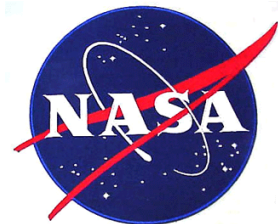


EXES Redux Pipeline Users Manual

SOF-US-HBK-OP10-2005

Date: 8 September 2021
Revision: B



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

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

EXES Redux Pipeline Users Manual

SOF-US-HBK-OP10-2005

AUTHOR:

Melanie Clarke, USRA, SOFIA DPS Development Lead

Date

William Vacca, USRA, SOFIA Deputy Associate Director for
Science Ops

Date

APPROVAL:

Ed Chambers, USRA, SOFIA DPS Science Coordinator

Date

William Vacca, USRA, SOFIA Deputy Associate Director
for Science Ops

Date

Elizabeth Moore, USRA, SOFIA Associate Director for
Software Systems

Date

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

REVISION HISTORY

REV	DATE	DESCRIPTION
-	3/30/15	Initial Release
A	2/1/16	Updates/clarifications based on experiences with processing EXES data.
B	9/8/21	Changes related to the EXES Redux Pipeline v2.0.0 release which accommodates the EXES instrument transitioning to FSI support. <ul style="list-style-type: none">- Algorithm description updates for readout, flat processing, and image coaddition- Updated software package descriptions- Updated parameter descriptions- Added change notes appendix

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

EXES Redux User's Manual

Release : SCI-US-HBK-OP10-2005 Rev. B

M. Clarke, W. Vacca

Sep 08, 2021

Contents

I	Introduction	3
II	SI Observing Modes Supported	3
1	EXES observing techniques	3
2	EXES instrument configurations	4
III	Algorithm Description	4
3	Overview of data reduction steps	4
4	Reduction algorithms	4
4.1	Correct nonlinearity and coadd readout frames	5
4.2	Process flat field	7
4.3	Despike	8
4.4	Debounce	9
4.5	Subtract sky	9
4.6	Flat correct	11
4.7	Clean bad pixels	11
4.8	Undistort	11
4.9	Coadd frames	13
4.10	Identify apertures	14
4.11	Extract spectra	15
4.12	Combine multiple spectra	17
5	Other Resources	17
IV	Flux calibration	17
V	Data products	17
6	Filenames	18

7 Pipeline Products	18
8 Data Format	20
VI Grouping LEVEL_1 data for processing	20
VII Configuration and execution	21
9 Installation	21
10 Configuration	22
11 Input data	22
12 Automatic mode execution	22
13 Manual mode execution	23
13.1 Basic workflow	23
13.2 Display features	27
14 EXES Reduction	30
VIII Data quality assessment	35
IX Appendix A: Sample configuration files	35
X Appendix B: Change notes for the EXES pipeline	45
15 Significant changes	45
15.1 EXES Redux v2.0.0	46
15.2 EXES Redux v1.6.0 (2019-04-02)	46
15.3 EXES Redux v1.0.2 (2015-07-17)	46
15.4 EXES Redux v1.0.1 (2015-05-14)	46
15.5 EXES Redux v1.0.0 (2015-03-17)	46

Part I

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 Pipeline, flux calibrate the results, 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.

This manual applies to EXES Redux version 2.0.0.

Part II

SI Observing Modes Supported

1 EXES observing techniques

EXES is a mid-infrared instrument, which implies that its observations must be taken with care in order to remove the effects of the spatially and temporally variable sky background. Unlike some other mid-infrared instruments, EXES does not make use of the chopping secondary mirror, but it does use telescope nods to alternate the object observation with a sky observation.

EXES has four basic nod patterns: stare, nod-on-slit, nod-off-slit, and map. The stare mode is simply a continuous observation of a source, with no telescope offsets to sky. This mode is primarily used for testing, rather than for science observations. In either of the nodding modes, the telescope moves in order to take frames at two different positions (the A position and the B position). For nod-along-slit mode, the frames alternate between placing the source at the bottom and top of the slit. This mode can be used for relatively compact sources (FWHM less than a quarter of the slit). For nod-off-slit mode, the frames alternate between the source and a patch of blank sky. This mode is useful for extended sources or crowded fields. Sky background for either nodding mode is removed by subtracting the two adjacent nod beams. Typically, EXES does a BA nod pattern, so that the nod pair is every second frame minus the previous frame. In map mode, the telescope starts in blank sky, then steps the source across the slit, and then returns to blank sky. The sky frames are averaged together to reduce their noise, and then subtracted from each source frame. This mode is used to map extended sources.

Before each observation sequence, EXES takes a series of calibration frames: typically a spectrum of a blackbody in the same configuration as the science observation and a dark frame. The dark frame is subtracted from the black frame, and the difference is used as a flat field for the 2D object spectrum. The black frame is also used to determine the edges of the orders in cross-dispersed modes.

2 EXES instrument configurations

EXES is a spectroscopic instrument that uses a slit to provide a spatial dimension and gratings to provide a spectral dimension. EXES has two gratings used for science: the echelle grating and the echelon. The echelle grating, used alone, produces single-order long-slit spectra. The Medium resolution configuration for EXES uses the echelle grating as the primary disperser at angles 30-65 degrees; the Low resolution configuration uses the echelle grating at angles less than 30 degrees. When the echelon grating is used as the primary disperser and the echelle grating is used as a secondary disperser, EXES produces multi-order, cross-dispersed spectra. In this configuration, the wavelength coverage and the number of orders is determined by the echelle angle. The High-Medium resolution configuration uses an echelle angle of 30-65 degrees. The High-Low resolution configuration uses an angle of less than 30 degrees.

The slit length and width for all modes can be varied depending on the observational needs, although the length is limited in the cross-dispersed modes by the requirement that echelon orders must not overlap.

EXES also has an imaging configuration, but it is used primarily for testing and for acquiring targets for spectroscopic observations. It is not used for science, so it will not be addressed in this document.

Part III

Algorithm Description

3 Overview of data reduction steps

This section will describe, in general terms, the major algorithms that the EXES Redux pipeline uses to reduce an EXES observation.

See Fig. 1 for a flowchart of the processing steps used by the pipeline.

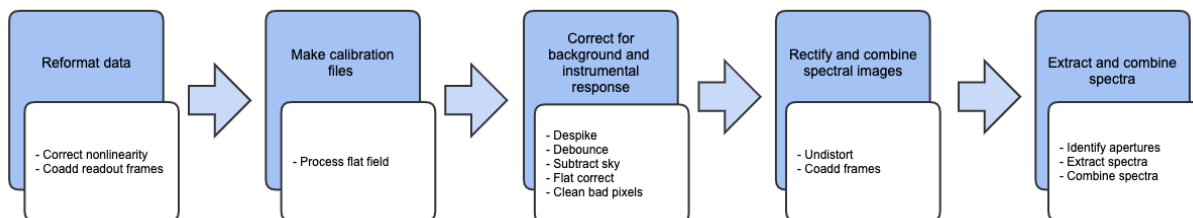


Fig. 1: Processing steps for EXES data.

4 Reduction algorithms

The following subsections detail each of the data reduction pipeline steps:

- Correct nonlinearity and coadd readout frames
- Process flat field
- Correct outliers (despike)
- Correct optical shifts (debounce)

- Subtract sky
- Flat correct
- Clean bad pixels
- Rectify spectral orders (undistort)
- Coadd frames
- Identify apertures
- Extract spectra
- Combine multiple spectra

4.1 Correct nonlinearity and coadd readout frames

Raw EXES data is stored in FITS image files, as a 3D data cube in which each plane corresponds to a digitized readout of the array. Readouts can be destructive, or non-destructive. The readout method can vary for EXES, so the pattern of readout actions used for a particular observation is recorded in the OTPAT header keyword. The value for OTPAT is a combination of the letters S, T, N, D, and C, each with a digit afterwards. The letters have the following meaning:

- S = spin (no reset, no digitization). This action effectively increases the integration time.
- T = trash (reset, no digitization). This action resets the array without storing the previous value.
- N = non-destructive read (no reset, but digitized). This action stores the current value of the array.
- D = destructive read (reset and digitization). This action stores the current value, and resets the array.
- C = coadd (reset and digitization). This action coadds previous reads in hardware and resets the array.

The digit after each letter gives the number of times the action was taken, minus one. For example: S0 is one spin, N3 is four non-destructive reads.

The pattern listed in the OTPAT can then be repeated any number of times (n_{pat}), so that the final number of frames in the raw file is $n_{pat}(n_{nd} + n_d)$, where n_{nd} is the number of nondestructive reads and n_d is the number of destructive reads per pattern.

Since near-infrared detectors do not have completely linear responses to incoming flux, each frame in the raw file may optionally be corrected for nonlinearity before proceeding. The nonlinearity coefficients for EXES were determined by taking a series of nondestructive flat exposures, followed by a single destructive read. Each readout was subtracted from a bias frame, then the resulting counts in each pixel were fit with a polynomial. The linearity correction algorithm reads in these polynomial coefficients from a 3D array and uses them to calculate a correction factor for each pixel in each raw readout. Pixels with counts below the bias level, or above the maximum value used to calculate the nonlinearity coefficients at that pixel, have no correction applied.

After linearity correction, in order to create a usable estimate of the input flux, the non-destructive and destructive reads must be combined in a way that maximizes the signal-to-noise. The EXES pipeline currently has support for combining frames taken with the patterns listed in the following subsections. Equations for the net signal and the associated variance come from Vacca et. al., 2004 (see *Other Resources*). After this initial calculation of the variance for each image pixel, the variance plane is propagated along with the image through all pipeline steps.

After the frames for each pattern are combined according to the algorithms below, the pipeline checks whether multiple patterns were taken at the same nod position. This is recorded in the header keyword NINT. If NINT is greater than 1, then NINT coadded patterns are averaged together before proceeding. Optionally, one or more of the initial patterns may be discarded before averaging.

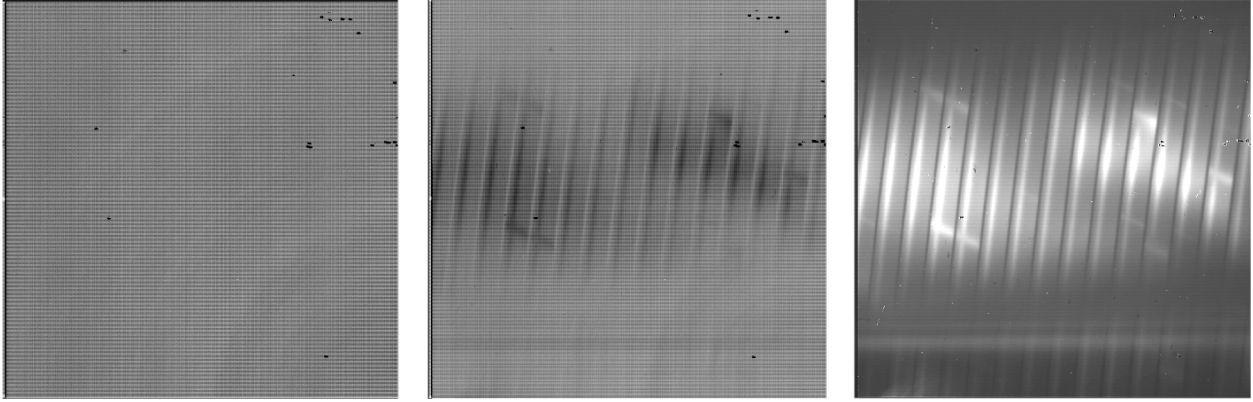


Fig. 2: Sample raw frames in nod-off-slit, High-Medium configuration: a pedestal image (left), a final destructive read (middle), and the Fowler coadd (right). The bright spectrum is visible as a dark line in the final raw readout and a bright line in the coadd.

All destructive mode

In this mode, each frame is a destructive read. There may be spins in between reads to increase integration time, but there are no non-destructive reads. In this case, the flux is simply the read signal (s), minus a standard bias frame (b) stored on disk, subtracted from a zero value (z), corresponding to a dark current, and divided by the integration time (Δt) and the pre-amp gain factor (g_p):

$$I_{net} = z - \frac{(s - b)}{\Delta t g_p}.$$

The dark current is typically zero for EXES.

The variance associated with this readout mode is the Poisson noise (first term) plus the readout noise (second term):

$$V = \frac{I_{net}}{g \Delta t} + \frac{\sigma_{read}^2}{g^2 \Delta t^2},$$

where σ_{read} is the read noise, and g is the gain factor (to convert from ADU to electrons).

This readout mode may optionally be used for any readout pattern, ignoring the non-destructive reads and using the final destructive read only.

Fowler mode

In this mode, a number of non-destructive reads are taken at the beginning of the integration, the array integrates for a while, then a number of non-destructive reads and a single destructive read are taken at the end of the integration. The initial reads are averaged together to estimate the “pedestal” level; the final reads are averaged together to estimate the total signal. The pedestal is then subtracted from the total signal to get the net signal. We then subtract the negative signal from a zero value (z) and divide by the pre-amp gain factor (g_p) and the integration time (Δt , the time between the first pedestal read and the first signal read). The net flux is then:

$$I_{net} = z - \frac{1}{n_r \Delta t g_p} \left(\sum_{i=1}^{n_r} s_i - \sum_{i=1}^{n_r} p_i \right).$$

We assume that the number of reads (n_r) is the same for the pedestal and the signal.

The variance associated with this readout mode is:

$$V = \frac{I_{net}}{g \Delta t} \left(1 - \frac{\delta t (n_r^2 - 1)}{3 \Delta t n_r} \right) + \frac{2 \sigma_{read}^2}{g^2 \Delta t^2 n_r}.$$

Here, δt is the frame time, σ_{read} is the read noise, and g is the gain factor to convert from ADU to electrons.

Sample OTPAT values for Fowler mode look like this:

- **N0 D0**: a single non-destructive/destructive pair. This is the minimal Fowler case: the first read is the pedestal, the second read is the signal. There are 2 frames per pattern.
- **N3 S15 N2 D0**: a Fowler 4 set. This set comprises 4 non-destructive reads, 16 spins to increase the integration time, 3 non-destructive reads, and a destructive read. There are 8 frames per pattern.

Sample-up-the-ramp mode

In this mode, non-destructive reads are taken at even intervals throughout the integration. The count values for all reads are fit with a straight line; the slope of this line gives the net flux. As with the other modes, we then subtract from the zero value and divide by the pre-amp gain and the integration time. The net flux for the evenly-sampled case is calculated as:

$$I_{net} = z - \frac{12}{n_r (n_r + 1) \Delta t g_p} \left(\sum_{i=1}^{n_r} s_i \left[i - \frac{n_r + 1}{2} \right] \right),$$

where n_r is the total number of reads (non-destructive and destructive).

The variance associated with this readout mode is:

$$V = \frac{6 I_{net} (n_r^2 + 1)}{5 g \Delta t n_r (n_r + 1)} + \frac{12 \sigma_{read}^2 (n_r - 1)}{g^2 \Delta t^2 n_r (n_r + 1)}.$$

Sample OTPAT values for sample-up-the-ramp mode look like this:

- **N0 S3 N0 S3 N0 S3 D0**: Sample-up-the-ramp with 4 spins in between each non-destructive read, ending with a destructive read. There are 4 frames per pattern.

Coadding

If the OT pattern contains a coadd (C), it is treated as a destructive read for which the appropriate combination has already been done. That is, if the pattern indicates Fowler mode, for example, the intensity is simply calculated as for a destructive read:

$$I_{net} = z - \frac{s}{\Delta t g_p},$$

but the variance is calculated from the net intensity as for the Fowler mode, above.

4.2 Process flat field

Each EXES reduction requires a flat observation taken in the same mode as the science data. The instrument configuration (eg. HIGH_MED or MEDIUM), central wavenumber, slit length, slit width, and grating angle must all match. This flat observation is used to correct for spectral and spatial instrumental gain variations, and to calibrate the source intensity of the science observation.

The flat observation consists of an observation of a blackbody in EXES's calibration box (*black*) and a dark frame (*dark*). If the blackbody's temperature is known, then dividing the sky-subtracted science frame by *black-dark*, normalized by the blackbody function, gives a calibrated intensity as follows (Lacy et. al., 2002, see *Other Resources*).

$$\begin{aligned} S_\nu(\text{black} - \text{dark}) &= B_\nu(T_{\text{black}}) R_\nu \\ S_\nu(\text{obj} - \text{sky}) &\approx I_\nu(\text{obj}) R_\nu \\ I_\nu(\text{obj}) &\approx S_\nu(\text{obj} - \text{sky}) \frac{B_\nu(T_{\text{black}})}{S_\nu(\text{black} - \text{dark})} \end{aligned}$$

where S_ν is the measured signal, $B_\nu(T)$ is the blackbody function at temperature T , and R_ν is the instrumental responsivity. The master flat frame produced by the EXES pipeline, therefore, is the blackbody function calculated at the temperature recorded in the flat frame's header (BB_TEMP), divided by *black-dark* (Fig. 3).

In all instrument configurations, the *black* frame is also used to determine an illumination mask that defines the good pixels for extraction. In the cross-dispersed modes, the *black* frame is further used to determine the edges of the orders for extraction and to check the optical distortion parameters. The pipeline cleans and undistorts the *black* frame, takes its derivative, then performs a 2D FFT of the resulting image to determine the spacing and orientation of the orders on the array. In particular, the value of k_{rot} , the angle of the order rotation, is adjusted automatically at this step: it is calculated from the characteristics of the FFT, then used to recompute the undistortion of the *black* frame, and redo the FFT, until the value of k_{rot} converges, or 5 iterations are reached. This process must be closely monitored: if the *black* frame has insufficient signal, or the optical parameters used to calculate the distortion correction are insufficiently accurate, the spacing and rotation angle may be wrongly calculated at this step. These values can be manually overridden in parameters for the pipeline if necessary.

The distortion parameters and order definitions, as determined from the *black* frame, are written to the header of the master flat, to be used in later reduction steps.

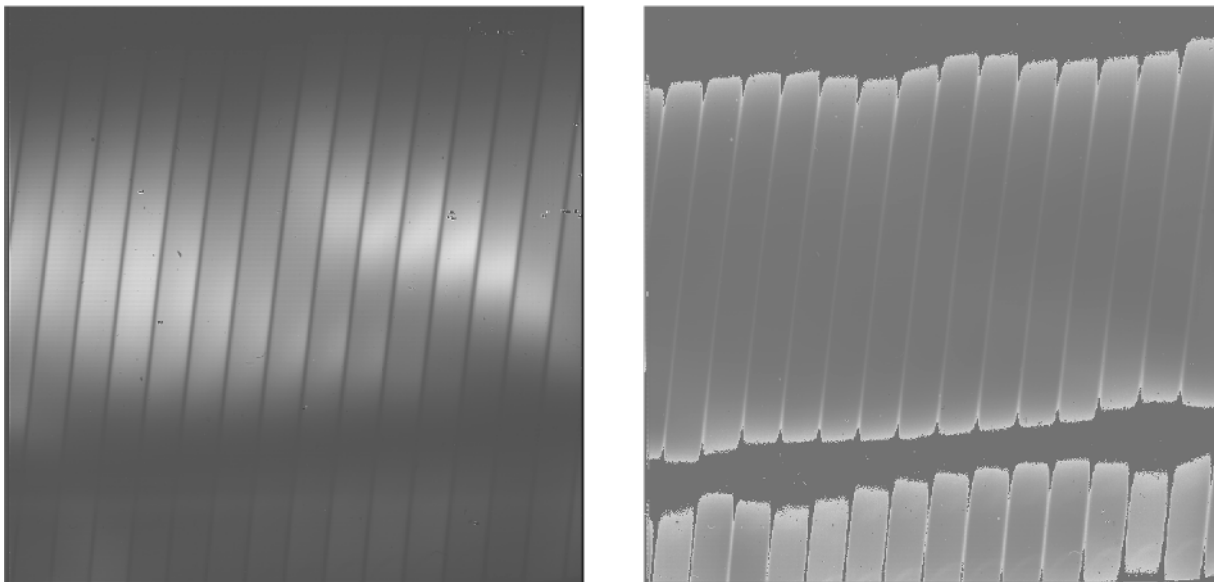


Fig. 3: Sample flat in High-Medium configuration. The left image is the black frame; the right image is the final processed flat frame. Unilluminated regions are set to zero in the final frame.

4.3 Despiking

After coadding the raw frames to make A and B nod frames, the pipeline attempts to identify and correct spikes (outlier data points) in the individual images. All A frames are compared, and any pixels with values greater than a threshold number of standard deviations from the mean value across the other frames are replaced with that mean value. B frames are similarly compared with each other. The threshold for identifying spikes is a tunable parameter; the default value is 20 sigma. Also in the despiking step, frames with significantly different overall background levels (“trashed” frames) may be identified automatically and removed from subsequent reduction. It is common, for example, for the first frame to look significantly different from the rest of the frames. In this case, leaving out this frame may improve the signal-to-noise of the final result.

4.4 Debounce

After despiking, there is an optional step called debouncing, which may help alleviate the effects of small optics shifts (“bounces”) between the nod beams. If there is a slight mismatch in the order placement on the array, it can lead to poor background subtraction when the nods are subtracted. In the debouncing algorithm, each nod pair is undistorted, then the B nod is shifted slightly in the spatial direction and differenced with the A nod. The shift direction that results in a minimum squared difference (summed over the spectral direction) is used as the bounce direction. The amplitude of the shift is controlled by the bounce parameter, which should be set to a number whose absolute value is between 0 and 1 (typically 0.1). If the bounce parameter is set to a positive number, only the above shift (the first-derivative bounce) will be corrected. If the bounce parameter is set to a negative value (e.g. -0.1), the debouncing algorithm will also attempt to fix the second-derivative bounce by smoothing the A or B nod slightly; the amount of smoothing is also controlled by the absolute value of the bounce parameter. Note that if the observed source is too near the edge of the order, it may confuse the debouncing algorithm; in this case, it is usually best to turn debouncing off (i.e. set the bounce parameter to 0). The default is not to use the debounce algorithm.

4.5 Subtract sky

In either nod-on-slit or nod-off-slit mode, each B nod is subtracted from each adjacent A nod (Fig. 4). This step usually removes most of the background emission from the image, but if there were changes in the sky level between the two nod frames, there may still be some residual sky signal. For the nod-off-slit mode, this residual signal can be estimated and corrected for before subtracting the nods, by subtracting a weighted fraction of the B data from the A data. The weighting is chosen to minimize the squared difference between the A and B nods. For the nod-on-slit mode, the mean background at each wavelength may be subtracted after nod subtraction and distortion correction, so that the wavelengths align with the rows or columns. The pipeline performs this step immediately before coadding, if desired.

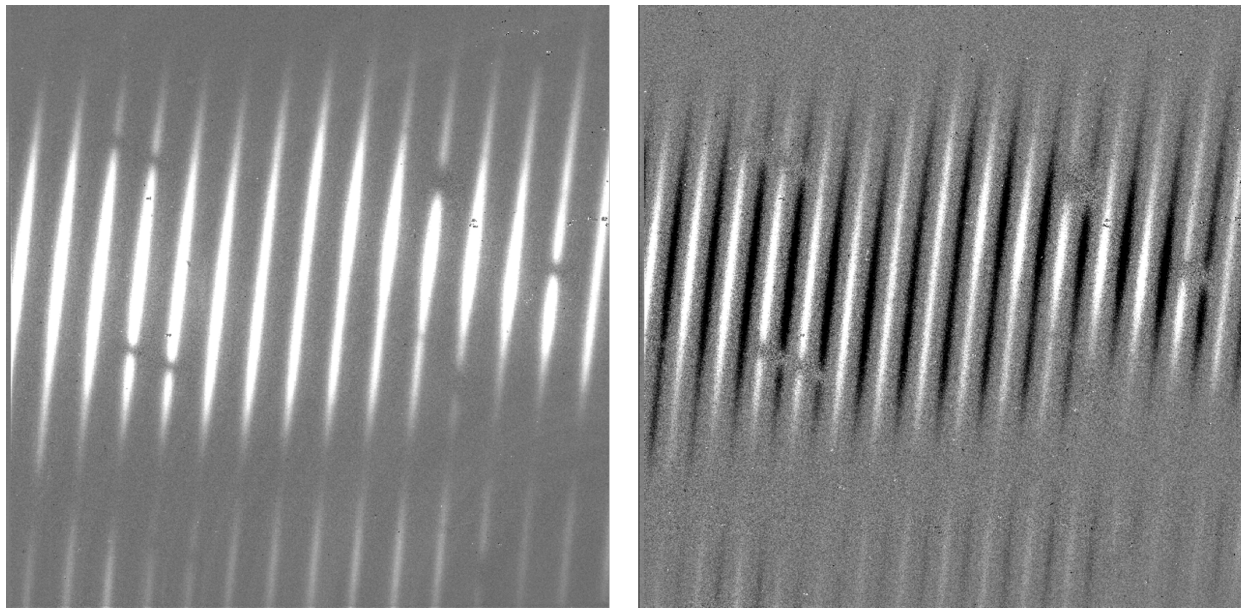


Fig. 4: Background subtracted frames in nod-off-slit (left) and nod-on-slit (right), in High-Medium configuration.

For the mapping mode, each of the steps across the source is treated as an A frame. The three sky frames taken at the end of the map are averaged together and this average is subtracted from each A frame (Fig. 5). This mode is usually used for extended sources that fill the entire slit, in which case there is no way to estimate or correct for any residual background remaining after sky subtraction. The three sky frames at the end of the map can be part of the map (without science target signal) or dedicated sky positions away from the source.

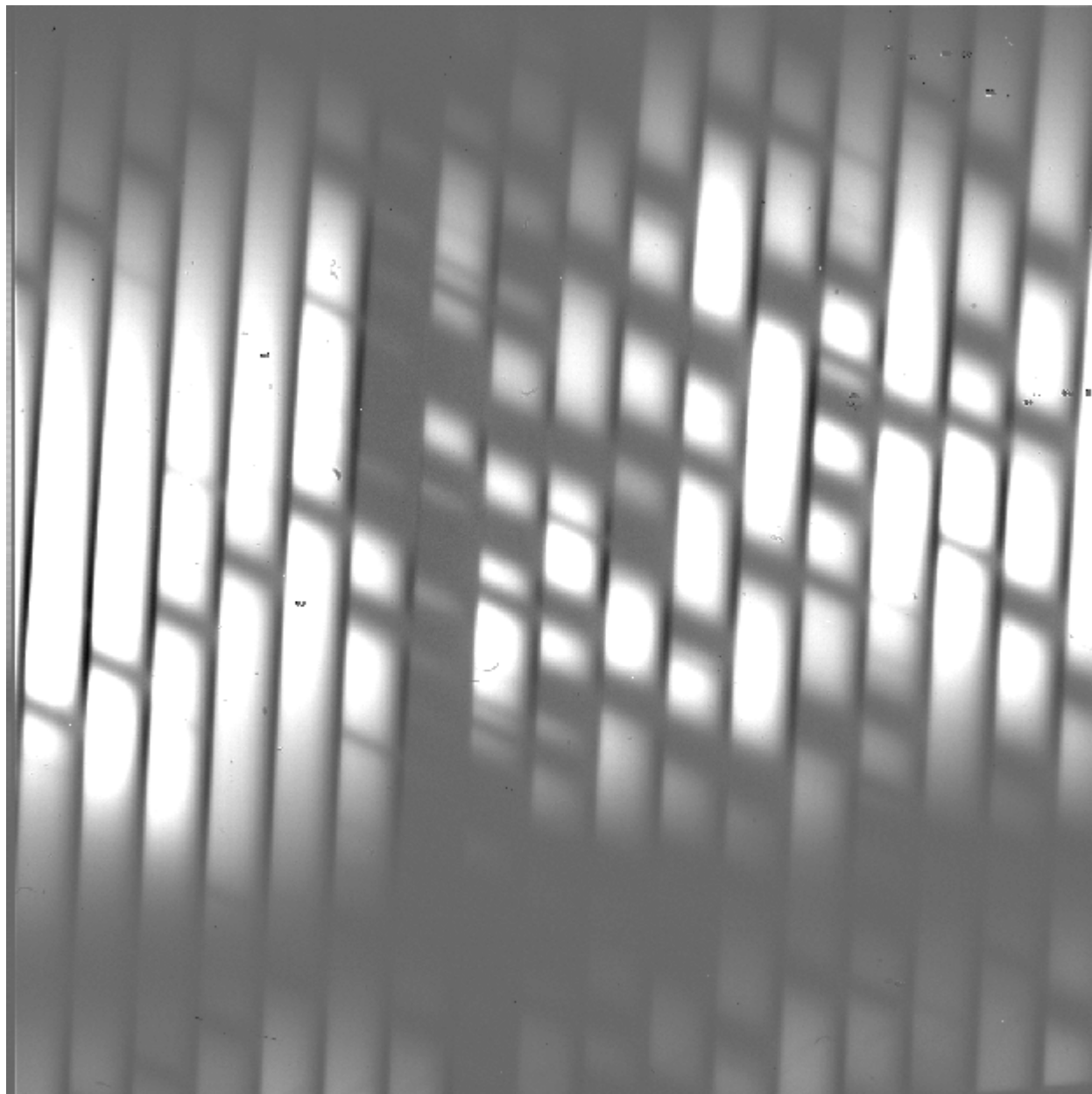


Fig. 5: Background subtracted image in High-Medium configuration with map mode. The extended source fills the entire slit.

4.6 Flat correct

After background subtraction, each science frame is multiplied by the processed flat calibration frame, described above. This has the effect of both correcting the science frame for instrumental response and calibrating it to intensity units.

4.7 Clean bad pixels

In this step, bad pixels are identified and corrected. Bad pixels may first be identified in a bad pixel mask, provided by the instrument team. In this FITS image, pixels that are known to be bad are marked with a value of 0; good pixels are marked with a value of 1. Alternately, bad pixels may be identified from their noise characteristics: if the error associated with a pixel is greater than a threshold value (by default: 20) times the mean error for the frame, then it is marked as a bad pixel. Unlike the despiking algorithm, which identifies outliers by comparing separate frames, outliers in this algorithm are identified by comparing the values within a single frame.

Bad pixels are corrected by using neighboring good values to linearly interpolate over the bad ones. The search for good pixels checks first in the y-direction, then in the x-direction. If good pixels cannot be identified within a 10-pixel radius, then the bad pixel will not be corrected. Note that bad pixels are also fixed or ignored in the extraction process, so it is not critical to correct all bad pixels at this stage.

4.8 Undistort

Next, the pipeline corrects the image for optical distortion, resampling the image onto a regular grid of spatial and spectral elements. The EXES distortion correction uses knowledge of the optical design of the instrument, as well as data recorded in the header of the observation, to calculate and correct for the optical distortion in each observation.

The optical parameters used to calculate the distortion correction are listed in the table below. Most of these parameters should not change, or should change rarely, so they have default values listed in configuration tables available to the pipeline (the *.dat* files listed below). Values in the *tortparm.dat* configuration table tend to change over time, as the instrument is adjusted, so their defaults vary by date. Some of the distortion parameters must come from the headers and do not have default values. One parameter is tuned at run-time (*krot*), and one must be manually optimized by the user (*waveno0*).

Table 1: Optical distortion parameters.

Parameter	Default value	Source	Comment
<i>hrfl</i>	(varies by date)	<i>tortparm.dat</i>	High resolution chamber focal length
<i>xdfl</i>	(varies by date)	<i>tortparm.dat</i>	Assumed cross- dispersed focal length
<i>slitrot</i>	(varies by date)	<i>tortparm.dat</i>	Slit rotation
<i>brl</i>	0.00	<i>tortparm.dat</i>	Barrel distortion parameter
<i>x0brl</i>	0.00	<i>tortparm.dat</i>	X-coordinate for the center of the barrel distortion
<i>y0brl</i>	0.00	<i>tortparm.dat</i>	Y-coordinate for the center of the barrel distortion
<i>hrg</i>	-0.015	<i>headerdef.dat</i>	Gamma (out of plane) angle for the echelon grating (radians)
<i>xdg</i>	0.033	<i>headerdef.dat</i>	Gamma angle for cross- dispersion grating (radians)
<i>xdlrdgr</i>	0.001328	<i>headerdef.dat</i>	Low resolution groove spacing (cm)
<i>xdmrdgr</i>	0.003151	<i>headerdef.dat</i>	Medium resolution groove spacing (cm)
<i>pixelwd</i>	0.0025	<i>headerdef.dat</i>	Pixel width
<i>detrot</i>	0.10	<i>headerdef.dat</i>	Detector rotation
<i>hrr</i>	9.8	<i>headerdef.dat</i> / Optimized at run-time	R number for echelon grating
<i>waveno0</i>	(none)	Header keyword WAVENO0	Planned central wavenumber
<i>echelle</i>	(none)	Header keyword ECHELLE	Echelle grating angle (deg)
<i>krot</i>	(varies by date)	<i>tortparm.dat</i> / Optimized at run-time	Rotation between chambers

Several of these parameters are then combined or recalculated before being used in the distortion correction. The cross-dispersed R number (*xdr*) is calculated from the groove spacing (*xdlrdgr* or *xdmrdgr*), the grating angle (*echelle*), and the central wavenumber (*waveno0*). The echelon R number (*hrr*) is recalculated from the above default number, the echelon gamma angle (*hrg*), and the central wavenumber. It may also be optionally tuned by the user. The expected order spacing for cross-dispersed modes is calculated from the cross-dispersed R number (*xdr*), the cross-dispersed focal length (*xdfl*), the pixel width (*pixelwd*), and the echelon gamma angle (*hrg*). The order spacing and the rotation angle (*krot*) are optimized for cross-dispersed modes when the calibration frame is processed, as described above.

The distortion correction equations calculate undistorted coordinates by correcting for the following effects for the cross-dispersed modes:

- slit skewing within orders due to the echelon “smile” (*hrg*, *hrr*, *slitrot*)
- nonlinearity of the echelon spectrum (*hrr*, *pixelwd*, *hrfl*)
- skewing by spectrum rotation on the detector due to the angle between the chambers, and the cross-dispersion smile (*xdg*, *xdr*, *krot*, *pixelwd*, *xdfl*, *hrfl*, *hrr*)
- nonlinearity of the cross-dispersion spectrum (*xdr*, *pixelwd*, *xdfl*)
- barrel distortion (*brl*, *x0brl*, *y0brl*)

For the long-slit modes, the distortion correction accounts for:

- skewing by the cross-dispersion smile (*xdg*, *xdr*, *slitrot*, *pixelwd*, *xdfl*)
- nonlinearity of the cross-dispersion spectrum (*xdr*, *pixelwd*, *xdfl*)
- barrel distortion (*brl*, *x0brl*, *y0brl*)

When the undistorted coordinates have been calculated, the 2D science image is interpolated from raw (x,y) pixel coordinates onto these coordinates. By default, a cubic interpolation is used that closely approximates a sinc interpolation function; bilinear interpolation is also available.

After distortion correction, the spatial and spectral elements should be aligned with the columns and rows of the image (Fig. 6). For the long-slit modes, the x-axis is the dispersion axis; for cross-dispersed modes, the y-axis is the dispersion axis.

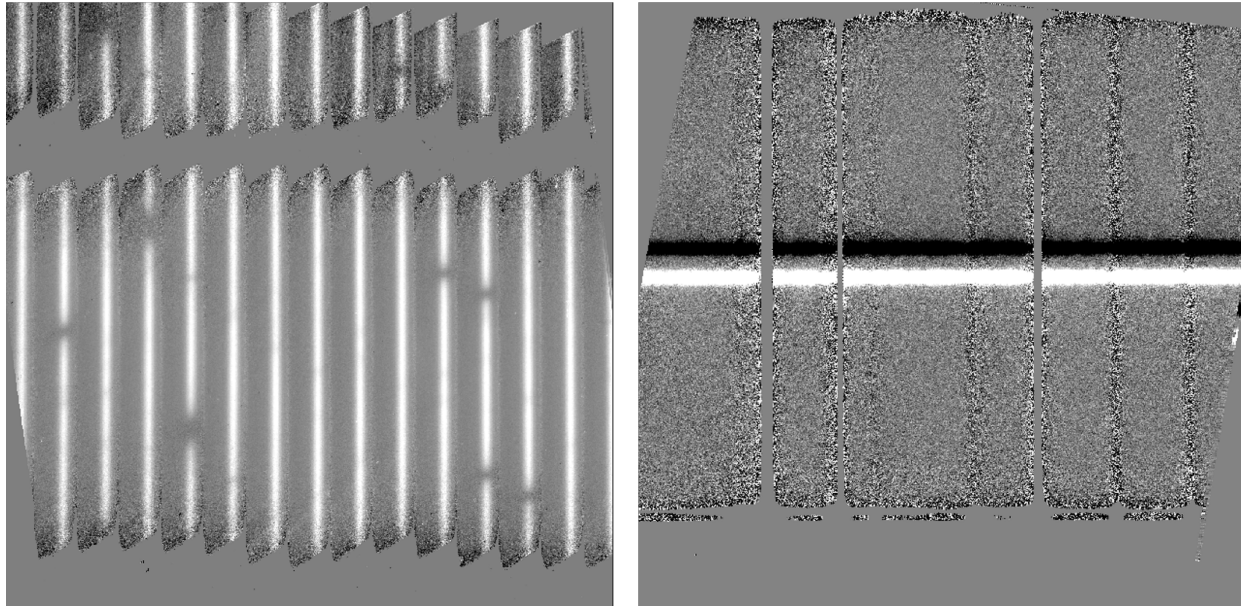


Fig. 6: Undistorted images in High-Medium configuration (left) and Medium configuration (right).

4.9 Coadd frames

The final step in the image processing prior to spectral extraction is coadding of the background-subtracted, undistorted frames. If desired, coaddition can be skipped, and individual spectra can be extracted from each frame, to be combined later. For faint sources, however, coaddition of the 2D image is recommended, for more accurate extraction.

Prior to coadding, the spectra can be shifted in the spatial direction to account for slight shifts in the position of the object in the slit between nod pairs. The shift is calculated by creating a template of the spectrum from all frames, averaged in the spectral direction, and cross-correlating each frame with the template.

By default, the pipeline weights all input frames by their correlation with this spectral template, so that if a frame is unusually noisy, or missed a nod, or had some other error, it will not contribute significantly to the coadded frame. However, it is possible to force the pipeline to use an unweighted coadd, if desired, or to specify the weights to use directly, or to explicitly identify frames for exclusion from the coadd.

The pipeline typically only coadds frames within a single file. Optionally, it may instead coadd frames from all files. This action is performed as a robust mean across all input frames, with or without weights. In cases where the target has not moved across the array between input files, this option may help increase signal-to-noise prior to extraction.

The final action in the coadd step is to rotate images taken in the cross-dispersed mode in order to align the spectra with the image rows (Fig. 7). This enables the pipeline to handle the spectral extraction in a consistent way for both the long-slit and the cross-dispersed data.

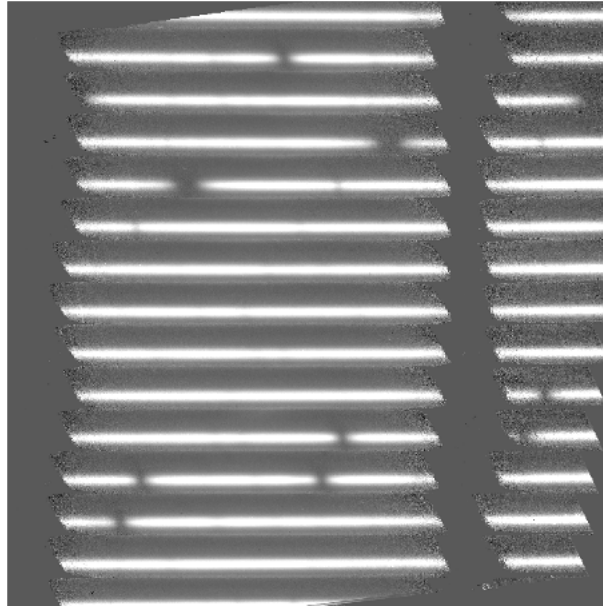


Fig. 7: Coadded image in High-Medium configuration, rotated to align the dispersion axis with the image rows.

4.10 Identify apertures

In order to aid in spectral extraction, the pipeline constructs a smoothed model of the relative intensity of the target spectrum at each spatial position, for each wavelength. This spatial profile is used to compute the weights in optimal extraction or to fix bad pixels in standard extraction (see next section). Also, the pipeline uses the median profile, collapsed along the wavelength axis, to define the extraction parameters.

To construct the spatial profile, the pipeline first subtracts the median signal from each column in the rectified spectral image to remove the residual background. The intensity in this image in column i and row j is given by

$$O_{ij} = f_i P_{ij}$$

where f_i is the total intensity of the spectrum at wavelength i , and P_{ij} is the spatial profile at column i and row j . To get the spatial profile P_{ij} , we must approximate the intensity f_i . To do so, the pipeline computes a median over the wavelength dimension (columns) of the order image to get a first-order approximation of the median spatial profile at each row P_j . Assuming that

$$O_{ij} \approx c_i P_j,$$

the pipeline uses a linear least-squares algorithm to fit P_j to O_{ij} and thereby determine the coefficients c_i . These coefficients are then used as the first-order approximation to f_i : the resampled order image O_{ij} is divided by f_i to derive P_{ij} . The pipeline then fits a low-order polynomial along the columns at each spatial point s in order to smooth the profile and thereby increase its signal-to-noise. The coefficients of these fits can then be used to determine the value of P_{ij} at any column i and spatial point j (see Fig. 8, left). The median of P_{ij} along the wavelength axis generates the median spatial profile, P_j (see Fig. 8, right).

The pipeline then uses the median spatial profile to identify extraction apertures for the source. The aperture centers can be identified automatically by iteratively finding local maxima in the absolute value of the spatial profile, or can be specified directly by the user. By default, a single aperture is expected and defined for nod-off-slit mode; two apertures are expected for nod-along-slit mode.

Besides the aperture centers, the pipeline also specifies a PSF radius, corresponding to the distance from the center at which the flux from the source falls to zero. By default, this value is automatically determined from the width of a

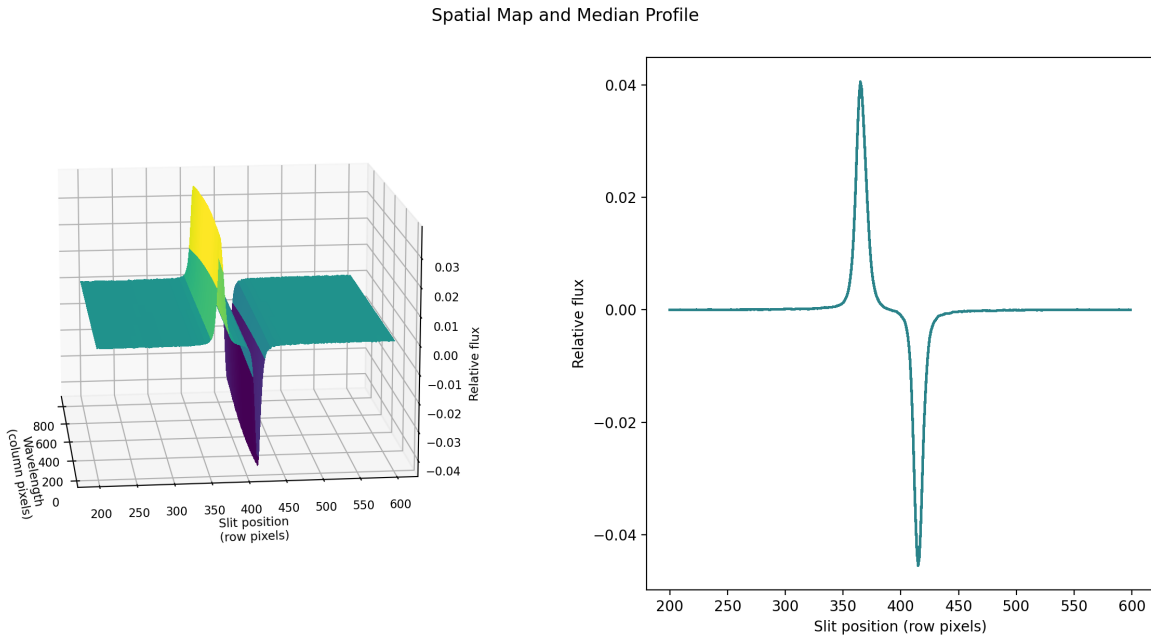


Fig. 8: Spatial model and median spatial profile, for the Medium configuration image in Fig. 6. The spatial model image here is rotated for comparison with the profile plot: the y-axis is along the bottom of the surface plot; the x-axis is along the left.

Gaussian fit to the peak in the median spatial profile, as

$$R_{psf} = 2.15 \cdot \text{FWHM}.$$

For optimal extraction, the pipeline also identifies a smaller aperture radius, to be used as the integration region:

$$R_{ap} = 0.7 \cdot \text{FWHM}.$$

This value should give close to optimal signal-to-noise for a Moffat or Gaussian profile. The pipeline also attempts to specify background regions outside of any extraction apertures, for fitting and removing the residual sky signal. All aperture parameters may be optionally overridden by the pipeline user.

4.11 Extract spectra

The spectral extraction algorithms used by the pipeline offer two different extraction methods, depending on the nature of the target source. For point sources, the pipeline uses an optimal extraction algorithm, described at length in the Spextool paper (see the *Other Resources* section, below, for a reference). For extended sources, the pipeline uses a standard summing extraction.

In either method, before extracting a spectrum, the pipeline first uses any identified background regions to find the residual sky background level. For each column in the 2D image, it fits a low-order polynomial to the values in the specified regions, as a function of slit position. This polynomial determines the wavelength-dependent sky level (B_{ij}) to be subtracted from the spectrum (D_{ij}).

Standard extraction

The standard extraction method uses values from the spatial profile image (P_{ij}) to replace bad pixels and outliers, then sums the flux from all pixels contained within the PSF radius. The flux at column i is then:

$$f_{i,\text{sum}} = \sum_{j=j_1}^{j_2} (D_{ij} - B_{ij})$$

where j_1 and j_2 are the upper and lower limits of the extraction aperture (in pixels):

$$\begin{aligned} j_1 &= t - R_{PSF} \\ j_2 &= t + R_{PSF} \end{aligned}$$

given the aperture center (t). This extraction method is the only algorithm available for extended sources.

Optimal extraction

Point sources may occasionally benefit from using standard extraction, but optimal extraction generally produces higher signal-to-noise ratios for these targets. This method works by weighting each pixel in the extraction aperture by how much of the target's flux it contains. The pipeline first normalizes the spatial profile by the sum of the spatial profile within the PSF radius defined by the user:

$$P'_{ij} = P_{ij} / \sum_{j=j_1}^{j_2} P_{ij}.$$

P'_{ij} now represents the fraction of the total flux from the target that is contained within pixel (i,j) , so that $(D_{ij} - B_{ij})/P'_{ij}$ is a set of j independent estimates of the total flux at column i . The pipeline does a weighted average of these estimates, where the weight depends on the pixel's variance and the normalized profile value. Then, the flux at column i is:

$$f_{i,\text{opt}} = \frac{\sum_{j=j_3}^{j_4} M_{ij} P'_{ij} (D_{ij} - B_{ij}) / (V_{D_{ij}} + V_{B_{ij}})}{\sum_{j=j_3}^{j_4} M_{ij} P'_{ij}{}^2 / (V_{D_{ij}} + V_{B_{ij}})}$$

where M_{ij} is a bad pixel mask and j_3 and j_4 are the upper and lower limits given by the aperture radius:

$$\begin{aligned} j_3 &= t - R_{ap} \\ j_4 &= t + R_{ap} \end{aligned}$$

Note that bad pixels are simply ignored, and outliers will have little effect on the average because of the weighting scheme.

The variance for the standard spectroscopic extraction is a simple sum of the variances in each pixel within the aperture. For the optimal extraction algorithm, the variance on the i^{th} pixel in the extracted spectrum is calculated as:

$$V_i = \sum_{j=j_3}^{j_4} \frac{M_{ij}}{P'_{ij}{}^2 V_{ij}}$$

where P'_{ij} is the scaled spatial profile, M_{ij} is a bad pixel mask, V_{ij} is the variance at each background-subtracted pixel, and the sum is over all spatial pixels j within the aperture radius. The error spectrum for 1D spectra is the square root of the variance.

Wavelength calibration

Wavelength calibration for EXES is calculated from the grating equation, using the optical parameters and central wavelength for the observation. It is stored as a wavelength calibration map identifying the wavelength for each column in each spectral order.

The wavelength calibration is expected to be good to within one pixel of dispersion, if the central wavelength is accurate. To refine the wavelength solution, the pipeline allows the user to identify a single spectral feature in the extracted spectrum. Using this feature, the central wavelength is recalculated and the wavelength calibration is adjusted to match. However, the distortion correction also depends on the central wavelength, so if the recalculated value is significantly different from the assumed value, the reduction should be repeated with the new value to ensure an accurate distortion correction and dispersion solution.

4.12 Combine multiple spectra

The final pipeline step is the combination of multiple spectra of the same source, from separate observations, apertures, and orders. For all configurations, the individual extracted 1D spectra for each order are combined with a robust weighted mean, by default.

For cross-dispersed configurations, the pipeline performs an automatic merge of the spectra from all orders. This algorithm uses the signal-to-noise ratio in overlapping regions to determine which pixels to exclude, and performs a weighted mean of any remaining overlapping regions. Artifacts near the edges of orders may optionally be manually trimmed before the merge.

5 Other Resources

For more information on the reduction algorithms used in the EXES package, adapted from the TEXES pipeline, see the TEXES paper:

[TEXES: A Sensitive High-Resolution Grating Spectrograph for the Mid-Infrared](#), J.H. Lacy, M.J. Richter, T.K. Greathouse, D.T. Jaffe and Q. Zhu (2002, PASP 114, 153)

For more information on the reduction algorithms used in FSpextool, see the Spextool papers:

[Spextool: A Spectral Extraction Package for SpeX, a 0.8-5.5 Micron Cross-Dispersed Spectrograph](#), Michael C. Cushing, William D. Vacca and John T. Rayner (2004, PASP 116, 362).

[A Method of Correcting Near-Infrared Spectra for Telluric Absorption](#), William D. Vacca, Michael C. Cushing and John T. Rayner (2003, PASP 115, 389).

[Nonlinearity Corrections and Statistical Uncertainties Associated with Near-Infrared Arrays](#), William D. Vacca, Michael C. Cushing and John T. Rayner (2004, PASP 116, 352).

Part IV

Flux calibration

EXES spectra are calibrated to physical intensity units ($erg/s/cm^2/sr/cm^{-1}$) when they are multiplied by the blackbody-normalized calibration frame. This process does not account for residual atmospheric effects. These may be corrected by dividing by a telluric standard spectrum, or a model of the atmosphere.

Part V

Data products

6 Filenames

EXES output files are named according to the convention:

$$\text{FILENAME} = F[\textit{flight}]_EX_SPE_AOR-ID_SPECTELISPECTEL2_type_FNI[-FN2][_SN].fits$$

where *flight* is the SOFIA flight number, *EX* is the instrument identifier, *SPE* specifies that it is a spectral file, *AOR-ID* is the AOR identifier for the observation, *SPECTELISPECTEL2* are the keywords specifying the spectral elements used, *type* is the three-letter identifier for the product type (listed in the table below), and *FNI* is the file number corresponding to the input file. *FNI-FN2* is used if there are multiple input files for a single output file, where *FNI* is the file number of the first input file and *FN2* is the file number of the last input file. If the output file corresponds to a single frame of the input file, there may also be a trailing serial number (SN) that identifies the input frame associated with the output file.

7 Pipeline Products

The following table lists all intermediate products generated by the pipeline for EXES spectral modes, in the order in which they are produced. The product type is stored in the FITS headers under the keyword PRODTYPE. By default, the *raw*, *undistorted*, *coadded*, *spec*, *combspec*, *mrgordspec*, *flat*, *mask*, and *wavemap* products are saved. If a subarray readout was used, the second dimension for the image products may be smaller than 1024. Currently the pipeline does not handle or create any spectral mapping products (e.g. data cubes).

Table 2: Intermediate and final data products for EXES reductions

Step	Description	Product Type	File code	Saved by default	Size
Load Data	Raw frames coadded according to OTPAT; variance planes appended at the end of all raw frames	<i>raw</i>	RAW	Y	1024 x 1024 x 2 N_{pat} where N_{pat} is the number of readout patterns
Make Flat	Calibration frame created from <i>black-dark</i>	<i>inverseflat</i>	IFT	N	1024 x 1024 x 2
Make Flat	Wavelength calibration map	<i>wavemap</i>	WVM	Y	1024 x 1024 x 2
Despike	Outlier pixels identified and corrected. Comparison is across similar frames.	<i>despiked</i>	DSP	N	1024 x 1024 x 2 N_{pat}

continues on next page

Table 2 – continued from previous page

Step	Description	Product Type	File code	Saved by default	Size
Debounce	Corrected for optics shifts between nods	<i>debounced</i>	DBC	N	1024 x 1024 x 2 N_{pat}
Subtract Nods	B nods or sky frames subtracted from A nods or map steps	<i>nods subtracted</i>	NSB	N	1024 x 1024 x 2 N_A where N_A is the number of A nods or map steps
Flat Correct	Multiplied by flat	<i>flatted</i>	FTD	N	1024 x 1024 x 2 N_A
Clean Bad Pixels	Bad pixels identified from variance or bad pixel mask	<i>cleaned</i>	CLN	N	1024 x 1024 x 2 N_A
Undistort	Corrected for optical distortion	<i>undistorted</i>	UND	Y	1024 x 1024 x 2 N_A
Undistort	Undistorted flat frame (1/inverseflat) with order definition keywords	<i>flat</i>	FLT	Y	1024 x 1024 x 2
Coadd Pairs	All frames averaged together, if desired	<i>coadded</i>	COA	Y	1024 x 1024 x N_C where N_C is the number of coadded frames (typically $N_C = 1$)
Coadd Pairs	All remaining frames, separated into different files	<i>image</i>	IMG	N	1024 x 1024 x 2
Extract Spectra	1D extracted spectra from all orders and apertures	<i>spec</i>	SPC	Y	$N_w \times 4 \times N_{ord} N_{ap}$, where N_w is the number of wavelength bins, N_{ord} is the number of orders, and N_{ap} is the number of apertures
Extract Spectra	Bad pixel mask indicating pixels ignored during extraction	<i>mask</i>	MSK	Y	1024 x 1024

continues on next page

Table 2 – continued from previous page

Step	Description	Product Type	File code	Saved by default	Size
Combine Spectra	1D spectra from separate files or apertures combined together	<i>combspec</i>	CMB	Y	$N_w \times 4 \times N_{ord}$
Merge Orders	1D spectrum from all cross-dispersed orders merged together	<i>mrgordspec</i>	MRD	Y	$N_{mw} \times 4$ where N_{mw} is the number of merged wavelength bins

8 Data Format

All files produced by the pipeline are FITS single-extension image files. All image products are 3-D arrays of data, where the first half of the planes are the image frames and the second half of the planes are the variances associated with each pixel in each frame. The extracted spectral products are one-dimensional spectra, stored in four rows of data. The first row is the wavelength, the second is the flux, the third is the error (standard deviation), and the fourth is a reference model atmospheric transmission spectrum. If there were multiple orders in the spectrum (e.g. the High-Medium configuration) or multiple apertures selected (e.g. for the nod-on-slit mode), then the spectrum for each aperture and order is stacked into a different plane. The length of the row varies depending on the data, but is typically around 1000 pixels before merging.

Part VI

Grouping LEVEL_1 data for processing

For EXES spectroscopy modes, there are three kinds of data: darks, flats, and sources. Flats and darks are required for all reductions, as they are used to calibrate the data, correct for instrumental response, and identify the edges of the slit. Dark frames should be taken near in time to the flat frame. Flat frames should have the same instrument configuration, slit length, slit width, and grating angle as the science observation. In order for separate science files to be grouped together, these same conditions apply, and they should additionally share a common target and be taken with the same nodding mode.

These requirements translate into a set of FITS header keywords that must match in order for a set of data to be usefully reduced together: see the table below for a listing of these keywords. All match requirements apply to science targets; flat exceptions are noted.

Note that data grouping must be carried out before the pipeline is run. The pipeline expects all inputs to be reduced together as a single data set.

Table 3: Grouping criteria

Keyword	Datatype	Match Criterion
INSTCFG	STR	Exact
SPECTEL1	STR	Exact
SPECTEL2	STR	Exact
SLIT	STR	Exact
ECHELLE	FLT	Exact
SDEG	FLT	Exact
WAVENO0	FLT	Exact
MISSN-ID	STR	Exact
ALTI_STA	FLT	+/- 500
ALTI_END	FLT	+/- 500
ZA_START	FLT	+/- 2.5
ZA_END	FLT	+/- 2.5
OBSTYPE	STR	Exact (unless flat)
OBJECT	STR	Exact (unless flat)
INSTMODE	STR	Exact (unless flat)
PLANID	STR	Exact (unless flat)
AOR-ID (optional)	STR	Exact (unless flat)

Part VII

Configuration and execution

9 Installation

Running Redux for EXES requires IDL 8.1 or later, as well as the latest version of the IDL Astronomy User's Library, the Coyote graphics library, and the EXES Redux package. The EXES Redux code is 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.

EXES Redux was developed under Linux and MacOS X operating systems. Other operating systems and versions of IDL may also work, but have not been tested.

SOFIA may distribute the EXES Redux code as a gzipped tar file. If so, unpack it as for example:

```
tar xvzf exes_redux.tar.gz
```

This will create a directory called *exes_redux* which will contain a number of subdirectories. This directory should be added to the IDL_PATH as well.

10 Configuration

For EXES algorithms, default parameter and header keyword values are specified in a configuration table called *headerdef.dat*, located in the *exes* package subdirectory. 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). A sample of this configuration file is given in Appendix A. All keywords in the table with default values will be written to output files produced by the EXES package.

For spectral extraction algorithms, default options are specified in a configuration file called *EXES.dat*, located in the *fspextool* package subdirectory. This file contains keyword-value pairs, in the format *parameter=value*. The parameters must all be present and in the correct order, but can have any number of spaces or comments between them. Comment lines begin with the *%* or *#* character. See Appendix A for a sample of this configuration file as well.

In automatic pipeline mode, parameters set in the configuration files are the values actually used by the pipeline, unless overridden by an input parameter file. In interactive mode, the configuration files set the default values, but the parameter values used can be modified at run-time.

11 Input data

Redux takes as input raw EXES FITS data files, which are image cubes composed of 1024 x 1024 pixel image arrays. 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 the header configuration file in Appendix A for a full list of these keywords.

It is assumed that the input data have been successfully grouped before beginning reduction: EXES Redux considers all input files in a reduction to be part of a single homogeneous reduction group, to be reduced together with the same parameters. A single input flat frame is required to complete a reduction past the Clean Bad Pixels step; Redux will abort at the Undistort step if a flat frame is not provided along with the science data.

12 Automatic mode execution

Redux is an object-oriented program whose basic unit is a reduction object (*exes_reduction__define.pro*). To run the pipeline from the IDL command line as a DCS 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 appropriate reduction object according to the mode specified in the input FITS headers, then calls the object's reduce method. This method calls each processing step in order, as appropriate for the given mode. 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, 'infile.txt', PARAM_FILE='exes_param.json'
```

where *exes_param.json* is a file that lists parameter keywords and values in JSON format for any step in the pipeline reduction. 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.

13 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.

13.1 Basic workflow

To start an interactive reduction, open a set of EXES files, using the File menu (**File->Open New Reduction**). This will bring up a file dialog window (Fig. 9). All files selected will be reduced together as a single reduction set. For EXES, a flat and dark file are required to complete a science reduction, so these should be selected along with the science data.

Redux will decide the appropriate reduction steps from the input files, and load them into the GUI (Fig. 10).

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 (Fig. 11). Within the parameter editor, all values may be edited, but will not be used unless **Save** or **Done** is selected. Clicking **Save** will leave the parameter editor window open; clicking **Done** will save values and close the window. Clicking **Reset** will restore any edited values to their defaults; clicking **Cancel** will discard all changes to the parameters.

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**). Note that edited parameters will retain their values for future reductions unless they are manually reset.

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 (Fig. 12). The table rows displayed can be filtered by entering a search string into the **Filter** text box.

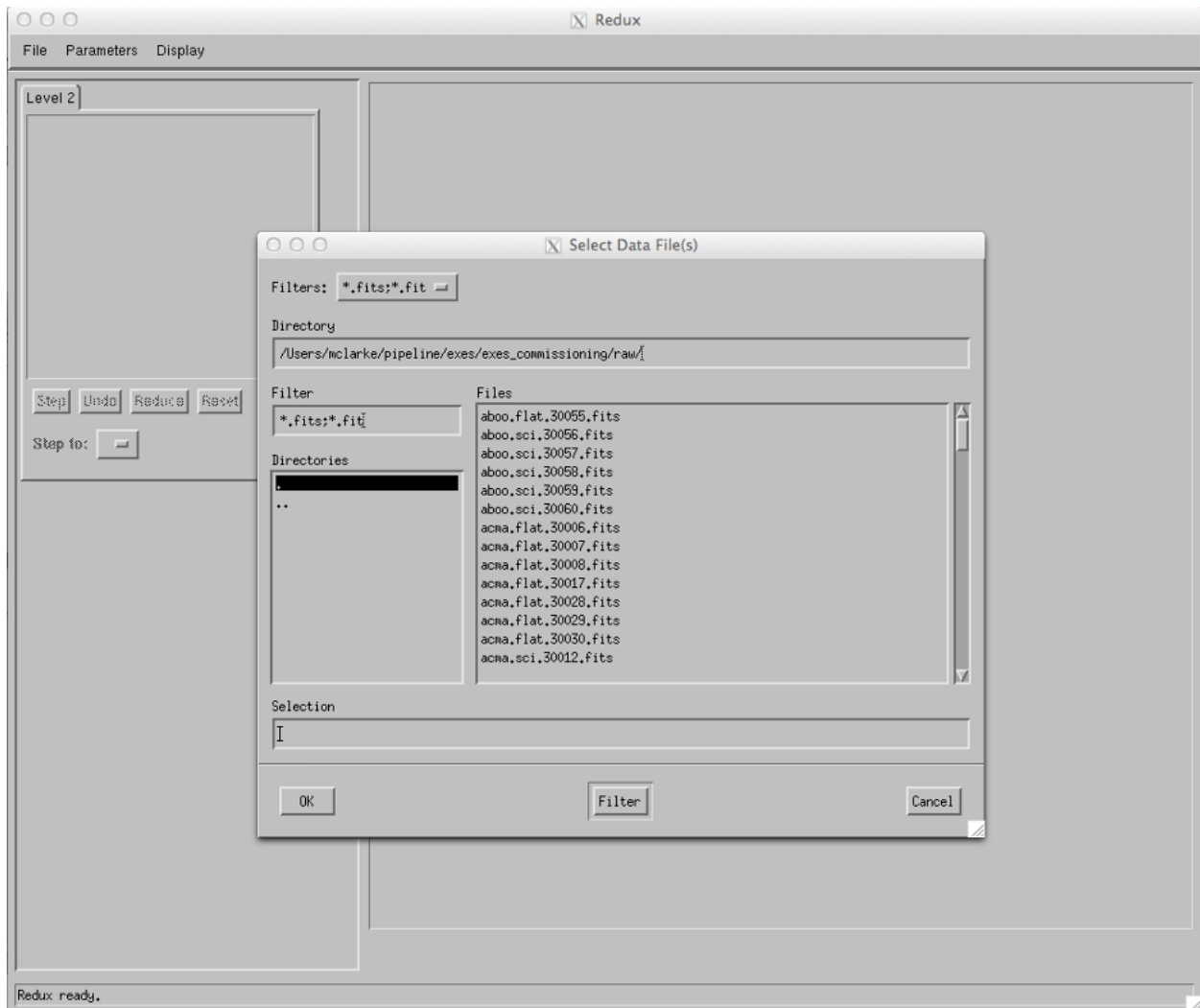


Fig. 9: Open new reduction.

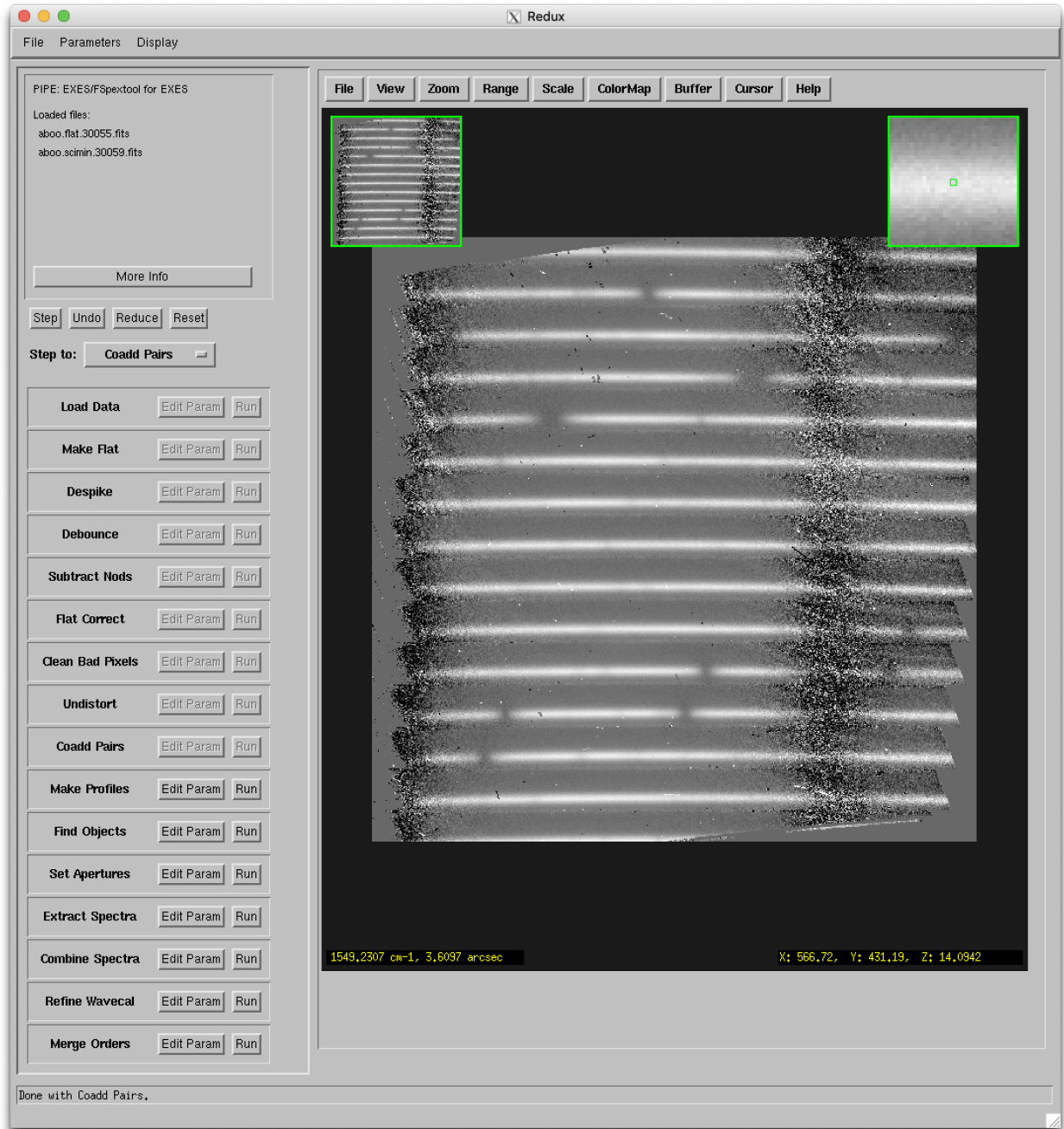


Fig. 10: Sample reduction steps for a High-Medium observation.

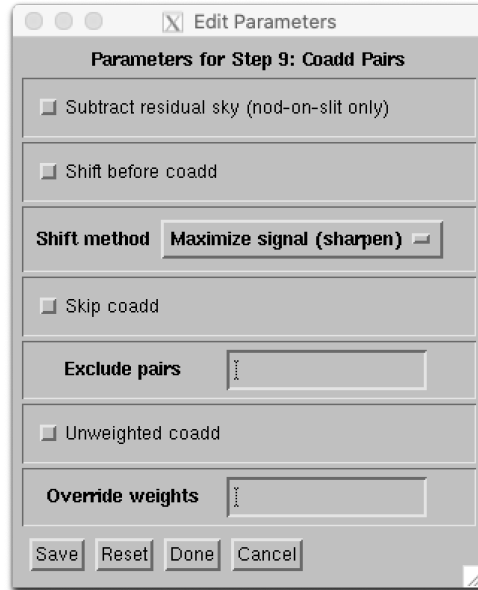


Fig. 11: Sample parameter editor for a pipeline step.

	Filename	Dimensions	Instrument	ObsID	AOR ID	Object	Obstype	Source Type	Spectel1
0	aboo.flat,30055	1024x1024x56	EXES	2014-04-08_EX_F15830055	86_0001_14	alpha boo	FLAT	POINT_SOURCE	EXE_ELONG
1	aboo.sci,30059	1024x1024x168	EXES	2014-04-08_EX_F15830059	86_0001_14	alpha boo	OBJECT	POINT_SOURCE	EXE_ELONG

Filter:

Close

Fig. 12: File information table.

13.2 Display features

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

Table 4: Display tool 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
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

Ximgtool has five buffers available for simultaneous display of images. If there are more than five files loaded into Redux, they can only be viewed by selecting **Display->Quick Look** from the Redux menu. This will cycle through the data 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.

Median spatial profiles generated from the 2D spectra are displayed in an independent plot window, with a separate plot for each order (Fig. 13). When the aperture parameters are set, they will be overplot on the spatial profile in this window.

Extracted spectra are displayed using xvspec (Fig. 14), another display tool packaged with the pipeline. Like ximgtool, this tool also has a number of useful features (Table 5).

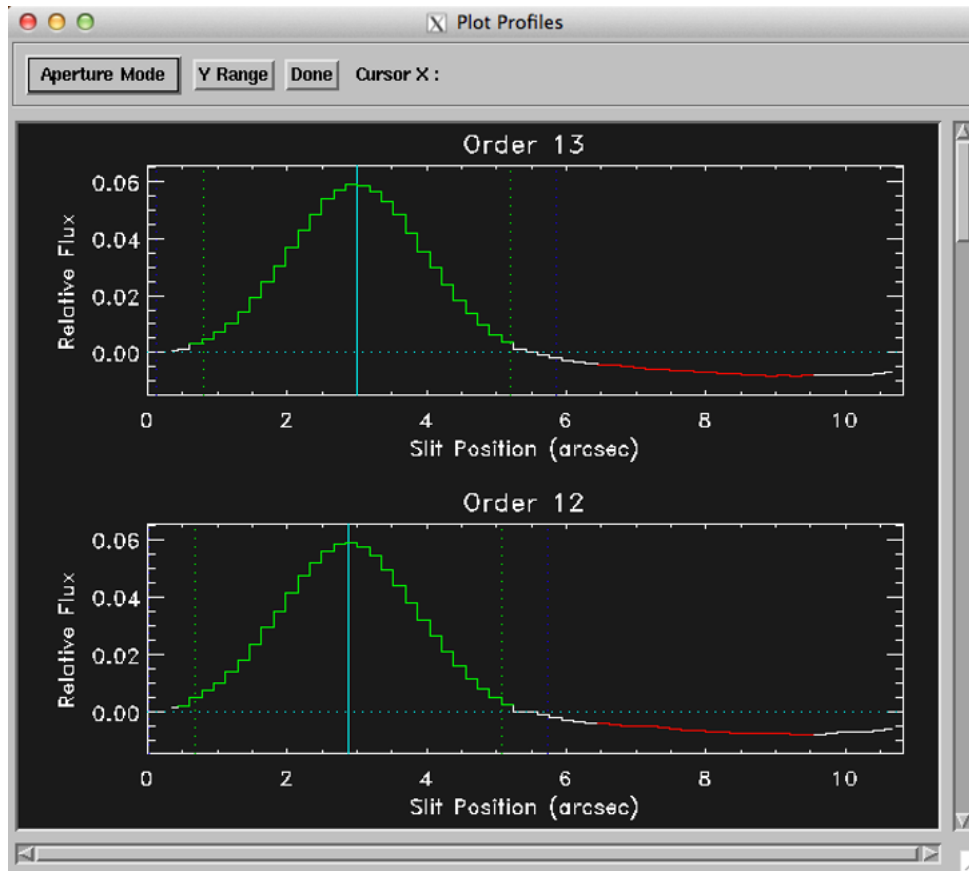


Fig. 13: Aperture location automatically identified and over-plotted on the spatial profile. The cyan line indicates the aperture center. Green lines indicate the integration aperture for optimal extraction, dark blue lines indicate the PSF radius (the point at which the flux goes to zero), and red lines indicate background regions.

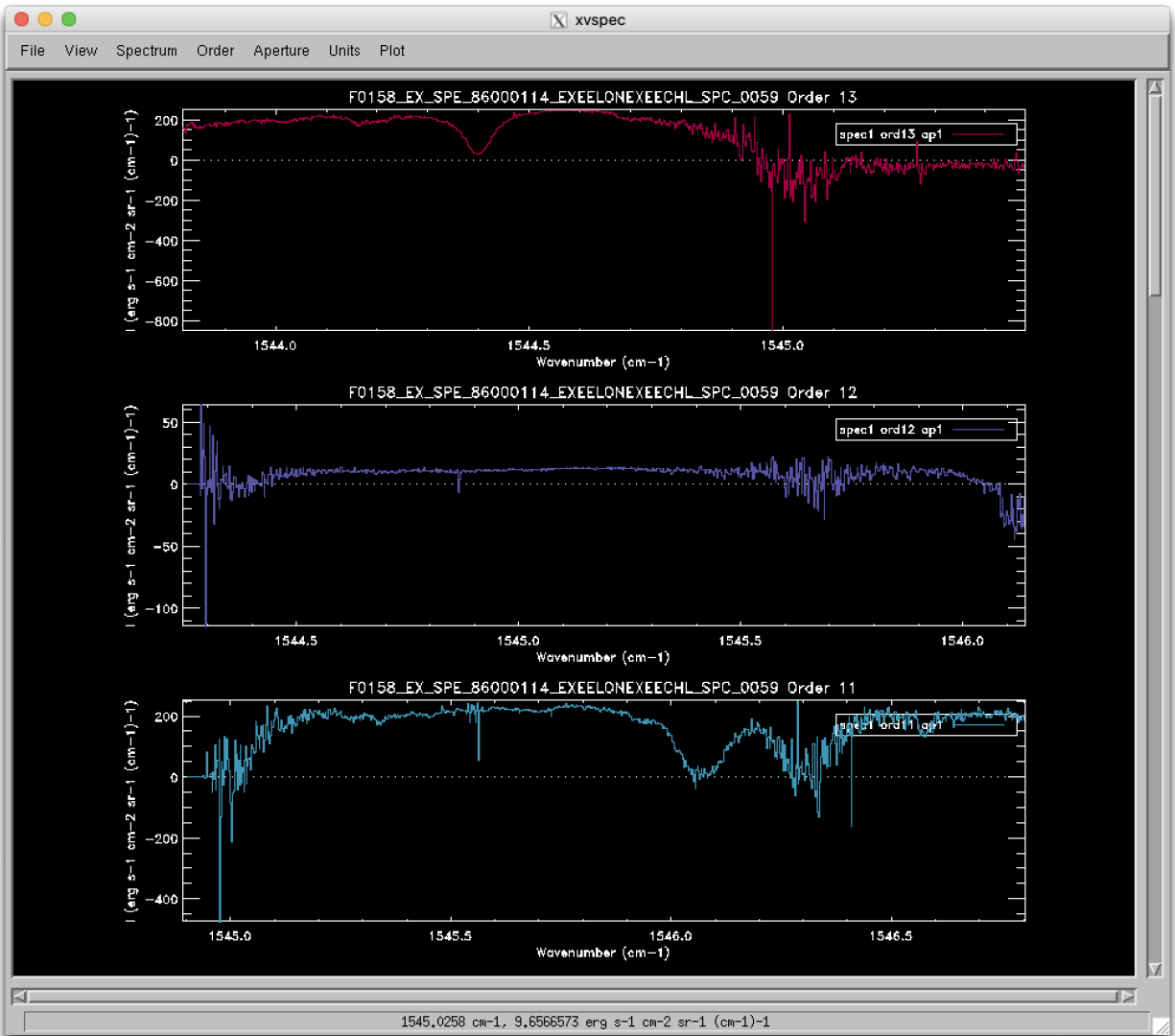


Fig. 14: Final extracted spectrum, displayed in an interactive plot window.

Table 5: Useful xvspec features

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.	–
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 atmospheric transmission	Plot->Overplot->Transmission	–
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 .

14 EXES Reduction

EXES data reduction with Redux follows the data reduction flowchart in Fig. 1. 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:**

- *Check headers:* By default, Redux will abort reduction if the input header keywords do not meet requirements. Uncheck this box to attempt the reduction anyway.
- *Save all intermediate files:* Check this box to save a file after each reduction step.
- *Apply linear correction:* If unchecked, the linearity correction will not be applied.
- *Toss INTs:* If set to 0, all integration sets will be used. If set to 1 or 2, that many integrations will be discarded from the beginning of the raw input file. Settings for multiple files may be specified, separated by a semi-colon.
- *Central wavenumber:* If the wavenumber is known more precisely than the value in the WAVENO0 keyword, enter it here to use it in the distortion correction calculations and wavelength calibration. This value

will be filled in automatically at the end of a reduction, if the Refine Wavecal step modifies the central wavenumber, so that the reduction may be reset and re-run with the new parameter.

- *HR focal length*: This distortion parameter may be adjusted to tune the dispersion solution for cross-dispersed spectra. Reducing HRFL decreases the wavenumber of the left end of the order.
- *XD focal length*: This distortion parameter may be adjusted to tune the dispersion solution for long-slit spectra. Reducing XDFL decreases the wavenumber of the left end of the order for a correctly identified central wavenumber. This parameter may also be adjusted to tune the predicted spacing between orders in cross-dispersed mode. In this case, increasing XDFL increases the predicted spacing.
- *Slit rotation*: This distortion parameter may be adjusted to tune the slit skewing, to correct for spectral features that appear tilted across an order. For cross-dispersed modes, increasing the slit rotation angle rotates the lines clockwise in the undistorted image. For long-slit modes, decreasing the slit rotation angle rotates the lines clockwise in the undistorted image.
- *Detector rotation*: This parameter may be adjusted to change the detector rotation from the default value.
- *HRR*: This parameter may be adjusted to change the R number for the echelon grating.
- *Read selection*: Set the readout mode. The default is 0, indicating that only the final destructive read should be used, regardless of readout pattern. 1 is the all-destructive read mode, 2 is the first/last non-destructive mode, 3 is the second/penultimate read mode, 4 is Fowler mode, and 5 is sample-up-the-ramp.
- *Refpix/BPM*: Set the bad pixel mask to used, by the location of the reference pixels. Set to 0 for reference pixels on the right side of the array (default), 1 for both sides, and 2 for the left side.

- **Make Flat:**

- *Threshold factor*: This value defines the illumination threshold for a flat, which determines where the edges of cross-dispersed orders are placed. Enter a number between 0 and 1. If the pipeline reports that it can not determine order edges, increasing the threshold slightly, to widen the gaps between orders, may help.
- *Optimize rotation angle*: Uncheck this box to prevent the pipeline from using the 2D FFT of the edge-enhanced flat to try to determine the best rotation angle (*krot*).
- *Edge enhancement method*: Depending on the noise and illumination characteristics of the input flat, it may work better to take the 2D FFT of the squared derivative, or the Sobel filtered image, rather than the derivative, of the flat. Use the drop-list to select an alternate algorithm.
- *Starting rotation angle*: Enter a number here to use as the starting value for *krot*. Otherwise, the starting value will be taken from the default parameters in configuration tables. After the Make Flat step is run, this value will be automatically filled in with the best calculated value.
- *Predicted spacing*: Enter a value here, in pixels, to use as the predicted spacing for cross-dispersed orders. This number will be used as a first guess for the spacing parameter in the fit to the 2D FFT, overriding the value calculated from header parameters.
- *Bottom pixel for undistorted order*: For medium and low configurations, specifying a value here sets the bottom edge of the order mask in the output wavemap (WVM) file to this value.
- *Top pixel for undistorted order*: Sets the top edge of the order mask in the wavemap to this value (medium and low only).
- *Starting pixel for undistorted order*: Sets the left edge of the order mask in the wavemap to this value (medium and low only).
- *Ending pixel for undistorted order*: Sets the right edges of the order mask in the wavemap to this value (medium and low only).
- *Interrupt TestTort for custom Hi-Res wavemap*: If set, a ‘customWVM.dat’ file is expected in the current directory. This file should contain bottom, top, left, and right edges to set in the output wavemap file for

cross-dispersed modes. There should be one line for each order with edge values specified as white-space separated integers.

- *Debug*: Check this box to view diagnostic images to determine whether the distortion parameters are being set correctly. If checked, Redux will display the undistorted flat, the derivative of the flat, the 2D FFT of the derivative, the fit region for the fundamental and second harmonic of the 2D FFT (this should show two round peaks, if all went well), and the illumination mask resulting from the order determination. If *Debug* is checked, Redux will raise an error and refuse to continue reduction: uncheck *Debug* and run the step again to continue with processing.

- **Despike:**

- *Spike factor*: Enter a value for the threshold for a pixel to be considered a spike. This value is multiplied by the standard deviation of the pixel across all frames: e.g. a value of 20 marks pixels with values over 20 sigma as bad pixels.
- *Mark trashed frames for exclusion*: If checked, frames that have significantly more noise compared to the rest will be identified and excluded from future processing. These frames will not be spike-corrected.

- **Debounce:**

- *Bounce factor*: Enter a number to use as the amplitude of the bounce correction. A value of 0 means no bounce correction will be applied. Values greater than 0 apply the first derivative bounce (shifting) only. Values less than 0 also apply the second derivative bounce (smoothing). Typical nonzero values are 0.1 or -0.1.
- *Spectral direction* If set, the bounce correction will be applied in the spectral direction instead of the spatial direction.

- **Subtract Nods:**

- *Skip nod subtraction*: If checked, this option will cause the pipeline to continue without performing nod-subtraction. This option may be used to extract reference sky spectra.
- *Subtract residual sky*: This option applies only to nod-off-slit mode. If checked, it will attempt to remove any residual sky signal before subtracting pairs. This option will remove any average continuum value along the slit; if this is not desired, background regions should be set at the time of extraction.
- *Dark subtraction*: If set, nod subtraction will be skipped and a dark frame will be subtracted from all planes instead. The dark file must have been loaded into the reduction with the science data. This option may produce better reference sky spectra than simply skipping nod subtraction.

- **Flat Correct:**

- *Skip flat correction*: If set, the data will not be multiplied by the flat. This option is primarily used for testing.

- **Clean Bad Pixels:**

- *Bad pixel threshold*: Enter a value for the threshold for a pixel to be considered a bad pixel. This value is multiplied by the standard deviation of all good pixels in the frame.

- **Coadd Pairs:**

- *Subtract residual sky*: This option applies only to nod-on-slit mode. If checked, the mean value at each wavelength bin will be subtracted to remove any residual sky noise.
- *Shift before coadd*: Set to attempt to shift spectra in the spatial direction to align with each other before coadding.
- *Shift method*: Select whether to maximize the signal or the signal-to-noise when determining the spatial shift.

- *Skip coadd*: Check to skip coadding images and instead extract separate spectra from each frame. If selected, each frame will be treated as a separate file after this step. For map mode, coadding is always skipped.
 - *Coadd all files*: If set, all input files will be coadded together, as if they were in a single file.
 - *Exclude pairs*: Enter comma-separated numbers to identify specific nod-subtracted frames to exclude from the coadd. For example, entering *1,4,7* will exclude the first, fourth, and seventh frame from the coadd. If multiple files are loaded, lists for each file may be entered, separated by a semi-colon. For example, *1,4;;3,10* will exclude the first and fourth frames from the coadd for the first file and the third and tenth from the coadd for the third file.
 - *Unweighted coadd*: Check to force the pipeline to use equal weights for all input frames.
 - *Override weights*: To override the default calculated weights, enter a fractional weight for each input frame, separated by commas. If multiple files are loaded, lists for each file may be entered, separated by a semi-colon. Weights must add to 1.0 for each file.
 - *Robust threshold*: Set to a number greater than zero to enable outlier rejection before coadding. The threshold value is the number of standard deviations away from the median for which a value is rejected.
- **Make Profiles:**
 - *Subtract median background*: In nod-on-slit mode or in long-slit observations of non-extended sources, the median level of the spatial profile is subtracted to account for a residual background level. If this is not desired (eg. it results in negative values in the spatial profile), uncheck this box.
- **Find Objects:**
 - *Number of auto apertures*: Enter the number of apertures to try to find automatically. The default is to look for two apertures for nod-on-slit mode, and one otherwise. If the observation is marked as SRC-TYPE=EXTENDED_SOURCE, or the slit is very short, the default is to mark a single aperture at the center of the order; to override this behavior, enter a value for this parameter, or for *Guess* or *Fix*.
 - *Guess*: Enter a guess value for the aperture to use as a starting point. Values are in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons. For example, *3,8;2,7* will look for apertures near 3" and 8" in the first image and 2" and 7" in the second image. If there are multiple files loaded, but only one aperture list is given, the aperture parameters will be used for all images.
 - *Width*: Enter a value to use as the starting value for the FWHM for automatically detecting apertures with a guess value. The default is 1".
 - *Fix position*: Enter a value to use as the aperture center. The format of this parameter is the same as for the *Guess* parameter, but no fit will be done to attempt to refine the position: it will be used as entered.
 - *Exclude orders*: Enter comma-separated numbers to identify specific orders to exclude from spectral extraction.
 - **Set Apertures:**
 - *Auto*: If checked, the pipeline will use the header parameters to determine how to set the apertures automatically. If SRC-TYPE=EXTENDED_SOURCE, or the slit is very short, it will extract the full slit. Otherwise, it will set a PSF radius to use in optimal extraction. If the data is cross-dispersed, it will not attempt to set background regions, since the slit is typically too short to determine them robustly. Uncheck *auto* to use either of the next two parameters.
 - *Subtract background*: If checked, and *auto* is not checked, then the pipeline will attempt to determine suitable background regions from the median spatial profile.

- *Set PSF radius for optimal extraction*: If checked, and *auto* is not checked, then the pipeline will attempt to determine a PSF radius from the median spatial profile. If a PSF radius is not set at this step, standard extraction will be used as the extraction method.
- *Override parameters*: If numbers are entered for any of the following parameters, they will be used as the aperture values. No automatic fit will be done for the specified parameter. For each one, values should be given in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons.
 - * *Override aperture radius*: Enter a value to use as the aperture radius.
 - * *Override aperture start*: Enter a value to use as lower edge of the aperture.
 - * *Override aperture end*: Enter a value to use as upper edge of the aperture.
 - * *Override PSF radius*: Enter a value to use as the PSF radius. This number should be **larger** than the aperture radius.
 - * *Override background regions*: Enter a range to use as the background region. For example, *0-1,8-10* will use the regions between 0" and 1" and between 8" and 10" to determine the background level to subtract in extraction.
- *Override aperture signs*: enter either 1 or -1 to override the automatic determination of the aperture sign from the spatial profile. If the value is -1, the spectrum will be multiplied by -1. Separate multiple apertures by commas; separate values for multiple files by semi-colons.

- **Extract Spectra:**

- *Extraction algorithm*: If set to *auto*, the pipeline will use standard extraction for SRC-TYPE=EXTENDED_SOURCE and optimal extraction otherwise. To override this, select either *optimal* or *standard*.
- *Use median profile*: By default, the pipeline uses a wavelength-dependent spatial profile for extraction, but this method may occasionally give poor results, if the signal-to-noise in the profile is low. Check this option to use the median spatial profile across all wavelengths instead.
- *Background fit order*: Set to a number greater than or equal to zero for the polynomial order of the fit to the background regions. The default is zero for EXES.
- *Fix bad pixels*: The pipeline usually uses the spatial profile to attempt to fix bad pixels during extraction. Occasionally, this results in a failed extraction. Uncheck this box to extract the spectra without bad pixel correction.
- *Bad pixel threshold*: Enter a value for the threshold for a pixel to be considered a bad pixel. This value is multiplied by the standard deviation of all good pixels in the aperture at each wavelength bin.
- *Extract sky emission*: If set, the output file name and product type (PRODTYPE) will be modified to include a '_SkySpec' suffix.

- **Combine Spectra:**

- *Combine apertures*: If unchecked, spectra from separate files will be combined, but separate apertures will remain separate in the output file.
- *Scale to median*: If checked, each spectrum will be scaled to the median across all spectra before combination.
- *Correct spectral shape*: If checked, each spectrum's shape will be corrected to the shape of the first spectrum before combination.
- *Combination statistic*: Select the combination method. The default is a robust weighted mean.
- *Extract sky emission*: If set, the output file name and product type (PRODTYPE) will be modified to include a '_SkySpec' suffix.

- **Refine Wavecal:**

- *Select line interactively:* If unchecked, no wavelength calibration refinement will be done. If checked, a GUI window will pop up, allowing selection and editing of a spectral feature. To select the feature, click on the spectrum, then in the new window, type f and click on either side of the desired feature. From the GUI window, click **Add from Plot**, then edit the wavelength to the known value. Click **Done** when finished.

- **Merge Orders:**

- *Selection threshold:* By default, pixels in overlapping regions are used if their signal-to-noise ratio is greater than 0.1 times the maximum signal-to-noise in the order. Increasing or decreasing this value may reduce artifacts like discontinuous jumps or excessively noisy overlap regions.
- *Trim before merge:* If set, the spectra from each order can be trimmed before merging. Over the `xvspec` window, type t and click on the left and right edge of the area to trim. Click **Done** in the pop-up window when finished. Should be used only for cross-dispersed configurations.
- *Extract sky emission:* If set, the output file name and product type (PRODTYPE) will be modified to include a ‘_SkySpec’ suffix.

Part VIII

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 coadded 2D image (*COA*) and an extracted spectrum (*SPC* or *CMB*).
- Look at the reduced image. Check that the distortion correction appears to have been calculated and applied correctly: order edges and spectral lines should appear to be aligned with the image columns and rows. If they are not, there may be a problem with the flat or with the optical parameters used.
- Look at the intermediate extracted spectra in *xvspec*. The spectra from positive and negative apertures should look similar. Spectra from separate observations of the same source in the same mode should look similar.
- Compare the signal-to-noise in the extracted spectrum to the reduced image. If the target seems bright in the reduced image, but the spectrum looks noisy, it may be that the aperture centers were not defined correctly.
- Check the aperture parameters recorded in the pipeline output and in the output file headers. The calculated aperture and PSF radii should have similar values for all apertures. The aperture centers for each order should also be similar in cross-dispersed spectra.
- Check the extracted spectra for excessive outliers. Most bad pixels should be removed in the extraction process. If they are not, it may be helpful to use the median spatial profile, or set the bad pixel threshold lower.

Part IX

Appendix A: Sample configuration files

Sample EXES configuration file, typically located in *exes_redux/exes/data/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. Some output keywords have defaults
# defined in other configuration files
#
# keyword reqd? default type min max enum
#-----
ADDTIME N -9999. float 0. . .
AIRSPEED N -9999. float . . .
ALTI_END Y -9999. float -60000. 60000. .
ALTI_STA Y -9999. float -60000. 60000. .
AOR_ID Y UNKNOWN string . . .
AOT_ID N UNKNOWN string . . .
ASSC_AOR N UNKNOWN string . . .
ASSC_MSN N UNKNOWN string . . .
ASSC_OBS N UNKNOWN string . . .
BB_TEMP Y 273.16 float 1 9999. .
BEAMTIME N 0.0 float . . .
BOUNCE N 0.0 float . . .
BPM N UNKNOWN string . . .
BRL N -9999. float . . .
CARDMODE N BLKSKY string . . NONE|BLK|SKY|SHINY|BLKSKY|BLKSHINY|
↳BLKOBJ|OBJ
CHOPPING N F bool . . .
CLOUD N 0.0 float . . .
CREATOR N UNKNOWN string . . .
CUMTIME N -9999. float 0. . .
DARKVAL N 0. float 0. . .
DATAPATH N UNKNOWN string . . .
DATASRC Y UNKNOWN string . . ASTRO|CALIBRATION|LAB|TEST|OTHER|FIRSTPOINT
DATATYPE N UNKNOWN string . . IMAGE|SPECTRAL|OTHER
DATE N UNKNOWN string . . .
DATE-OBS Y UNKNOWN string . . .
DEPLOY N UNKNOWN string . . .
DETECTOR N UNKNOWN string . . .
DETROT N 0.10 float . . .
DETSEC N UNKNOWN string . . .
DETSIZE N UNKNOWN string . . .
DITHER N F bool . . .
DRKFILE N UNKNOWN string . . .
DTHINDEX N -9999 int . . .
DTHNPOS N -9999 int . . .
```

DTHOFFS	N	-9999.	float	.	.	.
DTHPATT	N	UNKNOWN	string	.	.	3-POINT 5-POINT 9-POINT CUSTOM
EHELLE	Y	60.0	float	-35.	70.	.
ECTPAT	Y	UNKNOWN	string	.	.	.
EFL	N	-9999.	float	.	.	.
EFL0	N	-9999.	float	.	.	.
EPERADU	Y	75.0	float	1.	9999.	.
EQUINOX	N	-9999.	float	.	.	.
EXCLUDE	N	None	string	.	.	.
EXPTIME	N	-9999.	float	0.	.	.
FBC-STAT	N	UNKNOWN	string	.	.	FBC_OFF FBC_QS FBC_DY FBC_ON
FDATE	N	0.0	float	0.	9999.	.
FILENAME	Y	UNKNOWN	string	.	.	.
FILENUM	N	.	string	.	.	.
FILEREV	N	UNKNOWN	string	.	.	.
FLIGHTLG	N	UNKNOWN	string	.	.	.
FOCUS_EN	N	-9999.	float	-5000	5000	.
FOCUS_ST	N	-9999.	float	-5000	5000	.
FRAMETIM	Y	1.0	float	0.	9999.	.
GRATANGL	N	0.0	float	-360.	360.	.
GRDSPEED	N	-9999.	float	.	.	.
HEADING	N	-9999.	float	.	.	.
HEADSTAT	N	UNKNOWN	string	.	.	ORIGINAL UNKNOWN CORRECTED ERROR
HRDGR	N	0.7588	float	.	.	.
HRFL	N	-9999.	float	.	.	.
HRFL0	N	-9999.	float	.	.	.
HRG	N	-0.015	float	.	.	.
HRR	N	9.8	float	0.	9999.	.
IMAGEID	N	-9999	int	0	.	.
INSTCFG	Y	UNKNOWN	string	.	.	HI-MED MED HI-LO LO CAM PUP HIMED HILOW LOW CAMERA HIGH_MED HIGH_LOW MEDIUM HIGH-MED HIGH-LOW
INSTMODE	Y	UNKNOWN	string	.	.	STARE NOD_ON_SLIT NOD_OFF_SLIT MAP
INSTRUME	Y	EXES	string	.	.	EXES TEXES
KROT	N	-9999.	float	.	.	.
KWDICT	N	UNKNOWN	string	.	.	.
LASTREW	N	UNKNOWN	string	.	.	.
LAT_END	N	-9999.	float	.	.	.
LAT_STA	N	-9999.	float	.	.	.
LINFILE	N	UNKNOWN	string	.	.	.
LO_LIM	N	0.0	float	.	.	.
LON_END	N	-9999.	float	.	.	.
LON_STA	N	-9999.	float	.	.	.
MAPCRSYS	N	UNKNOWN	string	.	.	EQUATORIAL GALACTIC ECLIPTIC USER
MAPINTX	N	-9999.	float	.	.	.
MAPINTY	N	-9999.	float	.	.	.
MAPNXPOS	N	-9999	int	.	.	.
MAPNYPOS	N	-9999	int	.	.	.
MAPPING	N	F	bool	.	.	.
MCCSMODE	N	UNKNOWN	string	.	.	.
MISSN-ID	Y	UNKNOWN	string	.	.	.
NBELOW	N	-9999	int	0	.	.
NAIF_ID	N	.	int	.	.	.
NEXP	N	-9999	int	1	.	.
NFRAME	N	-9999	int	1	.	.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

NINT	Y	1	int	1	.	.
NODAMP	N	-9999.	float	.	.	.
NODANGLE	N	-9999.	float	-180	180	.
NODBEAM	N	UNKNOWN	string	.	.	.
NODCRSYS	N	UNKNOWN	string	.	.	ERF ECRF GALRF TARF FPIRF FFIRF <u>W</u>
↳WFIRF SIRF USER						
NODDING	N	F	bool	.	.	.
NODN	N	-9999	int	.	.	.
NODPATT	N	UNKNOWN	string	.	.	ABBA AB CUSTOM
NODSETL	N	-9999.	float	.	.	.
NODSTYLE	N	UNKNOWN	string	.	.	NMC NPC C2NC2 CUSTOM
NODTIME	N	-9999.	float	.	.	.
NORDERS	N	-9999	int	0	9999	.
NSPAT	N	1024	int	0	1025	.
NSPEC	N	1024	int	0	1025	.
NSUM	N	1	int	0	9999	.
NT	N	1024	int	0	1024	.
OBJECT	Y	UNKNOWN	string	.	.	.
OBS_ID	Y	UNKNOWN	string	.	.	.
OBSDEC	N	-9999.	float	-90.	90.	.
OBSERVER	N	UNKNOWN	string	.	.	.
OBSRA	N	-9999.	float	0.	24.	.
OBSTYPE	Y	UNKNOWN	string	.	.	OBJECT STANDARD_FLUX STANDARD_TELLURIC <u>W</u>
↳LAMP FLAT DARK BIAS SKY						
OMEGAP	N	-9999.	float	.	.	.
OPERATOR	N	UNKNOWN	string	.	.	.
ORDR_B	N	UNKNOWN	string	.	.	.
ORDR_E	N	UNKNOWN	string	.	.	.
ORDR_S	N	UNKNOWN	string	.	.	.
ORDR_T	N	UNKNOWN	string	.	.	.
ORIGIN	N	UNKNOWN	string	.	.	.
OTPAT	Y	UNKNOWN	string	.	.	.
PAGAIN	Y	1.0	float	0.	9999.	.
PINHOLE	N	0	int	-1	2	.
PIPELINE	N	UNKNOWN	string	.	.	.
PIPEVERS	N	UNKNOWN	string	.	.	.
PIXELWD	N	0.0025	float	0.	9999.	.
PIXSCAL	N	-9999.	float	.	.	.
PKGPATH	N	UNKNOWN	string	.	.	.
PLANID	Y	UNKNOWN	string	.	.	.
PLTSCALE	N	-9999.	float	.	.	.
PROCSTAT	Y	UNKNOWN	string	.	.	.
PRODTYPE	N	UNKNOWN	string	.	.	.
READNOIS	N	30.	float	0.	9999.	.
RESOLUN	N	-9999.	float	.	.	.
SATVAL	N	16000.	float	0.	65536	.
SCANNING	N	F	bool	.	.	.
SDEG	Y	180.	float	-45.	405.	.
SIBS_X	N	-9999	int	.	.	.
SIBS_Y	N	-9999	int	.	.	.
SLIT	Y	UNKNOWN	string	.	.	.
SLITOFF	N	-0.3	float	.	.	.
SLITROT	N	-9999.	float	.	.	.
SLITVAL	N	-9999.	float	.	.	.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

SLITWID	N	-9999.	float	.	.	.
SPACING	N	-9999.	float	.	.	.
SPECTEL1	Y	UNKNOWN	string	.	.	.
SPECTEL2	Y	UNKNOWN	string	.	.	.
SPIKEFAC	N	20.0	float	0.	9999.	.
SRCTYPE	Y	UNKNOWN	string	.	.	POINT_SOURCE EXTENDED_
↪SOURCE OTHER UNKNOWN						
STDFAC	N	20.0	float	0.	9999.	.
SUBARRNO	N	-9999	int	.	.	.
TELCONF	N	UNKNOWN	string	.	.	.
TELEL	N	-9999.	float	0.	90.	.
TELEQUI	N	UNKNOWN	string	.	.	.
TELESCOP	N	UNKNOWN	string	.	.	.
TELDEC	N	-9999.	float	-90.	90.	.
TELLOS	N	-9999.	float	-180	180	.
TELRA	N	-9999.	float	0.	24.	.
TELVPA	N	-9999.	float	0.	360.	.
TELXEL	N	-9999.	float	-90.	90.	.
TEMP_OUT	N	-9999.	float	.	.	.
TEMPPRI1	N	-9999.	float	-273.	.	.
TEMPPRI2	N	-9999.	float	-273.	.	.
TEMPPRI3	N	-9999.	float	-273.	.	.
TEMPSEC1	N	-9999.	float	.	.	.
THRFACT	N	0.5	float	0.	1.0001	.
TORTMODE	N	CARD	string	.	.	NONE OBJ CARD
TRACERR	N	F	bool	.	.	.
TRACKANG	N	-9999.	float	.	.	.
TRACMODE	N	UNKNOWN	string	.	.	OFF CENTROID ROF LIMB OFFSET ROF+OFFSET
↪CENTROID+INERTIAL ROF+INERTIAL OFFSET+INERTIAL ROF+OFFSET+INERTIAL						
TRASH	N	100.0	float	.	.	.
TSC-STAT	N	UNKNOWN	string	.	.	.
UP_LIM	N	11000.0	float	.	.	.
UTCSTART	N	UNKNOWN	string	.	.	.
UTCEND	N	UNKNOWN	string	.	.	.
WAVECENT	Y	0.0	double	0.	.	.
WAVENO0	Y	0.0	double	0.	.	.
WNO0	N	-9999.	double	0.	.	.
WVZ_END	N	-9999.	float	0.	.	.
WVZ_STA	N	-9999.	float	0.	.	.
X0BRL	N	-9999.	float	.	.	.
XD0	N	0.0	float	.	.	.
XDDELTA	N	0.0235	float	.	.	.
XDDGR	N	-9999.	float	.	.	.
XDFL	N	-9999.	float	.	.	.
XDFL0	N	-9999.	float	.	.	.
XDG	N	0.033	float	.	.	.
XDLRDGR	N	0.001328	float	.	.	.
XDMRDGR	N	0.003151	float	.	.	.
XDR	N	-9999.	float	.	.	.
XORDER1	N	0.0	float	0.	1024	.
Y0BRL	N	-9999.	float	.	.	.
ZA_END	Y	-9999.	float	-99999.	90.	.
ZA_START	Y	-9999.	float	-99999.	90.	.

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

Sample Spextool-style configuration file for spectroscopic algorithms, typically located in *exes_redux/fspextool/Instruments/EXES/Data/EXES.dat*

```
#
# Generated by M. Clarke, Jan. 2014
#
# EJM: Updated Mar 2019 to include WAVECENT in SPC, CMB, MRD Files
# This is the calibration file for the EXES spectrograph on SOFIA.
# Note the values must be in the correct order, but can have any number of
# spaces/comments between them.
#
#=====
#
INSTRUMENT=EXES
NCOLS=1024
NROWS=1024
STDIMAGE=1024
PLOTWINSIZE=700 512
FILENAME=FILENAME
EXPTIME=EXPTIME
TIME=TIME_OBS
POSANGLE=None
HA=None
AIRMASS=None
NINT=4
BADPIXMASK=None
%
%CAL BASE
%
CALMODULE=mc_exescals1d
%
%FILE READ MODE
%
FILEREADMODE=Filename
IPREFIX=F
OPREFIX=r
SUFFIX=*.fits*
FITSREADPROGRAM=mc_readexesfits
HEADCOMBPROGRAM=mc_exesdcshdr
YUNITS=erg s-1 cm-2 sr-1 (cm-1)-1
YTITLE=I (erg/s/cm!u2!n/sr/cm!u-1!n)
XUNITS=cm-1
XTITLE=Wavenumber (cm!u-1!n)
%
% Reduction Mode
%
REDUCTIONMODE=A
%
% Combine Base Information
%
COMBMODE=A
COMBSTAT=Median (Median Error)
COMBTHRESH=8.0
COMBODIR=proc/
```

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

```
%  
% Sky Base Information  
%  
  SKYSTAT=Robust Weighted Mean  
  SKYTHRESH=8.0  
%  
% Profile Parameters  
%  
  YBUFFER=2  
  OVERSAMP=1  
  ATMOSTHRESH=0  
%  
% Point Source Base  
%  
  PSNAPS=2  
  PPSFRAD=9.0  
  PSAPRAD=3.0  
  PSBGSUB=1  
  PSBGSTART=9.0  
  PSBGWIDTH=10  
  PSBGDEG=0  
  PSBGMULT=2.0  
%  
% Extended Source Base  
%  
  XSBGSUB=1  
  XSBG=0-9,24-36,54-60  
  XSBGDEG=0  
%  
% Additional processing base  
%  
  ADDLMODULE=None  
%  
%Other Base Parameters  
%  
  TRACEDEG=2  
  TRACESTEP=7  
  TRACESUMAP=7  
  TRACESIGTHRESH=1  
  TRACEWINTHRESH=5  
  BADPIXELTHRESH=5  
  PLOTSATURATEDPIXELS=0  
  SATURATION=3000  
  CHECKSEEING=0  
  SEEINGTHRESH=1.0  
  LINCORRECT=0  
  ERRORPROPAGATION=1  
  FLATFIELD=0  
  FIXBADPIXELS=1  
  OPTIMALEXTRACTION=1  
%  
% FITS Header Keywords to Grab  
% -----  
%
```

% Keywords required for processing, may also be required
% by SOFIA DCS
%
%
KEYWORD=ADDTIME
KEYWORD=AIRSPEED
KEYWORD=ALTI_END
KEYWORD=ALTI_STA
KEYWORD=AOR_ID
KEYWORD=AOT_ID
KEYWORD=ASSC_AOR
KEYWORD=ASSC_MSN
KEYWORD=ASSC_OBS
KEYWORD=BB_TEMP
KEYWORD=BEAMTIME
KEYWORD=BOUNCE
KEYWORD=BPM
KEYWORD=BRL
KEYWORD=CARDMODE
KEYWORD=CHOPPING
KEYWORD=CLOUD
KEYWORD=CREATOR
KEYWORD=CUMTIME
KEYWORD=DARKVAL
KEYWORD=DATAPATH
KEYWORD=DATASRC
KEYWORD=DATATYPE
KEYWORD=DATE
KEYWORD=DATE-OBS
KEYWORD=DEPLOY
KEYWORD=DETECTOR
KEYWORD=DETROT
KEYWORD=DETSIZE
KEYWORD=DETSEC
KEYWORD=DITHER
KEYWORD=DRKFILE
KEYWORD=DTHINDEX
KEYWORD=DTHNPOS
KEYWORD=DTHOFFS
KEYWORD=DTHPATT
KEYWORD=ECHELLE
KEYWORD=ECTPAT
KEYWORD=EFL
KEYWORD=EFL0
KEYWORD=EPERADU
KEYWORD=EQUINOX
KEYWORD=EXCLUDE
KEYWORD=EXPTIME
KEYWORD=FBC-STAT
KEYWORD=FDATE
KEYWORD=FILENAME
KEYWORD=FILENUM
KEYWORD=FILEREV
KEYWORD=FLIGHTLG

KEYWORD=FOCUS_EN
KEYWORD=FOCUS_ST
KEYWORD=FRAMETIM
KEYWORD=GRATANGL
KEYWORD=GRDSPEED
KEYWORD=HEADING
KEYWORD=HEADSTAT
KEYWORD=HRDGR
KEYWORD=HRFL
KEYWORD=HRFL0
KEYWORD=HRG
KEYWORD=HRR
KEYWORD=IMAGEID
KEYWORD=INSTCFG
KEYWORD=INSTMODE
KEYWORD=INSTRUME
KEYWORD=KROT
KEYWORD=KWDICT
KEYWORD=LASTREW
KEYWORD=LAT_END
KEYWORD=LAT_STA
KEYWORD=LINFILE
KEYWORD=LO_LIM
KEYWORD=LON_END
KEYWORD=LON_STA
KEYWORD=MAPCRSYS
KEYWORD=MAPINTX
KEYWORD=MAPINTY
KEYWORD=MAPNXPOS
KEYWORD=MAPNYPOS
KEYWORD=MAPPING
KEYWORD=MCCSMODE
KEYWORD=MISSN-ID
KEYWORD=NAIF_ID
KEYWORD=NBELOW
KEYWORD=NFRAME
KEYWORD=NEXP
KEYWORD=NINT
KEYWORD=NODAMP
KEYWORD=NODANGLE
KEYWORD=NODBEAM
KEYWORD=NODCRSYS
KEYWORD=NODDING
KEYWORD=NODN
KEYWORD=NODPATT
KEYWORD=NODSETL
KEYWORD=NODSTYLE
KEYWORD=NODTIME
KEYWORD=NORDERS
KEYWORD=NSPAT
KEYWORD=NSPEC
KEYWORD=NSUM
KEYWORD=NT
KEYWORD=OBJECT

KEYWORD=OBS_ID
KEYWORD=OBSDEC
KEYWORD=OBSERVER
KEYWORD=OBSRA
KEYWORD=OBSTYPE
KEYWORD=OMEGAP
KEYWORD=OPERATOR
KEYWORD=ORDR_B
KEYWORD=ORDR_E
KEYWORD=ORDR_S
KEYWORD=ORDR_T
KEYWORD=ORIGIN
KEYWORD=OTPAT
KEYWORD=PAGAIN
KEYWORD=PINHOLE
KEYWORD=PIPELINE
KEYWORD=PIPEVERS
KEYWORD=PIXELWD
KEYWORD=PIXSCAL
KEYWORD=PLANID
KEYWORD=PLTSCALE
KEYWORD=PROCSTAT
KEYWORD=PRODTYPE
KEYWORD=READNOIS
KEYWORD=RESOLUN
KEYWORD=SATVAL
KEYWORD=SCANNING
KEYWORD=SDEG
KEYWORD=SIBS_X
KEYWORD=SIBS_Y
KEYWORD=SLIT
KEYWORD=SLITOFF
KEYWORD=SLITROT
KEYWORD=SLITVAL
KEYWORD=SLITWID
KEYWORD=SPACING
KEYWORD=SPECTEL1
KEYWORD=SPECTEL2
KEYWORD=SPIKEFAC
KEYWORD=SRCTYPE
KEYWORD=STDFAC
KEYWORD=SUBARRNO
KEYWORD=TELCNF
KEYWORD=TELEL
KEYWORD=TELEQUI
KEYWORD=TELESCOP
KEYWORD=TELDEC
KEYWORD=TELLOS
KEYWORD=TELRA
KEYWORD=TELVPA
KEYWORD=TELXEL
KEYWORD=TEMP_OUT
KEYWORD=TEMPPRI1
KEYWORD=TEMPPRI2

VERIFY THAT THIS IS THE CORRECT REVISION BEFORE USE

KEYWORD=TEMPPRI3
KEYWORD=TEMPSEC1
KEYWORD=THRAC
KEYWORD=TORTMODE
KEYWORD=TRACERR
KEYWORD=TRACKANG
KEYWORD=TRACMODE
KEYWORD=TRASH
KEYWORD=TSC-STAT
KEYWORD=UP_LIM
KEYWORD=UTCSTART
KEYWORD=UTCEND
KEYWORD=WAVECENT
KEYWORD=WAVENO0
KEYWORD=WNO0
KEYWORD=WVZ_END
KEYWORD=WVZ_STA
KEYWORD=X0BRL
KEYWORD=XD0
KEYWORD=XDELTA
KEYWORD=XDDGR
KEYWORD=XDFL
KEYWORD=XDFL0
KEYWORD=XDG
KEYWORD=XDLRDGR
KEYWORD=XDMRDGR
KEYWORD=XDR
KEYWORD=XORDER1
KEYWORD=Y0BRL
KEYWORD=ZA_END
KEYWORD=ZA_START
KEYWORD=HISTORY

Part X

Appendix B: Change notes for the EXES pipeline

15 Significant changes

Below are listed the most significant changes for the EXES pipeline over its history, highlighting impacts to science data products. See the data handbooks or user manuals associated with each release for more information.

For previously processed data, check the PIPEVERS keyword in the FITS header to determine the pipeline version used.

15.1 EXES Redux v2.0.0

User manual: Rev. B

- Adopted EXES PI team version for SOFIA DPS support as a facility class instrument.
- Modified NAIF ID handling to not write default -9999 value for sidereal targets.
- Added option to coadd frames across all files with outlier rejection.
- Added ASSC_MSN keyword to track all input mission IDs and ASSC_OBS to track all input OBS-IDs.

15.2 EXES Redux v1.6.0 (2019-04-02)

User manual: Rev. A

- Added handling for darks to allow for black-dark flat generation method
- Added option to reject 1 or 2 initial readout patterns before readout coadd
- Added option to use only destructive frames for readout coadds
- Modified options to allow manual selection of bad pixel masks
- Allowed manual overrides for the edges of order masks
- Added option to debounce in the spectral direction, as well as the spatial
- Added manual options to set asymmetric apertures with start and end values
- Added options to subtract a dark frame instead of nods, for sky emission spectra

15.3 EXES Redux v1.0.2 (2015-07-17)

User manual: Rev. A

- Modified nonlinearity correction to allow for a lower-limit plane in the coefficient reference file.

15.4 EXES Redux v1.0.1 (2015-05-14)

User manual: Rev. -

- Added ASSC_AOR keyword to track all input AOR-IDs.
- Attached reference atmospheric transmission data to spectra.
- Improved wavelength calibration.

15.5 EXES Redux v1.0.0 (2015-03-17)

User manual: Rev. -

- Initial release.