A Snakemake-Based Pipeline for Comprehensive Functional Annotation of Protein Sequences

João Ferreira[1], Andreia Salvador[2], Maria Fernanda Vieira[1]

[1] Department of Informatics, University of Minho, 4710-057 Braga, Portugal
[2] Centre of Biological Engineering, University of Minho, 4710-057 Braga, Portugal

Abstract. The functional annotation of protein sequences remains a challenge in bioinformatics, particularly with the growing influx of uncharacterized sequences from genomic and metagenomic studies. While traditional homology-based methods are effective for well-characterized proteins, they have limitations when applied to novel or divergent sequences. Emerging approaches based on protein structure and machine learning (ML) offer complementary advantages, enabling broader and more accurate functional inference. This project aims to contribute to this evolving field by identifying the most effective open-source tools across homology-based, structure-based, and ML-based strategies, rigorously evaluating their performance, and integrating them into a unified, automated, and reproducible workflow using Snakemake. This approach enhances reproducibility and supports the functional characterization of proteins with limited existing annotations.

Keywords: Functional annotation, protein sequences, workflow, Snakemake, homology-based methods, structure-based inference, machine learning, bioinformatics pipeline, reproducibility, automated annotation

1 Motivation and objectives

The exponential increase in genomic and metagenomic sequencing has led to a growing repository of protein sequences with unknown functions, posing a significant challenge for functional annotation. Traditional sequence homology-based methods are widely used and effective for annotating well-characterized proteins. However, they may be less successful when applied to proteins whose homologs are either uncharacterized or only distantly related, limiting their applicability in some cases. This limitation has driven the need for different approaches, such as leveraging structural analysis and machine learning (ML) techniques, which can infer function based on protein folds or learned patterns beyond sequence similarity. Despite the growing availability of powerful open source tools for protein annotation, whether based on sequence homology, structural analysis, or machine learning (ML), these methodologies are relatively recent and rapidly evolving. Many of these tools show great promise for uncovering functions of previously uncharacterized proteins. Given the complexity of protein annotation, relying on a single method is often insufficient. Instead, combining complementary approaches increases the likelihood of obtaining accurate predictions, since some methods may outperform others depending on the protein, and concordant results across methods can serve as a form of validation. However, such tools are rarely integrated into unified, automated, and reproducible workflows. A standardized and modular pipeline that brings together homology-based, structure-based, and ML-based annotation strategies would thus provide a more comprehensive and scalable solution for protein function prediction. This project aims to contribute to that goal by identifying the most effective open-source tools in each of these three categories, rigorously testing them, and integrating them into a cohesive workflow using a workflow manager such as Snakemake.

2 State of the art

Proteins play a vast array of biological roles, and functional annotation is the process of assigning information to proteins regarding their biochemical activity, cellular localization, and involvement in biological processes [39]. Functional annotation can rely on various sources of evidence, such as sequence similarity, structural characteristics, or patterns learned by computational models [41-43]. In this section, we provide an overview of the key concepts and current methodologies used in protein functional annotation.

2.1 Gene Ontology (GO) Terms and Enzyme Commission (EC) Numbers and Clusters of Orthologous Groups (COGs).

Accurate functional annotation of proteins requires standardized systems that clearly describe and categorize their biological roles. Three widely used systems in bioinformatics and protein annotation are Gene Ontology (GO) terms, Enzyme Commission (EC) numbers, and Clusters of Orthologous Groups (COGs). The Gene Ontology (GO) system provides a structured vocabulary for describing protein functions consistently and precisely. GO categorizes protein functions into three distinct areas: Molecular Function, which describes specific biochemical activities at the molecular level; Biological Process, capturing broader biological objectives or pathways in which a protein participates; and Cellular Component, indicating the specific cellular locations where a protein performs its function. GO terms are structured hierarchically, ranging from general terms to highly specific annotations, facilitating automated and consistent annotations across databases and studies [17]. In contrast, the Enzyme Commission (EC) classification specifically categorizes enzymes based on the chemical reactions they catalyze. EC numbers consist of four digits separated by periods (for example, EC 2.7.11.1), with each digit representing increasingly specific information about the enzyme reaction type. EC numbers provide clarity and precision in biochemical communication, significantly improving enzyme-related annotations, especially in metabolic pathway analyses and bioinformatics applications [18,19].

Another important resource is the Clusters of Orthologous Groups (COGs), which classify proteins into families of orthologous genes based on evolutionary relationships. Orthologous genes are genes found in different species that originated from a common ancestral gene and typically retain similar functions. Each COG represents a group of such genes inferred to be functionally equivalent across multiple organisms [38].

2.2 Workflow Management and Pipeline Development in Bioinformatics

Modern bioinformatics analyses often require the combination of multiple tools and processing steps, applied systematically to large datasets. Constructing such pipelines manually can be time-consuming, error-prone, and difficult to reproduce [26]. To address these challenges, workflow management systems have been developed to streamline the creation, execution, and maintenance of bioinformatics pipelines. These systems allow users to define each analysis step, manage data dependencies, and ensure that workflows are both reproducible and scalable across computing environments.

Among the most widely used workflow systems Snakemake, enables researchers to specify computational steps as modular rules, clearly defining inputs, outputs, and commands [16]. Its flexibility and strong adoption in genomics and proteomics make it particularly suitable for building comprehensive annotation workflows [16]. For this project, Snakemake was selected to integrate the chosen annotation tools into a unified and reproducible pipeline.

2.3 Homology-Based Functional Annotation Approaches

One of the most common and reliable ways to predict the function of a protein is by comparing its sequence to other proteins whose functions are already known [1]. Sequence similarity methods such as BLAST and PSI-BLAST are widely used to identify homologous proteins by comparing input sequences to annotated entries in reference databases. PSI-BLAST enhances sensitivity through iterative refinement of position-specific scoring matrices (PSSMs), improving detection of distant homologs [8].

However, this approach also has certain limitations. While it performs well for proteins that are closely related to well annotated sequences [1,8], it may not always provide reliable annotations for novel or highly divergent proteins lacking characterized homologs in current databases [8]. Nonetheless, homology-based methods remain a valuable first step in many annotation pipelines due to their robustness, interpretability, and widespread support across bioinformatics tools [1-4].

Another approach is domain-based methods that focus on conserved functional regions within proteins rather than full sequences. These regions are identified using profile models from databases like Pfam and CDD, which capture shared patterns across domain families. Tools such as reCOGnizer map query sequences to orthologous groups and curated domain models, enhancing functional annotation with evolutionary information [5–7]. Several existing tools follow this approach and are candidates for integration into the annotation pipeline. These include BLAST and PSI-BLAST (NCBI), which identify homologous sequences through pairwise and iterative alignments [1]; InterProScan, which detects functional motifs and domains using curated databases [27]; eggNOG-mapper, which assigns orthologs and annotations based on precomputed orthologous groups [4]; UPIMAPI, which performs fast similarity searches using DIAMOND and UniProt [7]; reCOGnizer, which refines domain-based annotations using HMMER and Reverse PSI-BLAST [7]; and GO-Figure, which groups and visualizes GO terms based on semantic similarity [20].

Several existing tools follow this approach and are candidates for integration into the annotation pipeline. These include BLAST and PSI-BLAST (NCBI), which identify homologous sequences through pairwise and iterative alignments [1]; InterProScan, which detects functional motifs and domains using curated databases [27]; eggNOG-mapper, which assigns orthologs and annotations based on precomputed orthologous groups [4]; UPIMAPI, which performs fast similarity searches using DIAMOND and UniProt [7]; and reCOGnizer, which refines domain-based annotations using HMMER and Reverse PSI-BLAST [7].

2.4 Structure-Based Functional Annotation Strategies

A powerful way to predict a protein's function is by analyzing its three-dimensional (3D) structure, which is often more conserved than its sequence [9,14]. Structural configurations frequently indicate shared functions, even when sequence similarity is low [13–15].

Traditionally, protein structures were determined using experimental methods like X-ray crystallography or NMR, which are time-consuming and costly [42]. Recent advances, such as AlphaFold2, allow accurate structure prediction from amino acid sequences. This open-source tool has enabled structural modeling for millions of proteins, including many without prior characterization [10,42].

Once a structure is available, it can be compared to known proteins in databases like the Protein Data Bank (PDB). Tools like FoldSeek perform fast alignments to identify proteins with similar structural features, even in the absence of sequence similarity [12]. This is especially useful for annotating proteins with no close homologs.

Structure-based annotation tools increasingly integrate machine learning. For example, DeepFRI represents structures as graphs and applies graph convolutional networks to predict Gene Ontology (GO) terms [15]. COFACTOR combines structure alignment and interaction data to infer GO terms, EC numbers, and ligand-binding sites [28].

Because structural similarity does not always imply identical function, results are often interpreted alongside other evidence, such as gene context or biochemical data [14].

A number of tools support structure-based functional annotation and are being considered for integration into the pipeline. These include DeepFRI, which uses graph-based deep learning on protein structures to predict GO terms [15]; COFACTOR, which combines structure alignment and interaction data to predict GO terms, EC numbers, and ligand-binding sites [28]; and FoldSeek, which performs fast structural alignments to identify proteins with similar 3D folds [12].

2.5 Machine Learning Approaches for Functional Protein Annotation

Machine learning has become a cornerstone in protein annotation, enabling the prediction of protein functions, structures, and interactions directly from amino acid sequences. By training on extensive datasets, machine learning models identify intricate patterns within protein sequences, facilitating accurate annotations [21]. A critical aspect of this process is feature extraction, where models discern relevant characteristics from sequences, such as physicochemical properties, motifs, domains, and evolutionary information. Advanced models, including protein language models, can autonomously learn these features from raw sequences, capturing complex patterns essential for precise predictions [22]. Machine learning applications in protein annotation are diverse. For instance, models can predict the biological roles of proteins by analyzing sequence patterns and comparing them to known functional motifs, assigning Gene Ontology (GO) terms. Tools like DeepGO-SE exemplify this application by leveraging pre-trained language models to predict GO functions from protein sequences [24]. In addition to function prediction, machine learning is also widely applied to other aspects of protein annotation, such as subcellular localization [21]. Predicting a protein's subcellular localization is vital for understanding its function, as many biological activities are compartment specific. Machine learning models analyze signal peptides and localization signals within sequences to predict subcellular compartments, aiding in elucidating protein roles and interactions [25].

Various machine learning-based tools are also being evaluated for inclusion in the pipeline. These include CLEAN, which predicts EC numbers using neural networks trained on enzyme data [29]; DeepLoc 2.0, which predicts subcellular localization using deep learning [25]; DeepGO-SE, which uses sequence embeddings and structural features to predict GO terms [24]; ProteInfer, which uses convolutional neural networks to classify protein functions [31]; and ProtENN, which predicts EC numbers using an ensemble of neural networks [32].

3 Methodology

To construct the functional annotation pipeline, a comprehensive survey and evaluation of bioinformatics tools was conducted, focusing on three main annotation strategies: sequence homology, protein structure, and machine learning. The selection process was guided by criteria such as open-source availability, ease of automation within a Snakemake-based pipeline, demonstrated performance in functional annotation tasks, interpretability of the results, and relevance in the bioinformatics community.

For the sequence homology-based annotation module, three tools were chosen. ReCOGnizer performs domain-based annotations using multiple domain databases and provides reliable GO and EC annotations by leveraging sequence profiles and PSI-BLAST results [7]. UPIMAPI offers a fast and flexible way to retrieve functional annotations from UniProtKB, including GO terms and EC numbers based on DIAMOND similarity searches[7]. EggNOG-mapper provides scalable annotations through precomputed orthologous groups and delivers high-quality outputs for GO, EC, and KEGG terms, making it ideal for large-scale protein datasets[33]. In the structure-based annotation module, DeepFRI was selected for its use of graph convolutional neural networks to predict Gene Ontology (GO) terms from protein 3D structures, offering residue-level resolution and direct applicability to AlphaFold-predicted models [15]. FoldSeek complements DeepFRI by enabling ultra-fast 3D structure alignment, supporting the identification of structural analogs even in the absence of sequence similarity [12]. For the machine learning-based annotation module, the tools selected include DeepGO-SE, which combines protein sequence embeddings with semantic modeling of the Gene Ontology to predict GO terms accurately and consistently[35]. CLEAN was chosen for its use of contrastive learning with deep transformer models trained on curated enzyme data to predict EC numbers, offering high interpretability and classification performance [36]. Finally, ProteInfer was incorporated due to its use of convolutional neural networks trained on large, curated protein datasets to predict functional classes and EC numbers directly from amino acid sequences [37].

All selected tools are open source and can be integrated into Python-based pipelines, ensuring reproducibility and ease of automation within the Snakemake workflow. The pipeline will be implemented using Snakemake, which provides a robust and flexible framework for defining and managing computational workflows. Each tool will be integrated as an independent module, with Snakemake rules clearly specifying inputs, outputs, and commands. The pipeline will begin with a FASTA file provided by the user, containing protein sequences to be annotated. These sequences will be preprocessed as needed and passed through each of the three modules. Outputs will be collected in organized directories by module and tool, allowing for straightforward downstream integration.

4 Implementation

4.1 Tool Architecture

The annotation pipeline was implemented using Snakemake, it allows users to define rules, small modular execution blocks that specify how to generate output files from input files. Snakemake then executes the rules in the correct order, in parallel where possible, ensuring that only the necessary steps are run and only when outputs are missing or outdated.

In this pipeline, the main workflow is organized into independent modules, each responsible for executing one step of the annotation process. The Snakefile is the core of the workflow, it dynamically includes rule definitions based on the user configuration defined in the configuration file.

Each rule explicitly declares its input, output, conda environment, and shell commands. This ensures tasks are executed in the correct order. The user can activate or deactivate specific rules through the configuration file, which also defines input paths and output directories.

To manage tool dependencies, each rule is associated with a dedicated Conda environment declared in separate .yaml files. This method guarantees compatibility between software and libraries, especially critical for tools requiring specific versions of certain packages. When the workflow is run, Snakemake automatically creates these environments as needed, allowing full reproducibility independent of time and across different systems.

The workflow accepts a FASTA-formatted input file and automatically organizes outputs by tool.

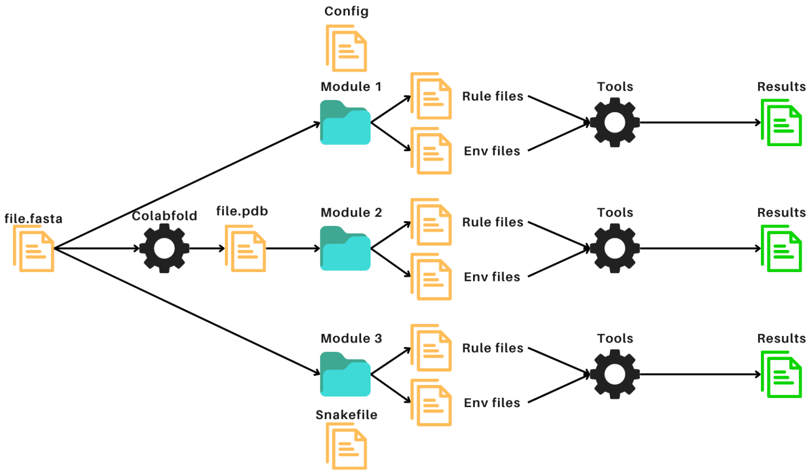

Fig. 1. Pipeline structure

4.2 Tool Integration

The workflow was divided into three distinct modules, each module groups tools according to their underlying annotation strategy.

The homology-based module integrates three tools: reCOGnizer, UPIMAPI, and eggNOG-mapper. These tools require only the input FASTA file with protein sequences and perform annotation based on sequence similarity to reference databases. Since these tools are available through Conda, their integration was relatively straightforward. Each was implemented using a dedicated rule in Snakemake, and their environments were defined in separate YAML files. Additional rules were created to automate the download of the required databases.

The structure-based module includes ColabFold, FoldSeek, and DeepFRI. The tools in this module require protein structures in PDB format. To generate these structures, the input FASTA file was first processed with ColabFold, which predicts 3D structures and outputs PDB files for each sequence. These files were then used as input for FoldSeek and DeepFRI. FoldSeek performs structural alignment against a reference database. In this case, only the PDB database was used, primarily due to storage limitations. Because the pipeline includes multiple tools and large databases, managing disk space was a critical constraint. DeepFRI, on the other hand, is a structure-based annotation tool that incorporates deep learning. It was executed using pre-trained models provided by the original authors, since training custom models was not feasible under the available hardware and time constraints.

The machine learning-based module incorporates CLEAN and DeepGO-SE, both tools were implemented from their publicly available repositories and running them with the pre-trained models supplied by their developers.

The output of each tool is stored in a structured and organized manner under the results directory by default. This directory is automatically creates separate sub-folders for each tool, facilitating easy access and post-processing of the annotation results.

5 Results and discussion

5.1 Annotation Coverage and Tool Comparison

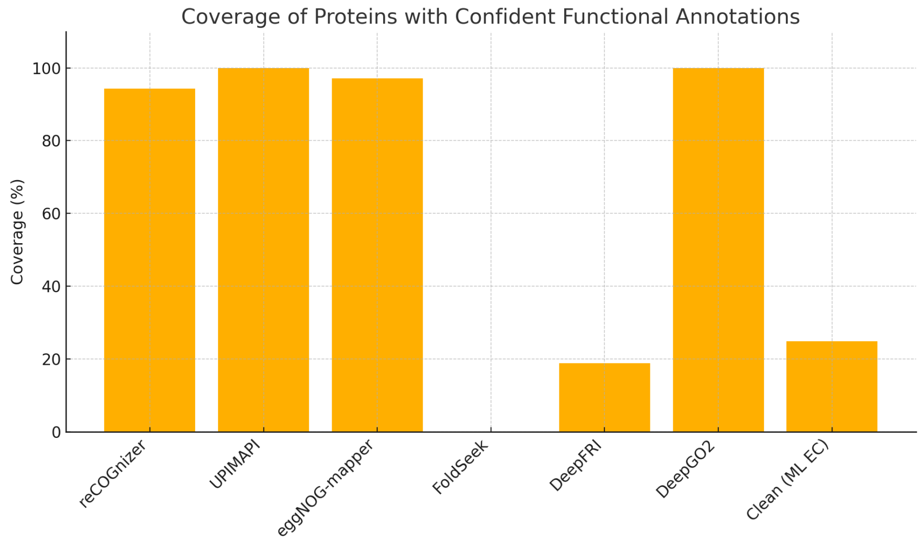

Fig. 2. Annotation Coverage and Tool Comparison

To evaluate the effectiveness of the proposed pipeline, a dataset composed of 1,802 randomly selected protein sequences from the NCBI Protein database was used. This dataset was chosen to facilitate controlled analysis, as most sequences are well-characterized and thus suitable for benchmarking. In future work, the pipeline will be applied to less-characterized or novel proteins, which will better test the limits and complementarity of the implemented tools. To ensure comparability across methods, only confident annotations were considered, defined as having E-values < 0.05 or scores > 0.5. The following section presents a general overview of the functional annotation results obtained through the pipeline. For readers interested in a more detailed data a comprehensive tabular data file is provided in the supplementary materials section below.

A comparative overview of annotation coverage per tool is presented in Figure 2. Among tools based on homology, UPIMAPI achieved the highest coverage, annotating 99.94% of the input proteins. This is likely due to its use of the comprehensive UniProt database and fast DIAMOND aligner. EggNOG-mapper also performed strongly, annotating 97.11% of proteins and enriching the output with GO, KEGG, and orthology-based annotations among other terms. ReCOGnizer annotated 94.29% of sequences based on domain detection using PSI-BLAST and curated domain databases. It is important to note that the protein dataset was composed of randomly selected sequences from the NCBI, which increases the likelihood of a successful homology-based annotation, as many of these proteins may already have close or identical entries in public databases such as UniProt or RefSeq.

Structure-based tools yielded more limited results. DeepFRI annotated 18.76% of proteins, which is relatively low but still informative. One contributing factor may be the use of pre-trained models, which could not be specifically optimized for the properties of this dataset or the quality of the predicted structures provided by colabfold. In future work, retraining DeepFRI with domain-specific data may improve both coverage and prediction quality. On the other hand, FoldSeek failed to produce any confident annotations under the defined criteria. This may be explained by the fact that the tool was configured to search only against the Protein Data Bank (PDB), which contains approximately 200,000 structures. While comprehensive for model organisms, the PDB lacks predicted structures such as those from AlphaFold, as well as hypothetical or uncharacterized proteins and sequences from non-model organisms in general. Given that the dataset in this study consists of 1,802 randomly selected proteins from the NCBI Protein database—many of which are likely not represented in PDB, this limitation likely contributed to the lack of confident matches. Future improvements could involve integrating more inclusive structural databases such as AlphaFold DB or MGnify, which may significantly enhance the performance of structure-based annotation in this pipeline.

Machine learning-based methods produced mixed results. DeepGO2 achieved 100.0% coverage, assigning at least one high-confidence GO term to every protein. This highlights the strength of deep learning methods that leverage transformer-based sequence embeddings trained on large-scale curated datasets. While CLEAN focused on predicting enzymatic function, annotated 24. 81% of proteins with confident EC numbers. Although this is lower than those of the other tools, it is expected given the narrower functional space it targets.

5.2 Pipeline performance

To assess the overall performance of the pipeline as a functional annotation system, we analysed how many tools provided confident annotations (E-value $<$ 0.05 or score ≥ 0.5) for each protein. As illustrated in Figure 3, the majority of proteins were annotated by exactly four tools, while a smaller subset received annotations from five or more. Only a minimal fraction of proteins were annotated by fewer than three tools.

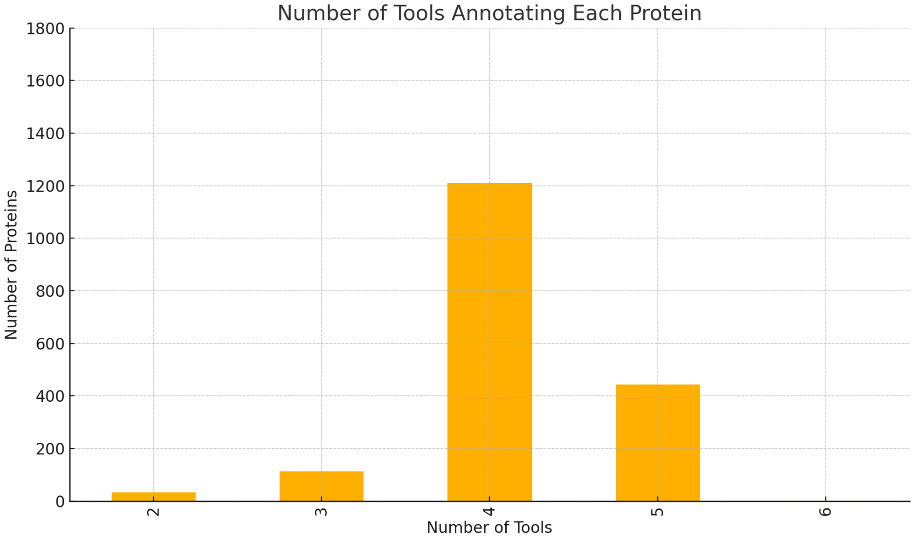

Fig. 3. Number of tools annotating each protein.

This distribution reflects the complementary nature of the tools implemented in the pipeline. Homology-based tools provided broad coverage, while ML-based and structural approaches contributed additional depth or specificity for certain proteins. The presence of multiple annotations per protein increases confidence in the predicted functions and allows users to cross-validate annotations from different sources.

6 Conclusion

The annotation coverage varied significantly across the different modules. The homology-based module achieved the highest coverage, providing confident functional annotations for the vast majority of proteins. This highlights the effectiveness of similarity-based approaches when working with well-characterized sequences. The machine learning-based module also contributed substantially to the annotation process, particularly for proteins lacking significant sequence or structural similarity to known entries. The structure-based module, although still useful for the overall pipeline, yielded limited results in this dataset.

An analysis of cross-tool agreement revealed that most proteins were annotated with confidence by at least four different tools, demonstrating strong complementarity between methods.

7 Pipeline Improvements and Future Work

In the future, this pipeline will be applied to less-characterized or novel proteomes to better assess its performance in real-world scenarios.

Further improvements will involve retraining DeepFRI using custom datasets to enhance the accuracy and specificity of structural predictions, extending the structural annotation module to include more comprehensive databases such as AlphaFold DB, integrating additional tools and implementing a dedicated step to automatically merge all outputs into a single unified annotation report.

Availability of Data and Materials

All datasets, annotation results, and source code used in this project are available at https://github.com/B-Neil/Projeto-Bioinform-tica.

References

1. Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., & Madden, T. L. (2009). BLAST+: architecture and applications. BMC Bioinformatics, 10, 421. https://doi.org/10.1186/1471-2105-10-421
2. Buchfink, B., Xie, C., & Huson, D. H. (2015). Fast and sensitive protein alignment using DIAMOND. Nature Methods, 12(1), 59–60. https://doi.org/10.1038/nmeth.3176
3. Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., & Tanabe, M. (2016). KEGG as a reference resource for gene and protein annotation. Nucleic Acids Research, 44(D1), D457–D462. https://doi.org/10.1093/nar/gkv1070
4. Huerta-Cepas, J., et al. (2016). eggNOG 4.5: a hierarchical orthology framework with improved functional annotations. Nucleic Acids Res, 44(D1), D286–D293. https://doi.org/10.1093/nar/gkv1248
5. Mistry, J., et al. (2021). Pfam: The protein families database in 2021. Nucleic Acids Res, 49(D1), D412–D419. https://doi.org/10.1093/nar/gkaa913
6. Eddy, S. R. (2011). Accelerated profile HMM searches. PLoS Comput Biol, 7(10), e1002195. https://doi.org/10.1371/journal.pcbi.1002195
7. Sequeira, J.C., et al. (2022). UPIMAPI, reCOGnizer and KEGGCharter. Comput Struct Biotechnol J, 20, 1798–1810. https://doi.org/10.1016/j.csbj.2022.03.042

8. Altschul, S.F., et al. (1997). Gapped BLAST and PSI-BLAST. Nucleic Acids Res, 25, 3389–3402.
9. Miao, Y., et al. (2025). GoBERT. arXiv. https://arxiv.org/abs/2501.01930
10. Litfin, T., Zhou, Y., & von Itzstein, M. (2025). Ultra-fast and highly sensitive protein structure alignment. bioRxiv. https://doi.org/10.1101/2025.03.14.643159
11. Kouba, P., et al. (2023). Machine learning-guided protein engineering. ACS Catal, 13(21), 13556–13571. https://doi.org/10.1021/acscatal.3c02743
12. van Kempen, M., et al. (2024). Fast and accurate protein structure search with Foldseek. Nat Biotechnol, 42, 243–246. https://doi.org/10.1038/s41587-023-01773-0
13. Koeppe, J. R., et al. (2025). Expanding the BASIL CURE. [Conference paper]
14. Shulman-Peleg, A., et al. (2004). Recognition of functional sites in protein structures. J Mol Biol, 339(3), 607–633. https://doi.org/10.1016/j.jmb.2004.05.013
15. Gligorijević, V., et al. (2021). Structure-based protein function prediction using GCNs. Nat Commun, 12, 3168. https://doi.org/10.1038/s41467-021-23303-9
16. Mölder, F., et al. (2021). Sustainable data analysis with Snakemake. F1000Res, 10, 33. https://doi.org/10.12688/f1000research.29032.2
17. Ashburner, M., et al. (2000). Gene ontology: Tool for the unification of biology. Nature Genet, 25(1), 25–29. https://doi.org/10.1038/75556
18. Webb, E. C. (1992). Enzyme Nomenclature 1992. Academic Press. ISBN: 978-0122271651.
19. Bairoch, A. (2000). The ENZYME database in 2000. Nucleic Acids Res, 28(1), 304–305. https://doi.org/10.1093/nar/28.1.304
20. Reijnders, M.J.M.F., & Waterhouse, R.M. (2021). Summary Visualizations of Gene Ontology Terms With GO-Figure! Front Bioinform, 1. https://doi.org/10.3389/fbinf.2021.638255
21. Zhou, N., et al. (2019). The CAFA challenge. Nature Methods, 16(7), 603–612. https://doi.org/10.1038/s41592-019-0598-1
22. Vitale, R., et al. (2024). Evaluating large language models for annotating proteins. Brief Bioinform, 25. https://doi.org/10.1093/bib/bbae177
23. Zhou, H., et al. (2021). Protein–protein interactions using supervised learning. Comput Struct Biotechnol J, 19, 2540–2548. https://doi.org/10.1016/j.csbj.2021.04.005
24. Chen, Y., et al. (2022). DeepGO-SE. Comput Struct Biotechnol J, 20, 5550–5561. https://doi.org/10.1016/j.csbj.2022.09.033
25. Thumuluri, V., et al. (2022). DeepLoc 2.0. Nucleic Acids Res, 50, W228–W234. https://doi.org/10.1093/nar/gkac278
26. Di Tommaso, P., et al. (2017). Nextflow enables reproducible workflows. Nat Biotechnol, 35(4), 316–319. https://doi.org/10.1038/nbt.3820
27. Jones, P., et al. (2014). InterProScan 5. Bioinformatics, 30(9), 1236–1240. https://doi.org/10.1093/bioinformatics/btu031
28. Zhang, C., et al. (2017). COFACTOR: Improved protein function prediction. Nucleic Acids Res, 45(W1), W291–W299. https://doi.org/10.1093/nar/gkx366
29. Yu, T., et al. (2023). Enzyme function prediction using contrastive learning. Science, 379, 1358–1363. https://doi.org/10.1126/science.adf2465
30. Elnaggar, A., et al. (2021). ProtTrans. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2021.3095381
31. Sanderson, T., et al. (2023). ProteInfer. eLife, 12, e80942. https://doi.org/10.7554/eLife.80942
32. Bileschi, M.L., et al. (2022). Annotating the protein universe. Nat Biotechnol, 40(4), 532–540. https://doi.org/10.1038/s41587-021-01179-w

33. Cantalapiedra, C.P., et al. (2021). eggNOG-mapper v2. Mol Biol Evol, 38(12), 5825–5829. https://doi.org/10.1093/molbev/msab293
34. Roy, A., et al. (2012). COFACTOR. Nucleic Acids Res, 40(W1), W471–W477. https://doi.org/10.1093/nar/gks372
35. Kulmanov, M., et al. (2024). Protein function prediction as semantic entailment. Nat Mach Intell, 6, 220–228. https://doi.org/10.1038/s42256-024-00795-w
36. Yu, T., et al. (2023). Enzyme function prediction using contrastive learning. Science, 379, 1358–1363. https://doi.org/10.1126/science.adf2465
37. Sanderson, T., et al. (2023). ProteInfer. eLife, 12, e80942. https://doi.org/10.7554/eLife.80942
38. Tatusov, R.L., et al. (2000). The COG database. Nucleic Acids Res, 28(1), 33–36. https://doi.org/10.1093/nar/28.1.33
39. Ouzounis, C.A. (2003). Alternative splicing and functional genomics. Nat Biotechnol, 21(5), 546–553. https://doi.org/10.1038/nbt0503-546
40. Radivojac, P., et al. (2013). Evaluation of protein function prediction. Nat Methods, 10(3), 221–227. https://doi.org/10.1038/nmeth.2340
41. Punta, M., et al. (2012). The Pfam protein families database. Nucleic Acids Res, 40(D1), D290–D301. https://doi.org/10.1093/nar/gkr1065
42. Jumper, J., et al. (2021). AlphaFold. Nature, 596(7873), 583–589. https://doi.org/10.1038/s41586-021-03819-2
43. Gligorijević, V., et al. (2021). Deep learning for protein function prediction. Mol Syst Biol, 17(3), e9882. https://doi.org/10.15252/msb.20209882
44. Zhou, N., et al. (2019). The CAFA challenge (update). Genome Biol, 20(1), 244. https://doi.org/10.1186/s13059-019-1835-8