FlatProt: 2D visualization eases protein structure comparison

Tobias Olenyi[1,♦], Constantin Carl[1], Tobias Senoner[1], Ivan Koludarov[1,•,♦], and Burkhard Rost[1,2,•]

1 School of Computation, Information, and Technology (CIT), Department of Informatics, Bioinformatics & Computational Biology, TUM (Technical University of Munich), 85748 Garching/Munich, Germany
2 TUM School of Life Sciences Weihenstephan (WZW), Alte Akademie 8, Freising, Germany

• Equally contributed as senior authors.
♦ Corresponding author: service+flatprot[at]rostlab.org, http://www.rostlab.org/, Tel: +49-289-17-811 (email rost: assistant[at]rostlab.org)

Abstract

Background: Understanding and comparing three-dimensional (3D) structures of proteins can advance bioinformatics, molecular biology, and drug discovery. While 3D models offer detailed insights, comparing multiple structures simultaneously remains challenging, especially on two-dimensional (2D) displays. Existing 2D visualization tools lack standardized approaches for comparing protein families, limiting their utility in structural analysis.

Results: We introduce FlatProt, designed to enhance protein structure comparison through standardized 2D visualization. Combining rotation algorithms with Foldseek structure alignments enables consistent family-wide visualization and comparison. FlatProt efficiently processes proteins of various sizes (2-34,350 residues), demonstrating its effectiveness through applications on Three-Finger Toxins (3FT), Kallikreins (KLK), and multi-domain proteins.

Conclusion: FlatProt provides clear, consistent, user-friendly visualizations that efficiently compare protein structures within and across families. It advances protein structure analysis by enabling standardized structural comparisons at scale.

Availability: GitHub (https://github.com/t03i/FlatProt); Zenodo (https://doi.org/10.5281/zenodo.15264810).

Keywords: protein visualization, bioinformatics tools, protein structure, sequence comparison

Abbreviations: 3D: three-dimensional, **2D**: two-dimensional, **DSSP**: Dictionary of Secondary Structure of Proteins (automatic assignment of secondary structure from 3D co-ordinates), **KLK**: Kallikreins, **3FTx**: Three-Finger Toxins, **ID**: Identifier, **LDDT**: Local Distance Difference Test, **PDB**: the Protein Data Bank (database of 3D coordinates of high-resolution experimental structures), **SCOP**: Structural Classification of Proteins, **SVG**: Scalable Vector Graphics (file format), **UniProtKB**: Universal Protein Resource Knowledgebase (database).

Background

2D visualization of 3D structures to empower comparisons on monitors. Proteins are the machinery of life, playing crucial roles in nearly every biological process. Insights into the intricate details of protein three-dimensional (3D) structures enable predictions of protein behavior, elucidate cellular mechanisms, and facilitate structure-based drug design, thereby

accelerating therapeutic development and improving efficacy [1]. 3D models offer detailed insights, but as our primary way of viewing and analyzing proteins is through two-dimensional (2D) computer displays, 3D models can be cumbersome, especially when comparing many proteins. Identifying specific features across multiple proteins necessitates simplified yet comprehensive visualization ready for 2D displays.

Current tools for protein visualization in 2D include *SSDraw* [2], *ProLego* [3], *Pro-origami* [4], *iCn3D* [5], and the database *2DProts* [6]. While these tools offer valuable visualization capabilities, they have notable limitations. SSDraw flattens protein structures into sequential diagrams and lacks comparative analysis features. Pro-origami and ProLego focus on single-protein visualization, and neither is actively maintained. iCn3D, while providing comprehensive 3D structure analysis with 2D diagrams as a supplementary feature, primarily serves as a web-based visualization platform. 2DProts, while extensive, is restricted to CATH domain classifications and pre-computed models. These constraints limit their utility for broader protein structure analysis, particularly when comparing multiple proteins or analyzing novel, unclassified structures.

To address these challenges, we introduce *FlatProt*, a novel tool designed to enhance protein comparability through standardized 2D visualizations. FlatProt employs advanced domain- and chain-splitting, protein orientation, and visualization techniques, leveraging a SCOP-derived domain family database [7] and *Foldseek* for fast structure alignment [8]. This ensures consistent and adjustable representations of proteins within the same family, facilitating direct comparisons and rapid understanding of structural similarities and differences. Developed to be both intuitive and lightweight, FlatProt simplifies protein structures without losing critical information. It supports efficient comparison of protein domains and chains, making it suitable for large-scale protein datasets and individual protein analysis. By highlighting intra-family variations, FlatProt advances significantly over existing visualization tools, providing a powerful resource for comprehensive protein analysis and comparison.

Here, we demonstrated the efficacy of *FlatProt* through several case studies. We also demonstrate its general applicability, measured through runtime efficiency. The FlatProt tool, documentation, example usages, and further information are available through GitHub (https://github.com/t03i/FlatProt).

Implementation

Data sets

Family mapping from SCOP. We created an initial FlatProt database for family mapping from the *SCOP2* database [7, 9] using the superfamily level of protein classification with 2,816 classes. We identified a representative for each superfamily as the structure with the highest average alignment TM-score [10] to all other family members. This yielded 2,783 representative structures. Although FlatProt uses SCOP [7] by default, users can preset a custom database, allowing individual adjustments while maintaining family-wide consistency. Alternative sources of family assignments, e.g., CATH [11], can be leveraged to derive a custom database described in the GitHub repository.

Benchmarking performance on the human proteome. We assessed how FlatProt scales with protein length using the human proteome from UniProtKB/SwissProt[12] (data accessed July 2024). The dataset initially included 20,420 proteins. We filtered this set by removing 279 proteins with missing coordinates and 814 orphans, yielding 19,327 proteins. Our final evaluation set was constructed by randomly sampling 1,000 proteins (lengths: 16 to 2,663 residues) from this filtered pool, ensuring the sample reflected the overall length distribution.

Kallikreins showcase family visualization. We used the protein family of *Kallikreins* (KLK) to give an example for FlatProt's visualization. KLK is a subgroup of serine proteases characterized by two attached beta sheets. We took a set of 437 proteins (123-226 residues; mean: 219) identified by experts as family members [13]. First, we predicted 3D structures for each protein using ColabFold [14].Then, *Foldseek* [8] clustered the family (coverage threshold: 0.9, minimal sequence identity: 0.5). This identified 68 clusters. We excluded 28 clusters as outliers (containing only one structure), resulting in 40 representative structures. The structures of these representatives and their UniProtKB IDs are attached in the Supporting Online Material (SOM-KLK.zip, Supporting Online Material).

3FTx showcase comparison clarity. We used the protein family of *Three-Finger Toxins (3FTx)* to highlight how FlatProt allows comparisons across many structures. Characterized by a three-finger fold consisting of beta-stranded loops and a central hydrophobic core formed by four conserved disulfide bonds [15], 3FTx proteins vary in the number of cysteine bonds and beta-sheets. Our dataset included 1,427 trimmed structures [16].

Algorithm

FlatProt generates 2D visualizations of protein 3D structures systematically (**Figure *1***). It inputs a 3D structure file in PDB format along with any optional annotations for domains. The tool then uses DSSP [17, 18] to identify and assign secondary structure segments such as helices, strands, and non-regular elements, dubbed *loops* (also referred to as coils). Next, FlatProt breaks down the protein into secondary structure blocks and simplifies each block accordingly. Helices and loops are represented as simple lines, while strands are depicted as triangles. FlatProt offers adjustable detail levels, allowing users to map individual amino acid residues or merge consecutive residues of identical secondary structure into simplified blocks, enabling visualization that emphasizes fine-grained details or overall structural patterns.

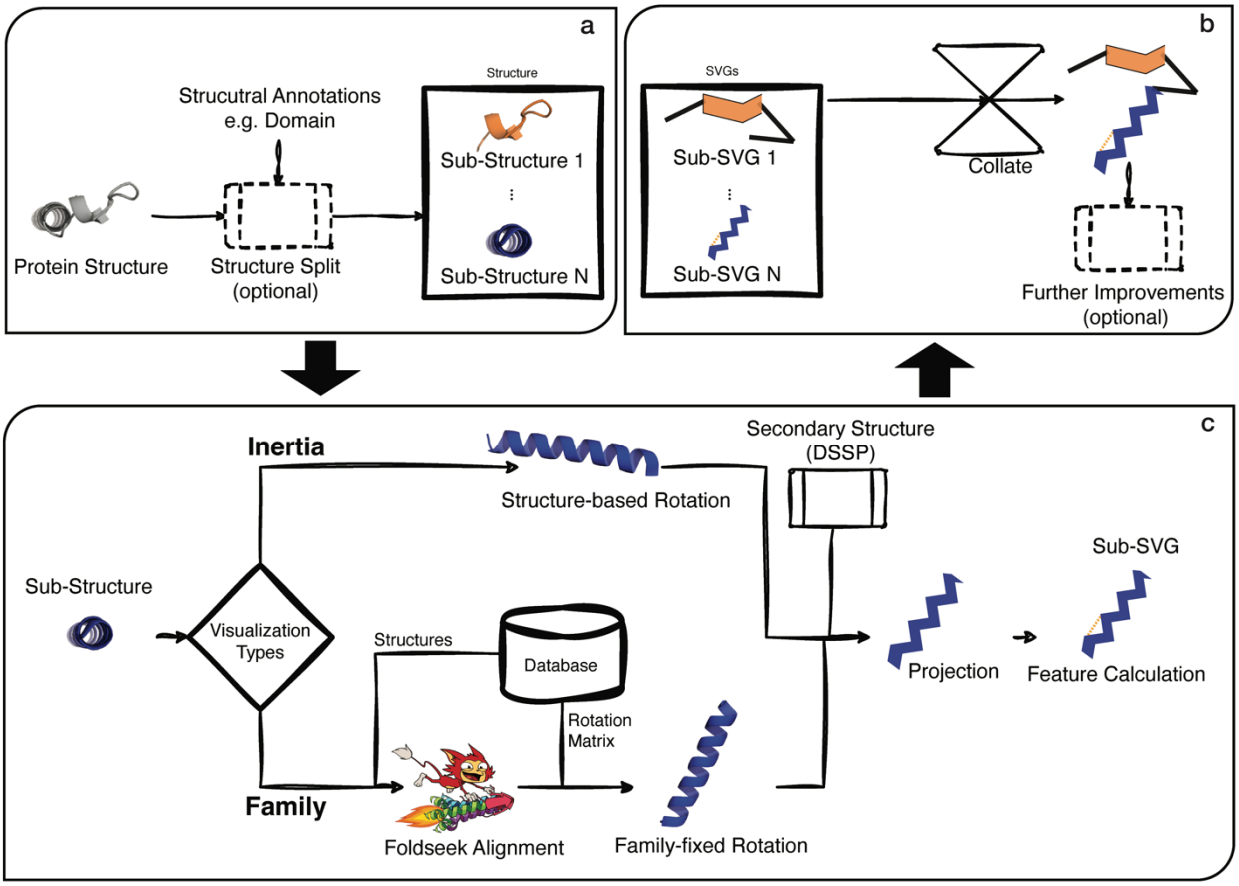

Figure* 1: *Systematic representation of FlatProt pipeline. *(a) Demonstrates how FlatProt processes several sub-structures based on external annotations, such as domain annotations. This process is optional, and the whole structure can be processed simultaneously. (b) Shows how FlatProt collates multiple 2D sub-representations created for each sub-structure. Optionally, these final representations can be collated into a family overlay or overlayed with the previous structure annotations. Colors are illustrative and chosen based on the secondary structure in the original program (c) Illustrates FlatProt's calculation of optimal projections based on the chosen visualization mode. The projection coordinates are then used to project the secondary structure into 2D space. Calculated features, such as disulfide bonds, can be rendered on top of the visualized structure.*

Algorithm

FlatProt systematically visualizes 3D structures in 2D (**Figure *1***). It starts by taking a 3D protein structure file in PDB format as input, and any optional annotations for domains. The tool then uses DSSP [17, 18] to assign secondary structure segments such as helices, strands, and non-regular elements, dubbed *loops* (also referred to as coils). Next, FlatProt breaks down the protein into secondary structure blocks: loops are represented as simple lines, helices as zig-zag area, and strands as triangles. FlatProt offers adjustable detail levels, allowing users to map individual amino acid residues or merge consecutive residues of identical secondary structure into simplified blocks, enabling visualization that emphasizes fine-grained details or overall structural patterns.

FlatProt optimizes the orientation of the 3D structure for the 2D projection using an inertia-based algorithm. It projects the 3D coordinates of the final rotation onto a 2D plane, using only the x and y coordinates. This forms the basis of the 2D representation of the protein structure.

Finally, FlatProt incorporates additional annotations, such as disulfide bonds and highlighted regions, with the latter indicating areas highly divergent from the structures of the entire family as measured by the lDDT-score scores [19].The resulting 2D representation, combining these elements and optimizations, is then saved in support vector graphics (SVG) format.

Structure rotation

Inertia rotation improves 2D representation. FlatProt enhances structural clarity using principal component analysis (PCA) for individual protein visualization, following an approach similar to established tools [20-22]. The algorithm (dubbed *inertia rotation*) aligns the protein's principal components with the coordinate axes, positioning the most considerable variance of atom coordinates along the x-axis, the second largest along the y-axis, and the smallest along the z-axis. This well-established alignment ensures optimal projection onto the 2D plane by maximizing the visible structural information while minimizing feature overlap in the final visualization.

Family-fixed rotation fosters comparability. FlatProt optionally aligns proteins to their respective family representatives through Foldseek 3Di identity searches [8] for consistent family-wide visualizations (dubbed *family rotation*). Users can customize the standard orientation for each family using common tools [20-22], which are stored in FlatProt's database. When visualizing a protein, FlatProt applies the family's predefined orientation after family assignment, ensuring consistent and comparable representations across all family members. This approach mirrors how structural biologists traditionally orient protein families, automating the process for large-scale analyses.

Annotation and visualization options

FlatProt generates a 2D visualization in the standard mode without family information. This provides a clear and simplified view of the secondary structure segments. Users may visualize a protein against its family to highlight the differences through a configurable normalized threshold in the lDDT-score of 0.5. This mode emphasizes regions that are similar throughout the family and those differing between family members to provide insights into structural variations. Users can achieve high comparability between same-family structures by defining preset protein orientations, ensuring consistent and clear visualizations.

Users can also supply domain annotations, enabling the overlay of domain structures on the 2D visualization. This feature helps to focus comparisons on specific domains, enhancing the understanding of domain organization and function.

FlatProt can visualize a protein family as an overlay. This feature allows to compare multiple proteins within the same family in a single 2D representation, highlighting both conserved and variable regions across family members.

FlatProt also supports visualizing other structural features, such as disulfide bonds. These features are calculated using the atom distances in the PDB file, with two sulfur atoms of Cysteine having distances below 3Å (0.3nm) annotated as bonds [23].

Hardware Requirements

We performed all tests on commodity hardware to ensure the accessibility and practicality of FlatProt for typical users. Example analyses were performed on a standard Apple MacBook Pro Intel, 2021, 16GB RAM.

Results and Discussion

Consistent visualization for 3FT family

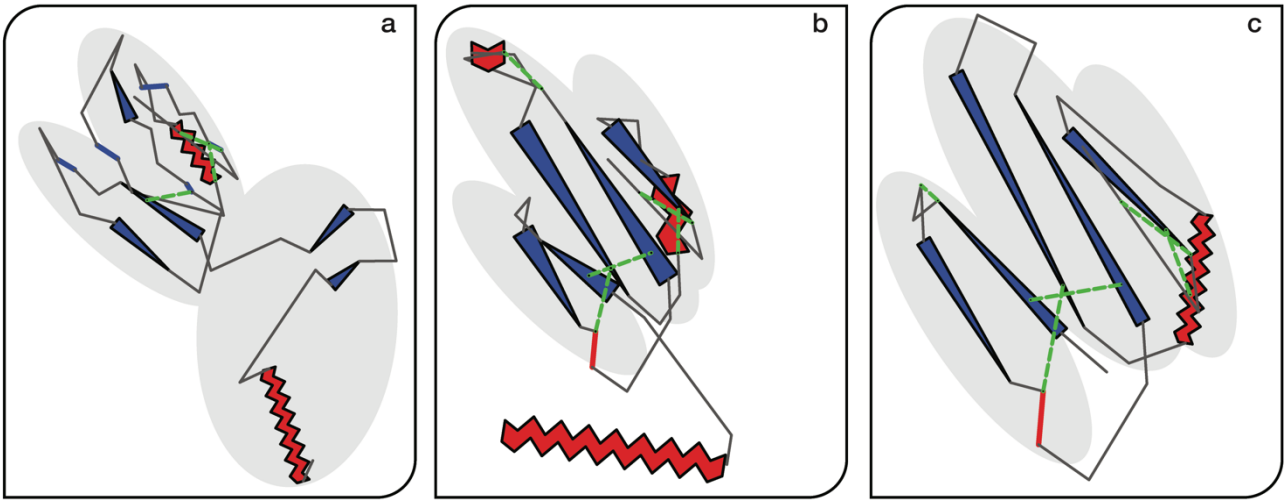

Figure 2: Consistent visualization of 3FTx family: *Each panel highlights the structural variations within the Three-finger Toxins (3FT) family, demonstrating FlatProt's capability to effectively capture and compare these differences. The core three-finger structure is manually shaded for comparison (a) A7X3S2 (present in Trimorphodon biscutatus) variation within the 3FTx family with a rotated outwards finger in the bottom right; (b) P01398 (present in Bungarus multicinctus) a long 3FTx protein (long helix manually rotated for comparability) with a small helix in the second finger; (c) 3FTx protein with typicial three-finger structure (present in Naja Naja [13]).*

Consistent family-wide visualization through standardized alignment. FlatProt achieves a consistent structure representation by aligning structures to family representatives and using a consistent representation. Applied to the protein family known as Three-finger Toxins (3FTxs), this approach reveals both the characteristic three-finger fold while capturing notable variations, such as the unusual two-finger structure in A7X3S2 (**Figure 2a**) and distinct helical elements in other variants (**Figure 2b,c**).

By maintaining consistent orientation and visualization parameters across the family, FlatProt enables direct structural comparison without requiring manual adjustments. Adding visualized cystine bonds (**Figure 2** in yellow) further enhances structural analysis. With generation times of approximately 2.5 seconds per structure, FlatProt efficiently processes entire protein families while maintaining visualization quality across different protein sizes (**Figure 5**).

Domain rotation improves clarity for large proteins

FlatProt effectively renders complex structures such as the 457-residue FKBP-like protein (PDB

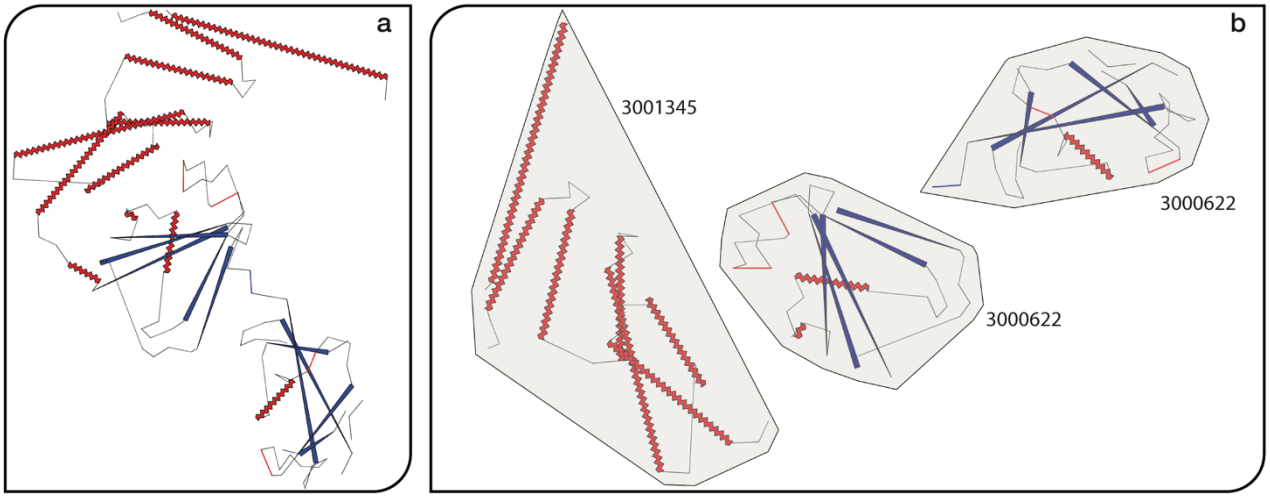

Figure 3: Visualization of 1KT0 Protein with Inertia Rotation and Domain Annotations*. (a) FlatProt visualization of 1KTO using the inertia rotation algorithm (b) Visualization of 1KT0 leveraging FlatProt's domain capabilities with Chainsaw predictions, performing per-domain family visualizations.*

ID 1KT0), involved in human steroid receptor complexes [24]. While the inertia rotation algorithm already provides a clear structural overview (**Figure 3a**) FlatProt can enhance visualization through structural grouping. Shown here is the grouping proposed by the *Chainsaw* domain predictions [25]; any structure grouping can be applied. The visualization splits the FKBP-like protein into three segments (**Figure 3b**), with two sharing the same "Rotamase-like" family classification and identical rotation angles. Each segment is annotated with its family assignment and the Foldseek confidence score, providing additional context for structural analysis.

Visualizing conserved family features through structure overlays

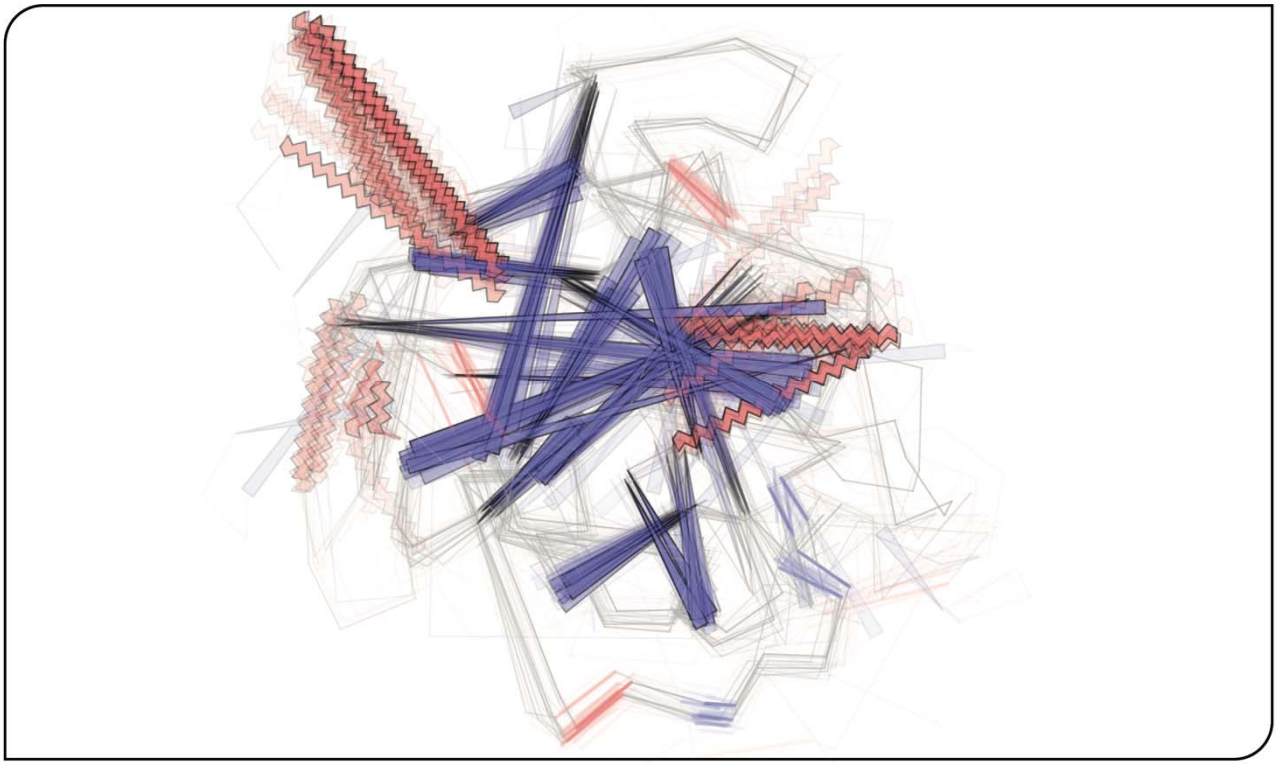

Figure 4: Superposition of traces for KLK family. *Shown are 40 representative structures for the Kallikrein (KLK) family [13] computed with Foldseek clustering. All representations are computed using family rotation and overlaid. High opacity (low transparency) and distinct lines indicate highly conserved structures, while lower opacity signifies fewer structures sharing specific sub-structures.*

We visualized the Kallikrein (KLK) protein family using 40 representative structures [13]. After excluding single-member Foldseek clusters, FlatProt's overlay visualization (**Figure 4**) uniquely enables simultaneous visualization of structural variation across the entire family while preserving the visibility of individual features. This revealed conserved elements: core sheet formations with additional helix structures surrounding them. Higher opacity in the visualization indicates structural conservation, allowing rapid identification of core family features.

Runtime analysis

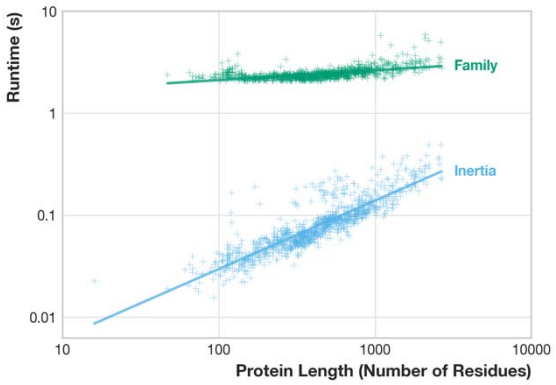

Figure 5: FlatProt Performance Scaling on the Human Proteome. *Runtime characteristics of FlatProt's inertia and family rotations were benchmarked using 1,000 protein structures derived from the human proteome [12]. The plot displays runtime in seconds against protein length in residues on a log-log scale, covering lengths from 16 to 2,663 residues. Individual measurements performed on commodity hardware are shown as crosses for the Inertia mode (n=1,000; blue) and the Family mode (n=962; green). Solid lines indicate the respective linear regression fits in log-log space (Inertia slope=0.67, R^2=0.78; Family slope=0.10, R^2=0.32). The reduced number of data points for the Family mode corresponds to structures for which Foldseek did not find a database match exceeding 50% structural similarity. The visualization highlights the distinct scaling behaviors: stronger length dependence for inertia calculations versus runtime dominated by the family search step (Foldseek).*

We benchmarked FlatProt using 1,000 structures derived from the human proteome [12] to evaluate the performance scaling of its inertia rotation and family rotation modes across protein lengths ranging from 16 to 2,663 residues (**Figure 5**). The analysis on a log-log scale reveals distinct scaling behaviors for the two modes. The runtime for the inertia mode shows a clear dependence on protein length (log-log regression slope = 0.67, R^2=0.78), with runtimes ranging from 0.02 s to 0.49 s. This steeper scaling likely reflects length-dependent computations such as eigenvector decomposition. In contrast, the family visualization mode exhibits noticeably higher absolute runtimes (2.06s-5.89s) but shows weak dependence on protein length (log-log slope = 0.10, R^2=0.32). This is consistent with the runtime being dominated by the highly optimized Foldseek search for family assignment, which scales less dramatically with length than the visualization calculations. The maximum observed runtime for the Family mode remained below 6 seconds, demonstrating practical performance on commodity hardware across the tested range. Notably, Foldseek's runtime could be further decreased by parallelism [8]. The plot includes 1000 points for Inertia mode and 962 for Family mode; the reduction for Family mode reflects instances where Foldseek did not find a match with at least 50% structural similarity in its database.

CONCLUSIONS

We introduced FlatProt, a novel tool designed to enhance the comparability and visualization of protein structures through standardized 2D representations. Building on well-established secondary structure segments and leveraging super-fast structure alignment techniques, FlatProt provides clear and consistent visualizations that are comprehensive and user-friendly. The tool combines internal rotation optimization for individual proteins with family-wide alignments to create standardized visualizations that facilitate structural comparison.

Our analysis demonstrates that FlatProt handles proteins across various sizes and complexities while maintaining efficient performance. By supporting domain annotations and family-specific orientations, FlatProt enables detailed structural analysis at multiple levels - from individual domains to entire protein families. The operation on standard hardware ensures broad accessibility, while its consistent performance across different visualization modes makes it particularly suitable for large-scale protein analysis and comparison.

A key innovation of FlatProt lies in its approach to family-based visualization, allowing researchers to identify both conserved structural elements and variations within protein families. This capability, combined with its support for domain visualization and structural feature annotation, positions FlatProt as a valuable addition to the bioinformatics toolkit. The tool's efficient processing of large structures and its ability to generate consistent family-wide visualizations address significant gaps in current protein structure visualization capabilities.

FlatProt is available with comprehensive documentation and examples on GitHub (https://github.com/t03i/FlatProt). As structural biology advances our understanding of proteins, tools like FlatProt that facilitate efficient visualization and comparison at scale will become increasingly valuable for research in molecular biology, drug design, and related fields.

Availability and requirements

Project name: FlatProt
Project home page: https://github.com/t03i/FlatProt
Operating system(s): Platform independent
Programming language: Python 3.12.5
Other requirements: python version “>=3.10, <3.13”, Foldseek, DSSP (mkdssp version >=4.2.2)
License: Apache 2.0
Any restrictions on use by non-academics: None

Declarations

Availability of data and materials

This published article and its supplementary information files include all data generated or analyzed during this study. The generated database is available on Zenodo https://doi.org/10.5281/zenodo.15264810

Competing interests

The authors declare no competing interests.

Authors' contributions

Tobias Olenyi: Conceptualization, Software, Methodology, Visualization, Writing - Original Draft, Writing - Review & Editing, Project Administration

Constantin Carl: Software, Methodology, Formal Analysis, Writing - Original Draft,

Tobias Senoner: Data Curation, Methodology, Writing - Review & Editing, Project Administration

Ivan Koludarov: Conceptualization, Supervision, Data Curation, Writing - Review & Editing. Funding Acquisition

Burkhard Rost: Supervision, Writing - Review & Editing, Funding Acquisition

Funding

The Bavarian Ministry of Education supported the work of TO, TS, and BR through funding to the TUM. IK and BR were supported by the German Ministry for Research and Education (BMBF: Bundesministerium für Bildung und Forschung); BMBF SSTDBB-16DKWN136

Declaration of generative AI and AI-assisted technologies in the writing process

While preparing this work, the authors used Claude.ai to check the language and grammar of this manuscript. After using this service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Acknowledgments

We thank Nikita Kugut (TUM) for his support with many aspects of this work. We thank the Technical University of Munich (TUM) for providing facilities and resources. Additionally, we are grateful to all contributors to open-source programming libraries. Finally, we thank those who deposit experimental data in public databases, maintain these databases, and develop methods to enrich experimental data.

References

1. O'Donoghue SI, Sabir KS, Kalemanov M, Stolte C, Wellmann B, Ho V, Roos M, Perdigão N, Buske FA, Heinrich J *et al*: **Aquaria: simplifying discovery and insight from protein structures**. *Nat Methods* 2015, **12**(2):98-99.
2. Chen EA, Porter LL: **SSDraw: software for generating comparative protein secondary structure diagrams**. In.; 2023.
3. Khan T, Panday SK, Ghosh I: **ProLego: tool for extracting and visualizing topological modules in protein structures**. *BMC Bioinformatics* 2018, **19**(1):167.
4. Stivala A, Wybrow M, Wirth A, Whisstock JC, Stuckey PJ: **Automatic generation of protein structure cartoons with Pro-origami**. *Bioinformatics* 2011, **27**(23):3315-3316.
5. Wang J, Youkharibache P, Marchler-Bauer A, Lanczycki C, Zhang D, Lu S, Madej T, Marchler GH, Cheng T, Chong LC *et al*: **iCn3D: From Web-Based 3D Viewer to Structural Analysis Tool in Batch Mode**. *Front Mol Biosci* 2022, **9**:831740.
6. Huta□ová Va□eková I, Huta□ J, Midlik A, Horský V, Hladká E, Svobodová R, Berka K: **2DProts: database of family-wide protein secondary structure diagrams**. *Bioinformatics* 2021, **37**(23):4599-4601.
7. Chandonia J-M, Guan L, Lin S, Yu C, Fox Naomi K, Brenner Steven E: **SCOPe: improvements to the structural classification of proteins – extended database to facilitate variant interpretation and machine learning**. *Nucleic Acids Research* 2022, **50**(D1):D553-D559.
8. Van Kempen M, Kim SS, Tumescheit C, Mirdita M, Lee J, Gilchrist CLM, Söding J, Steinegger M: **Fast and accurate protein structure search with Foldseek**. *Nat Biotechnol* 2024, **42**(2):243-246.
9. Fox NK, Brenner SE, Chandonia J-M: **SCOPe: Structural Classification of Proteins—extended, integrating SCOP and ASTRAL data and classification of new structures**. *Nucleic Acids Research* 2014, **42**(D1):D304-D309.
10. Zhang Y, Skolnick J: **Scoring function for automated assessment of protein structure template quality**. *Proteins* 2004, **57**(4):702-710.
11. Waman VP, Bordin N, Lau A, Kandathil S, Wells J, Miller D, Velankar S, Jones DT, Sillitoe I, Orengo C: **CATH v4.4: major expansion of CATH by experimental and predicted structural data**. *Nucleic Acids Res* 2025, **53**(D1):D348-D355.
12. The UniProt C, Bateman A, Martin M-J, Orchard S, Magrane M, Ahmad S, Alpi E, Bowler-Barnett EH, Britto R, Bye-A-Jee H *et al*: **UniProt: the Universal Protein Knowledgebase in 2023**. *Nucleic Acids Research* 2023, **51**(D1):D523-D531.
13. Barua A, Koludarov I, Mikheyev AS: **Co-option of the same ancestral gene family gave rise to mammalian and reptilian toxins**. *BMC Biol* 2021, **19**(1):268.
14. Mirdita M, Schutze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M: **ColabFold: making protein folding accessible to all**. *Nat Methods* 2022, **19**(6):679-682.
15. Kini RM, Doley R: **Structure, function and evolution of three-finger toxins: Mini proteins with multiple targets**. *Toxicon* 2010, **56**(6):855-867.
16. Koludarov I, Senoner T, Jackson TNW, Dashevsky D, Heinzinger M, Aird SD, Rost B: **Domain loss enabled evolution of novel functions in the snake three-finger toxin gene superfamily**. *Nat Commun* 2023, **14**(1):4861.
17. Joosten RP, Te Beek TAH, Krieger E, Hekkelman ML, Hooft RWW, Schneider R, Sander C, Vriend G: **A series of PDB related databases for everyday needs**. *Nucleic Acids Research* 2011, **39**(Database):D411-D419.

18. Kabsch W, Sander C: **Dictionary of protein secondary structure: Pattern recognition of hydrogen‐bonded and geometrical features**. *Biopolymers* 1983, **22**(12):2577-2637.
19. 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* 2013, **29**(21):2722-2728.
20. Meng EC, Goddard TD, Pettersen EF, Couch GS, Pearson ZJ, Morris JH, Ferrin TE: **UCSF ChimeraX : Tools for structure building and analysis**. *Protein Science* 2023, **32**(11):e4792.
21. Schrödinger LLC: **The PyMOL Molecular Graphics System, Version 1.8**. In.; 2015.
22. Schwieters CD, Kuszewski JJ, Tjandra N, Marius Clore G: **The Xplor-NIH NMR molecular structure determination package**. *Journal of Magnetic Resonance* 2003, **160**(1):65-73.
23. Sun M-a, Wang Y, Zhang Q, Xia Y, Ge W, Guo D: **Prediction of reversible disulfide based on features from local structural signatures**. *BMC Genomics* 2017, **18**(1):279.
24. Sinars CR, Cheung-Flynn J, Rimerman RA, Scammell JG, Smith DF, Clardy J: **Structure of the large FK506-binding protein FKBP51, an Hsp90-binding protein and a component of steroid receptor complexes**. *Proc Natl Acad Sci USA* 2003, **100**(3):868-873.
25. Wells J, Hawkins-Hooker A, Bordin N, Sillitoe I, Paige B, Orengo C: **Chainsaw: protein domain segmentation with fully convolutional neural networks**. In.; 2023.