The Journal of Open Source Software

PDR: The Planetary Data Reader

Sierra Brown [1], **Michael St. Clair** [1], **Chase Million** [1], **Sabrina A. Curtis** [1], **K.-Michael Aye** [2], and **Zack Weinberg** [1]

1 Million Concepts LLC **2** Institut für Geologische Wissenschaften, Freie Universität Berlin

DOI: 10.21105/joss.07256

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey

Reviewers:

- @AndrewAnnex
- @gaelccc

Submitted: 23 July 2024
Published: 30 September 2024

Summary

The Planetary Data Reader, `pdr`, is an open-source Python-language package that reads data stored in planetary science formats and converts it into standard Python objects. It typically loads images as `ndarrays`, tables as Pandas `DataFrames`, and metadata and ancillary data as strings or dicts. `pdr`'s interface is designed to be maximally accessible to the introductory Python user. To read a data file, a user must simply `import pdr`, then run `pdr.read(fn)`, where `fn` is the data file or a detached 'label' (metadata) file associated with it. `pdr` will immediately load the product's metadata, then lazily load data objects when referenced.

`pdr` reads data products held by the Planetary Data System (PDS) that follow either PDS3 (Jet Propulsion Laboratory, 2009) or PDS4 (Jet Propulsion Laboratory, 2024) standards – meaning that they have metadata labels that generally follow one of these two formats. (It also supports some common scientific interchange data formats that are not PDS, including FITS.) We knew from the outset that many products in the PDS were not fully standards-compliant, particularly those archived under PDS3; its holdings are extremely diverse and span over half a century. For this reason, we took a **data-driven development** approach rather than attempting to implement these standards to the letter. What this means in practice is that we built `pdr` around a core of extremely flexible heuristics that permit it to permissively accept and correctly handle many products that deviate from the standards.

We have developed these heuristics through an iterative design process centered on examination of actually-existing data. We created manifests of the holdings of each of the PDS nodes, then used them to help identify 'types' of data and retrieve representative samples of each 'type' (the PDS holds hundreds of millions of files with a total data volume in the petabytes, so examining every file is impractical).

When we examine a new type, we verify the correctness of `pdr`'s behavior across our sample of that type and make changes to `pdr` as necessary to support that type's characteristics. We then add one or two individual products of that type to a data corpus we use for regression testing. This has permitted us to design software that conforms to planetary data rather than planetary data standards. Our methods for adding dataset support are further described in Kaufman et al. (2022), and our testing toolchain can be found in Curtis et al. (2024).

`pdr` is an affliate package of `planetarypy` (PlanetaryPy Technical Committee, 2024). It is available on the Python Package Index and `conda-forge`.

Statement of need

Just accessing data is a major pain point for planetary scientists. Data archived under the PDS3 standards can be especially challenging due to inconsistent, specialized, or flatly incorrect formatting. While the newer PDS4 standards are significantly stricter, many of the holdings of the PDS have not yet been migrated to this standard. `pdr` can remove months of preparatory work, making it faster for scientists to get to core research tasks and making it much more practical for them to incorporate data sets they haven't worked with previously into their research.

The simplicity and consistency of `pdr`'s API, along with its speed and stability, make it ideal for use in automated data processing pipelines. `pdr` is currently used in a wide variety of planetary projects. These include the Perseverance rover's Mastcam-Z tactical pipeline (St. Clair, Million, et al., 2023) and PDS3 to PDS4 migration pipelines for data from Clementine (St. Clair et al., 2021), Chandrayaan-1 M3 (Pieters et al., 2021), and the Viking Orbiter cameras (St. Clair, Brown, et al., 2023). Its fast metadata parsing features make it especially appealing for converting metadata standards across tens of millions of products. For example, `pdr` is able to parse metadata and load the image arrays for 100 nominal-sized Mastcam-Z IOFs using their attached PDS3 labels in 1.5 seconds and in 0.75 seconds to parse the metadata alone. More complex labels such as the New Horizons ALICE calibrated Jupyter PDS3 labels took `pdr` 1.7 seconds to parse metadata for 100 files.

Other packages

There is a very wide variety of software intended to read data in planetary science formats. `pdr`'s most important distinctions are its emphasis on breadth, simplicity, and high compatibility with other tools. `pdr` incorporates some of this software, including `pds4_tools` (Small Bodies Node & Nagdimunov, 2021) and `astropy.io.fits` (Astropy Developers, 2024). `pdr` uses these packages to read, respectively, PDS4 and FITS files, converting their outputs into standard Python objects to provide users with a common interface regardless of file format.

It is important to note that many pieces of software with narrower *format* scope than `pdr` have wider *application* scope. For instance, GDAL (Rouault et al., 2024) and `rasterio` (Mapbox, 2024) (which uses GDAL) read a narrower range of data and do not provide as consistent or straightforward an interface, but will deal with map projection transformations; `plio` (USGS Astrogeology, 2024) only reads data in a few formats, but is capable of applying instrument-specific metadata-parsing rules. Many of these tools also offer write capabilities, which `pdr` does not. Users who require write capabilities or subdomain-specific behaviors might find narrowly-focused tools more appropriate; they might also find `pdr` useful as a preprocessor for such tools.

Acknowledgements

The development of pdr is supported by NASA grant No. 80NSSC21K0885. We would like to thank the Planetary Data System (PDS) for their continued cooperation with this project.

References

Astropy Developers. (2024). The Astropy Project. In *Github repository*. Github. https://github.com/astropy/astropy

Curtis, S. A., Brown, S. V., & St. Clair, M. A. (2024). Pdr-tests: The testing suite/toolchain for 'pdr'. In *Github repository*. Github. https://github.com/MillionConcepts/pdr-tests

Jet Propulsion Laboratory. (2009). *Planetary Data System Standards Reference Version 3.8*. nasa. https://pds.nasa.gov/datastandards/pds3/standards/sr/ StdRef_20090227_v3.8.pdf

Jet Propulsion Laboratory. (2024). *Planetary Data System Standards Reference Version 1.22.0*. nasa. https://doi.org/10.17189/2ass-x557

Kaufman, S. V., Million, C. C., & St. Clair, M. A. (2022). HOW WE'LL KNOW WE CAN READ ALL THE DATA IN THE PDS: A TESTING METHODOLOGY FOR THE

PLANETARY DATA READER!!! *53rd Lunar and Planetary Science Conference*. https://www.hou.usra.edu/meetings/lpsc2022/pdf/1119.pdf

Mapbox. (2024). Rasterio. In *Github repository*. Github. https://github.com/rasterio/rasterio/tree/main

Pieters, C., Lundee, S., & Sunshine, J. (2021). *Chandrayaan-1 Orbiter Moon Mineralogy Mapper Collected Data Sets*. NASA PDS: USGS Imaging Node. https://doi.org/10.17189/f8xf-6a29

PlanetaryPy Technical Committee. (2024). *Planetarypy*. https://planetarypy.org/

Rouault, E., Warmerdam, F., Schwehr, K., Kiselev, A., Butler, H., Łoskot, M., Szekeres, T., Tourigny, E., Landa, M., Miara, I., Elliston, B., Chaitanya, K., Plesea, L., Morissette, D., Jolma, A., Dawson, N., Baston, D., de Stigter, C., & Miura, H. (2024). GDAL: Geospatial Data Abstraction Library. In *Github repository*. Github. https://doi.org/https://zenodo.org/records/12545688

Small Bodies Node, P., & Nagdimunov, L. (2021). PDS4 Tools. In *Github repository*. Github. https://github.com/Small-Bodies-Node/pds4_tools

St. Clair, M. A., Brown, S., & Million, C. (2023). *Viking Orbiter Imaging Bundle*. NASA PDS: USGS Imaging Node. https://d1nxexkqx2p6yf.cloudfront.net/PDS4/bundle.xml

St. Clair, M. A., Million, C. C., Brown, S. V., & Rice, M. S. (2023). Automated Spectral Image Processing Techniques in the Marslab Family of Applications. *6th Planetary Data Workshop*. https://www.hou.usra.edu/meetings/planetdata2023/pdf/7050.pdf

St. Clair, M. A., Million, C., & Ianno, A. (2021). *Clementine Imaging Bundle*. NASA PDS: USGS Imaging Node. https://doi.org/10.17189/07q9-ph18

USGS Astrogeology. (2024). pilo: Planetary Input/Output. In *Github repository*. Github. https://github.com/DOI-USGS/plio