SPRAT_LHS6328_Hilt102

September 24, 2019

[1]:
```
import sys
import numpy as np
from astropy.io import fits
sys.path.append('aspired')
import aspired
import plotly.io as pio
pio.renderers.default = 'notebook+jpg'
```

```
WARNING: AstropyDeprecationWarning: astropy.extern.six will be removed in 4.0,
use the six module directly if it is still needed [astropy.extern.six]
```

[2]:
```
science_frame = aspired.ImageReduction('examples/lhs6328.list')
science_frame.reduce()
#science_frame.inspect()
#science_frame.savefits(overwrite=True)

standard_frame = aspired.ImageReduction('examples/hiltner102.list')
standard_frame.reduce()
#standard_frame.inspect()
```

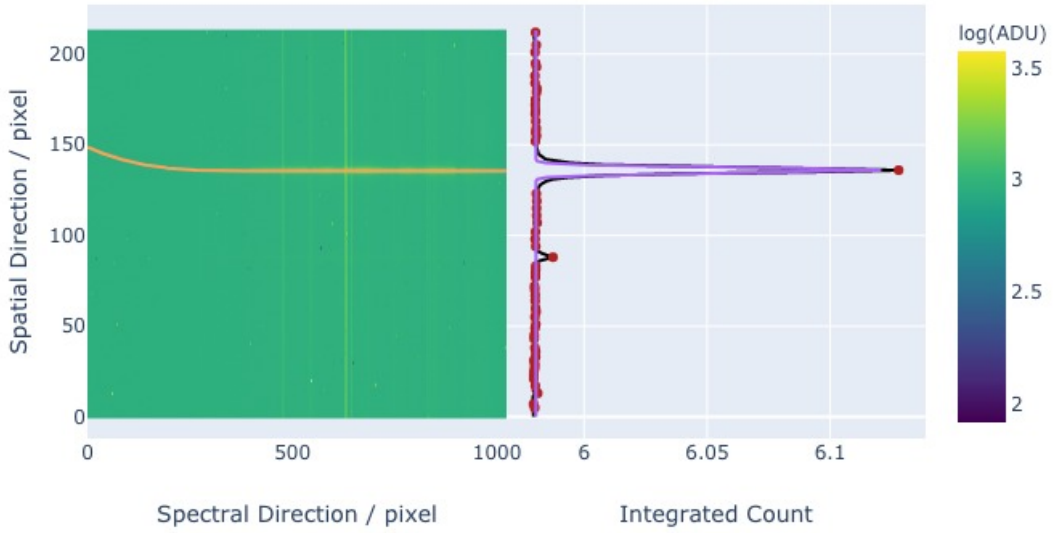
```
aspired/aspired.py:271: UserWarning:

No bias frames. Bias subtraction is not performed.

aspired/aspired.py:281: UserWarning:

No flat frames. Field-flattening is not performed.
```

[3]:
```
# Example data from SPRAT
# LHS6328 companion WD
science_data = science_frame.fits_data.data

# Example data from SPRAT
# Hiltner102
standard_data = standard_frame.fits_data.data
```

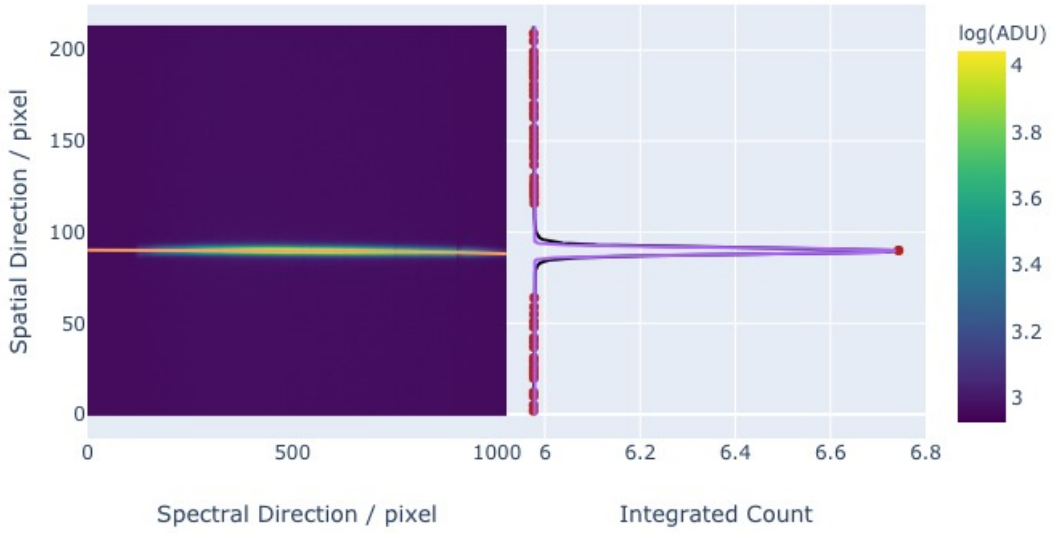
[4]:
```
# Set the spectral and spatial direction
Saxis = 1
Waxis = 0

# spec mask
spec_mask = np.arange(science_frame.fits_data.header['NAXIS1'])
spatial_mask = np.arange(40,science_frame.fits_data.header['NAXIS2'])

# initialise the two aspired.TwoDSpec()
lhs6328 = aspired.TwoDSpec(science_data, spatial_mask=spatial_mask,␣
↪spec_mask=spec_mask, rn=10., cr=False, gain=1., seeing=1.2, display=True)
hilt102 = aspired.TwoDSpec(standard_data, spatial_mask=spatial_mask,␣
↪spec_mask=spec_mask, rn=10., cr=False, gain=1., seeing=1.2, display=True)
```

[5]:
```
# automatically trace the spectrum
lhs6328.ap_trace(nsteps=20, recenter=False, prevtrace=(0, ), fittype='spline',␣
↪bigbox=8)
hilt102.ap_trace(nsteps=20, recenter=False, prevtrace=(0, ), fittype='spline',␣
↪bigbox=8)
```

```
Tracing Aperture using nsteps=20
Step 3 of 20 of spectrum 1 of 1 has a poor fit. Initial guess is used instead.
Spectrum 1 : Trace gaussian width = [0.62386191 0.26132169 0.         1.34726453
1.44325993 1.55227841
 1.48498931 1.45153484 1.44985531 1.44101176 1.41174052 1.39552287
 1.38963013 1.39080515 1.38125338 1.3753703  1.38674186 1.32418218
 1.23241782 0.        ] pixels

aspired/aspired.py:692: RuntimeWarning:

invalid value encountered in log10

/usr/local/lib/python3.7/site-packages/scipy/optimize/minpack.py:795:
OptimizeWarning:

Covariance of the parameters could not be estimated
```

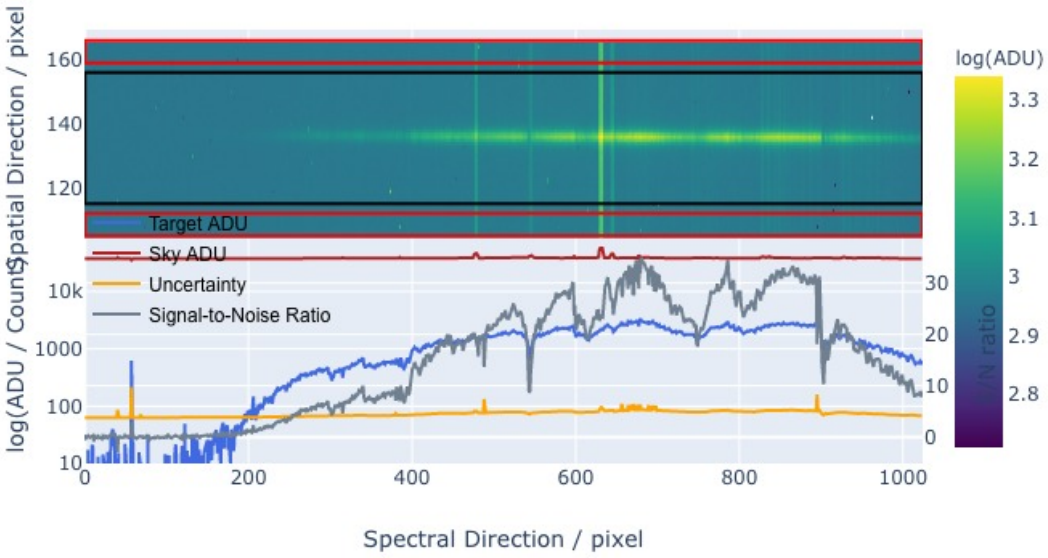


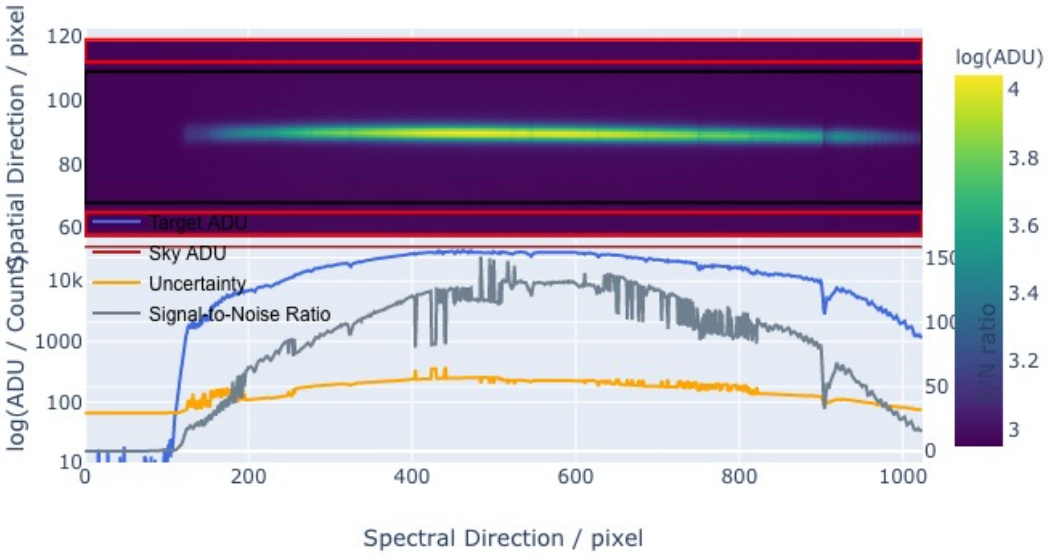

```
Tracing Aperture using nsteps=20
Spectrum 1 : Trace gaussian width = [0.01204678 2.69951126 1.90635725 1.49453735
1.3013829  1.22968109
 1.20013513 1.18278048 1.17835079 1.17293229 1.1807976  1.18837918
 1.2009496  1.21668759 1.237217   1.26093305 1.28901669 1.36933244
 1.37095872 0.        ] pixels
```



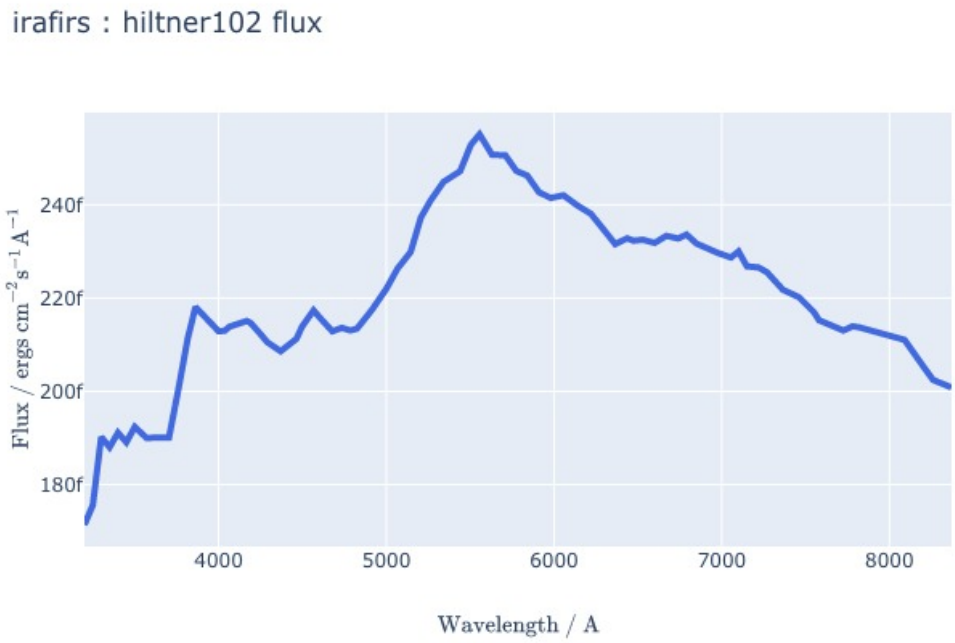
[6]:
```
# Optimal extracting spectrum by summing over the aperture along the trace
lhs6328.ap_extract(apwidth=20)
hilt102.ap_extract(apwidth=20)
```

```
aspired/aspired.py:1085: RuntimeWarning:

invalid value encountered in log10

aspired/aspired.py:1217: RuntimeWarning:

invalid value encountered in log10
```


Spatial Direction / pixel
160
140
120
Target ADU
Sky ADU
Uncertainty
Signal-to-Noise Ratio
log(ADU / Counts)
10k
1000
100
10
0
200
400
600
800
1000
30
20
10
0
log(ADU)
3.3
3.2
3.1
3
2.9
2.8
S/N ratio
Spectral Direction / pixel


Spatial Direction / pixel
120
100
80
60
Target ADU
Sky ADU
Uncertainty
Signal-to-Noise Ratio
log(ADU / Counts)
10k
1000
100
10
0
200
400
600
800
1000
150
100
50
0
log(ADU)
4
3.8
3.6
3.4
3.2
3
S/N ratio
Spectral Direction / pixel

[7]:
```
fluxcal = aspired.StandardFlux(target='hiltner102', group='irafirs', cutoff=0.
↪4, ftype='flux')
fluxcal.load_standard()
fluxcal.inspect_standard()
```

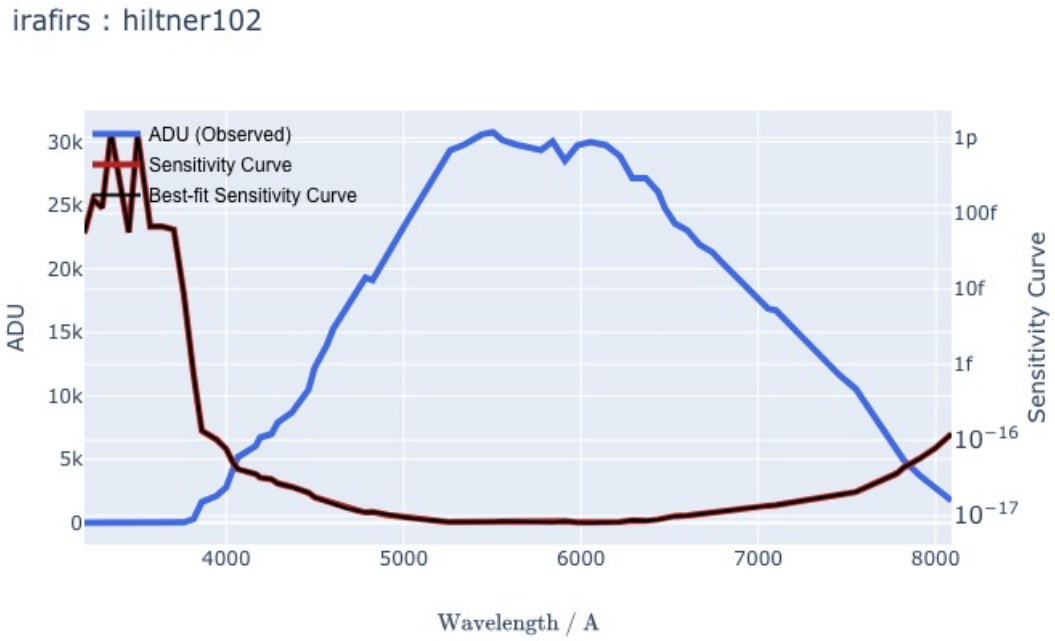


[8]:
```
# Placeholder of wavelength calibration
wavecal = type('', (), {})()
wavecal.pfit_type = 'poly'
wavecal.pfit = [3.17707768e-07, -6.15874242e-04,  5.13807283e+00,  3.
↪22724313e+03]
```

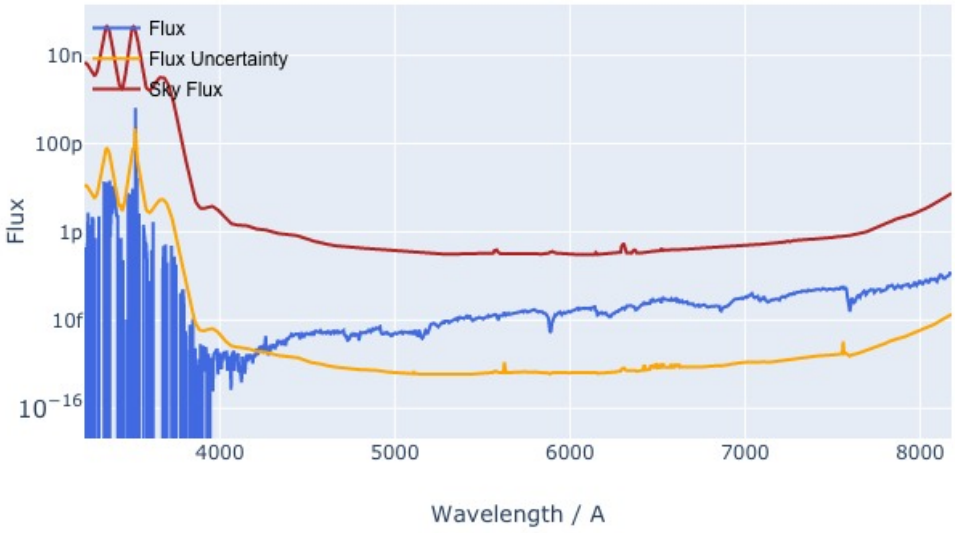
[9]:
```
# Get the sensitivity curves
lhs6328_reduced = aspired.OneDSpec(lhs6328, wavecal, standard=hilt102,␣
↪wave_cal_std=wavecal, flux_cal=fluxcal)
lhs6328_reduced.apply_wavelength_calibration('all')
lhs6328_reduced.compute_sencurve(kind='cubic')
lhs6328_reduced.inspect_sencurve()
```

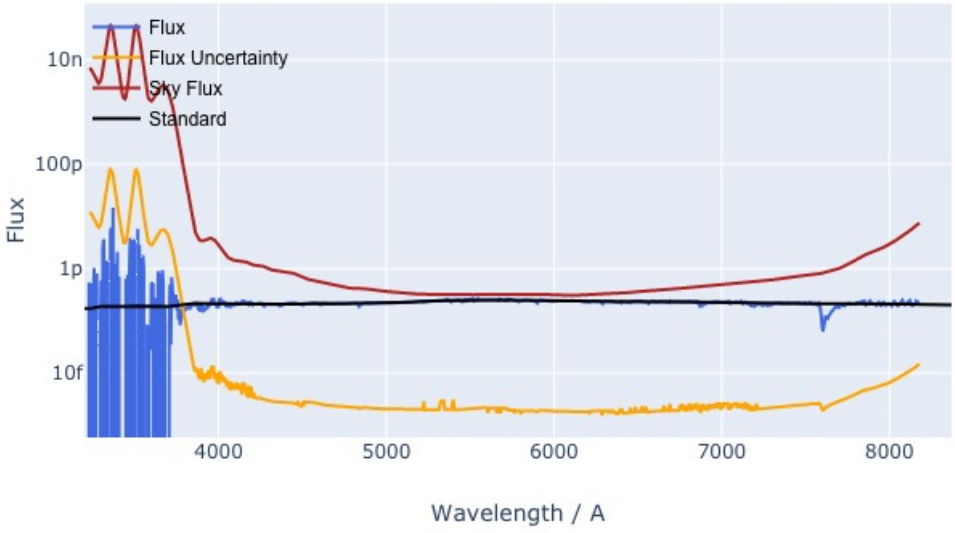
```
/Users/marcolam/git/SpectRes/spectres/spectral_resampling.py:74: UserWarning:

spectres: Part of the new wavelengths specified is outside the range of the
input data, they are filled with zeros.

aspired/aspired.py:1655: RuntimeWarning:
```

```
divide by zero encountered in true_divide
```



[10]:
```
lhs6328_reduced.apply_flux_calibration('all')
lhs6328_reduced.inspect_reduced_spectrum('all')
```


Flux
Flux Uncertainty
Sky Flux
10n
100p
1p
10f
10^{-16}
Flux
4000
5000
6000
7000
8000
Wavelength / A


Flux
Flux Uncertainty
Sky Flux
Standard
10n
100p
1p
10f
Flux
4000
5000
6000
7000
8000
Wavelength / A