-core CPU and 1,024 GB RAM memory. On SCOPe40, we measured or estimated the runtime for an all-versus-all comparison. To avoid excessive runtimes for TM-align, Dali and CE, we estimated the runtime by randomly selecting 10% of the 11,211 SCOPe domains as queries. We measured runtimes on AlphaFoldDB for searches with the same 100 randomly selected queries used for the sensitivity and alignment quality benchmarks (Fig. 2d,e). 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.

Reference-free multi-domain benchmarks

We devised two reference-free benchmarks that do not rely on any reference structural alignments. We clustered the AlphaFoldDB (version 1)[3] using SPICi[22]. For this, we first aligned all protein sequences all against all using an E value threshold $<10^{-3}$ using BLAST (2.5.0+)[5]. SPICi produced 34,270 clusters from the search result. For each cluster, we picked the longest protein as representative. We randomly selected 100 representatives as queries and searched the set of remaining structures. The top five alignments of all queries are listed at https://wwwuser.gwdg.de/~compbiol/foldseek/multi_domain_top5_alignments/.

For the evaluation, we needed to adjust the LDDT score function taken from AlphaFold2 (ref. 1). LDDT calculates for each residue i in the query the fraction of residues in the 15 Å neighborhood that have a distance within 0.5, 1, 2 or 4 Å of the distance between the corresponding residues in the target[23]. The denominator of the fraction is the number of 15 Å neighbors of i that are aligned to some residue in the target. This does not properly penalize non-compact models in which each residue has few neighbors within 15 Å. We, therefore, use as denominator the *total* number of neighboring residues within 15 Å of i.

For the alignment quality benchmark (Fig. 2e), we classified each aligned residue pair as TP or FP depending on its residue-wise LDDT score—that is, the fraction of distances to its 15 Å neighbors that are within 0.5, 1, 2 and 4 Å of the distance to the corresponding residues in the query, averaged over the four distance thresholds. TP residues are those with a residue-wise LDDT score of at least 0.6 and FPs below 0.25, ignoring matches in between. For the search sensitivity benchmark (Fig. 2d), TP residue–residue matches are those with an LDDT score of the query-target alignment of at least 0.6 and FPs below 0.25, ignoring matches in between. (The LDDT score of the query-target alignment is the average of the residue-wise LDDT score over all aligned residue pairs.) The choice of thresholds is illustrated in Supplementary Fig. 6. The benchmark for other thresholds is shown in Supplementary Fig. 7.

All-versus-all search of AlphaFoldDB with Foldseek

We downloaded the AlphaFoldDB (version 1)[3] 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 predicted LDDT (pLDDT) greater than or equal to 80 and (2) models of non-fragmented domains. We also computed the structural similarity for each pair using TM-align (default options).

Tools and options for benchmark comparison

Owing to dataset overlap, we excluded methods from the benchmark that are likely to be overfitted on SCOPe. This applies to methods that trained many thousands of parameters (for example, deep neural networks) with strong data leakage among training, validation and test sets. For example, several tools allowed up to 40% sequence identity between sets. The following command lines were used in the SCOPe as well as the multi-domain benchmark:

Foldseek

We used Foldseek commit `aeb5e` during this analysis. Foldseek was run with the following parameters: `--threads 64 -s 9.5 -e 10 --max-seqs 2000.`

Foldseek-TM

For the Foldseek-TM benchmark, we first run a regular 3Di/AA-based Foldseek search using the following parameters: `--threads 64 -s 9.5 -e 10 --max-seqs 4000 --alignment-mode 1`. All hits

passing are then aligned by Foldseeks's `tmalign --tmalign-fast 1 --tmscore-threshold 0.0 -a`. We used Foldseek commit `aeb5e` during this analysis. We expose Foldseek-TM in our command-line interface as a search mode that combines regular Foldseek 3Di/AA-based workflow with our TM-align implementation within the `tmalign` module.

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 version 4.80 (https://github.com/realbigws/PDB_Tool) to convert the benchmark structure set to CLE sequences. After the conversion, we used SSW[35] (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`); and (5) returning alignment (`-c`). The gap open and extend values were inferred from DeepAlign[41]. 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[34] (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) 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; (3) returning alignments (`-c`); (4) 3D-BLAST's optimized substitution matrix (`-a 3DBLAST`); and (5) protein alignment mode (`-p`): `ssw_test -o 8 -e 2 -c -a 3DBLAST -p`. We noticed that the 3D-BLAST matrix with Smith–Waterman resulted in a similar performance to CLE: 0.717, 0.230 and 0.011 for family classification, superfamily classification and fold classification, respectively. We excluded 3D-BLAST's measurement from the multi-domain benchmark because it produced occasionally high scores (>10^7) for single residue alignments.

TM-align

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. TM-align reports two TM-scores: (1) normalized by the length of 1st chain (query) or (2) normalized by the length of the 2nd chain (target). We used the average of TM-scores normalized by the 1st chain (query) and 2nd chain (target) in all our analyses. We evaluated TM-align's performance by sorting the results based on both the query TM-score and the minimum, maximum and average TM-score for both the query and target. Our results showed that the average TM-score performed the best in our single-domain benchmark.

Default: `TMalign query.pdb target.pdb`

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

Dali

We installed the standalone DaliLite.v5. For the SCOPe40 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 and 34,252 structures out of 34,270 SPICi clusters. 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 in descending order:

```
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[42] (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 with unlimited gap size (`maxGapSize -1`) to improve alignment results[14]. The results were sorted by *z*-score in descending order. For the multi-domain benchmark, we excluded one query that was running over 16 d. The Jar file of our implementation of CE calculation is provided (see 'Code availability').

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

Geometricus

We included Geometricus[20] in the SCOPe benchmark as a representative of alignment-free tools, which are fast but can find only globally similar structures. Geometricus discretizes fixed-length backbone fragments (shape-mers) using their 3D moment invariants and represents structures as a fixed-length count vector over the shape-mers. To calculate the shape-mer-based structural similarity of the benchmark set, we used Caretta-shape's Python implementation (1e3adb0) of multiple structure alignment (https://github.com/TurtleTools/caretta/caretta/multiple_alignment.py), which computes the Bray–Curtis similarity between the Geometricus shape-mer vectors. Our modified version extracts structural information from the input files and generates all-versus-all pairwise structural similarity score as an output. We ran Geometricus on a single core because it would require substantial engineering efforts to implement parallelization on multiple cores. With an efficient multi-core implementation, Geometricus might be as fast as MMseqs2 on 64 cores. The Python code of our implementation of Geometricus is provided:

```
python runGeometricus_caretta.py -i querylist.txt
-o OutputDirectory
```

HOMSTRAD alignment benchmark

The HOMSTRAD database contains expert-curated homologous structural alignments for 1,032 protein families[24]. We downloaded the latest HOMSTRAD version (https://mizuguchilab.org/homstrad/data/homstrad_with_PDB_2022_Aug_1.tar.gz) and picked the pairwise alignments between the first and last members of each family, which resulted in structures of a median length of 182 residues. We used the same parameters as in the SCOPe and multi-domain benchmark. We forced Foldseek, MMseqs2 and CLE-Smith–Waterman to return an alignment by switching off the prefilter and *E* value threshold. With the HOMSTRAD alignments as reference, we measured for each pairwise alignment the sensitivity (fraction of residue pairs of the HOMSTRAD alignment that were correctly aligned) and the precision (fraction of correctly aligned residue pairs in the predicted alignment). Dali, CE

and CLE-Smith–Waterman failed to produce an alignment for 35, 1 and 1 out of 1,032 pairs, respectively, which were rated with a sensitivity of 0. The mean sensitivity and precision are shown in Fig. 2e, and all individual alignments are listed in `homstrad_alignments.txt` at https://wwwuser.gwdg.de/~compbiol/foldseek/.

Limitations of benchmarks

The SCOPe benchmark to measure search sensitivity uses only single-domain proteins as queries and targets (Fig. 2a–c). It, therefore, cannot assess the ability of tools to find local similarities—for example, finding homologous domains shared between two multi-domain proteins. The alignment benchmark based on HOMSTRAD (Fig. 2e) has the same limitation, as the vast majority of proteins in HOMSTRAD have a single domain (median length = 182 residues). A drawback of our reference-free multi-domain benchmark is the need to choose thresholds for TPs and FPs (Supplementary Fig. 6).

Pre-built and ready-to-download databases

Foldseek includes the `databases` module to aid users with the download and setup of structural databases. Currently, we include the four variants of the AlphaFoldDB (version 4): UniProt (214 million structures), UniProt50, a clustered database to 50% sequence identity and 90% bi-directional coverage using MMseqs2 (parameters `-c 0.9 --min-seq-id 0.5 --cluster-reassign`; 54 million structures), Proteome (564,000 structures) and Swiss-Prot (542,000 structures). Additionally, we regularly build and offer a 100% sequence identity clustered PDB. The update pipeline is available in the `util/update_webserver_pdb` folder in the main Foldseek repository. These databases are hosted on Cloudflare R2 for fast downloading. We additionally link to and provide an automatic setup procedure for the ESM Atlas High-Quality Clu30[4] database.

Webserver

The Foldseek webserver is based on the MMseqs2 webserver[43]. To allow for searches in seconds, we implemented MMseqs2's pre-computed database indexing capabilities in Foldseek. Using these, the search databases can be fully cached 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 AlpaFoldDB/UniProt50 (version 4), AlphaFoldDB/Proteome (version 4), AlphaFoldDB/Swiss-Prot (version 4), CATH50, ESM Atlas High-Quality Clu30 and PDB100 require 191 GB, 3.8 GB, 3.4 GB, 1.4 GB, 110 GB and 2.0 GB RAM, respectively. The databases are kept in memory using `vmtouch` (https://github.com/hoytech/vmtouch). Databases are only required to remain resident in RAM if Foldseek is used as a webserver. During batch searches, Foldseek adapts its memory use to the available RAM of the machine. We implemented a structural visualization using the NGL viewer[44] to aid the investigation of pairwise hits. Because we only store C_α traces of the database proteins, we use PULCHRA[30] to complete the backbone of these sequences, and also of the query if necessary, to enable a ribbon visualization[45] of the proteins. For a high-quality superposition, we use TM-align[11] to superpose the structures based on the Foldseek alignment. Both PULCHRA and TM-align are executed within the users' browser using WebAssembly. They are available as `pulchra-wasm` and `tmalign-wasm` on the npm package repository as free open-source software.

Structure prediction in the webserver

We use the ESM Atlas API to predict structures of user-supplied sequences that are, at most, 400 residues long. This enables sequence-to-structure searches in the webserver.

Reporting summary

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

Data availability

Benchmark data are available at https://wwwuser.gwdg.de/~compbiol/foldseek.

Code availability

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

References

29. Burley, S. K. et al. Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.* **47**, D520–D528 (2019).
30. Kim, H., Mirdita, M. & Steinegger, M. Foldcomp: a library and format for compressing and indexing large protein structure sets. *Bioinformatics* **39**, btad153 (2023).
31. Rotkiewicz, P. & Skolnick, J. Fast procedure for reconstruction of full-atom protein models from reduced representations. *J. Comput. Chem.* **29**, 1460–1465 (2008).
32. Valasatava, Y. et al. Towards an efficient compression of 3D coordinates of macromolecular structures. *PLoS ONE* **12**, e0174846 (2017).
33. Hauser, M., Steinegger, M. & Söding, J. MMseqs software suite for fast and deep clustering and searching of large protein sequence sets. *Bioinformatics* **32**, 1323–1330 (2016).
34. Farrar, M. Striped Smith–Waterman speeds database searches six times over other SIMD implementations. *Bioinformatics* **23**, 156–161 (2007).
35. Zhao, M., Lee, W.-P., Garrison, E. P. & Marth, G. T. SSW library: an SIMD Smith–Waterman C/C++ library for use in genomic applications. *PLoS ONE* **8**, e82138 (2013).
36. Eddy, S. R. Accelerated profile HMM searches. *PLoS Comput. Biol.* **7**, e1002195 (2011).
37. Söding, J. & Remmert, M. Protein sequence comparison and fold recognition: progress and good-practice benchmarking. *Curr. Opin. Struct. Biol.* **21**, 404–411 (2011).
38. Benaglia, T., Chauveau, D., Hunter, D. R. & Young, D. mixtools: an R package for analyzing finite mixture models. *J. Stat. Softw.* **32**, 1–29 (2009).
39. Daily, J. Parasail: SIMD C library for global, semi-global, and local pairwise sequence alignments. *BMC Bioinformatics* **17**, 81 (2016).
40. Hung, L.-H. & Samudrala, R. fast_protein_cluster: parallel and optimized clustering of large-scale protein modeling data. *Bioinformatics* **30**, 1774–1776 (2014).
41. Jiménez-Moreno, A., Strelák, D., Filipovic, J., Carazo, J. M. & Sorzano, C. O. S. DeepAlign, a 3D alignment method based on regionalized deep learning for Cryo-EM. *J. Struct. Biol.* **213**, 107712 (2021).
42. Lafita, A. et al. BioJava 5: a community driven open-source bioinformatics library. *PLoS Comput. Biol.* **15**, e1006791 (2019).
43. Mirdita, M., Steinegger, M. & Söding, J. MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics* **35**, 2856–2858 (2019).
44. Rose, A. S. et al. NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics* **34**, 3755–3758 (2018).
45. Richardson, J. S. Early ribbon drawings of proteins. *Nat. Struct. Biol.* **7**, 624–625 (2000).

Acknowledgements

We thank N. Bordin, I. Sillitoe and C. Orengo for reporting issues and providing valuable feedback; Y. Zhang, P. Rotkiewicz and M. Wojdyr for making TM-align, PULCHRA and the Gemmi library freely accessible; and D.-Y. Kim for creating the Foldseek logo.

M.S. acknowledges support from the National Research Foundation of Korea (NRF) (grants 2019R1A6A1A10073437, 2020M3A9G7103933, 2021R1C1C102065 and 2021M3A9I4021220), the Samsung DS Research Fund and the Creative-Pioneering Researchers Program through Seoul National University. S.K. acknowledges support by NRF grant 2019R1A6A1A10073437. J.S. acknowledges support by the German Ministry for Education and Research (horizontal4meta). We used the compute cluster at the Gesellschaft für wissenschaftliche Datenverarbeitung mbH Göttingen (GWDG).

Author contributions

M.K., S.K., J.S. and M.S. designed the research. M.K., S.K., C.T., M.M. and M.S. developed code and performed analyses. M.K. and J.S. developed the 3Di alphabet. J.L. implemented the fast LDDT code. M.M. and C.L.M.G. developed the webserver. M.K., S.K., C.T., M.M., J.S. and M.S. wrote the manuscript.

Funding

Open access funding provided by Max Planck Society.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41587-023-01773-0.

Correspondence and requests for materials should be addressed to Johannes Söding or Martin Steinegger.

Peer review information *Nature Biotechnology* thanks the anonymous reviewers for their contribution to the peer review of this work.

nature research

Corresponding author(s): Johannes Söding, Martin Steinegger

Last updated by author(s): March 23, 2023

Reporting Summary

Nature Research 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 Research 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	No data collection was performed as the data was downloaded from publicly available databases (SCOPe, AlphafoldDB and HOMSTRAD).
Data analysis	Foldseek (commit aeb5e), MMseqs (realease 13-45111), CLE-SW (PDB Tool v4.80), 3D-BLAST (beta102, with BLAST+ 2.2.26), BLAST (2.5.0+), SPICi, TM-align (2019/08/22), Dali (DaliLite.v5), CE (BioJava 5.4.0), Geometricus (https://github.com/TurtleTools/caretta, commit 1e3adb0) The analysis scripts are publicly available at https://github.com/steineggerlab/foldseek-analysis.

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 Research 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 list of figures that have associated raw data
- A description of any restrictions on data availability

Used datasets:
https://scop.berkeley.edu/downloads/ver=2.01
https://ftp.ebi.ac.uk/pub/databases/alphafold/v1/
https://mizuguchilab.org/homstrad/data/ (version from 1. Aug 2022)

Foldseek, analysis scripts, Foldseek server:

https://github.com/steineggerlab/foldseek
https://github.com/steineggerlab/foldseek-analysis
https://github.com/soedinglab/mmseqs2-app

Benchmark data:
https://wwwuser.gwdg.de/~compbiol/foldseek

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	No samples were collected.
Data exclusions	No data was excluded from the public data sets.
Replication	We performed our analyses on public data sets (AlphaFoldDB, SCOPe, HOMSTRAD). The analysis scripts were published to ensure reproducibility.
Randomization	No randomization was performed as this was not relevant to this study
Blinding	No blinding was performed as this was not relevant to this study.

Behavioural & social sciences study design

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

Study description	*Briefly describe the study type including whether data are quantitative, qualitative, or mixed-methods (e.g. qualitative cross-sectional, quantitative experimental, mixed-methods case study).*
Research sample	*State the research sample (e.g. Harvard university undergraduates, villagers in rural India) and provide relevant demographic information (e.g. age, sex) and indicate whether the sample is representative. Provide a rationale for the study sample chosen. For studies involving existing datasets, please describe the dataset and source.*
Sampling strategy	*Describe the sampling procedure (e.g. random, snowball, stratified, convenience). Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient. For qualitative data, please indicate whether data saturation was considered, and what criteria were used to decide that no further sampling was needed.*
Data collection	*Provide details about the data collection procedure, including the instruments or devices used to record the data (e.g. pen and paper, computer, eye tracker, video or audio equipment) whether anyone was present besides the participant(s) and the researcher, and whether the researcher was blind to experimental condition and/or the study hypothesis during data collection.*
Timing	*Indicate the start and stop dates of data collection. If there is a gap between collection periods, state the dates for each sample cohort.*
Data exclusions	*If no data were excluded from the analyses, state so OR if data were excluded, provide the exact number of exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.*
Non-participation	*State how many participants dropped out/declined participation and the reason(s) given OR provide response rate OR state that no participants dropped out/declined participation.*
Randomization	*If participants were not allocated into experimental groups, state so OR describe how participants were allocated to groups, and if allocation was not random, describe how covariates were controlled.*

Ecological, evolutionary & environmental sciences study design

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

Study description	*Briefly describe the study. For quantitative data include treatment factors and interactions, design structure (e.g. factorial, nested, hierarchical), nature and number of experimental units and replicates.*
Research sample	*Describe the research sample (e.g. a group of tagged Passer domesticus, all Stenocereus thurberi within Organ Pipe Cactus National Monument), and provide a rationale for the sample choice. When relevant, describe the organism taxa, source, sex, age range and any manipulations. State what population the sample is meant to represent when applicable. For studies involving existing datasets, describe the data and its source.*
Sampling strategy	*Note the sampling procedure. Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.*
Data collection	*Describe the data collection procedure, including who recorded the data and how.*
Timing and spatial scale	*Indicate the start and stop dates of data collection, noting the frequency and periodicity of sampling and providing a rationale for these choices. If there is a gap between collection periods, state the dates for each sample cohort. Specify the spatial scale from which the data are taken*
Data exclusions	*If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.*
Reproducibility	*Describe the measures taken to verify the reproducibility of experimental findings. For each experiment, note whether any attempts to repeat the experiment failed OR state that all attempts to repeat the experiment were successful.*
Randomization	*Describe how samples/organisms/participants were allocated into groups. If allocation was not random, describe how covariates were controlled. If this is not relevant to your study, explain why.*
Blinding	*Describe the extent of blinding used during data acquisition and analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.*

Did the study involve field work? ☐ Yes ☐ No

Field work, collection and transport

Field conditions	*Describe the study conditions for field work, providing relevant parameters (e.g. temperature, rainfall).*
Location	*State the location of the sampling or experiment, providing relevant parameters (e.g. latitude and longitude, elevation, water depth).*
Access & import/export	*Describe the efforts you have made to access habitats and to collect and import/export your samples in a responsible manner and in compliance with local, national and international laws, noting any permits that were obtained (give the name of the issuing authority, the date of issue, and any identifying information).*
Disturbance	*Describe any disturbance caused by the study and how it was minimized.*

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
☒	☐ Human research participants
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

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

Antibodies

Antibodies used — *Describe all antibodies used in the study; as applicable, provide supplier name, catalog number, clone name, and lot number.*

Validation — *Describe the validation of each primary antibody for the species and application, noting any validation statements on the manufacturer's website, relevant citations, antibody profiles in online databases, or data provided in the manuscript.*

Eukaryotic cell lines

Policy information about cell lines

Cell line source(s) — *State the source of each cell line used.*

Authentication — *Describe the authentication procedures for each cell line used OR declare that none of the cell lines used were authenticated.*

Mycoplasma contamination — *Confirm that all cell lines tested negative for mycoplasma contamination OR describe the results of the testing for mycoplasma contamination OR declare that the cell lines were not tested for mycoplasma contamination.*

Commonly misidentified lines (See ICLAC register) — *Name any commonly misidentified cell lines used in the study and provide a rationale for their use.*

Palaeontology and Archaeology

Specimen provenance — *Provide provenance information for specimens and describe permits that were obtained for the work (including the name of the issuing authority, the date of issue, and any identifying information).*

Specimen deposition — *Indicate where the specimens have been deposited to permit free access by other researchers.*

Dating methods — *If new dates are provided, describe how they were obtained (e.g. collection, storage, sample pretreatment and measurement), where they were obtained (i.e. lab name), the calibration program and the protocol for quality assurance OR state that no new dates are provided.*

☐ Tick this box to confirm that the raw and calibrated dates are available in the paper or in Supplementary Information.

Ethics oversight — *Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.*

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

Animals and other organisms

Policy information about studies involving animals; ARRIVE guidelines recommended for reporting animal research

Laboratory animals — *For laboratory animals, report species, strain, sex and age OR state that the study did not involve laboratory animals.*

Wild animals — *Provide details on animals observed in or captured in the field; report species, sex and age where possible. Describe how animals were caught and transported and what happened to captive animals after the study (if killed, explain why and describe method; if released, say where and when) OR state that the study did not involve wild animals.*

Field-collected samples — *For laboratory work with field-collected samples, describe all relevant parameters such as housing, maintenance, temperature, photoperiod and end-of-experiment protocol OR state that the study did not involve samples collected from the field.*

Ethics oversight — *Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.*

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

Human research participants

Policy information about studies involving human research participants

Population characteristics — *Describe the covariate-relevant population characteristics of the human research participants (e.g. age, gender, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write "See above."*

Recruitment — *Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.*

Ethics oversight — *Identify the organization(s) that approved the study protocol.*

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

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

Clinical trial registration	*Provide the trial registration number from ClinicalTrials.gov or an equivalent agency.*
Study protocol	*Note where the full trial protocol can be accessed OR if not available, explain why.*
Data collection	*Describe the settings and locales of data collection, noting the time periods of recruitment and data collection.*
Outcomes	*Describe how you pre-defined primary and secondary outcome measures and how you assessed these measures.*

Dual use research of concern

Policy information about dual use research of concern

Hazards

Could the accidental, deliberate or reckless misuse of agents or technologies generated in the work, or the application of information presented in the manuscript, pose a threat to:

No	Yes	
☐	☐	Public health
☐	☐	National security
☐	☐	Crops and/or livestock
☐	☐	Ecosystems
☐	☐	Any other significant area

Experiments of concern

Does the work involve any of these experiments of concern:

No	Yes	
☐	☐	Demonstrate how to render a vaccine ineffective
☐	☐	Confer resistance to therapeutically useful antibiotics or antiviral agents
☐	☐	Enhance the virulence of a pathogen or render a nonpathogen virulent
☐	☐	Increase transmissibility of a pathogen
☐	☐	Alter the host range of a pathogen
☐	☐	Enable evasion of diagnostic/detection modalities
☐	☐	Enable the weaponization of a biological agent or toxin
☐	☐	Any other potentially harmful combination of experiments and agents

ChIP-seq

Data deposition

☐ Confirm that both raw and final processed data have been deposited in a public database such as GEO.

☐ Confirm that you have deposited or provided access to graph files (e.g. BED files) for the called peaks.

Data access links *May remain private before publication.*	*For "Initial submission" or "Revised version" documents, provide reviewer access links. For your "Final submission" document, provide a link to the deposited data.*
Files in database submission	*Provide a list of all files available in the database submission.*
Genome browser session (e.g. UCSC)	*Provide a link to an anonymized genome browser session for "Initial submission" and "Revised version" documents only, to enable peer review. Write "no longer applicable" for "Final submission" documents.*

Methodology

Replicates	*Describe the experimental replicates, specifying number, type and replicate agreement.*
Sequencing depth	*Describe the sequencing depth for each experiment, providing the total number of reads, uniquely mapped reads, length of reads and*

Sequencing depth	*whether they were paired- or single-end.*
Antibodies	*Describe the antibodies used for the ChIP-seq experiments; as applicable, provide supplier name, catalog number, clone name, and lot number.*
Peak calling parameters	*Specify the command line program and parameters used for read mapping and peak calling, including the ChIP, control and index files used.*
Data quality	*Describe the methods used to ensure data quality in full detail, including how many peaks are at FDR 5% and above 5-fold enrichment.*
Software	*Describe the software used to collect and analyze the ChIP-seq data. For custom code that has been deposited into a community repository, provide accession details.*

Flow Cytometry

Plots

Confirm that:

☐ The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).

☐ The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).

☐ All plots are contour plots with outliers or pseudocolor plots.

☐ A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation	*Describe the sample preparation, detailing the biological source of the cells and any tissue processing steps used.*
Instrument	*Identify the instrument used for data collection, specifying make and model number.*
Software	*Describe the software used to collect and analyze the flow cytometry data. For custom code that has been deposited into a community repository, provide accession details.*
Cell population abundance	*Describe the abundance of the relevant cell populations within post-sort fractions, providing details on the purity of the samples and how it was determined.*
Gating strategy	*Describe the gating strategy used for all relevant experiments, specifying the preliminary FSC/SSC gates of the starting cell population, indicating where boundaries between "positive" and "negative" staining cell populations are defined.*

☐ Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.

Magnetic resonance imaging

Experimental design

Design type	*Indicate task or resting state; event-related or block design.*
Design specifications	*Specify the number of blocks, trials or experimental units per session and/or subject, and specify the length of each trial or block (if trials are blocked) and interval between trials.*
Behavioral performance measures	*State number and/or type of variables recorded (e.g. correct button press, response time) and what statistics were used to establish that the subjects were performing the task as expected (e.g. mean, range, and/or standard deviation across subjects).*

Acquisition

Imaging type(s)	*Specify: functional, structural, diffusion, perfusion.*
Field strength	*Specify in Tesla*
Sequence & imaging parameters	*Specify the pulse sequence type (gradient echo, spin echo, etc.), imaging type (EPI, spiral, etc.), field of view, matrix size, slice thickness, orientation and TE/TR/flip angle.*
Area of acquisition	*State whether a whole brain scan was used OR define the area of acquisition, describing how the region was determined.*

Diffusion MRI ☐ Used ☐ Not used

Preprocessing

Preprocessing software — *Provide detail on software version and revision number and on specific parameters (model/functions, brain extraction, segmentation, smoothing kernel size, etc.).*

Normalization — *If data were normalized/standardized, describe the approach(es): specify linear or non-linear and define image types used for transformation OR indicate that data were not normalized and explain rationale for lack of normalization.*

Normalization template — *Describe the template used for normalization/transformation, specifying subject space or group standardized space (e.g. original Talairach, MNI305, ICBM152) OR indicate that the data were not normalized.*

Noise and artifact removal — *Describe your procedure(s) for artifact and structured noise removal, specifying motion parameters, tissue signals and physiological signals (heart rate, respiration).*

Volume censoring — *Define your software and/or method and criteria for volume censoring, and state the extent of such censoring.*

Statistical modeling & inference

Model type and settings — *Specify type (mass univariate, multivariate, RSA, predictive, etc.) and describe essential details of the model at the first and second levels (e.g. fixed, random or mixed effects; drift or auto-correlation).*

Effect(s) tested — *Define precise effect in terms of the task or stimulus conditions instead of psychological concepts and indicate whether ANOVA or factorial designs were used.*

Specify type of analysis: ☐ Whole brain ☐ ROI-based ☐ Both

Statistic type for inference (See Eklund et al. 2016) — *Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods.*

Correction — *Describe the type of correction and how it is obtained for multiple comparisons (e.g. FWE, FDR, permutation or Monte Carlo).*

Models & analysis

n/a	Involved in the study
☐	☐ Functional and/or effective connectivity
☐	☐ Graph analysis
☐	☐ Multivariate modeling or predictive analysis

Functional and/or effective connectivity — *Report the measures of dependence used and the model details (e.g. Pearson correlation, partial correlation, mutual information).*

Graph analysis — *Report the dependent variable and connectivity measure, specifying weighted graph or binarized graph, subject- or group-level, and the global and/or node summaries used (e.g. clustering coefficient, efficiency, etc.).*

Multivariate modeling and predictive analysis — *Specify independent variables, features extraction and dimension reduction, model, training and evaluation metrics.*