

# Astronomical échelle spectroscopy data analysis with muler

Michael A. Gully-Santiago<sup>1</sup>, Jessica L. Luna<sup>1</sup>, Caroline V. Morley<sup>1</sup>, Kyle Kaplan<sup>2</sup>, Aishwarya Ganesh<sup>1</sup>, Joel Burke<sup>1</sup>, and Daniel M. Krolikowski<sup>1</sup>

<sup>1</sup> The University of Texas at Austin Department of Astronomy, Austin, TX, USA <sup>2</sup> SOFIA Science Center, Universities Space Research Association, NASA Ames Research Center, Moffett Field, CA, USA

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) ↗
- [Repository](#) ↗
- [Archive](#) ↗

Editor: [Juanjo Bazán](#) ↗

## Reviewers:

- [@bmorris3](#)
- [@wtgee](#)

Submitted: 28 February 2022

Published: unpublished

## License

Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](#)).

## Summary

Modern astronomical échelle spectrographs produce information-rich 2D echellograms that undergo standard reduction procedures to produce extracted 1D spectra. The final post-processing steps of these 1D spectra are often left to the end-user scientists, since the order of operations and algorithm choice may depend on the scientific application. Implementing these post-processing steps from scratch acts as a barrier to entry to newcomers, taxes scientific innovation, and erodes scientific reproducibility. Here we assemble and streamline a collection of spectroscopic data analysis methods into a single easy-to-use Application Programming Interface (API), *muler*. The *specutils*-based fluent interface enables method chaining that yields compact 1-line code for many applications, a dramatic reduction in complexity compared to existing industry practices. When applicable, some algorithms are customized to the pipeline data products of three near-infrared spectrographs: HPF, Keck NIRSPEC, and IGRINS. The framework could be extended to other spectrographs in the future. The tutorials in *muler* may also serve as a central onboarding point for new entrants to the practice of near-infrared échelle spectroscopy data. The open-source permissively licensed Python 3 implementation lowers the barrier to entry and accelerates the investigation of astronomical spectra.

## Statement of need

In its most elemental form, an astronomical spectrum consists of a list of wavelength coordinates and their flux values. Additional metadata for uncertainties, date, time, sky pointing coordinates, and instrumental configuration are common. The Astropy-affiliated *specutils* package serves as a unifying container for virtually any astronomical spectrum, by providing a way to represent, load, manipulate, and analyze these data and metadata. It is therefore possible for each new practitioner to load a pipeline-produced output spectrum with, *numpy* arrays, *pandas* dataframes, or *specutils* objects and write standard post-processing operations from scratch. This bespoke approach requires expert knowledge, is time-consuming, and may be error prone. Ideally, we would have an easy-to-use post-processing tool that takes in the raw 1D pipeline output filename and produces a refined spectrum ready for scientific analysis. The *muler* framework fills this role.

Each spectrograph and pipeline has peculiarities, and so *muler* provides domain-specific layers for individual spectrographs. The *muler* framework therefore enables user code to be compact and easily transferable among projects.

The Python spectroscopy ecosystem has dozens or hundreds of packages. Here we compare *muler* to a few recent examples to show how it fits into that ecosystem. The *wobble* ([Bedell et al., 2019](#)) framework performs advanced modeling on groups of dozens of échelle spectra of

the same celestial object, assuming they have already been normalized, continuum-flattened, sky subtracted, etc. So in this case, `muler` serves as the layer that takes in the pipeline output, processes each spectrum identically, and then feeds these refined spectra into `wobble`. Similarly, the starfish framework (Czekala et al., 2015) expects spectra to reside in an HDF5 format with telluric lines either corrected or masked. `muler` can satisfy these requirements. You can therefore view `muler` as taking care of these necessary pre-processing steps to get from the pipeline products to the inputs of virtually any other Python spectroscopy analysis package.

Some packages such as `igrins_rv` (Tang et al., 2021) conduct pre-processing steps as a matter of course in their analyses. These packages could hypothetically be refactored to make `muler` a dependency, with the boilerplate steps taken over by `muler`. This refactoring step would simplify the code, possibly making it more maintainable in the long term. In other cases, popular methods in `muler` may move upstream into `specutils`.

`muler` depends on `astropy` (Astropy Collaboration et al., 2018, 2013), `numpy` (Harris et al., 2020), `specutils`, `scipy` (Virtanen et al., 2020), and others.

## Architecture and design considerations

The design of `muler` shares three key inspirations from `lightcurve` (Lightcurve Collaboration et al., 2018), which provides a well-liked open-source fluent interface to NASA Kepler/K2/TESS data. First, pipeline products have reliable and predictable formats that make them a *de facto* standard. You can then make an API that relies on that standard for accessing attributes and metadata, and build methods that combine the data, metadata, and optionally user inputs to conduct essentially all routine operations.

The second key idea is that the output of an operation is usually self-similar: a `lightcurve` operation usually takes in a `lightcurve` and outputs a `lightcurve`. Here a `muler` operation takes in a `Spectrum1D`-like object and returns a `Spectrum1D`-like. This key idea makes method chaining both possible and desirable.

Finally, `lightcurve` objects inherit key behaviors from the `Astropy TimeSeries` module, reducing the duplication of code relevant to time series data. Here in `muler` we gain similar extensibility through an object-oriented design with `specutils`, abstracting away many routine spectroscopy tasks, and keeping the `muler` code lean.

## Supported spectrographs

We currently support custom operations for three spectrographs: the Immersion Grating Infrared Spectrograph, IGRINS (Mace et al., 2018; Park et al., 2014) reduced with the `plp` (Lee et al., 2017); the Habitable Zone Planet Finder Spectrograph, HPF (Mahadevan et al., 2012, 2014, 2018) with either the Goldilocks or Penn State pipelines; and the Keck Near-Infrared Spectrograph, NIRSPEC (McLean et al., 1998, 2000) with the NSDRP pipeline.

In principle, `muler` could be extended with devoted classes for other échelle spectrographs, so long as the pipeline outputs deliver standardized files with common metadata, such as date, time, celestial coordinates, etc.

## Acknowledgements

This material is based upon work supported by the National Aeronautics and Space Administration under Grant Numbers 80NSSC21K0650 for the NNH20ZDA001N-ADAP:D.2 program, and 80NSSC20K0257 for the XRP program issued through the Science Mission Directorate. C.V.M., J.L., and M.A.G-S. acknowledge the National Science Foundation, which supported

the work presented here under Grant No. 1910969. This research has made use of NASA's Astrophysics Data System Bibliographic Services.

## References

- Astropy Collaboration, 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., Aldcroft, T. L., Cruz, K. L., Robitaille, T. P., Tollerud, E. J., ... Astropy Contributors. (2018). The Astropy Project: Building an Open-science Project and Status of the v2.0 Core Package. *156*, 123. <https://doi.org/10.3847/1538-3881/aabc4f>
- 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. *558*, A33. <https://doi.org/10.1051/0004-6361/201322068>
- Bedell, M., Hogg, D. W., Foreman-Mackey, D., Montet, B. T., & Luger, R. (2019). WOBBLE: A Data-driven Analysis Technique for Time-series Stellar Spectra. *158*(4), 164. <https://doi.org/10.3847/1538-3881/ab40a7>
- Czekala, I., Andrews, S. M., Mandel, K. S., Hogg, D. W., & Green, G. M. (2015). Constructing a Flexible Likelihood Function for Spectroscopic Inference. *812*, 128. <https://doi.org/10.1088/0004-637X/812/2/128>
- 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., R'io, 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>
- Lee, J.-J., Gullikson, K., & Kaplan, K. (2017). *Igrins/plp 2.2.0*. Zenodo. <https://doi.org/10.5281/zenodo.845059>
- Lightkurve Collaboration, Cardoso, J. V. d. M., Hedges, C., Gully-Santiago, M., Saunders, N., Cody, A. M., Barclay, T., Hall, O., Sagar, S., Turtelboom, E., Zhang, J., Tzanidakis, A., Mighell, K., Coughlin, J., Bell, K., Berta-Thompson, Z., Williams, P., Dotson, J., & Barentsen, G. (2018). *Lightkurve: Kepler and TESS time series analysis in Python*. Astrophysics Source Code Library.
- Mace, G., Sokal, K., Lee, J.-J., Oh, H., Park, C., Lee, H., Good, J., MacQueen, P., Oh, J. S., Kaplan, K., Kidder, B., Chun, M.-Y., Yuk, I.-S., Jeong, U., Pak, S., Kim, K.-M., Nah, J., Lee, S., Yu, Y.-S., ... Jaffe, D. T. (2018). IGRINS at the Discovery Channel Telescope and Gemini South. In C. J. Evans, L. Simard, & H. Takami (Eds.), *Ground-based and airborne instrumentation for astronomy VII* (Vol. 10702, p. 107020Q). <https://doi.org/10.1117/12.2312345>
- Mahadevan, S., Anderson, T., Balderrama, E., Bender, C. F., Bevins, E., Blakeslee, S., Cole, A., Conran, D., Diddams, S., Dykhouse, A., Darling, J., Fredrick, C., Halverson, S., Hearty, F., Jennings, J., Kaplan, K., Kanodia, S., Levi, E., Lubar, E., ... Wright, J. (2018). The habitable-zone planet finder: engineering and commissioning on the Hobby Eberly telescope (Conference Presentation). In C. J. Evans, L. Simard, & H. Takami (Eds.), *Ground-based and airborne instrumentation for astronomy VII* (Vol. 10702). International Society for Optics; Photonics; SPIE. <https://doi.org/10.1117/12.2313835>
- Mahadevan, S., Ramsey, L. W., Terrien, R., Halverson, S., Roy, A., Hearty, F., Levi, E., Stefansson, G. K., Robertson, P., Bender, C., Schwab, C., & Nelson, M. (2014). The

- 132 Habitable-zone Planet Finder: A status update on the development of a stabilized fiber-fed  
133 near-infrared spectrograph for the for the Hobby-Eberly telescope. In S. K. Ramsay, I. S.  
134 McLean, & H. Takami (Eds.), *Ground-based and airborne instrumentation for astronomy*  
135 v (Vol. 9147, p. 91471G). <https://doi.org/10.1117/12.2056417>
- 136 Mahadevan, S., Ramsey, L., Bender, C., Terrien, R., Wright, J. T., Halverson, S., Hearty,  
137 F., Nelson, M., Burton, A., Redman, S., Osterman, S., Diddams, S., Kasting, J., Endl,  
138 M., & Deshpande, R. (2012). The habitable-zone planet finder: a stabilized fiber-fed  
139 NIR spectrograph for the Hobby-Eberly Telescope. In I. S. McLean, S. K. Ramsay, & H.  
140 Takami (Eds.), *Ground-based and airborne instrumentation for astronomy IV* (Vol. 8446,  
141 p. 84461S). <https://doi.org/10.1117/12.926102>
- 142 McLean, I. S., Becklin, E. E., Bendiksen, O., Brims, G., Canfield, J., Figer, D. F., Graham,  
143 J. R., Hare, J., Lacayanga, F., Larkin, J. E., Larson, S. B., Levenson, N., Magnone, N.,  
144 Teplitz, H., & Wong, W. (1998). Design and development of NIRSPEC: a near-infrared  
145 echelle spectrograph for the Keck II telescope. In A. M. Fowler (Ed.), *Infrared astronomical*  
146 *instrumentation* (Vol. 3354, pp. 566–578). <https://doi.org/10.1117/12.317283>
- 147 McLean, I. S., Graham, J. R., Becklin, E. E., Figer, D. F., Larkin, J. E., Levenson, N. A., &  
148 Teplitz, H. I. (2000). Performance and results with the NIRSPEC echelle spectrograph  
149 on the Keck II telescope. In M. Iye & A. F. Moorwood (Eds.), *Optical and IR telescope*  
150 *instrumentation and detectors* (Vol. 4008, pp. 1048–1055). [https://doi.org/10.1117/12.](https://doi.org/10.1117/12.395422)  
151 [395422](https://doi.org/10.1117/12.395422)
- 152 Park, C., Jaffe, D. T., Yuk, I.-S., Chun, M.-Y., Pak, S., Kim, K.-M., Pavel, M., Lee,  
153 H., Oh, H., Jeong, U., Sim, C. K., Lee, H.-I., Nguyen Le, H. A., Strubhar, J., Gully-  
154 Santiago, M., Oh, J. S., Cha, S.-M., Moon, B., Park, K., ... Park, B.-G. (2014). Design  
155 and early performance of IGRINS (Immersion Grating Infrared Spectrometer). *Society*  
156 *of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 9147, 1. <https://doi.org/10.1117/12.2056431>
- 157  
158 Tang, S.-Y., Stahl, A., Johns-Krull, C., Prato, L., & Llama, J. (2021). IGRINS RV: A Python  
159 Package for Precision Radial Velocities with Near-Infrared Spectra. *The Journal of Open*  
160 *Source Software*, 6(62), 3095. <https://doi.org/10.21105/joss.03095>
- 161 Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D.,  
162 Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M.,  
163 Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., ...  
164 SciPy 1.0 Contributors. (2020). SciPy 1.0: fundamental algorithms for scientific computing  
165 in Python. *Nature Methods*, 17, 261–272. <https://doi.org/10.1038/s41592-019-0686-2>