FIFI-LS Data Reduction and Analysis

W. Vacca

FIFI-LS Pipeline Team:

W. Vacca, M. Clarke, J. Holt, D. Fadda, C. Iserlohe, C. Fischer

A Few Preliminaries…

- GO's ***do not*** need to reduce FIFI-LS data
  - SOFIA Data Pipeline Team is responsible for FIFI-LS data reduction and calibration
- GO's ***do*** need to be aware of limitations of pipeline reductions
  - The WVM on-board SOFIA has only recently been brought into service and has not yet been calibrated
- GO's are responsible for all subsequent analysis of FIFI-LS data
  - Remember that FIFI-LS data products are cubes!
  - There are now tools available that allow you to examine and analyze your data
  - The SOFIA team is ready and prepared to help you

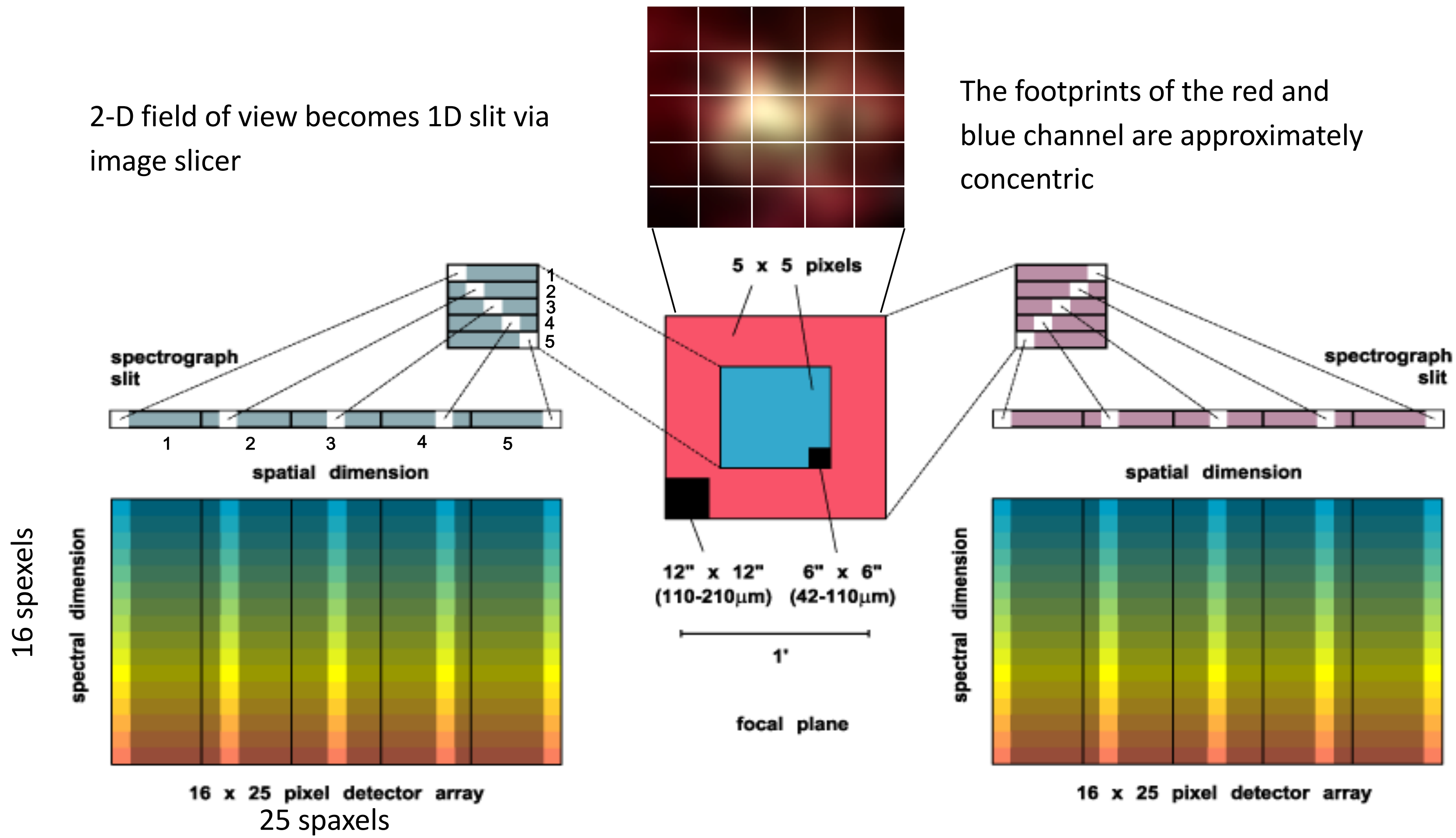
FIFI-LS Pipeline Status Summary

- Current version of FIFI-LS pipeline (v1.5.0) is written in IDL
  - Used extensively during flights to produce ‘Quick-look’ reductions
  - Incorporates all known FIFI-LS observing modes, including ‘Total Power’ and ‘Focus_Loop’
  - Previous version have been used to reduce all FIFI-LS data
  - Incorporates parallel processing (huge increase in speed!)
  - Generates L2, L3, and L4 (multi-mission) data cubes
  - Incorporates nominal telluric corrections using pre-computed ATRAN models at various Alt’s and ZAs
  - Incorporates flux calibration/response curves derived from observations of Mars
- Works within existing DPS infrastructure (REDUX)
  - Allows steps to be performed sequentially by hand or automatically
  - Displays results of each step when run manually

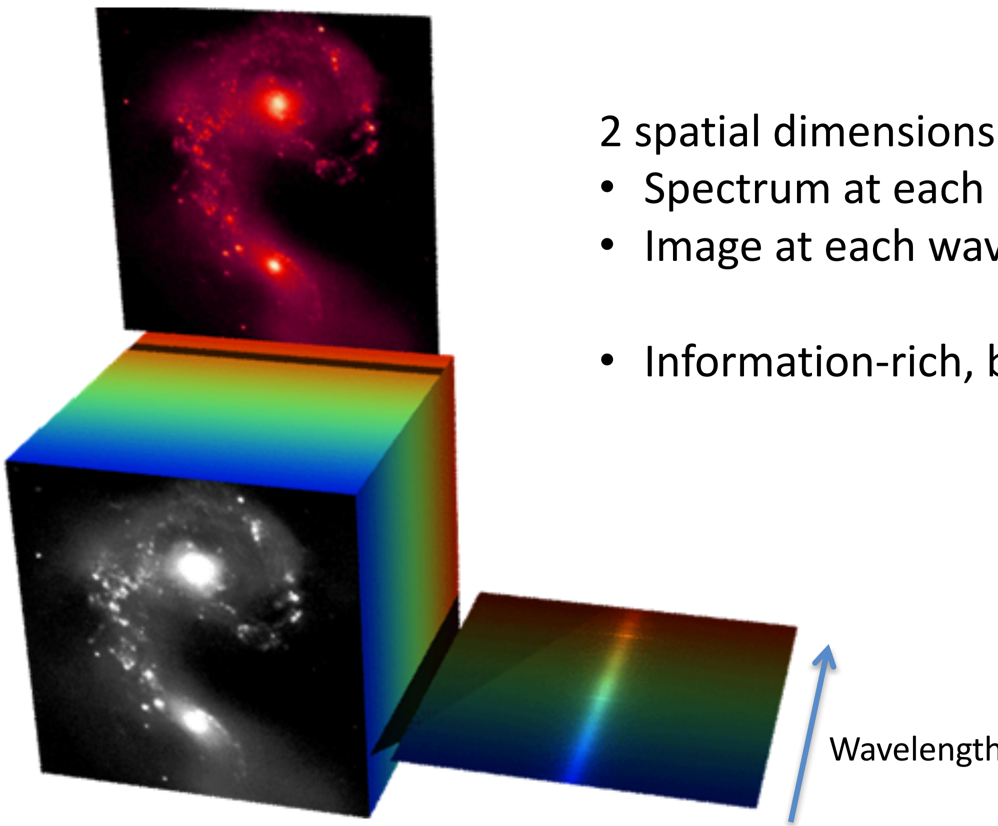
Basic Design of FIFI-LS:

2-D field of view becomes 1D slit via image slicer

The footprints of the red and blue channel are approximately concentric



FIFI-LS is an Integral Field Spectrograph: output file is a 3-D cube

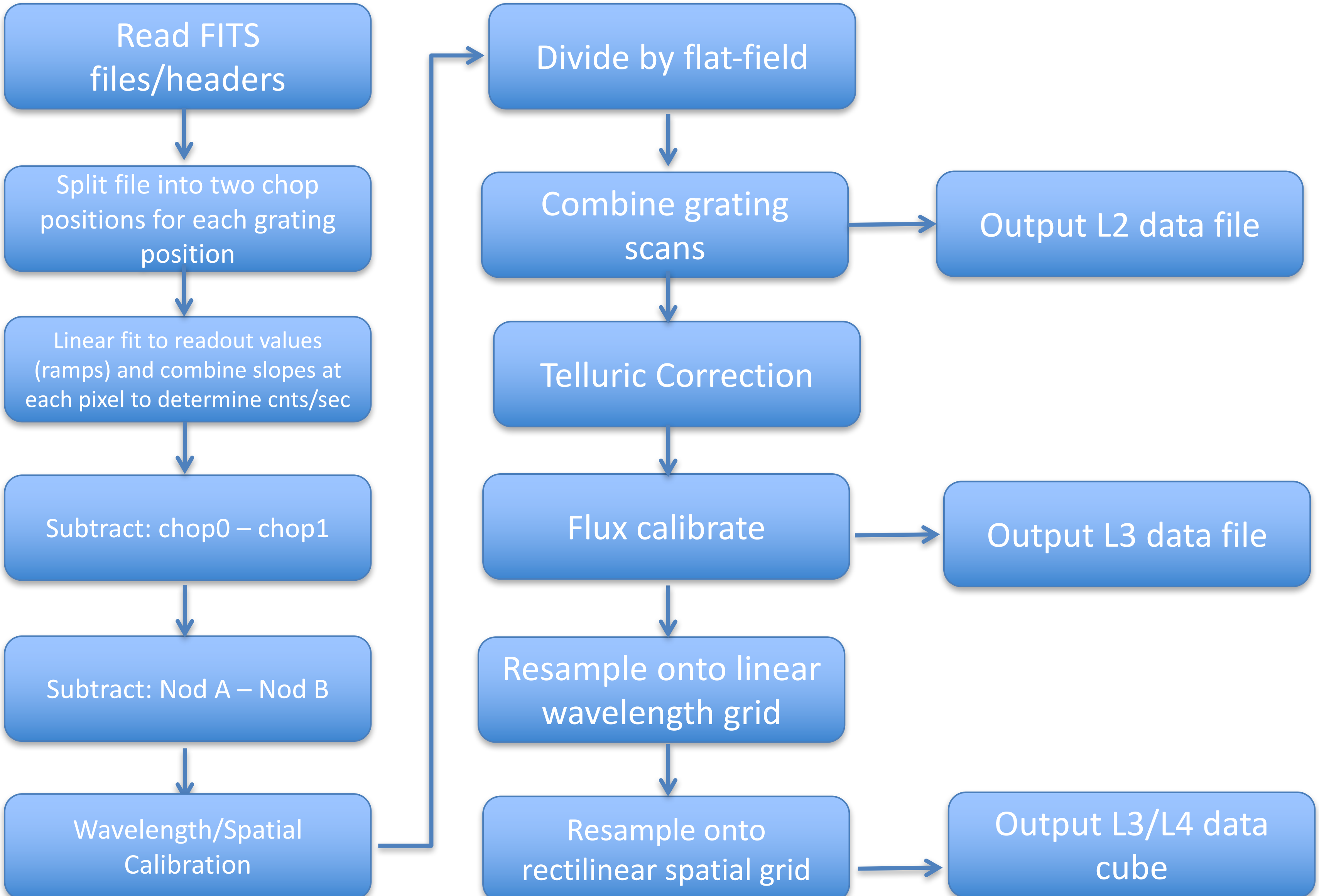


2 spatial dimensions x 1 wavelength dimension:

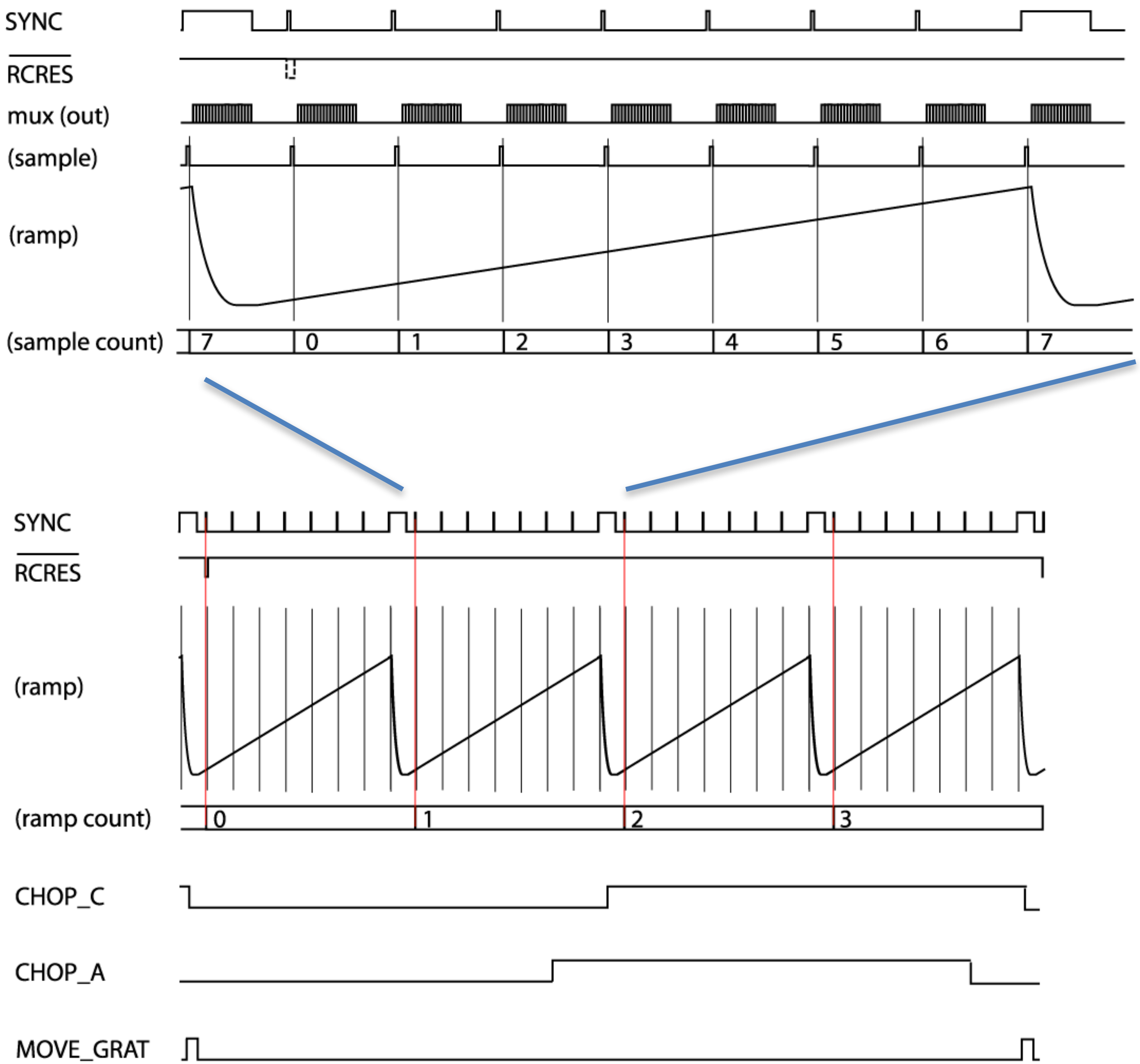
- Spectrum at each spatial pixel (spaxel)
- Image at each wavelength (spexel)

- Information-rich, but a more complex, format

Major FIFI-LS Pipeline Steps:



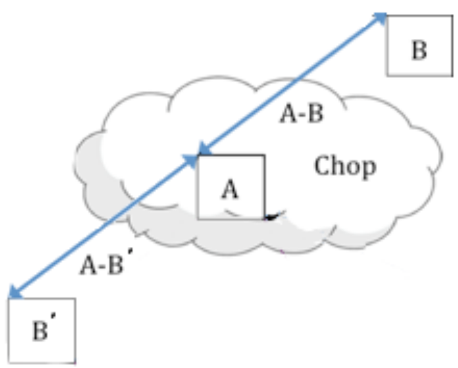
Readouts



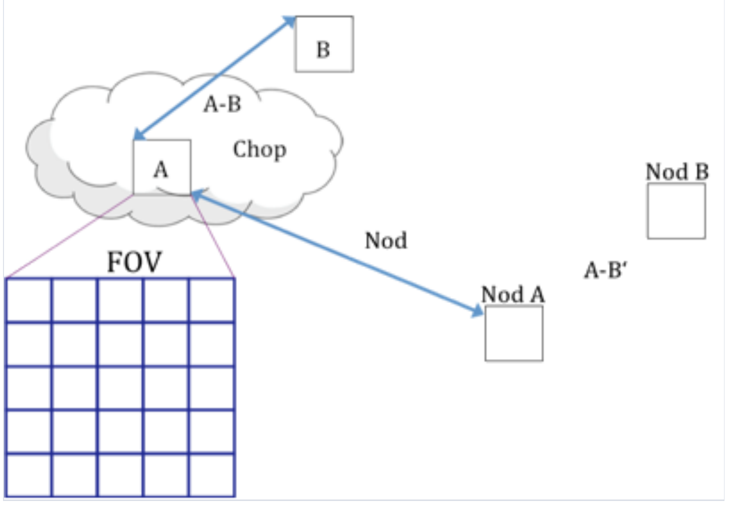
The output file contains a table of each of the readout values for each pixel. The reduction code separates the values for the two chop positions, separates the values for the grating positions, and performs a linear fit to each ramp to determine the count rate (slope = counts/sec) and uncertainty at each pixel

Chopping and Nodding

- Readout every 4 msec; 32 readouts per ramp = 128 msec per ramp
- Chop every ~0.25 sec ⟺2 ramps per chop position
- Step the grating every ~10 sec
- Nod every ~30 sec (dwell time); typically 3-4 grating positions at each nod position
- All the readouts (ramp values) for both chop positions and all grating positions at a single nod position are written into a single file
- Each nod position is a separate file (A or B)
- ~8 sec to move the telescope between nod positions
- Nods are usually ABBA: 3 grating positions at first AB nod pair (~76 sec) and 3 more during the second nod pair (~76 sec)
- This is repeated for each dither/sky position in a spatial map



Symmetric Chop/Nod (NMC)

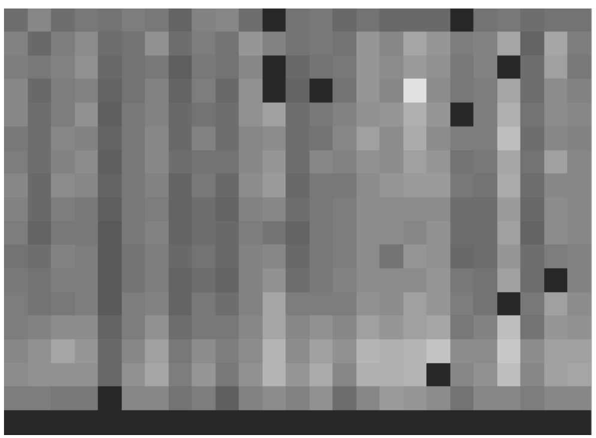


Asymmetric Chop/Nod (C2NC2)

Initial Steps: Fit Ramps, Chop & Nod Subtract

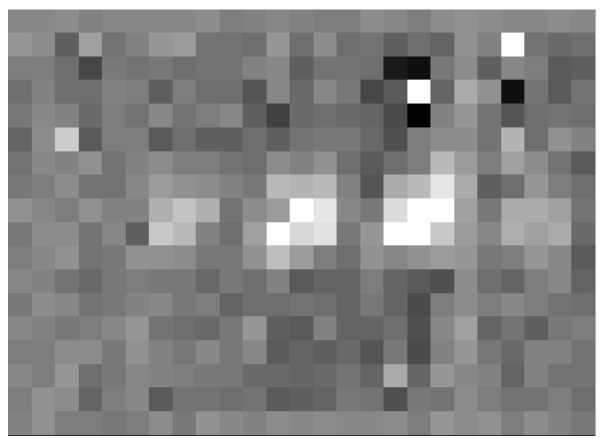
- For each pixel, at each chop position:
  - Exclude readouts at ends of ramps (chop transitions)
  - Exclude saturated values
  - Perform linear fit to each ramp to derive slopes and errors (V/s)
  - Exclude ramps during grating transitions
  - Combine slopes with weighted mean to derive instrumental flux and error
- For each pixel, subtract the values for the two chop positions
- For each pixel, combine the A and B nod values

16 spexels

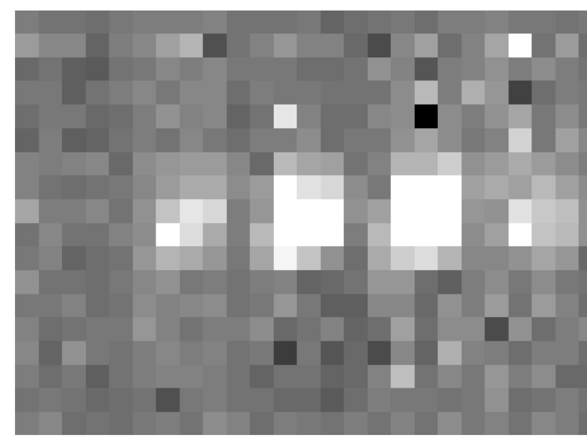


25 spaxels

Chop0



Chop0 – Chop1
Nod A



Nod A – Nod B

Wavelength calibration:

- Each spexel *j* in spaxel *i* has a wavelength $\lambda_{i,j}$ given by:

$$\phi_i = 2\pi\, ISF \frac{ind + ISOFF_i}{2^{24}}$$
$$\delta_j = [j - 8.5] * PS + sign[j - QOFF] * [j - QOFF]^2 * QS$$
$$g_i = g_0 * \cos\left(\tan^{-1}\left(\frac{SlitPos_i - NP}{a}\right)\right)$$
$$\lambda_{ij} = 1000 \frac{g_i}{m}\left[\sin(\phi_i - \gamma) + \sin\left(\phi_i + \gamma + \delta_j\right)\right]$$

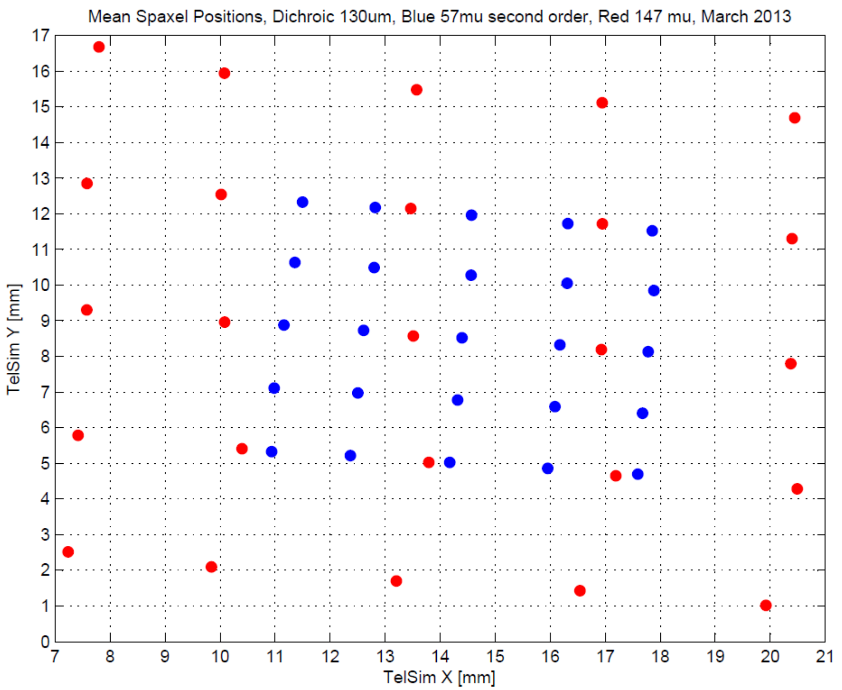
- Each of the 25 spaxels, *i* , has its own (and different) wavelength calibration, $\lambda_{i,j}$, for the 16 spectral pixels, *j*
- Sebastian confirms that wavelength calibration is good to 10-15% of a resolution element
- Each pixel has its own (and different) spectral width

$$d\lambda_{ij}/dp = 1000 \frac{g_i}{m}[PS + 2 * sign[j - QOFF] * (j - QOFF) * QS]\left[\cos\left(\phi_i + \gamma + \delta_j\right)\right]$$

- To conserve flux, when resampling, each point must be scaled by ratio of new spectral (and spatial) width to original spectral (and spatial) width

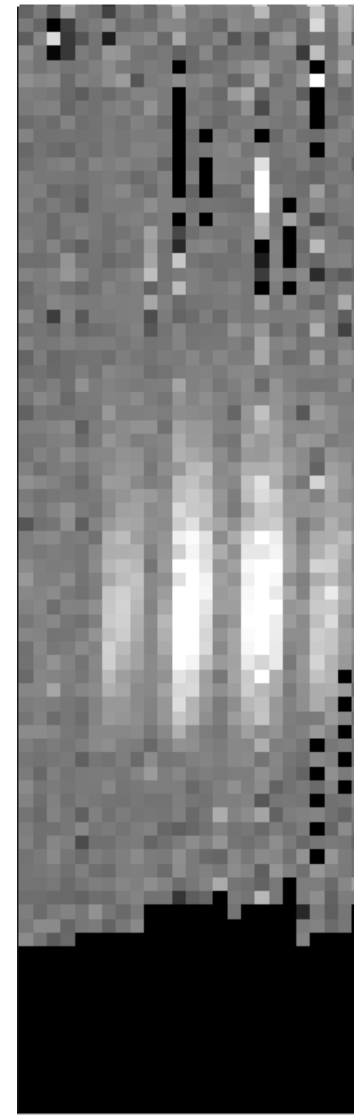
Spatial Calibration

- Spaxels are not on a rectilinear, regular grid
- The relative sky position of each pixel is computed and stored
- WCS for each cube is computed from dither offsets and reference position
- Offsets between red and blue arrays are accounted for
- Final cube is rotated so N up and E left

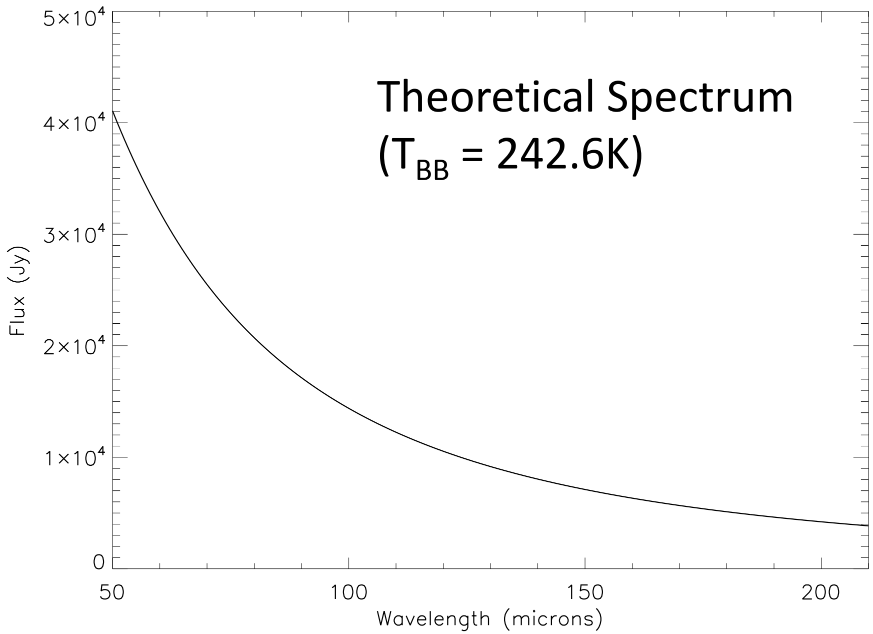


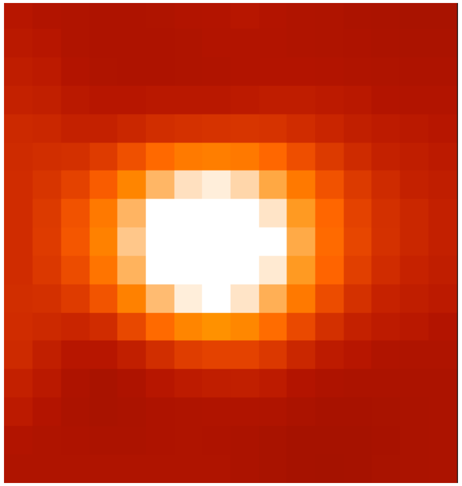
Combine grating scans, apply telluric corrections and flux calibration

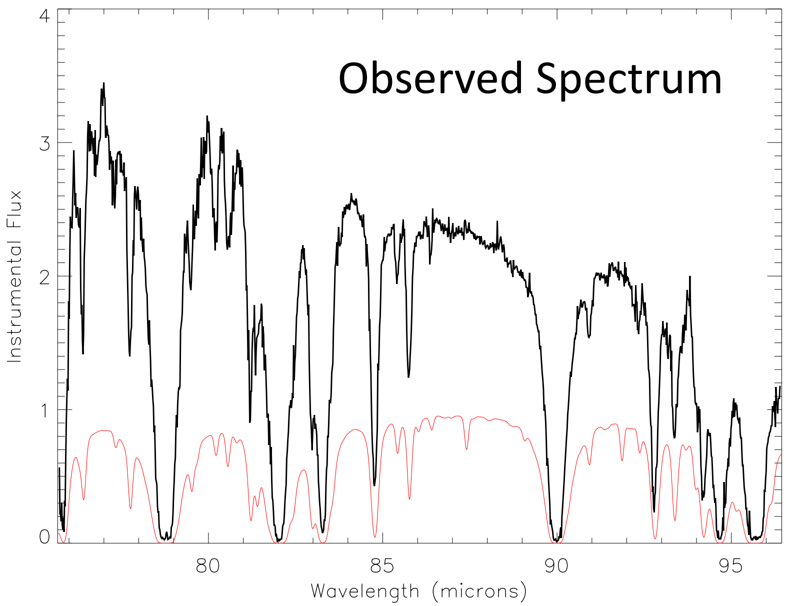
- Combine grating scans:
  - Scale each pixel value to a common spectral width
  - Subtract bias offsets between grating scans
  - Sort data by wavelength
  - Data are still irregularly sampled in wavelength
- Apply telluric correction
  - WVM is still uncalibrated, so no PWV values available
  - Pipeline uses ATRAN models for Alt and ZA of observation
  - Divide observed data by model telluric spectrum, smoothed to resolution, binned, and resampled to the observed data
  - Corrections are reasonable for broad shallow features, but not for:
    - deep sharp features which are extremely sensitive to precise PWV
    - narrow features (e.g., ozone) because $(F^*T_{atm}) \otimes P/(T_{atm} \otimes P) \neq F \otimes P$ unless T_{atm} is smooth
  - Exclude pixels where $T_{atm} < 60\%$
  - Retain uncorrected fluxes in separate cube
- Apply flux calibration
  - Divide by instrumental response curves (V/s/Jy) derived from observations of Mars and Callisto and theoretical models
  - Applied to both telluric-corrected and uncorrected cubes

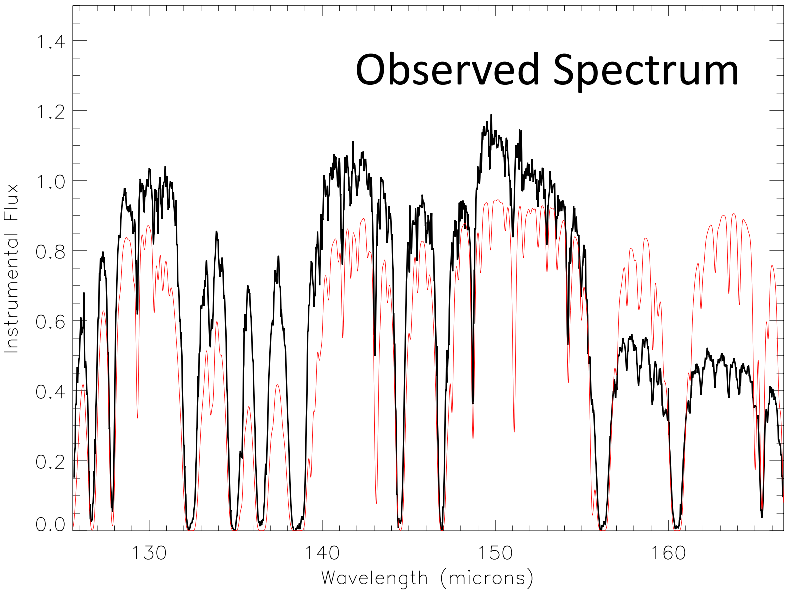


FIFI-LS Flux Calibration: Mars spectrum

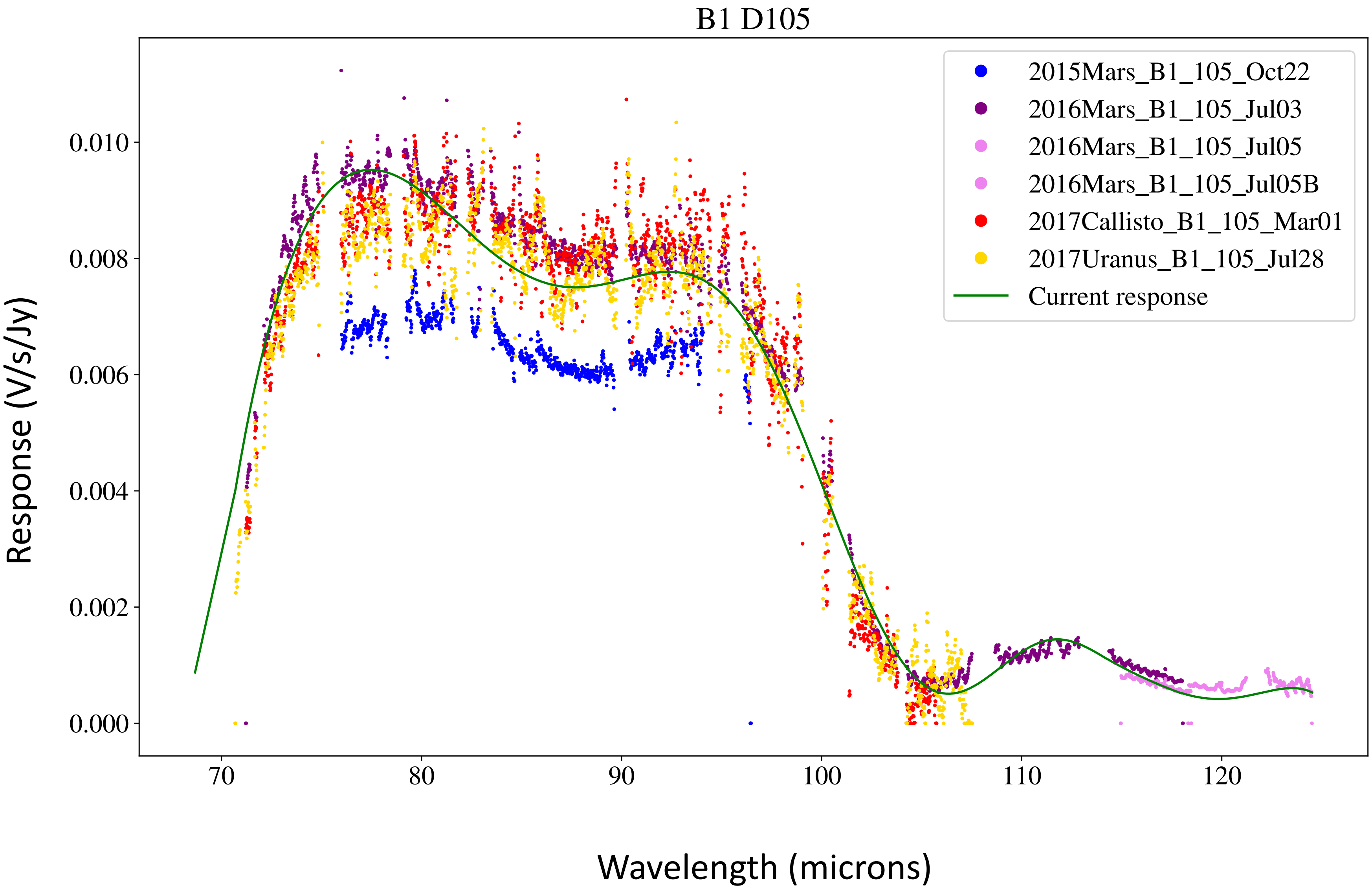






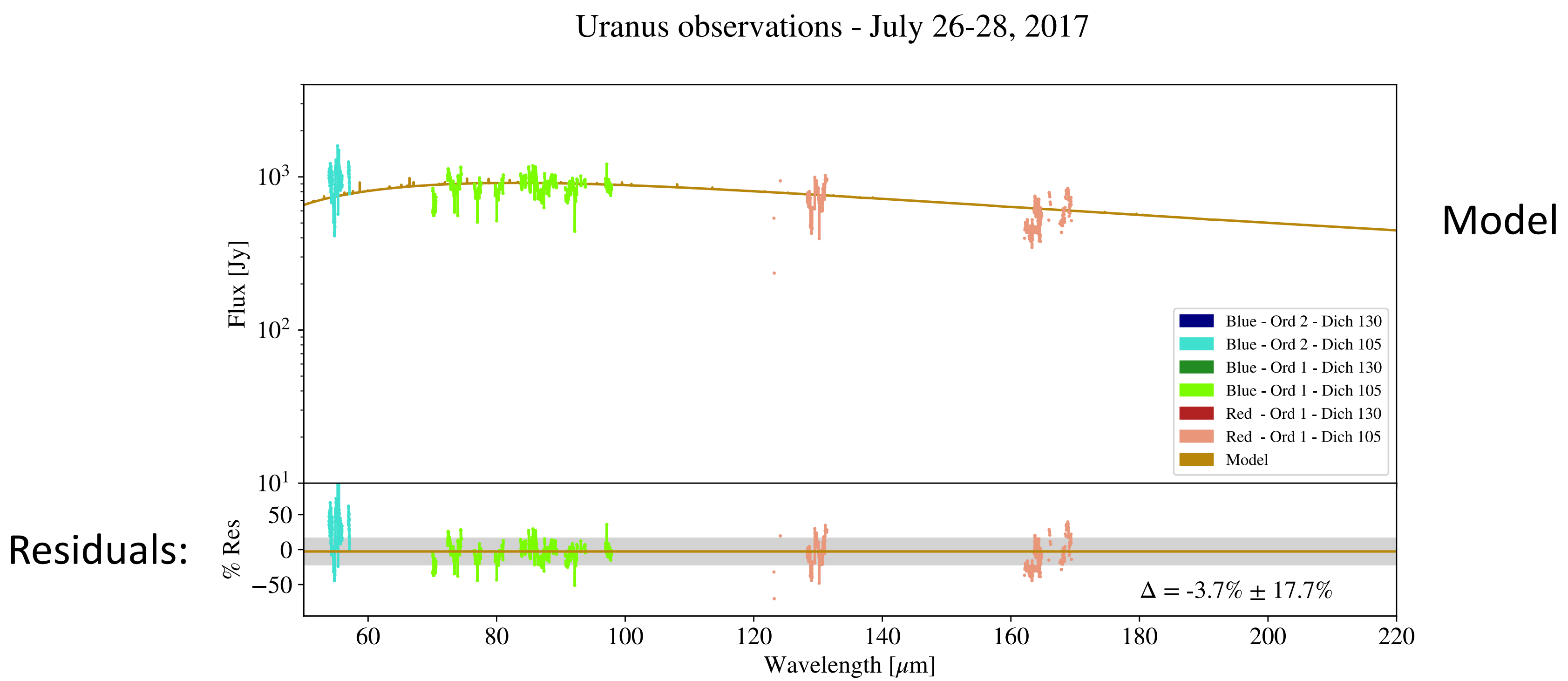


Pipeline Flux Calibration:



Pipeline Flux Calibration:

- Comparisons with Uranus and Callisto indicate:
  - Mean deviations < few% (where T_{atm} > 0.6)
  - RMS deviations < 20% (where T_{atm} > 0.8)
- Comparisons with *Herschel*/PACS indicate agreement to within < 15%



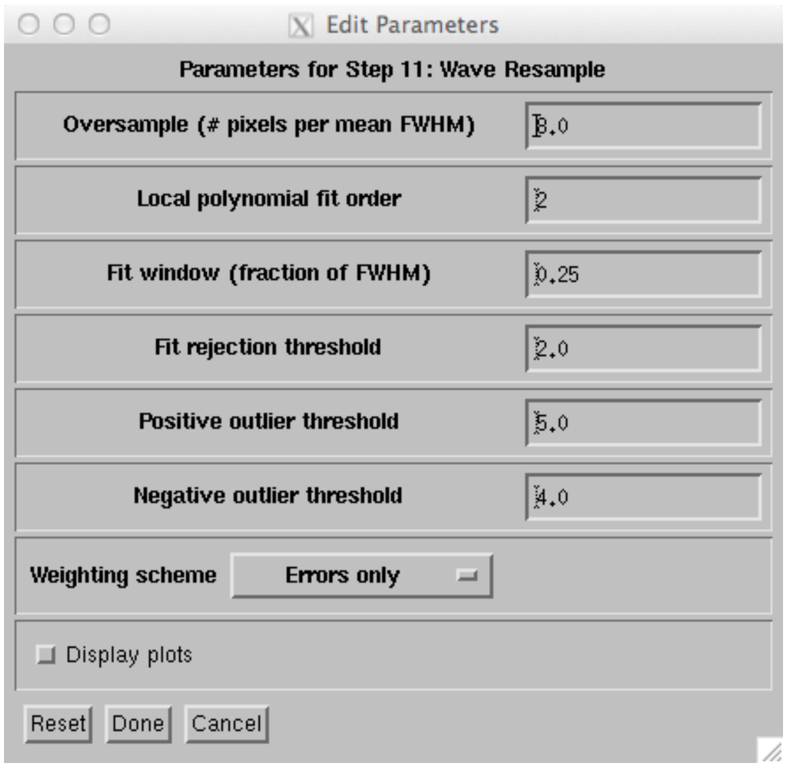
Generating a regularly sampled cube

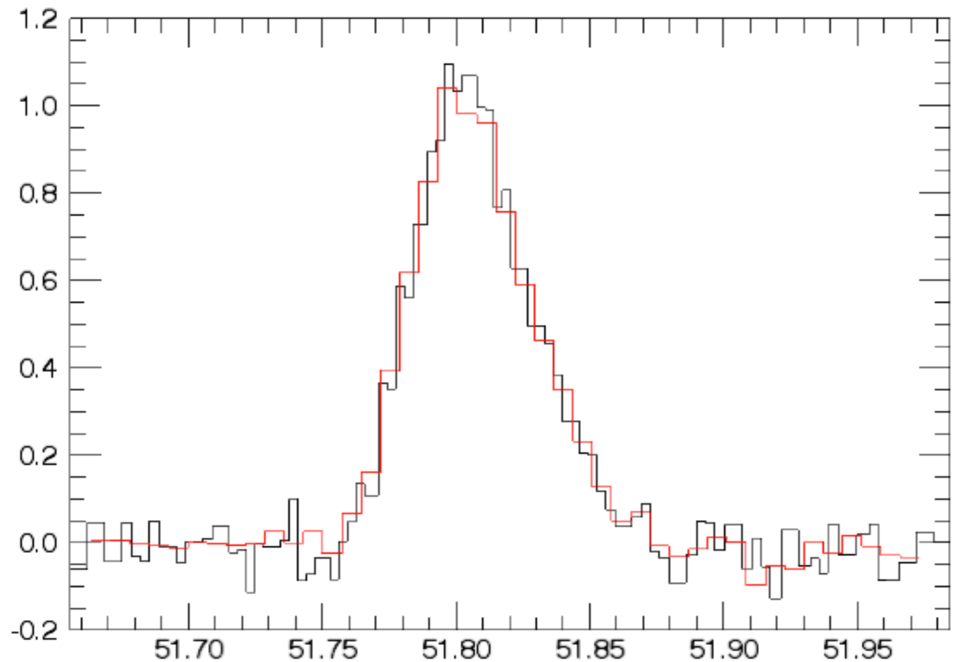
- The combined data set for an observation consists of a cloud of data points, irregularly spaced in both space and wavelength
- Ideally, one would resample once, in 3 dimensions (space and wavelength) simultaneously – but this is technically difficult
- The pipeline resamples first in wavelength and then in space

Wavelength Resampling

- Determine absolute λ_{min} and λ_{max} of the observations, from all grating positions for all spaxels
- Determine sampling $\Delta\lambda$ from mean λ and R (or FWHM) and n_{samp} number of points per FWHM ($\Delta\lambda$=FWHM/n_{samp}; $n_{samp} \sim 8$)
- Establish regular grid of wavelength points from λ_{min}, λ_{max} and $\Delta\lambda$
- At each grid point, for each spaxel, determine all *n* flux points within *f* x FWHM ($f \sim 0.25$)
- Fit an Nth (N ~ 2) order 1-D polynomial to the *n* flux points, excluding local outliers (> s σ; $s \sim 5$), with weights from errors
- Exclude 'bad' points as determined by the fit
- Flux and error at grid point determined from fit
- Repeat for all wavelength points and all spaxels
- Now all spaxels have the same wavelength grid

Wavelength Resampling



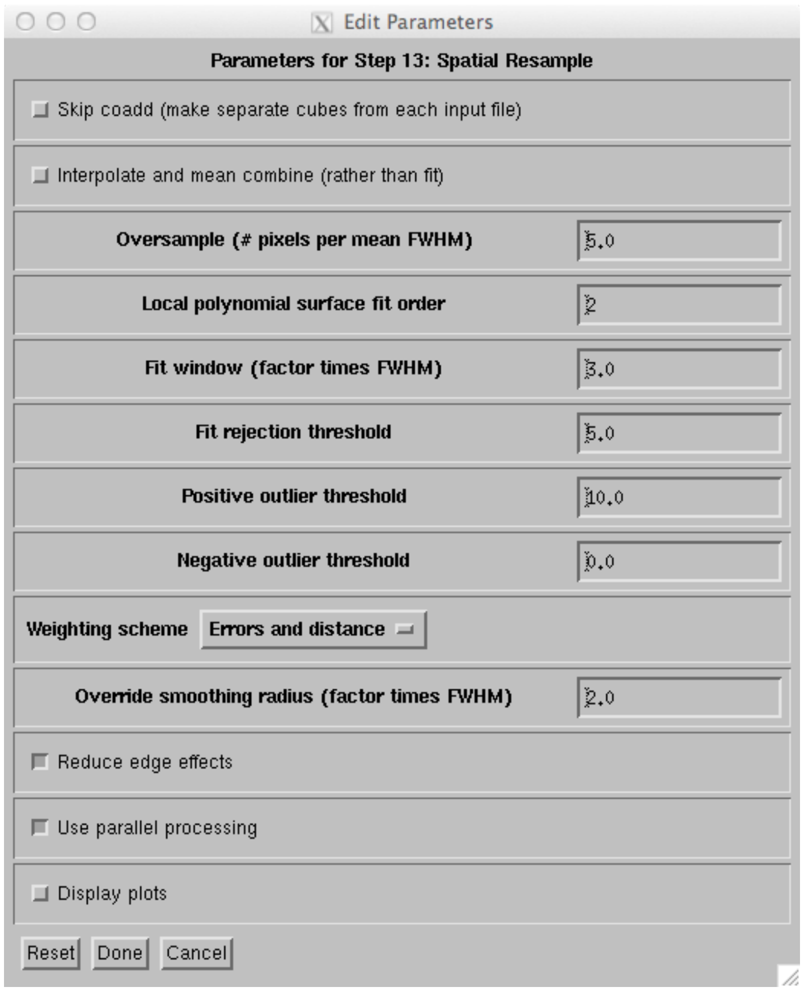


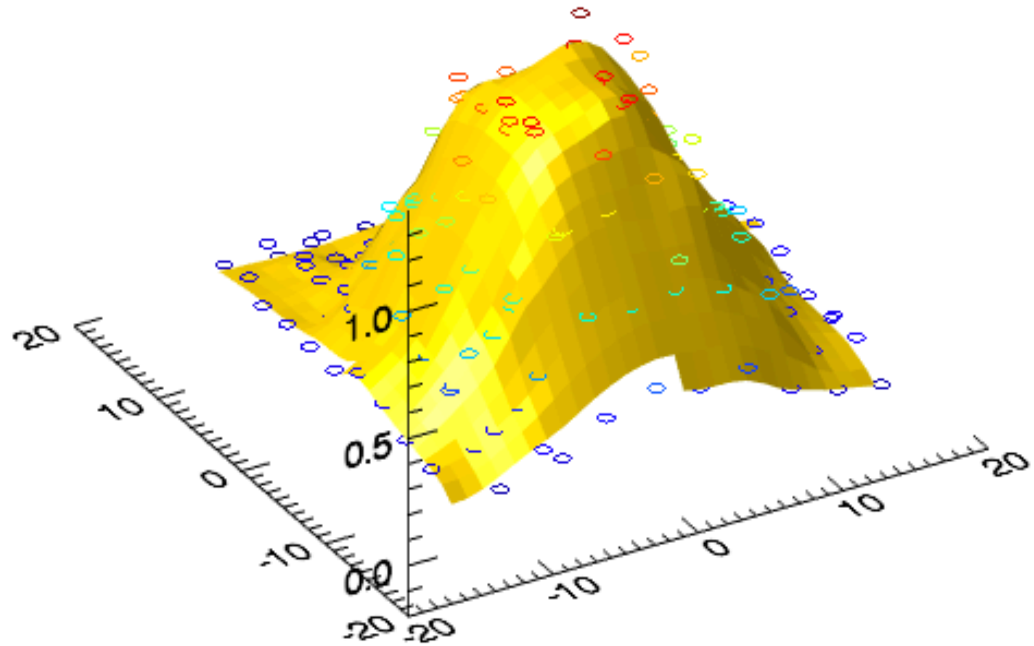
Black = input fluxes on irregular λ grid
Red = resampled fluxes on regular λ grid

Spatial Resampling

- Determine absolute spatial positions of the observations, from all dither/offset positions and all spaxels
- Determine sampling Δx from mean λ and R (or FWHM) and n_{samp} number of points per FWHM ($\Delta x = FWHM/n_{samp}$)
- Set (re-)sampling to 1” for blue array and 2” for red array
- Establish regular grid of spatial points (pixels)
- At each spatial grid point on each λ plane, determine all *n* flux points within *f* x FWHM (*f* ~ 3)
- Scale fluxes by ratio of area of new pixels to area of original spaxels to preserve flux
- Fit an Nth (N ~ 2) order 2-D polynomial to the *n* flux points, excluding local outliers (> *s* σ, *s* ~ 10), with weights derived from errors and distance from grid point
- Exclude ‘bad’ points as determined by the fit
- Flux and error at grid point determined from fit
- Repeat at all spatial points on each wavelength plane
- Now all data are on a regular wavelength and spatial grid

Spatial Resampling





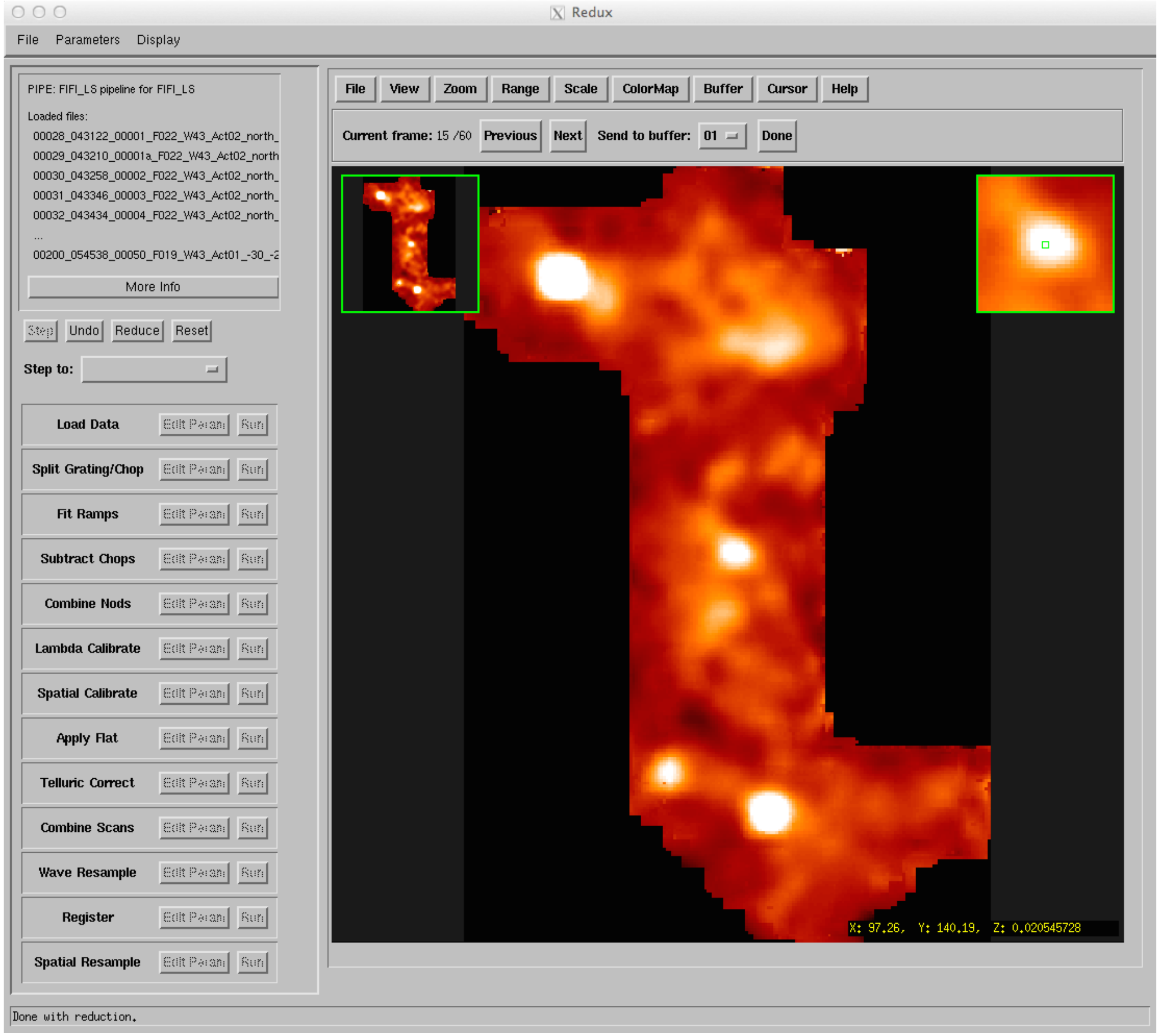
Circles = input fluxes on irregular spatial grid
Yellow = resampled fluxes on regular grid

Why ‘Drizzle’ Isn’t Optimal:

- “The flux in any given output spaxel, at each wavelength…is the sum of all contributions from the input spaxels…” (PACS manual)
- For Drizzle to work well, it needs
  - a regular grid of uniform detectors
  - a PSF highly oversampled in all dimensions
  - ‘perfect’ data, since flux values are being re-distributed
- None of these apply to FIFI-LS data!
- Our philosophy: each spectral and spatial sample point is an independent measurement of the physical flux distribution. We attempt to determine that true flux distribution directly by fitting surfaces in all dimensions

Pipeline Examples: W43 at 158 μm

Reduction steps



Data Formats and Output files:

- The pipeline produces a number of intermediate (L2) and final (L3/L4) products by default, for ease of re-processing, although the output at any stage in the reductions can be saved.

- SCM (L2) file, produced after combining scans into an unevenly spaced grid, with 1 extension
- CAL (L3) file, produced after flux calibration, with 1 extension
- WGR (L3) file, produced after wavelength re-sampling onto a regular wavelength grid, with 9 extensions (same as WXY files below, minus the exposure map)
- **WXY** (L3/L4) file, produced after spatial resampling onto a regular spatial grid, with 10 extensions:
  - FLUX: n_x x n_y x n_w cube of telluric-corrected, calibrated fluxes
  - ERR:OR n_x x n_y x n_w cube of errors on the fluxes
  - UNCORRECTED_FLUX: n_x x n_y x n_w cube of uncorrected, calibrated fluxes
  - UNCORRECTED_ERROR: n_x x n_y x n_w cube of errors on the uncorrected fluxes
  - WAVELENGTH: n_w array of wavelengths for each plane in the cube
  - X: n_x array of coordinate offsets from the reference position
  - Y: n_y array of coordinate offsets from the reference position
  - TRANSMISSION: n_w array of ‘nominal’ atmospheric transmission values
  - RESPONSE: n_w array of response values
  - EXPOSURE_MAP: n_x x n_y x n_w cube giving the number of exposures at each position and wavelength
- WXY Cubes can be viewed with DS9, FLUXER, or SOSPEX

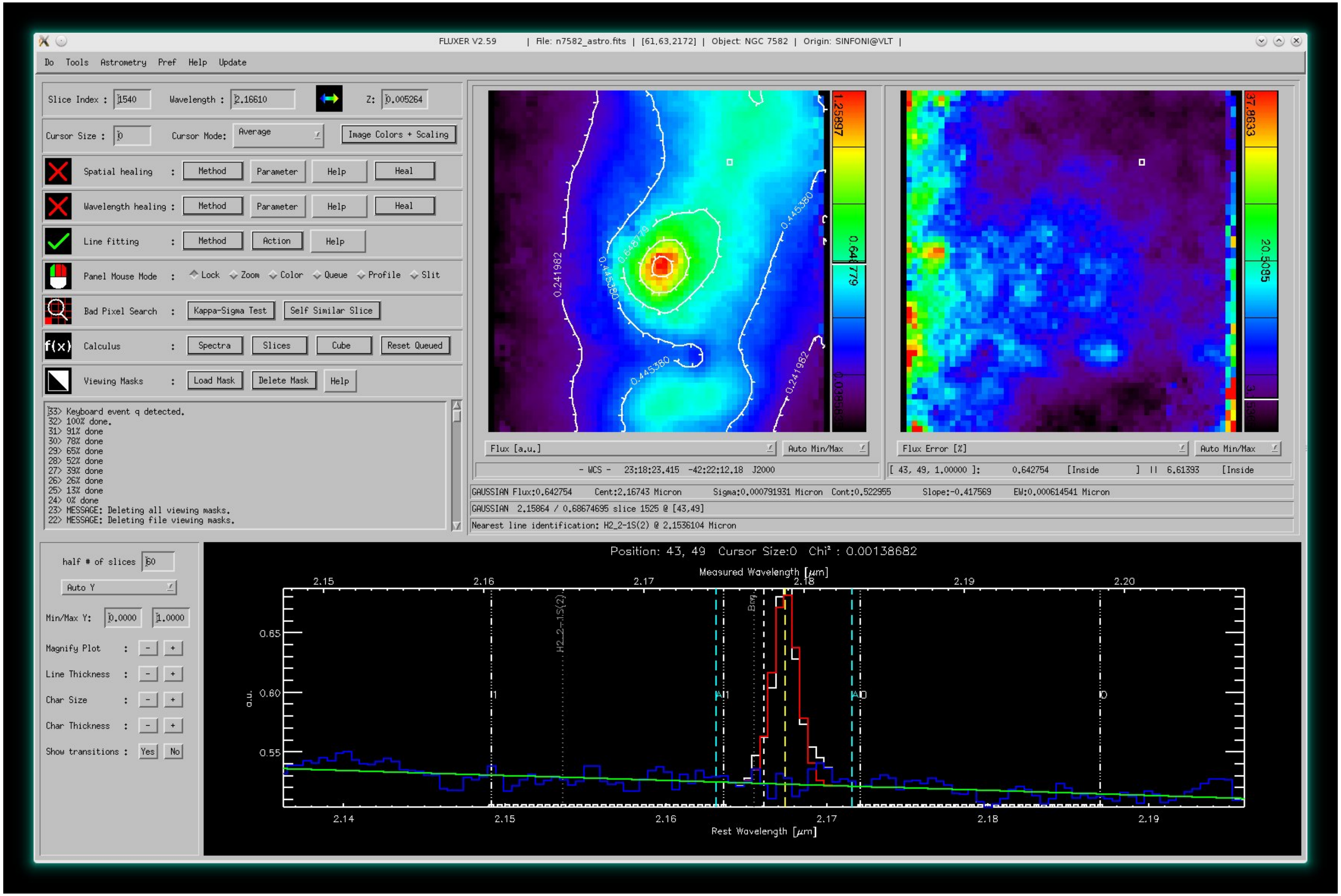
Now what?

- At this point, the data ***reduction*** is finished and the GO (you!) needs to perform the data ***analysis*** in order to produce scientific results
- Two software packages will help you examine the final data cubes, measure line fluxes, etc.
  - **FLUXER**:
    - Written in IDL by C. Iserlohe
    - http://www.ciserlohe.de/fluxer/fluxer.html
  - **SOSPEX**:
    - Written in Python by D. Fadda
    - https://github.com/darioflute/sospex

Fluxer (v2.72)

- IDL 8.3 code for Macs, Linux, and Windows
- Works with data cubes in FITS files
- Functionalities include:
- Visualization of cubes in 2-D (at each wavelength) and navigation through cubes (spatial slices)
- Extraction of spectra at various positions and in apertures (spectral slices)
- Fitting of continua and emission lines to generate line flux maps and velocity fields
- Generation and overplotting of ATRAN spectra
- Overplotting line IDs

Fluxer


FLUXER V2.59 | File: n7582_astro.fits | [61,63,2172] | Object: NGC 7582 | Origin: SINFONI@VLT |
Do Tools Astrometry Pref Help Update
Slice Index : 1540 Wavelength : 2.16610 Z: 0.005264
Cursor Size : 0 Cursor Mode: Average Image Colors + Scaling
Spatial healing : Method Parameter Help Heal
Wavelength healing : Method Parameter Help Heal
Line fitting : Method Action Help
Panel Mouse Mode : Lock Zoom Color Queue Profile Slit
Bad Pixel Search : Kappa-Sigma Test Self Similar Slice
Calculus : Spectra Slices Cube Reset Queued
Viewing Masks : Load Mask Delete Mask Help
33> Keyboard event q detected.
32> 100% done.
31> 91% done
30> 78% done
29> 65% done
28> 52% done
27> 39% done
26> 26% done
25> 13% done
24> 0% done
23> MESSAGE: Deleting all viewing masks.
22> MESSAGE: Deleting file viewing masks.
Flux [a.u.] Auto Min/Max
Flux Error [%] Auto Min/Max
- WCS - 23:18:23.415 -42:22:12.18 J2000
[43, 49, 1.00000]: 0.642754 [Inside] || 6.61393 [Inside
GAUSSIAN Flux:0.642754 Cent:2.16743 Micron Sigma:0.000791931 Micron Cont:0.522955 Slope:-0.417569 EW:0.000614541 Micron
GAUSSIAN 2.15864 / 0.68674695 slice 1525 @ [43,49]
Nearest line identification: H2_2-1S(2) @ 2.1536104 Micron
half # of slices 60
Auto Y
Min/Max Y: 0.0000 1.0000
Magnify Plot : - +
Line Thickness : - +
Char Size : - +
Char Thickness : - +
Show transitions : Yes No
Position: 43, 49 Cursor Size:0 Chi² : 0.00138682
Measured Wavelength [μm]
Rest Wavelength [μm]
a.u.

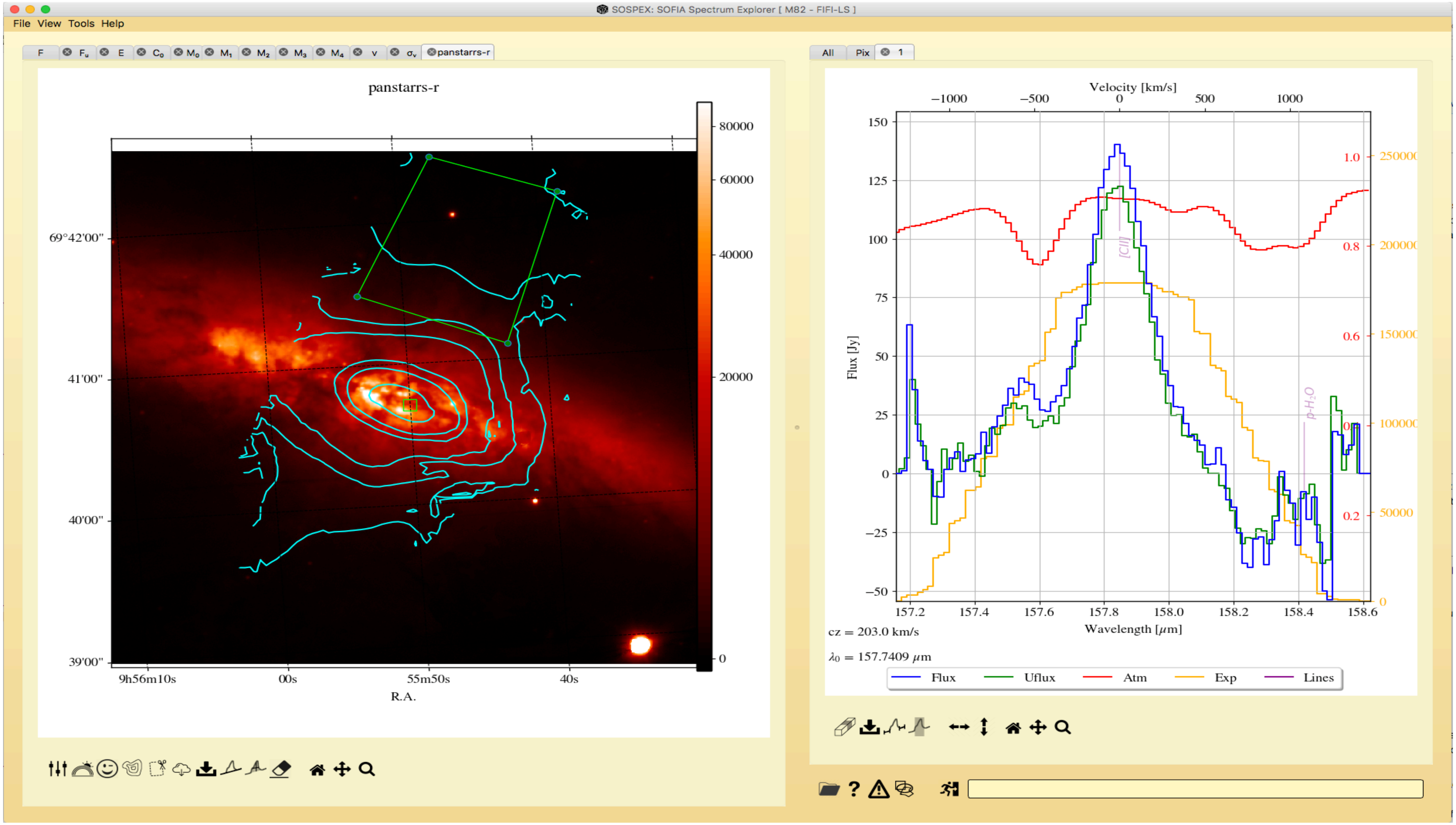
SOSPEX (SOFIA Spectral Explorer)

- Python 3 software, tested on Linux and Mac
- Directly installable with Anaconda:

  conda install –c darioflute sospex

- User-friendly interface with online help (? icon)
- Easy submission of issues through github (! icon)
- Displays FIFI-LS, GREAT, and PACS simple cubes.
- Navigates cube planes and spectra through tabs
- Allows cube manipulations (cut/crop)
- Computes continuum and moments across cubes
- Extracts flux in custom apertures
- Export/import defined apertures
- Overlaps contours on other images

SOSPEX (SOfia SPectral EXplorer)



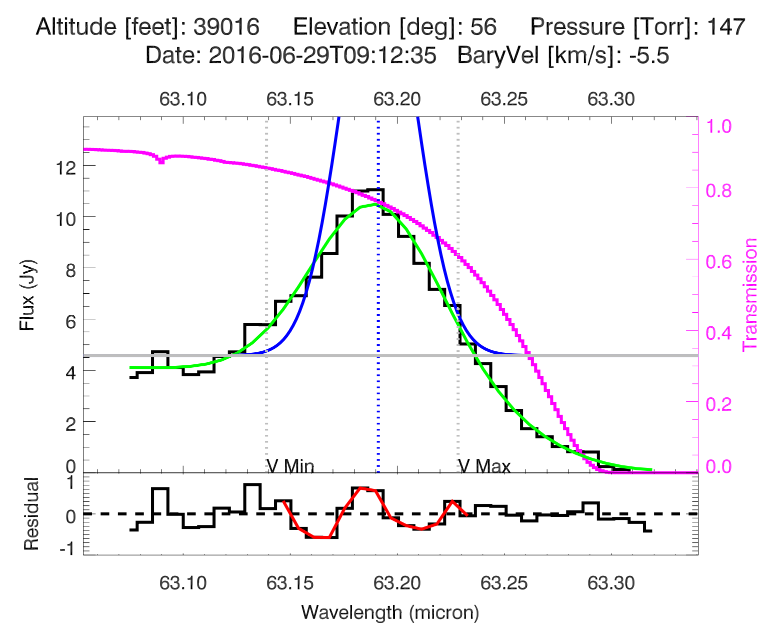
M82 FIFI-LS contours over a Panstarr-r' image. The spectrum from the outflow region inside the green polygon is shown on the right panel. The spectral panel shows flux, exposure, and atmospheric absorption, line names as well as the redshift and the reference wavelength.

Dealing with Telluric Absorption

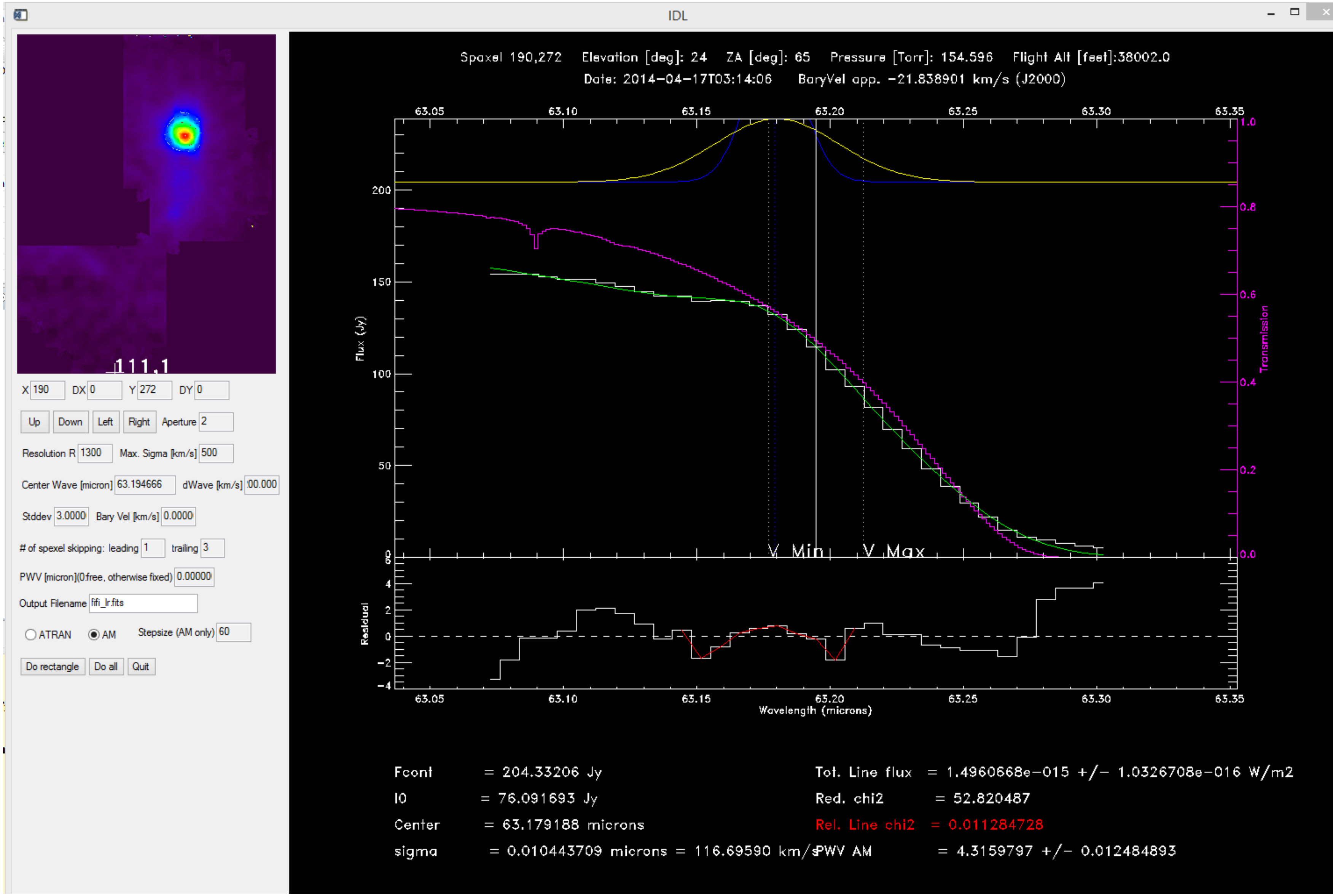
- Due to lack of a calibrated WVM, the pipeline uses pre-computed ATRAN telluric spectra appropriate for the altitude Alt and zenith angle ZA at the time of the observations
  - Based on a standard model atmosphere, with fixed zenith PWV values at each Alt
  - For much of FIFI-LS wavelength range, these spectra are perfectly adequate
  - Pipeline divides the observed spectrum by the model telluric spectrum
- However, at some wavelengths (e.g., 63 microns), where telluric absorption is very strong, true correction becomes very sensitive to precise PWV value
  - In these cases, the division can produce nonsensical results
  - Therefore, the pipeline blanks out regions where the std telluric transmission is below 0.6
- Under many circumstances, however, it is possible to recover the true spectrum (as long as the transmission > 0) using the *un-corrected*, calibrated data cube and the technique described on the next slide
- An IDL package (**fifi_recover**) has been developed by C. Iserlohe and W. Vacca that implements this method but has not been widely distributed because it requires a large library of ATRAN spectra, or a proprietary version of the AM models developed by the GREAT team
  - Used very successfully to measure the [O I] 63 µm line flux from the GC

Dealing with Telluric Absorption

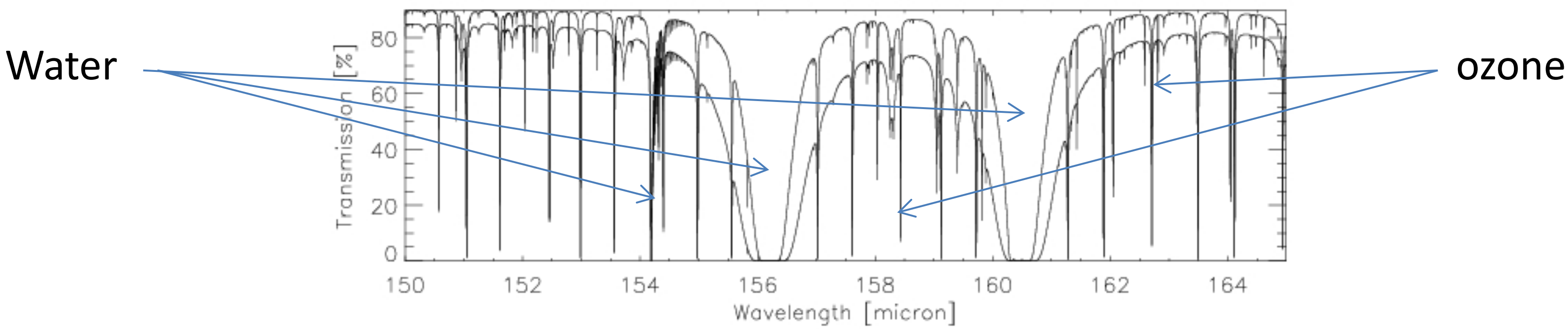
- Model the observed spectrum F_λ^{obs} as:
- $F_\lambda^{mod} = [(G(\lambda; I_0, \lambda_0, \sigma) + C(\lambda)) \cdot T_{atm}(\lambda; Alt, ZA, PWV)] \otimes P_{inst}(R)$
- Gaussian emission line
- Continuum
- Telluric absorption
- Instrumental response
- Compute χ^2 to find best-fit I_0 and PWV:
- $\chi^2 = \sum_\lambda \frac{\left(F_\lambda^{obs} - F_\lambda^{mod}\right)^2}{\sigma_\lambda^2}$
- Works only if spectrum has:
  - Obvious continuum
  - High S/N
  - Telluric feature, but $T_{atm} > 0$



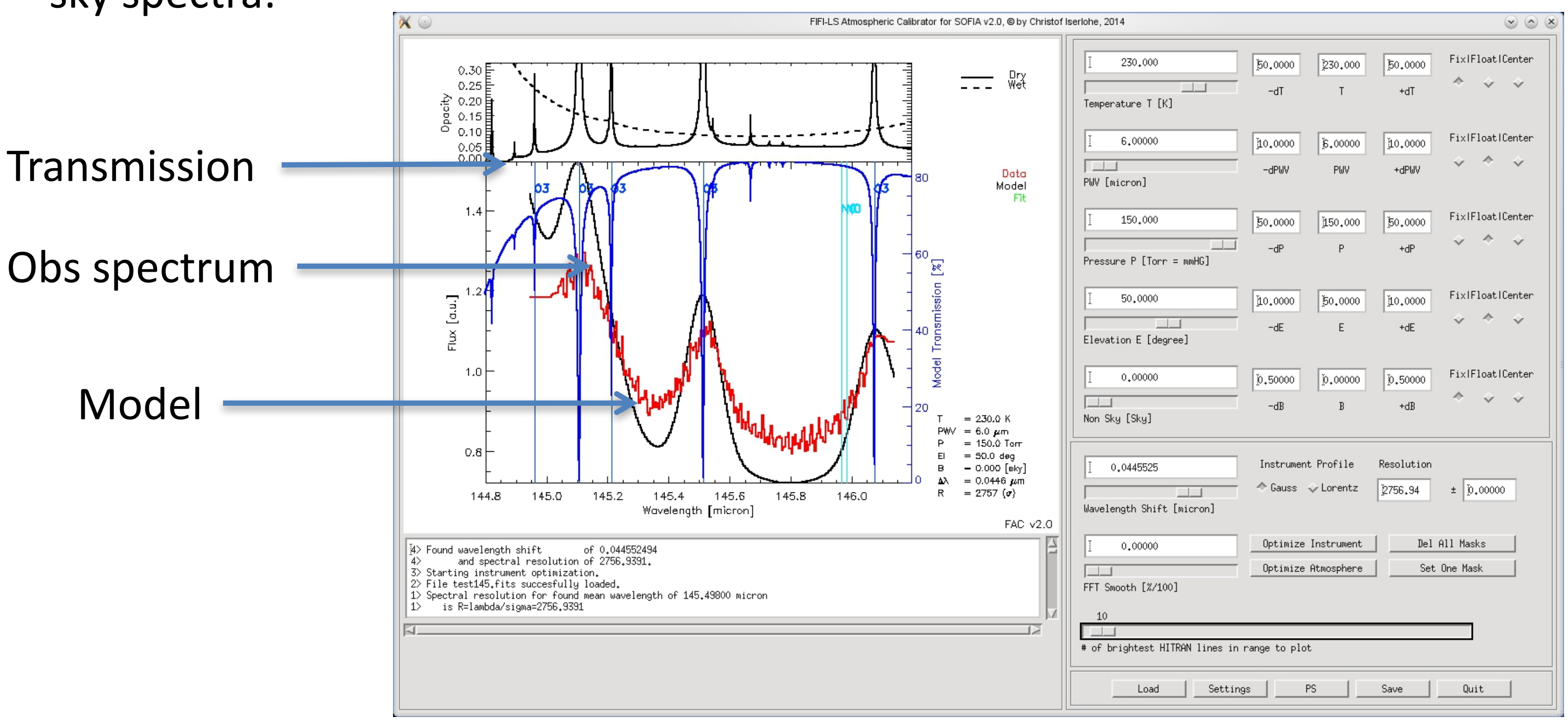
Dealing with Telluric Absorption



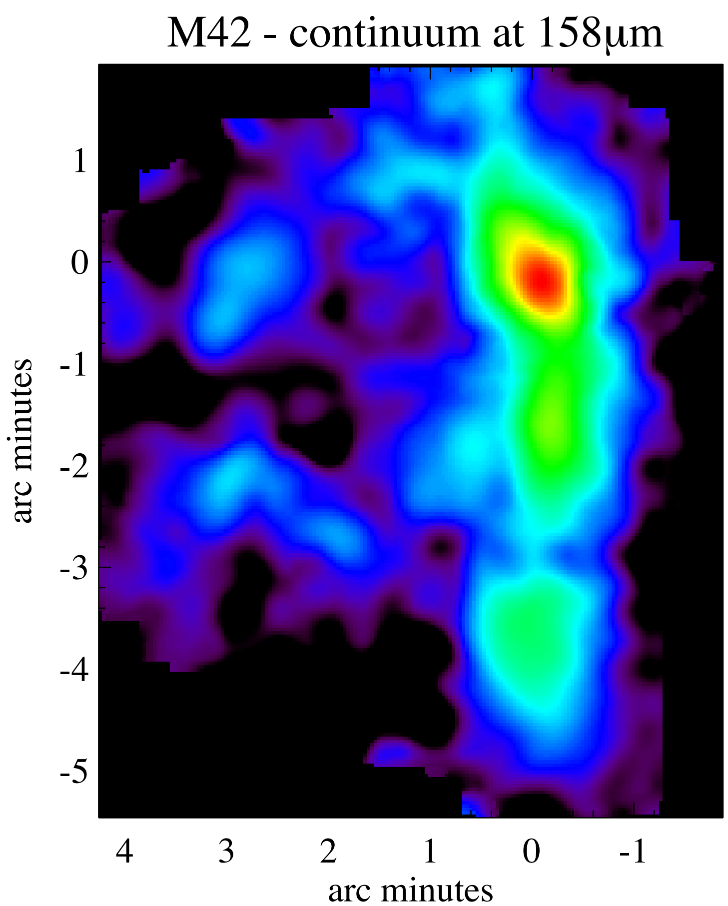
Telluric Absorption

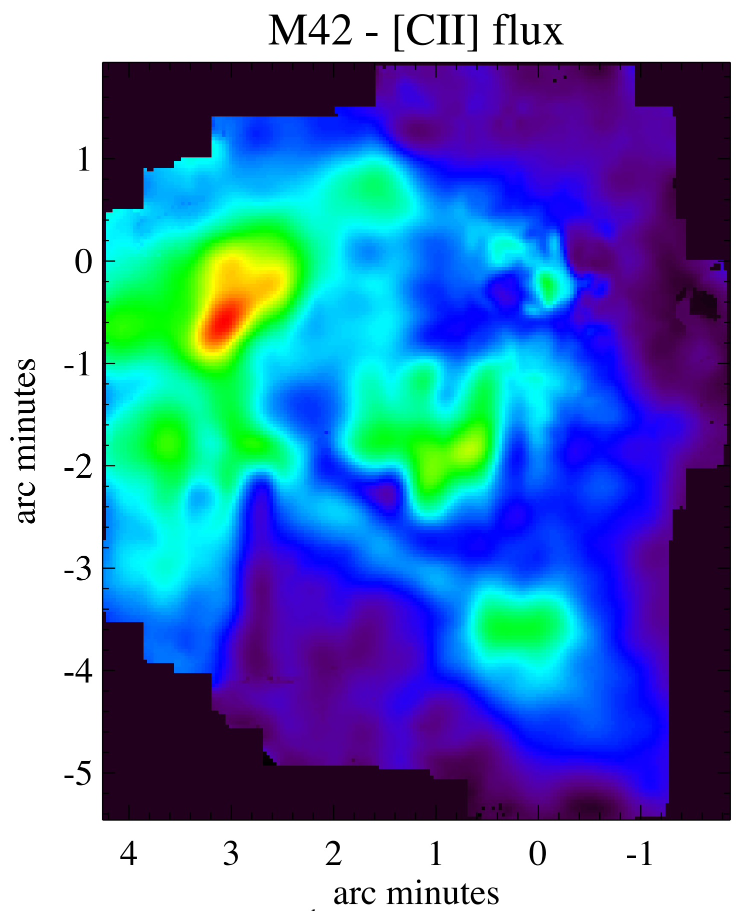


FAC (FIFI-LS Atmospheric Calibrator) by C. Iserlohe tries to fit AM models to observed sky spectra:



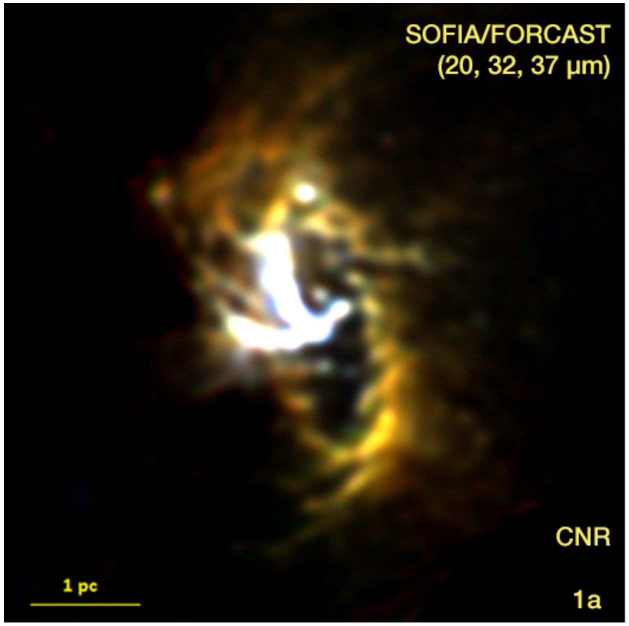
Pipeline Examples: Orion



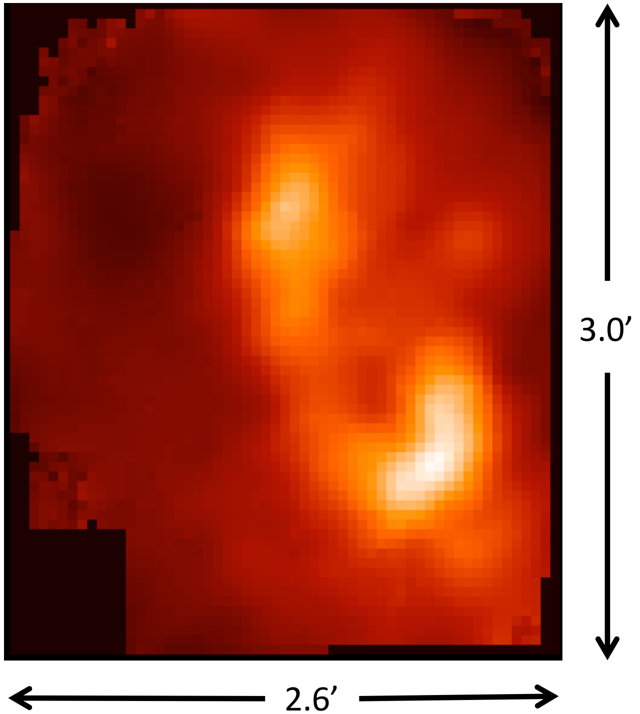


Pipeline Examples: Mapping the CNR in the GC

FORCAST Observations of the CNR
(Lau et al. 2013)

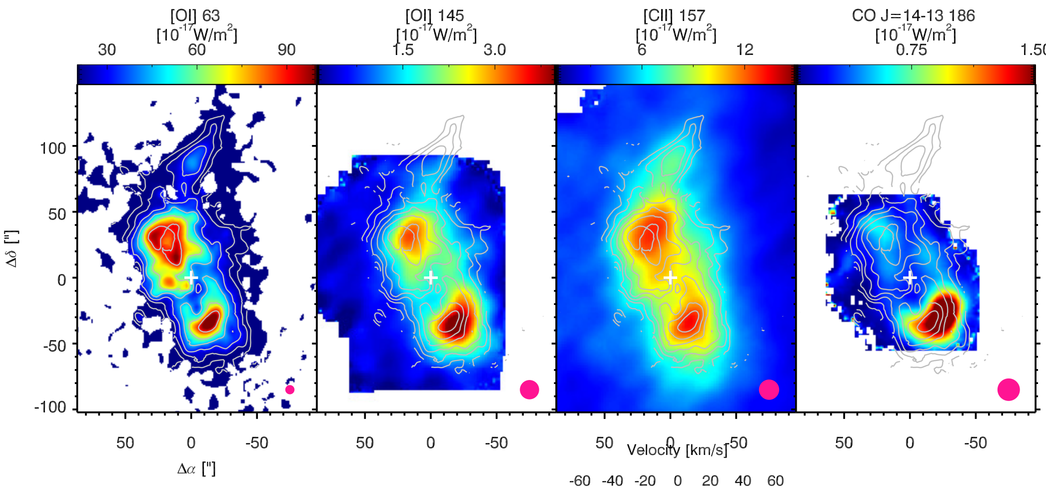


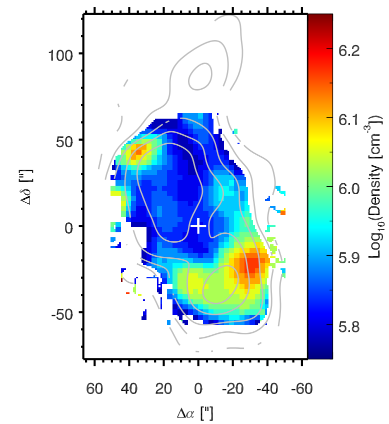
FIFI-LS Observations of the CNR at 145 μm
(Iserlohe et al. 2018)



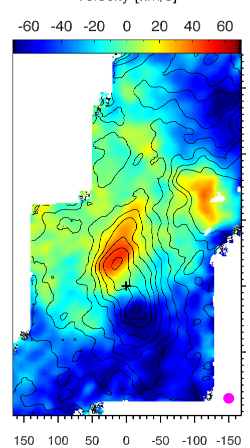
Pipeline Examples: CNR Mapping

Line Fluxes

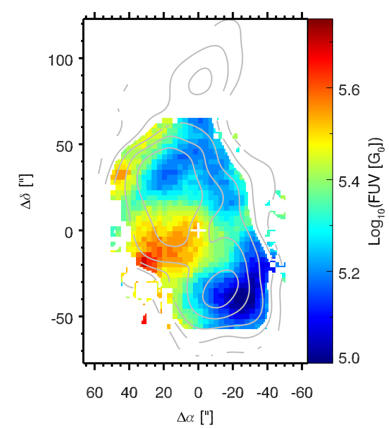




Density



Velocity Map



FUV Flux

Iserlohe et al. (2018)

Pipeline Summary

- IDL pipeline for the reduction of FIFI-LS data has been successfully developed and implemented. It automatically performs the following reduction steps:
  - Ramp fitting
  - Bad data detection and exclusion
  - Error propagation
  - Chop/Nod demodulation
  - Wavelength and spatial calibration of individual spaxels
  - Flat-fielding
  - Scan combining
  - Telluric correction
  - Flux calibration
  - Barycenter velocity corrections
  - Wavelength re-sampling
  - Spatial resampling
  - WCS computation
  - Data cube/map generation
- Software packages have been developed and are available to allow GOs to examine and analyze the data cubes produced by the pipeline.
- Dealing with telluric absorption is still somewhat problematic (and may always be, even with a calibrated WVM), but a technique/package has been developed that can used to recover the intrinsic line fluxes in many cases
- Contact the pipeline team for questions about, and assistance with interpreting, your data