The Journal of Open Source Software

Luci: A Python package for SITELLE spectral analysis

Carter L. Rhea[1, 2], **Julie Hlavacek-Larrondo**[1], **and Benjamin Vigneron**[1]

1 Département de Physique, Université de Montréal, Succ. Centre-Ville, Montréal, Québec, H3C 3J7, Canada **2** Centre de Recherche en Astrophysique du Québec (CRAQ), Québec, QC, G1V 0A6, Canada

DOI: 10.21105/joss.03518

Software

- Review
- Repository
- Archive

Editor: Pending Editor

Submitted: 20 July 2021
Published: 21 July 2021

Summary

High-resolution optical integral field units (IFUs) are rapidly expanding our knowledge of extragalactic emission nebulae in galaxies and galaxy clusters. By studying the spectra of these objects – which include classic HII regions, supernova remnants, planetary nebulae, and cluster filaments – we are able to constrain their kinematics (velocity and velocity dispersion). In conjunction with additional tools, such as the BPT diagram (e.g. Baldwin et al. (1981); Kewley et al. (2006)), we can further classify emission regions based on strong emission-line flux ratios. `LUCI` is a simple-to-use python module intended to facilitate the rapid analysis of IFU spectra. `LUCI` does this by integrating well-developed pre-existing python tools such as `astropy` and `scipy` with new machine learning tools for spectral analysis (Rhea et al. 2020a).

Statement of Need

Recent advances in the science and technology of IFUs have resulted in the creation of the high-resolution, wide field-of-view (11 arcmin x 11 arcmin) instrument SITELLE (Drissen et al. (2019)) at the Canada-France-Hawaii Telescope. Due to the large field-of-view and the small angular resolution of the pixels (0.32 arcseconds), the resulting data cubes contain over 4 million spectra. Therefore, a simple, fast, and adaptable fitting code is paramount – it is with this in mind that we created `LUCI`.

Functionality

At her heart, like any fitting software, `LUCI` is nothing more than a collection of pre-processing and post-processing functions to extract information from a spectrum using a fitting function (in this case a `scipy.optimize.minimize` function call). Since SITELLE data cubes are available as **HDF5** files, `LUCI` was built to parse the original file and create an instance of a `LUCI` cube which contains the 2D header information and a 3D numpy array (Spatial X, Spatial Y, Spectral). Once the data cube has been successfully converted to a `LUCI` cube, there are several options for fitting different regions of the cube (e.g., `fit_cube`, `fit_entire_cube`, `fit_region`) or fitting single spectra (e.g., `fit_spectrum_region`). The primary use case of `LUCI` is to fit a region of a cube defined either as a box (in this case, the user would employ the `fit_cube` method and pass the limits of the bounding box) or to fit a region of the cube defined by a standard *ds9* file (in this case, the user would pass the name of the region file to `fit_region`). Regardless of the region being fit, the user need only specify the lines they wish to fit and the fitting function. We currently support all standard lines in the SN1 filter ([OII3626] & [OII3629]), SN2 filter ([OIII4959], [OIII5007], & Hbeta), and SN3 filter ([SII6716], [SII6731], [NII6548], [NII6583], & Halpha).

JOSS
The Journal of Open Source Software

The user also must chose between three fitting functions: a pure Gaussian, and pure sinc function, or a sinc function convolved with a Gaussian (T. B. Martin et al. (2016)). In either case, `LUCI` will solve for the three primary quantities of interest which are the **amplitude** of the line, the **position** of the line (often described as the velocity and quoted in km/s), and the **broadening** of the line (often described as the velocity dispersion and quoted in units of km/s

The three fitting functions are mathematically described below where p_0 corresponds to the **amplitude**, p_1 corresponds to the **position**, and p_2 corresponds of the **broadening**.

The pure Gaussian function is expressed as

$$f(x) = p_0 * exp(-(x-p1)^2/(2*p_2^2)) \tag{1}$$

The pure since function is expressed as

$$f(x) = p_0 * (\frac{(x-p_1)/p_2}{(x-p_1)/p_2}) \tag{2}$$

The convolved sincgauss function is expressed as

$$f(x) = p_0 * exp(-b*^2) * ((erf(a-i*b) + erf(a+i*b))/(2*erf(a))) \tag{3}$$

where x represents a given spectral channel, $a = p_2/(\sqrt{2}*\sigma)$, $b = (x-p_1)/(\sqrt{(2)}*\sigma)$, where σ is the pre-defined width of the sinc function. We define this following (REF) as $\sigma = \frac{1}{2*MPD}$ where **MPD** is the maximum path difference.

In each case, after solving for these values, the velocity and velocity dispersion are calculated using the following equations:

$$v[km/s] = 3e5 * ((p_1' - v_0)/v_0) \tag{4}$$

where $3e5$ represents the speed of light in kilometers per second, p_1' is p_1 in nanometers, and v_0 is the reference wavelength of the line in nanometers.

$$\sigma[km/s] = 3e5 * (p_2/p_1) \tag{5}$$

where again $3e5$ represents the speed of light in kilometers per second.

Similarly, we define the flux for each fitting function as the following: *Flux for a Gaussian Function*:

$$Flux[erg/s/cm^2/Ang] = \sqrt{2\pi}p_0p_2 \tag{6}$$

Flux for a Sinc Function:

$$Flux[erg/s/cm^2/Ang] = \pi p_0p_2 \tag{7}$$

Flux for a SincGauss Function:

$$Flux[erg/s/cm^2/Ang] = p_0\frac{\sqrt{2\pi}p_2}{erf(\frac{p_2}{\sqrt{2}\sigma})} \tag{8}$$

A full Bayesian approach is implemented in order to determine uncertainties on the three key fitting parameters (p_0, p_1, and p_2) using the python `emcee` package (Foreman-Mackey et al. (2013)). Thus, we are able to calculate posterior distributions for each parameter.

Other Software

Several fitting software packages exist for fitting generalized functions to optical spectra (such as `astropy`; Robitaille et al. (2013)). Additionally, there exist software for fitting IFU datacubes for several instruments such as MUSE (Richard et al. (2012)) and SITELLE (T. Martin et al. (2012)). Although these are mature codes, we opted to write our own fitting package that is transparent to users and highly customize-able.

Acknowledgements

C. L. R. acknowledges financial support from the physics department of the Université de Montréal, IVADO, and le fonds de recherche – Nature et Technologie. J. H.-L. acknowledges support from NSERC via the Discovery grant program, as well as the Canada Research Chair program.

References

Baldwin, J. A., Phillips, M. M., & Terlevich, R. (1981). Classification parameters for the emission-line spectra of extragalactic objects. *Publications of the Astronomical Society of the Pacific*, *93*, 5. https://doi.org/10.1086/130766

Drissen, L., Martin, T., Rousseau-Nepton, L., Robert, C., Martin, R. P., Baril, M., Prunet, S., Joncas, G., Thibault, S., Brousseau, D., Mandar, J., Grandmont, F., Yee, H., & Simard, L. (2019). SITELLE: An imaging fourier transform spectrometer for the canada–france–hawaii telescope. *Monthly Notices of the Royal Astronomical Society*, *485*(3), 3930–3946. https://doi.org/10.1093/mnras/stz627

Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. (2013). Emcee: The MCMC hammer. *Publications of the Astronomical Society of the Pacific*, *125*(925), 306–312. https://doi.org/10.1086/670067

Kewley, L. J., Groves, B., Kauffmann, G., & Heckman, T. (2006). The host galaxies and classification of active galactic nuclei. *Monthly Notices of the Royal Astronomical Society*, *372*(3), 961–976. https://doi.org/10.1111/j.1365-2966.2006.10859.x

Martin, T. B., Prunet, S., & Drissen, L. (2016). Optimal fitting of gaussian-apodized or under-resolved emission lines in fourier transform spectra providing new insights on the velocity structure of NGC 6720. *Monthly Notices of the Royal Astronomical Society*, *463*(4), 4223–4238. https://doi.org/10.1093/mnras/stw2315

Martin, T., Drissen, L., & Joncas, G. (2012). ORBS: A data reduction software for the imaging fourier transform spectrometers SpIOMM and SITELLE. *Software and Cyberinfrastructure for Astronomy II*, *8451*, 84513K. https://doi.org/10.1117/12.925420

Richard, J., Bacon, R., Weilbacher, P. M., Streicher, O., Wisotzki, L., Herenz, E. C., Slezak, E., Petremand, M., Jalobeanu, A., Collet, C., & Louys, M. (2012). *Reduction and analysis of MUSE data*. 553–557. https://ui.adsabs.harvard.edu/abs/2012sf2a.conf..553R

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., Günther, H. M., … Streicher, O. (2013). Astropy: A community python package for astronomy. *Astronomy & Astrophysics*, *558*, A33. https://doi.org/10.1051/0004-6361/201322068