Bioinformatics, 38(9), 2022, 2633–2635
https://doi.org/10.1093/bioinformatics/btac121
Advance Access Publication Date: 23 February 2022
Applications Note

Structural bioinformatics

localpdb—a Python package to manage protein structures and their annotations

Jan Ludwiczak *, Aleksander Winski and Stanislaw Dunin-Horkawicz *

Laboratory of Structural Bioinformatics, Centre of New Technologies, University of Warsaw, 02-097 Warsaw, Poland

*To whom correspondence should be addressed.

Associate Editor: Lenore Cowen

Received on July 29, 2021; revised on January 7, 2022; editorial decision on February 19, 2022; accepted on February 21, 2022

Abstract

Motivation: The wealth of protein structures collected in the Protein Data Bank enabled large-scale studies of their function and evolution. Such studies, however, require the generation of customized datasets combining the structural data with miscellaneous accessory resources providing functional, taxonomic and other annotations. Unfortunately, the functionality of currently available tools for the creation of such datasets is limited and their usage frequently requires laborious surveying of various data sources and resolving inconsistencies between their versions.

Results: To address this problem, we developed *localpdb*, a versatile Python library for the management of protein structures and their annotations. The library features a flexible plugin system enabling seamless unification of the structural data with diverse auxiliary resources, full version control and powerful functionality of creating highly customized datasets. The *localpdb* can be used in a wide range of bioinformatic tasks, in particular those involving large-scale protein structural analyses and machine learning.

Availability and implementation: *localpdb* is freely available at https://github.com/labstructbioinf/localpdb. Documentation along with the usage examples can be accessed at https://labstructbioinf.github.io/localpdb/.

Contact: j.ludwiczak@cent.uw.edu.pl or s.dunin-horkawicz@cent.uw.edu.pl

1 Introduction

The size of the Protein Data Bank (PDB) has been growing steadily over the past decades (Burley *et al.*, 2019), encouraging the development of computational tools enabling classification (Andreeva *et al.*, 2020; Cheng *et al.*, 2014; Dawson *et al.*, 2017) and annotation (Dana *et al.*, 2019) of protein structures it encompasses. The wealth of the collected data has opened up many research opportunities ranging from studies focused on proteins of a particular function or evolutionary position to comprehensive surveys aiming at capturing the general aspects of the 'protein universe' (Alva *et al.*, 2010; Nepomnyachiy *et al.*, 2014). However, these innovations came at the cost of the dispersion of data sources, whose integration began to require expert knowledge and the tedious process of mapping structures to the corresponding sequence and functional information. Another problem frequently faced during research involving multiple PDB structures is related to the management of their versions—it is not uncommon for PDB structures to become updated, entirely changed or even removed after deposition. Similarly, the associated tools integrating the classifications and annotations, such as the PDBe-KB (Varadi *et al.*, 2020), RCSB (Rose *et al.*, 2021), SIFTS (Dana *et al.*, 2019), Proteo3Dnet (Postic *et al.*, 2021) or ECOD (Cheng *et al.*, 2014), are updated regularly to capture new structures or fix errors. Consequently, the datasets obtained prior to a given study may differ substantially from those generated later. This, in turn, makes the recreation of data at a given time point very hard, negatively impacting the research reproducibility.

The difficulties outlined above force researchers to manually collect and manage their datasets, bringing a risk of omitting important information, for example, during the collection of protein structures originating from a given taxonomic, evolutionary or functional group. Also, even cosmetic changes to a few structures, their identifiers or associated data may cause hard-to-track internal inconsistencies that need to be amended by hand. These research-hampering issues were partially alleviated by the development of computational tools such as the PDB module (Hamelryck and Manderick, 2003) of the *biopython* package (Cock *et al.*, 2009), CCPDB webserver (Agrawal *et al.*, 2019; Singh *et al.*, 2012) or PDB application programming interfaces (API) (Gilpin, 2015; Rose *et al.*, 2021). Unfortunately, these solutions have limitations, such as the lack of version management, the impossibility of building highly customized datasets based on complex queries, and last but not least, the unavailability of high-level integration with the programming tools typically used to carry out bioinformatics analyses.

The modern data analysis pipelines rely on the general developments in the Python and R programming languages. In particular, the so-called data frame structures, originating from the R language, become increasingly popular among Python developers, owing to

the development of the *pandas* package (McKinney, 2010). The main advantage of using data frames in data analyses stems from their plasticity, that is, a possibility to query, sort, search, merge or partition them with a handful of simple commands. The usefulness of *pandas* data frames inspired the development of its extensions devoted to the analyses of biological data. For example, the *biopandas* (Raschka, 2017) library enables representing the individual PDB structures as data frames, thus greatly facilitating their investigation at the atomic level. Another tool having *pandas* under the hood is *rstoolbox* (Bonet *et al.*, 2019), a Python library for the analysis of large-scale structural data, mostly in the context of protein design tasks.

The limitations of the currently available tools motivated us to develop *localpdb*, a lightweight Python library that utilizes the versatility of *pandas DataFrames* to offer a simple programming framework for handling a local copy of the PDB and auxiliary resources, such as ECOD (Schaeffer *et al.*, 2017), SIFTS (Dana *et al.*, 2019) or DSSP (Touw *et al.*, 2015) and thus enabling the creation of complex and reproducible bioinformatics workflows. The requirements to use *localpdb* are minimal and the package will run on any modern PC with enough disk space (around 100 GB if the user will decide to store PDB structures in both the PDB and mmCIF formats). In the following section, we provide a concise overview of the package features and functionalities. The full documentation can be accessed at https://labstructbioinf.github.io/localpdb/.

2 Overview of the *localpdb* package

The overview of the *localpdb* package is shown in Figure 1. In its basic functionality, it allows creating a local mirror image of the PDB (in either PDB or mmCIF formats) accessible via the *PDB* object and its *entries* and *chains* attributes (both being *pandas DataFrames*) that provide direct access to the whole structures and their components, respectively.

With the weekly releases of the PDB, the local copy can be updated to account for added, modified or outdated entries. Importantly, the update procedure is optional and does not have to follow a weekly routine. Another important feature of the *localpdb* is the ability to retain access to the previous versions of the data even after performing multiple updates. This enables the work on multiple projects concurrently without the need to store large separate datasets for each of them. Moreover, data corresponding to any of the *localpdb* versions can be seamlessly shared and independently recreated using the relatively small, exportable configuration file.

The basic functionality of the *localpdb* package is extendable with a flexible plugin system. The plugins enable augmenting the structural data with various annotations and making them available to the user in a uniform, searchable *DataFrame* format. We implemented several plugins featuring access to various resources such as domain databases (SCOP, ECOD, CATH and Pfam) or taxonomic and EC number mappings. While these plugins provide annotations related to the whole structure, chain or their segments, the others enable also per residue annotations. For example, the current release of the *localpdb* package includes plugins for DSSP (Touw *et al.*, 2015), a program for the annotation of secondary structure elements and Socket (Walshaw and Woolfson, 2001), a tool for the annotation of coiled-coil domains. Alike the core of the *localpdb* package, also the plugins are under the control of the versioning system, which is especially important in the context of periodically updated resources such as ECOD or SIFTS. Finally, in parallel to the plugin system, the *localpdb* package features access to the recently released RCSB search API (Rose *et al.*, 2021) allowing for queries based on the text descriptors, sequence similarity metrics and sequence or structural motifs.

In an effort to simplify the process of creation of the datasets integrating information from multiple sources, we implemented a filtering system that adjusts all the active *DataFrames* once the selection is performed on either of them. For example, performing a selection on the *entries DataFrame* automatically adjusts all the associated *DataFrames* such as *chains* and those related to the plugins.

To fully demonstrate the applicability of the package, we present two advanced use cases that accompany the manuscript and are available at the documentation webpage (https://labstructbioinf.github.io/localpdb/). The first example reproduces the ensemble analysis of the publicly available HIV protease structures to infer the conformational dynamics of this enzyme (Katebi *et al.*, 2015). We show that coupling *localpdb* with other common structural bioinformatics packages enables the recreation of this complex, multistep analysis purely in Python with a handful of lines of code. The second example focuses on the machine learning applications and describes steps performed to derive a dataset used to train our coiled-coil domain prediction algorithm—DeepCoil (Ludwiczak *et al.*, 2019). The presented approach can be easily adapted to facilitate dataset creation for similar machine learning tasks in which sequence similarity control and minority class oversampling are essential. Finally, the *localpdb* package has been also additionally tested during various projects conducted in our group; for example, it was used to construct and maintain specialized datasets used to train machine learning models allowing the prediction of protein-ligand interactions (Kamiński *et al.*, 2021) and to develop a pipeline for the annotation of coiled-coil motifs in protein structures (Szczepaniak *et al.*, 2021).

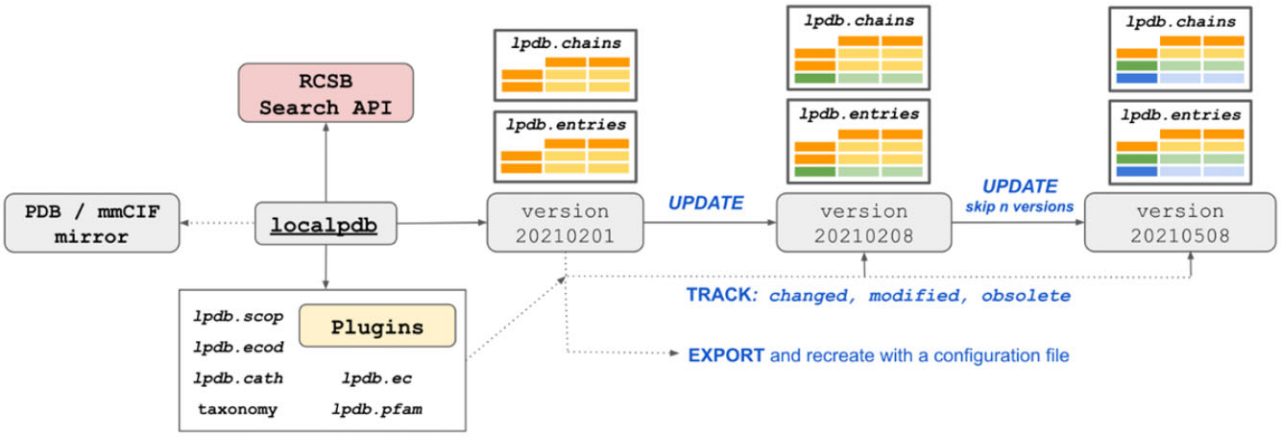

Fig. 1. General overview of the features and functionalities of the *localpdb* package. At its core, *localpdb* syncs the raw PDB data and entries (in PDB and mmCIF formats) and makes them available to the user through the *DataFrame* objects. With the weekly releases of new data, the local files can be updated, however, the possibility to access the previous versions is retained through the tracking mechanism. The functionalities of the *localpdb* can be further extended with the configurable plugin system that allows to fetch and track the updates from the additional data sources. *localpdb also* provides access to the RCSB search API that can be used for complex queries based on multiple criteria. Finally, each version of the *localpdb* can be independently recreated on a different machine or by other users by exporting a small configuration file

In sum, the *localpdb* package unifies and extends a variety of features offered by other tools by providing robust versioning and easily extendable plugin systems. We therefore envision that the *localpdb* package can be applicable to most structural bioinformatics tasks, in particular those related to building complex and reproducible workflows around the PDB data and deriving datasets for machine learning purposes.

Acknowledgements

The authors thank Kamil Kaminski, Krzysztof Szczepaniak, Adriana Bukala, Rafal Madaj and Maciej Jasinski for testing and validation of the method and useful comments on how to improve it.

Data availability

The underlying data are available within the article and the accompanying supplementary materials available at: https://github.com/labstructbioinf/localpdb, https://labstructbioinf.github.io/localpdb/.

Funding

This work was supported by the National Science Centre, Poland [2017/27/N/NZ1/00716 to J.L.]. S.D.-H. was supported by the First TEAM program of the Foundation for Polish Science co-financed by the European Union under the European Regional Development Fund [POIR.04.04.00-00-5CF1/18-00 to S.D.-H.].

Conflict of Interest: none declared.

References

Agrawal,P. *et al.* (2019) ccPDB 2.0: an updated version of datasets created and compiled from Protein Data Bank. *Database*, **2019**, bay142.

Alva,V. *et al.* (2010) A galaxy of folds. *Protein Sci.*, **19**, 124–130.

Andreeva,A. *et al.* (2020) The SCOP database in 2020: expanded classification of representative family and superfamily domains of known protein structures. *Nucleic Acids Res.*, **48**, D376–D382.

Bonet,J. *et al.* (2019) rstoolbox – a Python library for large-scale analysis of computational protein design data and structural bioinformatics. *BMC Bioinformatics*, **20**, 240.

Burley,S.K. *et al.* (2019) Protein Data Bank: the single global archive for 3D macromolecular structure data. *Nucleic Acids Res.*, **47**, D520–D528.

Cheng,H. *et al.* (2014) ECOD: an evolutionary classification of protein domains. *PLoS Comput. Biol.*, **10**, e1003926.

Cock,P.J.A. *et al.* (2009) Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics*, **25**, 1422–1423.

Dana,J.M. *et al.* (2019) SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins. *Nucleic Acids Res.*, **47**, D482–D489.

Dawson,N.L. *et al.* (2017) CATH: an expanded resource to predict protein function through structure and sequence. *Nucleic Acids Res.*, **45**, D289–D295.

Gilpin,W. (2015) PyPDB: a Python API for the Protein Data Bank. *Bioinformatics*, btv543.

Hamelryck,T. and Manderick,B. (2003) PDB file parser and structure class implemented in Python. *Bioinformatics*, **19**, 2308–2310.

Kamiński,K. *et al.* (2021) Rossmann-toolbox: a deep learning-based protocol for the prediction and design of cofactor specificity in Rossmann fold proteins. *Brief. Bioinf.*, **23**, bbab371.

Katebi,A.R. *et al.* (2015) The use of experimental structures to model protein dynamics. *Methods Mol. Biol.*, **1215**, 123–236.

Ludwiczak,J. *et al.* (2019) DeepCoil – a fast and accurate prediction of coiled-coil domains in protein sequences. *Bioinformatics*, **35**, 2790–2795.

McKinney,W. (2010) Data structures for statistical computing in Python. In: van der Walt,S. and Millman,J. (eds.) *Proceedings of the 9th Python in Science Conference*. Austin, TX, Vol. 445, pp. 56–61.

Nepomnyachiy,S. *et al.* (2014) Global view of the protein universe. *Proc. Natl. Acad. Sci. USA*, **111**, 11691–11696.

Postic,G. *et al.* (2021) Proteo3Dnet: a web server for the integration of structural information with interactomics data. *Nucleic Acids Res.*, **49**, W567–W572.

Raschka,S. (2017) BioPandas: working with molecular structures in pandas DataFrames. *J. Open Source Softw.*, **2**, 279.

Rose,Y. *et al.* (2021) RCSB Protein Data Bank: architectural advances towards integrated searching and efficient access to macromolecular structure data from the PDB archive. *J. Mol. Biol.*, **433**, 166704.

Schaeffer,R.D. *et al.* (2017) ECOD: new developments in the evolutionary classification of domains. *Nucleic Acids Res.*, **45**, D296–D302.

Singh,H. *et al.*; Open Source Drug Discovery Consortium. (2012) ccPDB: compilation and creation of data sets from Protein Data Bank. *Nucleic Acids Res.*, **40**, D486–D489.

Szczepaniak,K. *et al.* (2021) A library of coiled-coil domains: from regular bundles to peculiar twists. *Bioinformatics*, **36**, 5368–5376.

Touw,W.G. *et al.* (2015) A series of PDB-related databanks for everyday needs. *Nucleic Acids Res.*, **43**, D364–D368.

Varadi,M. *et al.* (2020) PDBE-KB: a community-driven resource for structural and functional annotations. *Nucleic Acids Res.*, **48**, D344–D353.

Walshaw,J. and Woolfson,D.N. (2001) SOCKET: a program for identifying and analysing coiled-coil motifs within protein structures. *J. Mol. Biol.*, **37**, 4575–4577.