PySSA: end-user protein structure prediction and visual analysis with ColabFold and PyMOL

Hannah Kullik; Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665, Recklinghausen, Germany; hannah.kullik@studmail.w-hs.de; ORCID: 0009-0004-5129-1298

Martin Urban; Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665, Recklinghausen, Germany; martin.urban@studmail.w-hs.de; ORCID: 0009-0008-6834-5714

Jonas Schaub; Institute for Inorganic and Analytical Chemistry; Friedrich Schiller University Jena, Lessing Strasse 8, 07743, Jena, Germany; jonas.schaub@uni-jena.de; ORCID: 0000-0003-1554-6666

Angelika Loidl-Stahlhofen; Laboratory of Protein Chemistry, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665, Recklinghausen, Germany; angelika.loidl-stahlhofen@w-hs.de; ORCID: 0000-0002-3158-9546

Achim Zielesny*; Institute for Bioinformatics and Chemoinformatics, Westphalian University of Applied Sciences, August-Schmidt-Ring 10, 45665, Recklinghausen, Germany; achim.zielesny@w-hs.de; ORCID: 0000-0003-0722-4229

*Corresponding author email: achim.zielesny@w-hs.de

Abstract

Computational methods for the accurate prediction of protein folding based on amino acid sequences have been researched for decades. The field has been significantly advanced in recent years by deep learning-based approaches, like AlphaFold, RoseTTAFold, or ColabFold. Although these can be used by the scientific community in various, mostly free and open ways, they are not yet widely used by bench scientists in relevant fields such as protein biochemistry or molecular biology, who are often not familiar with software tools such as scripting notebooks, command-line interfaces or cloud computing. In addition, visual inspection functionalities like protein structure displays, structure alignments, and specific protein hotspot analyses are required as a second step to interpret and apply the predicted structures in ongoing research studies.

PySSA (Python rich client for visual protein Sequence to Structure Analysis) is an open Graphical User Interface (GUI) application combining the protein sequence to structure prediction capabilities of ColabFold with the open-source variant of the molecular structure visualisation and analysis system PyMOL to make both available to the scientific end-user. PySSA enables the creation of managed and shareable projects with defined protein structure prediction and corresponding alignment workflows that can be conveniently performed by scientists without specialised computer skills or programming knowledge on their local computers. Thus, PySSA can help make protein structure prediction more accessible for end-users in protein chemistry and molecular biology as well as be used for educational purposes. It is openly available on GitHub, alongside a custom graphical installer executable for the Windows operating system: https://github.com/urban233/PySSA/wiki/Installation-for-Windows-Operating-System.

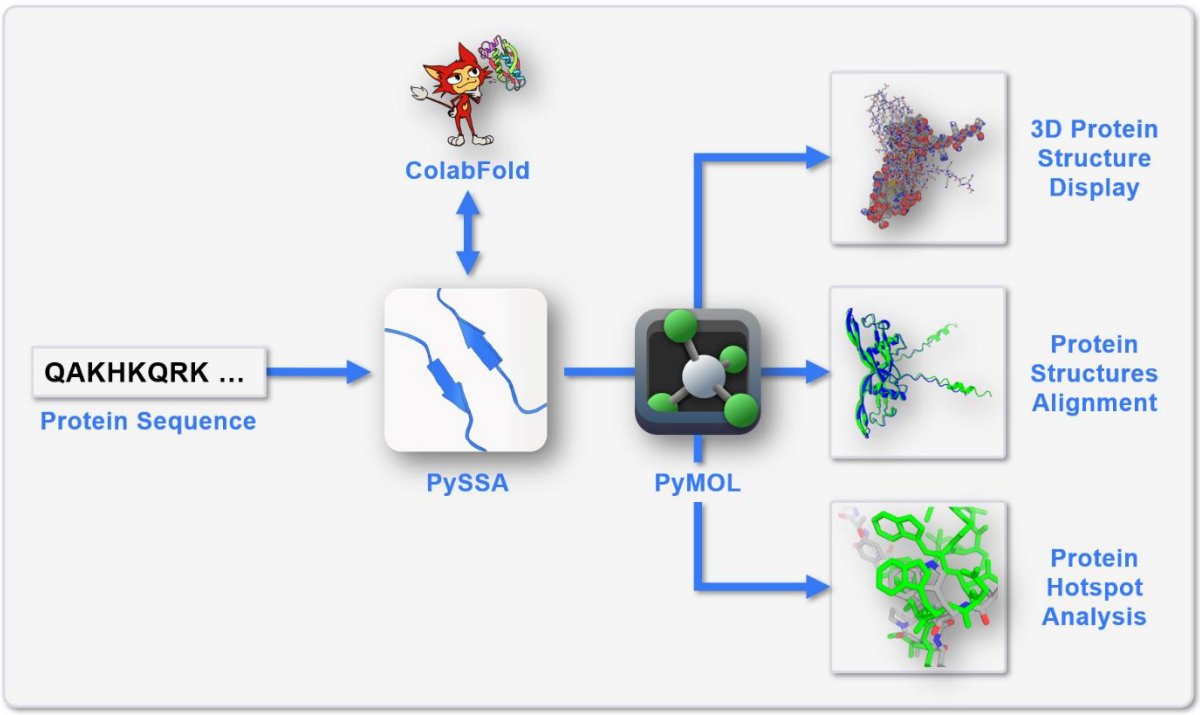
To demonstrate the capabilities of PySSA, its usage in a protein mutation study on the protein drug Bone Morphogenetic Protein 2 (BMP2) is described: the structure prediction results indicate that the previously reported BMP2-2Hep-7M mutant, which is intended to be less prone to aggregation, does not exhibit significant spatial rearrangements of amino acid residues interacting with the receptor.

Scientific contribution

PySSA (Python rich client for visual protein Sequence to Structure Analysis) is designed to provide accurate protein structure prediction based on amino acid sequences to the scientific end-user via its straightforward local installation, graphical user interface, and visual protein structure analysis functionalities, including three-dimensional display, structure alignment, and hotspot analysis. To this end, it combines ColabFold for structure predictions with the

visualisation capabilities of open-source PyMOL in an openly available, Python-based rich client application that can be installed and used without special computer knowledge or programming skills.

Graphical abstract



Keywords

protein structure prediction, protein structure alignment, protein analysis, AlphaFold, ColabFold, PyMOL, Bone Morphogenetic Protein, BMP, BMP2

Introduction

In biological systems, proteins play a pivotal role as enzymes, receptors, transporters, channels, or structural elements. They are also the most important targets for pharmacological interventions and medical treatments. Through increased research in genomics, transcriptomics, and proteomics over the last decades, the amino acid sequences of most human proteins are readily accessible in public databases, while their three-dimensional spatial structures are mostly undetermined. To illustrate, the UniProtKB/Swiss-Prot protein sequence database lists 20,434 human protein sequences in their second release version of 2024 [1, 2], whereas the Protein Data Bank (PDB) [3] reports 3,489 protein structures with *homo sapiens* as natural source organism at the time of writing of this manuscript [4], indicating that only about 17 % of the human proteome is structurally elucidated. Consequently, as the experimental methods are time-consuming and cumbersome, the *in silico* prediction of a protein's three-dimensional structure from its amino acid sequence (i.e. the prediction of protein folding *in vivo*) represents one of the fundamental challenges of structural bioinformatics and molecular biology. An atomic accuracy in the order of one angstrom (the typical length of a chemical bond in organic molecules [5]) must be achieved for the predicted structures to be of actual use in research fields like drug discovery.

In the 13th (2018) and 14th (2020) CASP (Critical Assessment of protein Structure Prediction) competition [6], the deep learning-based system AlphaFold (and its version 2 successor) developed by Google's DeepMind [7, 8] demonstrated superior performance to all other competitors, exhibiting significant advancement in the prediction of spatial protein structures based on their amino acid sequences. Subsequently, AlphaFold was employed to create the AlphaFold Protein Structure Database [9], comprising over 200 million predicted spatial protein structures, which collectively encompass almost all known proteins [10]. AlphaFold was able to predict 98.5 % of the protein structures of the human proteome [11], in contrast to only 17 % that were elucidated by experimental research over several decades (see above). While AlphaFold does not fundamentally solve the problem of protein folding, and many serious challenges remain, e.g. the prediction of dynamic protein-protein or protein-small-molecule interactions, its extraordinary impact on accurate protein structure predictions is beyond doubt [12]. The AlphaFold source code and model weights were published openly on GitHub. But deployment and accessibility remained an issue since a local or cloud installation of AlphaFold requires up to 3 TB of disk space and powerful Graphics Processing Units (GPU) to run [13]. Recently, AlphaFold version 3 was released which promises another significant improvement in the field, not just regarding protein structure prediction but also protein, nucleic acid, and small molecule interaction modelling. At the time of writing of this

manuscript, the source code and model weights of AlphaFold 3 have not been published and access has been possible only through a restricted web interface [14-16].

The success of AlphaFold also sparked other projects, like RoseTTAFold, that experimented with different model architectures and different methods of generating Multiple Sequence Alignments (MSA) as inputs for structure prediction [17] and achieved similar prediction accuracy. The open ColabFold software [18, 19] represents a more recent descendant of the AlphaFold breakthrough and exhibits similar prediction quality to AlphaFold 2 on CASP14 targets. It uses AlphaFold 2 or RoseTTAFold internally and can predict protein structures considerably faster by employing MMseqs2 [20] for generating the MSA of an input amino acid sequence which is then passed on to the deep-learning structure prediction models. ColabFold also represents an advancement in deployment and accessibility because it is available through Jupyter notebooks which can be executed (for free, in principle) in Google Colaboratory (Colab) or on a local, stand-alone machine with much fewer hardware requirements. It is also available as a local installation run via a command-line interface or as a Python package hosted on PyPI (Python Package Index) [21, 22]. This opens up the possibility of combining and integrating comparatively fast predictions of protein structures and homo- and heterodimeric complexes with visual structure alignment functionalities in a way that is accessible to the scientific end-user.

PySSA (Python rich client for visual protein Sequence to Structure Analysis) is a Python-based rich client application, intended to seize this opportunity. PySSA provides a Graphical User Interface (GUI) for setting up and executing protein structure predictions based on amino acid sequences which are internally run by AlphaFold 2 inside ColabFold. After prediction, the three-dimensional structures can be visualised and inspected: for its structure display, structure alignment (e.g. predicted structure vs. crystal structure or mutant vs. wild-type of the predicted protein), and protein hotspot analysis capabilities, PySSA employs the established open-source molecular structure visualisation and analysis system PyMOL [23, 24]. All structure prediction, display, and analysis functionalities can be executed on a local computer without specialised computer skills or programming experience via the PySSA GUI. Consequently, PySSA can assist in making protein structure prediction more accessible and adaptable for research and development in protein chemistry and molecular biology. Furthermore, PySSA can also be utilised for educational purposes, as the initial entry barrier is low and extensive documentation illustrates its functions and use cases. The PySSA source code is openly available on GitHub [25] and a convenient graphical installer can be used to deploy it on a local computer.

The initial idea to develop PySSA was conceived while working on a mutation study concerning the protein drug Bone Morphogenetic Protein 2 (BMP2), and its application in this ongoing research is reported here to demonstrate the main functionalities of PySSA. Bone

Morphogenetic Proteins (BMPs) represent the largest subgroup of the growth factor ß family [26] with predominantly dimeric forms, being characterised by a cystine knot for structural stabilisation. They exhibit biological activity as osteogenic growth factors [27] and exert further influences on broad physiological processes, including embryogenesis, myogenesis, and neural development [28, 29]. BMP2, in particular, has aroused significant medical interest due to its applications in bone healing and has been approved as a biological protein drug by the FDA since 2002 [30, 31]. To mitigate the severe adverse effects of aggregation-prone BMP2 [32, 33] and to facilitate its soluble expression in a bacterial host, a hydrophilisation mutation strategy was previously established to eliminate its aggregation hotspots while retaining all known receptor binding sites for the desired physiological activity [34]. In addition, heparin binding and the affinity to extracellular membrane components were improved by doubling the N-terminal heparin binding site. One resulting mutant with seven hydrophilising point mutations and an expanded heparin binding site, BMP2-2Hep-7M, did not form inclusion bodies after expression and could be purified as a soluble dimer with a cystine knot structure [34]. PySSA was used to predict a spatial model of this BMP2-2Hep-7M mutant and to align and inspect it in comparative studies with wild-type BMP2 regarding potential conformational effects of the point mutations or the expanded heparin binding site on the amino acid residues responsible for receptor binding and hence bioactivity.

Implementation

The general architecture of PySSA is described in Figure 1. The Python-based PySSA core acts as a graphical front end for user interaction, communicates with ColabFold to execute protein structure predictions from input amino acid sequences, and hosts PyMOL instances to enable protein structure display and visual analysis. PySSA is generally designed for offline use, but ColabFold internally communicates via the internet with a public web server hosting the MMseqs2 homology search engine [20], provided by the ColabFold/MMseqs2 developers, to generate Multiple Sequence Alignments (MSA) as inputs for the local AlphaFold 2 structure prediction model.

PySSA is exclusively available for the Windows Operating System (OS) version 10 or higher and is implemented as a Model-View-Controller (MVC) design pattern-based rich client application in Python 3.11 [35], using the PyQt5 GUI framework [36] and a local SQLite [37] database for internal data storage. PySSA was developed with Windows as the primary target OS since the platform is most widespread among scientific end-users. The Python-own virtual environment functionality (venv module) is used to create an isolated environment for PySSA and to set up the required libraries. All third-party libraries are obtained via the Python Package Index (PyPI) [21]. PySSA's protein structure prediction capabilities are realised via the ColabFold Python package [22] which is integrated as a microservice. Since ColabFold is based on Linux libraries, the Windows Subsystem for Linux version 2 (WSL2) is utilised with a custom AlmaLinux [38] distribution to create an isolated and standardised computing environment for the ColabFold microservice. PySSA communicates with the ColabFold microservice via the universal messaging library ZeroMQ [39] and the Python subprocess module. Prediction results are copied from the AlmaLinux file system to a directory for temporary files on the Windows host system. PySSA uses open-source PyMOL for molecular structure visualisation in two different instance types run by separate Python interpreters: it hosts a PyMOL user instance for user interaction and structure visualisation and auxiliary PyMOL instances for command execution, running independently of the PyMOL user session in the background. This enables the execution of parallelised tasks for generating ray traced images or for preparing analysis results like structure alignments. For communication between the PySSA core and the Python API of PyMOL via the ZeroMQ library, the PyMOL user instance implements a push/pull pattern, whereas the auxiliary PyMOL instances employ a request/reply pattern.

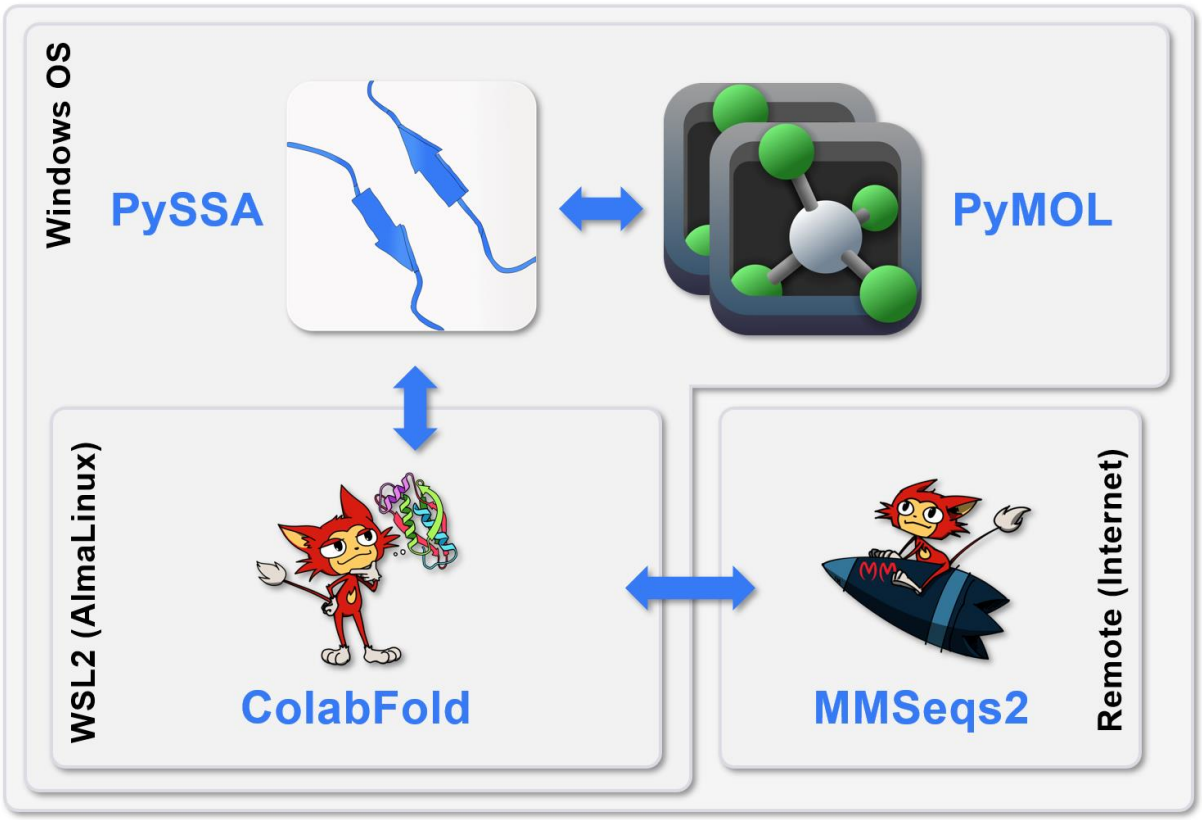


Figure 1. *PySSA architecture on a local computer. The MSA generation for the ColabFold prediction is performed on a remote web server running the MMseqs2 software (ColabFold, MMseqs2, and PyMOL logos taken from the respective GitHub repositories [23, 40, 41]).*

A graphical PySSA Windows installer executable was developed as part of this project for convenient installation of all PySSA components which are downloaded and installed in three successive steps: (1) WSL2 (if not already installed), (2) AlmaLinux and ColabFold, (3) Python 3.11, open-source PyMOL, all other third-party Python libraries, and PySSA itself. ColabFold deployed inside the AlmaLinux instance requires about 20 GB of disk space. The other components' disk space requirements are negligible in comparison to this. Concerning further hardware requirements, PySSA generally runs on the Central Processing Unit (CPU), so a powerful Graphics Processing Unit (GPU) is not required. After successful installation, PySSA can be launched from a desktop icon. The arrangement of the application windows (the PySSA main GUI and the PyMOL user instance) is managed by a compiled WinBatch executable [42].

The complete PySSA documentation is accessible via the *Help* menu in the GUI. More specific, context-based help texts can be accessed within the application via question mark icons or context menus. Moreover, use-case-specific tutorial videos are available that demonstrate PySSA workflows [25]. PySSA automatically logs internal processes using the Python logging API. A log file is written for every session and can be accessed via the *Help*

menu. The complete PySSA code is openly available on GitHub [25] and the PySSA installer project can be found in a separate repository [43].

Results and discussion

PySSA (Python rich client for visual protein Sequence to Structure Analysis) is an open, graphical end-user software application that enables the prediction of spatial monomeric and multimeric protein structures from their amino acid sequences using AlphaFold 2 via ColabFold. The Graphical User Interface (GUI) of PySSA enables scientists without specialised computer skills or programming knowledge to perform spatial protein structure predictions conveniently on their local computers. Protein structures are displayed in PySSA using the molecular visualisation platform PyMOL. Furthermore, the structural alignment functions offered by PyMOL are available for monomeric and multimeric protein pairs in subsequent analysis steps to evaluate their structural match. This feature can be used to study the structural matches of diverse kinds of protein pairs, e.g. wild-type vs. mutant or measured vs. predicted in any combination. All protein prediction and analysis jobs are organised in projects that support management functions and sharing via project file import/export. Projects themselves belong to a workspace directory as the basic level of organisation. Protein sequences can be added to a project via their FASTA file or by copy/paste operations. Existing protein structures for display or alignment can be retrieved via their PDB ID (if an internet connection is available) or imported from PDB files on the local file system. Multiple protein structure predictions with corresponding alignment procedures can also be bundled into a single batch job. However, since ColabFold does not support executing multiple prediction jobs in parallel, multiple predictions can be defined in PySSA and will then be executed one by one automatically. PySSA can export high-quality PNG images of displayed structures with optional ray tracing in high resolution, which can be particularly relevant for the visualisation of significant protein regions. For image generation, a user can select a background colour, different renderers, ray-trace modes, and ray textures in the global settings dialogue. All tabular data displayed in PySSA can be exported as a CSV file. Protein structures can be exported as PDB files.

The main functionalities of PySSA are showcased in the following section with the specific example of a mutation study of Bone Morphogenetic Protein 2 (BMP2).

Application example: BMP2

As preparatory work, the prediction quality of ColabFold for the monomeric form of the BMP2 wild-type variant (denoted BMP2-wt-mono) is investigated: within a new project, the amino acid sequence of BMP2-wt-mono is added for prediction (the corresponding predicted spatial protein structure will have the same name). As a reference for comparison, the previously determined experimental X-ray structure of the BMP2 wild-type homodimer is

included by retrieving it from PDB [44, 45]. The structure with PDB ID 6OMN (and hence named 6OMN in PySSA after retrieval) represents the glycosylated BMP2 wild-type homodimer which is cleared of solvent and sugar molecules by a cleaning function available in PySSA. The spatial structure of BMP2-wt-mono is predicted by setting up a prediction job. A subsequent structural alignment with the 6OMN reference structure is also specified in the job setup via the analysis option. Chain E of the reference 6OMN is compared with chain A of the predicted BMP2-wt-mono structure. Additional settings for the prediction job, that are available in PySSA, include the usage of template protein structures taken from PDB70: together with the MSA of the 20 most similar proteins generated by MMseqs2, their structures can also be included as templates in the AlphaFold 2 prediction input via this setting (enabled by default) [8, 18]. Another option is to perform an AMBER force field relaxation process after structure prediction (enabled by default) [8, 46, 47].

After completion of the calculation, which took about 2 h 15 min on a contemporary desktop computer (AMD Ryzen 5 3400G processor with Radeon Vega Graphics, 4 cores, 8 logical processors, base speed 3.70 GHz, 16 GB RAM; calculations done on CPU), the structural alignment is displayed in a PyMOL base scene. The *Results Summary* dialogue displays quality measures for the protein pair analysis, such as the number of aligned residues and the RMSD value of the structural alignment. The overall C_α RMSD value evaluates to only 0.82 Å, a good result within the experimental 6OMN X-ray resolution of 2.68 Å [44]. An interactive distance diagram of the residue pairs in combination with a distance frequency histogram and a distance table (with residue pair distances, residue number, position, and chain assignment) can be displayed in a separate dialogue window (see Figure 2). All amino acid (C_α) deviations are well within the experimental X-ray resolution as well (see above). The deviations of the five residues which are crucial for the interaction with the BMP2 receptor and hence for triggering the corresponding physiological effect (valine 26, tryptophan 31, proline 50, alanine 52, and serine 69) [48] demonstrate a good match (Table 1, line “BMP2-wt-mono (pred.) vs. 6OMN (exp.)”). Since PySSA appears to be capable of successfully predicting the basic spatial structure of BMP2, it may be utilised for BMP2 mutant investigations.

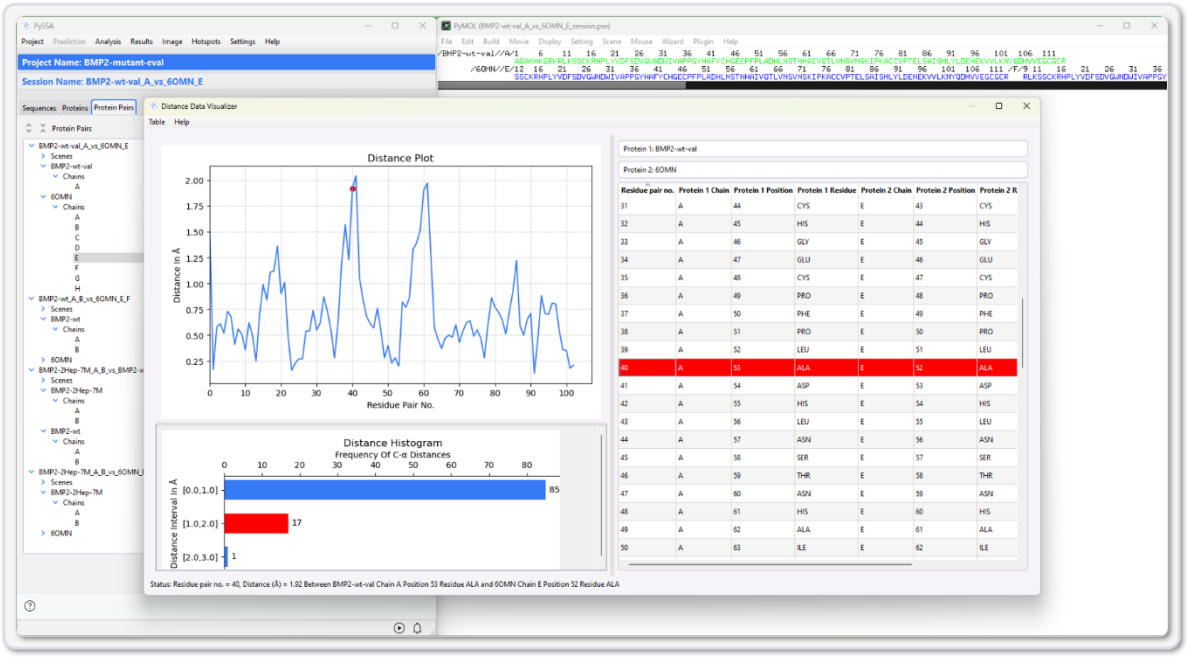


Figure 2. *Interactive distance diagram of the residue pairs in combination with a distance frequency histogram and a table overview for the structural alignment of the 6OMN reference with the predicted BMP2-wt-mono structure in PySSA.*

Table 1. *Structural alignment of protein pairs with deviations in angstrom. Residue columns give the deviations for the five most important residues for receptor binding (valine 26, tryptophan 31, proline 50, alanine 52, and serine 69). For protein pair BMP2-wt-mono vs. 6OMN, chain B is not included, since BMP2-wt-mono is a monomer only. Predicted (pred.) and experimental X-ray structures (exp.) are denoted accordingly. The AlphaFold 3 prediction in the last line is marked by an asterisk (*). RMSD values and deviations should be evaluated in relation to the 6OMN (exp.) experimental X-ray resolution of 2.68 Å.*

Residue		**V26**		**W31**		**P50**		**A52**		**S69**	
Chain		A	B	A	B	A	B	A	B	A	B
Protein pair	**RMSD [Å]**	**Deviation [Å]**									
BMP2-wt-mono (pred.) vs. **6OMN** (exp.)	0.82	0.70	-	1.36	-	0.63	-	1.92	-	1.33	-
BMP2-wt (pred.) vs. **6OMN** (exp.)	1.27	1.51	1.38	1.35	1.78	1.67	1.63	2.41	2.66	0.92	0.69
BMP2-2Hep-7M (pred.) vs. **6OMN** (exp.)	1.75	1.94	1.59	1.92	2.03	1.99	1.76	2.62	2.68	0.93	0.36
BMP2-2Hep-7M (pred.) vs. **BMP2-wt** (pred.)	3.54	0.86	1.02	1.46	1.37	1.01	1.13	0.80	1.05	0.91	0.98
BMP2-2Hep-7M (pred.)* vs. **6OMN** (exp.)	1.51	1.76	1.61	1.52	2.02	1.65	1.90	2.38	3.07	1.68	0.76

In order to predict and assess the spatial structure of the BMP2-2Hep-7M mutant (a mutant with seven point mutations to increase hydrophilicity and a doubled N-terminal heparin binding site, see above) [34] and compare it to the full wild-type BMP2 dimer, a new project is created, again using the experimental 6OMN X-ray structure of the wild-type BMP2 dimer as a reference for alignment with the predicted structures. Now, two protein structures are predicted from their amino acid sequences as a batch of two runs: the dimeric form of the BMP2 wild-type (denoted BMP2-wt) and the investigated BMP2-2Hep-7M mutant. The structure prediction of the BMP2-2Hep-7M dimer (2 x 122 residues) took about 4 h on the previously mentioned machine. Figure 3 shows the alignment of the predicted BMP2-2Hep-7M and the 6OMN reference structure in PySSA. The *Protein Regions* function provides a closer insight

into a protein structure or a protein pair by highlighting and zooming into a specified region of the structure. In Figure 4, this functionality is used to specifically inspect the alignment of tryptophan 31 between predicted BMP2-2Hep-7M and 6OMN. The results of all mutual structural alignment comparisons are provided in Table 1. The hydrophilisation mutation strategy and the duplication of the N-terminal heparin binding site in the BMP2-2Hep-7M mutant do not appear to significantly affect the spatial positions of the receptor-interacting amino acids. Their positional deviations between the predicted BMP2-2Hep-7M and the 6OMN reference structure are all within the experimental 6OMN X-ray resolution of 2.68 Å. Receptor binding and the elicitation of physiological effects should therefore not be significantly affected, an indication that needs to be confirmed by future experimental studies.

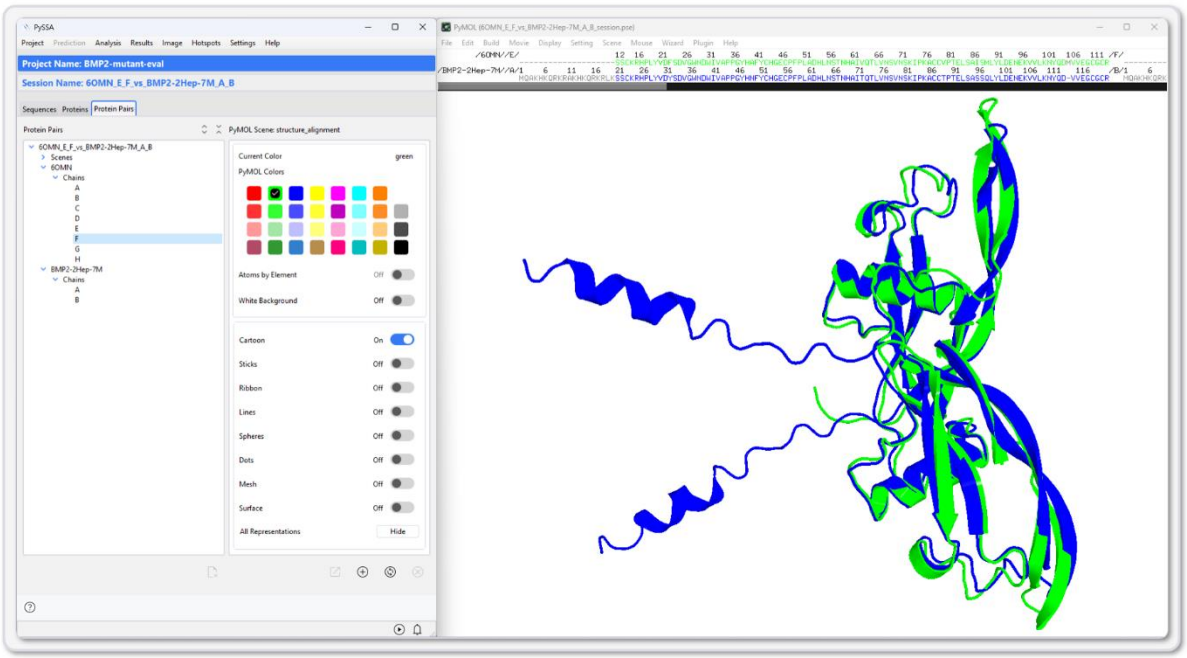


Figure 3. *Structural alignment scene of the 6OMN reference (green) with the predicted BMP2-2Hep-7M mutant (blue) in PySSA. The panel on the left can be used to customise the display, e.g. to set (background) colours or protein representations.*

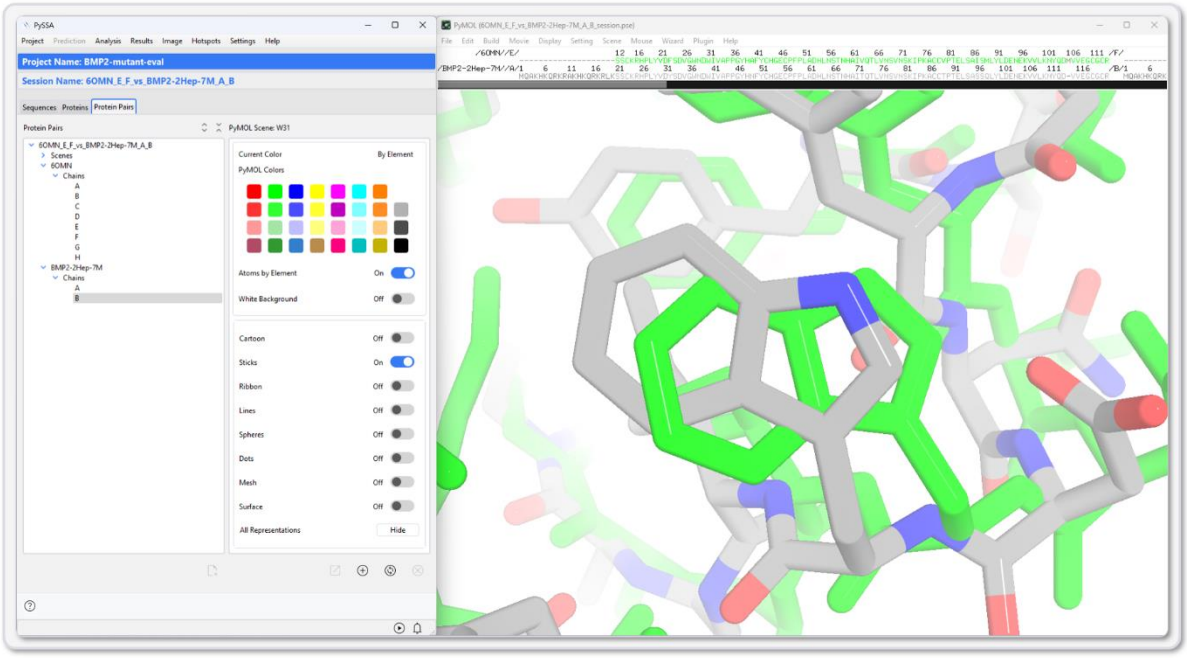


Figure 4. *PySSA protein region hotspot display with focus on tryptophan 31 for the structural alignment of the 6OMN reference (green) with the predicted BMP2-2Hep-7M mutant (coloured by elements).*

Finally, the BMP2-2Hep-7M mutant structure is predicted with the recently published closed-source AlphaFold 3 system [14]: the comparison of the BMP2-2Hep-7M mutant to the 6OMN reference shows slightly increased deviations for the receptor-interacting amino acids compared to the ColabFold (AlphaFold 2) prediction, with a slightly decreased overall RMSD value (1.75 to 1.51 Å, see Table 1). Thus, for the BMP2-2Hep-7M mutant, both prediction systems lead to comparable results.

Conclusion

PySSA aims to combine open-source PyMOL and ColabFold in a graphical software application to enable the prediction of spatial protein structures and the visual analysis of their structural alignments for the scientific end-user. It can be used for research and development as well as educational programs as it is easy to install and has a low entry barrier, enabling end-users to get familiar with the software within a few hours. Due to its modular architecture, future improvements in the prediction of spatial protein structures, such as alternative prediction engines (e.g. open systems comparable to the proprietary AlphaFold 3 system [16]) or additional analysis functions, can be easily implemented.

List of abbreviations

AMBER: Assisted Model Building with Energy Refinement

API: Advanced Programming Interface

BMP: Bone Morphogenetic Protein

CASP: Critical Assessment of protein Structure Prediction

CPU: Central Processing Unit

CSV: Comma-Separated Values

FDA: Food and Drug Administration

GPU: Graphics Processing Unit

GUI: Graphical User Interface

ID: Identifier

IP: Internet Protocol

MSA: Multiple Sequence Alignment

MVC: Model-View-Controller

OS: Operating System

PDB: Protein Data Bank (format)

PNG: Portable Network Graphic

PyPI: Python Package Index

PySSA: Python rich client for visual protein Sequence to Structure Analysis

RAM: Random Access Memory

RMSD: Root Mean Squared Deviation

WSL2: Windows Subsystem for Linux version 2

wt: wild-type

Availability and requirements

- Project name: PySSA
- Project home page: https://github.com/urban233/PySSA
- Documentation: https://github.com/urban233/PySSA
- Archived version: https://doi.org/10.5281/zenodo.12686853
- Current version: 1.0.1
- Operating system(s): Windows (x64)
- Programming language: Python
- Other requirements: none
- License: GPL-3.0 License

- Any restrictions to use by non-academics: None

Declarations

Availability of data and materials

Data and software are freely available under the GPL-3.0 License. The source code of PySSA is available on GitHub at https://github.com/urban233/PySSA. The source code of the Installer is also available on GitHub at https://github.com/urban233/ComponentInstaller.

Competing interests

AZ is co-founder of GNWI - Gesellschaft für naturwissenschaftliche Informatik mbH, Dortmund, Germany.

Funding

Not applicable.

Authors' contributions

HK and MU designed and developed the PySSA software and its documentation. JS revised and tested the system and its repository. ALS and AZ conceived the project. All authors wrote, read, and approved the final manuscript.

Acknowledgements

The authors would like to thank Uwe Urban for his help with the GUI programming and database design. The support of the Open Access Publication Fund of the Westphalian University of Applied Sciences is gratefully acknowledged. Special thanks go to the communities that created the open software systems used in this work.

References

[1] UniProtKB/Swiss-Prot (2024) UniProtKB/Swiss-Prot protein knowledgebase. https://ftp.uniprot.org/pub/databases/uniprot/previous_releases/release-2024_02/knowledgebase/UniProtKB_SwissProt-relstat.html. Accessed 6 Jun 2024

[2] The UniProt Consortium (2023) UniProt: the Universal Protein Knowledgebase in 2023. https://doi.org/10.1093/nar/gkac1052. Accessed 19 Jun 2024
[3] Burley et al (2023) RCSB Protein Data Bank (RCSB.org): delivery of experimentally-determined PDB structures alongside one million computed structure models of proteins from artificial intelligence/machine learning. https://doi.org/10.1093/nar/gkac1077.
[4] PDB Statistics: PDB Data Distribution by Natural Source Organism. https://www.rcsb.org/stats/distribution-source-organism-natural. Accessed 6 Jun 2024
[5] Allen et al Tables of bond lengths determined by X-ray and neutron diffraction. Part 1. Bond lengths in organic compounds. https://doi.org/10.1039/P298700000S1.
[6] Kryshtafovych A, Schwede T, Topf M, Fidelis K, Moult J (2021) Critical assessment of methods of protein structure prediction (CASP)-Round XIV. Proteins 89:1607–1617. https://doi.org/10.1002/prot.26237.
[7] Senior et al (2020) Improved protein structure prediction using potentials from deep learning. https://doi.org/10.1038/s41586-019-1923-7.
[8] Jumper J, Evans R, Pritzel A, Green T, Figurnov M, Ronneberger O, Tunyasuvunakool K, Bates R, Zídek A, Potapenko A et al (2021) Highly accurate protein structure prediction with AlphaFold. Nature 596:583–589. https://doi.org/10.1038/s41586-021-03819-2.
[9] Varadi M, Anyango S, Deshpande M, Nair S, Natassia C, Yordanova G, Yuan D, Stroe O, Wood G, Laydon A et al (2022) AlphaFold Protein Structure Database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. Nucleic Acids Res 50:439–444. https://doi.org/10.1093/nar/gkab1061.
[10] Callaway E (2022) The entire protein universe: AI predicts shape of nearly every known protein. Nature 608:15–16. https://doi.org/10.1038/d41586-022-02083-2.
[11] Tunyasuvunakool K, Adler J, Wu Z, Green T, Zielinski M, Zídek A, Bridgland A, Cowie A, Meyer C, Laydon A et al (2021) Highly accurate protein structure prediction for the human proteome. Nature 596:590–596. https://doi.org/10.1038/s41586-021-03828-1.
[12] Varadi M, Velankar S (2022) The impact of AlphaFold Protein Structure Database on the fields of life sciences. Proteomics. https://doi.org/10.1002/pmic.202200128.
[13] DeepMind (2024) AlphaFold repository on GitHub. https://github.com/google-deepmind/alphafold.
[14] Abramson J, Adler J, Dunger J, et al (2024) Accurate structure prediction of biomolecular interactions with AlphaFold 3. Nature 630:493–500. https://doi.org/10.1038/s41586-024-07487-w.
[15] AlphaFold3 — why did Nature publish it without its code? (2024) Nature 629:728 https://doi.org/10.1038/d41586-024-01463-0.

[16] Callaway E (2024) Who will make AlphaFold 3 open source? Scientists race to crack AI model. Nature 630:14-15. https://doi.org/10.1038/d41586-024-01555-x. Accessed 29 May 2024
[17] Baek et al (2021) Accurate prediction of protein structures and interactions using a three-track neural network. https://doi.org/10.1126/science.abj8754.
[18] Mirdita M, Schütze K, Moriwaki Y, Heo L, Ovchinnikov S, Steinegger M (2022) ColabFold: making protein folding accessible to all. Nat Methods 19:679–682. https://doi.org/10.1038/s41592-022-01488-1.
[19] G Kim, S Lee, E L Karin, H Kim, Y Moriwaki, S Ovchinnikov, M Steinegger, M Mirdita (2023) Easy and accurate protein structure prediction using ColabFold. PROTOCOL (Version 1) available at Protocol Exchange: https://doi.org/10.21203/rs.3.pex-2490/v1.
[20] Mirdita M, Steinegger M, Söding J (2019) MMseqs2 desktop and local web server app for fast, interactive sequence searches. Bioinformatics 35(16):2856–58. https://doi.org/10.1093/bioinformatics/bty1057.
[21] Python Software Foundation (2024) PyPI a repository of software for the Python programming language. https://pypi.org/. Accessed 18 Jun 2024
[22] ColabFold - v1.5.5 on PyPI. (2023) https://pypi.org/project/colabfold/. Accessed 18 Jun 2024
[23] Schrödinger LLC (2024) Open-Source PyMOL repository on GitHub. https://github.com/schrodinger/pymol-open-source. 19 Jun 2024
[24] Gohlke C (2024) PyMOL open-source wheels repository on GitHub. https://github.com/cgohlke/pymol-open-source-wheels. Accessed 20 Jun 2024
[25] IBCI (2024) PySSA repository on GitHub. https://github.com/urban233/PySSA. Accessed 20 Jun 2024
[26] Ruschke K, Hiepen C, Becker J, Knaus P (2012) BMPs are mediators in tissue crosstalk of the regenerating. Cell Tissue Res. 347(3):521-544. https://doi:10.1007/s00441-011-1283-6.
[27] Termaat MF, Den Boer FC, Bakker FC, Patka P, Haarman HJ (2005) Bone morphogenetic proteins. Development and clinical efficacy in the treatment of fractures and bone defects. J. Bone Joint Surg. Am. 87(6):1367-1378. https://doi:10.2106/JBJS.D.02585.
[28] Reddi AH (2005) BMPs: from bone morphogenetic proteins to body morphogenetic proteins. Cytokine Growth Factor Rev 16(3):249-250. https://doi:10.1016/j.cytogfr.2005.04.003.
[29] Hiepen C, Yadin D, Rikeit P, Dörpholz G, Knaus P (2016) Actions from head to toe: An update on Bone/Body Morphogenetic Proteins in health and disease. Cytokine Growth Factor Rev 27:1-11. https://doi.org/10.1016/j.cytogfr.2015.12.006.

[30] Gautschi OP, Frey SP, Zellweger R (2007) Bone morphogenetic proteins in clinical applications. ANZ J. Surg. 77(8):626-631. https://doi:10.1111/j.1445-2197.2007.04175.x.
[31] Nauth A, Ristiniemi J, McKee MD, Schemitsch EH (2009) Bone morphogenetic proteins in open fractures: past, present, and future. Injury 40(3):27-31. https://doi:10.1016/S0020-1383(09)70008-7.
[32] Bessa PC, Casal M, Reis RL (2008) Bone morphogenetic proteins in tissue engineering: the road from laboratory to clinic, part II (BMP delivery). J. Tissue Eng. Regen. Med. 2(2-3):81-96. https://doi:10.1002/term.74.
[33] James AW, LaChaud G, Shen J, Asatrian G, Nguyen V, Zhang X, Ting K, Soo C (2016) A Review of the Clinical Side Effects of Bone Morphogenetic Protein-2. Tissue Eng. Part B Rev 22(4):284-297. https://doi:10.1089/ten.TEB.2015.0357.
[34] Heinks T, Hettwer A, Hiepen C, Weise C, Gorka M, Knaus P, Mueller TD, Loidl-Stahlhofen A (2021) Optimized expression and purification of a soluble BMP2 variant based on in-silico design. Protein Expr. Purif. 186:105918. https://doi.org/10.1016/j.pep.2021.105918.
[35] Python Software Foundation (2024) Python 3.11. https://www.python.org/downloads/release/python-3119/. Accessed 18 Jun 2024
[36] Riverbank Computing (2023) PyQt. https://www.riverbankcomputing.com/software/pyqt/. Accessed 19 Jun 2024
[37] SQLite. https://www.sqlite.org/index.html. Accessed 25 May 2024
[38] AlmaLinux OS Foundation (2024) AlmaLinux. https://almalinux.org/. Accessed 20 Jun 2024
[39] ZeroMQ. https://zeromq.org/. Accessed 20 Jun 2024
[40] sokrypton (2024) ColabFold repository on GitHub. https://github.com/sokrypton/ColabFold. Accessed 9 May 2024
[41] Söding Lab (2024) MMSeqs2 repository on GitHub. https://github.com/soedinglab/MMseqs2.
[42] Island Lake Consulting LLC (2024) WinBatch. https://www.winbatch.com/. Accessed 19 Jun 2024
[43] IBCI (2024) Component Installer repository on GitHub. https://github.com/urban233/ComponentInstaller. Accessed 20 Jun 2024
[44] Protein Data Bank, 6OMN, Glycosylated BMP2 homodimer. https://doi.org/10.2210/pdb6OMN/pdb. Accessed 27 May 2024
[45] Seeherman et al (2019) A BMP/activin A chimera is superior to native BMPs and induces bone repair in nonhuman primates when delivered in a composite matrix. http://doi.org/10.1126/scitranslmed.aar4953.

[46] Hornak et al (2006) Comparison of multiple Amber force fields and development of improved protein backbone parameters. https://doi.org/10.1002/prot.21123
[47] Eastman et al. (2017) OpenMM 7: Rapid development of high performance algorithms for molecular dynamics. https://doi.org/10.1371/journal.pcbi.1005659.
[48] Kirsch T, Nickel J, Sebald W (2000) BMP-2 antagonists emerge from alterations in the low-affinity binding epitope for receptor BMPR-II. EMBO J. 19(13):3314-24. https://doi:10.1093/emboj/19.13.3314.