The Journal of Open Source Software

Eniric: Extended NIR Information Content

J.J. Neal[1, 2] **and P. Figueira**[3, 1]

1 Instituto de Astrofísica e Ciências do Espaço, Universidade do Porto, CAUP, Rua das Estrelas, 4150-762, Porto, Portugal **2** Departamento de Física e Astronomia, Faculdade de Ciências, Universidade do Porto, Rua do Campo Alegre, 4169-007, Porto, Portugal **3** European Southern Observatory, Alonso de Córdova 3107, Vitacura, Casilla 19001, Santiago 19, Chile

DOI: 10.21105/joss.01053

Software

- Review
- Repository
- Archive

Submitted: 11 September 2018
Published: 08 May 2019

With recent high-precision spectrographs targeting radial velocity (RV) precision at the 10 cms^{-1} level (e.g. F. Pepe et al., 2014) in the quest to find smallest planets, it is important to understand the theoretical precision attainable in stellar spectra. *Eniric* provides a simple way to calculate the theoretical spectral quality and RV precision (i.e., information content) of synthetic and/or observed stellar spectra given vectors of wavelength and photon flux.

Written in *Python 3*, *Eniric* calculates the fundamental photon noise RV precision as formulated in Connes (1985) and F. Bouchy, Pepe, & Queloz (2001). It is an improved version of the software used in Figueira et al. (2016) for calculating the RV precision of synthetic M-dwarf spectra in the near-infrared (NIR) bands. The code was refactored, with hard-coded constraints removed, making it faster and simpler to explore a larger combination of parameters (e.g. not limited to M-dwarfs and NIR wavelengths).

Eniric contains several independent functions to transform observed and synthetic spectra, such as wavelength selection, broadening, SNR normalization and to compute RV precisions.

Eniric performs rotational and instrumental broadening of spectra through convolution with a rotational kernel (Gray, 2005) and gaussian kernel respectively. Both kernels are wavelength dependant and do **not** require a uniformly spaced wavelength vector, unlike the convolution functions given in PyAstronomy. *Eniric* utilizes the *embarrassingly parallel* nature of the convolutions (each pixel can be calculated independently of its neighbours) to compute the convolutions in parallel; the convolution results are also cached using Joblib to avoid re-computation. This improves the convolution performance but not to the level achievable by algorithms that require an equal wavelength spacing and use fixed kernels (only valid for small wavelength regions), e.g. the "fast" convolutions provided in PyAstronomy.

Eniric enables the relative precision between synthetic spectra by allowing for normalization to a user defined signal-to-noise ratio (SNR) per pixel at a specific wavelength. Although user definable the default choice is a SNR of 100 at the center of the J-band (1.25 μm) as used in Figueira et al. (2016).

The precision calculations are not limited to the large spectroscopic bands, but can also be performed on narrow wavelength slices along the entire spectrum. This allows one to explore the RV precision across the entire spectrum and perform comparision between observations and synthetic libraries (e.g. É. Artigau et al., 2018).

Extraneous information not included in the spectra (i.e., not photon noise nor line content information) can be included in the precision calculation through the use of a spectral mask. This mask can be used to indicate which spectral lines are to be included/excluded (via a binary mask) or if some spectral lines should receive more statistical weight for an external reason. For example, masks derived from an atmospheric absorption spectrum can be used to explore the treatment and correction the atmospheric absorption on the