



**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

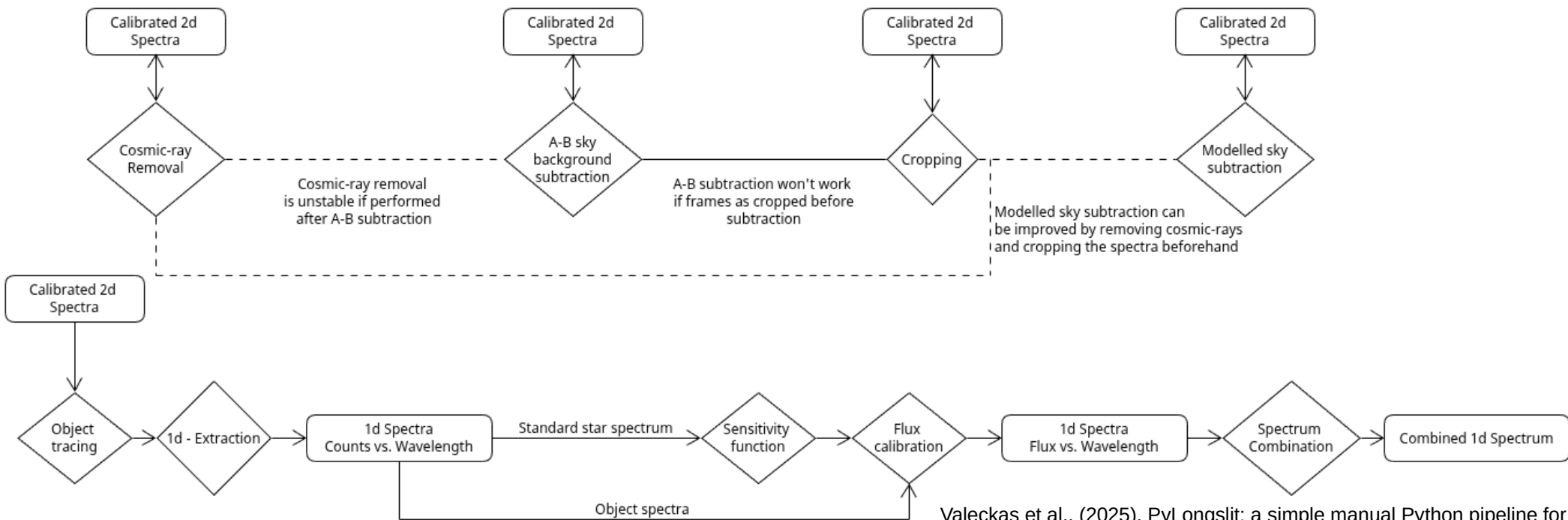
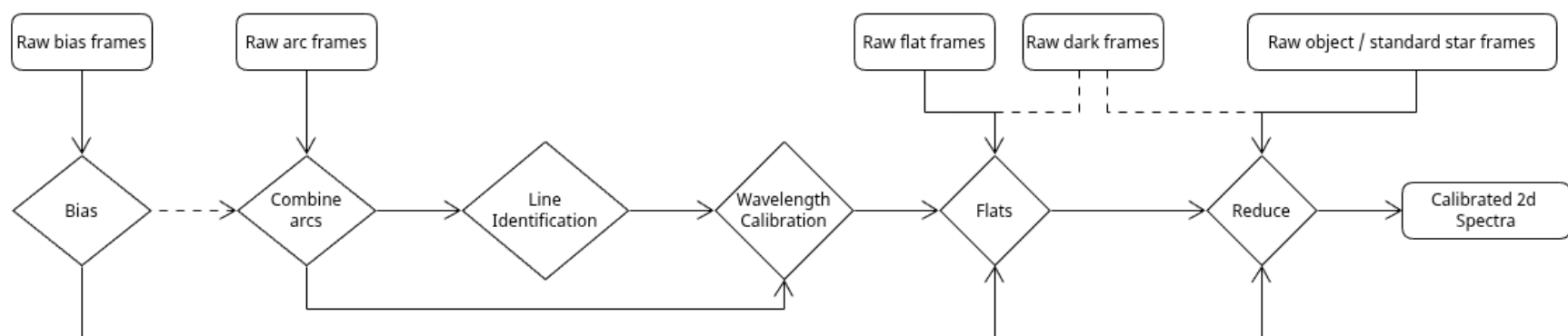
- **Robustness** – should come from **simplicity**.
- **Control** – should be **manually executed**.
- **Transparency** – should come from **documentation**.

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  
},
```

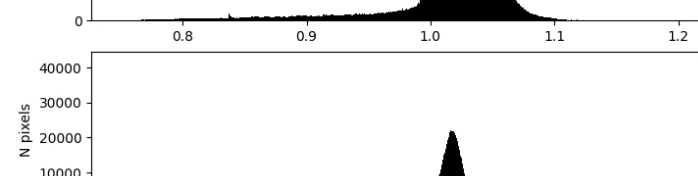
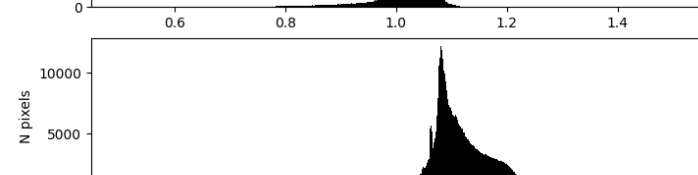
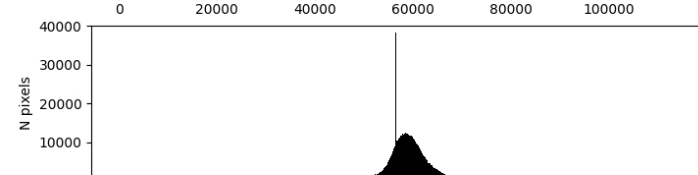
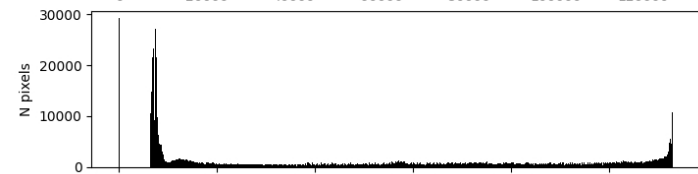
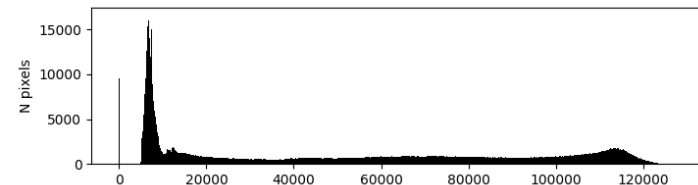
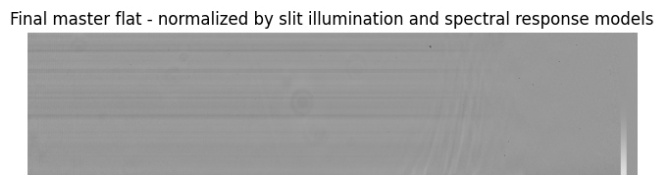
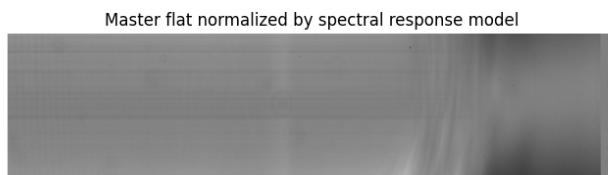
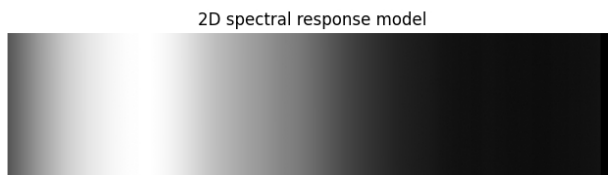
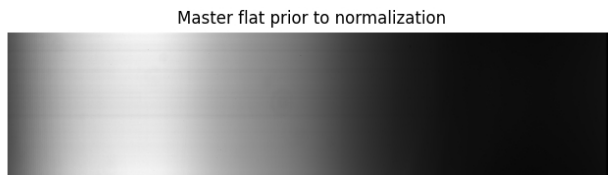
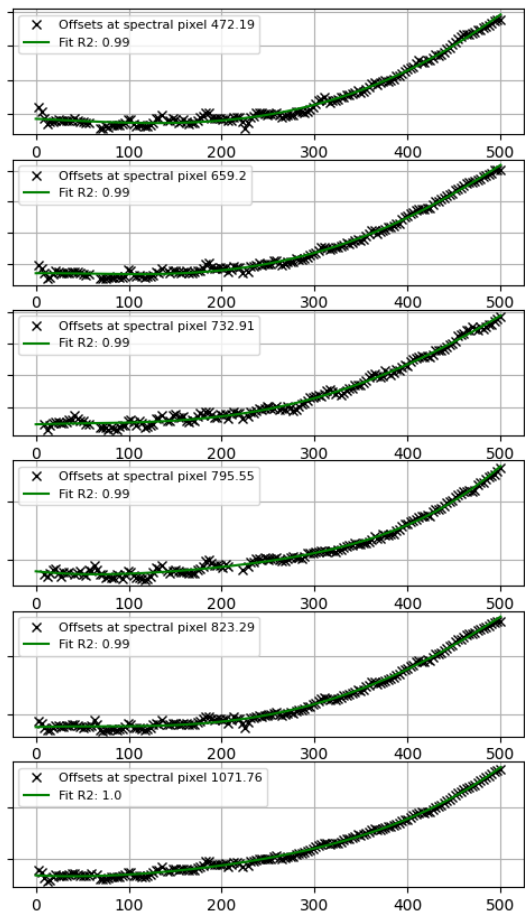
Configuration File

- 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 $\text{erg/s/cm}^2/\text{\AA}$ (for point-like objects).**

... and numerous QA plots.



Counts (ADU)

Applications

Applications

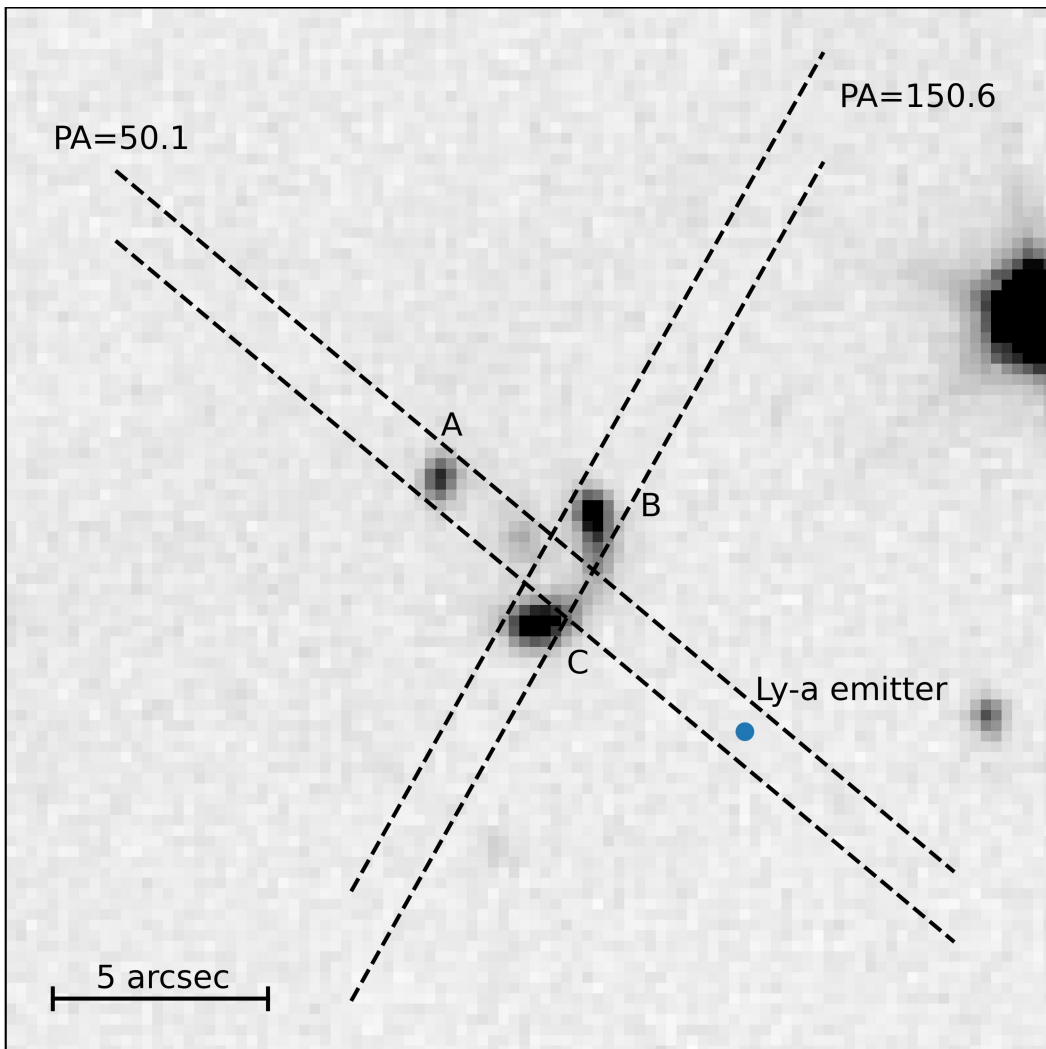
- Teaching.

Applications

- Teaching.
- Edge-case observations (example to come).
- Expandable framework for an automated and/or instrument-specific pipeline.

Use case – lensed quasars

DR3Gaia210752-161131



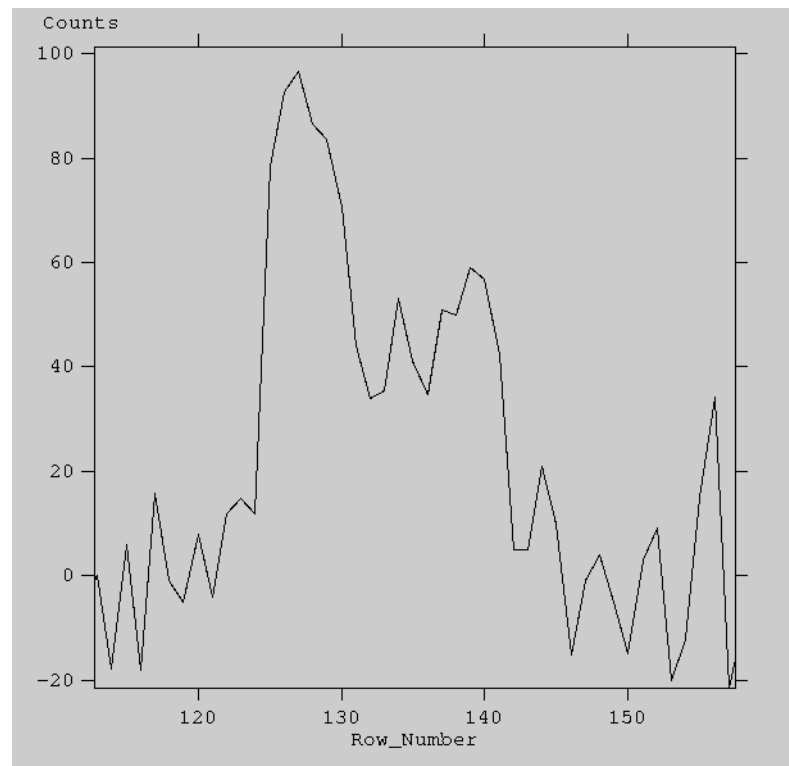
Spectroscopic characterization of a remarkable temporally varying, triple-lensed quasar at $z=2.67$

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



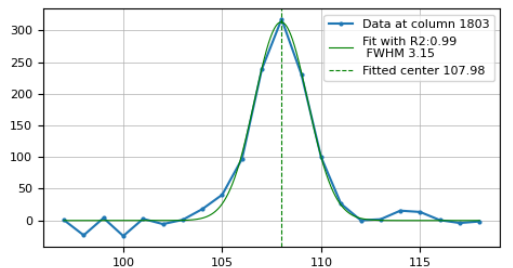
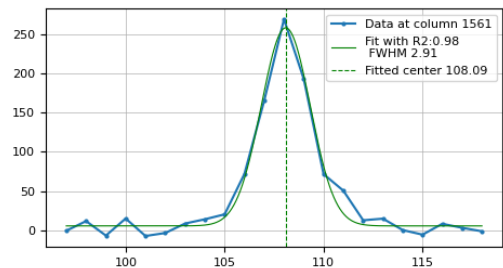
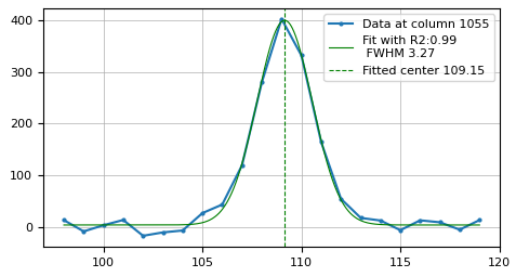
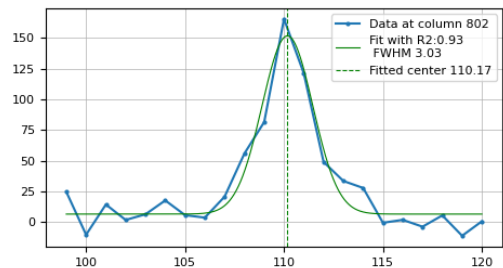


vertical 1D
slice

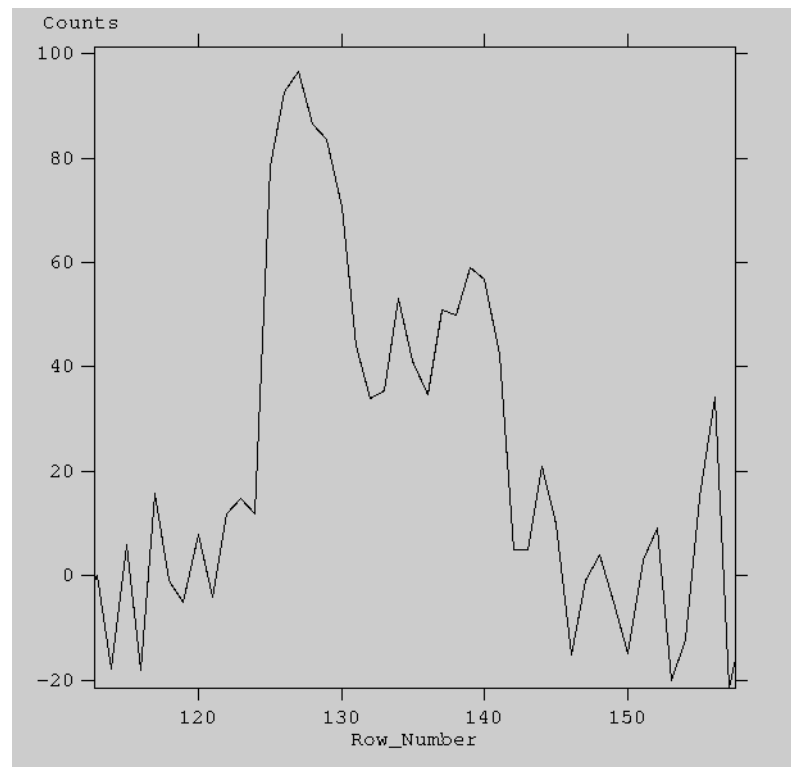




Counts



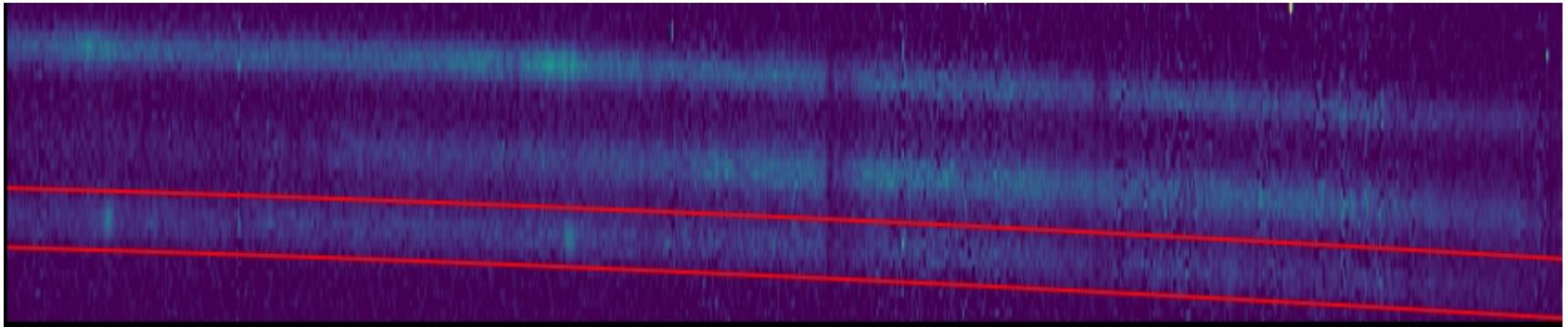
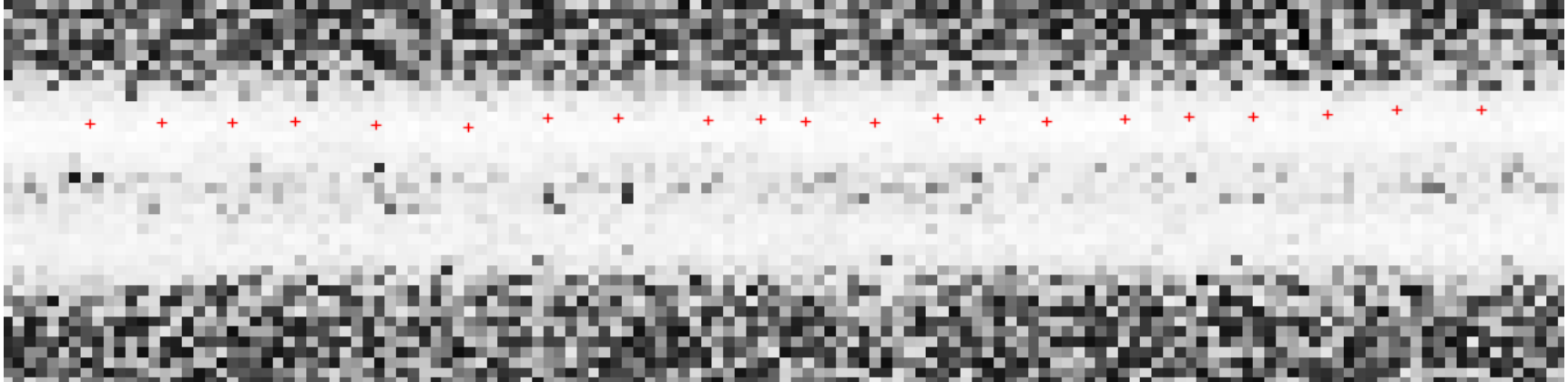
Spatial Pixels

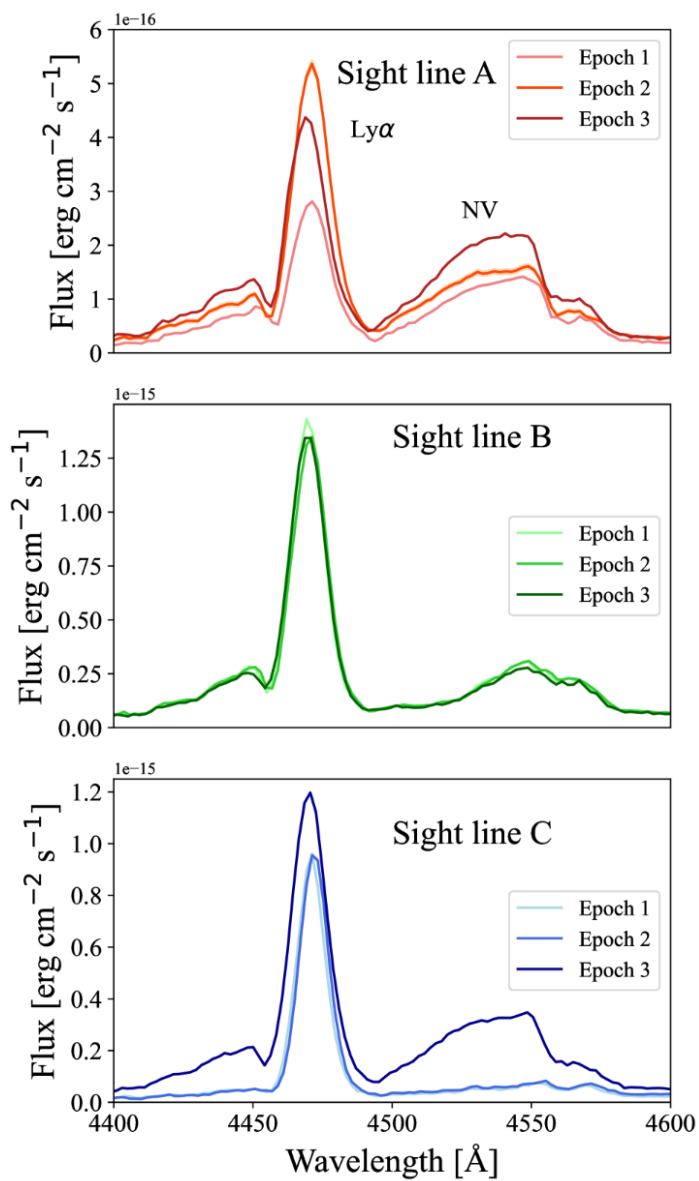


Vision

- **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



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

Navigation

Contents:

[Installation](#)

[Getting started](#)

[Overview of all pipeline procedures and their products](#)

[Tested Instruments and](#)

[Configurations](#)

[Special Use Cases](#)

[Interested in developing?](#)

[Note on how the software estimates uncertainties](#)

[Contact](#)

PyLongslit's documentation page

JOSS [10.21105/joss.09264](https://doi.org/10.21105/joss.09264) DOI [10.5281/zenodo.15091603](https://doi.org/10.5281/zenodo.15091603) arXiv [astro-ph](https://arxiv.org/abs/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 $\text{erg/s/cm}^2/\text{\AA}$ (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

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

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 $Flux(\lambda)$:

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

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):

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

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:

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

Code repository

<https://github.com/KostasValeckas/PyLongslit>

The screenshot shows the GitHub repository page for PyLongslit. The repository is public and has 22 branches, 33 tags, 474 commits, 2 stars, and 3 forks. The main branch is selected. The repository contains several files and folders, including .github/workflows, deprecated, docs, paper, pylongslit, .gitignore, LICENSE.rst, README.md, pyproject.toml, and setup.cfg. The README.md file is open, showing the repository name, a link to the index.html, and various identifiers like JOSS, DOI, and arXiv. The right sidebar shows the About section with a link to the repository, the MIT license, and 2 stars. The Releases section shows 17 releases, with the latest being v1.1.5 on Dec 7, 2025. The Packages section shows no packages published. The Contributors section shows 5 contributors. The Languages section shows Python (98.5%), TeX (1.3%), and Shell (0.2%).

PyLongslit Public

Unpin Watch 0 Fork 3 Star 2

main 22 Branches 33 Tags Go to file Add file Code

KostasValeckas fixed typo in reduce.py entry point ✓ 58d3a53 · 3 days ago 474 Commits

File/Folder	Description	Last Update
.github/workflows	cleaned up setup files	5 months ago
deprecated	docs deploy bugfix	last year
docs	citation info updated after paper acceptance	5 months ago
paper	minor paper bib. update	5 months ago
pylongslit	fixed typo in reduce.py entry point	3 days ago
.gitignore	more aggressive gitignore	8 months ago
LICENSE.rst	test 0.0.1 beta	last year
README.md	badges added to readme	last year
pyproject.toml	dependency bug fix	last year
setup.cfg	paper update and version increment	5 months ago

README MIT license

PyLongslit

All information is provided at: <https://kostasvaleckas.github.io/PyLongslit/index.html>

JOSS [10.21105/joss.09264](https://doi.org/10.21105/joss.09264)

DOI [10.5281/zenodo.15091603](https://doi.org/10.5281/zenodo.15091603)

arXiv [astro-ph](https://arxiv.org/abs/astro-ph)

About kostasvaleckas.github.io/PyLong...

Readme MIT license Activity 2 stars 0 watching 3 forks

Releases 17

v1.1.5 (Latest) on Dec 7, 2025

+ 16 releases

Packages

No packages published [Publish your first package](#)

Contributors 5

Languages

- Python 98.5%
- TeX 1.3%
- Shell 0.2%

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

`kostas.valeckas@nbi.ku.dk`

Thank you.