Spectre Normalization Tool (SNT)
User Guide

Description

This project was done in the context of an introduction to research grant (Bolsa de Iniciação à Investigação - BII) at Instituto de Astrofísica e Ciências do Espaço (http://www.iastro.pt/) by Diogo G.S. Marques under the guidance of professor Nuno C.Santos and Phd student André Silva.

This is a software, written in Python, that uses the computational geometry concept of alpha-shape to produce an upper envelope for the spectre. For this end, relevant points are selected by the algorithm, called "anchors" which are then, after some processing to remove outliers, interpolated by a linear or cubic polynomial. The final output of the program is the function that models a continuum line of the upper envelope of the flux values.

The algorithm is based on the methodology described in M. Cretignier et alt, 2020, RASSINE and Xu et alt. 2019.

How to Run

Before running make sure you have the latest version of python interpreter installed, and all the needed packages:

- Astropy (https://www.astropy.org/)
- Matplotlib (https://matplotlib.org/)
- Numpy (https://numpy.org/)
- Scipy (https://scipy.org/)

To run simply use the terminal with the following prompt, depending on your version:

python3 snt.py **spectrename.fits output.csv**

or

python snt.py **spectrename.fits output.csv**

Where **spectrename.fits** should be replaced by the name/path of the spectre you wish to analyse and **output.csv** is the name of the file where the output will be written.

The program also exports a file "anchors.csv" that contains the algorithm selected points before interpolation, this is useful to use different parameters for certain wavelength intervals. To do this, simple run the algorithm over the whole spectre again with the new parameters, and then select only the points in "anchors.csv" in the interval you wish to consider (make sure to save the "anchors.csv" file from the previous iteration). Then, run a linear or cubic interpolation on all the points. This will in fact produce a continuum using anchors from several different iterations by selecting the points in the different "anchors.csv" file in the intended intervals.

Parameters

The algorithm uses several parameters to produce the continuum, these can be modified altering the "config.py" file. They are as follows:

- **remove_n_first –** Removes **n** first points. If the first points in the spectre correspond to noise. For ESPRESSO data, this corresponds roughly to the first 3000 points, thus this is the default value for this parameter. If **n**=0 no points are removed.

Fig1- First points of the spectre. Removes the highlighted part.

- **radius_min** – Starting alpha-shape radius (rolling pin). Several values can produce a good continuum, the smaller the radius more detail will be included, should be a value compatible with the current **stretching** parameter. Good defaults in the range [20, 80] Å.
- **radius_max –** Maximum alpha-shape radius (rolling pin). This value should be at least the size of the largest gap, to make sure the pin does not "fall". Default for this value is 70 Å.
- **max_vicinity –** when first calculating the maxima that will be considered by the rolling pin algorithm this parameter imposes a required number of points between adjacent maxima. The larger it is, less points are considered, which makes the code run faster, but with less detail. This parameter is not very important with a significant enough large radius and stretching, because most adjacent non relevant points will not be selected, but it should have a small value if more detail is to be considered. Default is 10 points. Acceptable values between [0, 100] points.
- **stretching –** This, along with radius_min, is one of the most important parameters to produce different quality continua. This is the scaling of the Y-axis in relation to the X-axis, and can be seen as the inverse tension of the continuum line on the spectrum. If this value is too high the rolling pin will "fall" and will not be able to select relevant maxima. Several reasonable values will produce acceptable continua with different detail levels. Acceptable values in the range [20, 100]. This value should only be set very high, along with a very small **radius_min**, to capture zones with a very high detail (approximating the flux values in this zones), as seen in Fig 5.

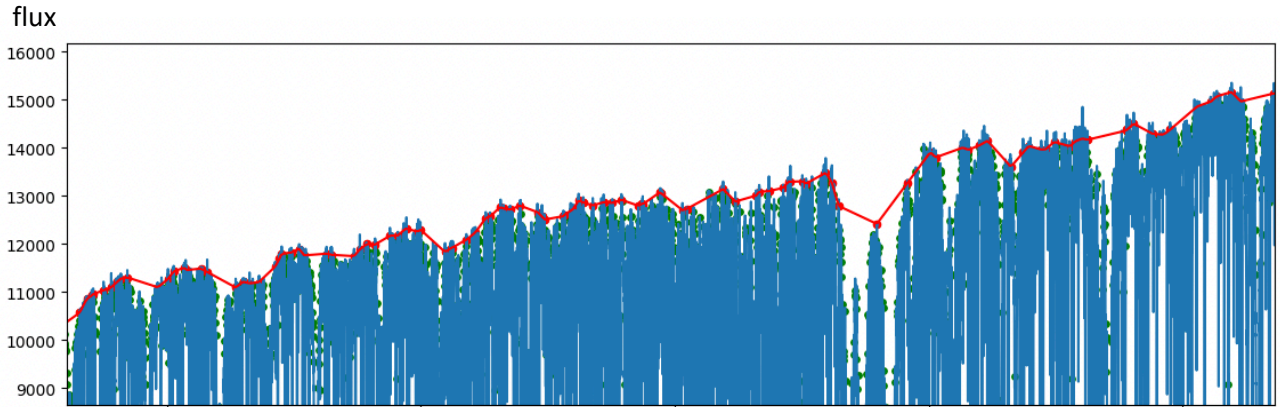

Fig 2 – **radius_min** = 20 Å, **stretching** = 100

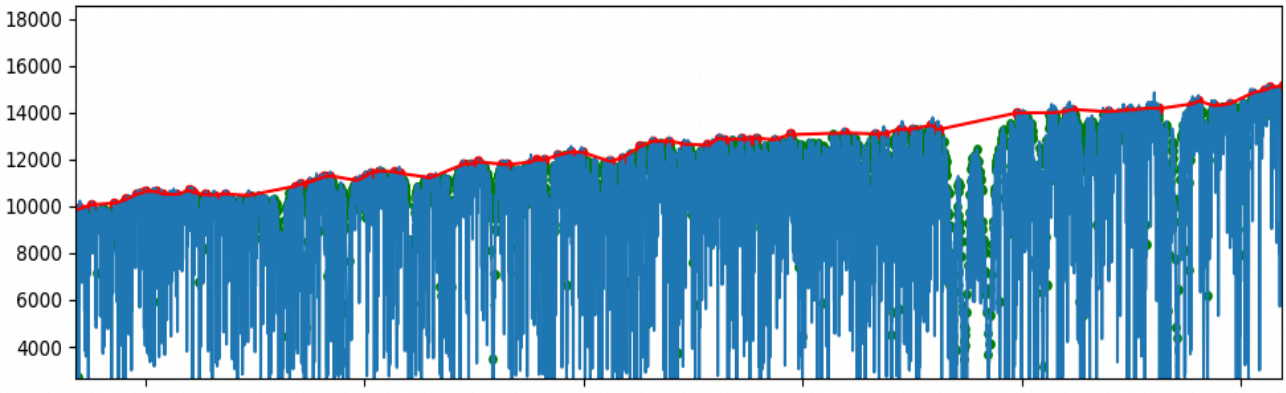

Fig 3 – **radius_min =** 20 Å, **stretching** = 40

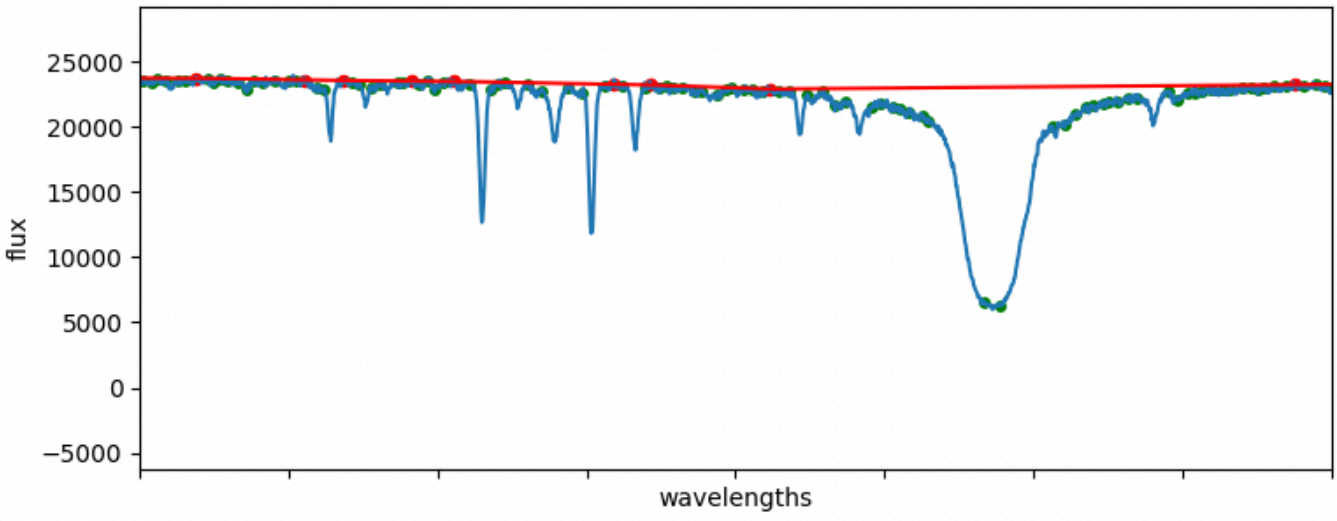

Fig 4 - **radius_min** = 10 Å, **stretching** = 100, **usefilter**=True, **use_RIC** = True

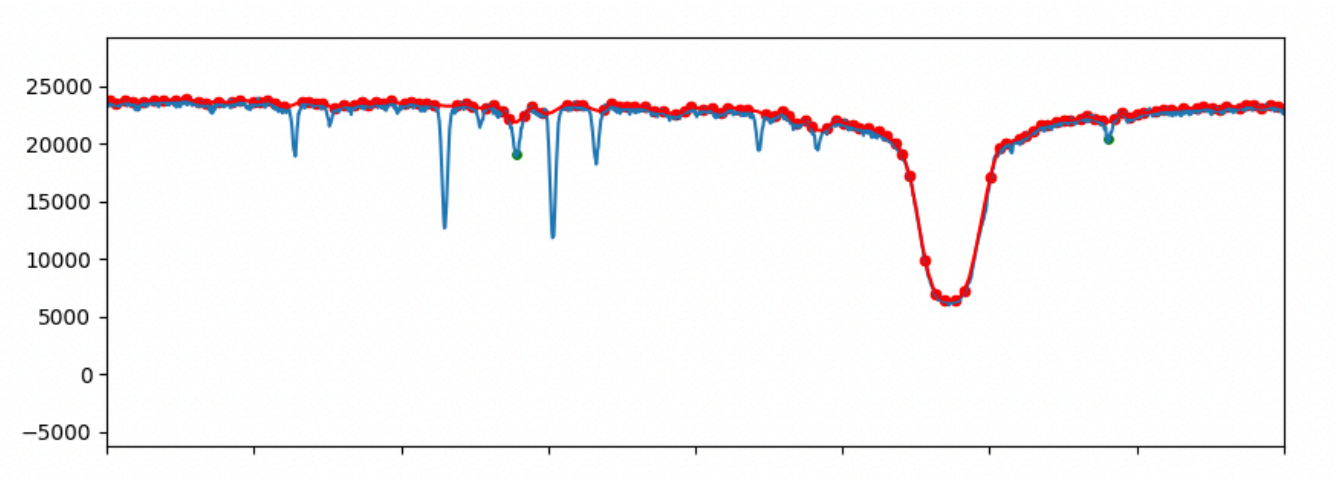

Fig 5 – X ∈ [6550, 6570], **radius_min** = 0.1 Å, **stretching** = 1500, **usefilter**=False, **use_RIC**= False

- **use_RIC–** Boolean value. Enables or disables the use of the radius inflation coefficient (RIC) function to compute the radius, if enabled the rolling pin will increase the radius near gaps in the spectre, and the line will pass above them. The following figure illustrates this, as all parameters are the same, the only change is the use of the RIC function. Note: this is an adaptation of a parameter referred as “penalty map” in M. Cretignier et alt. (2020).

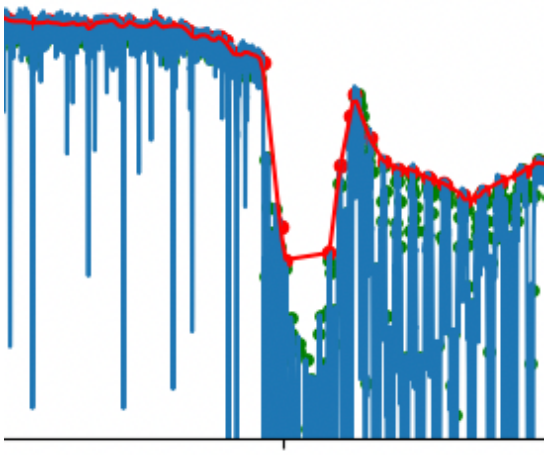

Fig 6.1 – **use_RIC** = False.

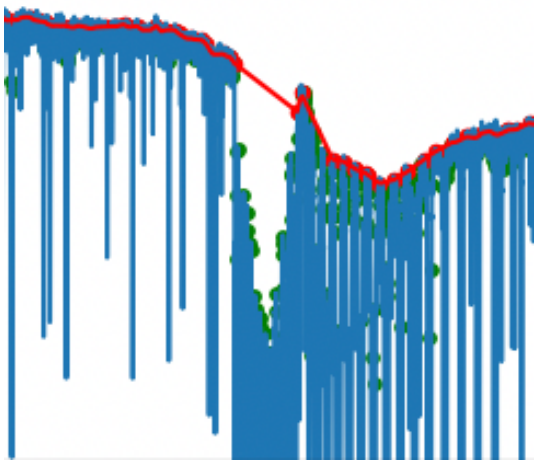

Fig 6.2 - **use_RIC**= True

- **Interp –** Interpolation type, cubic or linear.
- **use_denoise** - Boolean value. If enabled, uses the average of the flux value around the selected maximum to perform the interpolation. Should be enabled for low signal to noise ratio spectra.
- **usefilter** – Boolean value. Enables the use of a Savitzky-Golay filter to smooth the data before running the algorithm. The use of any filter can suppress true variations in the data, but in most cases, it should be enabled.
- **nu –** exponent of the polynomial function used to compute the radius according to the RIC map. If **nu** > 1, the RIC map is convex, and only high values will modify the radius substantially. In contrast, if **nu** < 1, the map is concave, and only low values will leave the radius unchanged. The best value for a given spectrum depends on its shape. Smoothed spectra can be effectively reduced with **nu** < 1, whereas spectra with long oscillations due to instrumental effects or poor merging of 1D extracted echelle-order spectra are better reduced with **nu** > 1. This value should be between 0.7 and 1.3 for most cases (M. Cretignier et alt(2020)). Default is **nu** = 1.
- **niter_peaks_remove** – Number of iterations of function to remove sharp peaks, this function removes the points in the top 99.5 percentile according to the sum of the absolute value of left and right derivatives (in case they have different signs). This is an additional step to smoothen the line before interpolation, this parameter should be low or zero, as the degree of detail or wiggles in the continuum should be controlled primarily through alpha-shape radius (**radius_min**) and **stretching**.
- **denoising_distance –** If **use_denoise** = True, this is the number of points to calculate the average around.

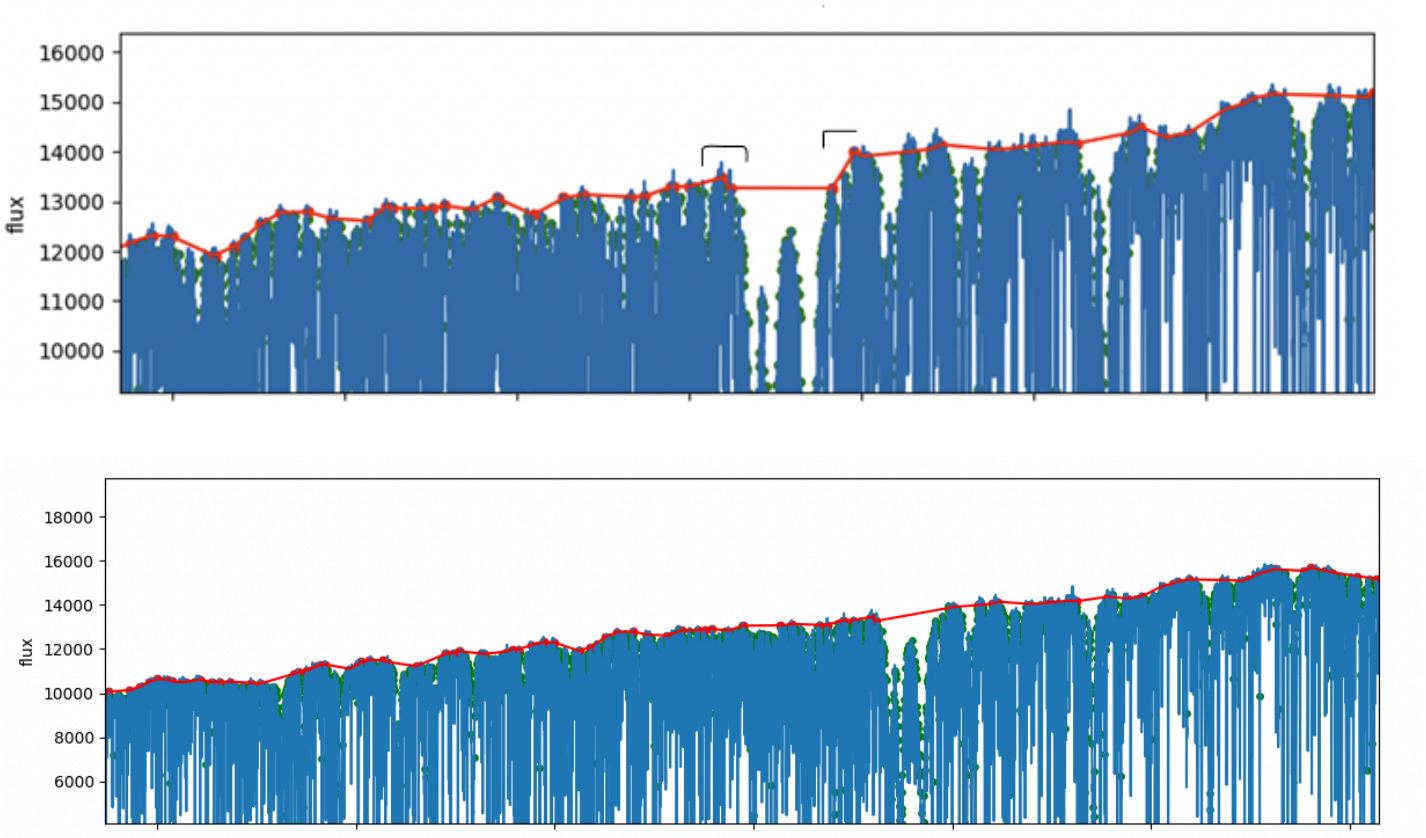

Fig 7 - Above: **niter_peaks_remove** = 0, Below: **niter_peaks_remove** = 10 the black line on the top picture highlights two pairs of close points with a sharp incline which are removed by this function.

References

Michael Cretignier, Jérémie Francfort, Xavier Dumusque, Romain Allart, Francesco Pepe, 2020, RASSINE: Interactive tool for normalising stellar spectra. https://arxiv.org/abs/2006.13098

Xu, X., Cisewski-Kehe, J., Davis, A. B., Fischer, D. A., & Brewer, J. M. 2019, arXiv e-prints, arXiv:1904.10065.