FIFI - LS Redux Pipeline Users Manual

SOF-US-HBK-OP10-2007

Date: July 13, 2017
Revision: C

AFRC
Armstrong Flight Research Center
Edwards, CA 93523

ARC
Ames Research Center
Moffett Field, CA 94035

German Space Agency, DLR
Deutsches Zentrum für Luft und
Raumfahrt

FIFI-LS Redux Pipeline Users Manual

SOF-US-HBK-OP10-2007

AUTHOR:

Melanie Clarke, USRA, SOFIA DPS Technical Lead	Date

William Vacca, USRA, SOFIA Assoc. Director for Science Data Systems	Date

APPROVAL:

William Vacca, USRA, SOFIA Assoc. Director for Science Data Systems	Date

Roberta Leftwich-Vann, USRA, Program Manager	Date

REVISION HISTORY

REV	DATE	DESCRIPTION
-	12/18/15	Initial release, with contributions from M. Clarke, W. Vacca, J. Holt, K. Nishikida, R. Klein
A	4/19/16	Various updates resulting from the following SPRs for the FIFI-LS Redux v1.2.0 release: PIPEDEV-254, PIPEDEV-253, PIPEDEV-252, PIPEDEV-250, PIPEDEV-249, PIPEDEV-248.
B	7/29/16	Various updates resulting from the following SPRs for the FIFI-LS Redux v1.3.0 and v1.3.1 release: PIPEDEV-259, PIPEDEV-260, PIPEDEV-271, PIPEDEV-272.
C	7/13/17	Minor updates for FIFI-LS Redux v1.3.2, v1.3.3, and v1.4.0 releases: PIPEDEV-278, PIPEDEV-279, PIPEDEV-288, PIPEDEV-289, and PIPEDEV-290.

FIFI-LS Redux Users Manual
SOF-US-HBK-OP10-2007

Table of Contents

FIFI-LS Redux Users Manual
SOF-US-HBK-OP10-2007

1 INTRODUCTION

The SI Pipeline Users Manual (OP10) is intended for use by both SOFIA Science Center staff during routine data processing and analysis, and also as a reference for General Investigators (GIs) and archive users to understand how the data in which they are interested was processed. This manual is intended to provide all the needed information to execute the SI Level 2 and Level 3 Pipeline, and assess the data quality of the resulting products. It will also provide a description of the algorithms used by the pipeline and both the final and intermediate data products.

A description of the current pipeline capabilities, testing results, known issues, and installation procedures are documented in the SI Pipeline Software Version Description Document (SVDD, SW06, DOCREF). The overall Verification and Validation (V&V) approach can be found in the Data Processing System V&V Plan (SV01-2232). Both documents can be obtained from the SOFIA document library in Windchill at location: / Software Management Development or Verification / Pipelines (DPS).

This manual applies to FIFI-LS Redux version 1.4.0.

2 SI OBSERVING MODES SUPPORTED

2.1 FIFI-LS instrument information

FIFI-LS has two separate and independent grating spectrometers with common fore-optics feeding two large Ge:Ga detector arrays (16 x 25 pixels each). The wavelength ranges of the two spectral channels are 42 – 110 microns and 110 – 210 microns, referred to as the BLUE and RED channels, respectively.

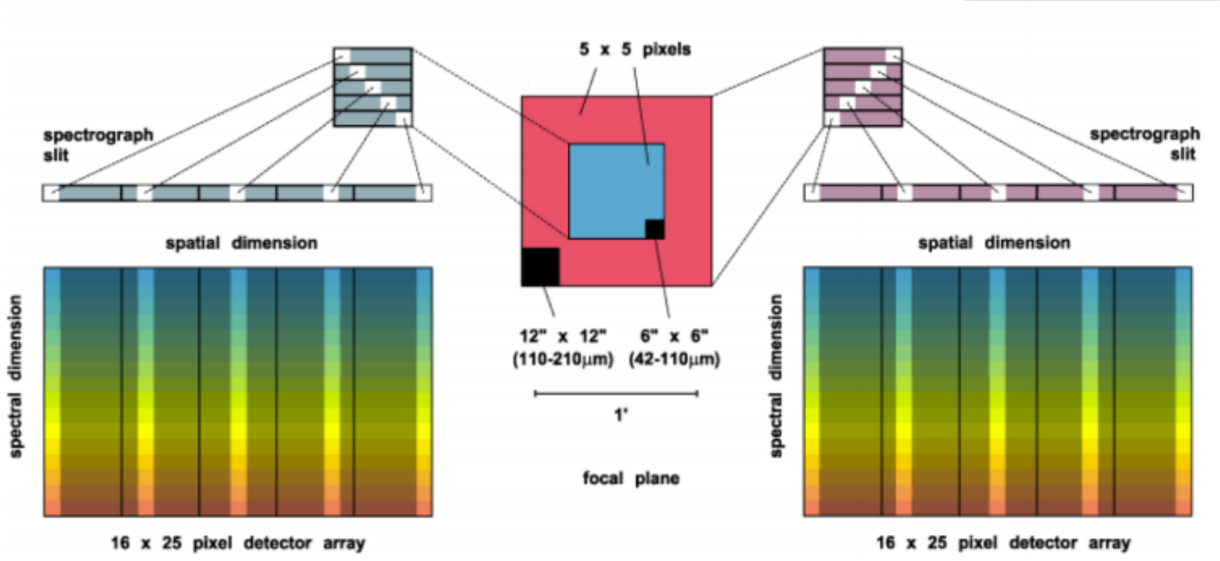
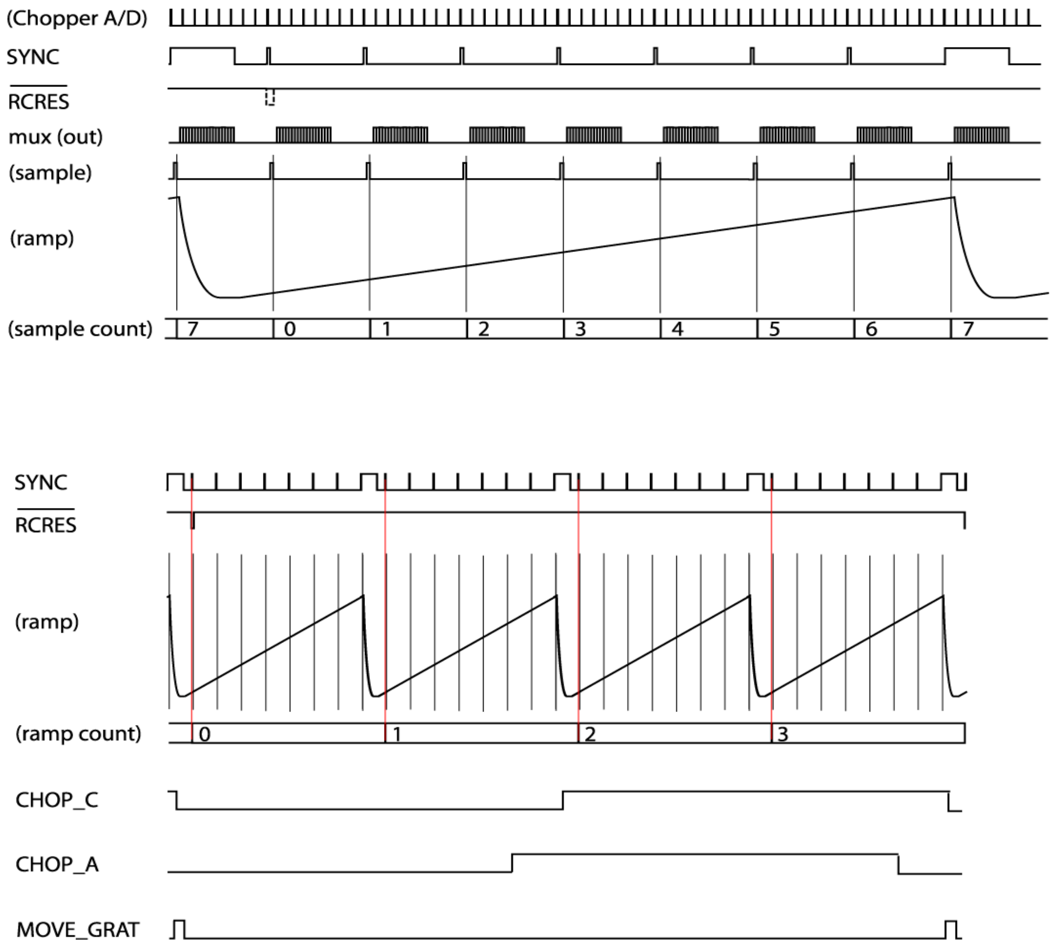


Figure 1: The integral field unit for each channel.

Multiplexing takes place both spectrally and spatially. An image slicer redistributes 5 x 5 pixel spatial fields-of-view (approximately diffraction-limited in each wave band) along the 1 x 25 pixel entrance slits of the spectrometers. Anamorphic collimator mirrors help keep the spectrometer compact in the cross-dispersion direction. The spectrally dispersed images of the slits are anamorphically projected onto the detector arrays, to independently match spectral and spatial resolution to detector size, thus enabling instantaneous coverage over a velocity range of ~ 1500 to 3000 km/s around selected FIR spectral lines, for each of the 25 spatial pixels ("spaxels").

The detectors are read out with integrating amplifiers: at each pixel a current proportional to the incident flux charges a capacitor. The resulting voltage is sampled at about 256Hz. After a certain number of read-outs (the ramp length), the capacitors are reset to prevent saturation. Thus, the data consist of linearly rising ramps for which the slope is proportional to the flux. See Figure 2 for an illustration of the read-out sequence.



Frequencies of pattern generator output lines:

CLK	8192 Hz
SYNC	256 Hz
RCRES	256 Hz / (ramps per complete chop cycle)
CHOP	256 Hz / (ramps per fundamental chop cycle)
MOVE_GRAT	256 Hz / [(ramps per complete chop cycle)x(chop cycles per grating step)]

Pattern generator clock frequency: 4 x 8192 Hz (CLK and everything else in pattern generator derived from this)

Digital multiplexer clock frequency: 64 x 8192 Hz (PLL-generated inside WRE from CLK)

Chopper signal sampling frequency (both x nd y): 2048 Hz = CLK/4, will produce 2 x 8 = 16 samples per 1/256 s ramp sampling interval which will be mapped onto 16 out of 18 "pixels" of one fake detector column

Figure 2: FIFI-LS readout sequence

2.2 FIFI-LS observing modes

Symmetric chop mode, also known as nod-match-chop mode, is the most efficient observing mode. In this mode, the telescope chops symmetrically to its optical axis, with a matched telescope nod to remove background. A typical observation sequence will cycle through the A nod position and the B nod position in an ABBA pattern.

Most observations will be taken using symmetric chop mode. However, if the object is very bright, the efficiency is improved by observing in an asymmetric chopping mode. This mode typically consists of two map positions and one off-position per nod-cycle (an AAB pattern, where the B position contains only empty sky). Asymmetric chopping may also be used if an object's size requires a larger chop throw than is possible with symmetric chopping.

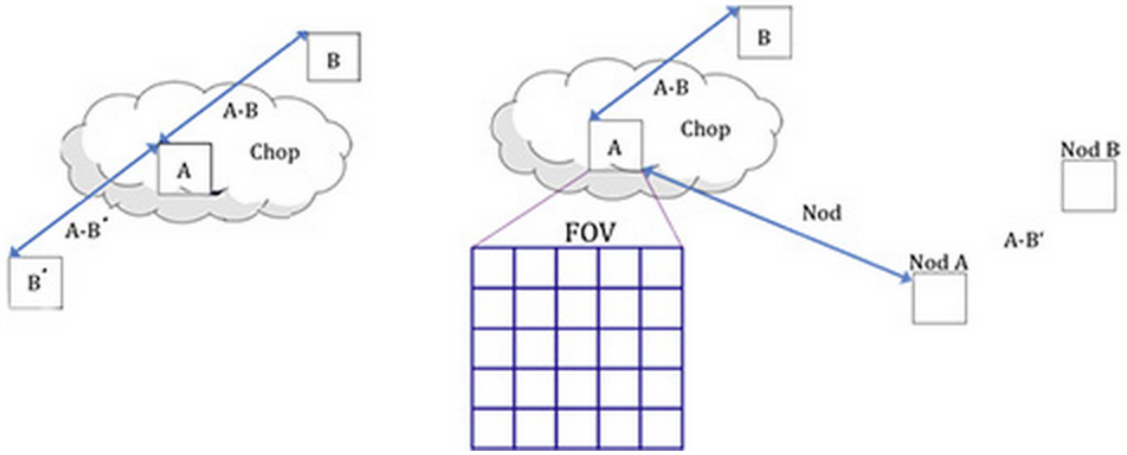


Figure 3: The geometry of chopping and nodding in the symmetric chop mode (left) and the asymmetric mode (right).

Occasionally, for very bright targets, it may be advantageous to take data with no chopping at all. This mode, called total power mode, may be taken with either symmetric or asymmetric nodding, or with no nods at all.

At each chop and nod position, it is common to step the grating through a number of positions before each telescope move. These additional grating scans effectively increase the wavelength coverage of the observation.

3 ALGORITHM DESCRIPTION

3.1 Overview of data reduction steps

This section will describe, in general terms, the major algorithms used to reduce a FIFI-LS observation. See the figure below for a flow chart showing how these algorithms fit together.

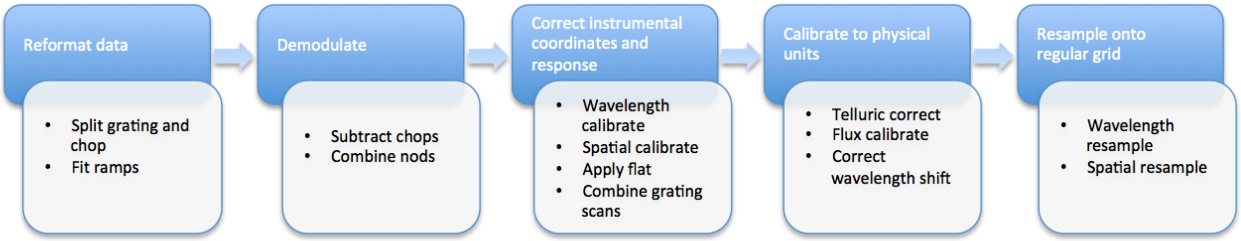


Figure 4: Processing steps for FIFI-LS data. The blue box describes an overview of steps and the white box contains the actual steps carried out.

3.2 Reduction Algorithms

The following subsections detail each of the data reduction pipeline steps outlined in the flowchart above.

3.2.1 Split Grating and Chop

A single FIFI-LS raw FITS file contains the data from both chop positions (on and off) for each grating step used in a single nod position. FIFI-LS records its grating positions in “inductosyn” units. These are long integer values that are used to convert the data to a micron scale in the wavelength calibrate step.

The raw FIFI-LS data consist of a header (metadata describing the observation) and a table of voltage readings from the detector pixels. Each data section contains one frame, i.e. simultaneous readouts of all detector pixels and chopper values.

The data header is sent before each frame. The following 8 unsigned 16-bit words contain the header information.

- Word 0: The word #8000 marks the start of the header.
- Word 1: The low word of the 32-bit frame counter.
- Word 2: The high word of the 32-bit frame counter.
- Word 3: The flag word. Bit 0 is the chopper signal. Bit 1 is the detector (0=red, 1=blue). Unused bits are fixed to 1 to recognize this flag word.
- Word 4: The sample count as defined in the timing diagrams (see Figure 2 above). This count gets advanced at every sync pulse and reset at every long sync pulse.
- Word 5: The ramp count as defined in the timing diagrams. This counter gets advanced with a long sync pulse and reset by RCRES.
- Word 6: The scan index.
- Word 7: A spare word (for now used as “end of header”: #7FFF).

Only columns 3, 4, and 5 are used in the split grating/chop step. The following shows an example header for a raw RED FIFI-LS file, as read in IDL:

```
IDL> data = mrdfits(‘fifi_raw.fits’, 1, /unsigned)
IDL> print, data.header
   32768   28160      15       1       0       0      89   32767
   32768   28161      15       1       1       0      89   32767
```

```
        32768   28162      15        1       2       0       89   32767
        32768   28163      15        1       3       0       89   32767
        32768   28164      15        1       4       0       89   32767
        32768   28165      15        1       5       0       89   32767
        32768   28166      15        1       6       0       89   32767
        32768   28167      15        1       7       0       89   32767
        32768   28168      15        1       8       0       89   32767
        32768   28169      15        1       9       0       89   32767
        32768   28170      15        1      10       0       89   32767
        32768   28171      15        1      11       0       89   32767
        32768   28172      15        1      12       0       89   32767
        32768   28173      15        1      13       0       89   32767
        32768   28174      15        1      14       0       89   32767
        32768   28175      15        1      15       0       89   32767
        32768   28176      15        1      16       0       89   32767
        32768   28177      15        1      17       0       89   32767
        32768   28178      15        1      18       0       89   32767
        32768   28179      15        1      19       0       89   32767
        32768   28180      15        1      20       0       89   32767
        32768   28181      15        1      21       0       89   32767
        32768   28182      15        1      22       0       89   32767
        32768   28183      15        1      23       0       89   32767
        32768   28184      15        1      24       0       89   32767
        32768   28185      15        1      25       0       89   32767
        32768   28186      15        1      26       0       89   32767
        32768   28187      15        1      27       0       89   32767
        32768   28188      15        1      28       0       89   32767
        32768   28189      15        1      29       0       89   32767
        32768   28190      15        1      30       0       89   32767
        32768   28191      15        1      31       0       89   32767
        32768   28192      15        1       0       1       89   32767
        32768   28193      15        1       1       1       89   32767
        32768   28194      15        1       2       1       89   32767
        32768   28195      15        1       3       1       89   32767
…
```

The third column is all ones, indicating that the data is for the RED channel, the fourth column counts the readouts from 0 to 31, and the fifth column indicates the ramp number.

Where each chop and grating position starts and stops in the raw data table is determined using the header keywords RAMPLN_[B,R], C_CYC_[B,R], C_CHOPLN, G_PSUP_[B,R], G_PSDN_[B,R] keywords. A RED data example is as follows:

```
RAMPLN_R=                    32 / number of readouts per red ramp
C_CHOPLN=                    64 / number of readouts per chop position
G_PSUP_R=                     5 / number of grating position up in one cycle
G_PSDN_R=                     0 / number of grating position down in one cycle
```

Here, C_CHOPLN / RAMPLN_R is 64 / 32 = 2; therefore, there are 2 ramps per chop.

Each chop switch index is determined using the 5th column in the header. It is chop 0 if the value is odd and chop 1 if the value is even. Grating scan information determines how that chop phase is split up into separate extensions using the following formula:

*binsize = (nreadout * ramplength) / (nstep * choplength)*

where *nreadout* is the total number of readouts (frames) in the file, *ramplength* is determined by the appropriate RAMPLN keyword, *nstep* is the number of grating steps (G_PSDN + G_PSUP), and *choplength* is the number of readouts per chop position (C_CHOPLN).

The binary data section is comprised of 468 signed 16-bit words: one each for 25 spaxels, plus one control value, times 18 spectral channels ("spexels"). The spaxels are read out one spectral channel at a time. Spectral channel zero of all 25 pixels are read out, and then a chopper value (analog readout from the secondary mirror) is recorded; then the next channel of all the pixels is read out, and then the next chopper value, and so on, through all the spectral channels. The chopper values are discarded during pipeline processing. Of the 18 spectral channels, channel 0 is the CRE resistor row and row 17 is the blind CRE row ("dummy channels"). These two channels are discarded; the other 16 channels are considered valid spexels.

In IDL, when the raw FITS file is read in using the *mrdfits* task, the binary table is represented as an array of structures consisting of an unsigned, one-dimensional eight-element array (element name HEADER), and a 26 x 18 x *nframe* signed long-integer array (element name DATA). The size of the third dimension of the array equals the number of frames acquired by the instrument.

The first step in the data reduction pipeline is to split out the data from each grating scan position into separate FITS tables, and to save all grating positions from a single chop position into a common file. For example, if there are five grating scans per chop, and two chop positions, then a single one-extension input raw FITS file will be reorganized into two files (chop 0 and chop 1) with 5 extensions each. For total power mode, there is only one chop position, so there will be one output file, with one extension for each grating step. Each extension in the output FITS files contains the data table corresponding to the grating position recorded in its header (keyword INDPOS). Hereafter, in the pipeline, until the Combine Grating Scans step, each grating scan extension is handled separately.

3.2.2 Fit Ramps

The flux measured in each spatial and spectral pixel is reconstructed from the readout frames by fitting a line to the voltage ramps. The slope of the line corresponds to the flux value.

Before fitting a line to a ramp, some likely bad frames are removed from the data. The chopper values (in the 26th spaxel position), the first ramp from each spaxel, and the first and last readout per ramp are all removed before fitting. Also, a ramp may be marked as saturated if it does not have its highest peak in the last readout of the ramp. If this occurs, the readout before the highest peak is removed before fitting, along with any readouts after it. This ensures that the slope is not contaminated by any non-linearity near the saturation point.

Typically, multiple ramps are taken at each chop position. After the slope of each ramp is derived, the slopes are combined with a robust weighted mean. This final averaged value is recorded as the flux for the pixel and the error on the mean is recorded as the error on the flux for that pixel. After this pipeline step, there is a flux value for each spatial and spectral pixel, recorded in a 5 x 5 x 18 data array in a separate FITS extension for each grating scan and chop position. The error values are recorded in a separate 5 x 5 x 18 table in the same FITS extension. The extension elements are named DATA and STDDEV, respectively.

Some pixels in the data array may be set to not-a-number (NaN). These are either known bad detector pixels, or pixels for which the ramp fits did not have sufficient signal-to-noise. These pixels will be ignored in all further reduction steps.

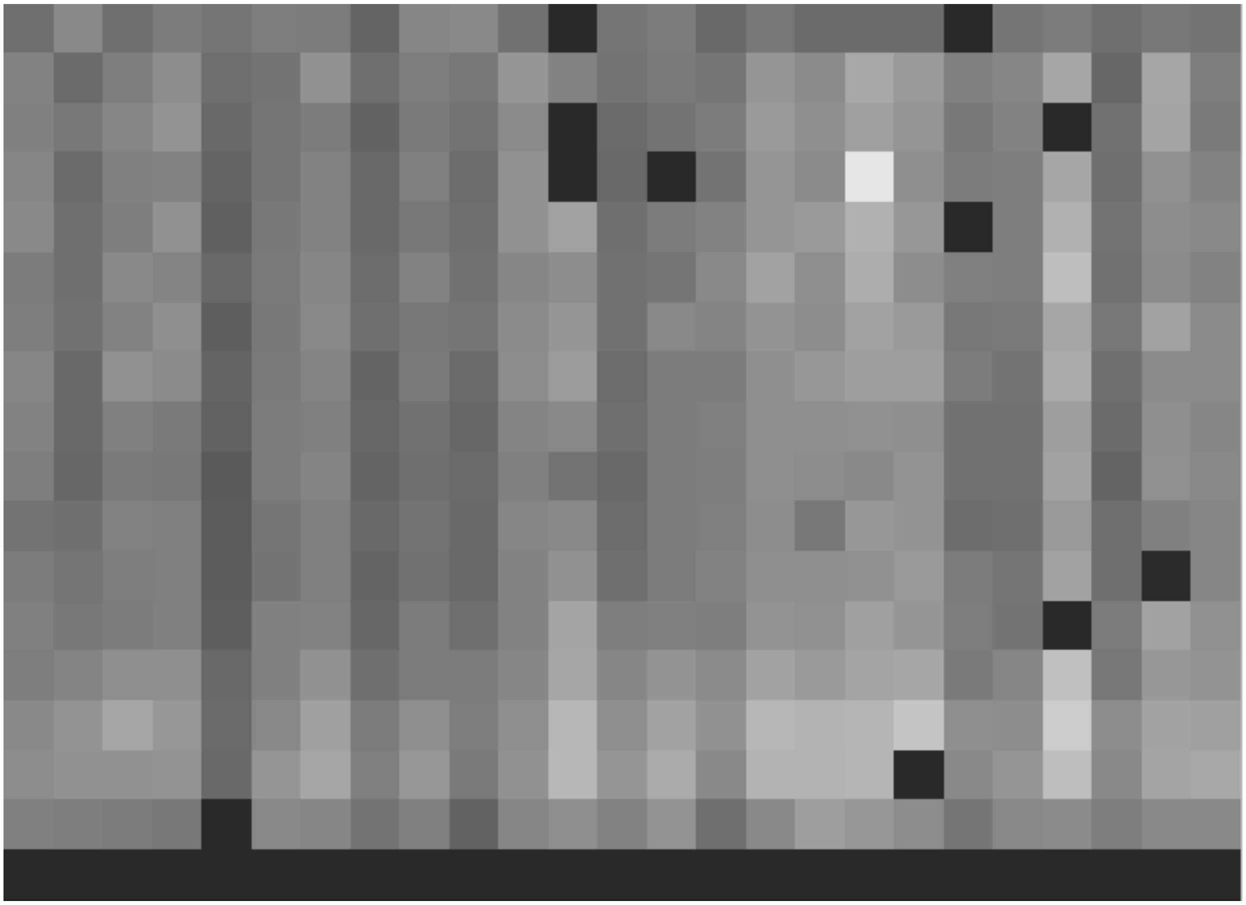


Figure 5: The flux array from a single grating scan at the chop 0 position in nod A after fitting ramps, flattened into a 25 x 18 array. The spectral dimension runs along the y-axis. The data was taken in symmetric chopping mode.

3.2.3 Subtract Chops

To remove instrument and sky background emission, the data from the two chop positions must be subtracted. For A nods, chop 1 is subtracted from chop 0. For B nods, chop 0 is subtracted from chop 1. All resulting source flux in the chop-subtracted data should therefore be positive, so that the nods are combined only by addition. This pipeline step produces one output file for each pair of input chop files. In total power mode, no chop subtraction is performed.

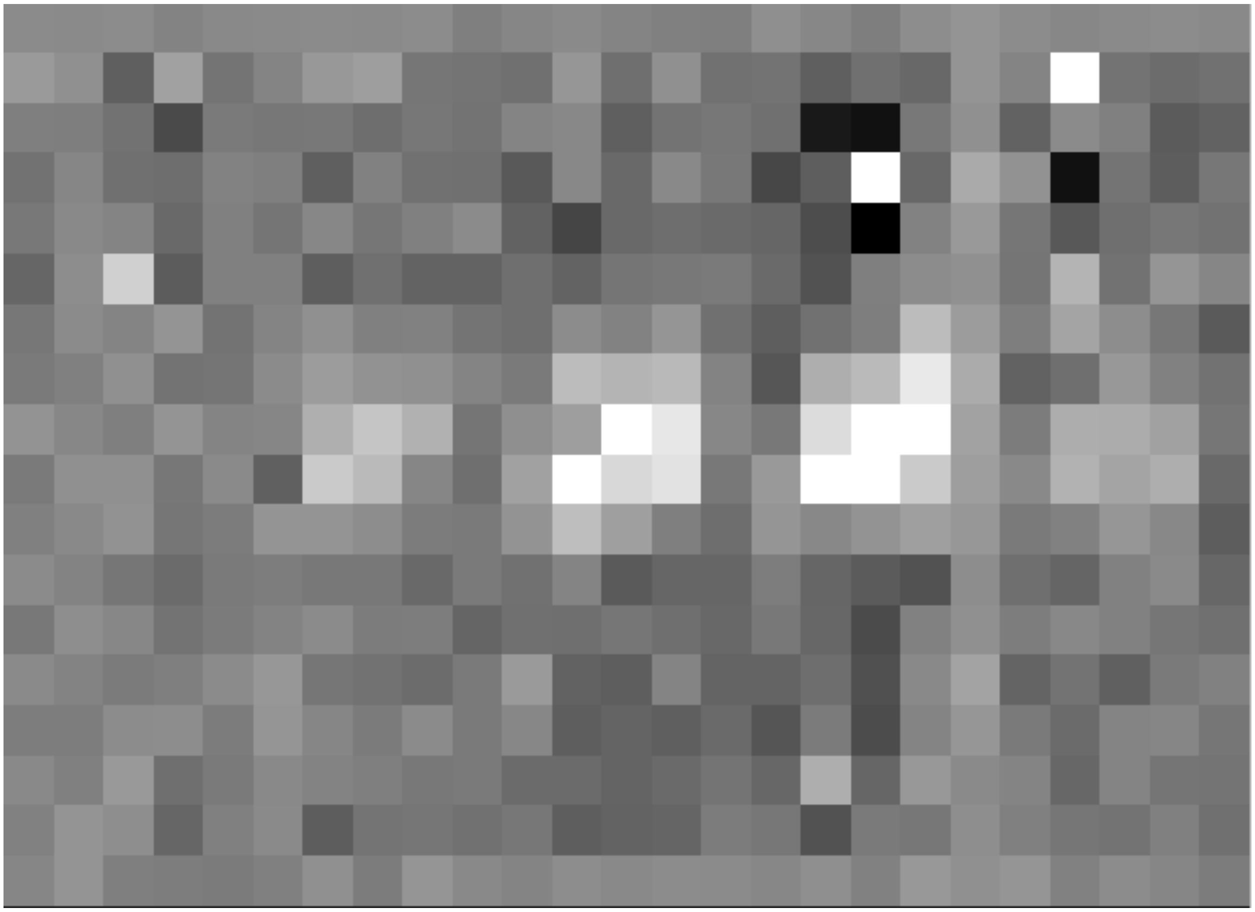


Figure 6: The same flux array as in Figure 5, with the corresponding chop 1 subtracted.

3.2.4 Combine Nods

After the chops are subtracted, the nods must be combined to remove residual background.

In symmetric chopping mode, the A nods are paired to adjacent B nods. In order to match a given A nod, a B nod must have been taken at the same dither position (FITS header keywords DLAM_MAP and DBET_MAP), and with the same grating position (INDPOS). The B nod meeting these conditions and taken nearest in time to the A nod (keyword DATE-OBS) is added to the A nod.

In asymmetric mode, a single B nod may be added to multiple A nods. For example, in an AAB pattern the B nod is combined with each preceding A nod. The B nods in this mode need not have been taken at the same dither position as the A nods, but the grating position must still match. The matching B nod taken nearest in time is added to each A nod.

This pipeline step produces an output file for each input A nod file, containing the chop- and nod-combined flux and error values for each grating scan.

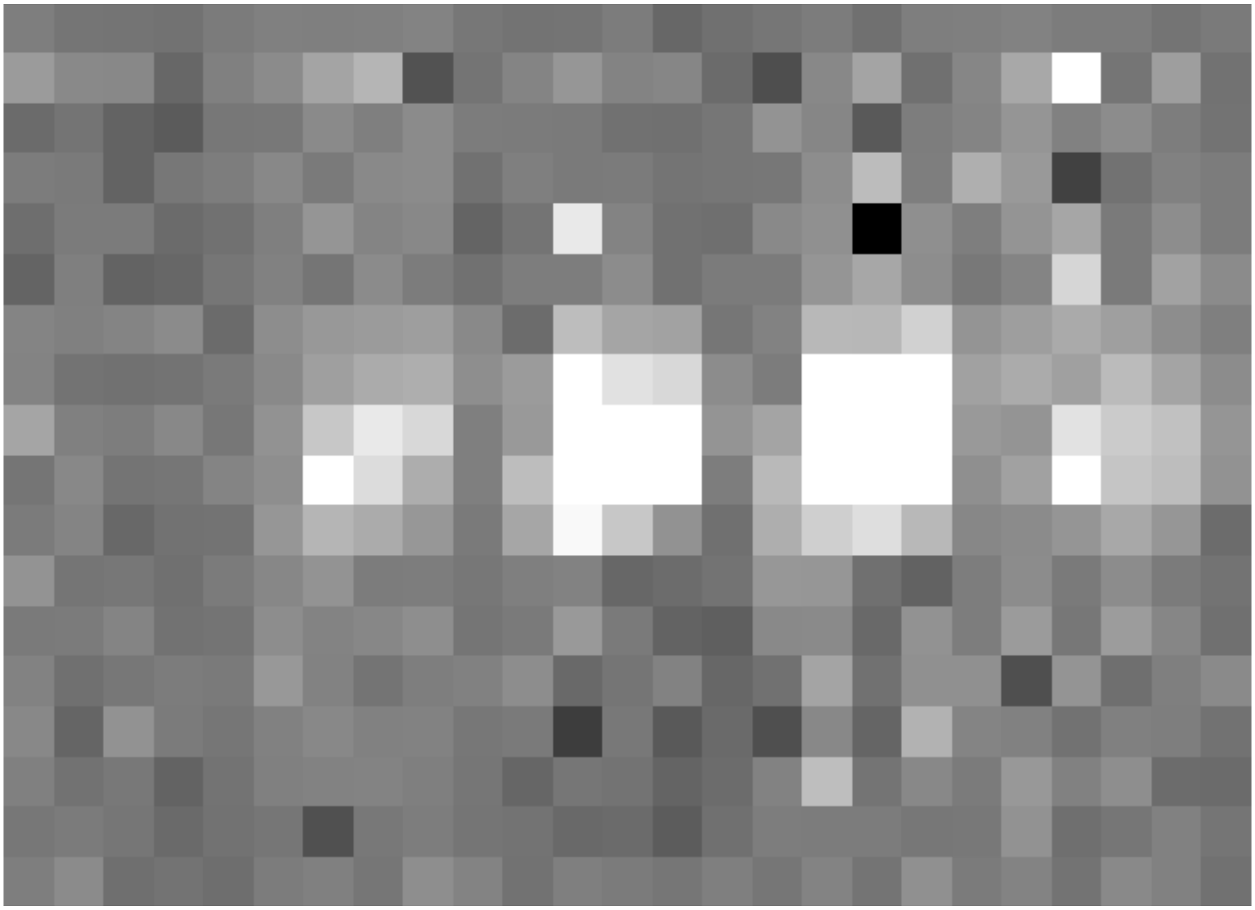


Figure 7: The chop-subtracted nod A flux array, with the corresponding nod B added.

3.2.5 Wavelength Calibrate

The wavelength calibrate step calculates wavelength values in microns for each of the spectral pixels in each grating scan, based on the known grating position, a model of the optical geometry of the instrument, and measurements of the positions of known spectral lines. The optics within FIFI-LS tend to drift with time and therefore the FIFI-LS team updates the wavelength solution every year. The wavelength equation (below) is stored in a script, while all relevant constants are stored in a reference table, with an associated date of applicability.

The wavelength (λ) for the pixel at spatial position *i* and spectral position *j* is calculated from the equation:

$$\phi_i = 2\pi \, ISF \frac{ind + ISOFF_i}{2^{24}}$$

$$\delta_j = [j - 8.5] * PS + sign[j - QOFF] * [j - QOFF]^2 * QS$$

$$g_i = g_0 * \cos\left(\tan^{-1}\left(\frac{SlitPos_i - NP}{a}\right)\right)$$

$$\lambda_{ij} = 1000 \frac{g_i}{m}\left[\sin(\phi_i - \gamma) + \sin(\phi_i + \gamma + \delta_j)\right]$$

where:

ind: the input inductosyn position

m: the spectral order of the observation (1 or 2)

are inputs that depend on the observation settings, and

ISF: inductosyn scaling factor
PS: pixel scale, in radians
QOFF: offset of quadratic pixel scale part from the "zero" pixel, in pixels
QS: quadratic pixel scale correcting factor, in radians/pixel^2
g_0: grating constant
NP: slit position offset
a: slit position scale factor
γ: offset from Littrow angle

are constants determined by the FIFI-LS team, and

$ISOFF_i$: offset of the home position from grating normal in inductosyn units for the *i*th spaxel
$SlitPos_i$: slit position of the *i*th spaxel

are values that depend on the spatial position of the pixel, also determined by the FIFI-LS team. The spaxels are ordered from 1 to 25 spatially as follows:

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

with corresponding slit position:

Slit position	Spaxel
1	21
2	22
3	23
4	24
5	25
6	
7	16
8	17
9	18
10	19
11	20
12	
13	11
14	12
15	13
16	14
17	15
18	
19	6
20	7
21	8
22	9
23	10
24	
25	1
26	2
27	3
28	4
29	5

Note that each spectral pixel has a different associated wavelength, but it also has a different effective spectral width. This width ($d\lambda/dp$) is calculated from the following equation:

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

where all variables and constants are defined above.

At this stage, the first and last spectral pixels (the dummy channels, top and bottom rows in Figure 7) are removed from the flux and error arrays, so that their sizes become 5 x 5 x 16. The wavelength values and spectral widths calculated by the pipeline for each pixel are stored in a new 5 x 5 x 16 table in each grating scan extension (elements LAMBDA and DLAMDPIX).

3.2.6 Spatial Calibrate

The locations of the spaxels are not uniform across the detector due to the optics not being perfectly aligned. See Figure 8 for a plot of the average of the center of each spaxel location as measured in the lab. This location is slightly different at each wavelength. These spaxel positions are determined by the FIFI-LS team and recorded in a look-up table.

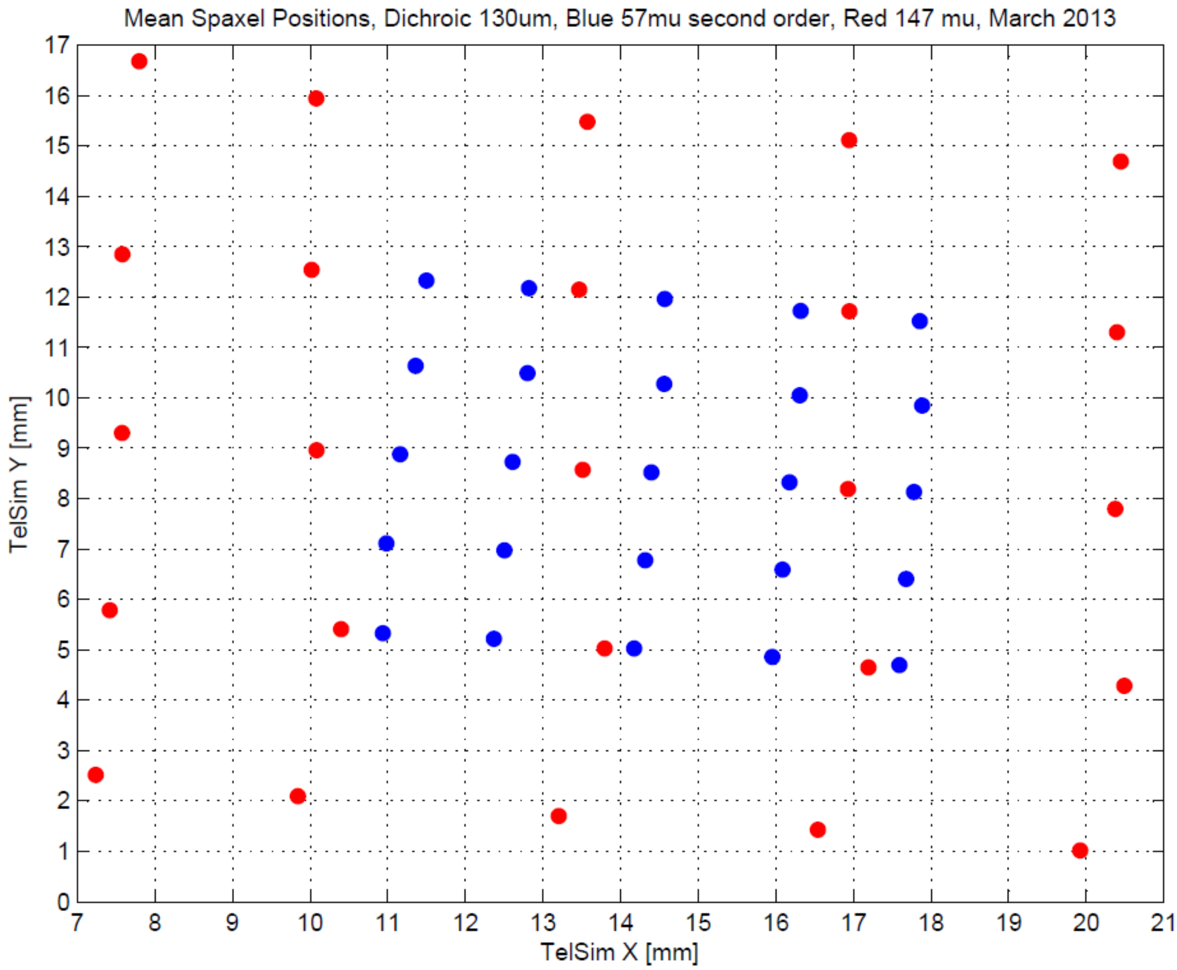


Figure 8: Average fitted spaxel positions in telescope simulator coordinates (red and blue channel) in the focal plane of the telescope. In these coordinates, blue spaxels are about 1.5 mm along an edge; red spaxels are about 3 mm. The plate scale is about 4 arcseconds per mm.

For a particular observation, the recorded dither offsets in arcseconds are used to calculate the x and y coordinates for the pixel in the *i*th spatial position and the *j*th spectral position using the following formulae:

$$x_{ij} = ps * (xpos_{ij} + dx) + d\lambda \cos(\theta) - d\beta \sin(\theta)$$
$$y_{ij} = ps * (ypos_{ij} + dy) + d\lambda \sin(\theta) + d\beta \cos(\theta)$$

where *ps* is the plate scale in arcseconds/mm (FITS header keyword PLATSCAL), $d\lambda$ is the right ascension dither offset in arcseconds (keyword DLAM_MAP), $d\beta$ is the declination dither offset in arcseconds (keyword DBET_MAP), θ is the detector angle, $xpos_{ij}$ and $ypos_{ij}$ are the fitted spaxel positions in mm for pixel *i, j*, and *dx* and *dy* are the spatial offsets between the primary array (usually BLUE), used for telescope pointing, and the secondary array (usually RED). The *dx* and *dy* offsets also take into account a small offset between the instrument boresight and the telescope boresight, for both the primary and secondary arrays. By default, the coordinates are then rotated by the detector angle (minus 180 degrees), and the y-coordinates are inverted in order to set North up and East left in the final coordinate system:

$$x'_{ij,} = -x_{ij} \cos(\theta) + y_{ij} \sin(\theta)$$
$$y'_{ij} = x_{ij} \sin(\theta) + y_{ij} \cos(\theta)$$

The pipeline stores these calculated x and y coordinates in two 5 x 5 x 16 data tables in each FITS extension (elements XS and YS).

3.2.7 Apply Flat

In order to correct for variations in response among the individual pixels, the FIFI-LS team has generated flat field arrays that correct for the average difference in spatial and spectral response. There are normalized flat fields available for a range of standard central wavelengths: 52, 57, 63, 70, 72, 79, 88, 96, 118, 145, 153, 158, 163, 185, and 200 microns. The pipeline selects the flat that is nearest the central wavelength for the observation (FITS keyword G_WAVE_R for the RED channel, G_WAVE_B for the BLUE channel), and divides the flux array by the flat. The output file format does not change for this step.

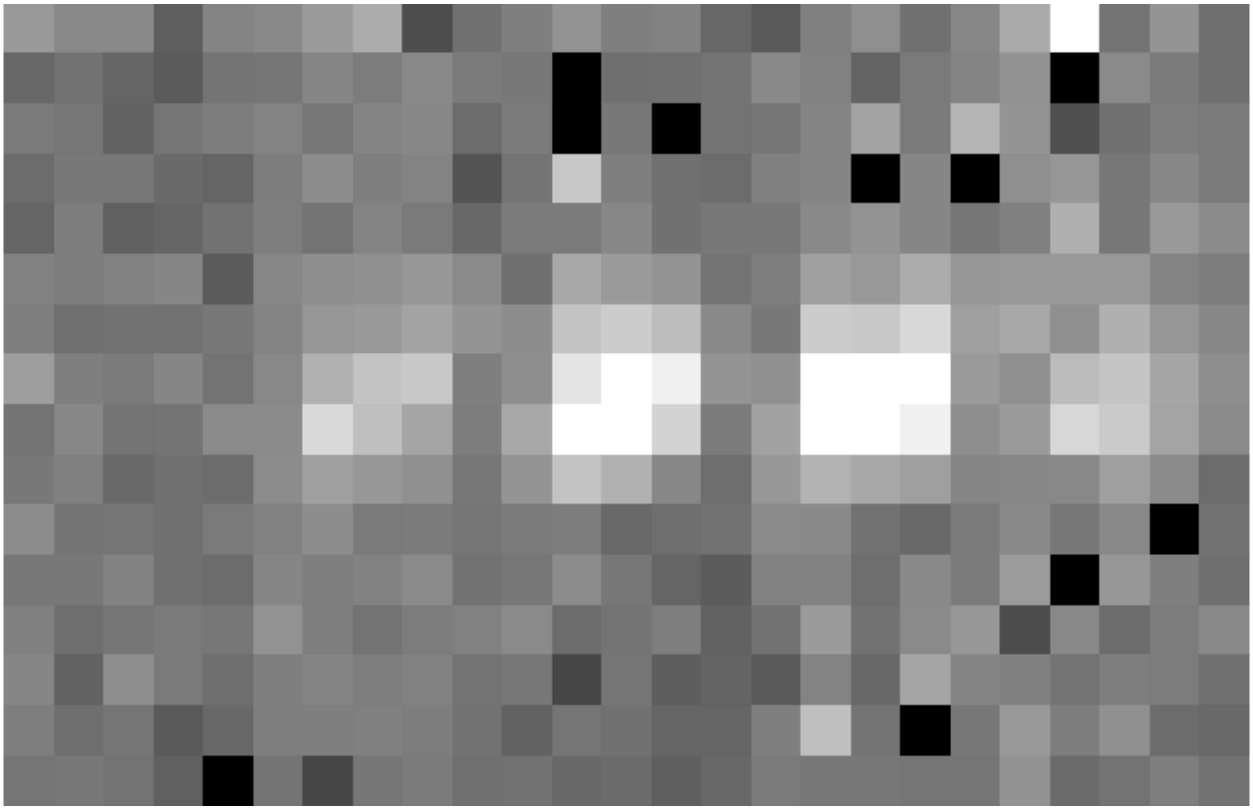


Figure 9: The flat-corrected flux array.

3.2.8 Combine Grating Scans

Up until this point, all processing has been done on each grating scan extension separately. The pipeline now combines the data from all grating scans, in order to fill in the wavelength coverage of the observation.

Since each pixel has a different spectral width, the pipeline first applies a scaling factor to each pixel to correct it to a common width. The scale factor for the pixel in the *i*th spatial position and the *j*th spectral position is the mean spectral width of the first extension in the input file, divided by $d\lambda/dp_{ij}$, as calculated above, in the wavelength calibration step. The integrated flux for each pixel in each grating scan should then be directly comparable. The common spectral width is recorded in the output FITS header under the keyword SPEXLWID.

Due to slight variations in the readout electronics, however, there may also be additive offsets in the overall flux level recorded in each grating scan. To correct for this bias offset, the pipeline calculates the mean value of all pixels in the overlapping wavelength regions for each grating scan. This value, minus the mean over all scans, is subtracted from each grating scan, in order to set all extensions to a common bias level.

Finally, the pipeline sorts the data from all grating scans by their associated wavelength values in microns, and stores the result in a single data array with dimensions 5 x 5 x (16 * *nscan*), where *nscan* is the total number of grating scans in the input file. Note that the wavelengths are still irregularly sampled at this point, due to the differing wavelength solutions for each grating scan and spatial pixel. All arrays in the output fits file (DATA, STDDEV, LAMBDA, XS, and YS) now have dimensions 5 x 5 x (16 * *nscan*). The DLAMDPIX array is removed, since the fluxes have been corrected to a common spexel width.

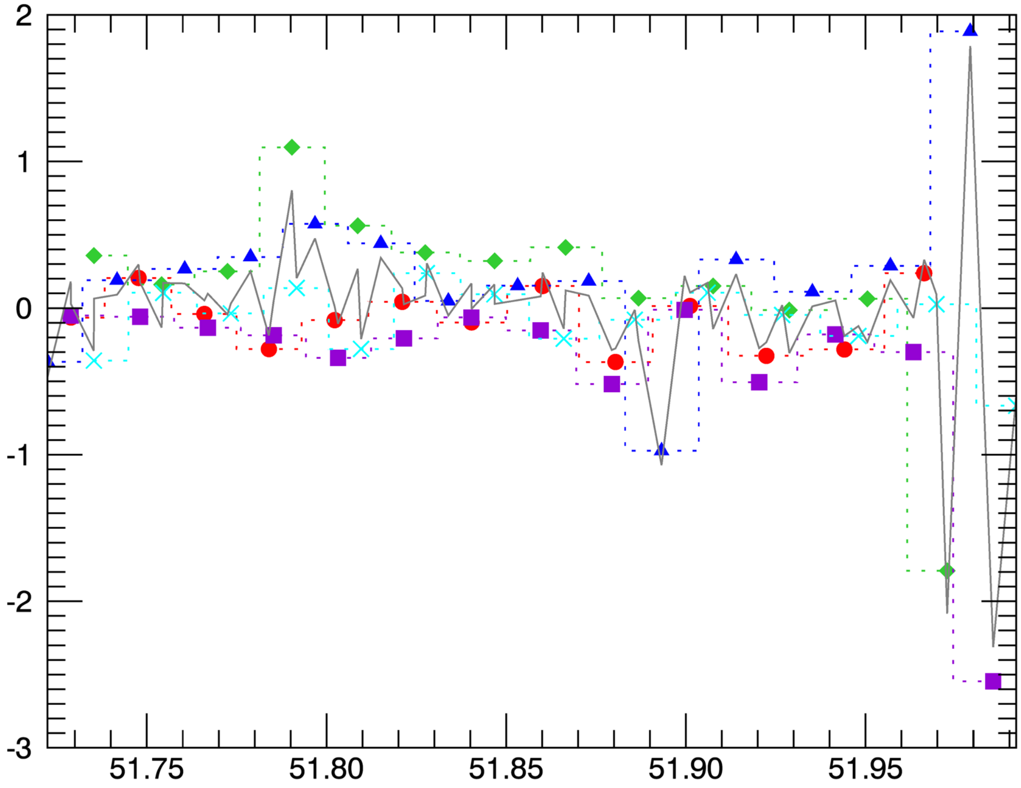


Figure 10: Example spectral flux from the center spaxel for a single dither position. Each different symbol and color represents a different grating scan, after correction to a common spectral width. The gray line indicates the combined flux array, after bias correction.

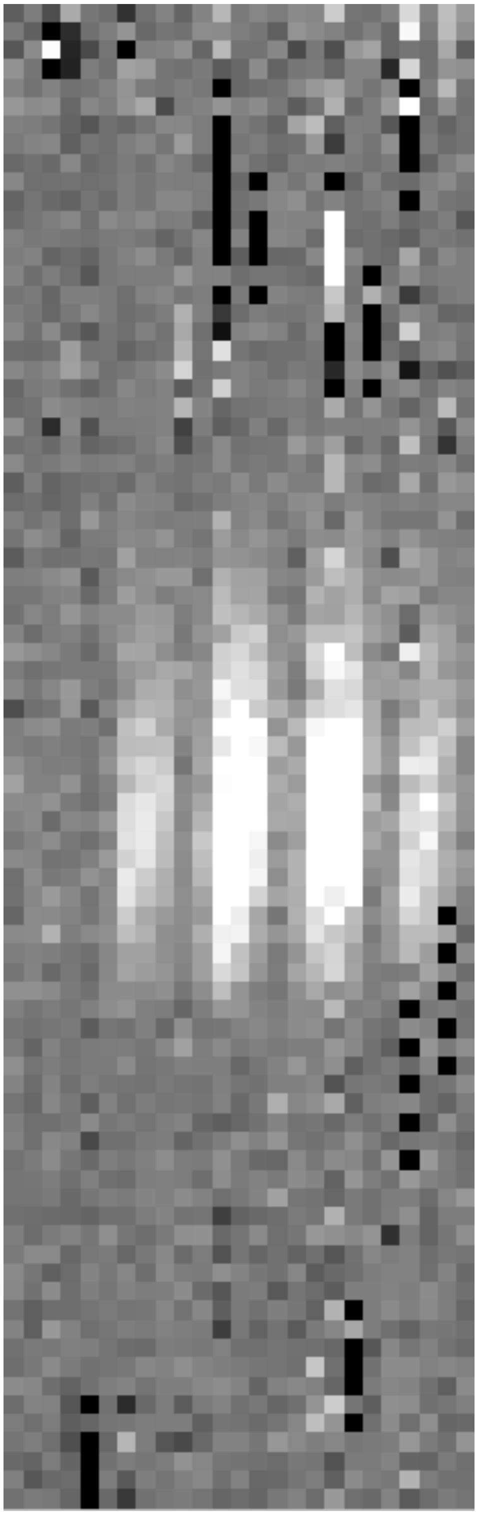


Figure 11: The full 5 x 5 x 80 flux array (flattened to 25 x 80), after combining 5 grating scans.

3.2.9 Telluric Correct

Telluric correction is the process of attempting to correct an observed spectrum for absorption by molecules in the earth's atmosphere, in order to recover the intrinsic ("exo-atmospheric") spectrum of the source. The atmospheric molecular components (primarily water, ozone, CO2) can produce both broad absorption features that are well resolved by FIFI-LS and narrow, unresolved features. The strongest absorption features are expected to be due to water. Because SOFIA travels quite large distances during even short observing legs, the water vapor content along the line of sight through the atmosphere can vary substantially on fairly short timescales during an observation. Therefore, observing a "telluric standard," as is usually done for ground-based observations, will not necessarily provide an accurate absorption correction spectrum. For this reason, telluric corrections of FIFI-LS data relies on models of the atmospheric absorption, as provided by codes such as ATRAN, in combination with the estimated line-of-sight water vapor content (precipitable water vapor, PWV) provided by the water vapor monitor (WVM) aboard SOFIA. Currently, the WVM does not generate PWV values that are inserted into the FITS headers of the FIFI-LS data files. It is expected that these values will become available in the near future and at that point the PWV values will be used to generate telluric correction spectra. Until then, correction spectra are generated using the expected PWV value for the flight altitude and airmass appropriate for the observations. Experience with both FIFI-LS and the FORCAST instrument has shown that this method can produce reasonably accurate corrections in the vicinity of broad, shallow telluric absorption features. However, corrections in regions with deep, sharp features (e.g., near 63 microns) are extremely sensitive to the actual PWV along the line of sight. Without accurate atmospheric models and measurements of the PWV, it cannot be expected that standard models will provide reliable telluric corrections. Furthermore, accurate correction of spectral lines in the vicinity of narrow telluric absorption features is problematic even with the use of good atmospheric models and knowledge of the PWV. This is due to the fact that the observed spectrum is the result of a multiplication of the intrinsic spectrum by the telluric absorption spectrum, and then a convolution of the product with the instrumental profile, whereas the correction derived from a model is the result of the convolution of the theoretical telluric absorption spectrum with the instrumental profile. The division of the former by the latter does not necessarily yield the correct results, and the output spectrum may retain telluric artifacts after telluric correction.

A set of ATRAN models appropriate for a range of altitudes and zenith angles has been generated for pipeline use. In this step, the pipeline selects the model closest to the observed altitude and zenith angle, smooths the transmission model to the resolution of the observed spectrum, bins the transmission data to the observed wavelength and spectral width of each spexel, and then divides the data by the transmission model. Very low transmission values result in poor corrections, so any pixel for which the transmission is less than 60% (by default) is set to NaN. For reference, the smoothed, binned transmission data is attached to the FITS file as a 5 x 5 x (16 * *nscan*) data table in the first FITS extension (element ATRAN).

Since the telluric correction may introduce artifacts, or may, at some wavelength settings, produce flux cubes for which all pixels are set to NaN, the pipeline also propagates the uncorrected flux cube through the remaining reduction steps. The telluric-corrected cube and its associated error are stored in the DATA and STDDEV fields of the binary table. The uncorrected cube and its associated error are stored in the UNCORRECTED_DATA and UNCORRECTED_STDDEV fields.

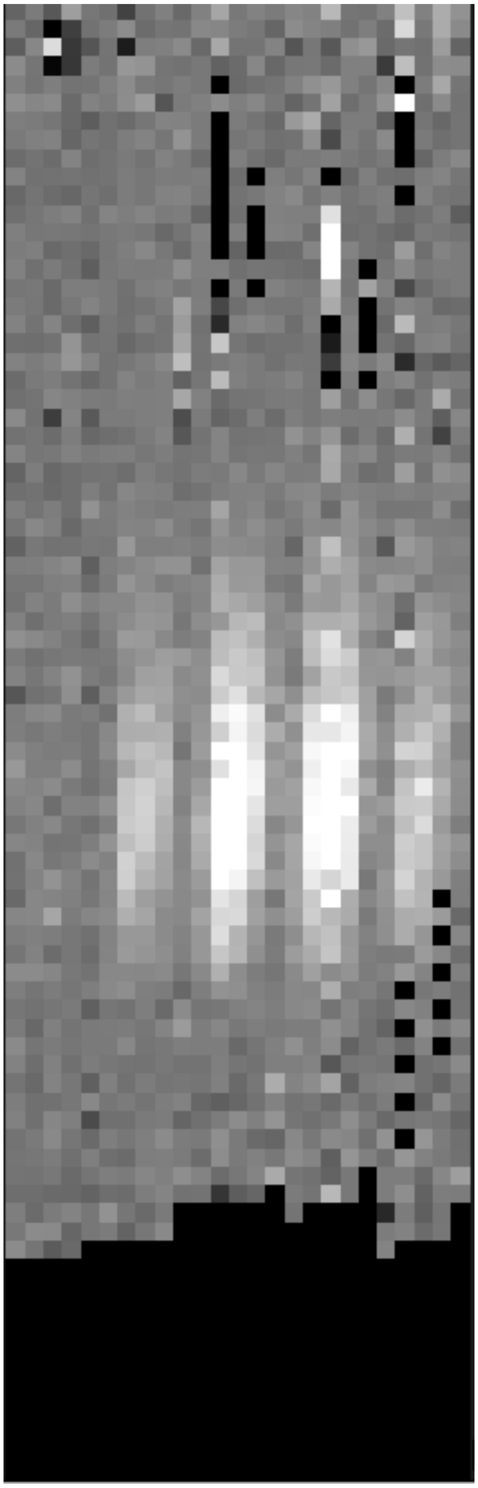


Figure 12: The telluric-corrected flux array. The lower pixels are set to NaN due to poor atmospheric transmission at those wavelengths.

3.2.10 Flux Calibrate

Flux calibration of FIFI-LS data is carried out via the division of the instrumental response, as recorded in response files appropriate for each grating setting, wavelength range, and dichroic. The response values have units of V/s/Jy and are derived from observations of “flux standards.” At the wavelengths at which FIFI-LS operates, there are very few stars bright enough to yield high signal-to-noise data useful for flux calibration purposes. Therefore, observations of asteroids, planets, and planetary moons are used, along with models of such objects, to derive the response curves. Since the observed fluxes of such solar system objects vary with time, the models must be generated for the time of each specific observation. To date, observations of Mars have been used as the primary flux calibration source. Predicted total fluxes for Mars across the FIFI-LS passband at the specific UT dates of the observations have been generated using the model of Lellouch and Amri.[1] Predicted fluxes at several frequencies have been computed and these have then been fit with blackbody curves to derive values at a large number of wavelength points. The deviations of the fits from the input predictions are much less than 1%. After the models have been generated, the telluric-corrected spectra of the standards, in units of V/s, are divided by the theoretical spectra, in Jy. The results are smoothed and then fit with a polynomial to derive response functions (V/s/Jy) that can then used to flux calibrate the telluric-corrected spectra of other astronomical sources.

The pipeline stores a set of response functions for each channel and dichroic value. To perform flux calibration, it selects the correct response function for each input observation, interpolates the response onto the wavelengths of each spexel, and divides the flux by the response value to convert it to physical units (Jy). From this point on, the data products are considered to be Level 3 (FITS keyword PROCSTAT=LEVEL_3). For reference, the resampled response data is attached to the FITS file as a 5 x 5 x (16 * *nscan*) data table in the first FITS extension (element RESPONSE). Flux calibration is applied to both the telluric-corrected cube and the uncorrected cube. The estimated systematic error in the flux calibration is recorded in the CALERR keyword in the FITS header, as an average fractional error. At this time, flux calibration is expected to be good to within about 20% (CALERR = 0.2).

3.2.11 Correct Wave Shift

Due to the motion of the earth with respect to the local standard of rest, the wavelengths of features in the spectra of astronomical sources will appear to be slightly shifted, by different amounts on different observation dates. In order to avoid introducing a broadening of spectral features when multiple observations obtained over different nights are combined, the wavelength calibration of FIFI-LS observations must be adjusted to remove the barycentric wavelength shift. This shift is calculated as an expected wavelength shift ($d\lambda/\lambda$), from the radial velocity of the earth with respect to the sun and the sun with respect to the local standard of rest, on the observation date, toward the RA and Dec of the observed target. This shift is recorded in the header keyword BARYSHFT, and applied to the wavelength calibration in the LAMBDA field as:

$$\lambda' = \lambda + \lambda(d\lambda/\lambda)$$

[1] See http://www.lesia.obspm.fr/perso/emmanuel-lellouch/mars/index.php

Since the telluric absorption lines do not change with the motion of the earth, the barycentric wavelength shift cannot be applied to non-telluric-corrected data. Doing so would result in a spectrum in which both the intrinsic features and the telluric lines are shifted. Therefore, the unshifted wavelength calibration is also propagated in the output file, in the field UNCORRECTED_LAMBDA.

3.2.12 Wavelength Resample

Next, the pipeline resamples the spectral flux for each spatial pixel onto a regular grid of wavelengths. It first determines the maximum and minimum wavelengths present in all input files, from all dither positions for the observation. This sets the range of the output grid. The spacing of the output grid is set by the desired oversampling. By default, the pipeline samples the average expected spectral FWHM for the observation with 8 output pixels. For example, for a BLUE order 2 observation, as in the Figures above, the average spectral resolution is 900. If the input files span a wavelength range of 51.65 to 51.99 microns, the average wavelength is 51.82 microns, and the expected spectral FWHM is 51.82 / 900 = 0.0576 microns. Sampling this FWHM with 8 pixels creates a grid with a spectral width of 0.0072 microns. This will sample the full range of wavelengths with 48 spectral pixels (see Figure 14). Since the pixel width changes in the resampling from the intrinsic spectral width, as recorded in the combine grating scans step (keyword SPEXLWID), the output flux will be corrected by a factor of the new width, divided by the old spectral width, in order to conserve integrated flux.

For each spatial pixel, the resampling is performed by looping over the output wavelength grid, and finding all flux values with assigned wavelengths within a fitting window (typically ±0.25 times the spectral FWHM). Outlier flux values (typically those greater than 5 sigma from the mean value) may be rejected. Then, a low-order one-dimensional polynomial is fit to all the good data points, with weighting by the error on the flux, as calculated by the pipeline, and the distance of the input data value from the grid location. The output flux value is the value of the polynomial, evaluated at the grid location, and the associated error is the error on the fit.

Both the telluric-corrected and the uncorrected flux cubes are resampled in this step, onto the same wavelength grid. However, the telluric-corrected cube is resampled using the wavelengths corrected for barycentric motion, and the uncorrected cube is resampled using the original wavelength calibration. The spectra from the uncorrected cube will appear slightly shifted with respect to the spectra from the telluric-corrected cube.

The pipeline stores the resampled data as a 2D FITS image extension (*nwave* x 25) with extension name FLUX. The associated error (also *nwave* x 25) is stored in a separate extension, with the name ERROR. The non-telluric-corrected cubes are stored in UNCORRECTED_FLUX and UNCORRECTED_ERROR extensions, respectively, and will also have size *nwave* x 25. The output wavelength values for the grid are stored in a 1D extension called WAVELENGTH, with size *nwave*. The x and y coordinates of the spatial pixels are recorded in 1D X and Y extensions, each with size 25. For reference, a model of the atmospheric transmission spectrum, smoothed to the resolution of the observation, and the instrumental response curve used in flux calibration are also attached to the FITS file in 1D extensions called TRANSMISSION and RESPONSE, with size *nwave*. In the example above, *nwave* is 48. Finally, an unsmoothed transmission spectrum is attached in a 2D image extension called UNSMOOTHED_TRANSMISSION. This extension will have size *ntrans* x 2, where *ntrans* is the number of data points in the spectrum, the first row is the wavelength array, and the second row is the transmission value. This spectrum may be useful for further analysis of the data (e.g. for determining the total flux in an emission line).

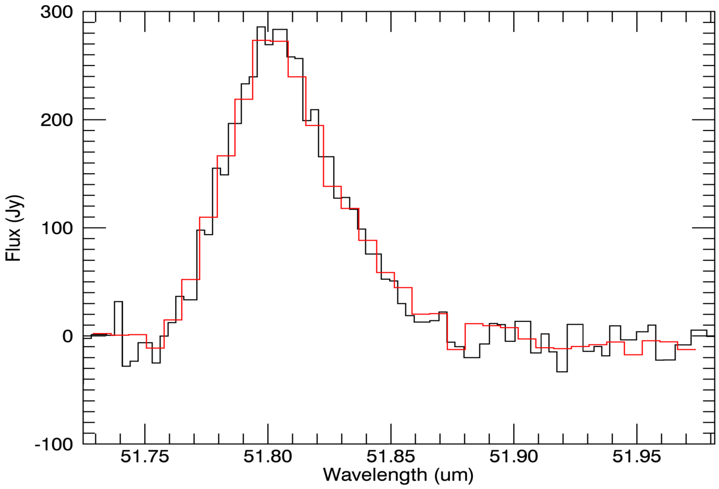


Figure 13: Example spectral flux from the center spaxel. Black is the input flux; red is the resampled flux.

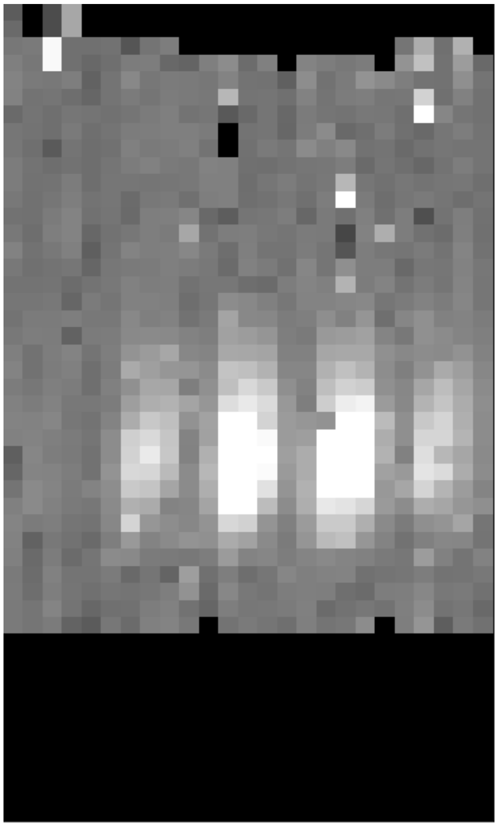


Figure 14: Wavelength resampled flux array, now 25 x 48

3.2.13 Spatial Resample

Finally, the pipeline resamples each wavelength plane onto a regular spatial grid. This step combines the spatial information from all input nod-combined dither positions into a single output map. Note that these input files are resampled separately in the wavelength resample step, but onto the same master wavelength grid, so that they may be compared directly. The maximum and minimum x and y offsets in the input data set the range of the spatial grid. The spacing (*dx*) is set by a reference spatial FWHM for the observed channel (5” for BLUE, 10” for RED), divided by the desired oversampling. For example, for a BLUE order 2 observation, the default oversampling of 5 pixels sets the output grid spacing to 1 arcsecond. If the range of x offsets is -14.0 to 15.8 arcseconds, and the range of y offsets is -15.9 to 15.7 arcseconds, then the output spatial grid will have dimensions 31 x 33. The pixel width again changes after resampling, so the output flux is multiplied by the area of the new pixel (dx^2), divided by the intrinsic area of the spaxel (approximately 36 arcseconds^2 for BLUE, 144 arcseconds^2 for RED). The output files are specified to have a spatial sampling of 1” per pixel in the BLUE and 2” per pixel in the RED.

The spatial resampling algorithm is similar to the wavelength-resampling algorithm. For each wavelength plane, the algorithm loops over the output spatial grid, finding values within a local fitting window. For the spatial grid, a larger fit window is typically necessary than for the spectral grid, since the observation setup usually allows more oversampling in wavelength than in space. The default value for the fit window is 3.0 times the average FWHM for the channel observed. Outlier flux values (typically those greater than 10 sigma away from the mean within the window) are rejected, and remaining good values are fit with a low order polynomial surface fit. The fits are usually weighted by the error on the flux and a Gaussian function of the distance from the data point to the grid location. The output flux for each pixel is the value of the surface polynomial, evaluated at the grid location, and the associated error value is the error on the fit.

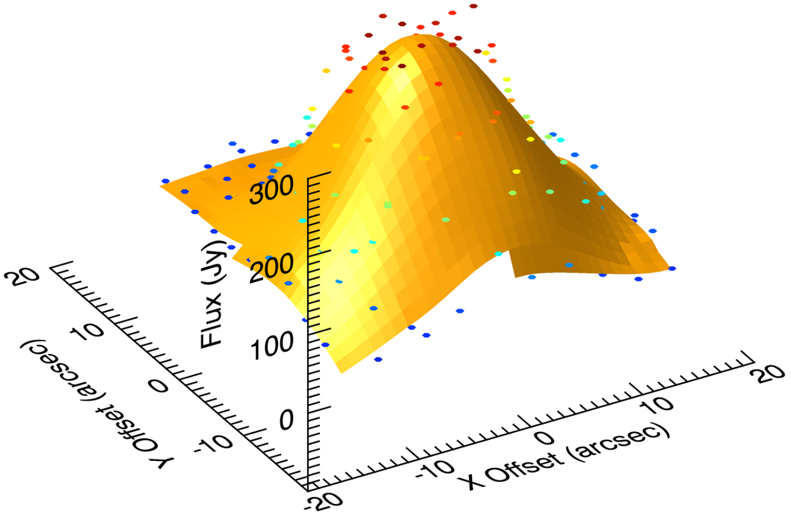


Figure 15: Fit surface at a single wavelength plane (51.804 μm), as calculated by the spatial resampling algorithm. Circles indicate the raw data points.

For some types of observations, especially undithered observations of point sources, for which the spatial FWHM is undersampled, the polynomial surface fits may not return good results. In these cases, it may be beneficial to use an alternate resampling algorithm. In this algorithm, the master spatial grid is determined as above, but each input file is interpolated onto it individually, for each wavelength plane, using a radial basis function interpolation. Areas of the spatial grid for which there is no data in the input file are set to NaN. The interpolated cubes are then mean-combined, ignoring any NaNs, to produce the final output cube.

For either algorithm, the pipeline also generates an exposure map cube indicating the number of observations of the source that were taken at each pixel (see Figure 16). If all input observations were taken on a single SOFIA flight, the output product is still considered to be Level 3 (FITS keyword PROCSTAT=LEVEL_3). If, however, observations taken across multiple flights are combined in this step, the output product is considered a higher-level mosaic, and is marked Level 4 (PROCSTAT=LEVEL_4).

The final output from the pipeline is a FITS file with 10 image extensions:

- FLUX: The *nx* x *ny* x *nw* cube of flux values.
- ERROR: The associated error values on the flux (also *nx* x *ny* x *nw*).
- UNCORRECTED_FLUX: The *nx* x *ny* x *nw* cube of flux values that have not been corrected for atmospheric transmission.
- UNCORRECTED_ERROR: The associated error values on the uncorrected flux (also *nx* x *ny* x *nw*).
- WAVELENGTH: The wavelength values associated with each plane of the cube (*nw*).
- X: The x-coordinates of the data, in arcsecond offsets from the base position (*nx*).
- Y: The y-coordinates of the data, in arcsecond offsets from the base position (*ny*).
- TRANSMISSION: The atmospheric transmission model (*nw*).
- RESPONSE: The instrumental response curve (*nw*).
- EXPOSURE_MAP: The exposure map (*nx* x *ny* x *nw*).
- UNSMOOTHED_TRANSMISSION: The unsmoothed atmospheric transmission model (*ntrans* x 2).

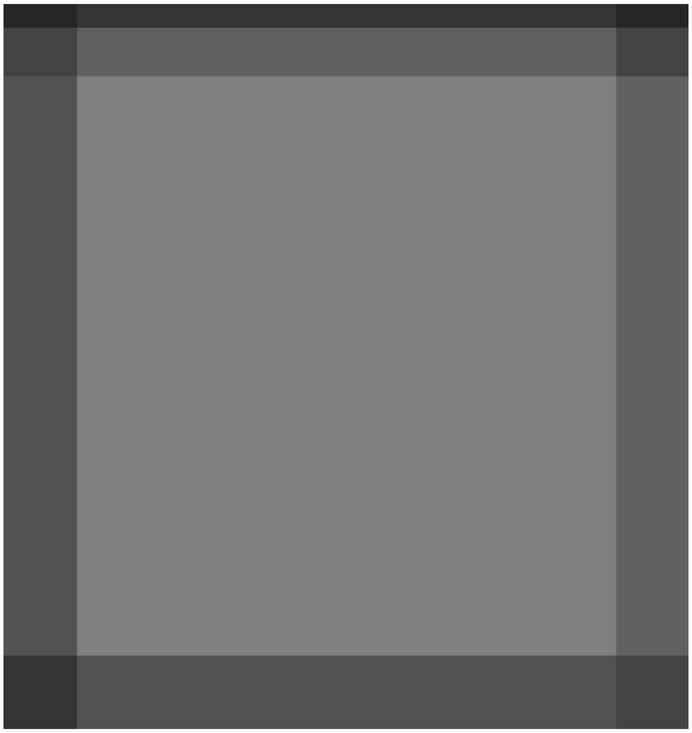


Figure 16: Exposure map of input dither positions. Values range from 4 to 16.

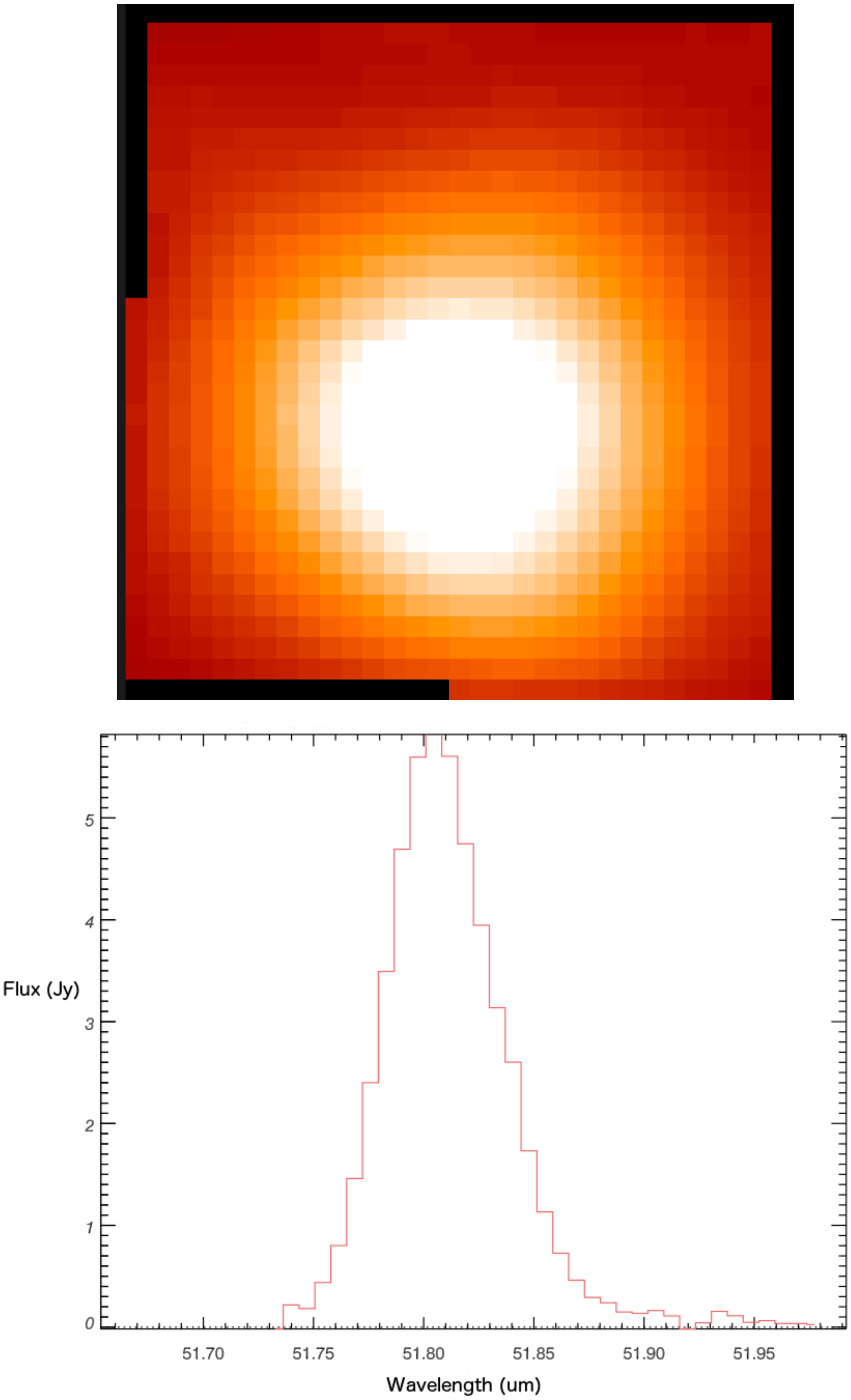


Figure 17: The final output flux cube. The image on top is a spatial slice at wavelength 51.804 µm. The plot on the bottom is a spectral slice at pixel 13, 14 (offset 0.26, -0.47 arcseconds).

4 DATA PRODUCTS

4.1.1 Filenames

FIFI-LS output files from Redux are named according to the convention:

FILENAME = F####_FI_IFS_AOR-ID_CHANNEL_Type_FN1[-FN2].fits,

where #### is the four-digit SOFIA flight number, FI is the instrument identifier, IFS specifies that it is integral field spectroscopy data, AOR-ID is the 8 digit AOR identifier for the observation, CHANNEL is either BLU or RED, Type is three letters identifying the product type (listed in the table below), and FN1 is the file number corresponding to the input file. FN1-FN2 is used if there are multiple input files for a single output file, where FN1 is the file number of the first input file and FN2 is the file number of the last input file.

4.1.2 Pipeline Products

The following table lists all intermediate products generated by Redux for FIFI-LS, in the order in which they are produced. The product type is stored in the primary FITS header of the file, under the keyword PRODTYPE. By default, the *scan_combined*, *flux_calibrated,* and *wavelength_resampled*, and *wxy_resampled* products are saved. Specifying the appropriate option in either the automatic or interactive modes will save all products. Note that as of pipeline version 1.3.2, the STDDEV and ERROR extensions do not include the systematic error on the flux calibration; this is recorded separately in the FITS header keyword CALERR in the primary extension.

Step	Description	Product Type	Proc. status	File code	Saved	Size
Split Grating / Chop	Data split into separate files by chop number and extensions by grating scans	*grating_ chop_split*	LEVEL_2	CP0, CP1	N	N_{scan} image extensions, each 26x18x1xN_{read}, where N_{scan} is the number of grating scans and N_{read} is the number of readout frames
Fit Ramps	Readout ramps fit and averaged	*ramps_fit*	LEVEL_2	RP0, RP1	N	N_{scan} binary table extensions, each containing DATA and STDDEV data cubes, each 5x5x18
Subtract Chops	Chop positions subtracted	*chop_ subtracted*	LEVEL_2	CSB	N	N_{scan} binary table extensions, each containing DATA and STDDEV data cubes, each 5x5x18
Combine Nods	Nod positions combined	*nod_ combined*	LEVEL_2	NCM	N	N_{scan} binary table extensions, each containing DATA and STDDEV data cubes, each 5x5x18
Lambda Calibrate	Wavelength values calculated and attached to data cube	*wavelength_ calibrated*	LEVEL_2	WAV	N	N_{scan} binary table extensions, each containing DATA, STDDEV, LAMBDA, and DLAMDPIX data cubes, each 5x5x16
Spatial Calibrate	Spatial values calculated and attached to data cube	*spatial_ calibrated*	LEVEL_2	XYC	N	N_{scan} binary table extensions, each containing DATA, STDDEV, LAMBDA, DLAMDPIX, XS, and YS data cubes, each 5x5x16
Apply Flat	Flat field correction applied to flux	*flat_fielded*	LEVEL_2	FLF	N	N_{scan} binary table extensions, each containing DATA, STDDEV, LAMBDA, DLAMDPIX, XS, and YS data cubes,

						each 5x5x16
Combine Scans	Combine grating scans into a single extension	*scan_ combined*	LEVEL_2	SCM	Y	1 binary table extension, containing DATA, STDDEV, LAMBDA, XS, and YS data cubes, each 5x5x(16*N_{scan})
Telluric Correct	Flux corrected for atmospheric absorption	*telluric_ corrected*	LEVEL_2	TEL	N	1 binary table extensions, containing DATA, STDDEV, UNCORRECTED_DATA, UNCORRECTED_STDDEV, LAMBDA, XS, YS, and ATRAN data cubes, each 5x5x(16*N_{scan})
Flux Calibrate	Flux values converted to physical units (Jy)	*flux_ calibrated*	LEVEL_3	CAL	Y	1 binary table extension, containing DATA, STDDEV, UNCORRECTED_DATA, UNCORRECTED_STDDEV, LAMBDA, XS, YS, ATRAN, and RESPONSE data cubes, each 5x5x(16*N_{scan})
Correct Wave Shift	Wavelength calibration corrected for barycentric moton	*wavelength_ shifted*	LEVEL_3	WSH	N	1 binary table extension, containing DATA, STDDEV, UNCORRECTED_DATA, UNCORRECTED_STDDEV, LAMBDA, UNCORRECTED_LAMBDA, XS, YS, ATRAN, and RESPONSE data cubes, each 5x5x(16*N_{scan})
Wave Resample	Resample data onto a regular wavelength grid	*wavelength_ resampled*	LEVEL_3	WGR	Y	9 image extensions: FLUX (N_w x 25), ERROR (N_w x 25), UNCORRECTED_FLUX (N_w x 25), UNCORRECTED_ERROR (N_w x 25), WAVELENGTH (N_w), X (25), Y (25), TRANSMISSION (N_w), RESPONSE (N_w), and UNSMOOTHED_TRANSMISSION (N_tx2), where N_w is the number of resampled wavelength pixels and N_t is the number of pixels in the unsmoothed transmission spectrum
Spatial Resample	Resample data onto a regular spatial grid	*wxy_ resampled*	LEVEL_3 or LEVEL_4 (if input data comes from multiple missions)	WXY	Y	10 image extensions: FLUX (N_x x N_y x N_w), ERROR (N_x x N_y x N_w), UNCORRECTED_FLUX (N_w x 25), UNCORRECTED_ERROR (N_w x 25), WAVELENGTH (N_w), X (25), Y (25), TRANSMISSION (N_w), RESPONSE (N_w), and EXPOSURE_MAP (N_x x N_y x N_w), and UNSMOOTHED_TRANSMISSION (N_tx2) where N_x and N_y are the number of resampled x and y spatial pixels

Table 1: Final and intermediate data products

4.1.3 Data Format

All files produced by the pipeline are multi-extension FITS files, for which the primary HDU contains only the primary header, and all data is contained in separate extensions. Before the combine scans step, all outputs will have a separate binary table extension for each grating position; the output of the combine scans and flux calibrate steps will have a single binary table extension. The binary tables contain the elements listed in the table above, along with a HEADER element. After the resampling steps, the data is contained in image extensions, as listed above.

5 GROUPING LEVEL_1 DATA FOR PROCESSING

FIFI-LS observations of a single object may dither in wavelength and/or in space. It is common to use a set of FIFI-LS observations to map out large regions of diffuse emission in a few small wavelength regions, but it is also common to use FIFI-LS to take a set of observations of a compact source over a wide range of wavelengths. As such, observations cannot simply be split on the central wavelength, or the base positions of the observation. Grouping relies on a keyword, FILEGPID, which defines a set of observations that should be reduced together to produce one final spectral map. In addition, however, observations must always be separated by detector channel, dichroic setting, nodding style (symmetric vs. asymmetric), observation type (source vs. sky flat), and the observational program ID. It may also be desirable to group file by the observation ID (AOR-ID), but this is considered optional. This is due to the fact that, because FIFI-LS incorporates a dichroic, different AORs can have the same grating setting for one of the two detectors. In such cases, the data sets with the same settings should be combined even though the AORs are different. All of these requirements together define a set of FITS keywords that must match in order for a group of input files to be reduced together (see Table 2).

Keyword	Data Type	Match Criterion
OBSTYPE	STR	Exact
DETCHAN	STR	Exact
DICHROIC	INT	Exact
NODSTYLE	STR	Exact
PLANID	STR	Exact
FILEGPID	STR	Exact
AOR-ID (optional)	STR	Exact

Table 2: Grouping criteria

6 CONFIGURATION AND EXECUTION

6.1 Installation

Redux is a software package written in IDL that is designed to be a framework for executing any number or combination of data reduction algorithms. Redux can run in an automatic batch mode, integrated with the SOFIA Data Pipeline System (DPS), or it can run with a graphical front end as a quick-look data viewer in flight or during manual data reduction and analysis. Redux with FIFI-LS and FSpextool was developed under MacOS X and Linux operating systems. Other operating systems may also work, but have not been tested.

Running Redux requires IDL 8.5 or later, as well as the latest version of the IDL Astronomy User's Library, the Coyote graphics library, the FIFI-LS package, the FSpextool package, and the Redux code. FIFI-LS, FSpextool, and Redux are under SOFIA DPS revision control and can be obtained directly from git repositories there. The IDL Astronomy User's Library (astrolib) is publicly available, and can be downloaded from the website at the following URL:

http://idlastro.gsfc.nasa.gov/homepage.html.

The Coyote graphics library (coyote) is also publicly available and can be downloaded from:

http://www.idlcoyote.com/documents/programs.php.

When these packages have been installed, their locations should be added to the IDL_PATH environment variable, so that their procedures are accessible to Redux.

SOFIA may distribute the Redux, FIFI-LS, and FSpextool codes as gzipped tar files. If so, unpack them, as, for example:

```
tar xvzf redux.tar.gz
tar xvzf fifi-ls.tar.gz
tar xvzf fspextool.tar.gz
```

This will create directories called *redux*, *fifi-ls*, and *fspextool*, which will contain a number of subdirectories. Each of these package directories should be added to the IDL_PATH as well.

6.2 Configuration

For FIFI-LS algorithms, default parameter values are defined by the Redux object that interfaces to them. These values may be overridden manually for each step, while running in interactive mode. They may also be overridden by an input parameter file, in JSON format, in either interactive or automatic mode. See Appendix A for an example of an input parameter file, which contains the current defaults for all parameters.

Requirements for input header keywords are also specified in a configuration file, called *headerdef.dat*, located in the *fifi-ls* package directory. This table lists the keyword name, whether it is a value required to be present in the input headers, its default value, the data type of the value, and any requirements on the value range (minimum value, maximum value, or enumerated value). The table also defines how keywords from multiple input files should be combined for a single output file (e.g. take the first value, take the sum, string-concatenate, etc.). A sample of this configuration file is also given in Appendix A. All keywords present in the table will be written to output files produced by the FIFI-LS Redux pipeline.

6.3 Input data

Redux takes as input raw FIFI-LS FITS data files, which are unsigned tables. The number of frames per raw data cube depends on the readout mode used to acquire the data. The FITS headers contain data acquisition and observation parameters and, combined with the pipeline configuration files, comprise the information necessary to complete all steps of the data reduction process. Some critical keywords are required to be present in the raw data in order to perform a successful grouping, reduction, and ingestion into the SOFIA archive (see Appendix A).

It is assumed that the input data have been successfully grouped before beginning reduction: Redux considers all input files in a reduction to be science files that are part of a single homogeneous reduction group, to be reduced together with the same parameters.

6.4 Automatic mode execution

To run the pipeline from the IDL command line as a fully automatic black box pipeline, run the pipeline wrapper (*redux_pipe.pro*). This wrapper takes as input the path to an input manifest file. This text file should contain a line specifying the number of input files, then the relative path to each input file, one per line. The script then reads these input files, instantiates the pipeline reduction object, then calls the object's reduce method. This method calls each processing step in order, as appropriate for FIFI-LS. Finally, the wrapper script will write an output manifest called *outfiles.txt* containing the names of the produced data files, and will save the as-run parameters to a file called *redux_param.json*.

This wrapper can be invoked from the IDL prompt, as

```
IDL> redux_pipe, ‘infiles.txt’
```

or directly from a terminal as

```
$ echo “redux_pipe, ‘infiles.txt’” | idl
```

The wrapper accepts a single input parameter on the command line, which allows the user to give it a parameter file that specifies any desired reduction parameters. This option is given as, for example:

```
IDL> redux_pipe, ‘infiles.txt’, PARAM_FILE=‘param.json’
```

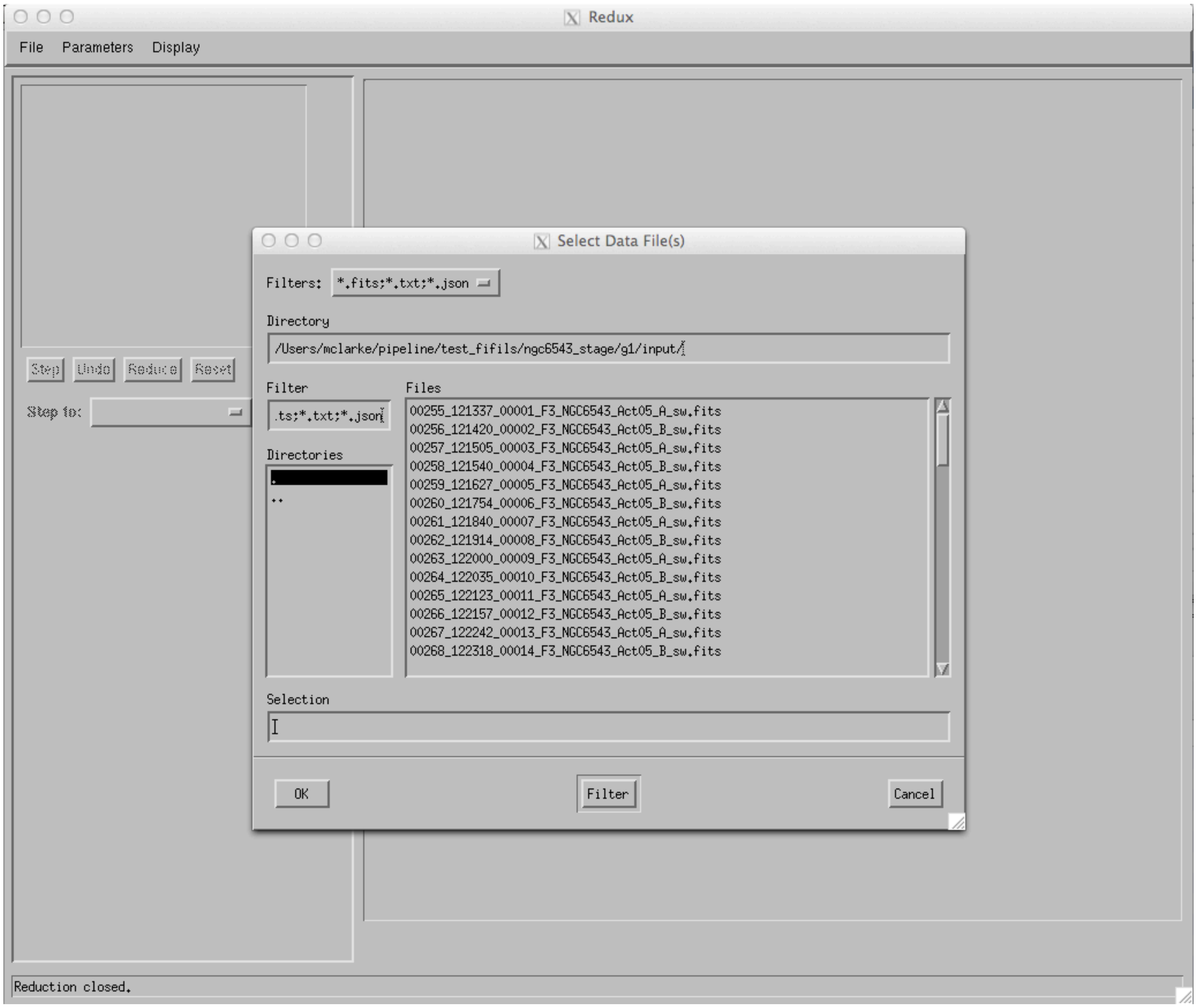
where *param.json* is a file that lists parameter keywords and values in JSON format for any step in the pipeline reduction (see Appendix A). Parameter files can be generated interactively with the GUI (see below), then saved and fed to the automatic pipeline for batch reduction of a large number of files.

6.5 Manual mode execution

It is also possible to run the pipeline interactively, using a graphical user interface. The IDL command `redux`, called without arguments, will launch the Redux GUI.

6.5.1 Basic workflow

To start an interactive reduction, select a set of FIFI-LS files, using the File menu (**File->Open New Reduction**). This will bring up a file dialog window (see figure, below). All files selected will be reduced together as a single reduction set.



Filters: *.fits;*.txt;*.json

Directory: /Users/mclarke/pipeline/test_fifils/ngc6543_stage/g1/input/

Filter: .ts;*.txt;*.json

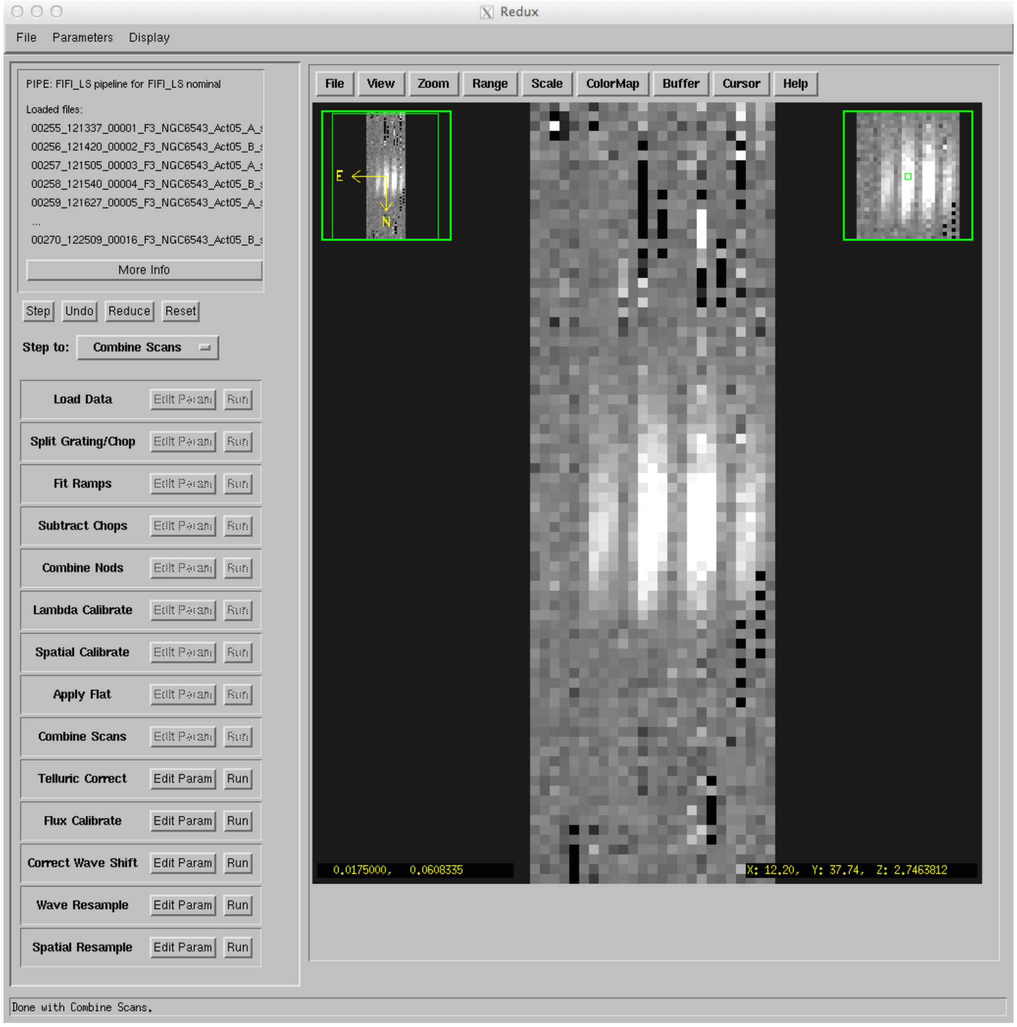
Files:

```
00255_121337_00001_F3_NGC6543_Act05_A_sw.fits
00256_121420_00002_F3_NGC6543_Act05_B_sw.fits
00257_121505_00003_F3_NGC6543_Act05_A_sw.fits
00258_121540_00004_F3_NGC6543_Act05_B_sw.fits
00259_121627_00005_F3_NGC6543_Act05_A_sw.fits
00260_121754_00006_F3_NGC6543_Act05_B_sw.fits
00261_121840_00007_F3_NGC6543_Act05_A_sw.fits
00262_121914_00008_F3_NGC6543_Act05_B_sw.fits
00263_122000_00009_F3_NGC6543_Act05_A_sw.fits
00264_122035_00010_F3_NGC6543_Act05_B_sw.fits
00265_122123_00011_F3_NGC6543_Act05_A_sw.fits
00266_122157_00012_F3_NGC6543_Act05_B_sw.fits
00267_122242_00013_F3_NGC6543_Act05_A_sw.fits
00268_122318_00014_F3_NGC6543_Act05_B_sw.fits
```

Reduction closed.

Figure 18: Open New Reduction

Redux will decide the appropriate reduction steps from the input files, and load them into the GUI, as in the figure below.



PIPE: FIFI_LS pipeline for FIFI_LS nominal

Loaded files:

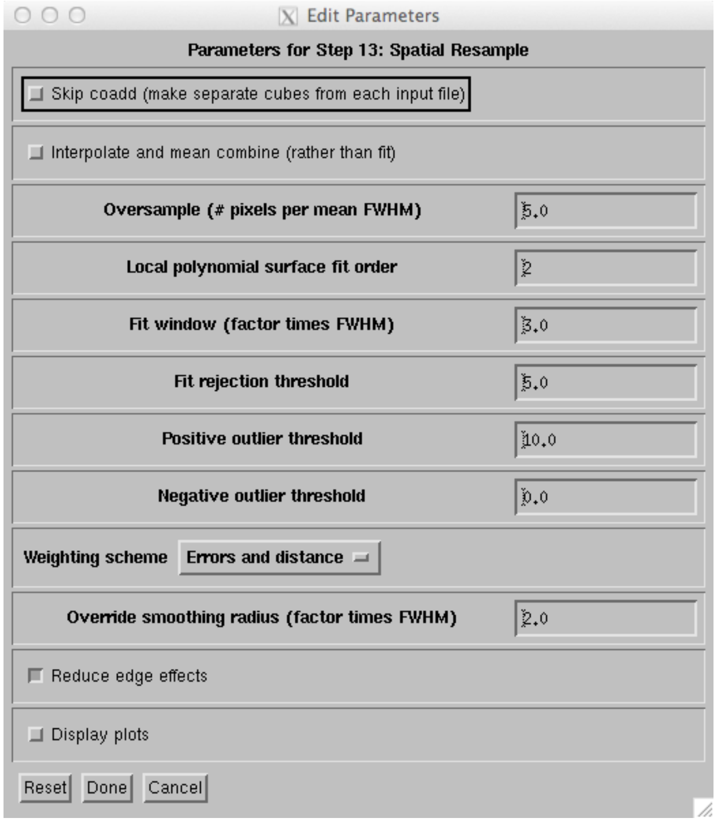
```
00255_121337_00001_F3_NGC6543_Act05_A_s
00256_121420_00002_F3_NGC6543_Act05_B_s
00257_121505_00003_F3_NGC6543_Act05_A_s
00258_121540_00004_F3_NGC6543_Act05_B_s
00259_121627_00005_F3_NGC6543_Act05_A_s
...
00270_122509_00016_F3_NGC6543_Act05_B_s
```

```
0.0175000,   0.0608335
X: 12.20,  Y: 37.74,  Z: 2.7463812
Done with Combine Scans.
```

Figure 19: Sample reduction step. Intermediate products are displayed in their current processing state after each data reduction step.

Each reduction step has a number of parameters that can be edited before running the step. To examine or edit these parameters, click the **Edit Param** button next to the step name to bring up the parameter editor for that step. Within the parameter editor, all values may be edited; clicking **Done** will save the edited values and close the window. Clicking **Reset** will restore any edited values to their defaults; clicking **Cancel** will discard all changes to the parameters and close the editor window.

The current set of parameters can be displayed, saved to a file, or reset all at once using the **Parameters** menu. A previously saved set of parameters can also be restored for use with the current reduction (**Parameters->Load Parameters**).



Edit Parameters

Parameters for Step 13: Spatial Resample

□ Skip coadd (make separate cubes from each input file)

□ Interpolate and mean combine (rather than fit)

Oversample (# pixels per mean FWHM): 5.0

Local polynomial surface fit order: 2

Fit window (factor times FWHM): 3.0

Fit rejection threshold: 5.0

Positive outlier threshold: 10.0

Negative outlier threshold: 0.0

Weighting scheme: Errors and distance

Override smoothing radius (factor times FWHM): 2.0

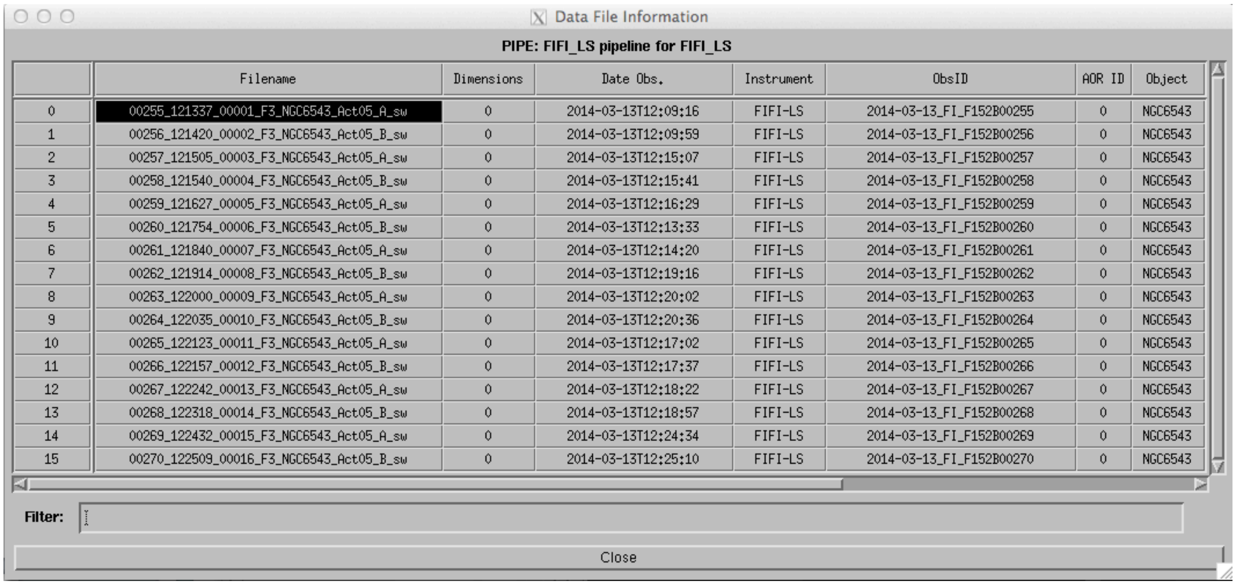
☑ Reduce edge effects

□ Display plots

Figure 20: Sample parameter editor (for Spatial Resample step)

After all parameters for a step have been examined and set to the user's satisfaction, a processing step can be run on all loaded files either by clicking **Step**, or the **Run** button next to the step name. Each processing step must be run in order, but if a processing step is selected in the **Step to:** widget, then clicking **Step** will treat all steps up through the selected step as a single step. When a step has been completed, its buttons will be grayed out and inaccessible. It is possible to undo one previous step by clicking **Undo**. All remaining steps can be run at once by clicking **Reduce**. After each step, the results of the processing will be displayed in the display window. Clicking **Reset** will restore the reduction to the initial state, without resetting parameter values.

Files can be added to the reduction step (**File->Add Files**) or removed from the reduction set (**File->Remove Files**), but either action will reset the reduction for all loaded files. Selecting **Display->Display File Information**, or the **More Info** button, will pull up a table of information about the currently loaded files (Figure 21). The table rows displayed can be filtered by entering a search string into the **Filter** text box.



PIPE: FIFI_LS pipeline for FIFI_LS

	Filename	Dimensions	Date Obs.	Instrument	ObsID	AOR ID	Object
0	00255_121337_00001_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:09:16	FIFI-LS	2014-03-13_FI_F152B00255	0	NGC6543
1	00256_121420_00002_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:09:59	FIFI-LS	2014-03-13_FI_F152B00256	0	NGC6543
2	00257_121505_00003_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:15:07	FIFI-LS	2014-03-13_FI_F152B00257	0	NGC6543
3	00258_121540_00004_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:15:41	FIFI-LS	2014-03-13_FI_F152B00258	0	NGC6543
4	00259_121627_00005_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:16:29	FIFI-LS	2014-03-13_FI_F152B00259	0	NGC6543
5	00260_121754_00006_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:13:33	FIFI-LS	2014-03-13_FI_F152B00260	0	NGC6543
6	00261_121840_00007_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:14:20	FIFI-LS	2014-03-13_FI_F152B00261	0	NGC6543
7	00262_121914_00008_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:19:16	FIFI-LS	2014-03-13_FI_F152B00262	0	NGC6543
8	00263_122000_00009_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:20:02	FIFI-LS	2014-03-13_FI_F152B00263	0	NGC6543
9	00264_122035_00010_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:20:36	FIFI-LS	2014-03-13_FI_F152B00264	0	NGC6543
10	00265_122123_00011_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:17:02	FIFI-LS	2014-03-13_FI_F152B00265	0	NGC6543
11	00266_122157_00012_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:17:37	FIFI-LS	2014-03-13_FI_F152B00266	0	NGC6543
12	00267_122242_00013_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:18:22	FIFI-LS	2014-03-13_FI_F152B00267	0	NGC6543
13	00268_122318_00014_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:18:57	FIFI-LS	2014-03-13_FI_F152B00268	0	NGC6543
14	00269_122432_00015_F3_NGC6543_Act05_A_sw	0	2014-03-13T12:24:34	FIFI-LS	2014-03-13_FI_F152B00269	0	NGC6543
15	00270_122509_00016_F3_NGC6543_Act05_B_sw	0	2014-03-13T12:25:10	FIFI-LS	2014-03-13_FI_F152B00270	0	NGC6543

Figure 21: File information table

6.5.2 Display features

Redux displays images using ximgtool, a full-featured display tool distributed with FSpextool. For more information, see the ximgtool help file, available from Redux via the **Help** button just above the display. See the table below for a quick listing of the most useful ximgtool features.

Feature	**Menu button**	**Keyboard shortcut**
Load new file	**File->Load FITS**	--
Load file into new frame	**File->New Frame**	--
View FITS header	**File->View Header**	--
Zoom	**Zoom->Zoom In**, **Zoom Out**, **Zoom To Fit**	Press **z** to enter zoom mode, then **i** to zoom in, **o** to zoom out, or **t** to zoom to fit.
Color stretch	**Cursor->Stretch**	Press **s** to enter stretch mode, click and drag to change brightness and contrast.
Set display range	**Cursor->Range**	Press **r** to enter range mode, click and drag to select the box that sets the display range.
Display distance	--	Press **d** to enter distance mode, then click and drag to identify start and end points.
Line cut	--	Press **l** to enter line cut mode, then click and drag to identify start and end points. A single click and release will do a cut through the spectral dimension.
Display image statistics	--	Press **m** to enter moments mode, then click and drag to identify box for which the statistics should be calculated.
Photometry	--	Press **a** over a star to do basic photometry.

Clear current mode	--	Press **c**.
Buffer select	**Buffer->Buffer 1, Buffer 2…**	Press **f** to move to the next buffer, **b** to move to the previous buffer.
Buffer math	**Buffer->Buffer Math**, then select buffers and arithmetic operation to perform	--
Blink buffers	**Buffer->Blink Buffers**	--
Cycle frames	**Buffer->Cycle frames**	Press **n** to move to the next available frame, **p** to move to the previous frame.

Table 3: Ximgtool features

Ximgtool has five buffers available for simultaneous display of images. If there are more than five images loaded into Redux, they can be viewed by selecting **Display->Quick Look** from the Redux menu. This will cycle through each data file in its current processing state, allowing interaction and analysis with each image in turn. To move between images, click the **Next File** or **Previous File** buttons, below the image. Click **Cancel** to quit the quick look display.

After the Combine Scans step, spectra from the brightest average spaxel in each file are displayed using xvspec (Figure 22), another display tool packaged with FSpextool. This tool also has a number of useful features – see Table 4.

Feature	**Menu button**	**Keyboard shortcut**
View FITS header	**File->View Header**	--
View select	**View->Flux** for flux data, **View-> Uncertainty** for error on the flux, **View-> S/N** for signal to noise ratio. If available, **View->Transmission** for transmission spectra and **View->Response** for instrumental response spectra.	Tab to move to the next view; shift-Tab to move to the previous view.
Spectrum select	**Spectrum->Select All** to select or unselect all loaded spectra, or **Spectrum->1** to select the first spectrum, for example.	--
Set units	**Units->Wavelength->pixels** to display the frame number for each data point.	--
Overplot reference data	**Plot->Overplot->Line List** for reference wavelengths. **Plot->Overplot->Error Bars** to show the error spectrum as error bars on the flux. **Plot->Overplot->Transmission** for reference atmospheric transmission. **Plot->Overplot->Response** for reference instrumental response. **Plot->Overplot->Model** for reference standard	--

	model.	
Zoom	--	Press **z** to zoom on both axes, **y** to zoom on y-axis only, **x** to zoom on x only. After entering zoom mode, press **i** to zoom in, or **o** to zoom out, or click to set the bounding box. Press **w** to restore original axes.
Fit line	--	Press **f** to enter fitting mode, then click to set bounding box. Fit centroid and FWHM for the selected region will be displayed in the bar below the plot. Additional statistics are printed to the terminal window.
Clear current mode	--	Press **c** or **q**.

Table 4: Xvspec features

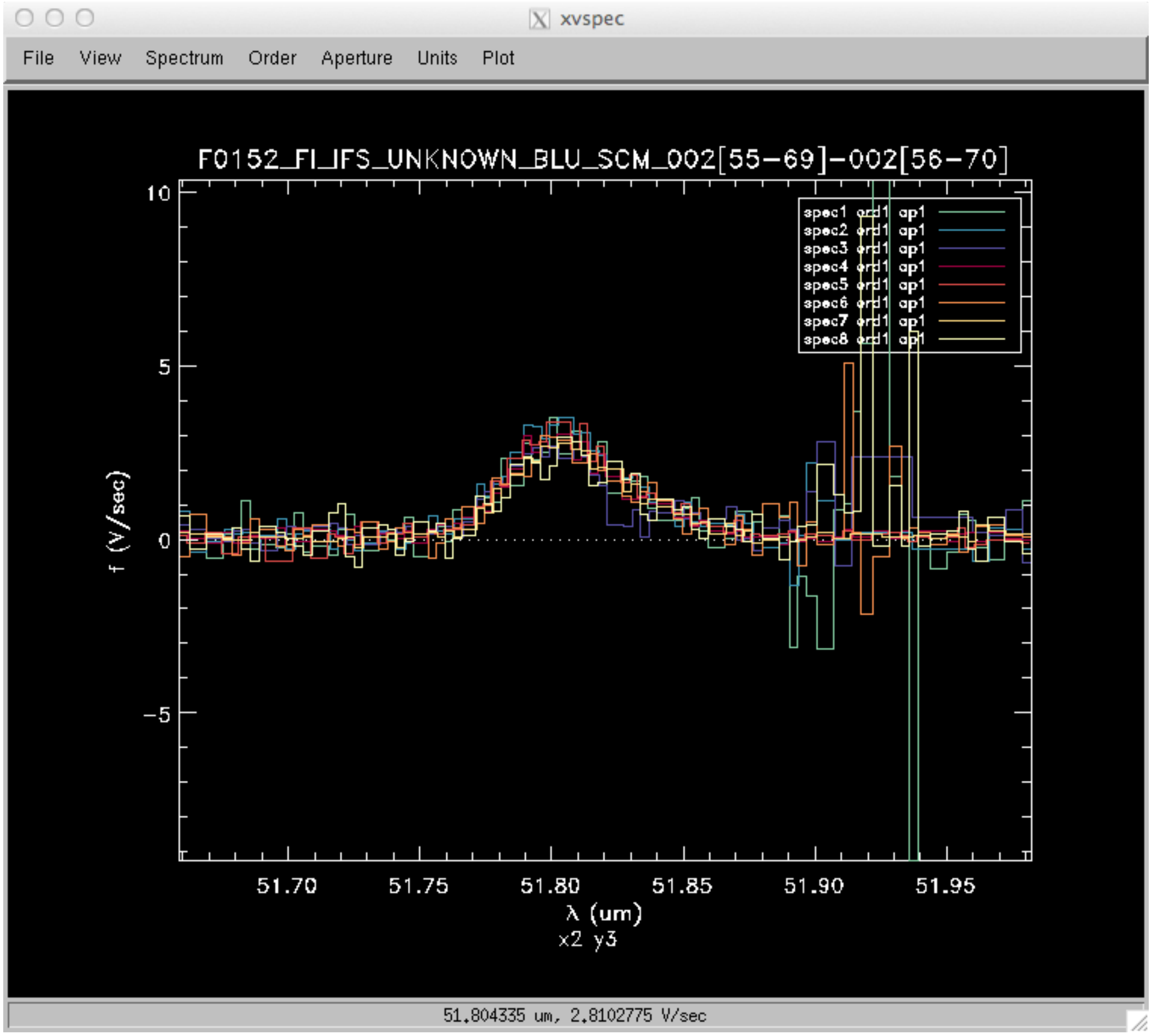


Figure 22: Spectrum from the brightest spaxel of each input image, displayed in xvspec.

6.6 FIFI-LS Reduction

FIFI-LS data reduction with Redux follows the data reduction flowchart given in section 3. At each step, Redux attempts to determine automatically the correct action, given the input data and default parameters, but each step can be customized as needed.

Some key parameters to note are listed below.

- **Load Data**
  - *Save intermediate files*: Check this box to save the files produced after each reduction step.
  - *Check header keywords*: By default, Redux will abort the reduction if the input header keywords do not meet requirements. Uncheck this box to attempt the reduction anyway.

  - *Use parallel processing*: By default, Redux will attempt to use all CPUs available (up to 12) to process data in parallel. Uncheck this box to use serial processing on a single CPU instead. Serial reduction is useful for debugging pipeline errors, and may be necessary for some architectures and/or IDL versions.
- **Fit Ramps**
  - *Signal-to-Noise Threshold*: This value defines the signal-to-noise cut-off to flag a ramp as bad and ignore it. Set to -1 to skip the signal-to-noise cut.
  - *Robust Combination Threshold (sigma)*: This value defines the rejection threshold for the robust mean combination of the ramp slopes. Set higher to reject fewer ramps, lower to reject more.
  - *Override default bad pixel file*: By default, the pipeline looks up a bad pixel mask in *fifi-ls/data/badpix_files*. To override the default mask, use this parameter to select a different text file. The file must be an ASCII table with two columns: the spaxel number (1-25, numbered left to right in displayed array), and the spexel number (1-16, numbered bottom to top in displayed array). This option may be used to block a bad spaxel for a particular observation, by entering a bad pixel for every spexel index in a particular spaxel.
  - *Remove first ramp(s)*: Select to remove the first ramp. This is on by default; if it is determined that the first ramp is okay to be used, it may be turned off.
  - *Display plots*: Select to display plots of ramp fits from the central spaxel. If selected, processing will be performed serially for this step.
- **Combine Nods**
  - *Make off-beam image instead of on-beam*: If selected, B beams will be treated as if they were A beams for all subsequent reductions. That is, the map will be produced at the B position instead of the A. All other filenames and header settings will remain the same, so use with caution.
- **Spatial Calibrate**
  - *Rotate by detector angle*: By default, Redux rotates the data by the detector angle to set North up and East to the left in the final map. Deselect this box to keep the final map in detector coordinates.
  - *DLAM/DBET sign convention (+, -, or default)*: For most data, the sign convention of the DLAM_MAP and DBET_MAP header keywords, which define the dither offsets, is determined automatically (parameter value “default”). Occasionally, for particular observations, these keywords may need their signs flipped (“-”), or used as is (“+”). This is usually determined by inspection of the results of the Spatial Resample step.
- **Apply Flat**
  - *Skip Flat Correction*: Check this box to skip flat-fielding the data. This option is mostly used for testing.
  - *Override default flat file*: Use this option to select a FITS file containing a normalized spectral cube to use as the flat file. This option is mostly used for testing.
- **Combine Scans**
  - *Correct bias offset between scans*: Check this box to subtract an overall bias offset between the individual scans.

  - *Display plots*: Select to display a plot of the individual and combined scans for each spaxel. If selected, processing will be performed serially for this step.
- **Telluric Correct**
  - *Skip telluric correction*: Select to skip correcting the data for telluric absorption. This option is mostly used for testing.
  - *Override default ATRAN file*: Use this option to select a FITS file containing a transmission spectrum to use in place of the default ATRAN model that matches the data. This file will be used to correct all input data files; if different models are desired for a set of reduction data, they should be loaded and processed up through telluric correction and/or flux calibration separately. The products may then be loaded together to finish processing into a final map.
  - *Cutoff transmission level*: Modify to adjust the transmission value below which the telluric-corrected data will be set to NaN.
  - *Display plots*: Select to display plots of the telluric-corrected spectra, in comparison to the raw spectra. If selected, processing will be performed serially for this step.
- **Flux Calibrate**
  - *Skip flux calibration*: Select to skip flux calibration of the data. The flux will remain in instrumental units (V/sec), with PROCSTAT=LEVEL_2.
  - *Override default response file*: Use this option to select a FITS file containing an instrumental response spectrum to use in place of the default file on disk.
  - *Display plots*: Select to display plots of the flux-calibrated spectra, in comparison to the raw spectra. If selected, processing will be performed serially for this step.
- **Correct Wave Shift**
  - *Skip wavelength shift correction*: Select to skip applying the correction to the wavelength calibration due to barycentric velocity. In this case, both telluric-corrected and uncorrected cubes will be resampled onto the original wavelengths.
- **Wave Resample**
  - *Oversample*: This parameter controls the resolution of the output wavelength grid. The value is given in terms of pixels per average FWHM for the detector channel at the wavelength used.
  - *Local polynomial fit order*: This parameter controls the order of the fit to the data at each grid point. Higher orders give more fine-scale detail, but are more likely to be unstable. Set to zero to do a weighted mean of the nearby data.
  - *Fit window*: This parameter controls how much data to use in the fit at each grid point. It is given in terms of a fraction of the average spectral FWHM. Higher numbers will lead to more smoothing in the output data.
  - *Fit rejection threshold*: If the fit value is more than this number times the standard deviation away from the weighted mean, the weighted mean is used instead of the fit value. This parameter is used to reject bad fit values.
  - *Positive outlier threshold*: Sets the rejection threshold for the input data, in sigma.
  - *Negative outlier threshold*: If non-zero, sets a separate rejection threshold for negative fluxes, to be used in a first-pass rejection.
  - *Weighting scheme*: Selects the weighting scheme for the fits. Options are errors and distance (default), errors only, distance only, or no weighting.

  - *Display plots*: Select to display a plot of the actual and resampled flux for each spaxel. If selected, processing will be performed serially for this step.
- **Spatial Resample**
  - *Skip coadd*: If selected, a separate flux cube will be made from each input file, using the interpolate and mean combine algorithm. This option is useful for identifying bad input files.
  - *Interpolate and mean combine*: If set, an alternate resampling algorithm will be used, rather than the local polynomial surface fits. This option may be preferable for data with small dither offsets.
  - *Oversample*: This parameter controls the resolution of the output spatial grid. The value is given in terms of pixels per reference FWHM for the detector channel used. For the BLUE camera, the reference FWHM is 5.0 arcseconds; for RED, it is 10.0 arcseconds.
  - *Local polynomial surface fit order*: This parameter controls the order of the surface fit to the data at each grid point. Higher orders give more fine-scale detail, but are more likely to be unstable. Set to zero to do a weighted mean of the nearby data.
  - *Fit window*: This parameter controls how much data to use in the fit at each grid point. It is given in terms of a factor times the average spatial FWHM. Higher values will lead to more smoothing in the output data; too-low values may result in missing data (holes) in the output map.
  - *Fit rejection threshold*: If the fit value is more than this number times the standard deviation away from the weighted mean, the weighted mean is used instead of the fit value. This parameter is used to reject bad fit values.
  - *Positive outlier threshold*: Sets the rejection threshold for the input data, in sigma.
  - *Negative outlier threshold*: If non-zero, sets a separate rejection threshold in sigma for negative fluxes, to be used in a first-pass rejection.
  - *Weighting scheme*: Selects the weighting scheme for the fits. Options are errors and distance (default), errors only, distance only, or no weighting.*Override smoothing radius*: If distance weights are used, the smoothing radius is set to the fit window by default. Lowering this value results in finer detail for the same input fit window. Too low values may result in noisy output data; too high values effectively negate the distance weights.
  - *Reduce edge effects*: If selected, the order of the fit will be lowered near the edges of the image, to reduce bad pixels due to unstable fits near the edges.
  - *Display plots*: Select to display a plot of the actual and resampled flux surface for each wavelength plane. If selected, processing will be performed serially for this step.

7 DATA QUALITY ASSESSMENT

After the pipeline has been run on a set of input data, the output products should be checked to ensure that the data has been properly reduced:

- Check the output to the terminal (or the log, in the case where the pipeline has been run by the automatic DPS system) for warnings or errors. Non-fatal warnings will be prepended with the string *WARNING*. Fatal errors will be prepended with the string *ERROR*.

- Check that the expected files were written to disk: there should, at a minimum, be a scan-combined file (*SCM*), a flux-calibrated file (*CAL*), a wavelength-resampled file (*WGR*) and a spatial-resampled file (*WXY*).
- Look at the spectrum of each spaxel in the *WGR* file. Verify that there are no bad spaxels, i.e. that the spectrum looks reasonably well behaved in each spaxel. If a spaxel seems to have wildly varying values, an alternate flat field may need to be used.
- Look at each plane of the reduced image in the *WXY* file. Check that the spatial resampling seems to have completed successfully: there should not be excessive holes in the map, or bad pixels away from the edges of the image. If there are, the spatial resampling may need to be redone with modified parameters.
- Look at the spectra for a sampling of spatial pixels in the *WXY* file. Check that there are no sudden dropouts or other discontinuities in the spectrum that are not associated with poor atmospheric transmission. If there are such discontinuities, the wavelength resampling may need to be redone with modified parameters.

8 APPENDIX A: SAMPLE CONFIGURATION FILES

Sample FIFI-LS Redux parameter override file in JSON format, with all values stored as strings. If present, the parameter value overrides the default defined by the FIFI-LS reduction object. If not present, the default value will be used. The parameters displayed here are the current default values.

```
{
 "FIFI_LS_REDUCTION":
  {
   "Load Data":
    {
     "procpath": ".",
     "verbose": "0",
     "save_intermediate": "0",
     "checkreq": "1",
     "parallel": "1"
    },
   "Split Grating/Chop":
    {
    },
   "Fit Ramps":
    {
     "s2n": "30",
     "thresh": "5",
     "badpixfile": "",
     "plot_this": "0"
    },
   "Subtract Chops":
    {
    },
   "Combine Nods":
    {
    },
   "Lambda Calibrate":
    {
    },
   "Spatial Calibrate":
    {
     "rotate": "1",
     "flipsign": "default"
    },
   "Apply Flat":
    {
     "skipflat": "0",
     "flatfile": ""
    },
   "Combine Scans":
    {
     "bias": "1",
     "plot_this": "0"
    },
   "Telluric Correct":
    {
     "skiptell": "0",
     "atranfile": "",
     "cutoff": "0.6",
     "plot_this": "0"
    },
   "Flux Calibrate":
    {
     "skipcal": "0",
     "respfile": "",
     "plot_this": "0"
    },
   "Correct Wave Shift":
    {
     "skipshift": "0"
    },
   "Wave Resample":
    {
     "oversample": "8.0",
     "order": "2",
     "window": "0.25",
     "fitthresh": "2.0",
     "posthresh": "5.0",
     "negthresh": "4.0",
     "weight": "Errors and distance",
     "plot_this": "0"
    },
   "Spatial Resample":
    {
     "skipcoadd": "0",
     "interp": "0",
     "oversample": "5.0",
     "order": "2",
     "window": "3.0",
     "fitthresh": "5.0",
     "posthresh": "10.0",
     "negthresh": "0.0",
     "weight": "Errors and distance",
     "smoothing": "2.0",
     "edge": "1",
     "plot_this": "0"
    }
  }
}
```

Sample FIFI-LS configuration file, located in *fifi-ls/data/header_info/headerdef.dat*. Values marked with a Y in the *reqd?* column are keywords required to be present in input data. They must meet the type and range requirements listed for grouping and data reduction to be successful.

```
#
# This table lists keywords, allowed values, and defaults for
# all keywords to be written to output files.  Keywords
# required to be present in input files for correct data
# processing are marked with a 'Y'. A '.' in min, max, or enum
# means no requirement.  The combine column designates the
# algorithm to be used to calculate the value for the
# header of a product made from multiple input files.
#
# keyword  reqd?  default  type    combine     min    max     enum
#------------------------------------------------------------------
AIRSPEED     N    -9999.   float   first       .      .       .
ALTI_END     Y    -9999.   float   last        0.     60000.  .
ALTI_STA     Y    -9999.   float   first       0.     60000.  .
AOR_ID       N    UNKNOWN  string  first       .      .       .
AOT_ID       N    UNKNOWN  string  first       .      .       .
ASSC_AOR     N    UNKNOWN  string  concatenate .      .       .
ASSC_MSN     N    UNKNOWN  string  concatenate .      .       .
ASSC_OBS     N    UNKNOWN  string  concatenate .      .       .
ATRNFILE     N    UNKNOWN  string  concatenate .      .       .
BDPXFILE     N    UNKNOWN  string  concatenate .      .       .
BGLEVL_A     N    -9999.   float   mean        .      .       .
BGLEVL_B     N    -9999.   float   mean        .      .       .
C_CHOPLN     Y    -9999    int     first       7      256     .
C_SCHEME     Y    UNKNOWN  string  first       .      .       2POINT
CHOPPING     Y    T        bool    first       .      .       .
CHPAMP1      N    -9999.   float   first       -1125  1125    .
```

```
CHPAMP2      N    -9999.   float   first       -1125  1125    .
CHPANGLE     N    -9999.   float   first       -360.  360.    .
CHPCRSYS     N    UNKNOWN  string  first       .      .       TARF|ERF|SIRF
CHPFREQ      Y    -9999.   float   first       0.25   25.     .
CHPPHASE     N    -9999    int     first       0      1000    .
CHPPROF      N    UNKNOWN  string  first       .      .       2-POINT|3-POINT
CHPSYM       N    UNKNOWN  string  first       .      .       .
CHPTIP       N    -9999.   float   first       -301.  301.    .
CHPTILT      N    -9999.   float   first       -301.  301.    .
CREATOR      N    UNKNOWN  string  first       .      .       .
DATAQUAL     N    UNKNOWN  string  first       .      .       NOMINAL|USABLE|TEST|PROBLEM|FAIL
DATASRC      Y    UNKNOWN  string  first       .      .       ASTRO|CALIBRATION|LAB|TEST|OTHER|FIRSTPOINT
DATATYPE     N    UNKNOWN  string  first       .      .       IMAGE|SPECTRAL|OTHER
DATE         N    UNKNOWN  string  first       .      .       .
DATE-OBS     Y    UNKNOWN  string  first       .      .       .
DBET_MAP     Y    -9999.   float   first       -36000 36000   .
DEPLOY       N    UNKNOWN  string  first       .      .       .
DETCHAN      Y    UNKNOWN  string  first       .      .       BLUE|RED
DETECTOR     N    UNKNOWN  string  first       .      .       .
DETSIZE      N    UNKNOWN  string  first       .      .       .
DICHROIC     Y    -9999    int     first       .      .       105|130
DITHER       N    F        bool    first       .      .       .
DLAM_MAP     Y    -9999.   float   first       -36000 36000   .
DTHCRSYS     N    UNKNOWN  string  first       .      .       TARF|ERF|SIRF
DTHINDEX     N    -9999    int     default     .      .       .
DTHNPOS      N    -9999    int     first       .      .       .
DTHOFFS      N    -9999.   float   first       .      .       .
DTHPATT      N    UNKNOWN  string  first       .      .       NONE|3-POINT|5-POINT|9-POINT|CUSTOM
DTHXOFF      N    -9999.   float   first       .      .       .
DTHYOFF      N    -9999.   float   first       .      .       .
EQUINOX      N    -9999.   float   first       .      .       .
EXPTIME      Y    -9999.   float   sum         0.02   1000    .
FBC-STAT     N    UNKNOWN  string  last        .      .       FBC_OFF|FBC_QS|FBC_DY|FBC_ON
FILENAME     Y    UNKNOWN  string  first       .      .       .
FILENUM      N    UNKNOWN  string  first       .      .       .
FILEREV      N    UNKNOWN  string  first       .      .       .
FLATFILE     N    UNKNOWN  string  concatenate .      .       .
FLIGHTLG     N    UNKNOWN  string  first       .      .       .
FOCUS_EN     N    -9999.   float   last        -5000  5000    .
FOCUS_ST     N    -9999.   float   first       -5000  5000    .
G_CYC_B      Y    -9999    int     first       0      100     .
G_CYC_R      Y    -9999    int     first       0      100     .
G_ORD_B      Y    -9999    int     first       1      2       .
G_PSDN_B     Y    -9999    int     first       0      100     .
G_PSDN_R     Y    -9999    int     first       0      100     .
G_PSUP_B     Y    -9999    int     first       0      100     .
G_PSUP_R     Y    -9999    int     first       0      100     .
G_STRT_B     Y    -9999    long    first       0      2098176 .
G_STRT_R     Y    -9999    long    first       0      2098176 .
G_SZDN_B     Y    -9999    long    first       0      20000   .
G_SZDN_R     Y    -9999    long    first       0      20000   .
G_SZUP_B     Y    -9999    long    first       0      20000   .
G_SZUP_R     Y    -9999    long    first       0      20000   .
GRDSPEED     N    -9999.   float   first       .      .       .
HEADING      N    -9999.   float   first       .      .       .
HEADSTAT     N    UNKNOWN  string  first       .      .       ORIGINAL|UNKNOWN|CORRECTED|ERROR|MODIFIED
IMAGEID      N    -9999    int     default     0      .       .
INSTCFG      N    UNKNOWN  string  first       .      .       DUAL_CHANNEL
INSTMODE     N    UNKNOWN  string  first       .      .
SYMMETRIC_CHOP|ASYMMETRIC_CHOP|BRIGHT_OBJECT|SPECTRAL_SCAN
INSTRUME     Y    FIFI-LS  string  first       .      .       FIFI-LS
KWDICT       N    UNKNOWN  string  first       .      .       .
LASTREW      N    UNKNOWN  string  first       .      .       .
LAT_END      N    -9999.   float   last        .      .       .
LAT_STA      N    -9999.   float   first       .      .       .
LON_END      N    -9999.   float   last        .      .       .
LON_STA      N    -9999.   float   first       .      .       .
MAPCRSYS     N    UNKNOWN  string  first       .      .       EQUATORIAL|GALACTIC|ECLIPTIC|USER
MAPINTX      N    -9999.   float   first       .      .       .
```

```
MAPINTY      N    -9999.   float   first       .      .       .
MAPNXPOS     N    -9999    int     first       .      .       .
MAPNYPOS     N    -9999    int     first       .      .       .
MAPPING      N    F        bool    first       .      .       .
MCCSMODE     N    UNKNOWN  string  first       .      .       .
MISSN-ID     Y    UNKNOWN  string  first       .      .       .
N_SPEC       N    -9999    int     sum         0      .       .
NEXP         N    1        int     sum         0      .       .
NODAMP       N    -9999.   float   first       .      .       .
NODANGLE     N    -9999.   float   first       -360   360     .
NODBEAM      Y    UNKNOWN  string  default     .      .       A|B
NODCRSYS     N    UNKNOWN  string  first       .      .
ERF|ECRF|GALRF|TARF|FPIRF|FFIRF|WFIRF|SIRF|USER
NODDING      Y    T        bool    first       .      .       .
NODN         N    -9999    int     first       .      .       .
NODPATT      Y    UNKNOWN  string  first       .      .       .
NODSETL      N    -9999.   float   first       .      .       .
NODSTYLE     Y    UNKNOWN  string  first       .      .       NMC|C2NC2
NODTIME      N    -9999.   float   first       .      .       .
OBJECT       Y    UNKNOWN  string  first       .      .       .
OBS_ID       Y    UNKNOWN  string  first       .      .       .
OBSBET       N    -9999.   float   first       -90.   90.     .
OBSDEC       N    -9999.   float   first       -90.   90.     .
OBSERVER     N    UNKNOWN  string  first       .      .       .
OBSLAM       N    -9999.   float   first       0.     360.    .
OBSRA        N    -9999.   float   first       0.     24.     .
OBSTYPE      Y    UNKNOWN  string  first       .      .
OBJECT|STANDARD_FLUX|STANDARD_TELLURIC|STANDARD_WAVECAL|LAMP|FLAT|DARK|BIAS|SKY|BB|GASCELL|LASER|FOCUS_LOO
P
OPERATOR     N    UNKNOWN  string  first       .      .       .
ORIGIN       N    UNKNOWN  string  first       .      .       .
PIPELINE     N    UNKNOWN  string  first       .      .       .
PIPEVERS     N    UNKNOWN  string  first       .      .       .
PIXSCAL      N    -9999.   float   first       .      .       .
PLANID       N    UNKNOWN  string  first       .      .       .
PLATSCAL     Y    -9999    float   first       .      .       .
PROCSTAT     Y    UNKNOWN  string  first       .      .       .
PRODTYPE     N    UNKNOWN  string  first       .      .       .
RAMPLN_B     Y    -9999    int     first       0      256     .
RAMPLN_R     Y    -9999    int     first       0      256     .
RAWUNITS     N    V/sec    string  first       .      .       .
RESFILE      N    UNKNOWN  string  concatenate .      .       .
RESOLUN      N    -9999.   float   first       .      .       .
RSPNFILE     N    UNKNOWN  string  concatenate .      .       .
SCANNING     N    F        bool    first       .      .       .
SCNDEC0      N    -9999.   float   first       -90.   90.     .
SCNDECF      N    -9999.   float   last        -90.   90.     .
SCNDIR       N    -9999.   float   last        .      .       .
SCNRA0       N    -9999.   float   first       0.     24.     .
SCNRAF       N    -9999.   float   last        0.     24.     .
SCNRATE      N    -9999.   float   last        .      .       .
SIBS_X       N    -9999    int     default     .      .       .
SIBS_Y       N    -9999    int     default     .      .       .
SLIT         N    UNKNOWN  string  first       .      .       .
SPATFILE     N    UNKNOWN  string  concatenate .      .       .
SPECTEL1     Y    UNKNOWN  string  first       .      .       NONE|FIF_BLUE
SPECTEL2     Y    UNKNOWN  string  first       .      .       NONE|FIF_RED
SRCTYPE      N    UNKNOWN  string  first       .      .       POINT_SOURCE|EXTENDED_SOURCE|OTHER|UNKNOWN
SUBARRNO     N    -9999    int     first       .      .       .
TELCONF      N    UNKNOWN  string  first       .      .       .
TELEL        N    -9999.   float   first       0.     90.     .
TELEQUI      N    UNKNOWN  string  first       .      .       .
TELESCOP     N    UNKNOWN  string  first       .      .       .
TELDEC       N    -9999.   float   first       -90.   90.     .
TELLOS       N    -9999.   float   first       -180   180     .
TELRA        N    -9999.   float   first       0.     24.     .
TELVPA       N    -9999.   float   first       0.     360.    .
TELXEL       N    -9999.   float   first       -90.   90.     .
TEMP_OUT     N    -9999.   float   first       .      .       .
```

```
TEMPPRI1     N    -9999.   float   first       -273.  .       .
TEMPPRI2     N    -9999.   float   first       -273.  .       .
TEMPPRI3     N    -9999.   float   first       -273.  .       .
TEMPSEC1     N    -9999.   float   first       .      .       .
TRACERR      N    F        bool    or          .      .       .
TRACKANG     N    -9999.   float   first       .      .       .
TRACMODE     N    UNKNOWN  string  first       .      .
OFF|CENTROID|ROF|LIMB|OFFSET|ROF+OFFSET|CENTROID+INERTIAL|ROF+INERTIAL|OFFSET+INERTIAL|ROF+OFFSET+INERTIAL
TSC-STAT     N    UNKNOWN  string  last        .      .       .
UTCEND       N    UNKNOWN  string  last        .      .       .
UTCSTART     N    UNKNOWN  string  first       .      .       .
WAVECENT     N    -9999.   float   first       0.     .       .
WAVEFILE     N    UNKNOWN  string  concatenate .      .       .
WVZ_END      N    -9999.   float   last        0.     .       .
WVZ_STA      N    -9999.   float   first       0.     .       .
ZA_END       Y    -9999.   float   last        0.     90.     .
ZA_START     Y    -9999.   float   first       0.     90.     .
```