REDUCING JWST DATA: FROM RAW DATA TO LIGHT CURVES (FEAT. EUREKA!)

Presented by Kevin Stevenson

2023 Sagan Summer Workshop

Pasadena, CA

GET YOUR EUREKA! STICKER TODAY

EUREKA! DATA REDUCTION TOPICS

What WILL be Discussed Today

- Introduction
- Data Download & Setup
- Stage 1 – Detector Processing
- Stage 2 – Calibrations
- Stage 3 – Data Reduction
- Stage 4 – Generate Light Curves

Important, but NOT Discussed

- Stages 5 + 6 (tomorrow)
- Installation (tutorial)
- GitHub/ReadTheDocs
  - Submitting an Issue
  - Contributing to Eureka!
  - Eureka! FAQ

PLEASE READ THE DOCS!!!

MOTIVATION FOR DEVELOPING EUREKA!

- **Programming skills should not drive scientific success.**
- Reducing JWST time-series observations requires customized software not provided by STScI's official "jwst" software package.
- Custom pipelines:
  - Seldom open-source or user friendly
  - Oftentimes lack documentation, thorough testing, and general user support
- Creates a members-only environment for those that have access to one of the few validated pipelines.
- Without access to or training for such software, new researchers trying to enter the field face a daunting barrier.
- Project philosophy:
  - To facilitate a community-supported, open-source pipeline that is modular in design and easy to use.

Sci-Fi/SyFy original (2006 – 2012) set in Eureka, Oregon

WELCOME TO THE EUREKA! COMMUNITY

Taylor James Bell
BAER Institute postdoc at NASA Ames

Kevin Stevenson

Laura Kreidberg
she/her

Eva-Maria Ahrer
PhD student at the University of Warwick

Yoni Brande

Aarynn Carter

Adina Feinstein
what I do -- stars

Giannina Guzman Caloca
1st year graduate student at UMD

Megan Mansfield
she/her

Sebastian Zieba

Caroline Piaulet

Erin May
she/her

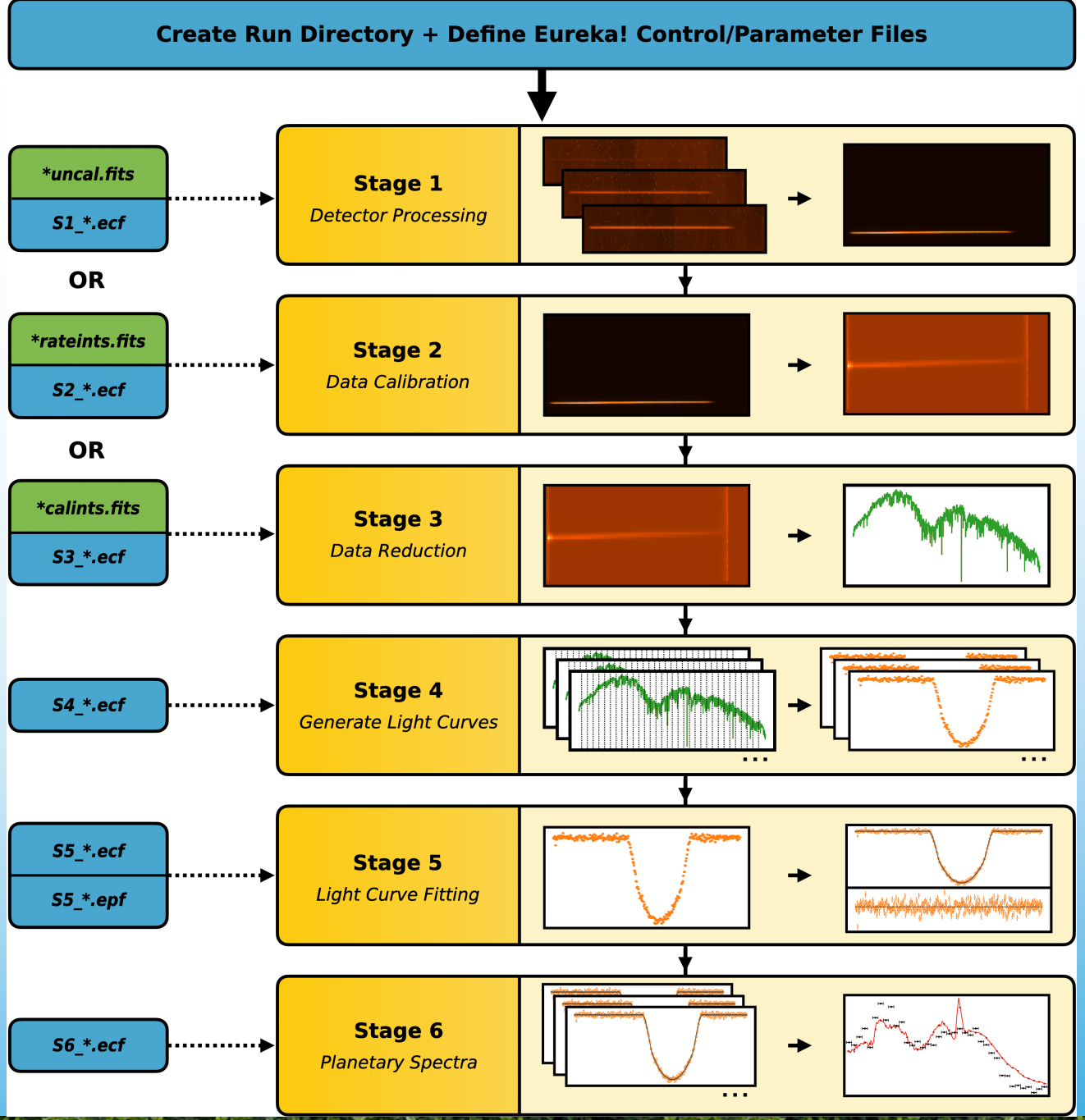
Björn Benneke

Louis-Philippe Coulombe

Pierre-Alexis Roy

BELL ET AL. (2022)

- 10.21105/joss.04503



JOSS
The Journal of Open Source Software

Eureka!: An End-to-End Pipeline for JWST Time-Series Observations

Taylor J. Bell [1], Eva-Maria Ahrer [2], Jonathan Brande [3], Aarynn L. Carter [4], Adina D. Feinstein [5], Giannina Guzman Caloca [6], Megan Mansfield [7,8], Sebastian Zieba [9], Caroline Piaulet [10], Björn Benneke [10], Joseph Filippazzo [11], Erin M. May [12], Pierre-Alexis Roy [10], Laura Kreidberg [9], and Kevin B. Stevenson [12]

1 BAER Institute, NASA Ames Research Center, Moffet Field, CA 94035, USA 2 Department of Physics, University of Warwick, Gibbet Hill Road, CV4 7AL Coventry, UK 3 Department of Physics and Astronomy, University of Kansas, 1082 Malott, 1251 Wescoe Hall Dr., Lawrence, KS 66045, USA 4 Department of Astronomy and Astrophysics, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA 5 Department of Astronomy & Astrophysics, University of Chicago, 5640 S. Ellis Avenue, Chicago, IL 60637, USA 6 Department of Astronomy, University of Maryland, College Park, MD USA 7 Steward Observatory, University of Arizona, Tucson, AZ 85719, USA 8 NHFP Sagan Fellow 9 Max-Planck-Institut für Astronomie, Königstuhl 17, D-69117 Heidelberg, Germany 10 Department of Physics and Institute for Research on Exoplanets, Université de Montréal, Montreal, QC, Canada 11 Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA 12 Johns Hopkins APL, 11100 Johns Hopkins Road, Laurel, MD 20723, USA

DOI: 10.21105/joss.04503

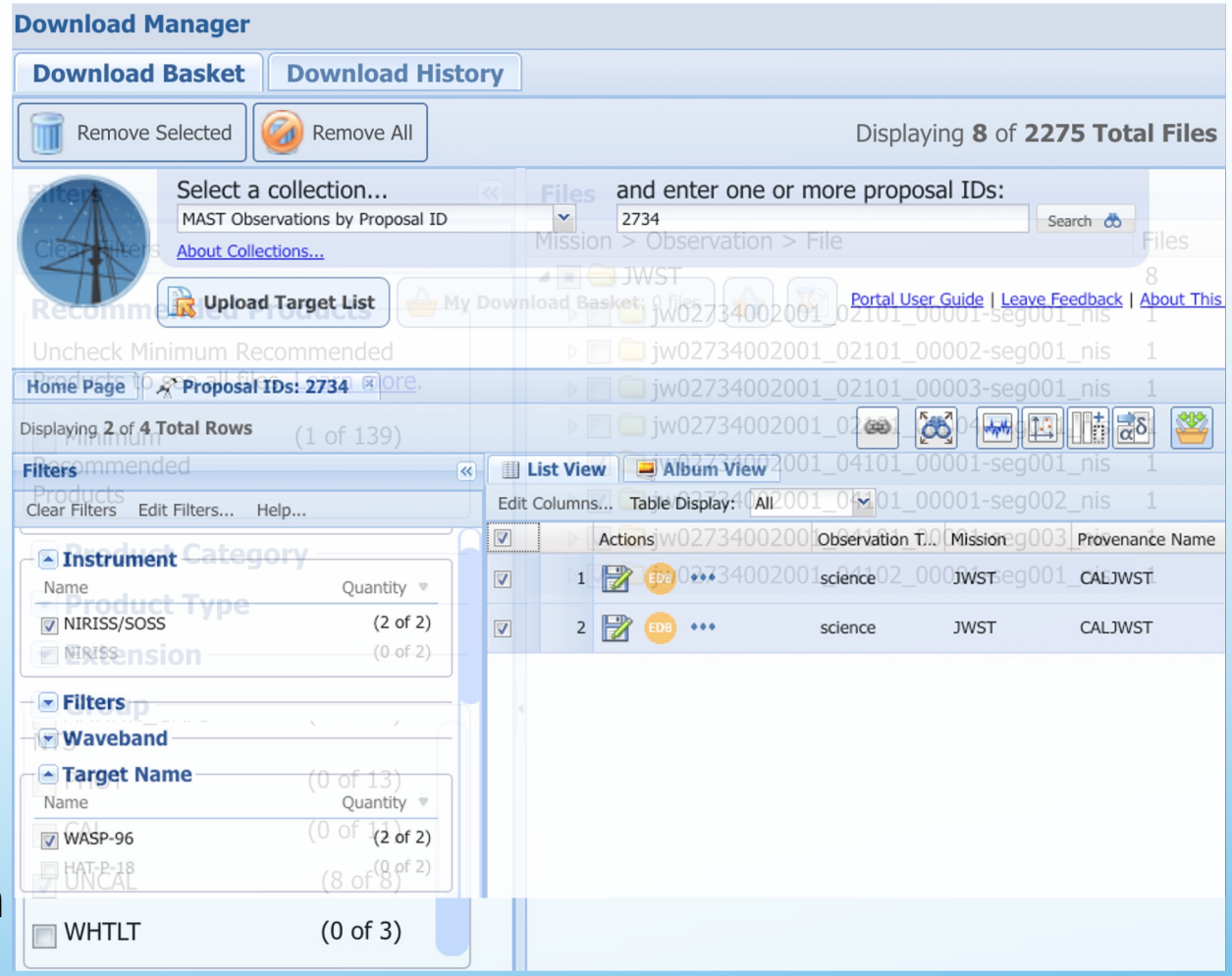
Software
- Review
- Repository
- Archive

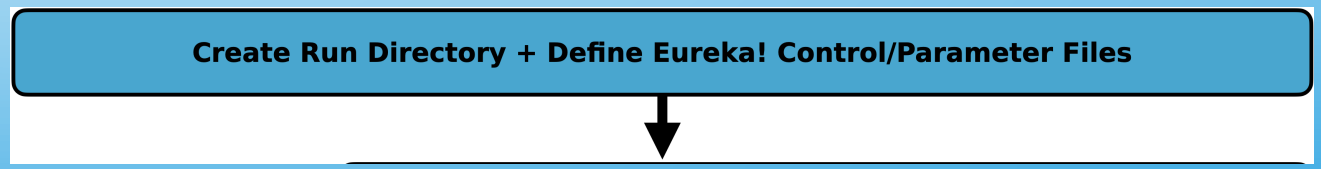
Summary

Eureka! is a data reduction and analysis pipeline for exoplanet time-series observations, with a particular focus on James Webb Space Telescope (JWST, Gardner et al., 2006) data. JWST was launched on December 25, 2021 and over the next 1-2 decades will pursue four main

DATA DOWNLOAD & SETUP

- Method 1 (MAST Website)
  - https://mast.stsci.edu/
  - Need to login to access exclusive access data
  - Filter on proposal ID
  - Select target
  - Add data products to Download Basket
  - Uncheck “Minimum Recommended Products”
  - Select “UNCAL” Group type
  - Differentiate science data (04) from target acquisition data (02)
  - Click “Download”

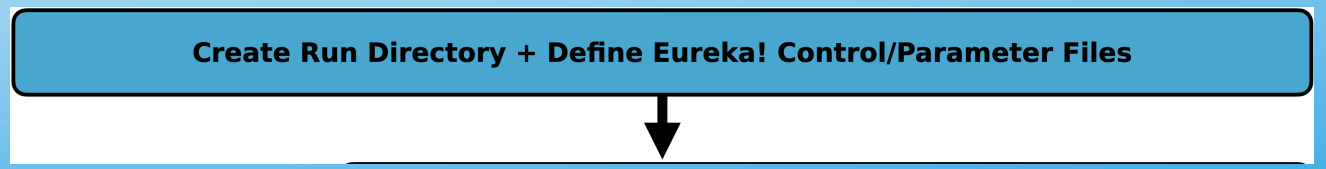




DATA DOWNLOAD & SETUP

- Method 2 (MAST API via Eureka!)
  - /Eureka/demos/JWST/download_data_JWST_template.py
  - Provide proposal ID, observation #, visit #, calibration level, subgroup, directories
  - Program information
    - https://www.stsci.edu/jwst/science-execution/program-information.html
  - Need MAST API token to access exclusive access data
    - Generate token here: https://auth.mast.stsci.edu/token

```
# Proposal/Program ID, can be string or int
proposal_id = '02734'

# Observation number
observation = 1

# Visit number
visit = 1

# Calibration level, list
# (0 = raw, 1 = uncalibrated, 2 = calibrated, 3 = science product,
# 4 = contributed science product)
calib_level = [1]

# FITS file type, varies by calib_level.
# 1: UNCAL, GS-ACQ1, GS-ACQ2, GS-FG, GS-ID, GS-TRACK
# 2: CAL, CALINTS, RATE, RATEINTS, X1DINTS, ANNNN_CRFINTS,
# GS-ACQ1, GS-ACQ2, GS-FG, GS-ID, GS-TRACK, RAMP
# 3: X1DINTS, WHTLT
subgroup = 'UNCAL'

# Temporary download directory will be 'download_dir'/mastDownload/...
download_dir = './wasp96b'

# Final destination of files after calling mast.consolidate
final_dir = './wasp96b/S0'
```

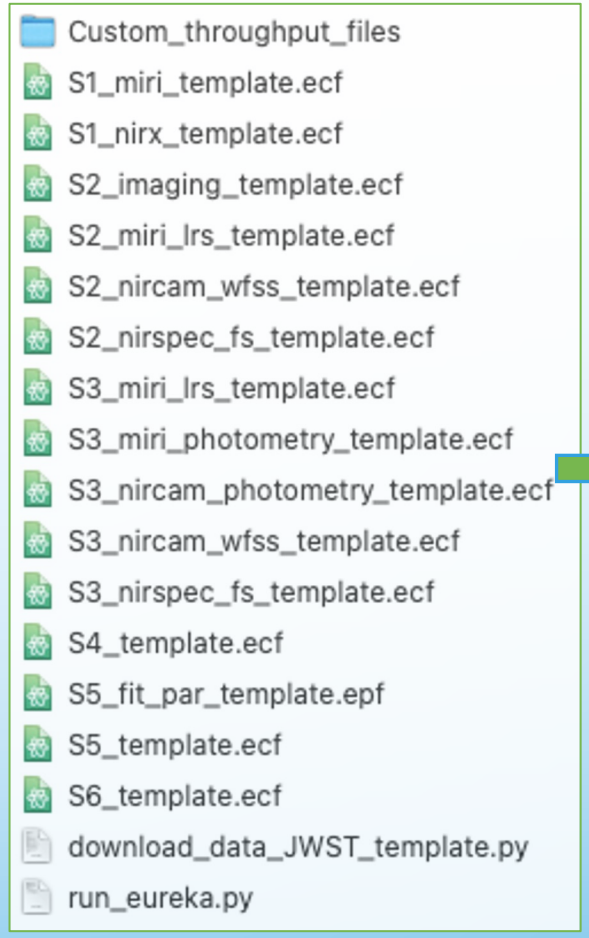


DATA DOWNLOAD & SETUP

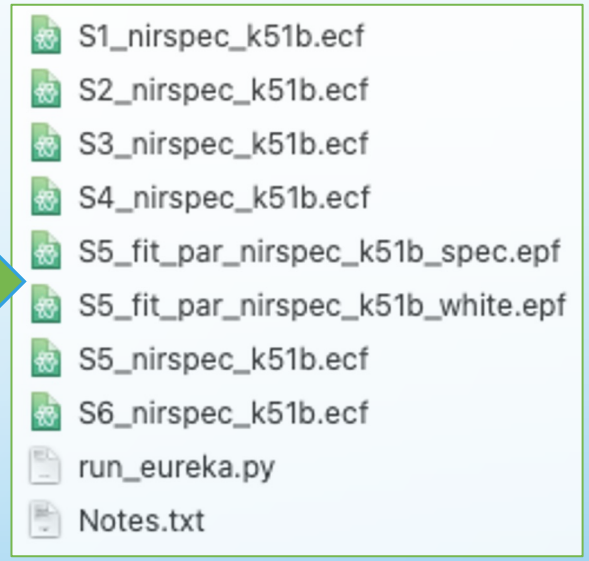
- Copy relevant files from /demos/JWST into your analysis directory
- Rename your Eureka! control files (ECFs)
- Review and edit each ECF

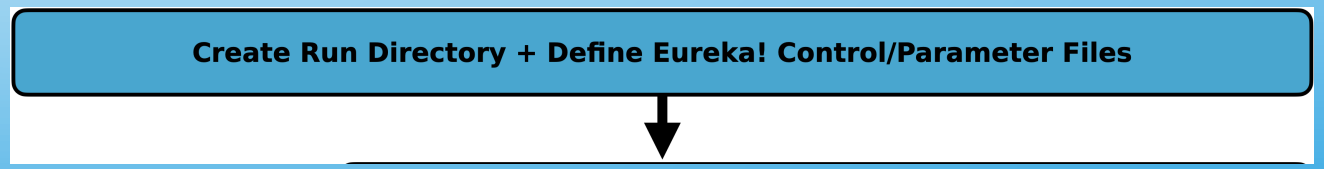
- Advice
  - Keep your directories organized
  - Take notes on what you did and why!

…/Eureka/demos/JWST



…/Analyses/Kepler-51b

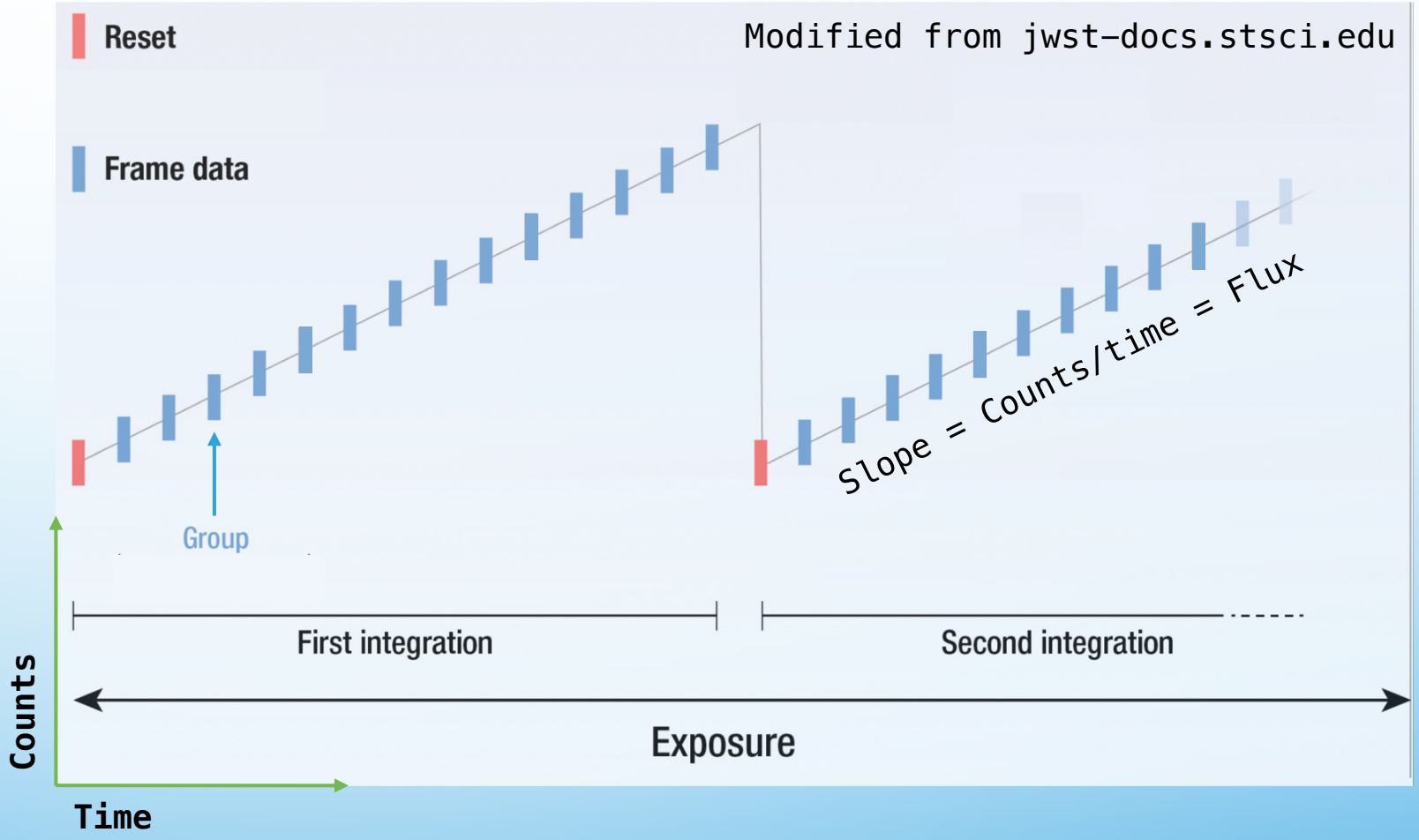


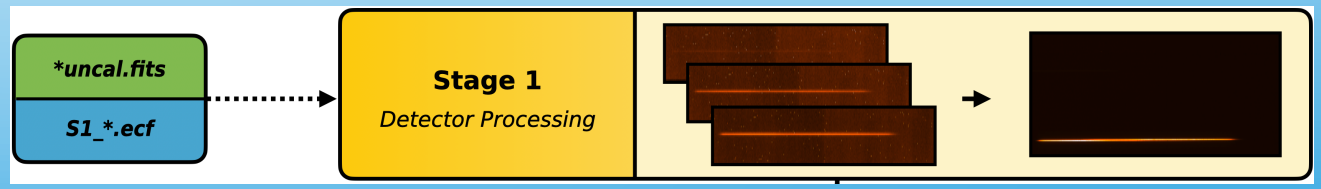


STAGE 1 – TERMINOLOGY

- What are Frames? Groups? Integrations? Exposures?

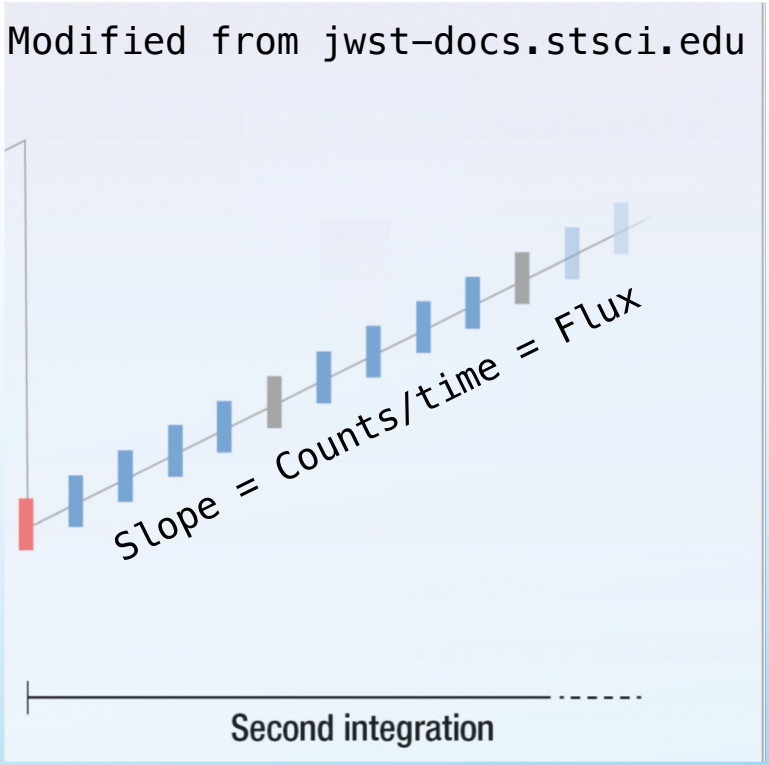
- Frame = Group
  - 1 frame per group for TSO
  - Non-destructive read
  - 5–100+ groups per integration
- Integration
  - Single point in your light curve
  - 1000s of integrations per exposure
- Exposure (or “visit”)
  - Transit or eclipse

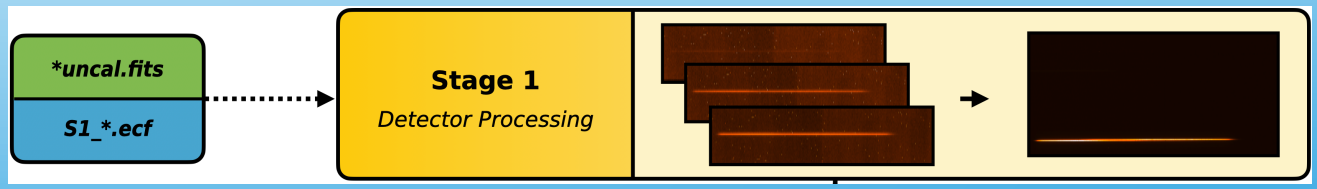




STAGE 1 – "RAMPS TO SLOPES"

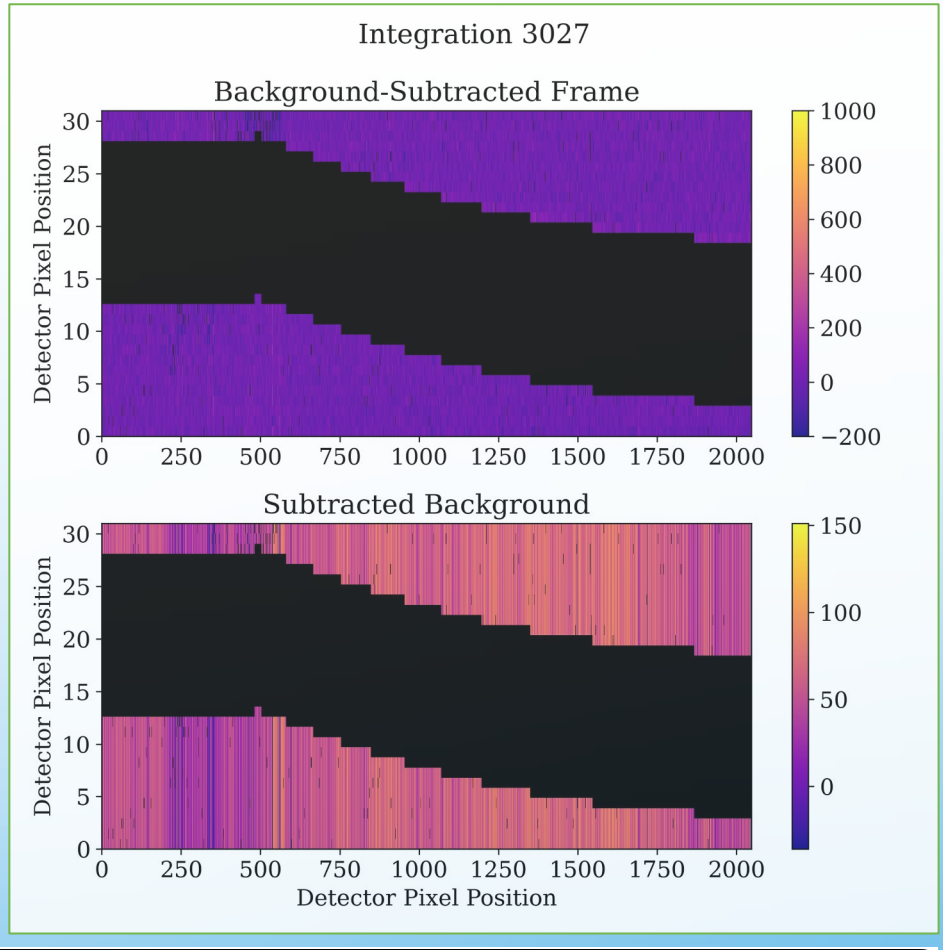
- Eureka! is a wrapper for jwst Stage 1 pipeline
- Online documentation
  - Calwebb_detector1.html
- Input: 4D raw data
  - ncols x nrows x ngroups x nintegrations
  - *_uncal.fits
- Output: 3D countrate product
  - ncols x nrows x nintegrations
  - *_rateints.fits
  - "Uncalibrated slopes"
- Run time ~ 1 – 24 hours

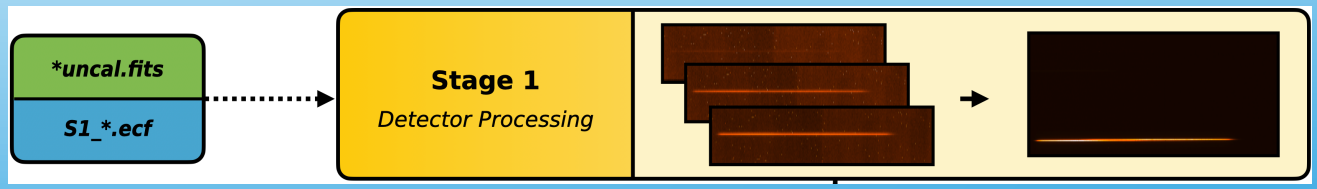




STAGE 1 – USER CONSIDERATIONS

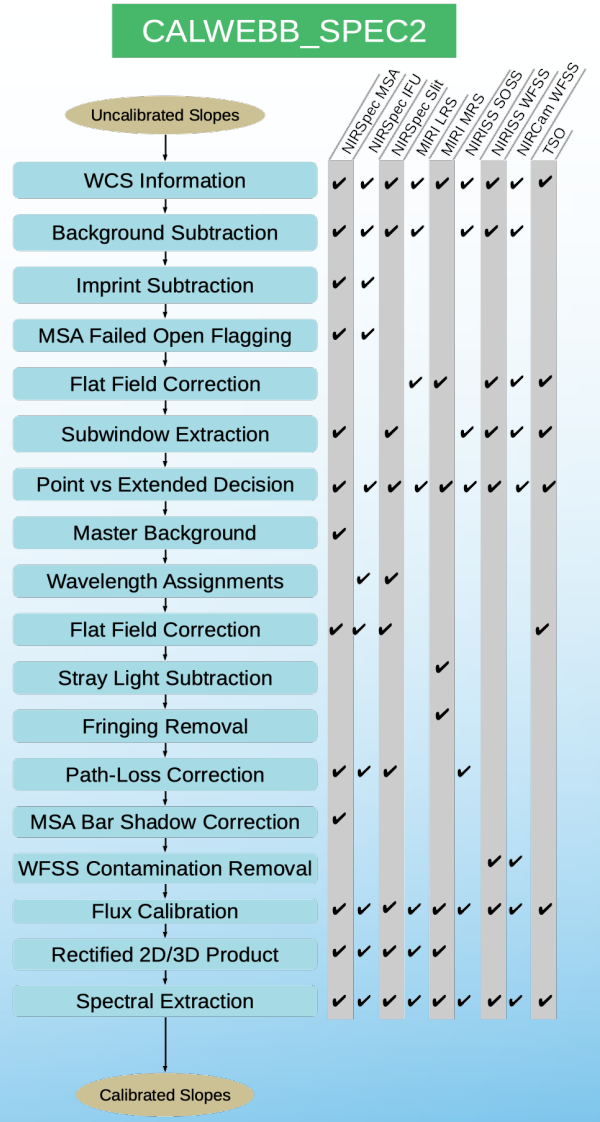
- Group-level background subtraction
  - Column-by-column BG subtraction on each group
  - Works well when # of groups is small
- Jump detection
  - Flags cosmic rays when fitting slope
  - Works well when # of groups is large and threshold is large (~15 for NIRSpec)
- Saturation
  - Flag entire columns
  - Expand saturation flags to previous group
- Custom bias correction
  - May help NIRSpec data

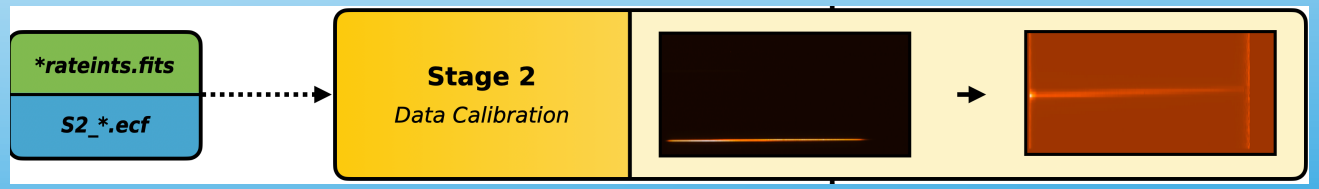




STAGE 2 – CALIBRATED SLOPES

- Eureka! is a wrapper for jwst Stage 2 pipeline
- Online documentation
  - Spectroscopy: calwebb_Spec2.html
  - Photometry: calwebb_image2.html
- Input: 3D “Uncalibrated slopes”
  - *_rateints.fits
- Output: 3D “Calibrated slopes”
  - *_calints.fits
- Run time ~ 1 minute



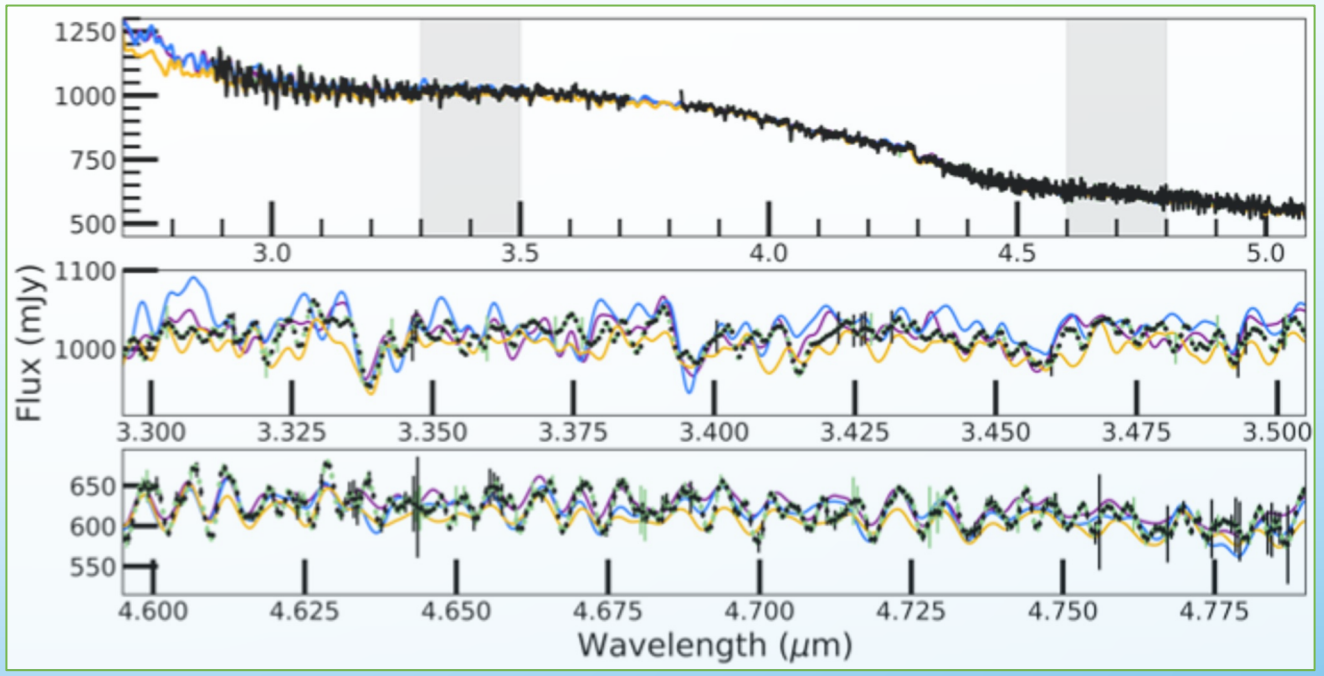


STAGE 2 – USER CONSIDERATIONS

- Subwindow extraction
  - Cross-dispersion extraction height
  - NIRCam: tsgrism_extract_height
  - NIRSpec: slit_y_low, slit_y_high

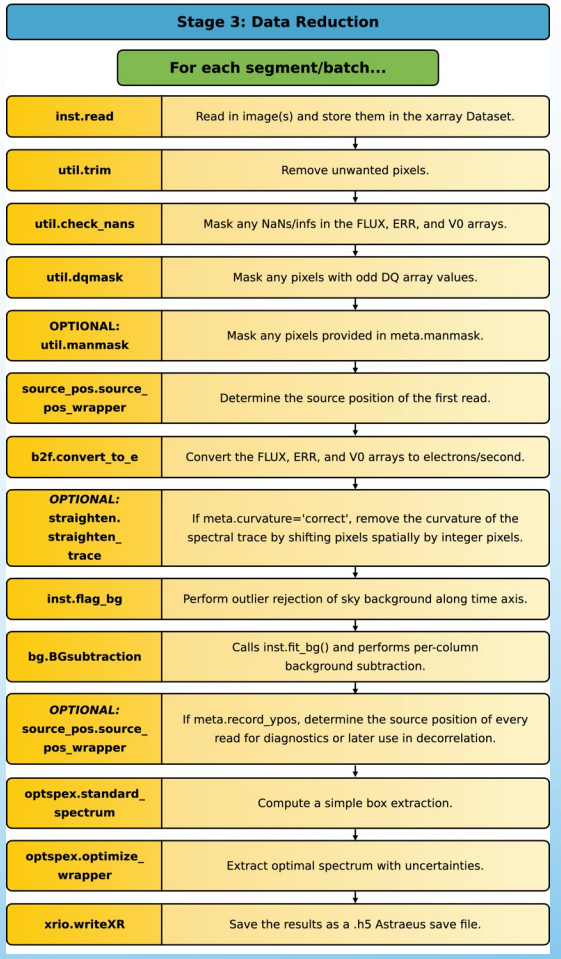
- Flux calibration
  - If skip_photom = True
    - Default for TSO reductions
    - Smaller uncertainties
    - Units of DN/s
  - If skip_photom = False
    - Calibrated stellar spectra
    - Units of mJy

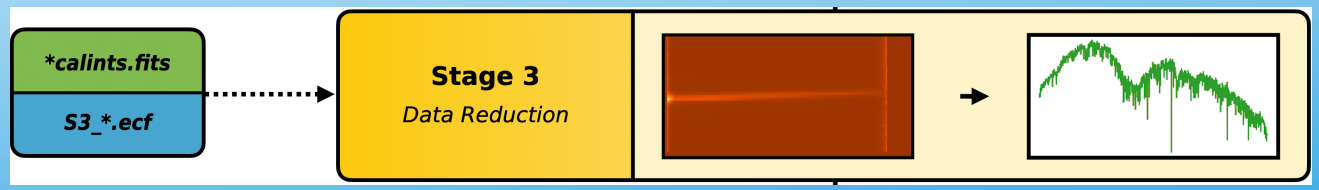
GJ 486 Spectrum – Moran & Stevenson et al. (2023)



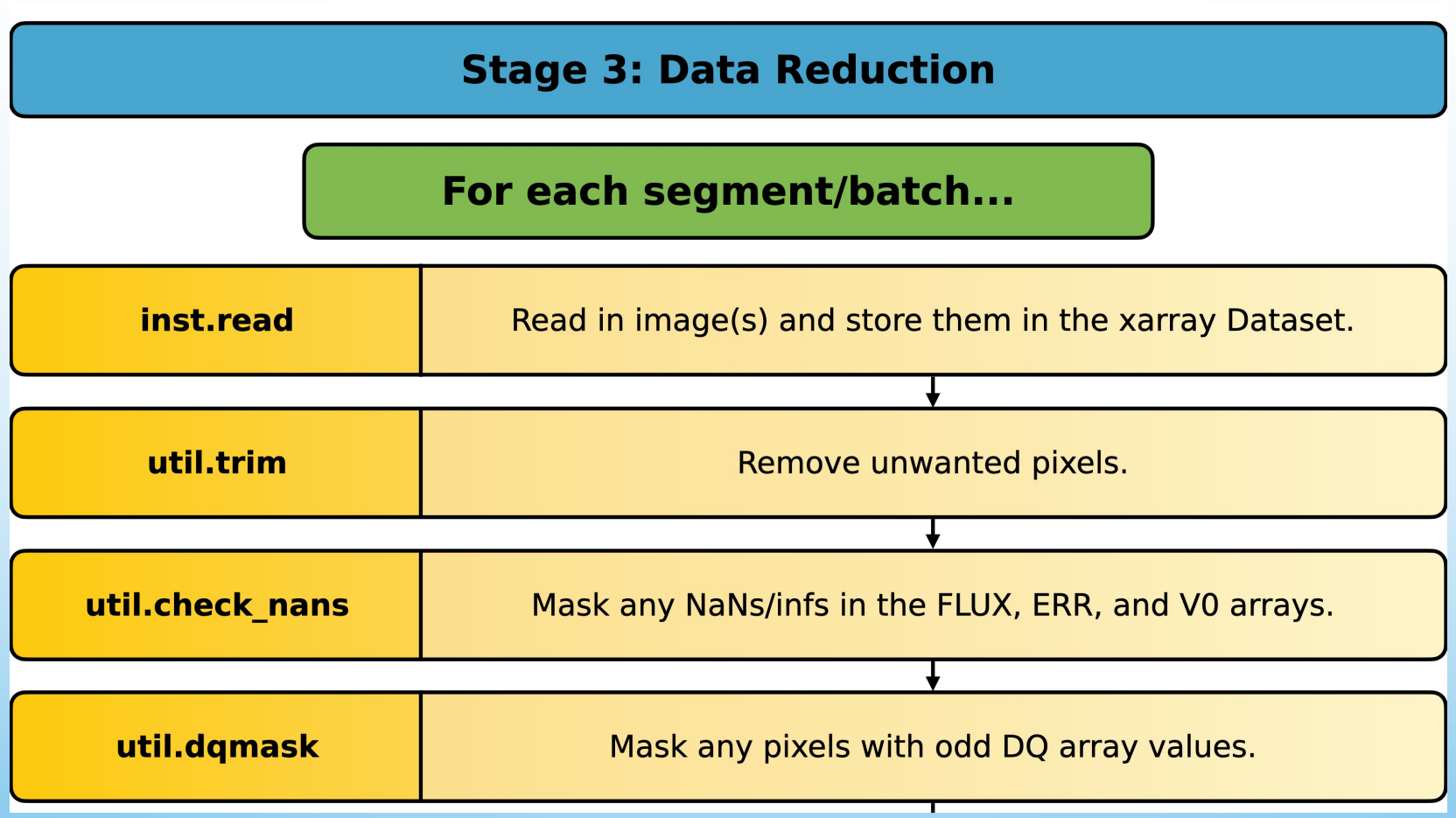
STAGE 3 – TIME SERIES OF 1D SPECTRA

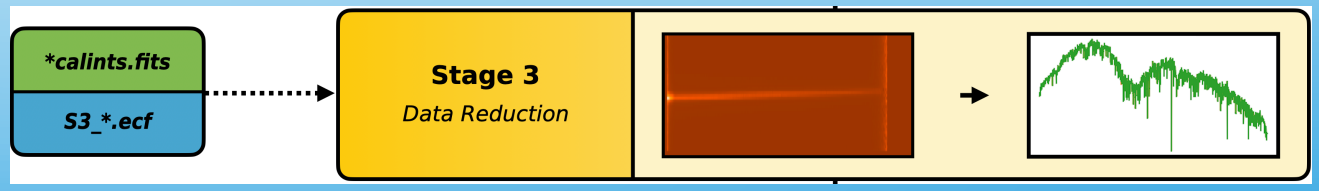
- Eureka! has its own Stage 3 pipeline
- Online documentation
  - Eureka! ReadTheDocs Stage 3
- Input: 3D “Calibrated slopes”
  - *_calints.fits
- Outputs: Time series of 1D spectra
  - S3_*_FluxData.h5 (time, wavelength)
  - S3_*_SpecData.h5 (time, x, y)
  - S3_*_Meta_Save.h5 (ECF parameters, etc.)
- Uses Xarray data structures
- Run time ~ 10 minutes





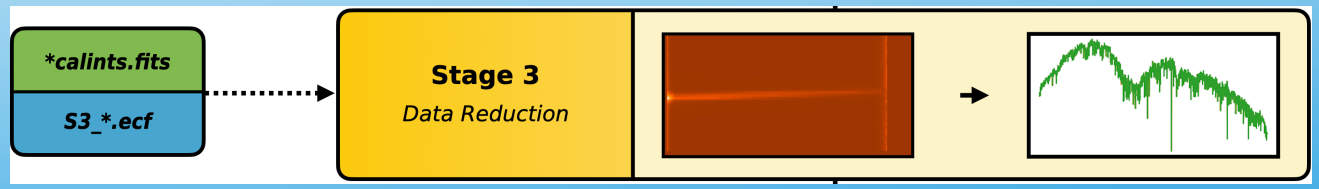
STAGE 3 – FLOWCHART





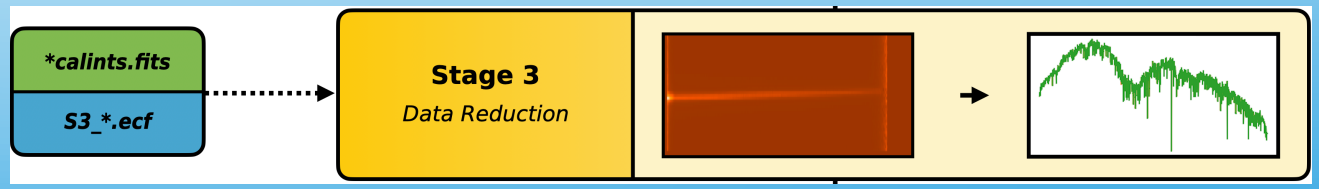
STAGE 3 – FLOWCHART

OPTIONAL: util.manmask	Mask any pixels provided in meta.manmask.
source_pos.source_pos_wrapper	Determine the source position of the first read.
b2f.convert_to_e	Convert the FLUX, ERR, and V0 arrays to electrons/second.
***OPTIONAL:* straighten. straighten_trace**	If meta.curvature='correct', remove the curvature of the spectral trace by shifting pixels spatially by integer pixels.
inst.flag_bg	Perform outlier rejection of sky background along time axis.

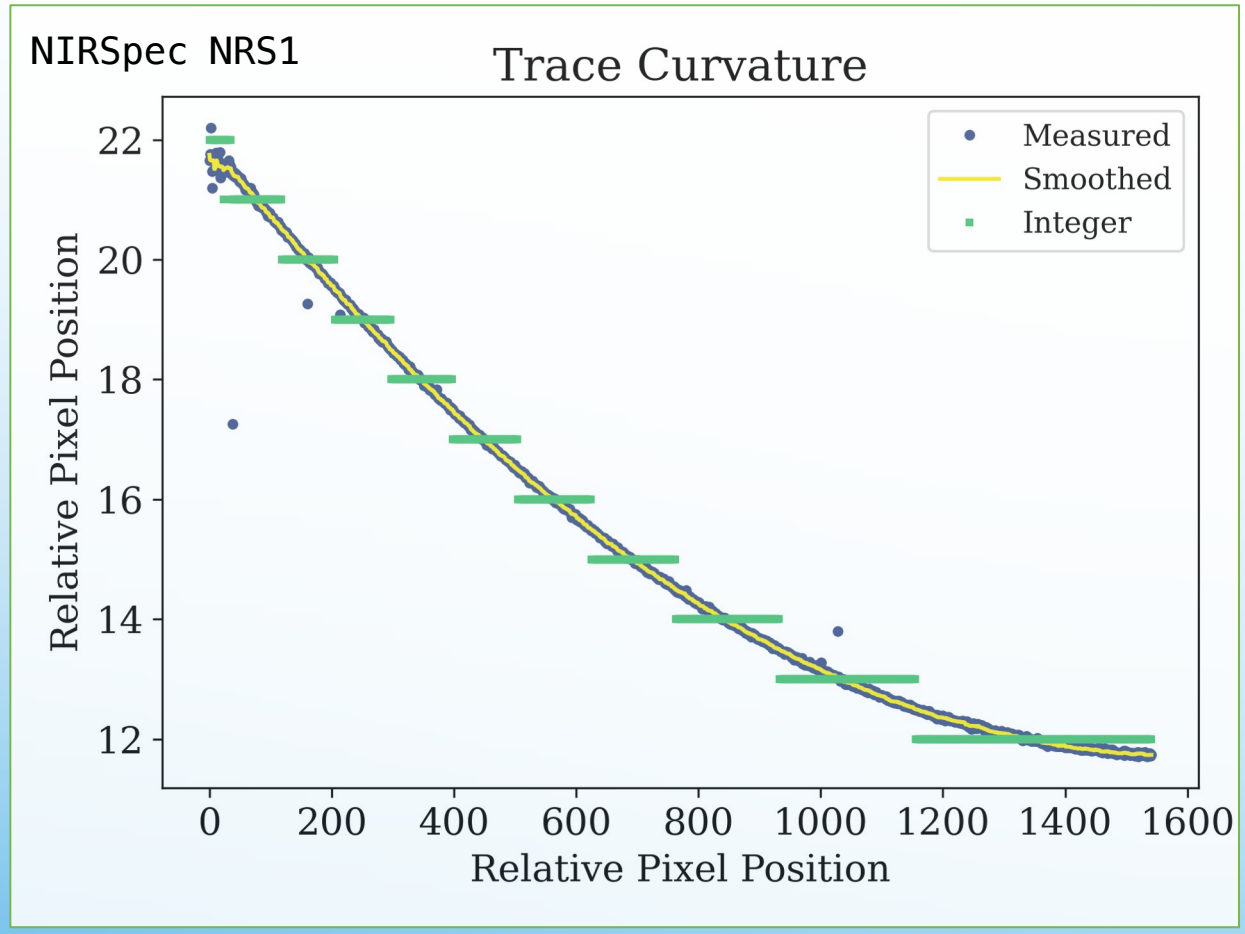


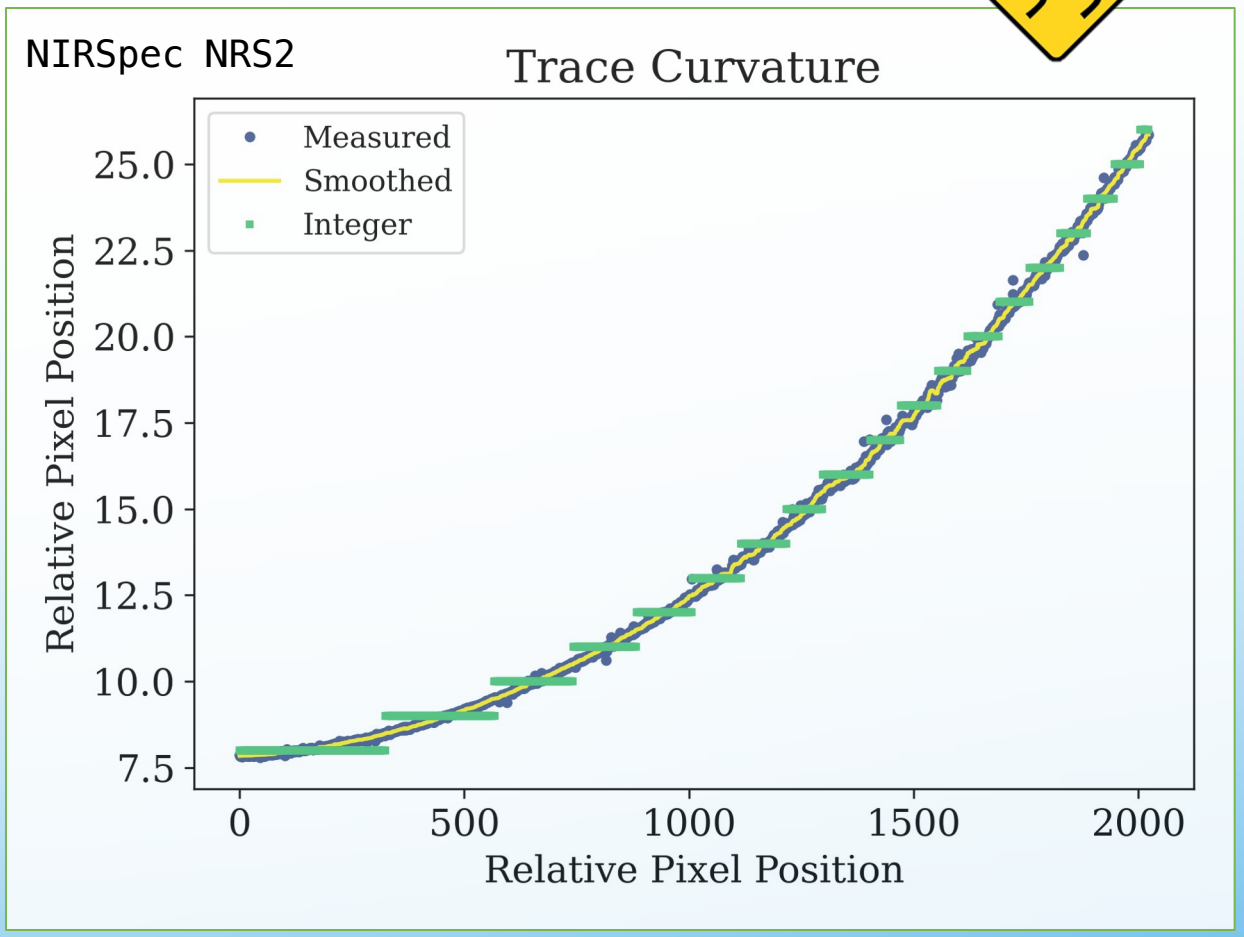
STAGE 3 – FLOWCHART

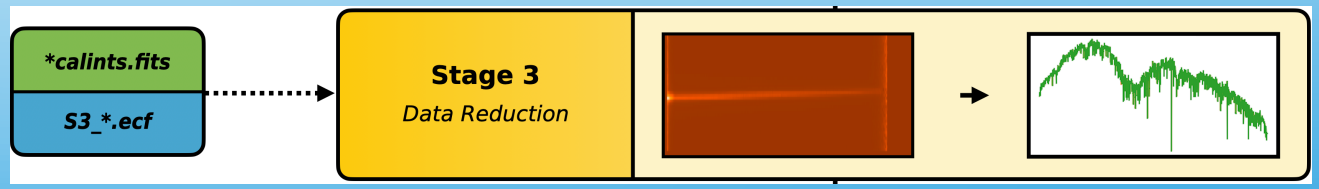
bg.BGsubtraction	Calls inst.fit_bg() and performs per-column background subtraction.
***OPTIONAL:* source_pos.source_pos_wrapper**	If meta.record_ypos, determine the source position of every read for diagnostics or later use in decorrelation.
optspex.standard_spectrum	Compute a simple box extraction.
optspex.optimize_wrapper	Extract optimal spectrum with uncertainties.
xrio.writeXR	Save the results as a .h5 Astraeus save file.



STAGE 3 – CURVATURE CORRECTION

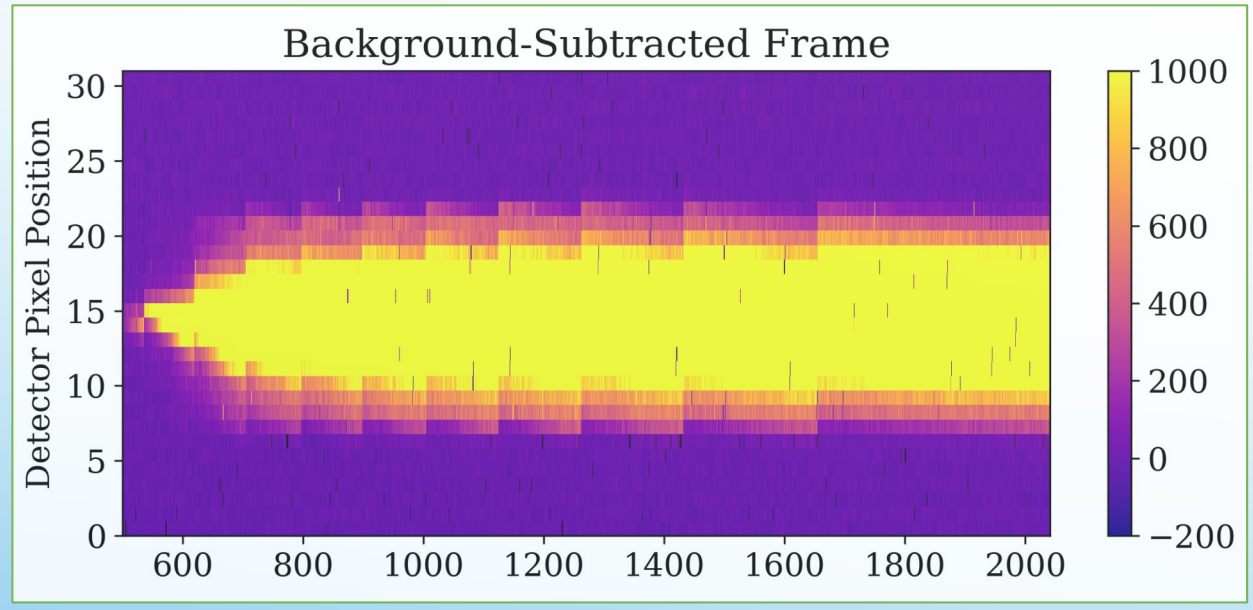




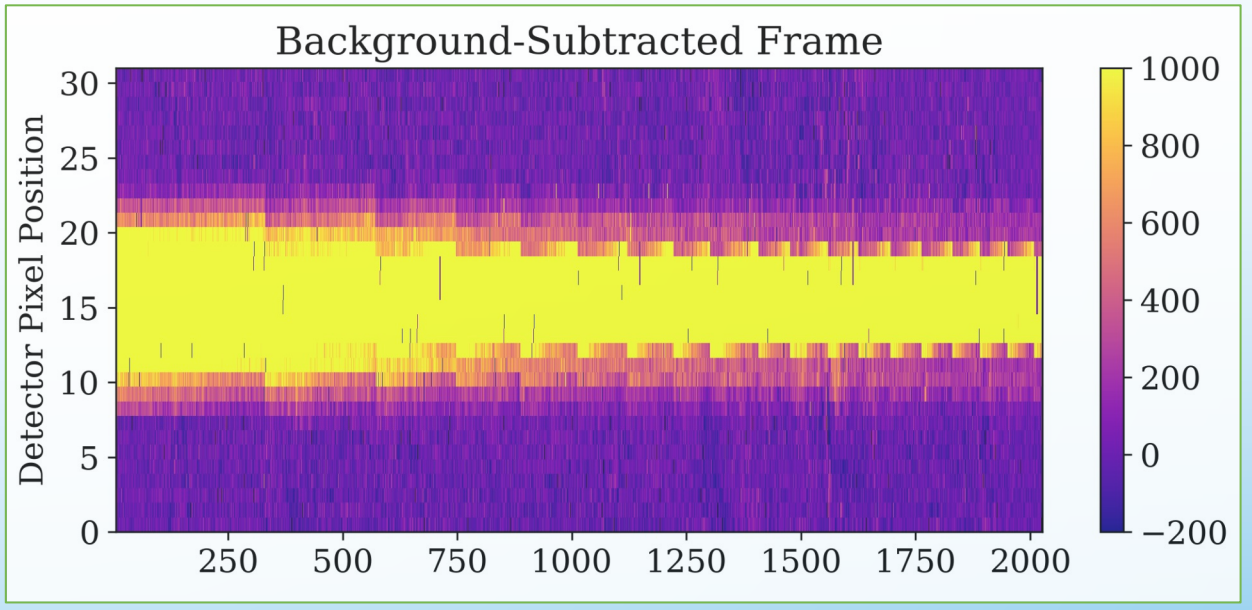


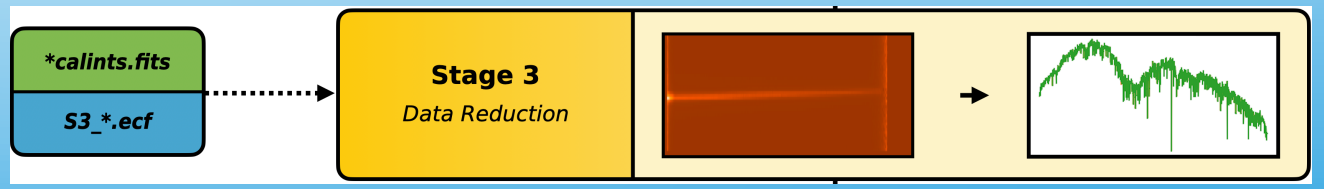
STAGE 3 – CURVATURE CORRECTION

NIRSpec NRS1



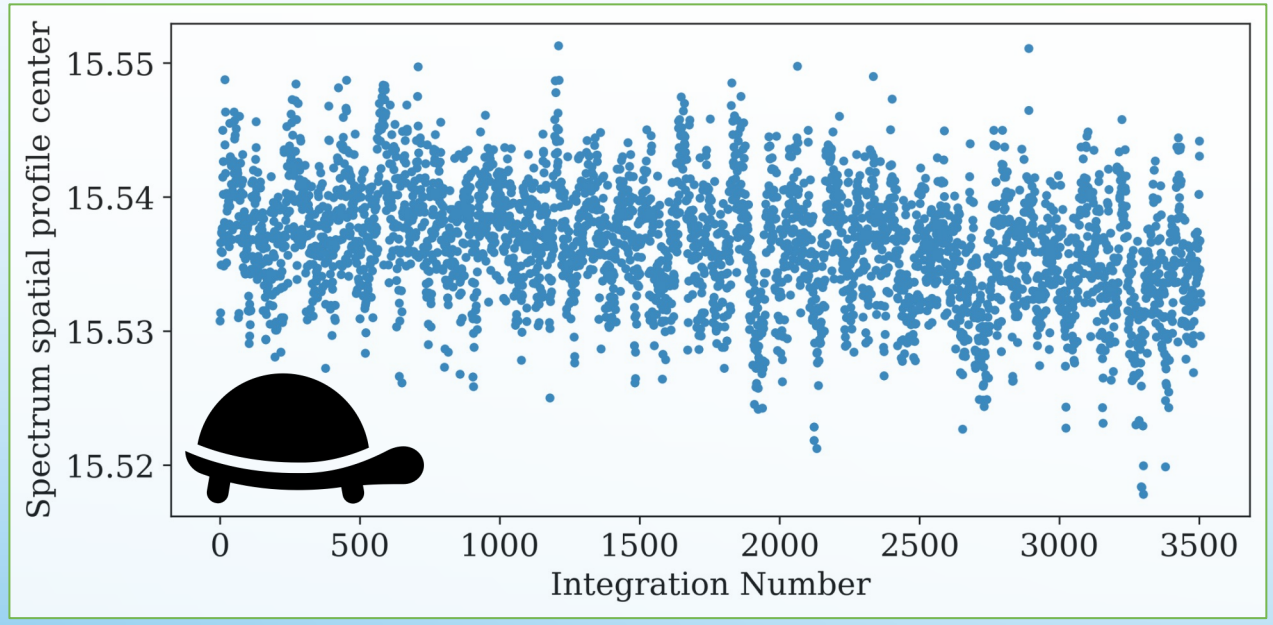
NIRSpec NRS2



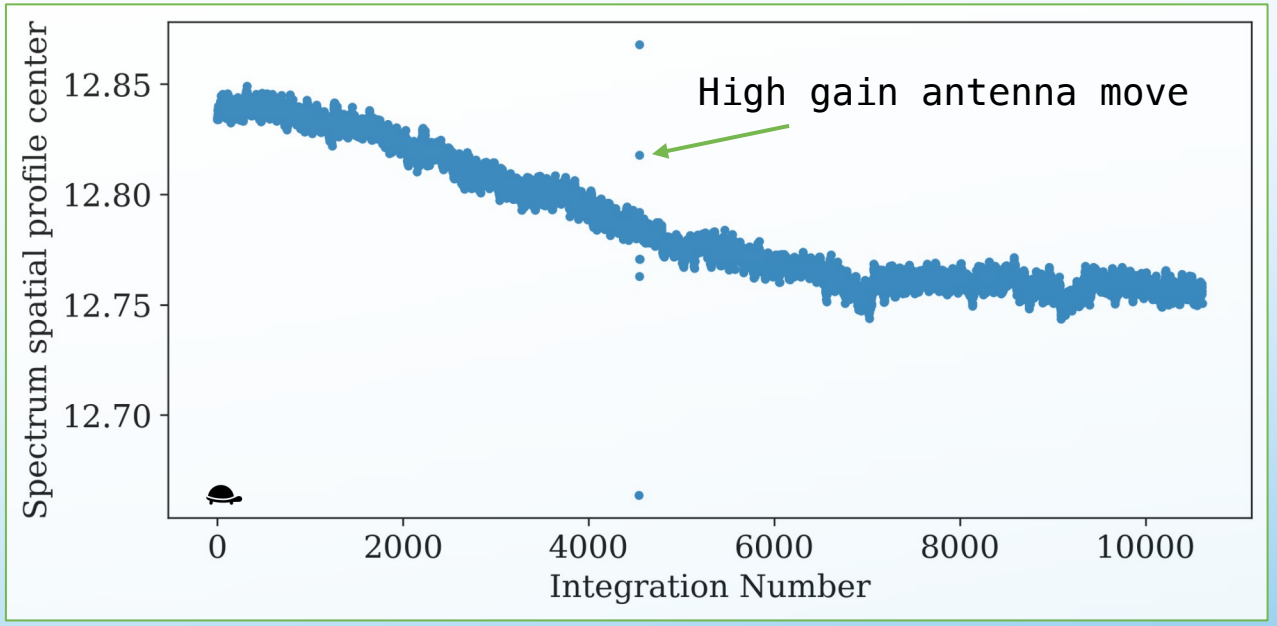


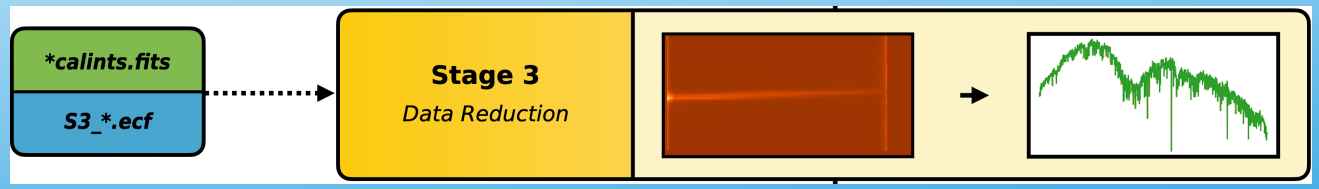
STAGE 3 – WATCH OUT FOR DRIFT!

Minimal Drift (<0.02 pixels)

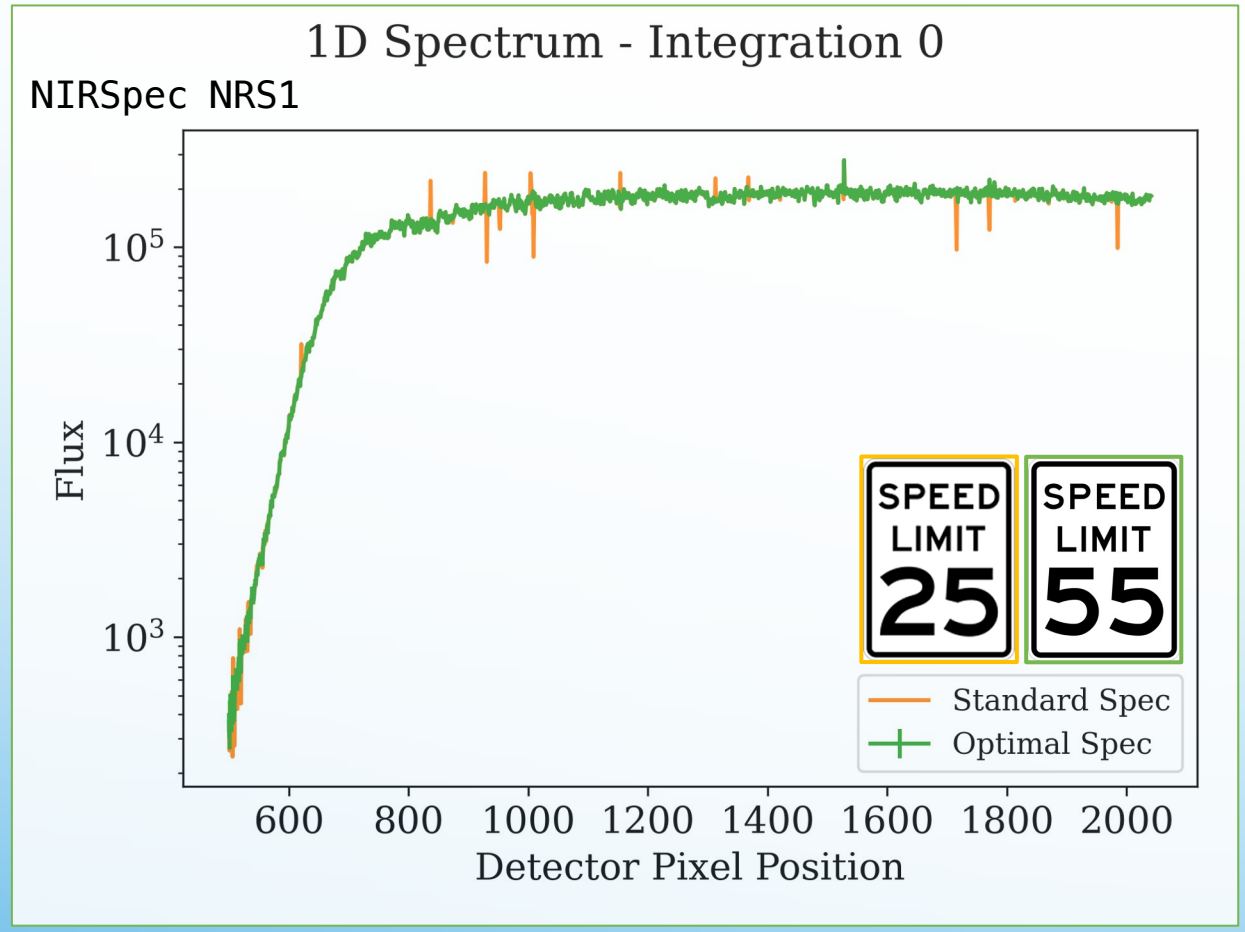


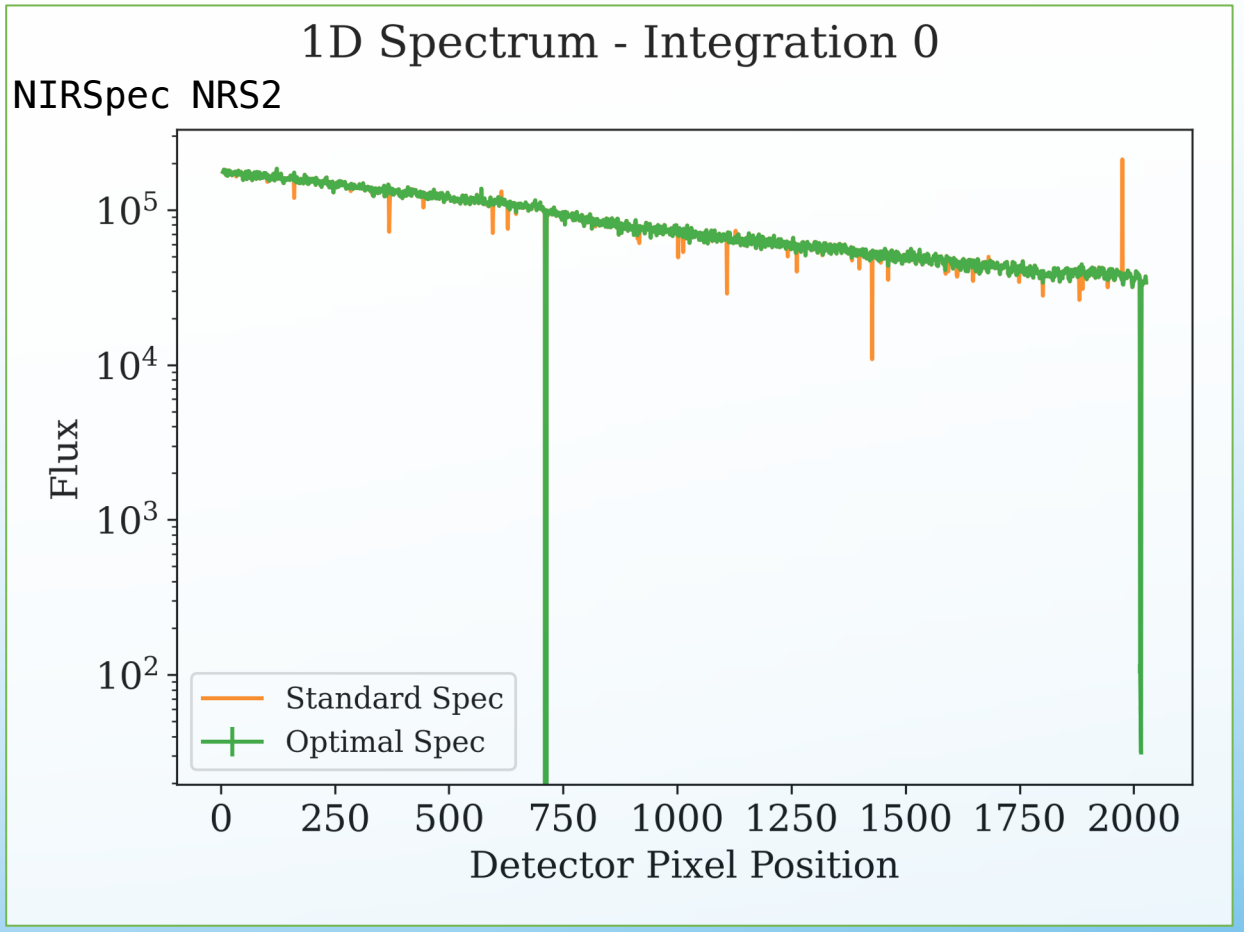
Significant Drift (~0.1 pixels)

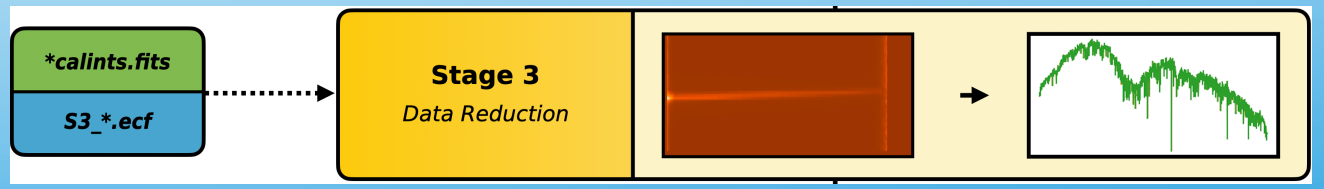




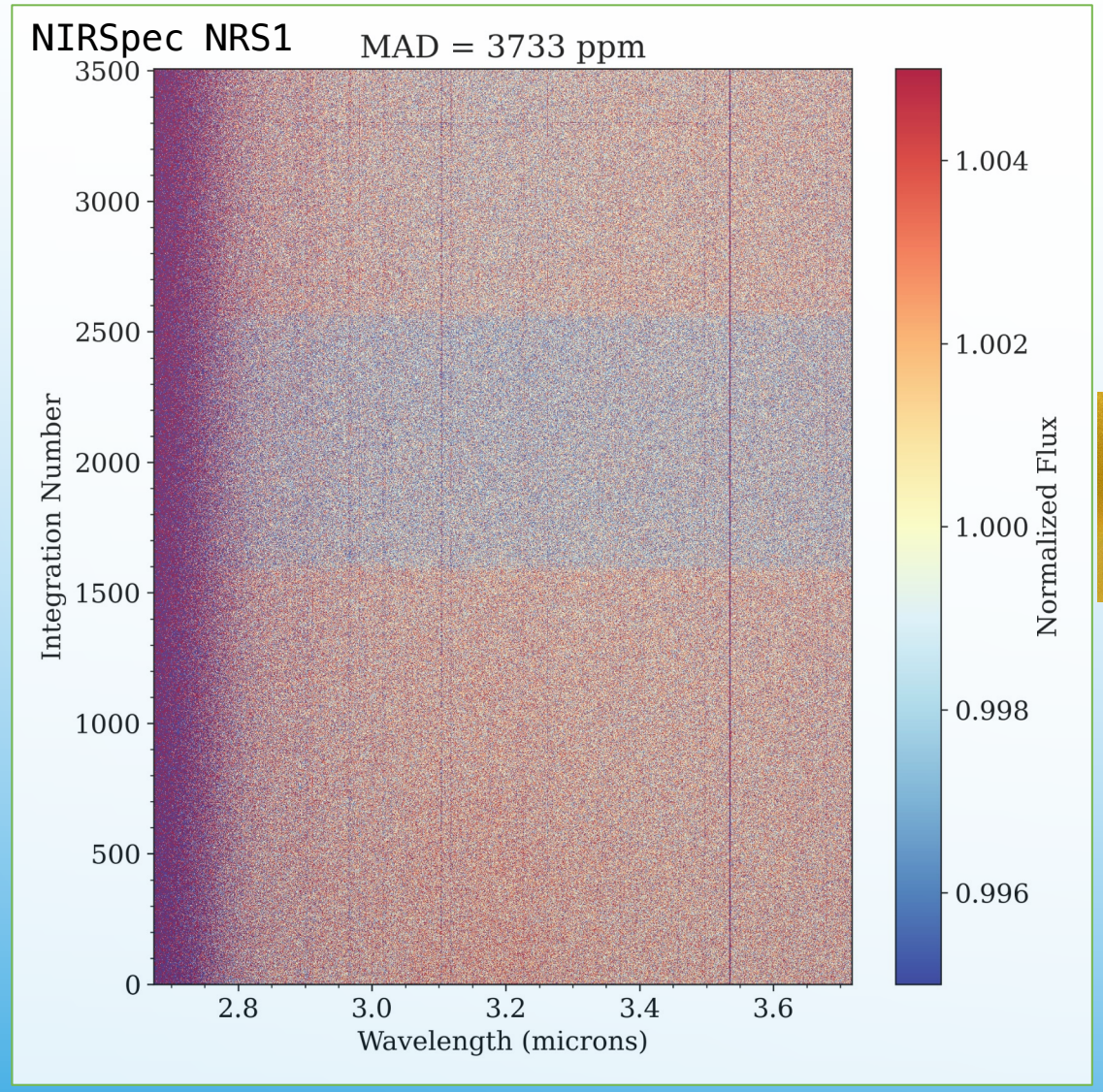
STAGE 3 – STANDARD VS. OPTIMAL

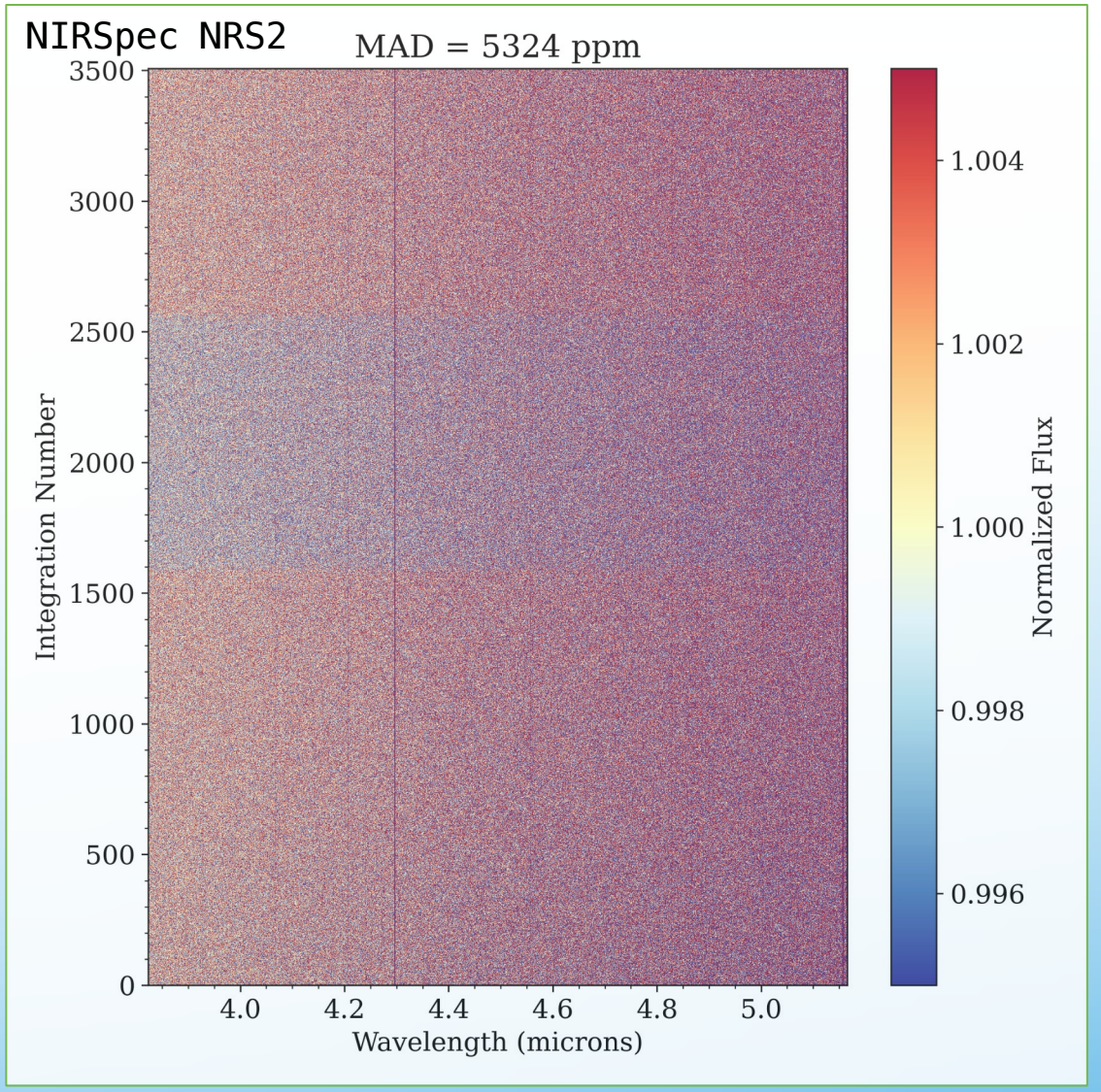


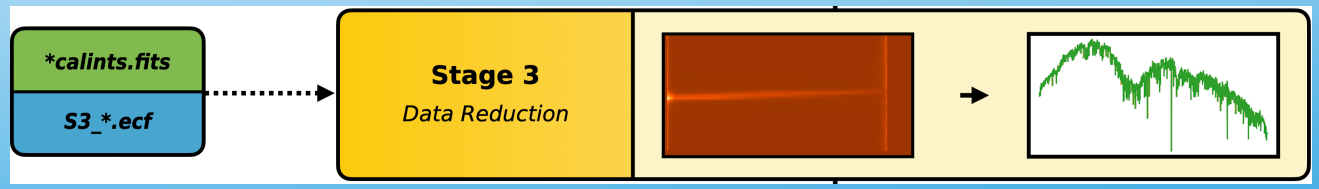




STAGE 3 – LIGHT CURVES



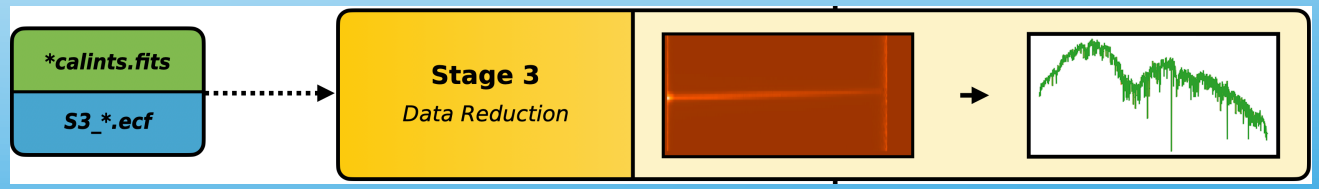




STAGE 3 – USER CONSIDERATIONS

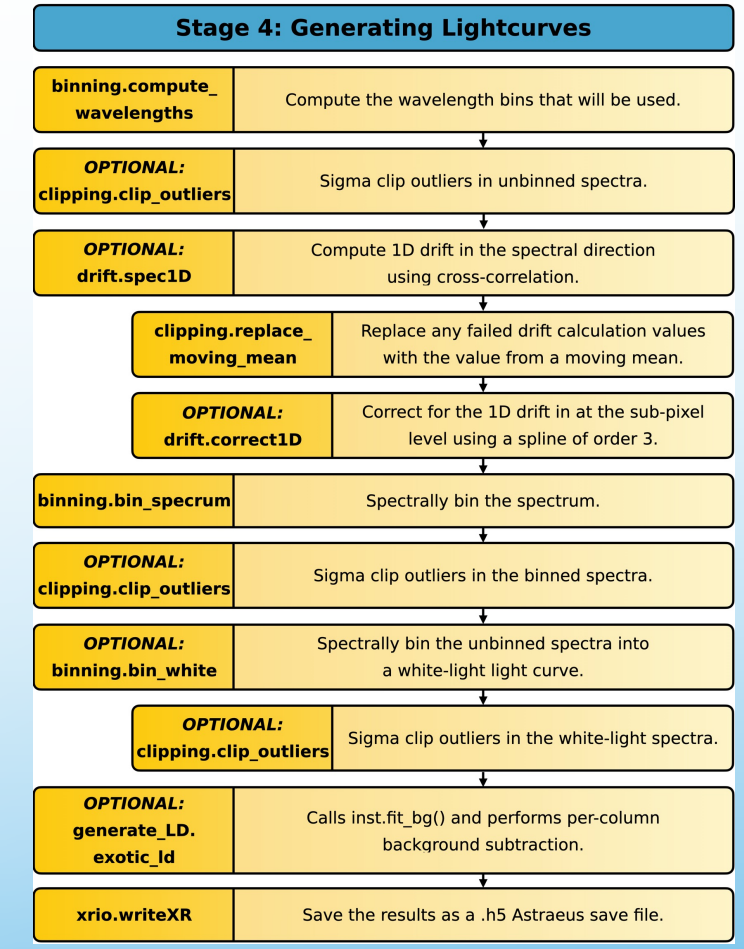
```
# Eureka! Control File for Stage 3: Data Reduction

# Stage 3 Documentation: https://eurekadocs.readthedocs.io/en/latest/ecf.html#stage-3

ncpu            1           # Number of CPUs
nfiles          1           # The number of data files to analyze simultaneously
max_memory      0.5         # The maximum fraction of memory you want utilized by read
suffix          calints     # Data file suffix

convert_to_e    True        # Whether or not a conversion to electrons should be perfo
photometry)

# Subarray region of interest
ywindow         [2,28]      # Vertical axis as seen in DS9
xwindow         [60,410]    # Horizontal axis as seen in DS9
src_pos_type    gaussian    # Determine source position when not given in header (Opti
record_ypos     True        # Option to record the y position and width for each integ
dqmask          True        # Mask pixels with an odd entry in the DQ array

# Outlier rejection along time axis
ff_outlier      False       # Set False to use only background region (recommended for
                            # Set True to use full frame (works well for shallow trans
bg_thresh       [4,4]       # Double-iteration X-sigma threshold for outlier rejection

# Background parameters
bg_hw           7           # Half-width of exclusion region for BG subtraction (relat
bg_deg          0           # Polynomial order for column-by-column background subtrac
p3thresh        10          # X-sigma threshold for outlier rejection during backgroun

# Spectral extraction parameters
spec_hw         6           # Half-width of aperture region for spectral extraction (r
fittype         smooth      # Method for constructing spatial profile (Options: smooth
median_thresh   5           # Sigma threshold when flagging outliers in median frame,
window_len      13          # Smoothing window length, for median frame or when fittyp
1 for no smoothing when computing median frame for fittype=meddata.
prof_deg        3           # Polynomial degree, when fittype = poly
p5thresh        10          # X-sigma threshold for outlier rejection while constructi
p7thresh        60          # X-sigma threshold for outlier rejection during optimal s
```

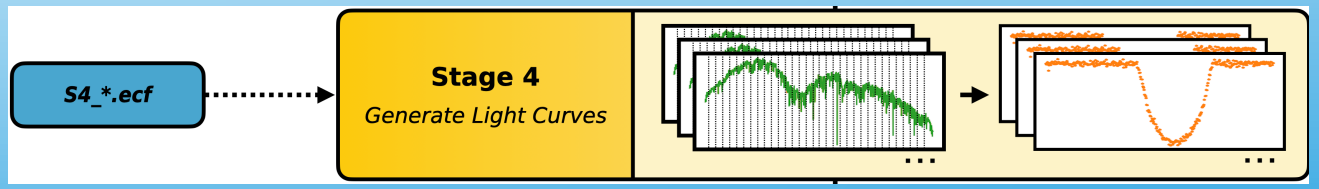
- Aperture size
  - spec_hw
- Background size
  - bg_hw
- Sigma thresholds
  - median_thresh
  - bg_thresh
  - p3thresh
  - p7thresh



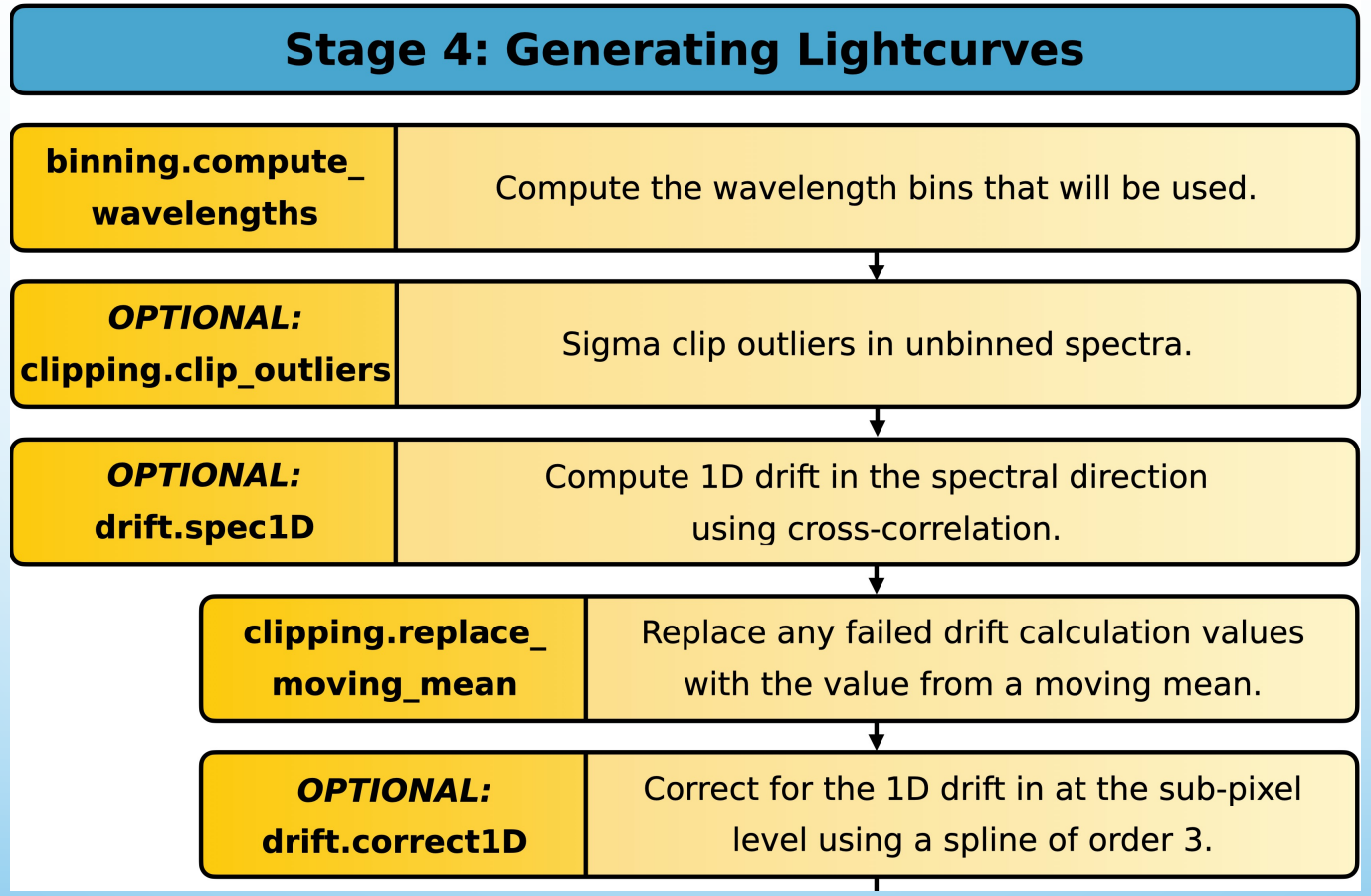
STAGE 4 – GENERATE LIGHT CURVES

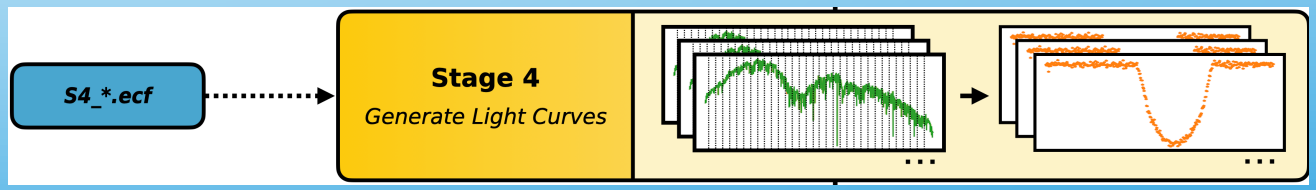
- Eureka! has its own Stage 4 pipeline
- Online documentation
  - Eureka! ReadTheDocs Stage 4
- Input: Time series of 1D spectra
  - S3_*_FluxData.h5 (time, wavelength)
  - S3_*_SpecData.h5 (time, x, y)
- Outputs: Spectroscopic light curves
  - S4_*_LC_Data.h5 (time, wavelength)
  - S4_*_SpecData.h5 (time, x, y)
  - S4_*_Meta_Save.h5 (ECF parameters, etc.)
- Run time ~ 2 minutes





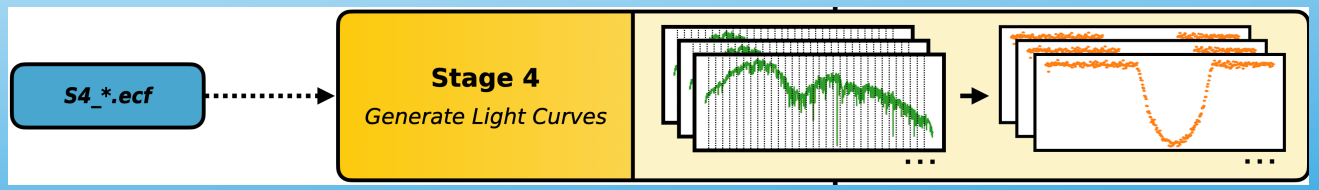
STAGE 4 – FLOWCHART





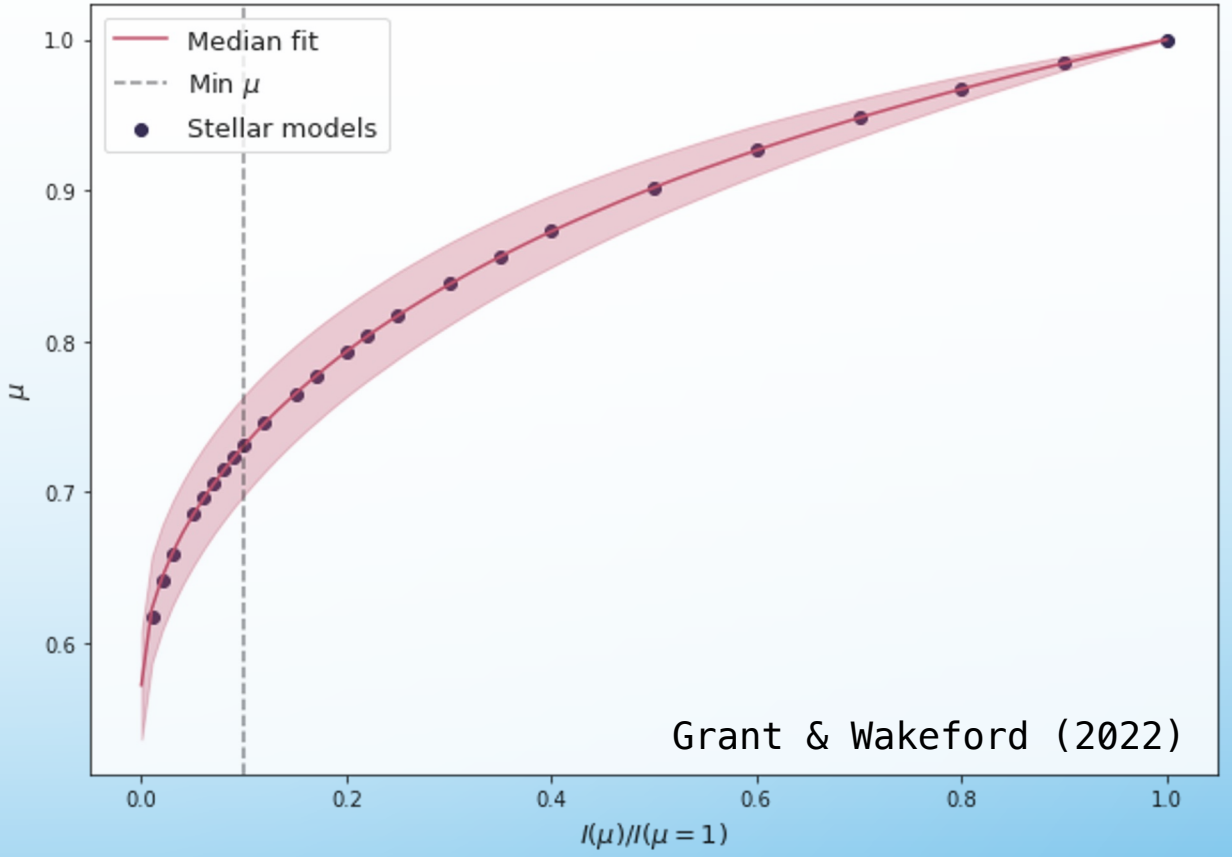
STAGE 4 – FLOWCHART

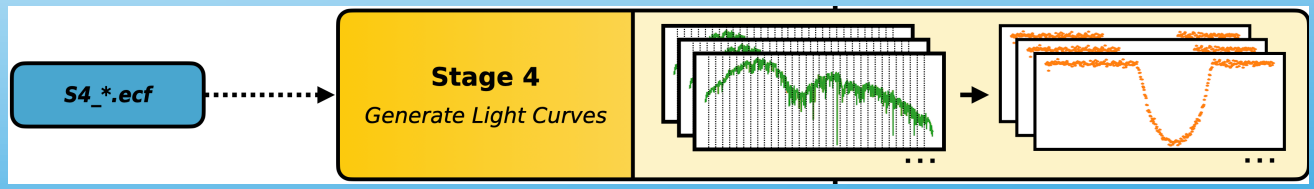
binning.bin_specrum	Spectrally bin the spectrum.
OPTIONAL: **clipping.clip_outliers**	Sigma clip outliers in the binned spectra.
OPTIONAL: **binning.bin_white**	Spectrally bin the unbinned spectra into a white-light light curve.
OPTIONAL: **clipping.clip_outliers**	Sigma clip outliers in the white-light spectra.
OPTIONAL: **generate_LD. exotic_ld**	Use ExoTiC-LD to generate various limb-darkening coefficients
xrio.writeXR	Save the results as a .h5 Astraeus save file.



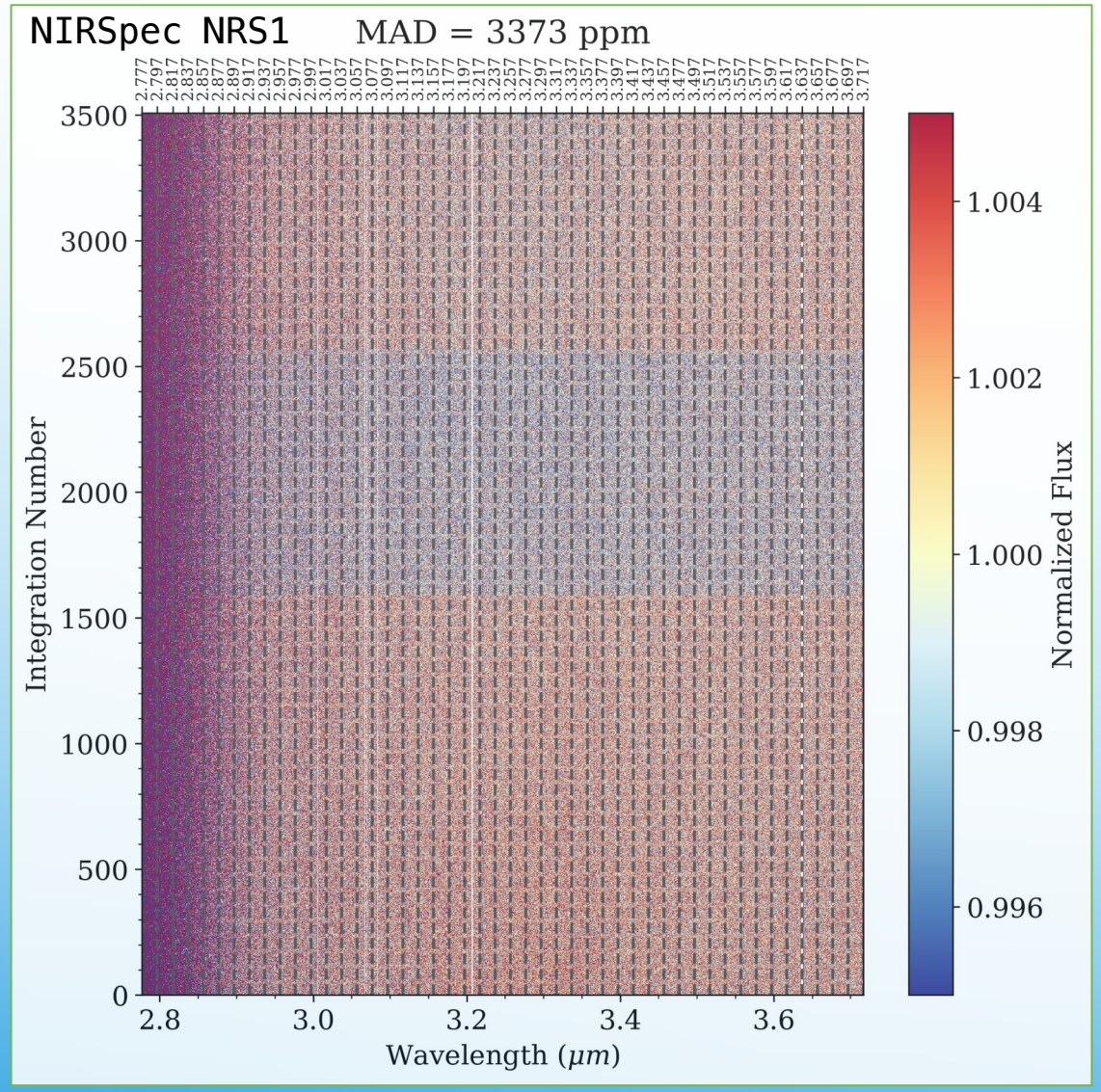
STAGE 4 – USER CONSIDERATIONS

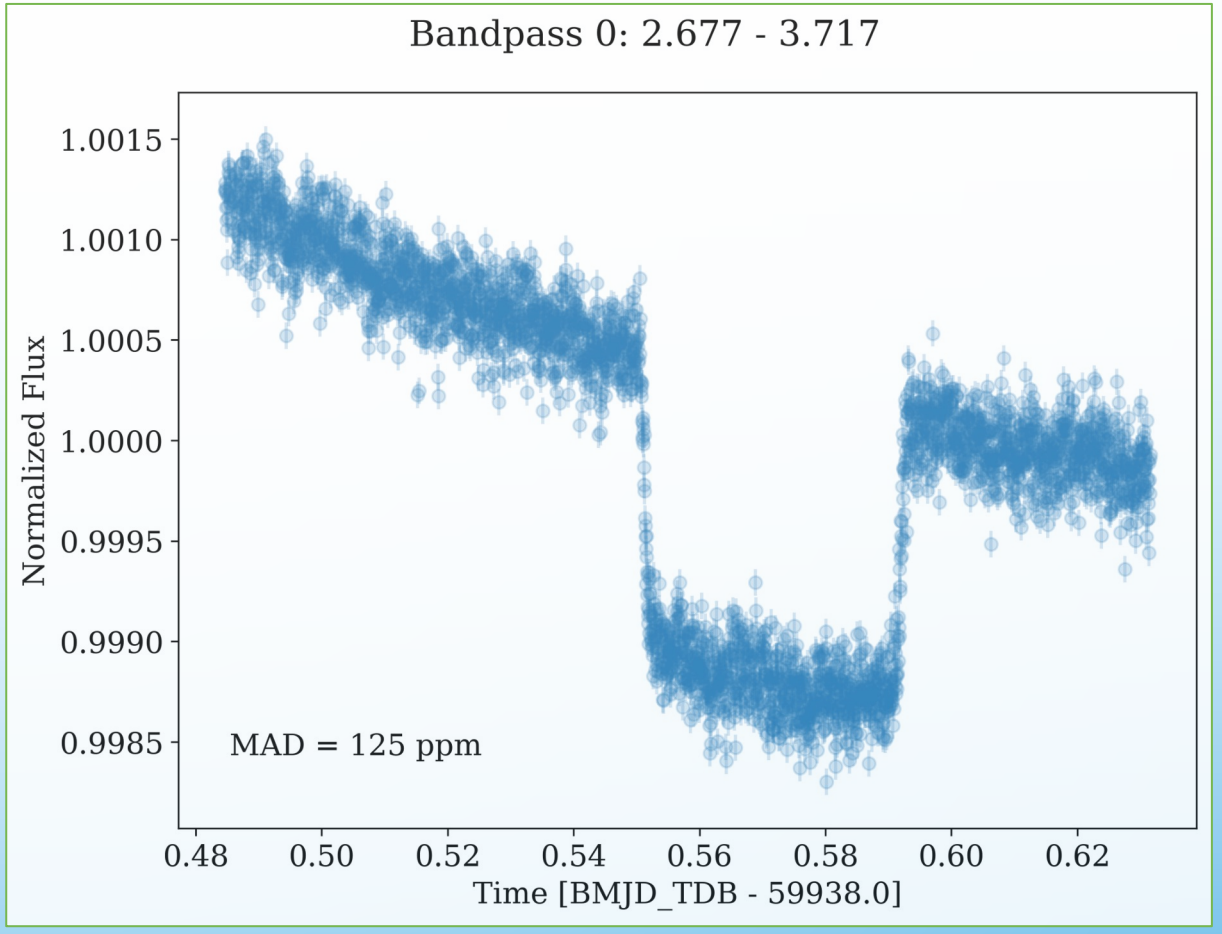
- Number of spectroscopic channels
  - nspecchan
- Wavelength range
  - wave_min
  - wave_max
- Manually mask pixel columns
  - mask_columns
- Light curve sigma clipping
  - clip_binned
- Compute limb-darkening parameters
  - Using ExoTiC-LD
  - compute_ld

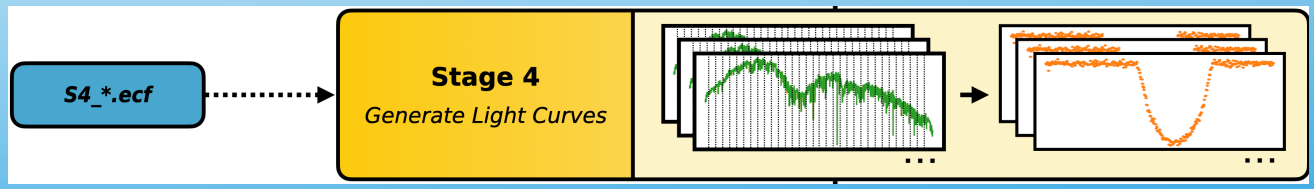




STAGE 4 – LIGHT CURVES








THANK YOU
FOR VISITING
EUREKA