The Journal of Open Source Software

Your: Your Unified Reader

Kshitij Aggarwal[1, 2], Devansh Agarwal[1, 2], Joseph W Kania[1, 2], William Fiore[1, 2], Reshma Anna Thomas[1, 2], Scott M. Ransom[3], Paul B. Demorest[4], Robert S. Wharton[5], Sarah Burke-Spolaor[1, 2], Duncan R. Lorimer[1, 2], Maura A. Mclaughlin[1, 2], and Nathaniel Garver-Daniels[1, 2]

1 West Virginia University, Department of Physics and Astronomy, P. O. Box 6315, Morgantown 26506, WV, USA **2** Center for Gravitational Waves and Cosmology, West Virginia University, Chestnut Ridge Research Building, Morgantown 26506, WV, USA **3** National Radio Astronomy Observatory, Charlottesville, VA 22903, USA **4** National Radio Astronomy Observatory, Socorro, NM, 87801, USA **5** Max-Planck-Institut für Radioastronomie, Auf dem Hügel 69, D-53121 Bonn, Germany

DOI: 10.21105/joss.02750

Software

- Review
- Repository
- Archive

Editor: Dan Foreman-Mackey

Reviewers:

- @pravirkr
- @paulray

Submitted: 28 August 2020
Published: 15 November 2020

Summary

The understanding of fast radio transients like pulsar single pulses, rotating radio transients (RRATs), and especially Fast Radio Bursts (FRBs) has evolved rapidly over the last decade. This is primarily due to dedicated campaigns by sensitive radio telescopes to search for transients. The advancement in signal processing and GPU processing systems has enabled new transient detectors at various telescopes to perform much more sensitive searches than their predecessors due to the ability to find and process FRB candidates in real-time or near-real-time. Typically the data output from the telescopes is in one of the two commonly used formats: psrfits (Hotan et al., 2004) and `Sigproc filterbank` (Lorimer, 2011). Software developed for transient searches often only works with one of these two formats, limiting their general applicability. Therefore, researchers have to write custom scripts to read/write the data in their format of choice before they can begin any data analysis relevant for their research. This has led to the development of several python libraries to manage one or the other data format (like pysigproc, psrfits, sigpyproc, etc). Still, no general tool exists which can work across data formats.

Statement of need

`Your` (Your Unified Reader) is a python-based library that unifies the data processing across multiple commonly used formats. `Your` was conceived initially to perform data ingestion for The Petabyte FRB search Project (TPP), which will uniformly search a large number of datasets from telescopes around the world for FRBs. As this project is going to process data in different formats from multiple telescopes worldwide, a unified reader was required to streamline the search pipeline. `Your` implements a user-friendly interface to read and write in the data format of choice. It also generates unified metadata corresponding to the input data file for a quick understanding of observation parameters and provides utilities to perform common data analysis operations. `Your` also provides several state-of-the-art radio frequency interference mitigation (RFI) algorithms (Agarwal, Lorimer, et al., 2020; Nita & Gary, 2010), which can now be used during any stage of data processing (reading, writing, etc.) to filter out artificial signals.

`Your` can be used at the data ingestion step of any transient search pipeline and can provide data and observation parameters in a format-independent manner. Generic tools can thus be used to perform the search and further data analysis. It also enables online processing like RFI flagging, decimation, subband search, etc.; functions for some of these are already available in `Your`. It can also be used to perform analysis of individual candidate events (using `Candidate` class): generate candidate data cutouts, create publication-ready visualizations, and perform GPU accelerated pre-processing for candidate classification (Agarwal, Aggarwal, et al., 2020). It also consists of functions to run commonly used single-pulse search software `Heimdall` (Barsdell, 2012) on any input data format.

`Your` will not only benefit experienced researchers but also new undergraduate and graduate students who otherwise have to face a significant bottleneck to understand various data formats and develop custom tools to access the data before any analysis can be done on it. Moreover, `Your` is written purely in python, which is a commonly used language within Astronomy. It also comes with comprehensive documentation and example notebooks to make it easier to get started.

`Your` uses the matplotlib library (Hunter, 2007) for plotting, and also makes use of various numpy (Harris et al., 2020), scipy (Virtanen et al., 2020), scikit-image (Van der Walt et al., 2014), numba (Lam et al., 2015) and Pandas (McKinney, 2010; The pandas development team, 2020) functions. `Your` also leverages several functions in the Astropy package (Astropy Collaboration et al., 2013; Price-Whelan et al., 2018): fits (astropy.io.fits), units (astropy.units), coordinates (astropy.coordinates) and time (astropy.time).

Acknowledgements

KA, DA, WF, SMR, PDB, SBS, DRL, MAM, and NGD are members of the NANOGrav Physics Frontiers Center, supported by NSF award number 1430284. MAM, DA, DRL, JWK, and SBS are also supported by NSF award number 1458952. DRL, MAM, DA and JWK acknowledge support from the NSF award AAG-1616042. WF acknowledges funding from the WVU STEM Mountains of Excellence graduate fellowship. RSW acknowledges financial support by the European Research Council (ERC) for the ERC Synergy Grant BlackHoleCam under contract no. 610058.

References

Agarwal, D., Aggarwal, K., Burke-Spolaor, S., Lorimer, D. R., & Garver-Daniels, N. (2020). FETCH: A deep-learning based classifier for fast transient classification. *Monthly Notices of the Royal Astronomical Society*, *497*(2), 1661–1674. https://doi.org/10.1093/mnras/staa1856

Agarwal, D., Lorimer, D. R., Surnis, M. P., Pei, X., Karastergiou, A., Golpayegani, G., Werthimer, D., Cobb, J., McLaughlin, M. A., White, S., Armour, W., MacMahon, D. H. E., Siemion, A. P. V., & Foster, G. (2020). Initial results from a real-time FRB search with the GBT. *Monthly Notices of the Royal Astronomical Society*, *497*(1), 352–360. https://doi.org/10.1093/mnras/staa1927

Astropy Collaboration, Robitaille, T. P., Tollerud, E. J., Greenfield, P., Droettboom, M., Bray, E., Aldcroft, T., Davis, M., Ginsburg, A., Price-Whelan, A. M., Kerzendorf, W. E., Conley, A., Crighton, N., Barbary, K., Muna, D., Ferguson, H., Grollier, F., Parikh, M. M., Nair, P. H., … Streicher, O. (2013). Astropy: A community Python package for astronomy. *Astronomy & Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068

Barsdell, B. R. (2012). *Advanced architectures for astrophysical supercomputing* [PhD thesis]. Swinburne University of Technology.

Harris, C. R., Millman, K. J., Walt, S. J. van der, Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., Kerkwijk, M. H. van, Brett, M., Haldane, A., Río, J. F. del, Wiebe, M., Peterson, P., … Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357–362. https://doi.org/10.1038/s41586-020-2649-2

Hotan, A. W., van Straten, W., & Manchester, R. N. (2004). PSRCHIVE and PSRFITS: An Open Approach to Radio Pulsar Data Storage and Analysis. *Publications of the Astronomical Society of Australia*, *21*(3), 302–309. https://doi.org/10.1071/AS04022

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Lam, S. K., Pitrou, A., & Seibert, S. (2015). Numba: A LLVM-Based Python JIT Compiler. *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*. https://doi.org/10.1145/2833157.2833162

Lorimer, D. R. (2011). *SIGPROC: Pulsar Signal Processing Programs* (p. ascl:1107.016).

McKinney, Wes. (2010). Data Structures for Statistical Computing in Python. In Stéfan van der Walt & Jarrod Millman (Eds.), *Proceedings of the 9th Python in Science Conference* (pp. 56–61). https://doi.org/10.25080/Majora-92bf1922-00a

Nita, G. M., & Gary, D. E. (2010). The generalized spectral kurtosis estimator. *Monthly Notices of the Royal Astronomical Society*, *406*(1), L60–L64. https://doi.org/10.1111/j.1745-3933.2010.00882.x

Price-Whelan, A. M., Sipőcz, B. M., Günther, H. M., Lim, P. L., Crawford, S. M., Conseil, S., Shupe, D. L., Craig, M. W., Dencheva, N., Ginsburg, A., VanderPlas, J. T., Bradley, L. D., Pérez-Suárez, D., de Val-Borro, M., Paper Contributors, (Primary., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., … Contributors, (Astropy. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *The Astronomical Journal*, *156*, 123. https://doi.org/10.3847/1538-3881/aabc4f

The pandas development team. (2020). *pandas-dev/pandas: Pandas* (latest) [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.3509134

Van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: image processing in Python. *PeerJ*, *2*, e453.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., … van Mulbregt, P. (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, *17*, 261–272. https://doi.org/10.1038/s41592-019-0686-2