Structural motif search across the protein-universe with Folddisco

Hyunbin Kim[1,2], Rachel Seongeun Kim[1,2], Milot Mirdita[2], and Martin Steinegger[1,2,3,4,✉]

Detecting similar protein structural motifs, functionally crucial short 3D patterns, in large structure collections is computationally prohibitive. Therefore, we developed Folddisco, which overcomes this through an index of position-independent geometric features, including side-chain orientation, combined with a rarity-based scoring system. Folddisco indexes 53 million AFDB50 structures into 1.45 terabyte within 24 hours, enabling rapid detection of discontinuous or segment motifs. Folddisco is more accurate and storage-efficient than state-of-the-art methods, while being an order of magnitude faster. Folddisco is free software available at folddisco.foldseek.com and a webserver at search.foldseek.com/folddisco.

Contact: *martin.steinegger@snu.ac.kr*

Structural motifs are short, recurring arrangements of tertiary structural elements that form recognizable patterns in proteins and are often associated with stability, binding interactions, or active sites (1, 2). Evolutionary constraints due to motif functionality often result in their conservation at sub-Angstrom resolution. Thus, identifying these motifs can provide functional insights, even for proteins with unknown function (3).

Notable examples for the links between structural motifs and function include the Cys2-His2 zinc finger motif in transcription factors and the CWxP, NPxxY and DRY motifs in G-protein-coupled receptors (GPCRs), which have been shown to bind zinc ions, thereby stabilizing DNA binding structure (4) and to be involved in receptor activation (5), respectively.

Although structural motifs can directly provide functional insights, most functional annotation methods rely predominantly on sequence information, despite its indirect relationship to function (6). This is largely due to the high throughput of sequencing and alignment techniques (7, 8), in contrast to the relative scarcity of structural data and the limited capabilities of structure-comparison methods until recently (9).

However, recent revolutionary advances by AlphaFold2 (10, 11) and other deep learning-based structure prediction methods now offer hundreds of millions (12, 13) of protein structures. These advances have motivated the development of rapid and scalable structural aligners, such as Foldseek (14), which exploits this potential by enabling direct structure-based functional annotation (15). Despite its strengths, Foldseek is not built for motif detection, as it assumes that residues match in linear order, in contrast to the non-linear path of far-apart matching pieces, common to structural motifs.

The RCSB motif search method (16) uses the Protein Data Bank (PDB) (17) as an input database. This method tackles the non-linearity problem by breaking each structure into proximal residue pairs, and extracting, for each pair, the residues' amino acid (AA) identities as well as geometric features: the distance between their Cα atoms, the distance between their Cβ atoms, and the intersecting angle between the Cα-Cβ vectors. Each such 5-feature set is saved in an inverted index, which maps it to the PDB entry and positions where it occurs. Since the number of proximal pairs scales roughly with each structure's residue count, indexing requires ∼75x more feature extraction and storage operations than the number of residues. As a result, the RCSB method took 3.5 days and 55GB to index 160,467 structures. pyScoMotif (18) is a faster Python-based motif finder utilizing the same pair representation, except that it uses side-chain centroids instead of Cβ atoms. It reduced the indexing time to 20.5 hours for 195,000 structures, but still required 73GB, making the indexing time and storage requirement the key limiting factors.

Another limitation of current motif search methods is their lack of flexibility in handling various query motif types and lengths. For instance, RCSB's service supports query motifs of up to 10 residues, restricting the method only to short motifs. Alignment-based fragment search methods, like MASTER (19), can handle longer, discontinuous queries, but struggle with short motifs like catalytic triads or zinc fingers.

Here, we present Folddisco (Fig. 1a-c), the first motif search algorithm that supports both short motif queries (Fig. 1d) and long, discontinuous segments (Fig. 1e) within a single framework. It operates efficiently on a massive scale, indexing 53 million structures in under 24 hours (<1.5 TB) with queries taking only a few seconds. This performance makes it >18x faster and requires <3.5x less storage than the state-of-the-art. Folddisco examines proximal residue pairs in each input structure, extracts a set of features from each pair, encodes the set numerically, and stores it in an index (Methods, Fig. 1b). Extending RCSB's feature set, Folddisco introduces two additional features: the torsion angles between the N–Cα and Cβ atoms, used by trRosetta for structure prediction (20). This feature set is more specific and eliminates the need to store the positions of the matching residues, allowing the index to map feature sets only to the structure identifiers (IDs) in which they occur, substantially reducing its size. Furthermore, the two angles capture side-chain orientation, which is crucial for detecting enzyme-activity related motifs.

Folddisco's full querying pipeline consists of four steps: feature extraction, pre-filtering, residue matching, and superposition (Fig. 1c). First, the query motif's feature sets are extracted and encoded as integers. Then, Folddisco pre-filters structures that share at least one feature set with the query motif—including variants within defined distance, angle, and amino acid tolerances—by looking up the query's encodings in the index. Folddisco prioritizes the most relevant candidate structures by introducing a coverage score to rank them by their specificity to the motif. This score (Methods) uses Inverse Document Frequency (21, IDF) weights computed over the entire index, rewarding rare feature sets and penalizing common ones (e.g., helices). A length penalty further reduces random hits in large proteins, ensuring consistency for queries

[1]Interdisciplinary Program in Bioinformatics, Seoul National University, Seoul, Republic of Korea. [2]School of Biological Sciences, Seoul National University, Seoul, Republic of Korea. [3]Institute of Molecular Biology and Genetics, Seoul National University, Seoul, Republic of Korea. [4]Artificial Intelligence Institute, Seoul National University, Seoul, Republic of Korea.

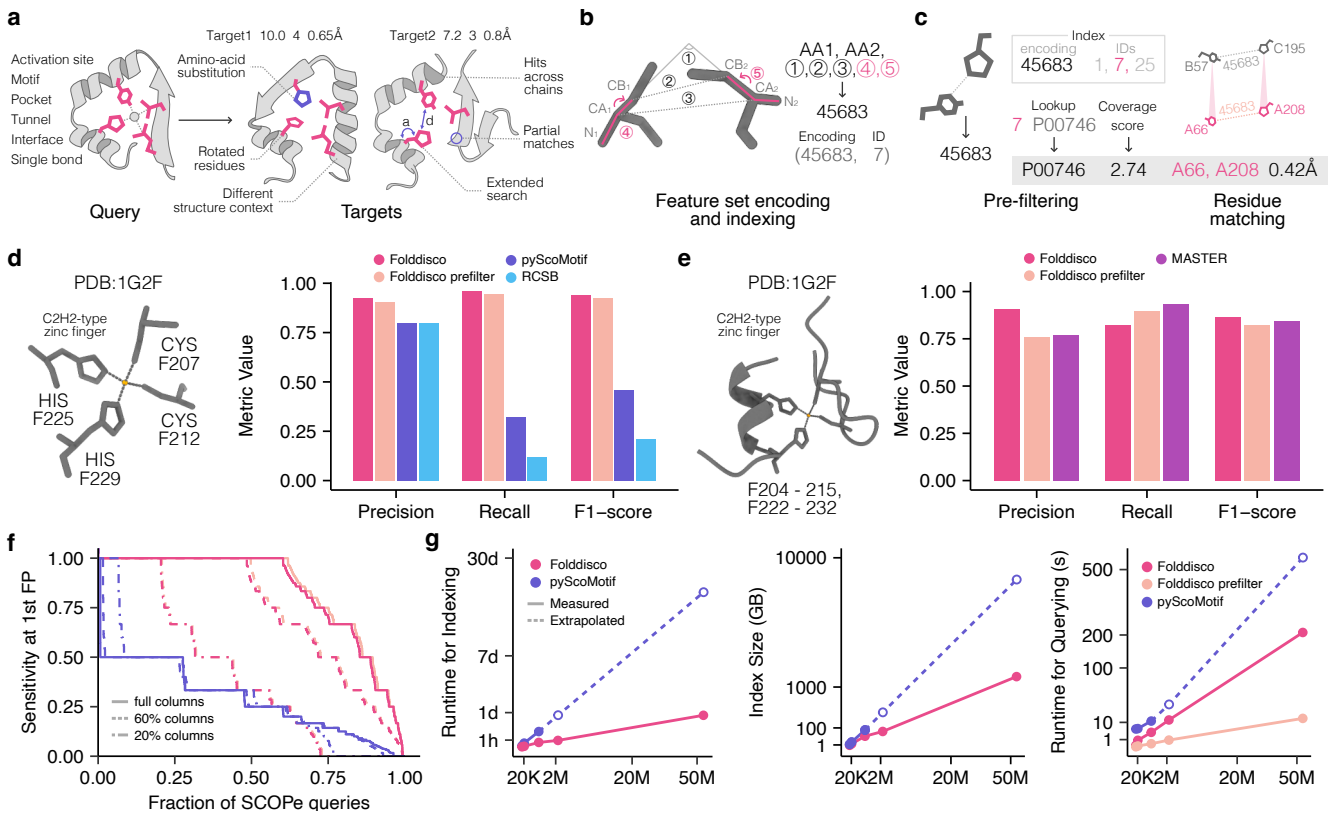


Fig. 1. Folddisco's workflow and benchmark. a, Folddisco is a fast tool for sensitive motif detection in millions of protein structures. Given motif-defining query residues (a, left), it examines proximal pairs (<20Å) and computes feature sets for each pair. To increase sensitivity, it can generate additional feature sets accounting for amino-acid substitutions, side-chain flexibility, and increased distances/angles (Methods "Extended search"). Each set is encoded (b) and rapidly searched (c) against a precomputed index of pairwise features from database structures. **b**, Feature set and index: Folddisco associates each structure with an ID and extracts 5 RCSB features (black) and 2 new features (pink) from its pairs of proximate residues. Each set of 7 features is bit-encoded and stored in an index that maps to all IDs in which the set was found. **c**, Querying: analogous feature extraction from proximal motif residues is followed by retrieval of structure IDs that share its feature sets ("pre-filter"). Pre-filtered structures can be further processed to match their residues (pink) to the query (gray). **d**,**e**, Folddisco is the most accurate method in querying the human fraction of the AFDB-proteome for zinc fingers, both when using a short motif query suitable for pyScoMotif and RCSB (**d**, left; residue labels, e.g. F207, denote chain and residue number) and when using the motif-containing segments suitable for MASTER (**e**, left). **f**, Folddisco achieves higher sensitivity than pyScoMotif on SCOPe-constructed benchmarks, where the goal is to match SCOPe sequences of the same family as the query before matching a different fold, using all conserved columns ("full") or a random subsample of them (60%, 20%). **g**, Scalability comparison on various sized databases. Indexing speed (left): Folddisco is 18x faster than pyScoMotif; Index size (middle): Folddisco's index is 3.5x smaller than pyScoMotif's; Querying speed (right): search time of the zinc finger motif (panel d) is up-to 48x (3x) shorter for Folddisco pre-filter (-full) compared to pyScoMotif.

ranging from a single residue pair to an entire structure.

Next, Folddisco can optionally identify the motif-forming residues in each candidate that passed the pre-filter. To do so, it constructs a graph where each candidate residue is a node, and directed edges are drawn between node pairs that match the query motif's residue pairs either by having the same feature set (as detected in the pre-filter) or a similar one (extended search, see Methods). Folddisco detects connected components in this graph, each of which is a proposed match to the motif that is superposed to it, and the match's root mean square deviation (RMSD) is computed.

We compared the accuracy of Folddisco to that of RCSB and pyScoMotif for detecting the zinc finger motif (partial or full) and the serine peptidase motif in 23,391 AlphaFold2-predicted structures of the human proteome (Methods). Each detection of the zinc finger motif was counted as true positive (TP) if it matched the PROSITE (22) rule PRU00042, and as false positive (FP) otherwise. For serine peptidase, TPs belonged to MEROPS' (23) family S1 and other detections were considered as FPs. Precision, recall, and F1 scores were calculated using these counts and the total number of positives (P) for the zinc finger (P=761) and serine peptidase (P=124) motifs in the human proteome. All three tools performed comparably on serine peptidase and the partial (three residues) zinc finger motif (Extended Data Fig. 1).

However, Folddisco outperformed both methods when querying the full, four-residue zinc finger motif, where RCSB and pyScoMotif had low recall (Fig. 1d). Folddisco was also more accurate than MASTER (Fig. 1e) when the query was provided as segments containing the zinc finger motif (Methods), while pyScoMotif failed to return any results (Supplementary Table 1). Thus, Folddisco is the only method that can search both discrete motifs and discontinuous segments. In all benchmarks, Folddisco's pre-filter alone achieved competitive accuracy, demonstrating the effectiveness of its extended feature set and coverage score ranking. We studied the runtime and scalability of Folddisco in a separate benchmark but even in this one with a single database, Folddisco's query time was >7 times faster than any other method (Extended Data Fig. 1).

To evaluate Folddisco's generalizability beyond zinc fingers and serine peptidases, we developed a benchmark based on SCOPe (24). Instead of relying on full SCOPe-domain alignments, we aimed to mimic motifs by selecting conserved and scattered residues from family-level multiple sequence align-

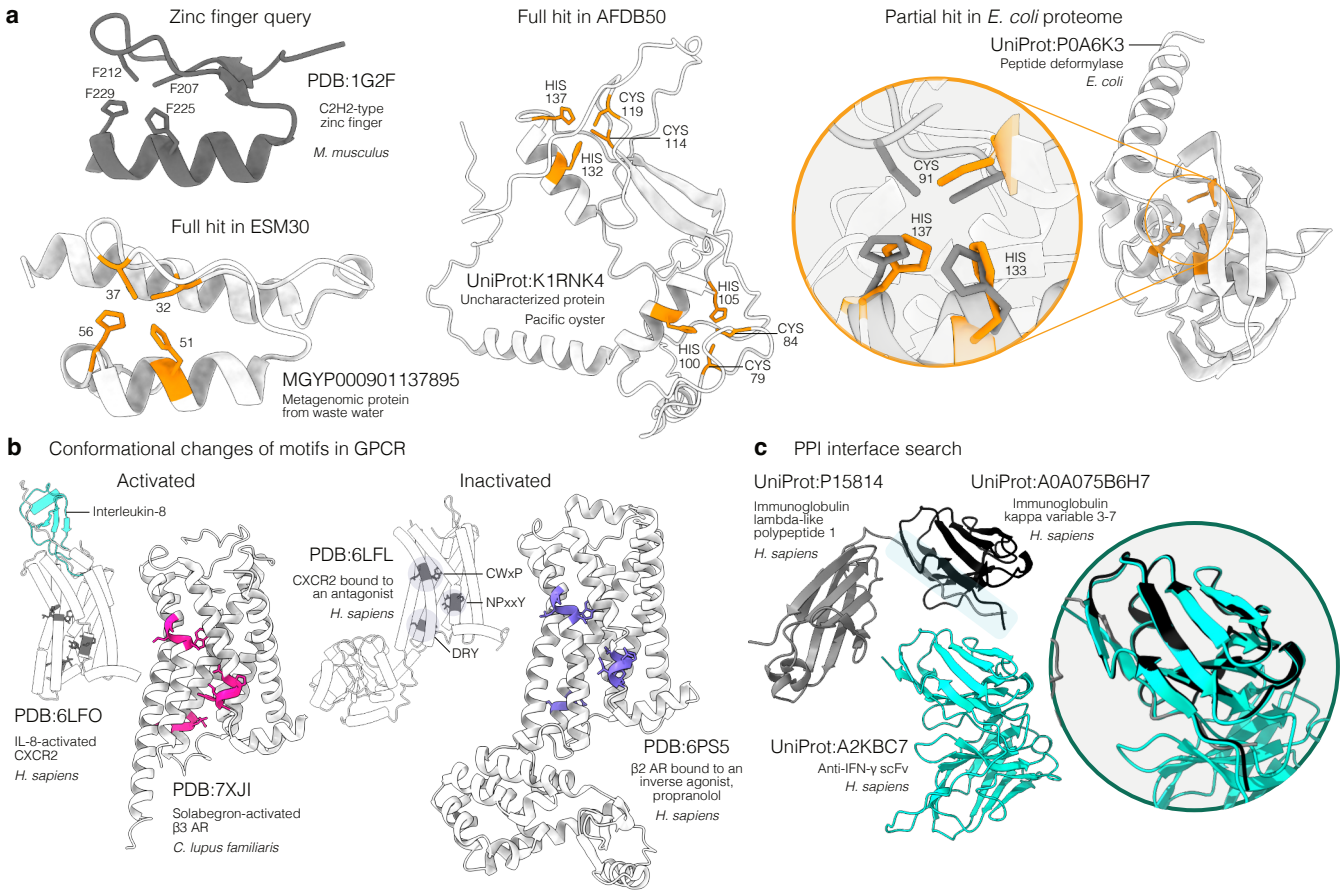


Fig. 2. Applications of Folddisco. a, Zinc finger motif detection. A query for a C2H2 zinc finger (top-left) identifies full hits in previously unannotated proteins (bottom-left, middle) and a partial hit corresponding to the known metal-coordinating site in an E. coli enzyme (right). **b**, Conformational state identification. Queries using motifs from activated (left, magenta) or inactivated (right, purple) G-protein-coupled receptors (GPCRs) successfully retrieve structures in the corresponding functional states. **c**, Protein interface search. A query using an immunoglobulin domain interface (left) retrieves a single-chain variable fragment that exhibits a similar binding geometry (right).

ments generated by the structural aligner FoldMason (25). In each alignment we identified columns with full occupancy (no gaps) and a "dominant" residue (occurring in >66% of the members) and constructed three benchmarks by using the dominant residues from all identified columns as a query (termed "full"); by randomly sampling 60% of the columns and using their dominant residues; and by sampling and using 20%. Each such query was searched against the SCOPe database and a match was counted as TP if it belonged to the same family as the query, FP if it belonged to a different fold, and ignored otherwise. Sensitivity was measured as the fraction of correctly identified family members (TP/P) before the first FP, where the ranking of the matches was by the coverage score for Folddisco's pre-filter or by RMSD for pyScoMotif and Folddisco's full pipeline. Folddisco was consistently more sensitive than pyScoMotif with Area Under the Sensitivity Curve values of 0.837, 0.733 and 0.414 compared to 0.300, 0.290, and 0.285 for the three benchmarks, respectively (Fig. 1f). Of note, while pyScoMotif's performance peaked when queries had fewest residues (on 20%) and was nearly the same otherwise (60% and full), Folddisco improved with every gain of information. Here too, Folddisco's pre-filter alone achieved performance comparable to the full pipeline, indicating that its coverage score ranking is in high agreement with RMSD-based sorting, despite being approximately 15-fold faster.

Next, we studied Folddisco's scalability in comparison to pyScoMotif, which is a faster implementation of the RCSB method. To that end, we used Folddisco to index five databases, holding between 4K and 53M structures (Methods) and pyScoMotif to index the three smallest of them. Index construction by Folddisco was faster than by pyScoMotif, taking 26.5 minutes for 540K structures using 64 cores, compared to 4.87 hours (Fig. 1g, left). Folddisco's storage requirement of 31.5GB was less than half of pyScoMotif's 79GB for the 540K database (Fig. 1g, middle). By extrapolating the requirements of pyScoMotif for larger databases, we find that Folddisco's index construction time and storage requirement improve even more as the input database increases. This means, for example, that indexing the 53M structures of the AFDB50 required 1.45TB by Folddisco, compared to 5.38 times more—7.8TB extrapolated for pyScoMotif. Folddisco's full pipeline's querying time of the zinc finger motif was shorter by a factor of ∼3, compared to pyScoMotif for all databases. Folddisco's pre-filter was 50x faster: nearly instantaneous for smaller databases and taking only ∼13 seconds for AFDB50 (Fig. 1g, right). Having established Folddisco's motif search capability, we examined three use cases: functional annotation of divergent sequences, search for protein state-defining motifs and interface detection.

Folddisco identified a zinc finger motif in metagenomic- (from ESM30) and uncharacterized oyster (from AFDB50) proteins, which lack sequence-level annotations such as InterPro (8) domains (Fig. 2a, left and middle). It also recognized a partial motif in *E. coli*, pinpointing known metal-coordinating sites (26) of peptide deformylase (Fig. 2a, right). These examples demonstrate the advantage of Folddisco over Foldseek for detecting motifs, rather than longer structural elements, as Foldseek scored the two uncharacterized proteins with high E-values of >20 (commonly above the cutoff for discarding) and could not at all align the query to the *E. coli* protein. These discoveries and similar ones (Supp. Fig. 2) underscore Folddisco's capacity to detect structurally conserved yet sequence-divergent features.

Next, Fig. 2b demonstrates that Folddisco searches related to GPCR activation—the CWxP, NPxxY, and DRY motifs of CXCR2—successfully distinguish between active and inactive β-adrenergic receptor structures in the PDB and their characteristic activation-state patterns. Since AlphaFold2 has been shown to sample different conformational states (27, 28), we sought to compare the prevalence of active versus inactive states in predicted and experimental databases. When run on the PDB, 54% of Folddisco's matches were in the active state and a similar fraction (53%) was found in the AFDB50. This suggests that the conformational landscape sampled by AlphaFold2 closely mirrors the prevalence of functional states found in the PDB.

For the last use case, Folddisco queried a cross chain protein–protein interface motif pattern (29) derived from immunoglobulin λ-like and immunoglobulin κ variable domains (Fig. 2c) in AFDB50, retrieving a single-chain variable fragment exhibiting the same interaction geometry. Demonstrating more capabilities, Folddisco successfully identified disulfide bonds (Supp. Fig. 3) and short linear motifs (Supp. Fig. 4).

To facilitate access to Folddisco motif searching, we developed a webserver (search.foldseek.com/folddisco). Queries can be provided as a standalone motif or as a full protein structure with specified motif residues. The webserver provides prebuilt indices of major databases: AFDB50, PDB100, AFDB-proteome (12) and ESM30. For each query, it returns up to 1,000 top-ranked matches per database, along with Folddisco scores and interactive structure visualizations. Searching for the full zinc finger motif (Fig. 1d) in all databases at once completes in approximately 100 seconds on a single core.

In conclusion, Folddisco's compact index enhances scalability by reducing storage and indexing time, enabling fast querying of large databases like AFDB50 and ESM30. Its features for side-chain orientations and rarity-based scoring allow accurate detection of both short motifs and long fragments. By uncovering motifs linked to catalysis, complex formation, and conformational regulation, Folddisco facilitates mechanistic insights across diverse taxonomic and functional landscapes.

References

1. Bartlett, G. J., Porter, C. T., Borkakoti, N. & Thornton, J. M. Analysis of Catalytic Residues in Enzyme Active Sites. *Journal of Molecular Biology* **324**, 105–121 (2002).
2. Fernandez-Fuentes, N., Dybas, J. M. & Fiser, A. Structural Characteristics of Novel Protein Folds. *PLoS Computational Biology* **6**, e1000750 (2010).
3. Redfern, O. C., Dessailly, B. & Orengo, C. A. Exploring the structure and function paradigm. *Current Opinion in Structural Biology* **18**, 394–402 (2008).
4. Pabo, C. O., Peisach, E. & Grant, R. A. Design and selection of novel Cys2His2 zinc finger proteins. *Annual Review of Biochemistry* **70**, 313–340 (2001).
5. Zhou, Q. *et al.* Common activation mechanism of class A GPCRs. *eLife* **8**, e50279 (2019).
6. Svedberg, D. *et al.* Functional annotation of a divergent genome using sequence and structure-based similarity. *BMC Genomics* **25**, 6 (2024).
7. Steinegger, M. & Söding, J. MMseqs2 enables sensitive protein sequence searching for the analysis of massive data sets. *Nature Biotechnology* **35**, 1026–1028 (2017).
8. Paysan-Lafosse, T. *et al.* InterPro in 2022. *Nucleic Acids Research* **51**, D418–D427 (2023).
9. Hasegawa, H. & Holm, L. Advances and pitfalls of protein structural alignment. *Current Opinion in Structural Biology* **19**, 341–348 (2009).
10. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).
11. Mirdita, M. *et al.* ColabFold: making protein folding accessible to all. *Nature Methods* **19**, 679–682 (2022).
12. Varadi, M. *et al.* AlphaFold Protein Structure Database in 2024: providing structure coverage for over 214 million protein sequences. *Nucleic Acids Research* **52**, D368–D375 (2024).
13. Lin, Z. *et al.* Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science* **379**, 1123–1130 (2023).
14. Van Kempen, M. *et al.* Fast and accurate protein structure search with Foldseek. *Nature Biotechnology* **42**, 243–246 (2024).
15. Ruperti, F. *et al.* Cross-phyla protein annotation by structural prediction and alignment. *Genome Biology* **24**, 113 (2023).
16. Bittrich, S., Burley, S. K. & Rose, A. S. Real-time structural motif searching in proteins using an inverted index strategy. *PLoS Computational Biology* **16**, e1008502 (2020).
17. Burley, S. K. *et al.* Updated resources for exploring experimentally-determined PDB structures and Computed Structure Models at the RCSB Protein Data Bank. *Nucleic Acids Research* **53**, D564–D574 (2024).
18. Cia, G., Kwasigroch, J., Stamatopoulos, B., Rooman, M. & Pucci, F. pyScoMotif: Discovery of similar 3D structural motifs across proteins. *Bioinformatics Advances* **3**, vbad158 (2023).
19. Zhou, J. & Grigoryan, G. Rapid search for tertiary fragments reveals protein sequence–structure relationships. *Protein Science* **24**, 508–524 (2015).
20. Yang, J. *et al.* Improved protein structure prediction using predicted interresidue orientations. *Proceedings of the National Academy of Sciences* **117**, 1496–1503 (2020).
21. Sparck Jones, K. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation* **28**, 11–21 (1972).
22. Sigrist, C. J. *et al.* ProRule: a new database containing functional and structural information on PROSITE profiles. *Bioinformatics* **21**, 4060–4066 (2005).
23. Rawlings, N. D. *et al.* The MEROPS database of proteolytic enzymes, their substrates and inhibitors in 2017 and a comparison with peptidases in the PANTHER database. *Nucleic Acids Research* **46**, D624–D632 (2018).
24. Chandonia, J.-M. *et al.* SCOPe: improvements to the structural classification of proteins–extended database to facilitate variant interpretation and machine learning. *Nucleic Acids Research* **50**, D553–D559 (2022).
25. Gilchrist, C. L., Mirdita, M. & Steinegger, M. Multiple Protein Structure Alignment at Scale with FoldMason. *bioRxiv* 2024.08.01.606130v3 (2024).
26. Becker, A. *et al.* Iron center, substrate recognition and mechanism of peptide deformylase. *Nature Structural Biology* **5**, 1053–1058 (1998).
27. Del Alamo, D., Sala, D., Mchaourab, H. S. & Meiler, J. Sampling alternative conformational states of transporters and receptors with AlphaFold2. *eLife* **11**, e75751 (2022).
28. Kalakoti, Y. & Wallner, B. AFsample2 predicts multiple conformations and ensembles with AlphaFold2. *Communications Biology* **8**, 373 (2025).
29. Zhang, J. *et al.* Computing the human interactome. *bioRxiv* 2024.10.01.615885v1 (2024).

Methods

General workflow

Indexing. Folddisco is designed to efficiently query a motif in input databases of many millions of protein structures. Therefore, we generate an index by assigning each database protein structure a numerical ID and examining its pairs of proximal residues (default radius: 20Å). From each proximal pair Folddisco extracts two sets of 7 features (see "Pairwise features") and jointly encodes each of them as a 32-bit unsigned integer (see "Encoding feature sets as integers"). Each unsigned integer serves as a key to retrieve the input IDs of structures in which its encoded feature set was found (see "Mode of indexing"). Folddisco's index does not include the structure positions of the proximate residues, but they can be optionally reconstructed (see "Residue matching"). This strategy in combination with delta compression of the IDs results in a more compact index compared to the indices of conventional position-storing motif search methods.

Querying. Pairs of proximal residues are identified in the query motif in the same way input structures are processed during indexing. For each pair at position i and j, two sets of features—(i,j) and its reverse (j,i), are extracted and encoded because of the asymmetry in the representation. To increase sensitivity, Folddisco can search for more encodings through amino-acid substitutions and adjustable distance/angle thresholds ("Extended search"). In the pre-filter step, the 32-bit integers computed for the query motif are used as keys to retrieve IDs of indexed structures that share at least one feature set with the query. Folddisco then ranks the IDs by their coverage of the motif, i.e., by the number of feature sets they share with it and the sets' rarity (see "Pre-filtering"). After pre-filtering, an optional step can match the residues of the query to those of pre-filtered structures. As Folddisco doesn't index positional information, this step is conducted by finding connected graph components (see "Residue matching").

Folddisco's feature set

Pairwise features. Folddisco extracts two feature sets from each pair of proximal residues in each input structure as well as the query. This set includes 5 features used by RCSB: the amino acid type of the residues (AA1 and AA2), the distance between their Cα atoms, the distance between their Cβ atoms, and the intersecting angle between the Cα-Cβ vectors. Since RCSB's features do not capture the side-chain orientations of the residues, we included 2 additional features in Folddisco's set: the two dihedral angles in the atoms of N1-Cα1-Cβ1-Cβ2 and N2-Cα2-Cβ2-Cβ1, which were proposed by trRosetta for structure prediction (20). Since these two features are not symmetric and may have different values, depending on which residue is considered as the first, Folddisco considers two feature sets, one in the direction of AA1-AA2 and another in the direction of AA2-AA1.

Encoding feature sets as integers. Folddisco encodes each of the 7 features in a set in bits and concatenates them bitwise as follows. AA1 and AA2 are treated numerically (0,1,...,19), requiring 5 bits each. The distance features are discretized into bins from 0 to 20Å (default number: 16), requiring 4 bits each. To keep the cyclic nature of the angle features, we discretize their cosine and sine values from -1 to 1 (default: 4 bins for cosine, 4 bins for sine), requiring 4 bits per angle. In total, this encoding requires 30 bits, which can be represented as a 32-bit unsigned integer (first 2 bits are always '0').

Index building

Index format. Folddisco's index consists of 4 files; main index file (`*.value`), offset (`*.offset`), lookup (`*.lookup`), and metadata (`*.type`). The main index file stores the numerical IDs of the protein structures as values of the keys. The offset file stores the offset of each key in the main index file from the first key to enable random access to the index. The lookup file is for mapping the numerical IDs to the original textual identifiers of the protein structures (e.g., 0 mapped to 'P00568'). The metadata file stores the information required for querying, such as the path to the input protein structures, binning information, and the indexing mode used (see section).

Mode of indexing. Folddisco supports two modes of indexing based on the data structure of the offset file: array-based or hashmap-based. The array-based offset file's size is fixed and determined by the theoretical potential of unique feature set encodings (keys). As Folddisco's default encoding uses 30 bits, there can be up-to 2^{30} encodings, each of which requires 8 bytes for its offset, thus the array-based offset file requires 8 GB of memory $\left(2^{30} \times 8 \text{ bytes} = 8 \text{ GB}\right)$. In contrast, Folddisco's hashmap depends on the number of unique feature set encodings computed for the input and requires a memory allocation overhead three times this number to avoid hash collisions. For small input (rule of thumb: fewer than 65k protein structures), the hashmap-based offset is more space-efficient because it avoids allocating the full 8 GB required by the array-based offset. However, for large datasets, the array-based offset file is more efficient in terms of memory usage because it does not require the overhead allocation. Both offset formats support random access to the index with multiple threads. Folddisco handles the numerical IDs for each mode differently; delta compression is applied to 64-bit unsigned integers in the case of array-based indexing, while 16-bit unsigned integers are used for the hashmap-based index. In array-based mode, the compressed IDs are stored using multiple threads, so it is recommended to use SSDs for index construction to avoid an Input/Output bottleneck.

Protein structure input formats

Folddisco accepts input protein structures either as files in PDB or mmCIF format (optionally gzip-compressed), or as Foldcomp-compressed (30) files. When reading Foldcomp-compressed protein structures, Folddisco can iterate through them at a comparable speed compared to reading uncompressed protein structures in PDB/mmCIF format.

Pre-filtering

When querying a motif, Folddisco first applies a computationally inexpensive pre-filter to eliminate most non-matching

structures before any structures are read from disk. The query motif is provided to Folddisco in PDB/mmCIF format. By default (but see also "Extended search"), pairs of proximal residues are identified in the query motif, and two sets of features are extracted and encoded from each of them in the same way the input structures are processed during indexing, resulting in a set of 32-bit unsigned integer keys. Folddisco uses these keys to list candidate structures that share at least one feature set with the query motif as follows.

Encodings' rarity. Folddisco computes Inverse Document Frequency (IDF) weights for each 32-bit encoding to represent its rarity among the input protein structures.

$$\mathrm{IDF}_e = \log_2\left(\frac{\#\text{ of structures in the index } e}{\#\text{ of structures containing encoding}}\right)$$

These weights are used to rank the candidate structure list by computing coverage scores.

Coverage scores. Each candidate is scored by the sum of the IDF weights of the encodings it shares with the query:

$$\mathrm{Score}_{\text{candid. struct.}} = L^{-\alpha}\sum_{i=1}^{n}\mathrm{IDF}_{e_i},$$

where n is the number of shared encodings, L is the length of the candidate structure in residues, and α is a length-penalty exponent (default 0.5) to avoid length-dependent random matches ranking high.

Motif completeness score based on query residues. In addition to the coverage score, Folddisco computes a motif completeness score for each candidate. Notably, if a given structure candidate shares two feature sets (encodings) with the query, they can involve either 3 or 4 distinct residues in the query: the two encodings from two pairs that share a residue (x-y and x-z) or two encodings from two distinct pairs (x-y and z-t). Since Folddisco has access to the query's residues also during pre-filtering, it counts for each candidate the number of distinct query residues its shared features with the query involve. This number is reported separately from the coverage score and can be used for optional post-filtering.

Extended search. The distance and angle features are controlled by thresholds (default = 0). When these thresholds are greater than 0, Folddisco will compute more encodings for each query proximal pair (e.g., with a slightly shorter/longer distance between Cα atoms), potentially matching more input structures. During benchmarking, we applied thresholds of 0.5Å for distances and 5° for angles to allow minor conformational differences. Folddisco also allows extended amino acid matching by providing alternative uppercase one-letter amino acid codes (e.g., 'A' for Alanine and 'X' for any amino acid) after the position of the query residue to extend. Users can also use custom lowercase one-letter codes to represent amino acid groups by properties: 'p' (positive), 'n' (negative), 'h' (hydrophilic), 'b' (hydrophobic), and 'a' (aromatic). Setting an amino acid alternative will prompt Folddisco to compute more query encodings, extending the search in the same way as the distance and angle thresholds do.

Residue matching and superposition

Graph construction. Despite not holding position information in its index, Folddisco can optionally match the residues of an input candidate to the query after the pre-filter step by establishing a graph where the nodes represent the candidate's residues. At this stage Folddisco reads the full information of each candidate, including residue positions from disk. Folddisco examines all candidate residue pairs that match at least one query residue pair in terms of amino acid identities (e.g., a candidate Cys-His pair will be examined if the query has Cys-His as one of its pairs). If a residue pair shares a feature set with some query residue pair, Folddisco links the residue pair's nodes with an edge in the direction indicated by the shared feature. In this stage, more than two feature sets are considered for each query residue pair by setting the distance and angle thresholds as described in the section "Extended search". This means that the residues of all pairs discovered by the pre-filter will be linked with at least one directed edge, but potentially additional off-by-a-little pairs will be linked as well. In addition, residue pairs with the same amino acid identities and a similar Cα–Cα distance to some query residue pair (<0.5Å difference by default) will be linked with an edge.

Connected components detection. Folddisco lists connected components in the graph built for the candidate, as these cohesive subgraphs indicate groups (and not just pairs) of residues that together are likely to form the motif. Folddisco identifies two connectivity types: strongly connected components with Tarjan's algorithm (31) and weakly connected components by first ignoring edge directions and then performing a DFS-based search (32).

Superposition computation. After residue matching, Folddisco superposes the query motif on the matched residues using the Quaternion Characteristic Polynomial algorithm (33). RMSD is calculated using the coordinates of the Cα and Cβ atoms of the query motif and the matched residues.

Benchmarks

Version of databases and software. We used pyScoMotif version 20231119 (commit 916b515), MASTER version 1.5, and Folddisco commit cc4cd7f in all benchmarks. PDB archive was downloaded on March 7th, 2024. We used AlphaFold database version 4 (released in October 2022), ESMatlas v0 (released in November 2022) and SCOPe v2.08. We additionally derived subsets from these resources: AFDB rep., the non-redundant representatives produced by Foldseek clustering of AFDB v4; AFDB50, the same AFDB v4 models clustered at a 50% sequence-identity threshold; and ESM30, high-confidence ESMatlas v0 models clustered at 30% sequence identity.

Human proteome benchmark

We benchmarked Folddisco on the human subset of the AFDB-proteome, which consists of 23,391 protein structures and compared it to RCSB's motif search, MASTER, and pyScoMotif. We used the motif set reported in Bittrich et al. (16), which includes the catalytic triad of serine protease in trypsin (PDB ID: 4CHA Residues: B57, B102, C195), the first letter of the residue's identifier is the chain ID, and the number refers to the residue's position in the respective chain. and the C2H2 zinc finger motif in TATA-binding early growth response protein 1 (EGR1). For the zinc finger motif, we used both the three residue-partial motif (PDB ID 1G2F: F207, F225, F229) and the four residue-full motif (PDB ID 1G2F: F207, F212, F225, F229) in our benchmark. As MASTER is designed for fragment searching, we prepared a segment query that includes the neighboring residues of the zinc-finger motif residues (PDB ID: 1G2F F204-215, F222-232) and used this segment in the benchmark. Detections were counted as true/false positives as described in the Main text.

Computing resource and resource measurement

All benchmarks—except the AFDB50 indexing step—were conducted with a server equipped with two 64-core AMD EPYC 7742 CPUs, 2TB of RAM and 15.3TB NVMe disk. AFDB50 indexing was carried out on a separate machine equipped with four Intel Xeon Gold 6328HL CPUs, 6 TB RAM, and the same 15.3 TB NVMe storage. Maximum RAM usage (maximum resident set size) and elapsed time of each tool were measured with the GNU `time -v` command.

Webserver

We integrated Folddisco into the MMseqs2 webserver platform (34). The Folddisco motif search is available when the webserver is launched in structure mode, in conjunction with Foldseek, Foldseek-Multimer and FoldMason. Users can search through Folddisco databases for AFDB50, AFDB-proteome, ESM30 and PDB100. The latter is built directly from the original PDB files, while the others were built from Foldcomp databases (Data availability). We prioritize matches by selecting the best 1,000 database structures according to their pre-filter coverage scores using the `--top 1000` parameter in Folddisco. These are passed to Folddisco's residue matching step, and the final result list is then sorted by number of matched nodes and RMSD. To visualize query and target residue matching, the matched target structures are decompressed from the Foldcomp database, except for PDB100 entries, which are retrieved directly as PDB files. All structure visualizations are rendered using the NGL viewer library (35).

Code availability

Folddisco is GPLv3-licensed free open-source software. The source code and ready-to-use binaries, as well as precomputed databases, can be downloaded at folddisco.foldseek.com. The scripts used for the benchmarks and plotting are available at https://github.com/steineggerlab/folddisco-analysis. The webserver code is available at github.com/soedinglab/mmseqs2-app.

References

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. Tarjan, R. Depth-first search and linear graph algorithms. *SIAM Journal on Computing* **1**, 146–160 (1972).
32. Sharir, M. A strong-connectivity algorithm and its applications in data flow analysis. *Computers & Mathematics with Applications* **7**, 67–72 (1981).
33. Theobald, D. L. Rapid calculation of RMSDs using a quaternion-based characteristic polynomial. *Foundations of Crystallography* **61**, 478–480 (2005).
34. Mirdita, M., Steinegger, M. & Söding, J. MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics* **35**, 2856–2858 (2019).
35. Rose, A. S. & Hildebrand, P. W. NGL Viewer: a web application for molecular visualization. *Nucleic Acids Research* **43**, W576–W579 (2015).

Acknowledgements

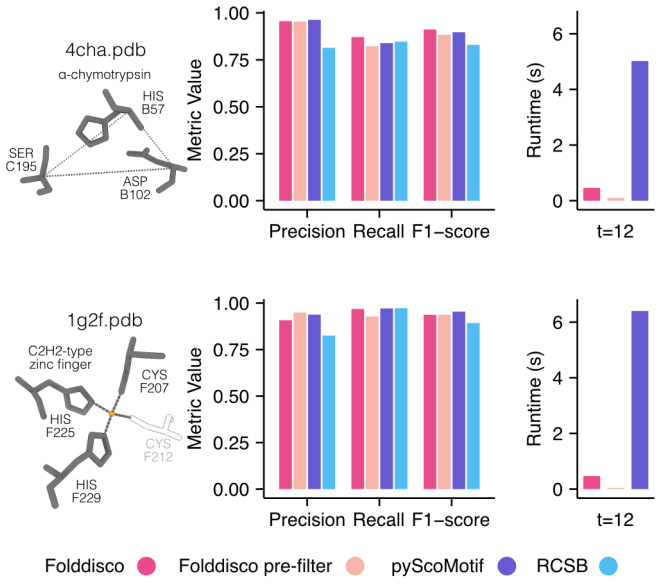
We thank Eli Levy Karin from ELKMO and Jaebeom Kim for their comments on the manuscript draft. M.S. acknowledges support by the National Research Foundation of Korea (grants 2020M3A9G7103933, RS-2021-NR061659 and RS-2021-NR056571 and RS-2024-00396026), Samsung DS Research Fund, Creative-Pioneering Researchers Program, AI-Bio Research Grant through Seoul National University, and Novo Nordisk Foundation (NNF24SA0092560). M.M. acknowledges support from the National Research Foundation of Korea (grant RS-2023-00250470).

Author contributions

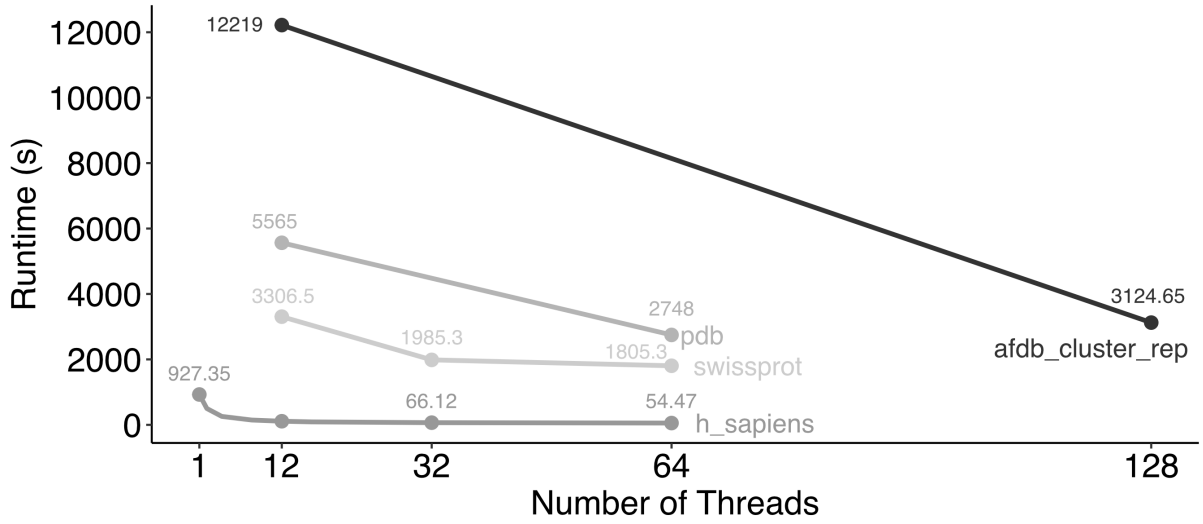
H.K., R.S.K., and M.S. designed the Folddisco algorithm. H.K., R.S.K., and M.M. developed the software. H.K. performed the benchmarks, designed figures, and wrote the manuscript, with contributions from all authors.

Competing interests

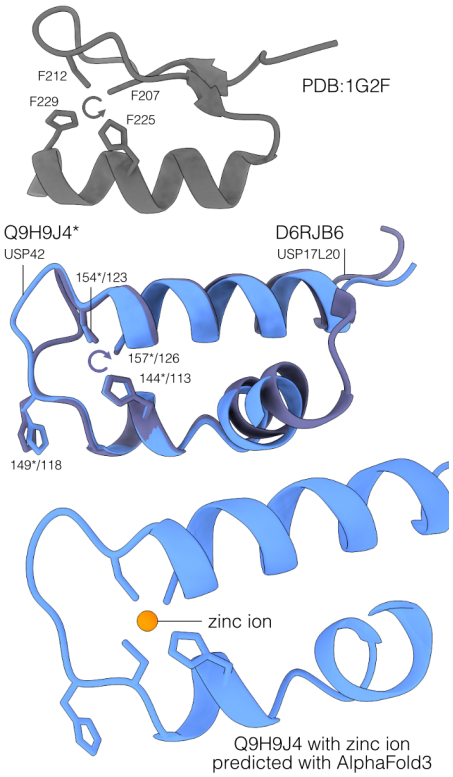
M.S. acknowledges outside interest in Stylus Medicine. The remaining authors declare no competing interests.



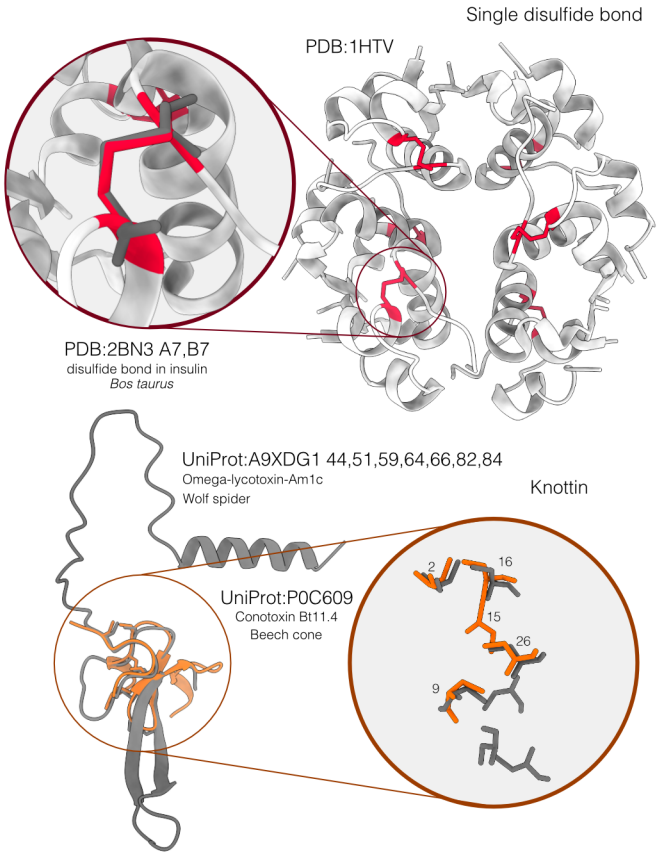
Extended Data Fig. 1. Performance comparison of Folddisco to pyScoMotif and RCSB Precision, recall, F1-score and search time (in seconds) with 12 threads were evaluated for **top** serine peptidase motif query and **bottom** zinc-finger motif with 3 residues.



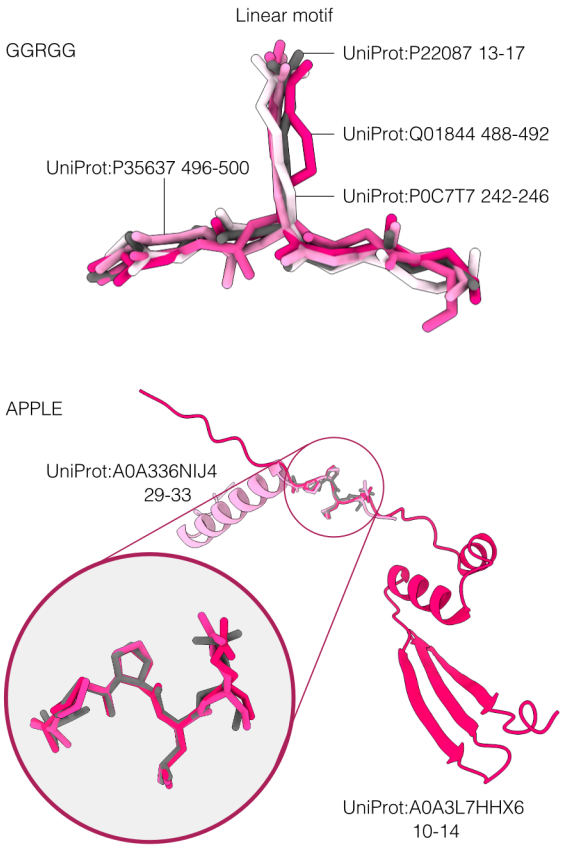
Supplementary Figure 1. Runtime benchmarking of Folddisco index construction across databases and CPU cores. Index building time (in seconds) was measured for four databases — PDB, Swiss-Prot, human subset of the AFDB-proteome, and AFDB50 cluster representatives — using 1, 12, 32, 64, and 128 CPU cores.



Supplementary Figure 2. Partial zinc finger motifs identified in ubiquitin-specific peptidases. Folddisco detected partial matches of the zinc finger motif (**top**) in ubiquitin-specific peptidases USP42 and USP17L20 (**middle**). Notably, the residue order is reversed in these matches. AlphaFold3 predictions confirmed zinc coordination within these motifs (**bottom**).



Supplementary Figure 3. Detection of single disulfide bonds and knottin motifs Folddisco identified single disulfide bonds (red) in insulin (**top**) and complex knottin motifs formed by multiple disulfide bonds (**bottom**). From search of a knottin motif in a spider toxin, a conotoxin was retrieved with a partial match.



Supplementary Figure 4. Identification of short linear motifs by Folddisco. Folddisco searched and detected the known "GGRGG" motif (**top**) and a randomly generated "APPLE" motif (**bottom**). The query "APPLE" motif structure (gray) was predicted using ColabFold.