LUCI: A Python package for SITELLE spectral analysis

Carter Rhea,¹, 2 Julie Hlavacek-Larrondo,¹ Laurie Rousseau-Nepton,³ Benjamin Vigneron,¹ and Louis-Simon Guité¹

¹Département de Physique, Université de Montréal, Succ. Centre-Ville, Montréal, Québec, H3C 3J7, Canada
²Centre de recherche en astro physique du Québec (CRAQ)
³Canada-France-Hawaii Telescope, Kamuela, HI, United States

(Received June 1, 2019; Revised January 10, 2019; Accepted August 31, 2021)

Submitted to RNAAS

ABSTRACT

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. 2020). Furthermore, LUCI provides several easy-to-use tools to access and fit SITELLE data cubes.

Keywords: SITELLE data reduction, IFU, Machine Learning

STATEMENT OF NEED

Recent advances in the science and technology of integral field units (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 a collection of pre-processing and post-processing functions coupled with a fitting algorithm (in this case, a scipy.optimize.minimize function call) to extract information from a spectrum. That being said, LUCI is more than a simple wrapper for a fitting 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 needs only specify the lines they wish to fit, the fitting function, and the constraint relations between the lines.

Corresponding author: Carter Lee Rhea
carterrhea@astro.umontreal.ca
We currently support all standard lines in the SN1 filter ([O\textsc{ii}]3626 & [O\textsc{ii}]3629), SN2 filter ([O\textsc{ii}]4959, [O\textsc{iii}]5007, & H\textbeta), and SN3 filter ([S\textsc{ii}]6716, [S\textsc{ii}]6731, [N\textsc{ii}]6548, [N\textsc{ii}]6583, & H\alpha). The user chooses between three fitting functions: a pure Gaussian, and pure sinc function, or a sinc function convolved with a Gaussian (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). LUCI’s suite of post-processing tools allows us to extract kinematic parameters and line fluxes directly from these quantities.

**Fitting Functions**

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

The pure Gaussian function is expressed as

$$f(x) = p_0 \exp \left\{ \frac{-(x - p_1)^2}{(2 * p_2^2)} \right\}$$  \hspace{1cm} (1)

The pure sinc function is expressed as

$$f(x) = p_0 \frac{\sin((x - p_1)/p_2)}{(x - p_1)/p_2}$$  \hspace{1cm} (2)

The convolved sincgauss function is expressed as

$$f(x) = p_0 \exp \{-b^2\} \left( \frac{\text{erf}(a - ib) + \text{erf}(a + ib)}{2\text{erf}(a)} \right)$$  \hspace{1cm} (3)

where $x$ represents a given spectral channel, $a = \frac{p_2}{\sqrt{2} \sigma}$, $b = \frac{x - p_1}{\sqrt{2} \sigma}$, where $\sigma$ is the pre-defined width of the sinc function. We define this as $\sigma = \frac{1}{2 \text{MPD}}$ where MPD is the maximum path difference. We note that erf is the error function.

**Kinematic Equations**

In each case, after solving for these values, the velocity and velocity dispersion (also called the broadening) are calculated using the following equations:

$$v[\text{km/s}] = 3c5 \ast \left( \frac{p_1^\prime - v_0}{v_0} \right)$$  \hspace{1cm} (4)

where $3c5$ represents the speed of light in kilometers per second, $p_1^\prime$ is $p_1$ in nanometers, and $v_0$ is the reference wavelength of the line in nanometers.

$$\sigma[\text{km/s}] = 3c5 \ast \left( \frac{p_2}{p_1} \right)$$  \hspace{1cm} (5)

where again $3c5$ represents the speed of light in kilometers per second.

**Flux Equations**

Similarly, we define the flux for each fitting function.

Flux for a Gaussian Function:

$$\text{Flux}[\text{erg/s/cm}^2] = \sqrt{2\pi} p_0 p_2$$  \hspace{1cm} (6)

Flux for a sinc Function:

$$\text{Flux}[\text{erg/s/cm}^2] = \pi p_0 p_2$$  \hspace{1cm} (7)

Flux for a sincgauss Function:

$$\text{Flux}[\text{erg/s/cm}^2] = p_0 \frac{\sqrt{2\pi} p_2}{\text{erf}(\frac{p_2}{\sqrt{2} \sigma})}$$  \hspace{1cm} (8)
Uncertainty Estimates

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 emcee python package (Foreman-Mackey et al. 2013). It can be activated by added bayes_bool=True to the argument of any fitting function. Thus, we are able to calculate posterior distributions for each parameter. There is a dedicated discussion to this in our online documentation. Additionally, LUCI can calculate the uncertainties by calculating the covariance matrix of the fit; this allows users to calculate uncertainties using a classical method by adding the argument uncertainty_bool=True to any fitting function (again, there is a dedicated discussion on this calculation in our online documentation).

DISCUSSION

Installation, Documentation, & Examples

LUCI is available as an open-source software package on our GitHub page: https://github.com/crhea93/LUCI. Moreover, we have included, on this page, a detailed set of instructions to install and use the software package. We have included detailed documentation outlining the features of LUCI and the mathematics behind her at https://crhea93.github.io/LUCI/index.html. On this site, we have included several key examples that can be adapted to any use-case of the software.

Other Software

Several fitting software packages exist for fitting generalized functions to optical spectra (such as astropy; Collaboration et al. 2013; Astropy Collaboration et al. 2018). Additionally, there exist software for fitting IFU datacubes for several instruments such as MUSE (Richard et al. 2012) and SITELLE (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.

ACKNOWLEDGMENTS

The authors would like to thank the Canada-France-Hawaii Telescope (CFHT), which is operated by the National Research Council (NRC) of Canada, the Institut National des Sciences de l’Univers of the Centre National de la Recherche Scientifique (CNRS) of France, and the University of Hawaii. The observations at the CFHT were performed with care and respect from the summit of Maunakea, which is a significant cultural and historic site.

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.

Software: astropy (Collaboration et al. 2013; Astropy Collaboration et al. 2018), scipy (Virtanen et al. 2020), numpy (Harris et al. 2020), matplotlib (Hunter 2007), ipython (Perez & Granger 2007), tensorflow (Abadi et al. 2015), keras (Chollet 2015), emcee (Foreman-Mackey et al. 2013)

REFERENCES

Abadi, M., Agarwal, A., Barham, P., et al. 2015, 19
Astropy Collaboration, Price-Whelan, A. M., Sipőcz, B. M., et al. 2018, The Astronomical Journal, 156, 123,
doi: 10.3847/1538-3881/aabc4f
Baldwin, J. A., Phillips, M. M., & Terlevich, R. 1981,
Publications of the Astronomical Society of the Pacific, 93, 5, doi: 10.1086/130766
Chollet, F. 2015, Keras. https://keras.io
Collaboration, A., Robitaille, T. P., Tollerud, E. J., et al.
2013, Astronomy and Astrophysics, 558, A33,
doi: 10.1051/0004-6361/201322068
Drissen, L., Martin, T., Rousseau-Nepton, L., et al. 2019,
Monthly Notices of the Royal Astronomical Society, 485, 3930, doi: 10.1093/mnras/stz627
Foreman-Mackey, D., Hogg, D. W., Lang, D., & Goodman, J. 2013, Publications of the Astronomical Society of the Pacific, 125, 306, doi: 10.1086/670067
Harris, C. R., Millman, K. J., van der Walt, S. J., et al.
2020, Nature, 585, 357, doi: 10.1038/s41586-020-2649-2
Hunter, J. D. 2007, Computing in Science Engineering, 9, 90, doi: 10.1109/MCSE.2007.55
Kewley, L. J., Groves, B., Kauffmann, G., & Heckman, T. 2006, Monthly Notices of the Royal Astronomical Society, 372, 961, doi: 10.1111/j.1365-2966.2006.10859.x

Martin, T., Drissen, L., & Joncas, G. 2012, in Software and Cyberinfrastructure for Astronomy II, Vol. 8451 (International Society for Optics and Photonics), 84513K, doi: 10.1117/12.925420

Martin, T. B., Prunet, S., & Drissen, L. 2016, Monthly Notices of the Royal Astronomical Society, 463, 4223, doi: 10.1093/mnras/stw2315

Perez, F., & Granger, B. E. 2007, Computing in Science Engineering, 9, 21, doi: 10.1109/MCSE.2007.53

Rhea, C. L., Rousseau-Nepton, L., Prunet, S., Hlavacek-larrondo, J., & Fabbro, S. 2020, Astrophysical Journal, 901. http://arxiv.org/abs/2008.08093

Richard, J., Bacon, R., Weilbacher, P. M., et al. 2012, 553. https://ui.adsabs.harvard.edu/abs/2012sf2a.conf..553R

Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nature Methods, 17, 261, doi: 10.1038/s41592-019-0686-2