



Electronic Communications of the EASST
Volume 86 (1) Year 2026

deRSE26 - Selected Contributions of the 6th Conference for Research Software Engineering in Germany (Part 1)

Edited by Bernd Flemisch, René Caspart, Bernadette Fritzsch, Florian Goth, Jean-Noël Grad, Florian Thiery

The Bio-Structure Hub: Accessible AI Structure Prediction for Life Science Researchers

Christine E. Schulz, Inga Ulusoy, Rebecca C. Wade

DOI: 10.14279/eceasst.v86.1.2784

License: This article is licensed under a CC-BY 4.0 License.

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

The Bio-Structure Hub: Accessible AI Structure Prediction for Life Science Researchers

Christine E. Schulz [1,2], **Inga Ulusoy** [1], **Rebecca C. Wade** [1,2,3]

[1] Scientific Software Center, Interdisciplinary Center for Scientific Computing, Heidelberg University, 69120 Heidelberg, Germany
[2] Molecular and Cellular Modeling Group, Heidelberg Institute for Theoretical Studies (HITS), 69118 Heidelberg, Germany
[3] Faculties of Engineering Sciences and Biosciences, Center for Molecular Biology of Heidelberg University (ZMBH), DKFZ-ZMBH Alliance, Heidelberg University, 69120 Heidelberg, Germany

Abstract: AI-based biomacromolecular structure prediction has revolutionized fields from fundamental molecular function studies to drug discovery. However, successfully exploiting these new computational tools requires specialized knowledge, robust compute hardware, and customized software setups, which creates significant accessibility barriers for life scientists. To maximize the scientific potential of these methods, the Scientific Software Center (SSC) at Heidelberg University established the Bio-Structure Hub. The Hub serves as a central infrastructure and accessible entry point, offering optimized AI software installations on suitable resources alongside crucial domain expertise and consultation. The Bio-Structure Hub thus empowers the local research community to optimally utilize state-of-the-art structure prediction methods. In the first 12 months of providing our service we completed 15 projects of various lengths, including long term mentoring and collaborations.

Keywords: biological molecules, structure prediction, research software engineering support

1 Introduction

The three-dimensional structure of a protein, which determines its function, is encoded in its amino-acid sequence.[Anf73] Predicting the structure from a given sequence, known as the folding problem, has been a core challenge since the first atomic-resolution crystal structures were honored with the 1962 Nobel Prize in Chemistry.[DM12] In 1995, the critical assessment of structure prediction (CASP) challenge started to systematically monitor the state of the art in modeling protein structure from sequence.[Bou03, Mou05] The AI revolution of 3D structure prediction started with the performance of AlphaFold2[JEP+21] at CASP14 in 2020.[Cal20] The AlphaFold2 algorithm outperforms other competitors of CASP14 in both easy and difficult targets, providing a consistent folding method to the scientific community.[PSH+21] Both the model and model weights were published open-source. This open source availability led to a series of improvements, such as the AlphaFold-Multimer mode[EOP+22] which was inspired by

user discussion on Twitter / now X (see discussion in [EOP+22]). Using MMseqs2[MSS19] instead of JackHMMER / HHBlits to calculate the Multi-Sequence Alignment (MSA) allowed for a google Colab notebook adaptation termed ColabFold[MSM+22], which led to an even broader availability in the community. Soon, a range of functional AlphaFold2 clones emerged, such as OpenFold[ABF+24] using another AI framework (PyTorch). OpenFold was extended to multimer use and comes with a GoogleColab Notebook.

In May 2024, the last improvement in the AlphaFold series was published: AlphaFold3 and its accompanied AlphaFold Server.[AAD+24] It allows the prediction of not only protein structures, but was able to co-fold small molecules, DNA, and RNA with higher accuracy than specialized tools. In contrast to its predecessors, AlphaFold3 came with a restrictive licensing: Originally published closed source, the AlphaFold3 code and model weights became available for academic use after community backlash.[Cal24] Model weights are available on request and for non-commercial academic use only, and the outputs of the web service AlphaFold3 server cannot be used for subsequent modeling. In addition, the web server only allows a fraction of the functionality that AlphaFold3 is capable of. Small molecules, ions and post-translational modifications cannot be added by using SMILES code, but have to be selected from a drop down menu. Sampling and template options are limited. It should be noted that the options for sampling, templates and small molecules were improved since the first draft of this article was prepared. In November of 2024, the Boltz-1 model was published with the provocative goal to ”democratize bio-molecular interaction modeling”.[WCP+24] It provided the community with an AlphaFold3-level accuracy mode, capable of predicting the same range of outputs. Additionally, they release the training and inference code, model weights, datasets and benchmarks under MIT open license, in stark contrast to AlphaFold3. Subsequently, Boltz-2 was released.[PCW+25] It includes a physical steering capability that reduces the amount of hallucinations and non-physical predictions and includes affinity predictions for co-folded small molecules. However, both Boltz-1 and Boltz-2 only provide the inference model and need to be manually interfaced with a tool to calculate the MSA, such as MMseqs2[MSS19] or the ColabFold MSA server.[MSM+22] Although their installation procedure is easier than AlphaFold3, they do not provide a web server. The Boltz architecture can even be used to design peptide and small molecule binders to protein sequences and rank them via co-folding and affinity predictions.[SFC+25]

With the availability of these methods to the research community, a lack of best practices, tutorials, and guidelines became apparent. This gap has been filled by the EMBL-EBI tutorial series[EE], providing an entry point into structural bioinformatics, as well as a practical guide to AlphaFold2. For AlphaFold3 a more concise tutorial is available on the AlphaFold3 server. For research questions that go beyond these tutorials, interested researchers would turn to local research groups, which in Heidelberg soon experienced an influx in requests that they could not handle. Hence the idea of a central access point for scientific and technical questions around 3D structure prediction and connected methodology was born.

1.1 Existing Solutions

Most users obtain access to the aforementioned structure prediction tools via web servers such as the AlphaFold3 web server[AAD+24] or Google ColabFold web server[KLL+25, MSM+22] that uses AlphaFold2 (multimer)[JEP+21] and MMseqs2[MSS19]. This however raises a few issues: There were so many quick releases of AlphaFold2, that users of web servers reported difficulties to stay in a consistent version while predicting a series of structures. Benchmark evaluations became obsolete within months[ZSKE23], making it difficult to judge the expected quality of a given prediction. The AlphaFold3 web server is an even more obscure black box, making it difficult to recognize the current model version and settings used.[ZWGL26] When software, libraries, or model versions are not explicitly recorded, it becomes impossible to reliably reproduce scientific findings from these experiments. Another argument is data security concerns. The web server requires users to upload proprietary, confidential, or patient-derived data to external servers, exposing those datasets to potential interception, unauthorized access, or inadvertent sharing due to misconfigured cloud storage. A hybrid solution that combines web services with local high performance computing (HPC) access is the ProteinAI service of GWDG[GWD], which places a strong emphasis on the security of user data. However, at present it only provides access to AlphaFold2 and Boltz-1.

An obvious solution to the limited functionality, questionable data privacy and reproducibility is to run the models on a local computer. In principle, a modest workstation can host the software, but realistic use cases quickly exceed the capabilities of a typical lab computer. In addition, installation and upkeep with the latest released models is not a standard task for lab scientists or technicians. Especially when the easy-to-use web server of AlphaFold3 provides an equivalent of a high-end GPU without the trouble. The alternative is the use of local HPC infrastructure, with pre-installed modules for the different prediction methods. For users who are used to the convenience of a web-based graphical interface, moving to a command-line driven HPC environment can seem intimidating, despite the obvious benefits of increased data protection and functionality. In addition, the bwForCluster Helix reported an increased demand of large GPUs by new users, indicating difficulties in the resource selection process. Neither AlphaFold3 nor Boltz-2 provide straight-forward guidance aside from their methodology section[AAD+24, PCW+25]. Independent benchmarks (e.g. [XFQ+25]) focus on the accuracy and limits of the methods, rather than the minimum technical requirements for a given sequence length. Upon addition of small molecules, these requirements vary depending on tokenization, but are not yet systematically assessed. Currently, this problem can only be solved with experience or guidance from experienced users. Hence, an easy access point to the HPC system is needed, that provides pre-installed modules and environments as well as guidance on the resource selections to allow the user full functionality at a simplified interface.

We identified Jupyter notebooks via a JupyterLab interface as a reasonable solution. The success of the ColabFold notebooks[KLL+25, MSM+22] shows that this format works well for non-expert users. Although many HPC centers with JupyterHub connection could easily provide access to AlphaFold3, we found no direct example that comes close to our approach. There are many possible reasons for this. One reason could be the rapidly evolving field that especially

in 2025 produced new models each month. The lack of reliable benchmarks complicates the choice of the right model for a given user base. Keeping up to date with the differences between these models requires a domain expert in the niche between machine learning and computational biology. This profile is likely not available to most HPC centers. At the same time, the restrictive license model of AlphaFold3 complicates the easy pre-installation for HPC users. To this day, each user has to individually request the model parameters from Google and confirm that they adhere to the parameter and output terms of use. Lastly, the most common 3D structure prediction methods require large GPUs. AlphaFold3 built on Jackhmmer[JEP10] requires both CPU access for the MSA prediction and GPU access for the inference, which adds an additional constraint on the hardware.

Aside from existing technical solution, we draw inspiration from existing core facility models. These allow non-expert users to consult with domain experts to get assistance in conducting their experiments or collaborate in a series of experiments conducted by the core facility scientist. A local example in the field of Bioinformatics is the Bioinformatics Core Facility at the Institute of Molecular Biology in Mainz. To our knowledge there is no such service in the field of computational structural biology.

1.2 Project Scope

Here we present the Bio-Structure Hub, an entry point to 3D structure prediction for life science researchers with little or no background in computational biology. On the one hand, we provide expert consultations on project feasibility, choice of methodology and analysis options. On the other hand, we collect resources as well as provide tutorials and Jupyter notebooks, designed for use on the bwForCluster Helix[Hel] via bwVisu[bwv]. Our goal here is to foster independence and curiosity for computational experiments on HPC systems without overwhelming the user. We also do not want to hide functionality and create another blackbox, but enable a transition to traditional HPC usage if the user requires more resources. By providing our examples via GitHub, we also model good software practices for our target users like thorough documentation and testing. In this we aim to improve the documentation and reproducibility of computational 3D structure prediction experiments.

2 Project Results

2.1 Access and Submission

All information is collected on our GitHub website[Bio]: https://ssciwr.github.io/BioStructureHub/ For our tutorials we focus on the bwForCluster Helix[Hel], used for computational structural biology in the bwHPC ecosystem, and its access point bwVisu[bwv]. A schematic overview on how our tutorials integrate with Helix and bwVisu is given in Figure 1. As a cooperation of Heidelberg University, the Karlsruhe Institute of Technology and the High-Performance Computing Center Stuttgart (HLRS), bwVisu was introduced in December 2024 as a web-based service to enable user-friendly execution of interactive application on the supercomputer bwForCluster Helix. bwVisu is based on OpenOnDemand[HJC+18] and features several domain-specific

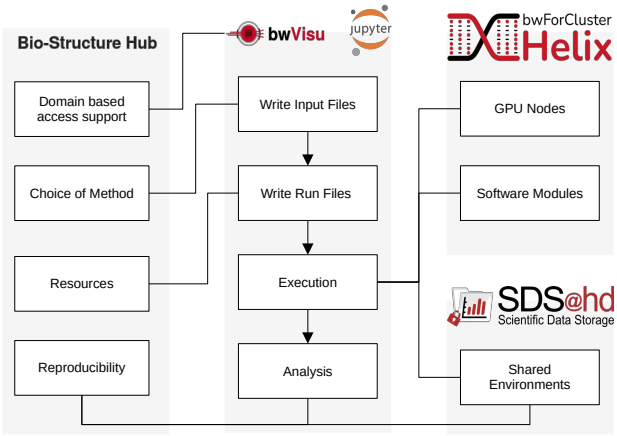


Figure 1: Schematic representation of the involved resources from tutorials to Jupyter notebooks to the underlying clusters.

applications, alongside JupyterLab and RStudio. In principle, our scripts and concepts can be transferred to any other JupyterHub. A key feature is the general user access to bwVisu.[bwv] Users do not require specific proposals for compute time, but can access bwVisu using the storage service SDS@hd.[SDS] This is of particular advantage to experimental research groups starting to work with the HPC system, as SDS@hd is used and available to them. As the environments for the structure prediction are shared via SDS@hd as well, users can even gain access via the Bio-Structure Hub without a separate access request.

To guide researchers during their first structure predictions, we provide step-by-step tutorials on our website[Bio]. We provide help with standard prediction methods like AlphaFold3[AAD+24] and Boltz-2[PCW+25], but also related topics like peptide design using e.g. Boltzgen[SFC+25]. Our tutorials and notebooks cover the entire process from access, to resource selection, to the prediction on the cluster and common analysis tools.

An important focus of these tutorials is the appropriate selection of resources: With bwVisu, users have access to GPUs of type A40 or A100. With a single A100(40GB), AlphaFold3 can run inputs up to 4,352 tokens[AAD+24], with one tolken per amino acid or small molecule atom. While more powerful GPUs are available, this filtering simplifies choices and prevents accidental overbooking. The interactive sessions have priority access in the SLURM job manager to decrease waiting times.

After accessing bwVisu, users simply upload the notebook and their input sequences to their working directory, either as uploaded FASTA files or plain text strings. We provide screenshots on how the final data structure should look like in each tutorial.

2.2 Notebooks

The core of the hub's resources are its Jupyter notebooks. The notebooks we provide can be categorized into two types, execution notebooks and analysis notebooks. Each notebook is intended to stand alone, even without the Tutorial, for example when shared or copied.

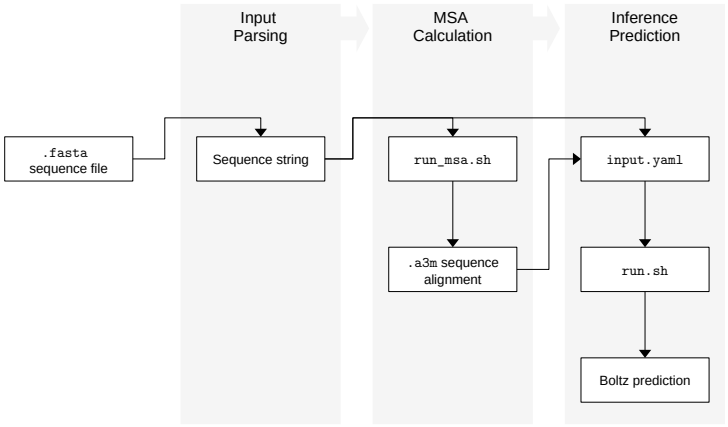


Figure 2: Example workflow for the structure prediction using Boltz2 from .fasta files containing the sequence. The Jupyter notebook executes each of these tasks subsequently.

The general structure of each execution notebook follows a common motif, independent of the prediction method chosen. An example of a structure prediction workflow is given in Figure 2. Here we follow the example of Boltz-2[PCW+25] with MMseqs2[MSS19] to first create the multi-sequence alignment (MSA), followed by the structure prediction. First, using the user provided input, the sequence is extracted for the respective input files. In case of Boltz-2, the notebook contains a template to prepare an input.yaml following the input documentation. For simpler applications like MMseqs2 that require fewer input parameters, these are parsed directly into bash files that runs the execution. The bash files mimic the style of the SLURM syntax, presenting commands that closely resemble those used on the command line, and thereby easing the transition to direct HPC access. These bash files are then executed within the notebook to start the multi-sequence alignment and subsequently the structure prediction. Within the run scripts, we access either pre-installed environments (e.g. for Boltz) or the bwForCluster Helix modules (e.g. for AlphaFold). In case of AlphaFold3, users have to provide the model weights by individually requesting them from Google and uploading them to the cluster. When we prepared the notebooks, we deliberately avoided hiding any functionality from the user. This transparent approach allows the user to get familiar with the HPC workflow rather than treating the prediction as a black box. By exposing each step, we improve both the understanding of the underlying methods and the overall reproducibility. If users later wish to transition to direct access to the HPC systems, only minimal changes to the already generated files are necessary.

Once the structure prediction is successful, we also provide notebooks and Python environments for standard analysis tools, such as visualizing the confidence measures. These notebooks are in the style of typical data analysis notebooks. The library of analysis methods is constantly evolving, depending on the feedback we get from the individual projects.

All notebooks are covered in a CI workflow, supplemented with a test suite and environments to ensure long term functionality of the notebooks and clearly document versions for users. Additionally, the repository setup provides a minimal example for users new to programming on how a Jupyter notebook analysis can be documented and published on Github.

2.3 User Profiles

To contextualize design choices, we include anonymized user profiles based on our consultations and requests to highlight goals, constraints, and interaction patterns.

User 1: As an enthusiastic user, I used the web server extensively and now want to transition to a local HPC system because I need functionality that the web server does not have.
What: Transition from easy to use web servers to the local system, and examples on how to use the full functionality locally.
How: Tutorials and Jupyter notebooks that describe how to access and how to execute the structure prediction.

User 2: As a curious user, I generated a lot of data on the web server and wanted to know how to analyze it all.
What: Tailored (semi-)automatic calculation of properties in an easy to understand format.
How: Easily customizable analysis notebooks.

User 3: As a master student, I want easy-to-understand information on how to add structure predictions to my thesis work.
What: Digestible theory and practical tips on how to predict structures from sequence.
How: Lecture materials and tutorials, as well as curated links to relevant online tutorials and webinars.

User 4: As a PI, I want (someone to work on) preliminary experiments that I need to plan new projects.
What: Preliminary experiments to evaluate project success and apply for necessary funding.
How: Consultation on project feasibility and short term project support.

User 5: As a researcher in biology, I need someone who can introduce me to machine learning and high performance computing, without getting overwhelmed.
What: Digestible background information to foster understanding of the computational experiment
How: Lecture materials and tutorials, as well as curated links to relevant online tutorials and webinars.

3 Status, Challenges and Future Work

The Bio-Structure Hub started in April 2025 with one full-time RSE. Our initial estimate gave highest priority to usable tutorials and notebooks, and the connected HPC access. However, it became obvious that in-person consultations on the design of the predictions and choice of method were the highest demand of the user base. In the first year, we provided over 15 project consultations to evaluate feasibility, resources and strategies of various computational experiments centered around the prediction of biomolecular structures. From this we completed ten small-scale projects that took one to three weeks of development time to complete. These projects centered around functionality that only local installations of prediction methods provide, such as custom molecules or the use of experimental structures as templates in the folding process. On the analysis spectrum, projects ranged from providing custom scripts to compare relative positions in complex predictions, to scaling existing analysis workflows to evaluate hundreds of structure combinations.

We also offered long-term mentorship to two PhD students for recurring support. In this we observed an increased benefit of an external consultant compared to a co-PI, as we provide independent, solution oriented feedback depending on the skill and commitment level of the student. Lastly, we provided seminars and training as stand-alone sessions on how to use 3D structure prediction methods or within the context of general molecular modeling methods as block courses. We observed an increased interest from students during their Bachelor and Master studies, which we attribute to the growing interest and demand in interdisciplinary research. This demographic profits explicitly from the easy-to-access computational tools we provide, as they often lack support of a PI to gain access to HPC traditionally.

Now we are extending our service beyond the Heidelberg research campus. The use of bwVisu is limited to the user base of the bwHPC clusters, specifically academic users at research institutions in Baden-Württemberg. The concept of the Bio-Structure Hub is in principle transferable to other Jupyter hubs, as long as the required resources, in particular the GPUs, are available. We also hope to inspire professionals in other disciplines to use tools like Jupyter notebooks to improve the availability for computational tools for non-standard users.

Acknowledgements: The Bio-Structure Hub is enabled by the Klaus-Tschira Stiftung through project number 00.026.2024, Scientific Software funding line. The authors acknowledge support by the state of Baden-Württemberg through bwHPC and the German Research Foundation (DFG) through grant INST 35/1597-1 FUGG. The authors gratefully acknowledge the data storage service SDS@hd supported by the Ministry of Science, Research and the Arts Baden-Württemberg (MWK) and the German Research Foundation (DFG) through grant INST 35/1503-1 FUGG. The authors acknowledge support by the state of Baden-Württemberg through bwVisu.

Data availability: The code for the Bio-Structure Hub website (including tutorials) and notebooks (including environments) can be found at https://github.com/ssciwr/BioStructureHub.

Declaration on the use of AI: All AI programs used in this work have been cited in the text and in the bibliography. In this document AI assistance was undertaken solely for grammatical improvements.

Bibliography

[AAD+24] J. Abramson, J. Adler, J. Dunger, R. Evans, T. Green, A. Pritzel, O. Ronneberger, L. Willmore, A. J. Ballard, J. Bambrick, S. W. Bodenstein, D. A. Evans, C. C. Hung, M. O'Neill, D. Reiman, K. Tunyasuvunakool, Z. Wu, A. Žemgulytė, E. Arvaniti, C. Beattie, O. Bertolli, A. Bridgland, A. Cherepanov, M. Congreve, A. I. Cowen-Rivers, A. Cowie, M. Figurnov, F. B. Fuchs, H. Gladman, R. Jain, Y. A. Khan, C. M. Low, K. Perlin, A. Potapenko, P. Savy, S. Singh, A. Stecula, A. Thillaisundaram, C. Tong, S. Yakneen, E. D. Zhong, M. Zielinski, A. Žídek, V. Bapst, P. Kohli, M. Jaderberg, D. Hassabis, J. M. Jumper. Accurate structure prediction of biomolecular interactions with AlphaFold 3. *Nature* 630(8016):493–500, June 2024.
doi:10.1038/s41586-024-07487-w

[ABF+24] G. Ahdritz, N. Bouatta, C. Floristean, S. Kadyan, Q. Xia, W. Gerecke, T. J. O'Donnell, D. Berenberg, I. Fisk, N. Zanichelli, B. Zhang, A. Nowaczynski, B. Wang, M. M. Stepniewska-Dziubinska, S. Zhang, A. Ojewole, M. E. Guney, S. Biderman, A. M. Watkins, S. Ra, P. R. Lorenzo, L. Nivon, B. Weitzner, Y.-E. A. Ban, S. Chen, M. Zhang, C. Li, S. L. Song, Y. He, P. K. Sorger, E. Mostaque, Z. Zhang, R. Bonneau, M. AlQuraishi. OpenFold: retraining AlphaFold2 yields new insights into its learning mechanisms and capacity for generalization. *Nature Methods* 21(8):1514–1524, Aug. 2024.
doi:10.1038/s41592-024-02272-z
https://www.nature.com/articles/s41592-024-02272-z

[Anf73] C. B. Anfinsen. Principles that Govern the Folding of Protein Chains. *Science* 181(4096):223–230, July 1973.
doi:10.1126/science.181.4096.223
https://www.science.org/doi/10.1126/science.181.4096.223

[Bio] Bio-Structure Hub.
https://ssciwr.github.io/BioStructureHub/

[Bou03] P. E. Bourne. CASP and CAFASP Experiments and Their Findings. In *Structural Bioinformatics*. Pp. 499–507. John Wiley & Sons, Ltd, 2003.
doi:10.1002/0471721204.ch24
https://onlinelibrary.wiley.com/doi/abs/10.1002/0471721204.ch24

[bwv] bwVisu - University Computing Centre Heidelberg.
https://www.urz.uni-heidelberg.de/en/service-catalogue/software-and-applications/bwvisu

[Cal20] E. Callaway. 'It will change everything': DeepMind's AI makes gigantic leap in solving protein structures. *Nature* 588(7837):203–204, Dec. 2020.
doi:10.1038/D41586-020-03348-4

[Cal24] E. Callaway. Who will make AlphaFold3 open source? Scientists race to crack AI model. *Nature* 630(8015):14–15, May 2024.
doi:10.1038/d41586-024-01555-x
https://www.nature.com/articles/d41586-024-01555-x

[DM12] K. A. Dill, J. L. MacCallum. The Protein-Folding Problem, 50 Years On. *Science* 338(6110):1042–1046, Nov. 2012.
doi:10.1126/science.1219021
https://www.science.org/doi/full/10.1126/science.1219021

[EE] EMBL-EBI. AlphaFold - A practical guide.
https://www.ebi.ac.uk/training/online/courses/alphafold

[EOP+22] R. Evans, M. O'Neill, A. Pritzel, N. Antropova, A. Senior, T. Green, A. Žídek, R. Bates, S. Blackwell, J. Yim, O. Ronneberger, S. Bodenstein, M. Zielinski, A. Bridgland, A. Potapenko, A. Cowie, K. Tunyasuvunakool, R. Jain, E. Clancy, P. Kohli, J. Jumper, D. Hassabis. Protein complex prediction with AlphaFold-Multimer. *bioRxiv*, p. 2021.10.04.463034, Mar. 2022.
doi:10.1101/2021.10.04.463034
https://www.biorxiv.org/content/10.1101/2021.10.04.463034v2

[GWD] GWDG. Protein AI :.
https://docs.hpc.gwdg.de/services/ai-services/protein-ai/index.html

[Hel] bwForCluster Helix - University Computing Centre Heidelberg.
https://www.urz.uni-heidelberg.de/en/service-catalogue/
high-performance-computing/bwforcluster-helix

[HJC+18] D. Hudak, D. Johnson, A. Chalker, J. Nicklas, E. Franz, T. Dockendorf, B. L. McMichael. Open OnDemand: A web-based client portal for HPC centers. *Journal of Open Source Software* 3(25):622, May 2018.
doi:10.21105/joss.00622
https://joss.theoj.org/papers/10.21105/joss.00622

[JEP10] L. S. Johnson, S. R. Eddy, E. Portugaly. Hidden Markov model speed heuristic and iterative HMM search procedure. *BMC Bioinformatics* 11(1):431, Aug. 2010.
doi:10.1186/1471-2105-11-431
https://doi.org/10.1186/1471-2105-11-431

[JEP+21] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko, A. Bridgland, C. Meyer, S. A. Kohl, A. J. Ballard, A. Cowie, B. Romera-Paredes, S. Nikolov, R. Jain, J. Adler, T. Back, S. Petersen, D. Reiman, E. Clancy, M. Zielinski, M. Steinegger, M. Pacholska,

T. Berghammer, S. Bodenstein, D. Silver, O. Vinyals, A. W. Senior, K. Kavukcuoglu, P. Kohli, D. Hassabis. Highly accurate protein structure prediction with AlphaFold. *Nature* 596(7873):583–589, July 2021.
doi:10.1038/s41586-021-03819-2
https://www.nature.com/articles/s41586-021-03819-2

[KLL+25] G. Kim, S. Lee, E. Levy Karin, H. Kim, Y. Moriwaki, S. Ovchinnikov, M. Steinegger, M. Mirdita. Easy and accurate protein structure prediction using ColabFold. *Nature Protocols* 20(3):620–642, Mar. 2025.
doi:10.1038/s41596-024-01060-5
https://www.nature.com/articles/s41596-024-01060-5

[Mou05] J. Moult. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Current Opinion in Structural Biology* 15(3):285–289, June 2005.
doi:10.1016/j.sbi.2005.05.011
https://www.sciencedirect.com/science/article/pii/S0959440X05000953

[MSM+22] M. Mirdita, K. Schütze, Y. Moriwaki, L. Heo, S. Ovchinnikov, M. Steinegger. ColabFold: making protein folding accessible to all. *Nature Methods* 19(6):679–682, May 2022.
doi:10.1038/s41592-022-01488-1
https://www.nature.com/articles/s41592-022-01488-1

[MSS19] M. Mirdita, M. Steinegger, J. Söding. MMseqs2 desktop and local web server app for fast, interactive sequence searches. *Bioinformatics* 35(16):2856–2858, Aug. 2019.
doi:10.1093/bioinformatics/bty1057
https://doi.org/10.1093/bioinformatics/bty1057

[PCW+25] S. Passaro, G. Corso, J. Wohlwend, M. Reveiz, S. Thaler, V. R. Somnath, N. Getz, T. Portnoi, J. Roy, H. Stark, D. Kwabi-Addo, D. Beaini, T. Jaakkola, R. Barzilay. Boltz-2: Towards Accurate and Efficient Binding Affinity Prediction. *bioRxiv*, p. 2025.06.14.659707, June 2025.
doi:10.1101/2025.06.14.659707
https://www.biorxiv.org/content/10.1101/2025.06.14.659707v1

[PSH+21] J. Pereira, A. J. Simpkin, M. D. Hartmann, D. J. Rigden, R. M. Keegan, A. N. Lupas. High-accuracy protein structure prediction in CASP14. *Proteins: Structure, Function, and Bioinformatics* 89(12):1687–1699, Dec. 2021.
doi:10.1002/PROT.26171
https://onlinelibrary.wiley.com/doi/full/10.1002/prot.26171

[SDS] SDS@hd – Scientific Data Storage - University Computing Centre Heidelberg.
https://www.urz.uni-heidelberg.de/en/service-catalogue/storage/sdshd-scientific-data-storage

[SFC+25] H. Stark, F. Faltings, M. Choi, Y. Xie, E. Hur, T. O'Donnell, A. Bushuiev, T. Uçar, S. Passaro, W. Mao, M. Reveiz, R. Bushuiev, T. Pluskal, J. Sivic, K. Kreis, A. Vahdat, S. Ray, J. T. Goldstein, A. Savinov, J. A. Hambalek, A. Gupta, D. A. Taquiri-Diaz, Y. Zhang, A. K. Hatstat, A. Arada, N. H. Kim, E. Tackie-Yarboi, D. Boselli, L. Schnaider, C. C. Liu, G.-W. Li, D. Hnisz, D. M. Sabatini, W. F. DeGrado, J. Wohlwend, G. Corso, R. Barzilay, T. Jaakkola. BoltzGen: Toward Universal Binder Design. *bioRxiv*, p. 2025.11.20.689494, Nov. 2025.
doi:10.1101/2025.11.20.689494
https://www.biorxiv.org/content/10.1101/2025.11.20.689494v1

[WCP+24] J. Wohlwend, G. Corso, S. Passaro, M. Reveiz, K. Leidal, W. Swiderski, T. Portnoi, I. Chinn, J. Silterra, T. Jaakkola, R. Barzilay. Boltz-1: Democratizing Biomolecular Interaction Modeling. *bioRxiv*, p. 2024.11.19.624167, Dec. 2024.
doi:10.1101/2024.11.19.624167
https://www.biorxiv.org/content/10.1101/2024.11.19.624167v2

[XFQ+25] S. Xu, Q. Feng, L. Qiao, H. Wu, T. Shen, Y. Cheng, S. Zheng, S. Sun. Benchmarking all-atom biomolecular structure prediction with FoldBench. *Nature Communications* 17(1):442, 2025.
doi:10.1038/s41467-025-67127-3
https://www.nature.com/articles/s41467-025-67127-3

[ZSKE23] W. Zhu, A. Shenoy, P. Kundrotas, A. Elofsson. Evaluation of AlphaFold-Multimer prediction on multi-chain protein complexes. *Bioinformatics* 39(7):btad424, July 2023.
doi:10.1093/bioinformatics/btad424
https://doi.org/10.1093/bioinformatics/btad424

[ZWGL26] J. Zhang, S. Wang, L. Gao, Q. Li. Deep learning revolutionizes protein research: Advances in structure prediction, functional annotation, and engineered design. *Journal of Biotechnology* 414:91–110, June 2026.
doi:10.1016/j.jbiotec.2026.03.012
https://www.sciencedirect.com/science/article/pii/S0168165626000854