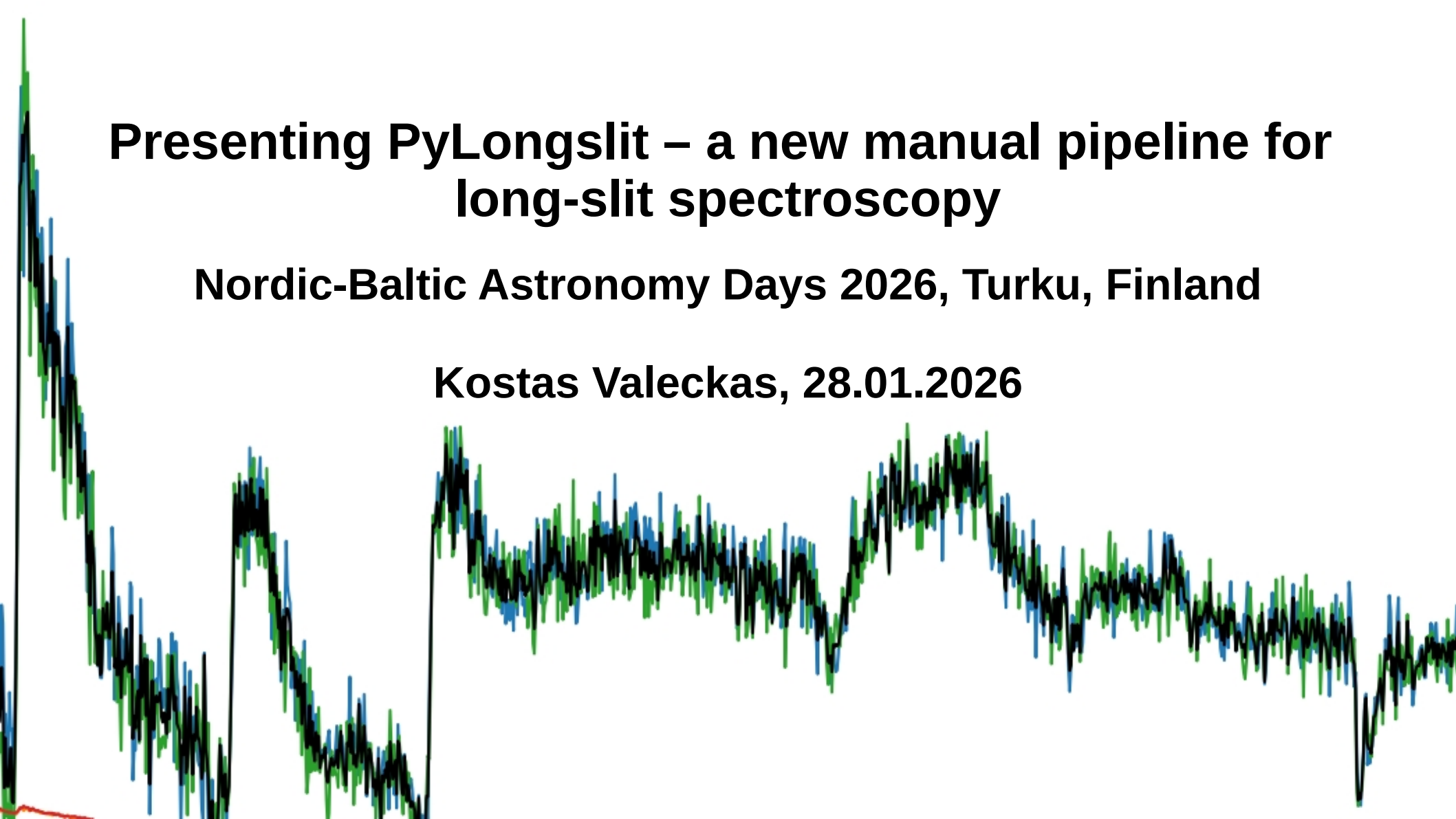
Presenting PyLongslit – a new manual pipeline for long-slit spectroscopy

Nordic-Baltic Astronomy Days 2026, Turku, Finland

Kostas Valeckas, 28.01.2026



Agenda

Agenda

- Motivation
- How it works (briefly)
- Example use case

Motivation

Motivation

… Why (yet another) pipeline?

A counterpart to “black-box” automated solutions

Vision

Vision

- **Robustness.**

Vision

- **Robustness** – should come from **simplicity**.

Vision

- **Robustness** – should come from **simplicity**.
- **Control.**

Vision

- **Robustness** – should come from **simplicity**.
- **Control –** should be **manually executed.**

Vision

- **Robustness** – should come from **simplicity**.
- **Control –** should be **manually executed.**
- **Transparency**.

Vision

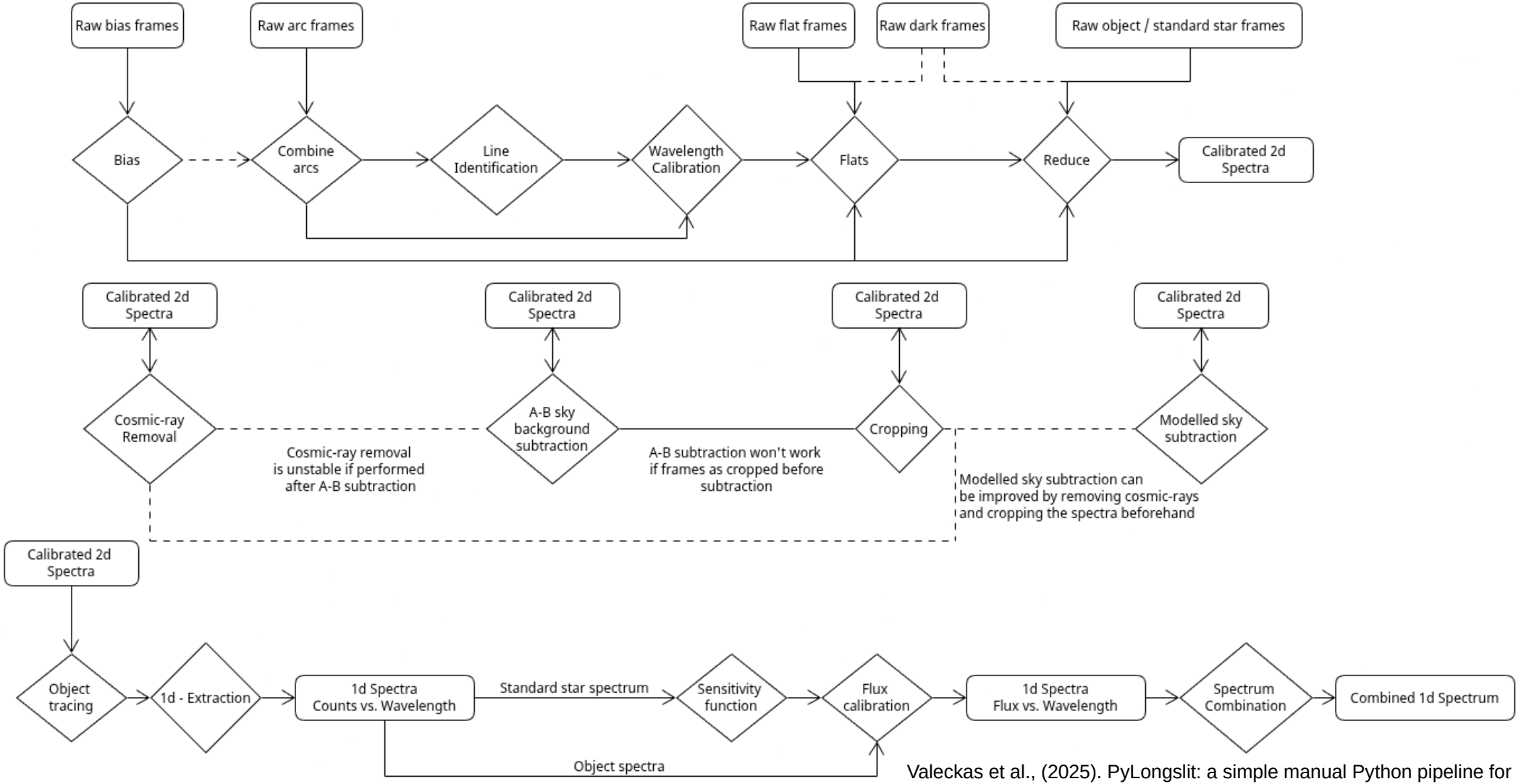
- **Robustness** – should come from **simplicity**.
- **Control –** should be **manually executed.**
- **Transparency –** should come from **documentation**.

Vision

- **Robustness** – should come from **simplicity**.
- **Control –** should be **manually executed.**
- **Transparency –** should come from **documentation**.

LIMITATIONS

How it works (briefly)



Valeckas et al., (2025). PyLongslit: a simple manual Python pipeline for processing of astronomical long-slit spectra recorded with CCD detectors. Journal of Open Source Software, 10(116), 9264,

Works as a command-line tool

Example of a procedure call:

```
pylongslit_bias SDSS_J213510+2728.json
```

Works as a command-line tool

Example of a procedure call:

```
pylongslit_bias SDSS_J213510+2728.json
```

Configuration File

Configuration File

```
"detector": {
    "xsize": 500,
    "ysize": 2102,
    "dispersion": {
        "spectral_dir": "y",
        "wavelength_grows_with_pixel": false
    },
    "gain": 0.16,
    "read_out_noise": 4.3,
    "overscan" : {
        "use_overscan": true,
        "overscan_x_start": 0,
        "overscan_x_end": 499,
        "overscan_y_start": 2064,
        "overscan_y_end": 2102
    }
},

"data": {
    "raw_data_hdu_index": 1
},
```

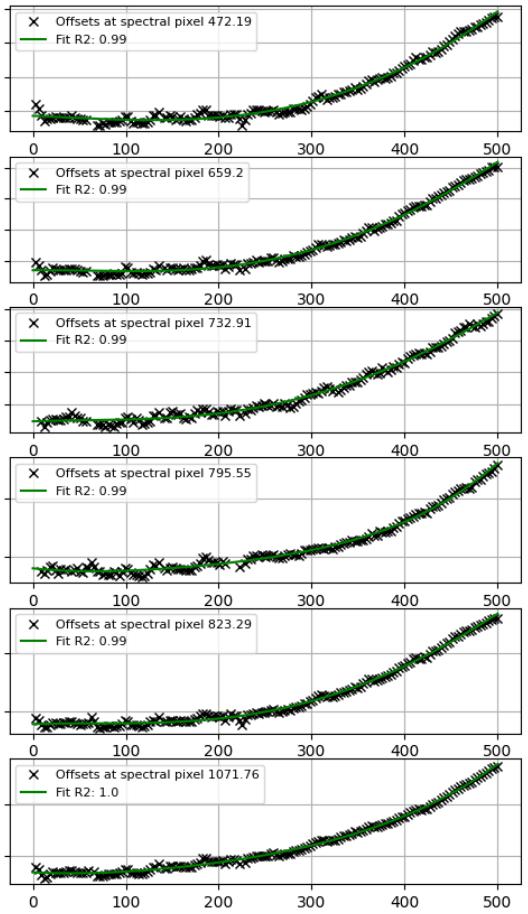
Configuration File

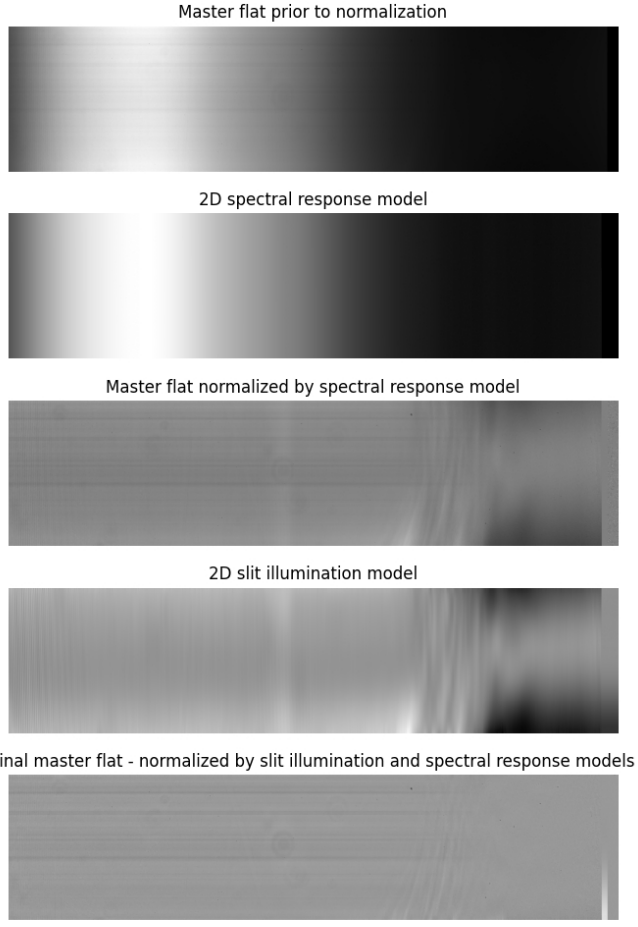
- Pros: extreme flexibility – instrument independent. Configuration files can be saved to reproduce results later.
- Cons: requires a lot of user input (time), and can be subject to wrong input producing faulty results.

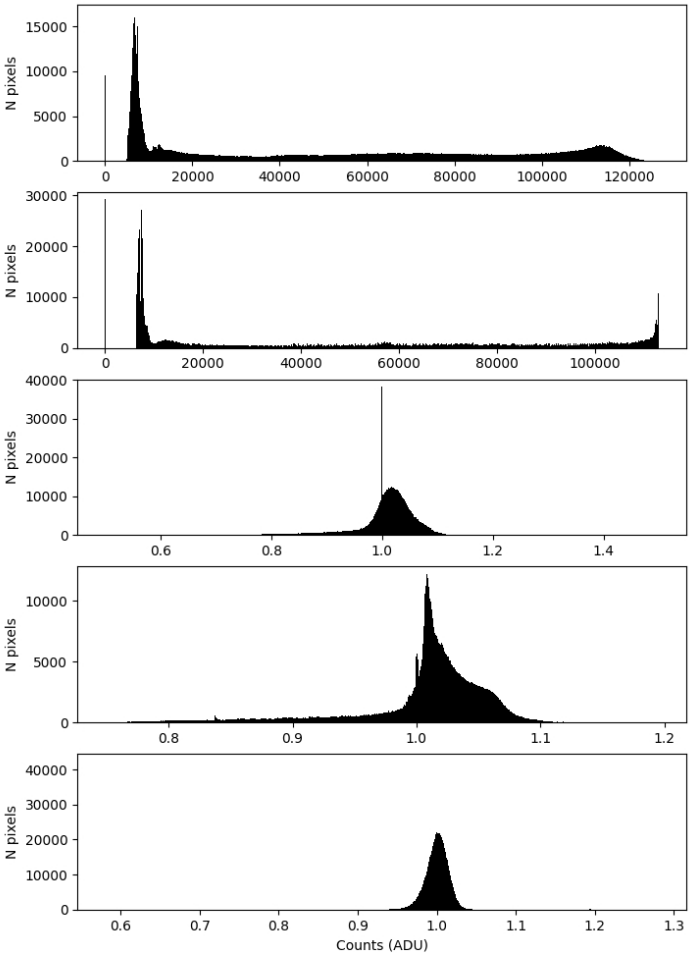
Products

- **A calibrated 2D spectrum in counts/wavelength.**
- **A 1D spectrum extracted from the 2D spectrum, in counts/wavelength (for point-like objects).**
- **A flux-calibrated 1D spectrum in erg/s/cm2/Å (for point-like objects).**

… and numerous QA plots.







Applications

Applications

- **Teaching.**

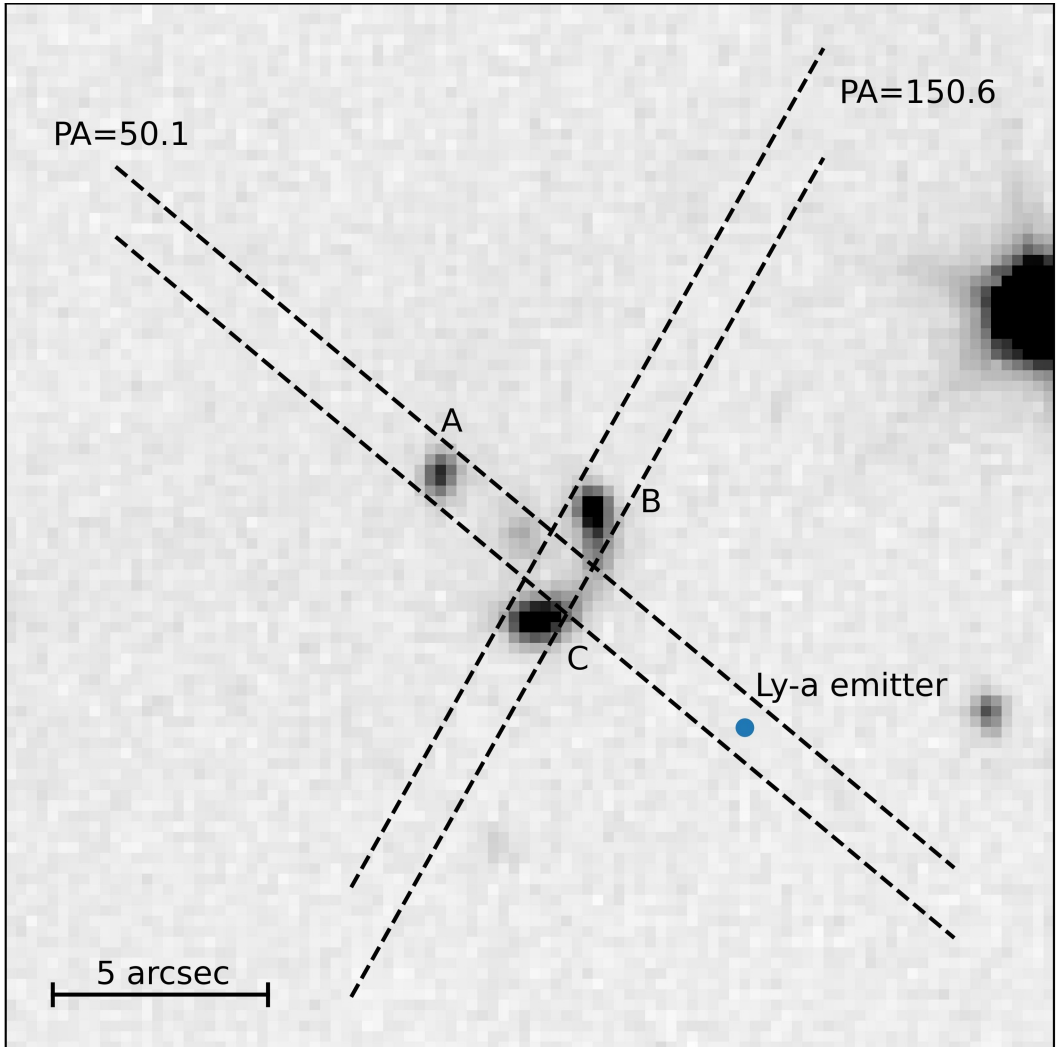
Applications

- **Teaching.**

- Edge-case observations (example to come).

- Expandable framework for an automated and/or instrument-specific pipeline.

Use case – lensed quasars

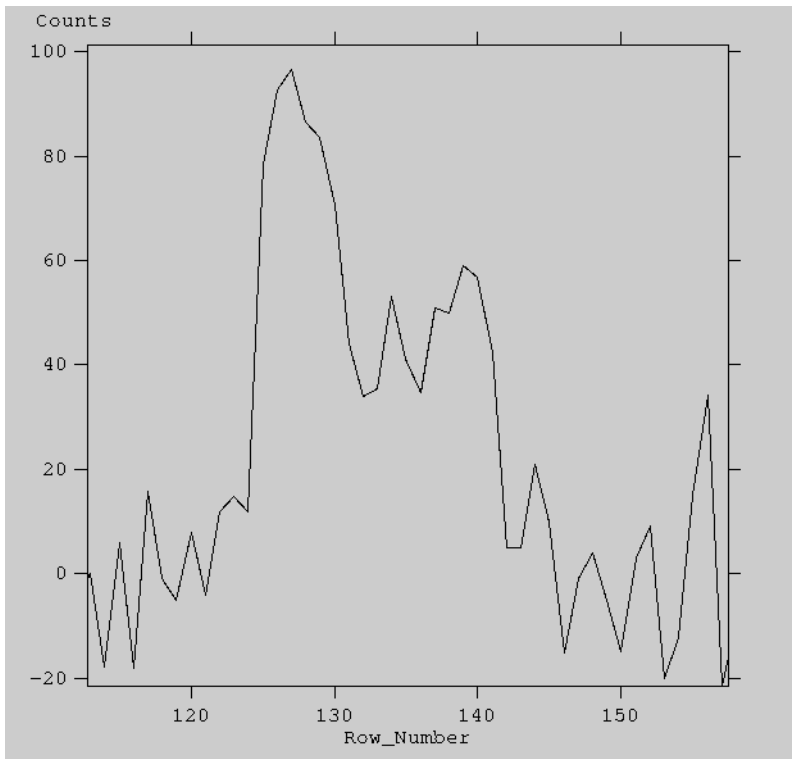


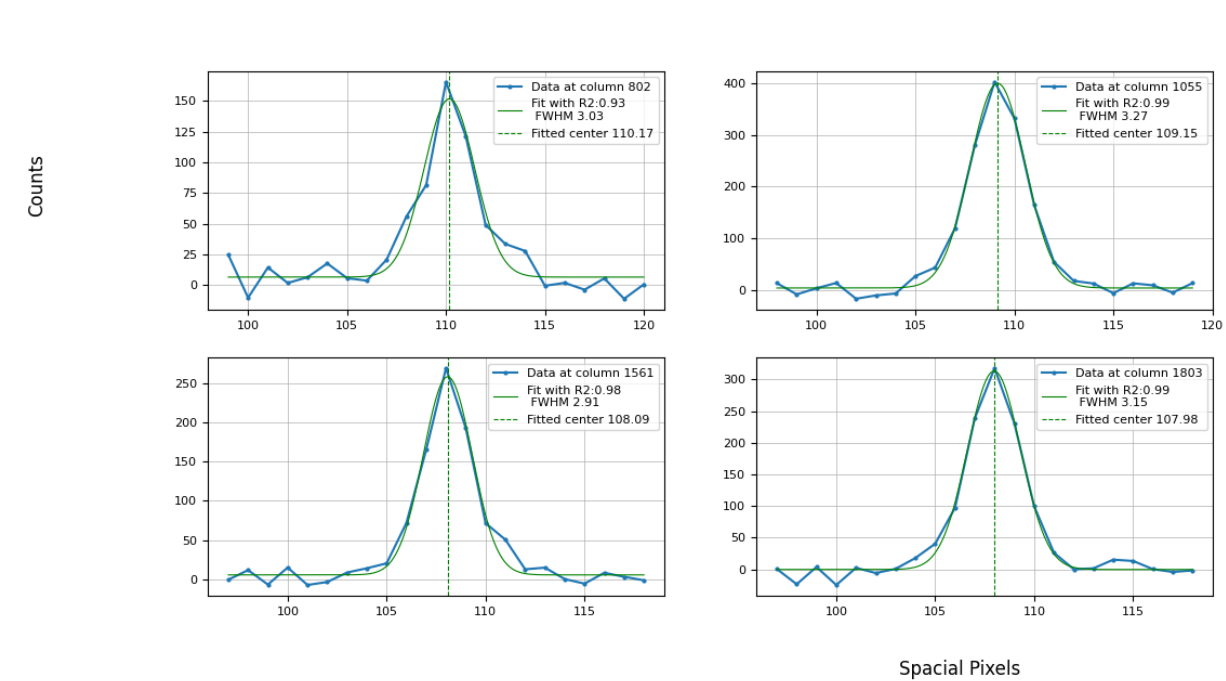
DR3Gaia210752-161131

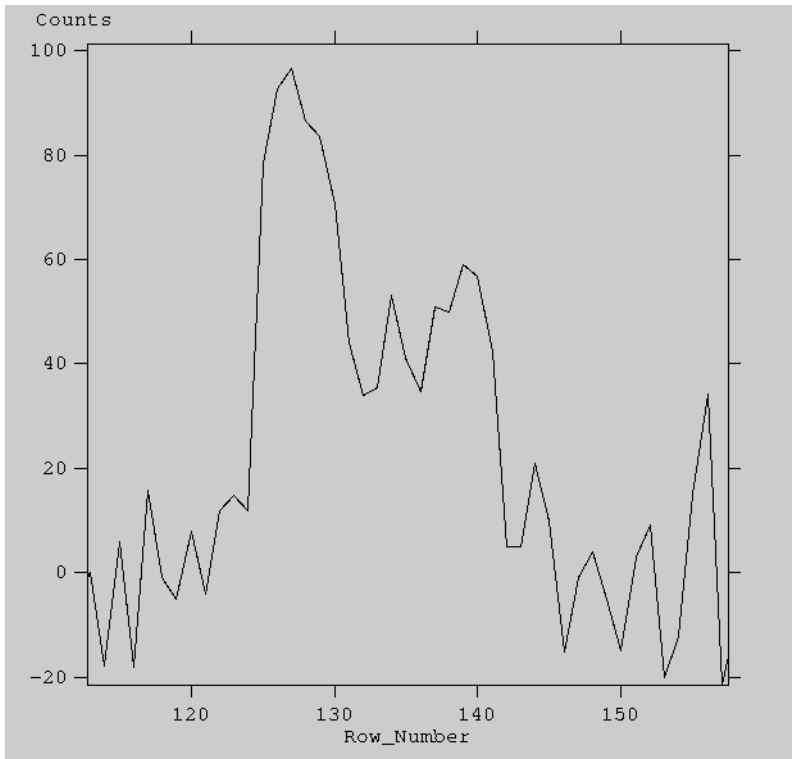
Spectroscopic characterization of a remarkable temporally varying, triple-lensed quasar at z=2.67

(Lind-Thomsen et al., 2026, in prep.)

vertical 1D slice



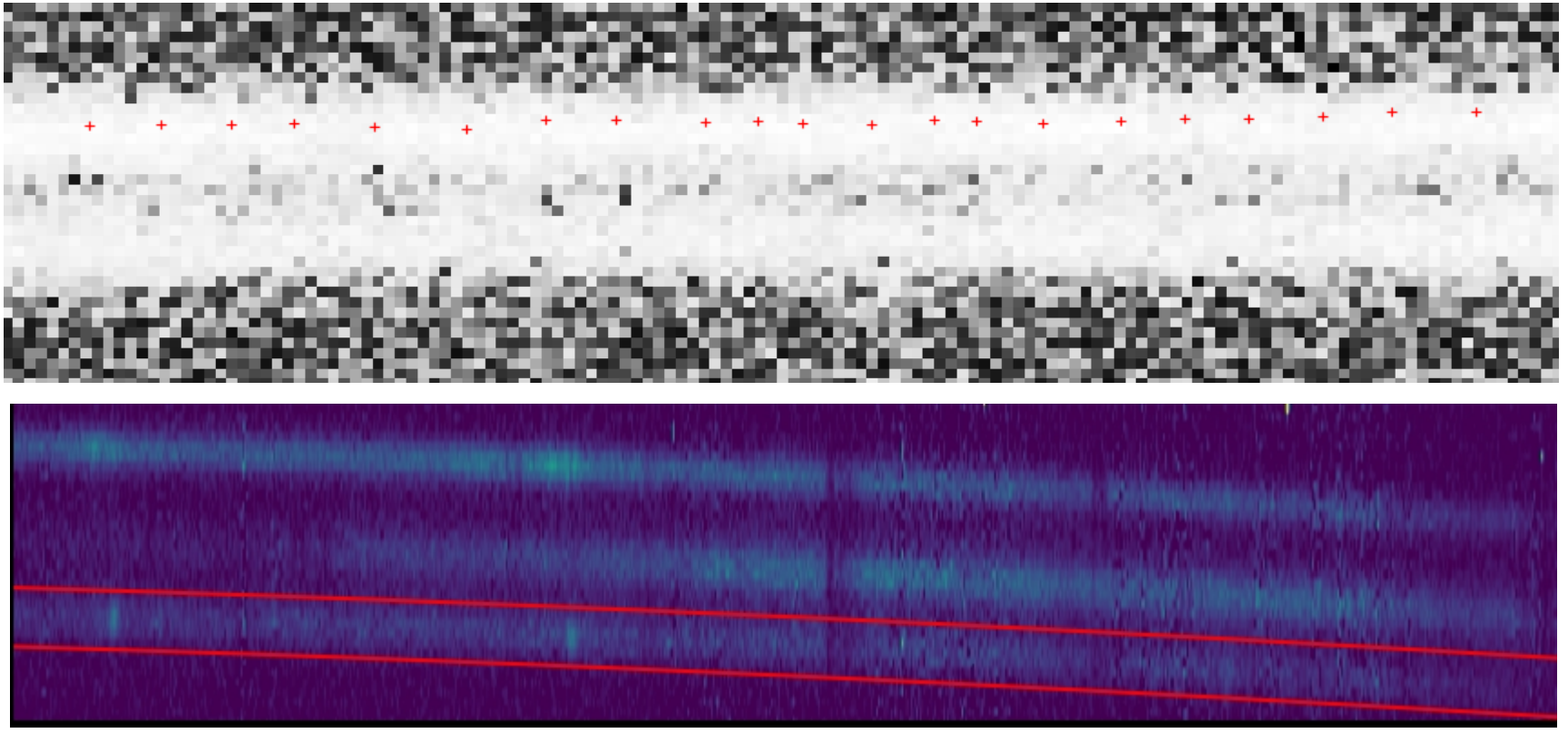

Counts
Data at column 802
Fit with R2:0.93
FWHM 3.03
Fitted center 110.17
Data at column 1055
Fit with R2:0.99
FWHM 3.27
Fitted center 109.15
Data at column 1561
Fit with R2:0.98
FWHM 2.91
Fitted center 108.09
Data at column 1803
Fit with R2:0.99
FWHM 3.15
Fitted center 107.98
Spacial Pixels

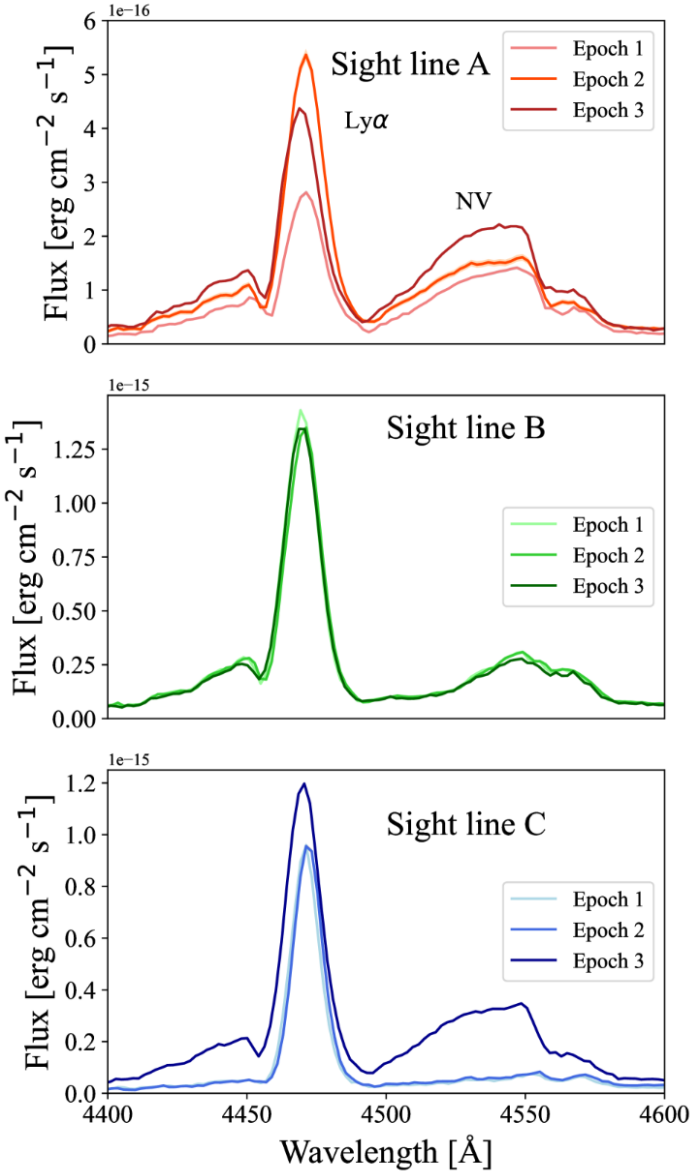

Counts
Row_Number

Vision

- **Robustness** – should come from **simplicity**.
- **Control –** should be **manually executed.**
- **Transparency –** should come from **documentation**.

pylongslit_objtrace_manual





Spectroscopic characterization of a remarkable temporally varying, triple-lensed quasar at z=2.67

(Lind-Thomsen et al., 2026, in prep.)

A few useful references

The Journal of Open Source Software

PyLongslit: a simple manual Python pipeline for processing of astronomical long-slit spectra recorded with CCD detectors

Kostas Valeckas [1,4], **Johan Peter Uldall Fynbo** [2], **Jens-Kristian Krogager** [3], **and Kasper Elm Heintz** [2]

1 Niels Bohr Institute, Copenhagen University, Denmark **2** Cosmic Dawn Center, Niels Bohr Institute, Copenhagen University, Denmark **3** Centre de Recherche Astrophysique de Lyon, France **4** Nordic Optical Telescope, Spain

Valeckas et al., (2025). PyLongslit: a simple manual Python pipeline for processing of astronomical long-slit spectra recorded with CCD detectors. Journal of Open Source Software, 10(116), 9264, https://doi.org/10.21105/joss.09264

PyLongslit docs

https://kostasvaleckas.github.io/PyLongslit/index.html

PyLongslit

Search Go

Navigation

Contents:

PyLongslit's documentation page

JOSS 10.21105/joss.09264 DOI 10.5281/zenodo.15091603 arXiv astro-ph

PyLongslit is a simple manual Python pipeline for processing of astronomical long-slit spectra recorded with CCD detectors. At the current version, the software can produce the following products:

1. A calibrated 2D spectrum in counts/wavelength.
2. A 1D spectrum extracted from the 2D spectrum in counts/wavelength (for point-like objects).
3. A flux-calibrated 1D spectrum in $\mathrm{erg/s/cm^2/Å}$ (for point-like objects).

The software is designed to be instrument-independent, and is **designed to work with data that has the following characteristics**:

1. The data is taken by a long-slit spectrograph. I.e. the data is 2D, with spatial information along one axis and spectral information along the other.
2. The data is recorded with a single CCD detector (or a mosaic of CCD detectors that are treated as one).
3. The data is in FITS format.

Math from the engine room

https://github.com/KostasValeckas/PyLongslit_dev/blob/main/documents/Note_on__PyLongslit_uncertainties.pdf

$$S_{points}(\lambda)\left[\frac{erg/cm^2/\AA}{counts}\right] = \frac{\text{1d-spec flux-calibrated spectrum}\left[erg/s/cm^2/\AA\right]}{\text{1d-spec observed spectrum i counts pr. second}\left[counts/s\right]} \quad . \tag{19}$$

Fitting a model to these points gives a conversion factor $S(\lambda)$ between observed counts pr. second $C_{1d}(\lambda)/s$ to flux in physical units $Flux(\lambda)$:

$$Flux(\lambda) = \frac{C_{1d}(\lambda)}{\text{exposure time}}\, S(\lambda) \quad . \tag{20}$$

In the software, the fit for $S(\lambda)$ is performed in (10-base) log-space ($S_{log}(\lambda)$). This is because the observed 1d-standard star spectrum in counts will still have some artifacts such as absorption lines from the sky, and these might corrupt the fit. Fitting in logarithmic space scales these artifacts down. The error of the fit in logspace is calculated using eq. (1):

$$\sigma_{S_{log}} = \text{RMS}(\log\left(S_{points}(\tilde{\lambda})\right) - S_{log}(\tilde{\lambda})) \quad , \tag{21}$$

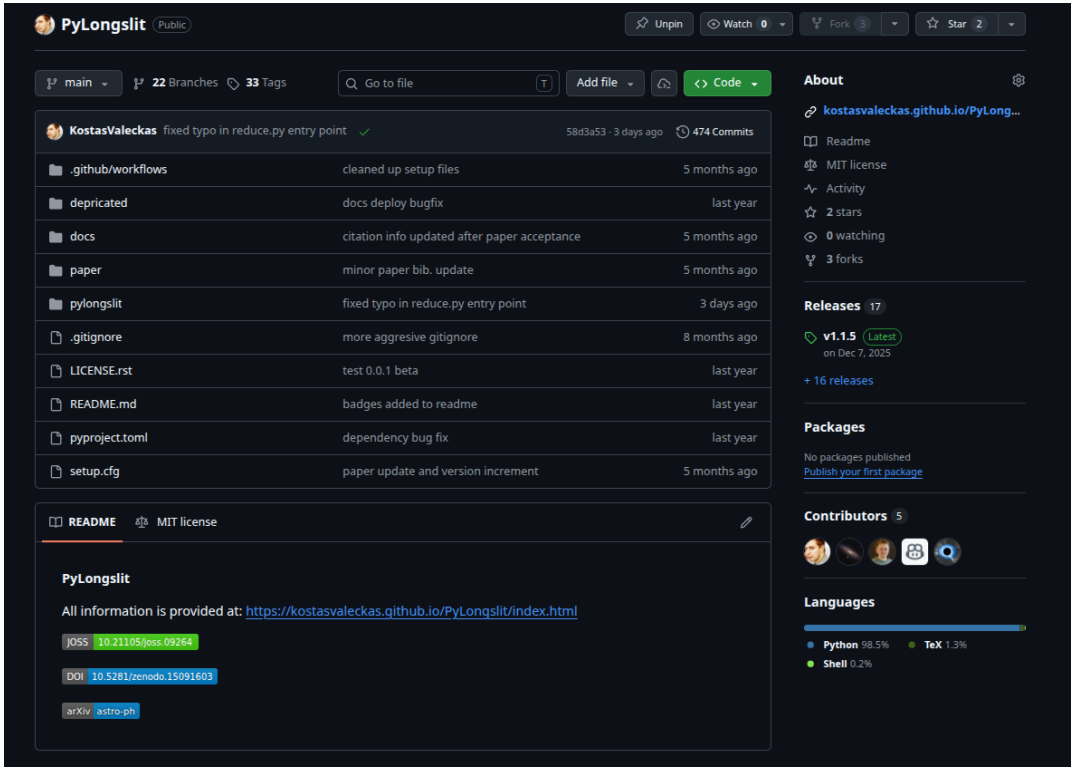
where $\tilde{\lambda}$ are the wavelengths at which the sensitivity points were evaluated.

To convert from $S_{log}(\lambda)$ to $S(\lambda)$, the following expression is used:

$$S(\lambda) = 10^{S_{log}(\lambda)} \quad , \tag{22}$$

Code repository

https://github.com/KostasValeckas/PyLongslit



This is an open source project, and contributions are (very) welcome!

kostas.valeckas@nbi.ku.dk

Thank you.