The Journal of Open Source Software

pridepy: A Python package to download and search data from PRIDE database

Selvakumar Kamatchinathan[1], **Suresh Hewapathirana**[1], **Chakradhar Bandla**[1], **and Yasset Perez-Riverol**[1]

1 European Molecular Biology Laboratory, European Bioinformatics Institute (EMBL-EBI), Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK

DOI: N/A

Software

- Review
- Repository
- Archive

Editor: Open Journals

Reviewers:

- @openjournals

Submitted: 01 January 1970
Published: 01 January 1970

Summary

The Proteomics Identification Database (PRIDE) (Perez-Riverol et al., 2022) is the world's largest repository for proteomics data and a founding member of ProteomeXchange (Deutsch et al., 2023). Here, we introduce `pridepy`, a Python client designed to access PRIDE Archive data, including project metadata and file downloads. `pridepy` offers a flexible programmatic interface for searching, retrieving, and downloading data via the PRIDE REST API. This tool simplifies the integration of PRIDE datasets into bioinformatics pipelines, making it easier for researchers to handle large datasets programmatically.

Statement of Need

The PRIDE Archive storages an extensive collection of proteomics data (Perez-Riverol et al., 2022), but manually accessing this data can be inefficient and time-consuming. With the growing need for cloud-based (Dai et al., 2024) and HPC bioinformatics tools (Mehta et al., 2023), command-line utilities that seamlessly interact with the PRIDE API are increasingly important. `pridepy` addresses this by enabling researchers to programmatically access PRIDE using Python, a widely adopted language. It allows efficient dataset integration into automated workflows, with support for large-scale data transfers via Aspera, Globus, FTP, and HTTPS, making it ideal for scalable, reproducible pipelines.

Methods

`pridepy` is built in Python and interacts with the PRIDE Archive REST API. The library and package not only provide data models for each data structure of the API but also a set of commandline to facilitate their use by users. The main features of `pridepy` is dataset search and file download.

The client is available on PyPI and can be installed using `pip`. The source code is hosted on GitHub and is open-source under the Apache 2.0 license. In addition, a conda recipe is available for easy installation in conda environments. The package is continuously tested using GitHub Actions and has been successfully deployed on the EMBL-EBI HPC cluster.

Downloading files from PRIDE Archive

By 2024, PRIDE Archive stores the data in two different storage systems (**Figure 1**), one for public data and another for private data. The public data is stored in a S3-like storage system, called FIRE (Thakur et al., 2024), which also includes other major EMBL-EBI archives such as ENA (European Nucleotide Archive) and EGA (European Genome-phenome Archive). FIRE

has limited capabilities for data updates and deletions, making it ideal for long-term data storage. FIRE data is accessible via multiple protocols including FTP, Aspera and Globus. In contrast, private datasets are stored in a different file-system based on NFS, which is more flexible for data updates and deletion; but it is only accessible via PRIDE streaming protocol.

The `pridepy` client provides a simple command line interface to download files from PRIDE Archive using the following protocols. Each protocol offers different advantages: - **FTP**: Widely supported and easy to use - **Aspera**: High-speed file transfers, especially for large files or over long distances - **Globus**: Reliable transfers for very large datasets. It is important to notice that in the current implementation, pridepy uses the https endpoint of the Globus service, which means that the data is in fact transfer using the https protocol.

These are currently the only supported protocols for file downloads.

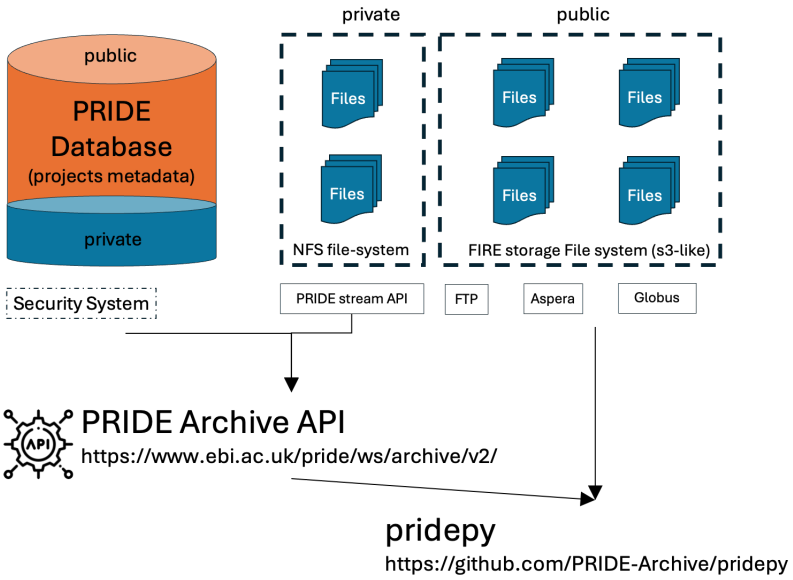

Figure 1: Architecture of transfer protocols supported by PRIDE Archive

Users can download files from PRIDE Archive using the following command options: - `download-all-public-raw-files`: command downloads all raw files from a dataset, it is useful for large-scale data retrieval and public datasets. - `download-file-by-name`: command downloads a single file by name. Users can specify the output directory, protocol (FTP, Aspera or Globus), and other options to customize the download process. For private datasets, the user and password are required to access the data.

One example of downloading all raw files using Aspera from a dataset is shown below:

```
$ pridepy download-all-raw-files \
  -a PXD012353 \
  -o /Users/yourname/Downloads/foldername/ \
  -p aspera
```

This makes the client suitable for handling large-scale proteomics data in automated workflows, particularly in environments requiring bulk downloads of proteomics datasets.

Benchmark of the download speed

We conducted a benchmark to compare the download speeds of the three protocols supported by the PRIDE Archive. The test was carried out on files of varying sizes (~14MB, ~230MB,

~3GB, and ~7GB). We reached out to several PRIDE users, providing them with a benchmark script (available at https://github.com/PRIDE-Archive/pridepy/tree/master/benchmark), and the benchmark was run across multiple locations, including the USA, UK, Europe, and Asia (Hong Kong). The results are presented in Figure 2-3.

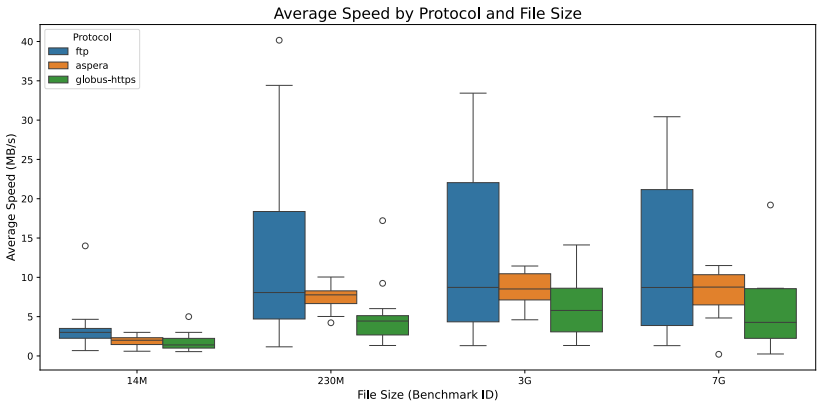

Figure 2: Benchmark of download speed for different protocols

For small files (~14MB), the three have similar performance. For medium (~230MB) and large files (~3G and ~7G), Globus and Aspera outperformed FTP. More importantly, FTP performance can decrease significantly with increasing file size, while Aspera and Globus maintain a more consistent download speed. The newly introduced Globus protocol showed the best performance for large files, making it the preferred choice for large-scale data transfers (Figure 2). In Figure 3 we present the download speed for different file sizes, protocols and locations.

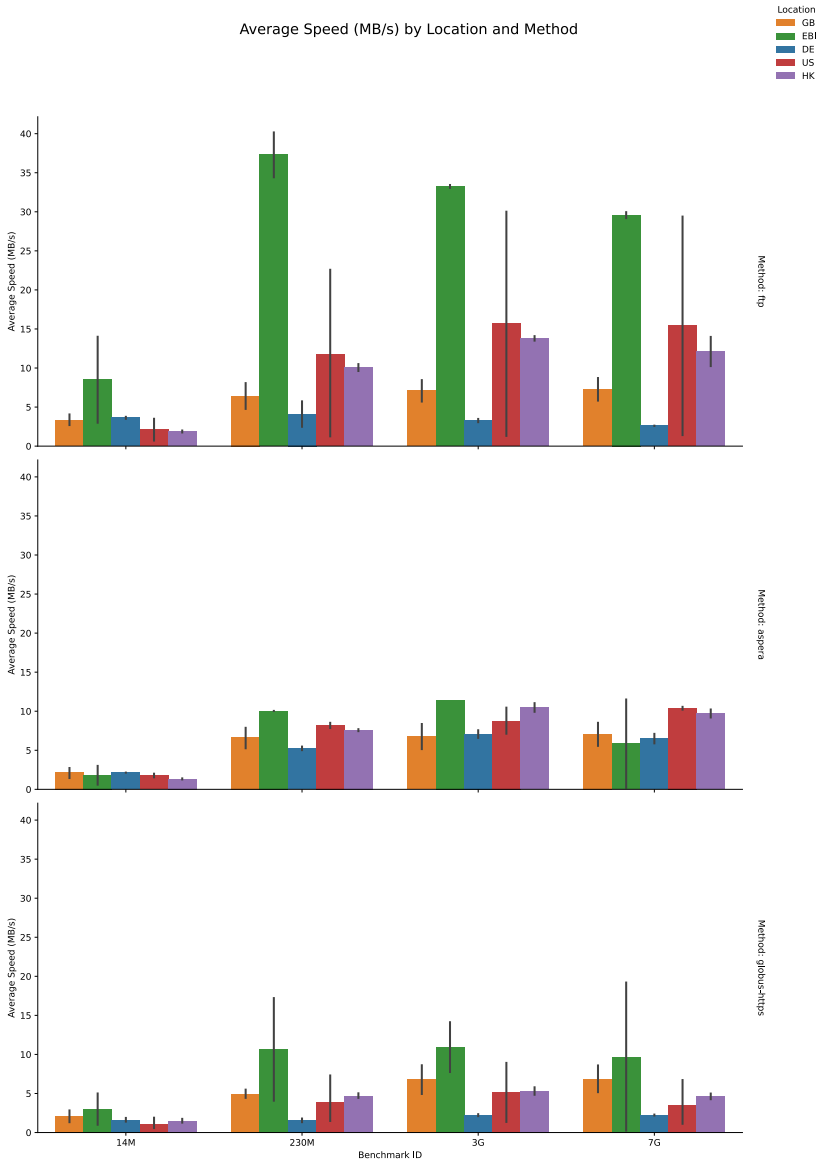

Figure 3: Benchmark of download speed for different protocols, file sizes and locations

As file sizes increase, Aspera and Globus consistently deliver the fastest download speeds, especially for larger files (~3GB and ~7GB), making it more efficient for large data transfers. Globus performs well but shows higher variability based on location, while FTP generally exhibits slower speeds, particularly with larger files. The benchmark highlights that both the choice of protocol and geographical location significantly impact download speeds, with Aspera being the most robust option across various conditions.

Discussion and Future Directions

`pridepy` (https://github.com/PRIDE-Archive/pridepy) successfully simplifies access to the PRIDE Archive data, but future development could focus on improving the handling of large downloads by implementing parallel downloads. Additionally, we will expand the user documentation and examples could help broaden its use within the scientific community; and at the same time produce a group of benchmarks to evaluate the performance of the client in

different scenarios. We plat to continue extending the library to support more features of the PRIDE Archive API, such as dataset metadata streaming, and submission of new datasets to the PRIDE Archive.

Acknowledgments

We would like to thank the PRIDE Archive team and contributors to this project for their invaluable input and feedback. The work is supported by core funding from the European Molecular Biology Laboratory (EMBL) and the Wellcome Trust [grant numbers 208391/Z/17/Z and 223745/Z/21/Z], and the BBSRC grant 'DIA-Exchange' [BB/X001911/1]. Thanks to Enrique Audain, Jonas Scheid, J. Sebastian Paez, and Dai Chengxin for their contributions to the benchmarking study. Thanks to Santiago Insua from an EBI instrastructure team for his support deploying multiple AWS machines in different locations for the benchmark.

References

Dai, C., Pfeuffer, J., Wang, H., Zheng, P., Käll, L., Sachsenberg, T., Demichev, V., Bai, M., Kohlbacher, O., & Perez-Riverol, Y. (2024). Quantms: A cloud-based pipeline for quantitative proteomics enables the reanalysis of public proteomics data. *Nat. Methods*, *21*(9), 1603–1607.

Deutsch, E. W., Bandeira, N., Perez-Riverol, Y., Sharma, V., Carver, J. J., Mendoza, L., Kundu, D. J., Wang, S., Bandla, C., Kamatchinathan, S., Hewapathirana, S., Pullman, B. S., Wertz, J., Sun, Z., Kawano, S., Okuda, S., Watanabe, Y., MacLean, B., MacCoss, M. J., … Vizcaíno, J. A. (2023). The ProteomeXchange consortium at 10 years: 2023 update. *Nucleic Acids Res.*, *51*(D1), D1539–D1548.

Mehta, S., Bernt, M., Chambers, M., Fahrner, M., Föll, M. C., Gruening, B., Horro, C., Johnson, J. E., Loux, V., Rajczewski, A. T., Schilling, O., Vandenbrouck, Y., Gustafsson, O. J. R., Thang, W. C. M., Hyde, C., Price, G., Jagtap, P. D., & Griffin, T. J. (2023). A galaxy of informatics resources for MS-based proteomics. *Expert Rev. Proteomics*, *20*(11), 251–266.

Perez-Riverol, Y., Bai, J., Bandla, C., García-Seisdedos, D., Hewapathirana, S., Kamatchinathan, S., Kundu, D. J., Prakash, A., Frericks-Zipper, A., Eisenacher, M., Walzer, M., Wang, S., Brazma, A., & Vizcaíno, J. A. (2022). The PRIDE database resources in 2022: A hub for mass spectrometry-based proteomics evidences. *Nucleic Acids Res.*, *50*(D1), D543–D552.

Thakur, M., Buniello, A., Brooksbank, C., Gurwitz, K. T., Hall, M., Hartley, M., Hulcoop, D. G., Leach, A. R., Marques, D., Martin, M., Mithani, A., McDonagh, E. M., Mutasa-Gottgens, E., Ochoa, D., Perez-Riverol, Y., Stephenson, J., Varadi, M., Velankar, S., Vizcaino, J. A., … McEntyre, J. (2024). EMBL's european bioinformatics institute (EMBL-EBI) in 2023. *Nucleic Acids Res.*, *52*(D1), D10–D17.