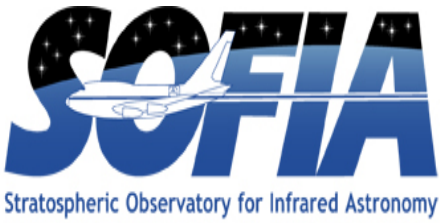


FORCAST Redux Users Manual

SCI-US-HBK-OP10-2003

Date: February 19, 2019
Revision: E

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

FORCAST Redux Users Manual

SCI-US-HBK-OP10-2003

AUTHOR:

Melanie Clarke, USRA, SOFIA DPS Development Lead	Date

William Vacca, USRA, SOFIA Associate Director for Science Data Systems	Date

APPROVAL:

William Vacca, USRA, SOFIA Associate Director for Science Data Systems	Date

REVISION HISTORY

REV	DATE	DESCRIPTION
-	12/30/13	Initial Release
A	1/7/15	Updated for Redux 1.0.1; re-organized to streamline distribution to GIs. Updates for Redux 1.0.3 OC2-2 release: Grism processing and grouping rules.
B	2/1/16	Numerous editorial and other updates to improve readability and use. Updates to reflect SPR PIPEDEV-226 Produce L3 FORCAST imaging products with physical units (Jy).
C	1/31/17	Updates for SPRs PIPEDEV-258 and PIPEDEV-276: Improvements to imaging and grism calibration processes.
D	4/19/2017	Updates for SPRs PIPEDEV-280, PIPEDEV-282, and PIPEDEV-283: Incorporating grism response files, modifying imaging exposure maps, and supporting imaging mosaics.
E	2/19/19	Updates for the FORCAST Redux 1.4.0 release: PIPEDEV-340, support slit scan reductions.

FORCAST Redux User's Manual

Release : SCI-US-HBK-OP10-2003 Rev. E

M. Clarke, W. Vacca

Feb 19, 2019

Contents

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 at location: / Software Management Development or Verification / Pipelines (DPS).

This manual applies to FORCAST Redux version 1.4.0.

Part II

SI Observing Modes Supported

1 FORCAST observing techniques

Because the sky is so bright in the mid-infrared (MIR) relative to astronomical sources, the way in which observations are made in the MIR is considerably different from the more familiar way they are made in the optical. Any raw image of a region in the MIR is overwhelmed by the background sky emission. The situation is similar to trying to observe in the optical during the day: the bright daylight sky swamps the detector and makes it impossible to see astronomical sources in the raw images.

In order to remove the background from the MIR image and detect the faint astronomical sources, observations of another region (free of sources) are made and the two images are subtracted. However, the MIR is highly variable, both spatially and temporally. It would take far too long (on the order of seconds) to reposition a large telescope to observe this sky background region: by the time the telescope had moved and settled at the new location, the sky background level would have changed so much that the subtraction of the two images would be useless. In order to avoid this problem, the secondary mirror of the telescope (which is considerably smaller than the primary mirror) is tilted, rather than moving the entire telescope. This allows observers to look at two different sky positions very quickly (on the order of a few to ten times per second), because tilting the secondary by an angle θ moves the center of the field imaged by the detector by θ on the sky. Tilting the secondary between two positions is known as "chopping". FORCAST observations are typically made with a chopping frequency of 4 Hz. That is, every 0.25 sec, the secondary is moved between the two observing positions.

Chopping can be done either symmetrically or asymmetrically. Symmetric chopping means that the secondary mirror is tilted symmetrically about the telescope optical axis (also known as the boresight) in the two chop positions. The distance between the two chop positions is known as the chop throw. The distance between the boresight and either chop position is known as the chop amplitude and is equal to half the chop throw (see Fig. 1).

Asymmetric chopping means that the secondary is aligned with the telescope boresight in one position, but is tilted away from the boresight in the chop position. The chop amplitude is equal to the chop throw in this case (see Fig. 2).

Unfortunately, moving the secondary mirror causes the telescope to be slightly misaligned, which introduces optical distortions (notably the optical aberration known as coma) and additional background emission from the telescope

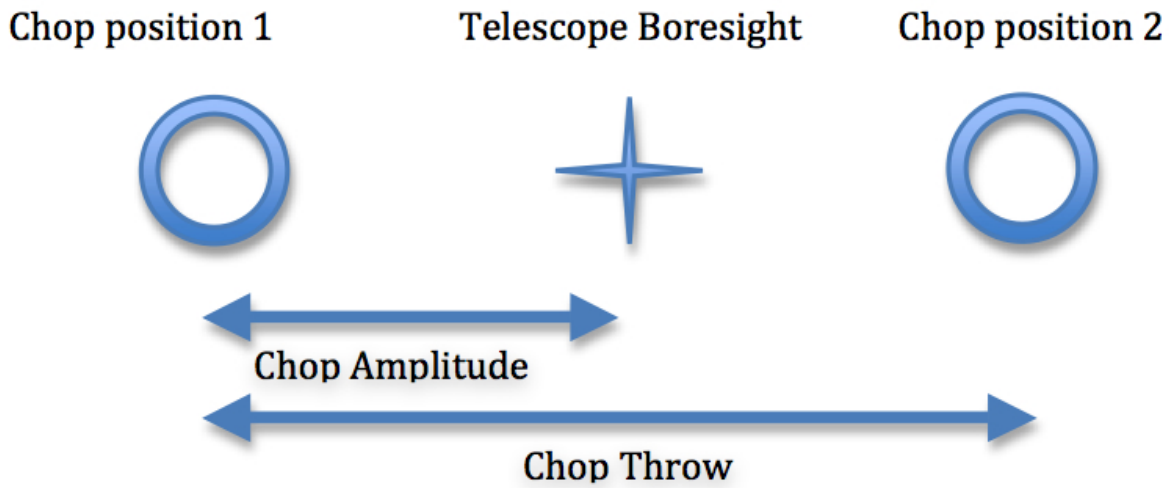


Fig. 1: Symmetric Chop

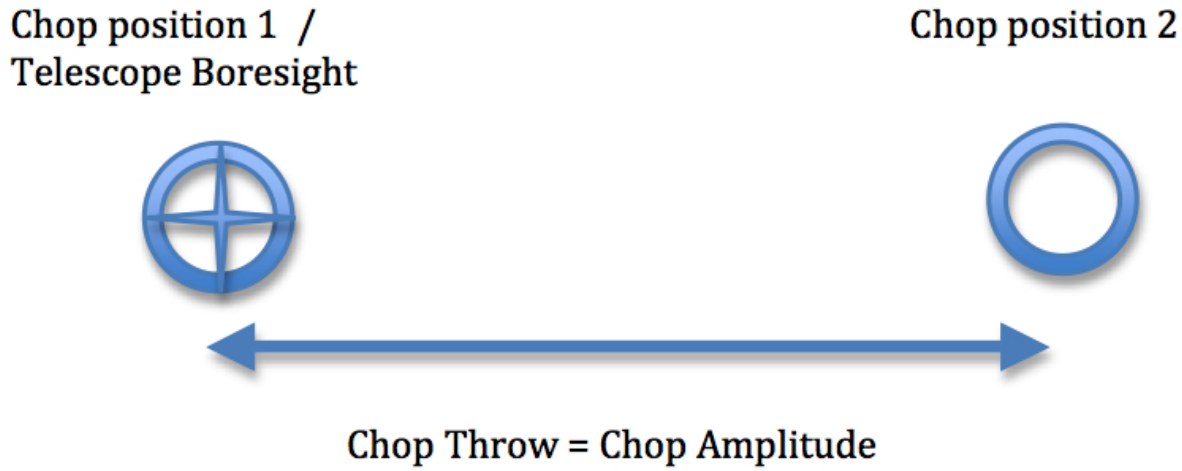


Fig. 2: Asymmetric Chop

(considerably smaller than the sky emission but present nonetheless) in the images. The optical distortions can be minimized by tilting the secondary only tiny fractions of a degree. The additional telescopic background can be removed by moving the entire telescope to a new position and then chopping the secondary again between two positions. Subtracting the two chop images at this new telescope position will remove the sky emission but leave the additional telescopic background due to the misalignment; subtracting the result from the chop-subtracted image at the first telescope position will then remove the background. Since the process of moving to a new position is needed to remove the additional background from the telescope, not the sky, it can be done on a much longer timescale. The variation in the telescopic backgrounds occurs on timescales on the order of tens of seconds to minutes, much slower than the variation in the sky emission.

This movement of the entire telescope, on a much longer timescale than chopping, is known as nodding. The two nod positions are usually referred to as nod A and nod B. The distance between the two nod positions is known as the nod throw or the nod amplitude. For FORCAST observations, nods are done every 5 to 30 seconds. The chop-subtracted images at nod position B are then subtracted from the chop-subtracted images at nod position A. The result will be an image of the region, without the sky background emission or the additional emission resulting from tilting the secondary during the chopping process. The sequence of chopping in one telescope position, nodding, and chopping again in a second position is known as a chop/nod cycle.

Again, because the MIR sky is so bright, deep images of a region cannot be obtained (as they are in the optical) by simply observing the region for a long time with the detector collecting photons continuously. As stated above, the observations require chopping and nodding at fairly frequent intervals. Hence, deep observations are made by "stacking" a series of chop/nod images. Furthermore, MIR detectors are not perfect, and often have bad pixels or flaws. In order to avoid these defects on the arrays, and prevent them from marring the final images, observers employ a technique known as "dithering." Dithering entails moving the position of the telescope slightly with respect to the center of the region observed each time a new chop/nod cycle is begun, or after several chop/nod cycles. When the images are processed, the observed region will appear in a slightly different place on the detector. This means that the bad pixels do not appear in the same place relative to the observed region. The individual images can then be registered and averaged or median-combined, a process that will eliminate (in theory) the bad pixels from the final image.

2 Available chopping modes

2.1 Symmetric chopping modes: C2N and C2ND

FORCAST acquires astronomical observations in two symmetric chopping modes: two-position chopping with no nodding (C2) and two-position chopping with nodding (C2N). Dithering can be implemented for either mode; two-position chopping with nodding and dithering is referred to as C2ND. The most common observing methods used are C2N and C2ND. C2ND is conceptually very similar the C2N mode: the only difference is a slight movement of the telescope position after each chop/nod cycle.

FORCAST can make two types of C2N observations: Nod Match Chop (NMC) and Nod Perpendicular to Chop (NPC). The positions of the telescope boresight, the two chop positions, and the two nod positions for these observing types are shown below (Fig. 3 and Fig. 4).

C2N: Nod Match Chop (NMC)

In the NMC mode, the telescope is pointed at a position half of the chop throw distance away from the object to be observed, and the secondary chops between two positions, one of which is centered on the object. The nod throw has the same magnitude as the chop throw, and is in a direction exactly 180 degrees from that of the chop direction. The final image generated by subtracting the images obtained for the two chop positions at nod A and those at nod B, and then subtracting the results. This will produce three images of the star, one positive and two negative, with the positive being twice as bright as the negatives.

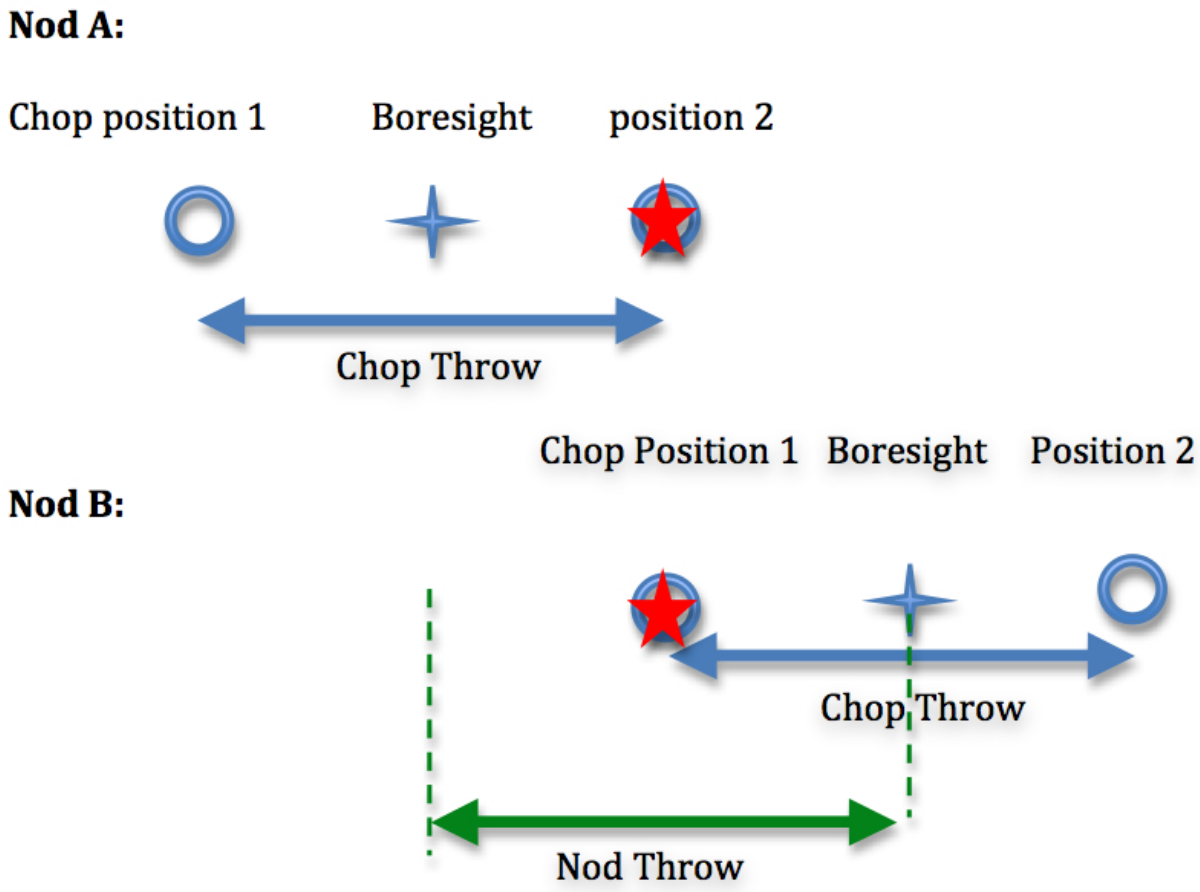


Fig. 3: Nod Match Chop mode

C2N: Nod Perpendicular to Chop (NPC)

In the NPC mode, the telescope is offset by half the nod throw from the target in a direction perpendicular to the chop direction, and the secondary chops between two positions. The nod throw usually (but not necessarily) has the same magnitude as the chop, but it is in a direction perpendicular to the chop direction. The final image is generated by subtracting the images obtained for the two chop positions at nod A and those at nod B, and then subtracting the results. This will produce four images of the star in a rectangular pattern, with the image values alternating positive and negative.

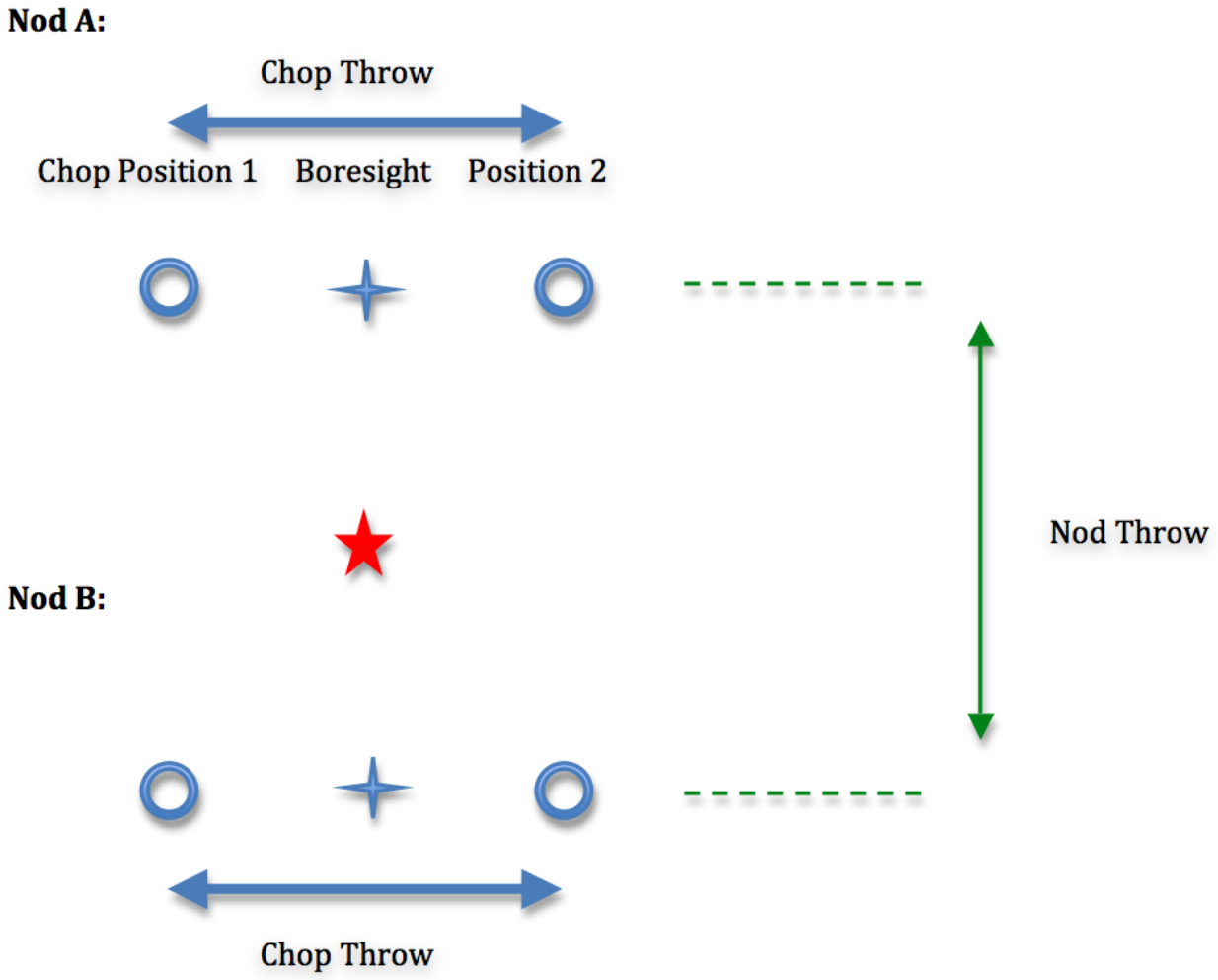


Fig. 4: Nod Perpendicular to Chop mode

2.2 Asymmetrical chopping modes: C2NC2 and NXCAC

FORCAST also has an asymmetrical chop mode, known as C2NC2. In this mode, the telescope is first pointed at the target (position A). In this first position, the secondary is aligned with the boresight for one observation and then is tilted some amount (often 180-480 arc seconds) for the second (asymmetrically chopped) observation. This is an asymmetric C2 mode observation. The telescope is then slewed some distance from the target, to a sky region without sources (position B), and the asymmetric chop pattern is repeated. The time between slews is typically 30 seconds.

There is an additional asymmetric mode chopping mode, called NXCAC (nod not related to chop/asymmetrical chop).

This mode replaces the C2NC2 mode when the GI wants to use an asymmetrical chop for a grism observation. This mode is taken with an ABBA nod pattern, like the C2N mode (not ABA, like C2NC2). The nods are packaged together, so data from this mode will reduce just like the C2N mode. The reason for adding this mode stems from the need to define our large chops and nods in ERF (equatorial reference frame), and dither in SIRF (science instrument reference frame) along the slit.

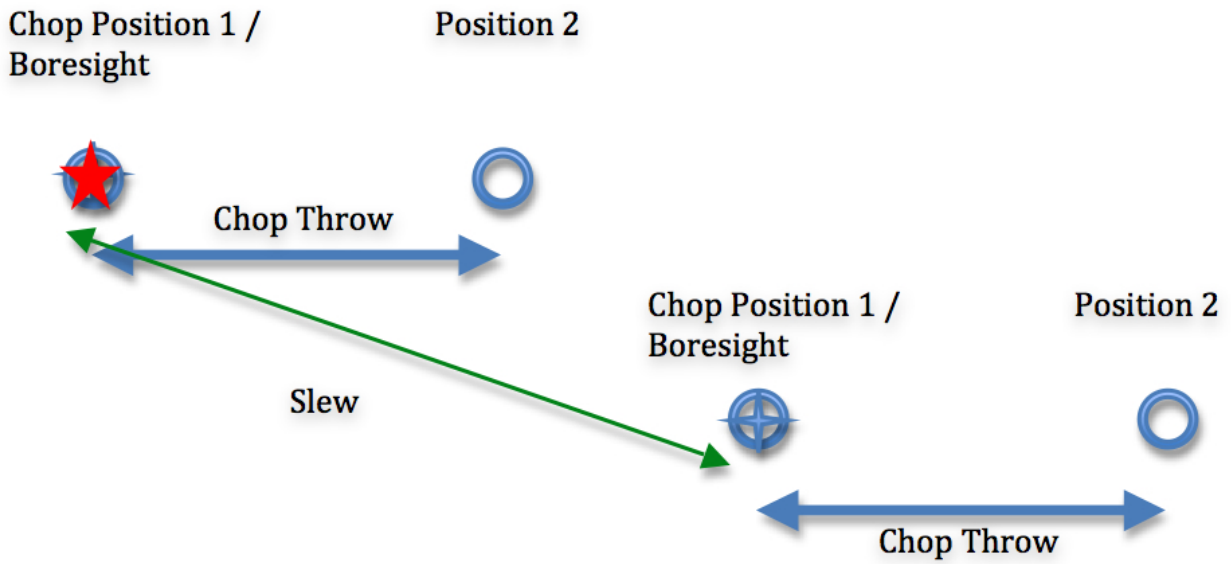


Fig. 5: C2NC2 mode

2.3 Spectral imaging mode: SLITSCAN

Similar to the C2ND mode for imaging, the SLITSCAN mode for grism observations allows a combination of chopping and nodding with telescope moves to place the spectral extraction slit at different locations in the sky.

In slit-scan observations, a chop/nod cycle is taken at a series of positions, moving the slit slowly across an extended target after each cycle. In this mode, the different telescope positions may be used to generate both extracted spectra at each position and a spatial/spectral cube that combines all the observations together into a spectral map of the source.

Part III

Algorithm Description

3 Overview of data reduction steps

This section will describe, in general terms, the major algorithms used by the FORCAST Redux pipeline, to reduce a FORCAST observation.

Redux is a software package written in IDL that is designed to be a framework for executing any number or combination of data reduction algorithms. For FORCAST, it has been developed to support seamlessly running image processing algorithms from the DRIP IDL package, originally developed by the FORCAST team, alongside spectral

extraction algorithms from the FSpextool IDL package, originally developed for the SpeX instrument, and photometry and flux calibration algorithms from the PipeCal IDL package, developed by the SOFIA Data Processing System (DPS) team. Redux can run in an automatic batch mode, integrated with DPS infrastructure, or it can run with a graphical front end as a quick-look data viewer in flight or during manual data reduction and analysis.

The pipeline applies a number of corrections to each input file, regardless of the chop/nod mode used to take the data. The initial steps used for imaging and grism modes are nearly identical; points where the results or the procedure differ for either mode are noted in the descriptions below. After preprocessing, individual images or spectra of a source must be combined to produce the final data product. This procedure depends strongly on the instrument configuration and chop/nod mode.

All raw files are first processed as images, with algorithms developed for the DRIP reduction package. Spectroscopy files then undergo spectral extraction and combination of the resulting one-dimensional spectra, using algorithms from the FSpextool reduction package. See Fig. 6 for a flowchart of all processing steps used by the pipeline.

4 Reduction algorithms

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

- Steps common to imaging and spectroscopy modes
  - Cleaning of bad pixels
  - Droop effect correction
  - Nonlinearity correction
  - Background subtraction (chop/nod stacking)
  - Jailbar removal (crosstalk correction)
- Imaging-specific steps
  - Optical distortion correction
  - Image shift and rotation (merging)
  - Image registration
  - Telluric correction
  - Combining multiple observations (coadding)
  - Flux calibration
- Spectroscopy-specific steps
  - Spectral extraction
  - Aperture merging
  - Flux calibration and telluric correction
  - Combining multiple observations
  - Spectral cube construction from slit-scan observations
  - Response file generation from standards

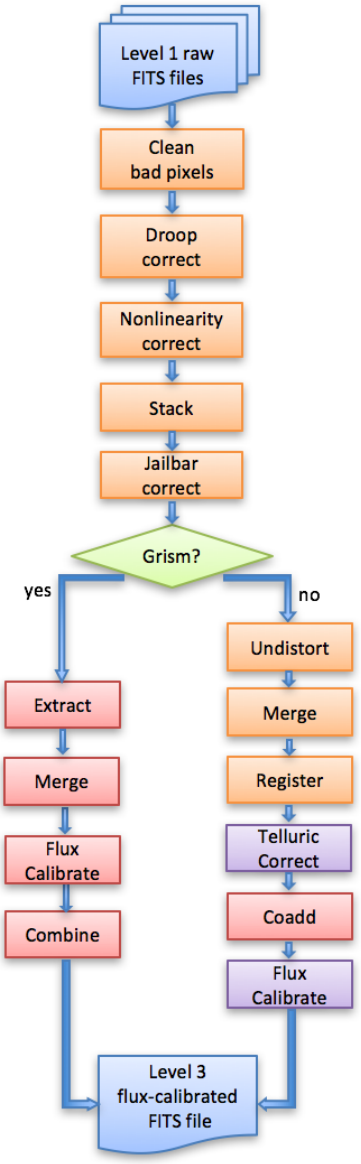


Fig. 6: Processing steps for imaging and grism data. Orange boxes indicate steps that use algorithms from the DRIP package. Red boxes indicate steps that use FSpextool algorithms. Purple boxes use algorithms from the PipeCal package.

4.1 Steps common to imaging and spectroscopy modes

Bad pixel cleaning

Bad pixels in the FORCAST arrays take the form of hot pixels (with extreme dark current) or pixels with very different response (usually much lower) than the surrounding pixels. The pipeline minimizes the effects of bad pixels by using a bad pixel mask to identify their locations and then replacing the bad pixels with values derived from the surrounding operational pixels. The DRIP clean function is built around the IDL procedure MASKINTERP, written by J. Harrington, which fits a 2-dimensional surface to an aperture in the image centered on the bad pixel(s) while ignoring the bad pixel(s) identified in the mask. MASKINTERP then replaces the bad pixels with the corresponding values of the surface fit. For FORCAST, MASKINTERP is set to use a planar surface with an aperture radius of 6 pixels.

The bad pixel map for both FORCAST channels is currently produced manually, independent of the pipeline. The mask is a 256x256 image with pixel value = 0 for bad pixels and pixel value = 1 otherwise.

Droop correction

The FORCAST arrays and readout electronics exhibit a linear response offset caused by the presence of a signal on the array. This effect is called 'droop' since the result is a reduced signal. Droop results in each pixel having a reduced signal that is proportional to the total signal in the 15 other pixels in the row read from the multiplexer simultaneously with that pixel. The effect, illustrated in Fig. 7, is an image with periodic spurious sources spread across the array rows. The droop correction removes the droop offset by multiplying each pixel by a value derived from the sum of every 16th pixel in the same row all multiplied by an empirically determined offset fraction: droopfrac = 0.0035. This value is a configurable parameter, as some data may require a smaller droop fraction to avoid overcorrection of the effect. Overcorrection may look like an elongated smear along the horizontal axis, near a bright source (see Figure 8). Note that while droop correction typically removes the effect near the source, there may be lingering artifacts in other areas of the image if the source was very bright, as in Fig. 7.

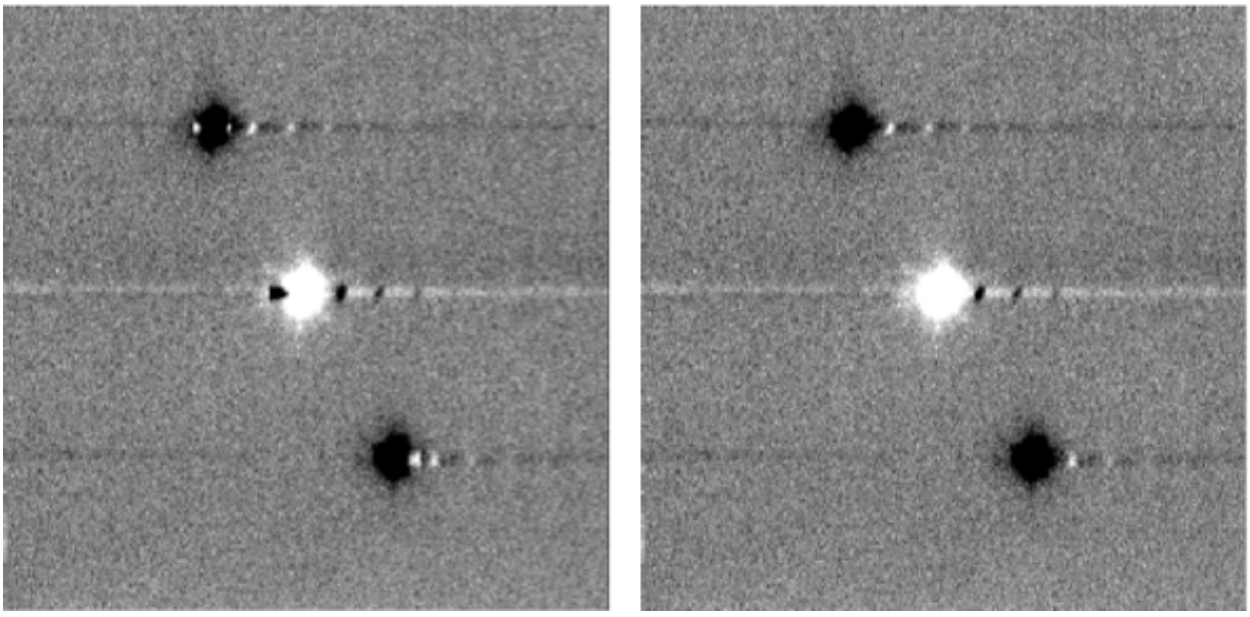


Fig. 7: Background-subtracted FORCAST images of a star with droop (left) and with the droop correction applied (right).

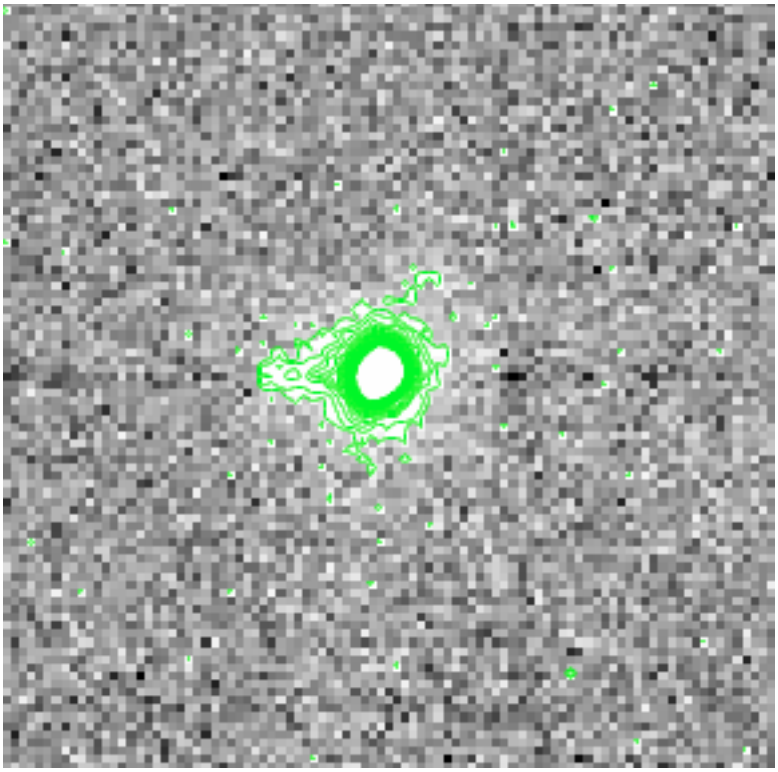


Fig. 8: Overcorrected droop effect, appearing as an elongated smear on the bright central source.

Nonlinearity correction

In principle, the response of each of the pixels in the FORCAST detector arrays should be linear with incident flux. In practice, the degree to which the detector is linear depends on the level of charge in the wells relative to the saturation level. Empirical tests optimizing signal-to-noise indicate that signal levels in the neighborhood of 60% of full well for a given detector capacitance in the FORCAST arrays have minimal departures from linear response and optimal signal-to-noise. For a given background level we can keep signal levels near optimal by adjusting the detector readout frame rate and detector capacitance. Since keeping signals near 60% of saturation level is not always possible or practical, we have measured response curves (response in analog-to-digital units (ADU) as a function of well depth for varying background levels) that yield linearity correction factors. These multiplicative correction factors linearize the response for a much larger range of well depths (about 15% – 90% of saturation). The linearity correction is applied globally to FORCAST images prior to background subtraction. The pipeline first calculates the background level for a sub-image, and then uses this level to calculate the linearity correction factor. The pipeline then applies the correction factor to the entire image.

Background subtraction (chop/nod stacking)

Background subtraction is accomplished by subtracting chopped image pairs and then subtracting nodded image pairs.

For C2N/NPC imaging mode with chop/nod on-chip (i.e. chop throws smaller than the FORCAST field of view), the four chop/nod images in the raw data file are reduced to a single stacked image frame with a pattern of four background-subtracted images of the source, two positive and two negative. For chop/nod larger than the FORCAST field of view the raw files are reduced to a single frame with one background-subtracted image of the source.

For the C2N/NPC spectroscopic mode, either the chop or the nod is always off the slit, so there will be two traces in the subtracted image: one positive and one negative. If the chop or nod throw is larger than the field of view, there will be a single trace in the image.

In the case of the C2N/NMC mode for either imaging or spectroscopy, the nod direction is the same as the chop direction with the same throw so that the subtracted image frame contains three background-subtracted images of the source. The central image or trace is positive and the two outlying images are negative. If the chop/nod throw is larger than the FORCAST field of view, there will be a single image or trace in the image.

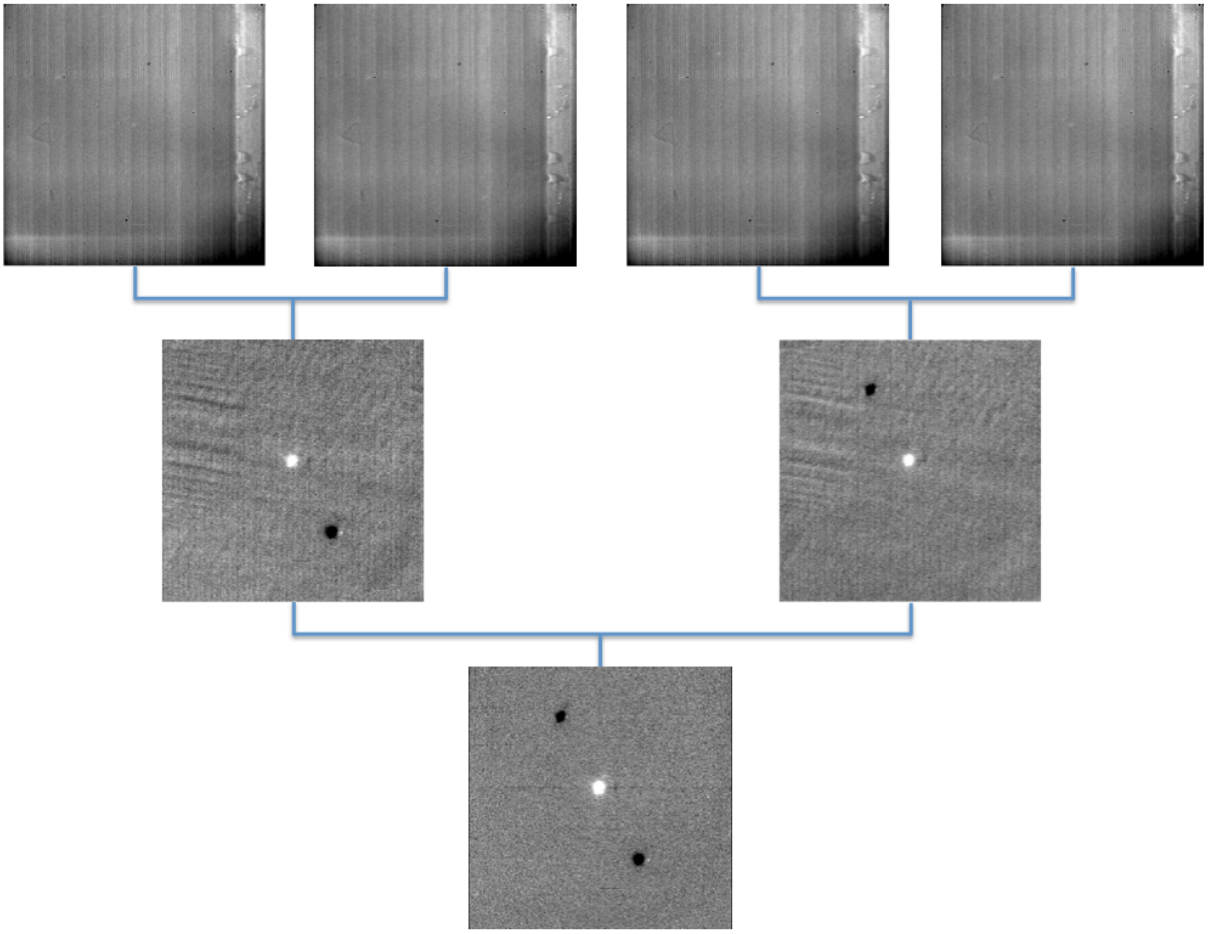


Fig. 9: Images at two stages of background subtraction in imaging NMC mode: raw frames (upper row), chop-subtracted (middle row), chop/nod-subtracted (lower row). Four raw frames produce a single stacked image.

C2NC2 raw data sets for imaging or spectroscopy consist of a set of 5 FITS files, each with 4 image planes containing the chop pairs for both the on-source position (position A) and the blank sky position (position B). The four planes can be reduced in the same manner as any C2N image by first subtracting chopped image pairs for both and then subtracting nodded image pairs. The nod sequence for C2NC2 is $A_1 B_1 A_2 A_3 B_2 A_4 A_5 B_3$, where the off-source B nods are shared between some of the files (shared B beams shown in bold):

File 1 = $A_1 \boldsymbol{B_1}$

File 2 = $\boldsymbol{B_1} A_2$

File 3 = $A_3 \boldsymbol{B_2}$

File 4 = $\boldsymbol{B_2} A_4$

File 5 = $A_5 \boldsymbol{B_3}$

At this point, the background in the chop/nod-subtracted stack should be zero, but if there is a slight mismatch between the background levels in the individual frames, there may still remain some small residual background level. After

stacking, the pipeline estimates this residual background by taking the mode of the image data in a central section of the image, and then subtracts this level from the stacked image.

The last step in the imaging stack pipeline step is to convert pixel data from analog-to-digital units (ADU) per frame to mega-electrons per second (Me/s) using the gain and frame rate used for the observation.

For grism data, this conversion is applied as well. Then, individual frames taken at the same dither position may be combined together to increase the signal-to-noise in the two-dimensional spectral image. Since the spectral image is not yet distortion-corrected, spectra taken at different dither positions are not combined together at this point.

Jailbar removal (Crosstalk correction)

The FORCAST array readout circuitry has a residual, or latent, signal that persists when pixels have high contrast relative to the surrounding pixels. This can occur for bad pixels or for bright point sources. This residual is present not only in the affected pixels, but is correlated between all pixels read by the same one of sixteen multiplexer channels. This results in a linear pattern of bars, spaced by 16 pixels, known as "jailbars" in the background-subtracted (stacked) images. Jailbars can interfere with subsequent efforts to register multiple images since the pattern can dominate the cross-correlation algorithm sometimes used in image registration. The jailbars can also interfere with photometry in images and with spectral flux in spectroscopy frames.

The pipeline attempts to remove jailbar patterns from the background-subtracted images by replacing pixel values by the median value of pixels in that row that are read by the same multiplexer channel (i.e. every 16th pixel in that row starting with the pixel being corrected). The jailbar pattern is located by subtracting a 1-dimensional (along rows) median filtered image from the raw image.

4.2 Imaging-specific steps

Optical distortion correction

The FORCAST optical system introduces anamorphic magnification and barrel distortion in the images. The distortion correction uses pixel coordinate offsets for a grid of pinholes imaged in the lab and a 2D polynomial warping function to resample the 256x256 pixels to an undistorted grid. The resulting image is 262x247 pixels with image scale of 0.768"/pixel for a corrected field of view of 3.4x3.2 arc minutes. The distortion-corrected image is centered in a 656x656 pixel array to accommodate the distortion correction and to make room for subsequent shifting and adding of chop/nod images and for image rotation prior to the final registration and coaddition steps of the reduction process. Pixels outside of the detector area are set to NaN to distinguish them from real data values (see Fig. 11).

There is no distortion correction for the grism mode since the extracted spectra have a wavelength and spatial calibration applied directly to the array rows.

Image shift and rotation (merging)

The stack step of the pipeline in imaging mode may produce images with multiple positive and negative source images, depending on the chop/nod mode used for data acquisition. These positive and negative sources may be merged by copying, shifting, and re-combining the image in order to increase the signal-to-noise of the observation. The final image must then be rotated by the nominal sky angle, so that North is up and East is left in the final image (see Fig. 11).

The merge pipeline step makes a number of copies of the stacked image, shifts them by the chop and nod throws used in data acquisition, and adds or subtracts them (depending on whether the image is a positive or negative background-subtracted image). The pipeline can use three different methods for registration in the merge process:

- centroid of the brightest point source in the stacked images
- cross-correlation, usually best for extended or nebulous sources

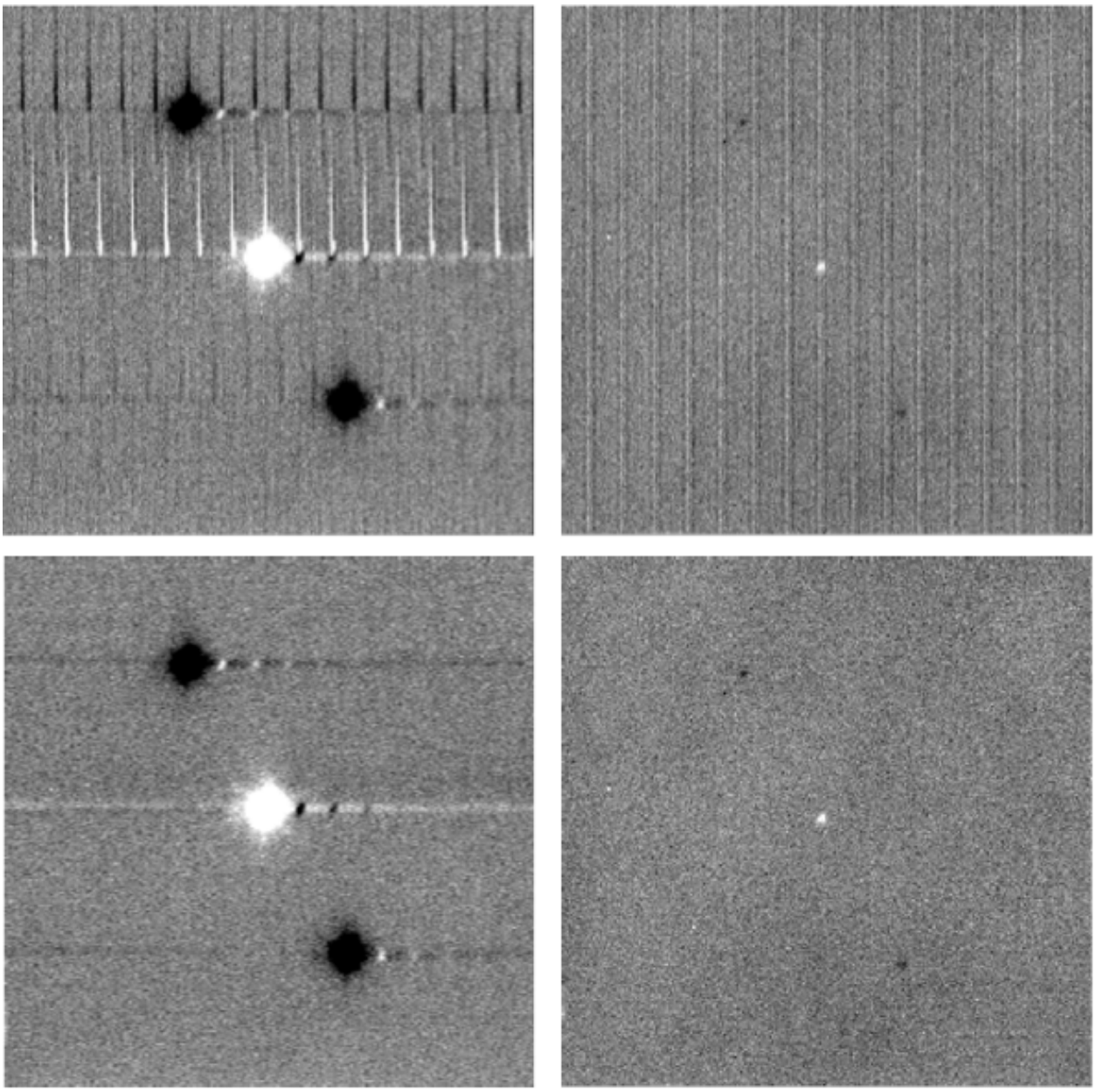


Fig. 10: Cross-talk correction for a bright point source (left), and faint source (right). Images on the top are before correction; images on the bottom are after correction.

- chop/nod data from the FITS header

The default for flux standards is to use centroiding, as it is usually the most precise method. If merging is desired for science images that do not contain a bright, compact source, the header data method is usually the most reliable. After the shifting and adding, the final merged image consists of a positive image of the source surrounded by a number of positive and negative residual source images left over from the merging process. The central image is the source to use for science.

For the NPC imaging mode with chop/nod amplitude smaller than the field of view, the stack step produces a single stacked image frame with a pattern of four background-subtracted images of the source, two of them negative. The merge step makes four copies of the stacked frame, then shifts each using the selected algorithm. It adds or subtracts each copy, depending on whether the source is positive or negative.

For the NMC imaging mode with chop/nod amplitude smaller than the field of view, the stacked image contains three background-subtracted sources, two negative, and one positive (see Fig. 9). The positive source has double the flux of the negative ones, since the source falls in the same place on the detector for two of the chop/nod positions. The merge step for this mode makes three copies of the image, shifts the two negative sources on top of the positive one, and then subtracts them (see Figure 11). Pixels with no data are set to NaN.

While performing the merge, the locations of overlap for the shifted images are recorded. For NPC mode, the final merged image is normalized by dividing by the number of overlapping images at each pixel. For NMC mode, because the source is doubled in the stacking step, the final merged image is divided by the number of overlapping images, plus one. In the nominal case, if all positive and negative sources were found and coadded, the signal in the central source, in either mode, should now be the average of four observations of the source. If the chop or nod was relatively wide, however, and one or more of the extra sources were not found on the array, then the central source may be an average of fewer observations.

For either NPC or NMC imaging modes, with chop/nod amplitude greater than half of the array, there is no merging to be done, as the extra sources are off the detector. However, for NMC mode, the data is still divided by 2 to account for the doubled central source. For C2NC2 mode, the chops and telescope moves-to-sky are always larger than the FORCAST field of view; merging is never required for this mode. It may also be desirable to skip the merging stage for crowded fields-of-view, as the merge artifacts may be confused with real sources. In all imaging cases, whether or not the shifting-and-adding is performed, the merged image is rotated by the sky angle at the end of the merge step. Any remaining NaN border padding the edges of the image is then removed, so that the image is only as large as it needs to be.

Image registration

In order to combine multiple imaging observations of the same source, each image must be registered to a reference image, so that the pixels from each image correspond to the same location on the sky. The registration algorithm uses the same three options for registration of images as the merge step (centroid, cross-correlation, or FITS header data). The first image in the set is treated as the reference image; the algorithm uses header data to shift this image to account for any initial dither offset. For all subsequent images, the specified algorithm is used to find the shift required to register it to the first image. The interpolation order of the shift may be 0 (integer pixel shifts), 1 (bilinear interpolation), or 3 (the default; cubic interpolation). Using a shift order of 1 or 3 will allow sub-pixel shifts to be performed.

Telluric correction

For accurate flux calibration, the pipeline must first correct for the atmospheric opacity at the time of the observation. In order to combine images taken in different atmospheric conditions, or at different altitudes or zenith angles, the pipeline corrects the flux in each individual registered file for the estimated atmospheric transmission during the observations, based on the altitude and zenith angle at the time when the observations were obtained, relative to that computed for a reference altitude (41,000 feet) and reference zenith angle (45 degrees), for which the instrumental response has been calculated. The atmospheric transmission values are derived from the ATRAN code kindly provided to the SOFIA

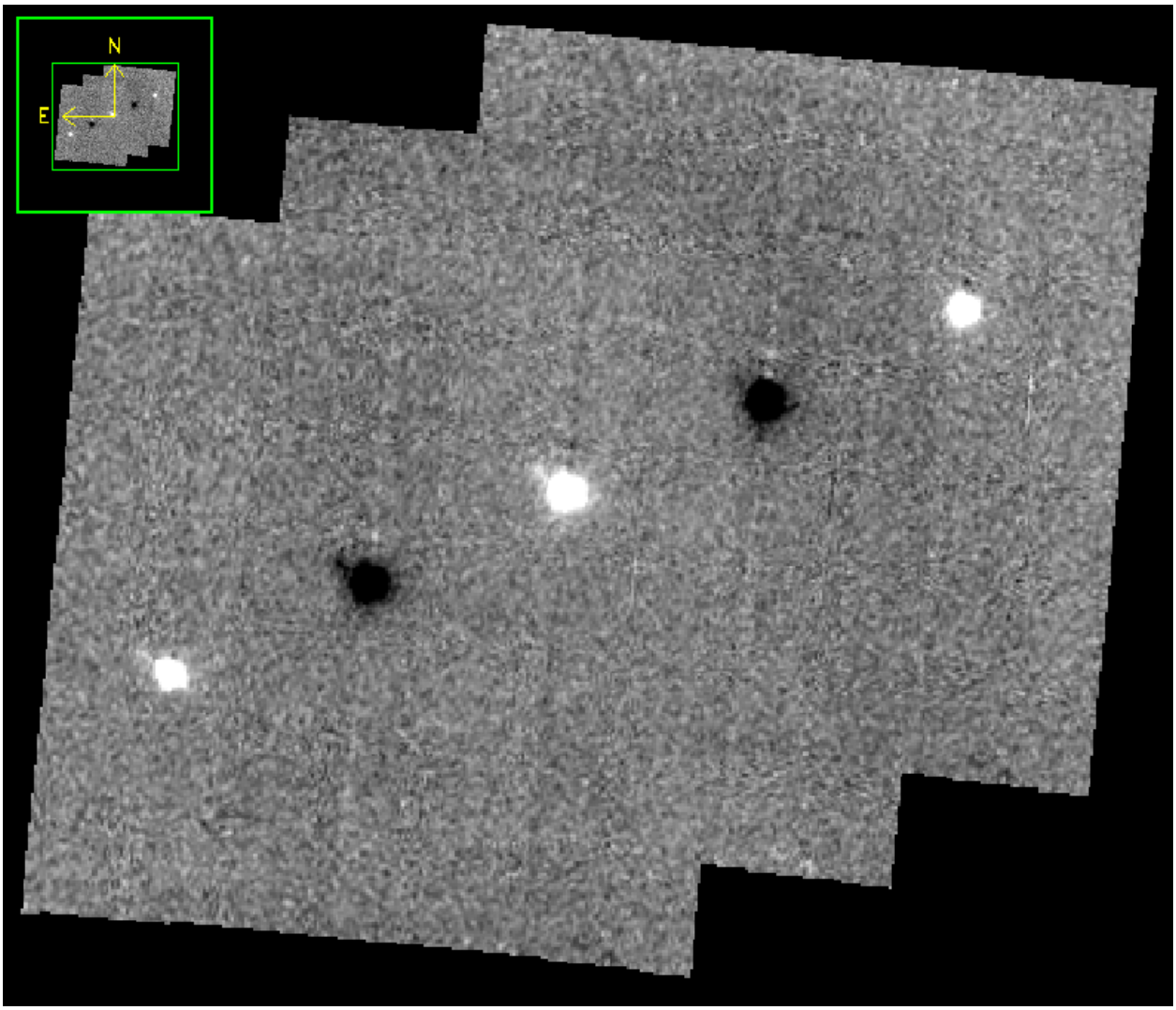


Fig. 11: The NMC observation of Fig. 9, after merging. Only the central source should be used for science; the other images are artifacts of the stacking and merging procedure. Note that the merged image is rotated to place North up and East left.

program by Steve Lord. The pipeline applies the telluric correction factor directly to the flux in the image, and records it in the header keyword TELCORR.

After telluric correction, the pipeline uses the PipeCal package to perform aperture photometry on all observations that are marked as flux standards (FITS keyword OBSTYPE = STANDARD_FLUX). The brightest source in the field is fit with a Moffat profile to determine its centroid, and then its flux is measured, using an aperture of 12 pixels and a background region of 15-25 pixels. The aperture flux and error, as well as the fit characteristics, are recorded in the FITS header, to be used in the flux calibration process.

Coadding multiple observations

After registration and scaling, the pipeline coadds multiple observations of the same source with the same instrument configuration and observation mode. The image combination is performed by an FSpextool algorithm that allows rejection of outlying values. The default combination statistic is a median; a robust weighted mean may also be used. For flux standards, the photometry is repeated on the coadded image, in the same way it was performed on the individual images.

Flux calibration

For imaging, flux calibration factors are typically calculated from all standards observed within a flight series, as detailed in the flux calibration section, below. These calibration factors are recorded in the headers of all telluric-corrected and coadded products, then as a final step in the pipeline, are used to produce a flux-calibrated image. The final Level 3 product has image units of Jy per pixel.[1] See the flux calibration section, below, for more information.

Mosaic

In some cases, it may be useful to stack together separate calibrated observations of the same target. In order to create a deeper image of a faint target, for example, observations taken across multiple flights may be combined together. Large maps may also be generated by taking separate observations, and stitching together the results. In these cases, the pipeline may register these files and coadd them, using the same methods as in the initial registration and coadd steps. The output product is a LEVEL_4 mosaic.

4.3 Spectroscopy-specific steps

Spectral extraction and merging

The FSpextool spectral extraction algorithms used by Redux offer two different extraction methods depending on the nature of the target source, as defined by the SRCTYPE FITS keyword. 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, which simply sums the flux over an aperture. The extraction aperture can be specified directly by the user or determined automatically from the spatial distribution of the flux over the slit (the spatial profile).

For the NPC grism mode, with chop/nod amplitude less than the field of view, there will be a positive and a negative spectral trace in the stacked image. The pipeline extracts both, multiplying the negative spectrum by -1 to make it positive. It then merges the spectra by coadding them and dividing by 2.

For the NMC grism mode, with chop/nod amplitude less than the field of view, and chopping or nodding along the slit, there will be a positive and two negative spectral traces in the stacked image. In this case, the pipeline extracts all three spectra, multiplying the negative ones by -1. It then merges the spectra by coadding them and dividing by four

[1] Note that earlier versions of this pipeline did not produce a final calibrated file. The final Level 3 products had image units of Me/sec, with the flux calibration factor (Me/sec/Jy) recorded in the FITS header keyword, CALFCTR.

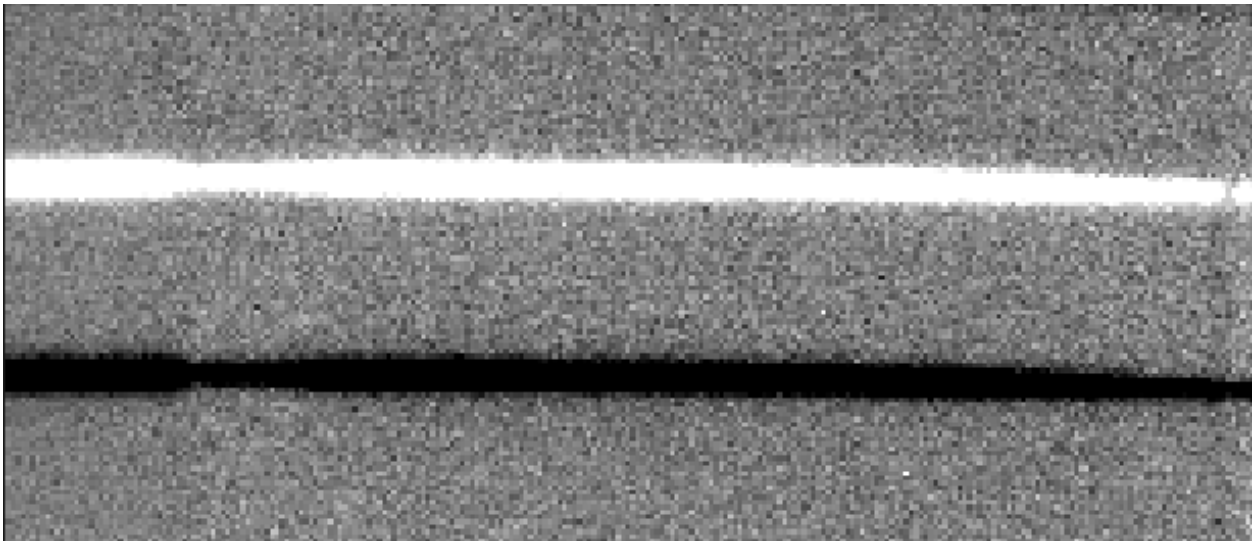


Fig. 12: NPC spectrum after stacking. Both spectra will be extracted.

(to account for the doubled central source). If the chop/nod amplitude for the NMC mode is larger than the field of view or the observation was chopped off the slit, there will be only a single spectral trace. In this case, the pipeline extracts this spectrum, and then simply divides it by two to account for the doubled source.

All other grism modes produce a single positive spectral trace in the stacked image, which the pipeline extracts directly.

All spectral extractions require a 2D wavelength calibration map, identifying the wavelength associated with each pixel in the array, to extract spectra along lines of constant wavelength. This simultaneously corrects for any distortion in the spatial or spectral directions, and wavelength-calibrates the output spectrum. Wavelength calibration maps are generated from identifications of sky emission and telluric absorption lines and a polynomial fit to centroids of those features in pixel space for each row (i.e. along the dispersion direction). The derivation of a wavelength calibration is an interactive process, but application of the derived wavelength calibration is an automatic part of the data reduction pipeline. The default wavelength calibration is expected to be good to within approximately one pixel in the output spectrum.

During extraction, the wavelength and spatial calibration images are used to resample the 2D spectral image into a spatially and spectrally rectified image. This image is saved to disk, alongside the extracted spectra.

Telluric correction and flux calibration

Before the extracted spectra are combined, a correction that accounts for spatial variations in the instrumental throughput is applied to each spectrum. This “slit correction function” is a function of the position of the science target spectrum along the slit relative to that used for the standard stars. The extracted spectra are then corrected individually for instrumental response and atmospheric transmission, a process that yields a flux-calibrated spectrum in units of Jy/pixel. See the section on flux calibration, below, for more detailed information. The individual spectra of each target can then be combined.

The rectified spectral image produced during extraction is also corrected for the slit function and atmospheric transmission, and is calibrated to physical units in the same manner. Each row of the image is divided by the same correction as the extracted spectrum. This image is suitable for custom extractions of extended fields: a sum over any number of rows in the image produces a flux-calibrated spectrum of that region, in the same units as the spectrum produced directly by the pipeline.

Note that the FITS header for this product (PRODTYPE=’calrectimg’) contains a full spatial and spectral WCS that can be used to identify the coordinates of any spectra so extracted. The primary WCS identifies the spatial direction

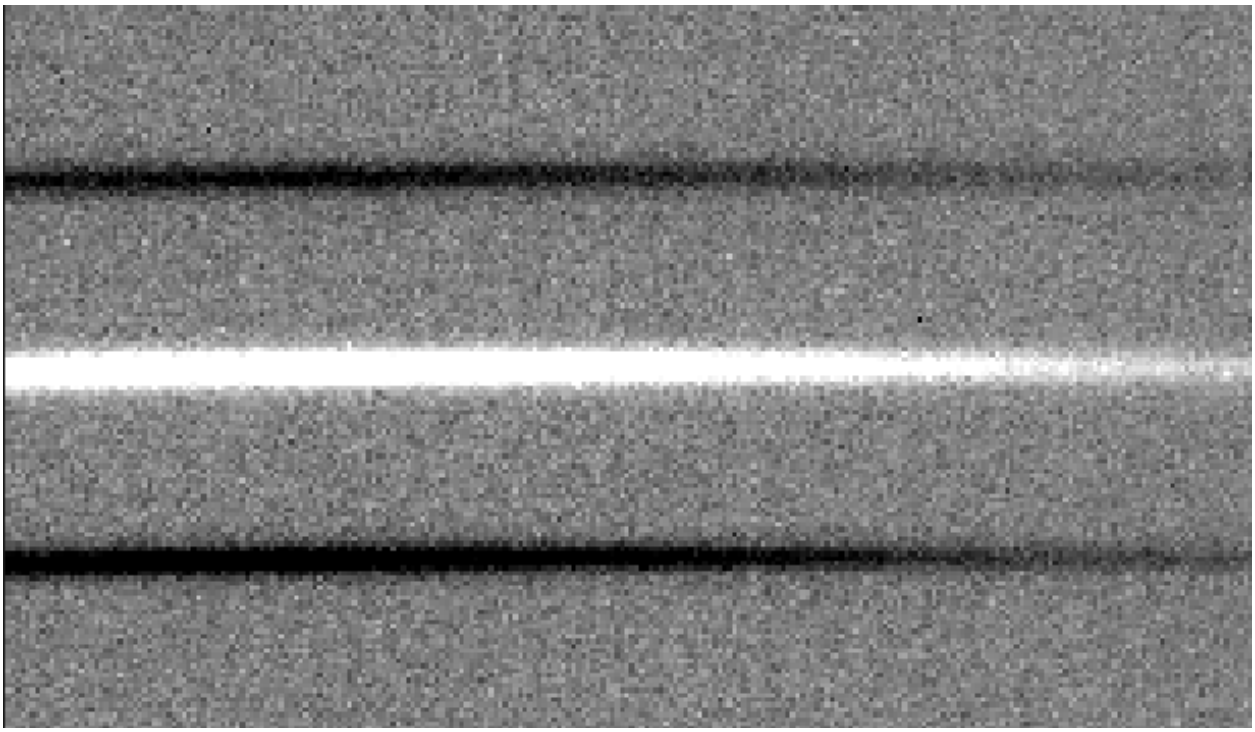


Fig. 13: NMC spectrum after stacking. All three spectra will be extracted.

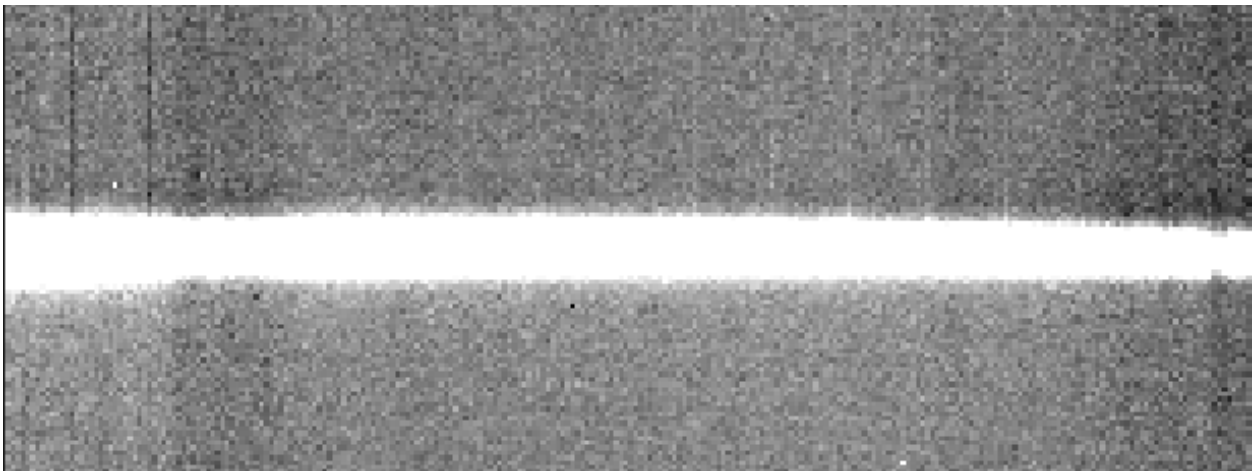


Fig. 14: NMC spectrum with wide chop after stacking.

as arcseconds up the slit, but a secondary WCS with key='A' identifies the RA, Dec, and wavelength of every pixel in the image. Either can be extracted and used for pixel identification with standard WCS manipulation packages, such as the astropy WCS package.

Coadding multiple observations

The final pipeline step for most grism observation modes is coaddition of multiple spectra of the same source with the same instrument configuration and observation mode. For spectroscopy, the FSpextool algorithm that combines the individual spectra scales them to a median value, by default, before combining them with a median statistic.

Reductions of flux standards have an alternate final product (see *Response spectra*, below). Slit-scan observations also produce an alternate final product instead of directly coadding spectra (see *Spectral cubes*, below).

Response spectra

The final product of pipeline processing of telluric standards is not a calibrated, combined spectrum, but rather an instrumental response spectrum that may be used to calibrate science target spectra. These response spectra are generated from individual observations of calibration sources by dividing the observed spectra by a model of the source multiplied by an atmospheric model. The resulting response curves are then combined with other response spectra from a flight series to generate a master response spectrum that is used in calibrating science spectra. See the flux calibration section, below, for more information.

Spectral cubes

For slit-scan observations, the calibrated, rectified images produced at the flux calibration step are resampled together into a spatial/spectral cube.

Since the pipeline rectifies all images onto the same wavelength grid, each column in the image corresponds to the same wavelength in all rectified images from the same grism. The pipeline uses the WCS in the headers to assign a spatial position to each pixel in each input image, then steps through the wavelength values, resampling the spatial values into a common grid.

The resampling algorithm proceeds as follows. At each wavelength value, the algorithm loops over the output spatial grid, finding values within a local fitting window. Outlier flux values (usually 10:math:`sigma` away from the mean value) are rejected, and the remaining values are fit with a low-order polynomial surface fit. These fits are weighted by the error on the flux, as propagated by the pipeline, and by a Gaussian function of the distance from the data point to the grid location. The output flux at each pixel is the value of the surface polynomial, evaluated at the grid location. The associated error value is the error on the fit. Grid locations for which there was insufficient input data are set to NaN. An exposure map cube indicating the number of observations input at each pixel is also generated and attached to the output FITS file.

5 Uncertainties

The pipeline calculates the expected uncertainties for raw FORCAST data as a variance image associated with the input data. FORCAST raw data is recorded in units of ADU per coadded frame. The variance associated with the *(i,j)*th pixel in this raw data is calculated as:

$$V_{ij} = \frac{N_{ij}\beta_g}{FR \cdot t \cdot g} + \frac{RN^2}{FR \cdot t \cdot g^2}$$

where N is the raw ADU per frame in each pixel, β_g is the excess noise factor, FR is the frame rate, t is the integration time, g is the gain, and RN is the read noise in electrons. The first term corresponds to the Poisson noise, and the

second to the read noise. Since FORCAST data are expected to be background-limited, the Poisson noise term should dominate the read noise term.

For all image processing steps, the pipeline propagates this calculated variance image alongside the flux in the standard manner. The variance image is written to disk as an extra plane in all FITS images produced at intermediate steps.

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:

$$V_i = \sum_j \frac{1}{P_{ij}^2 \cdot V_{ij}}$$

where P_{ij} is the spatial profile, V_{ij} is the variance at each pixel, and the sum is over all pixels j in the extraction aperture. This equation comes from the Spextool paper, describing optimal extraction.

For the 1D extracted spectra written to disk, the uncertainty is saved as a standard deviation (the square root of the propagated variance), in an extra dimension in the image file.

6 Other Resources

For more information on how to run the FSpextool interactive tools (*xspextool, ximgtool*, *xvspec*, *xwavecal2d, xcombspec*, *xtellcor*, and *xcleanspec*), see the help files distributed with the FSpextool code, under *fspextool/Helpfiles*.

For more information about the Redux, DRIP, and FSpextool software architecture, see the Redux Developer's Manual, located in *redux/helpfiles*.

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

7 Imaging Flux Calibration

The reduction process, up through image coaddition, generates Level 2 images with data values in units of mega-electrons per second (Me/s). After Level 2 imaging products are generated, the pipeline derives the flux calibration factors (in units of Me/s/Jy) and applies them to each image. The calibration factors are derived for each FORCAST filter configuration (filter and dichroic) from observations of calibrator stars.

After the calibration factors have been derived, they are written to the headers of the Level 2 telluric-corrected and coadded files in the FITS keyword CALFCTR. The coadded file is then divided by this factor to produce the Level 3 calibrated data file, with flux in units of Jy/pixel.

7.1 Reduction steps

The calibration is carried out in several steps. The first step consists of measuring the photometry of all the standard stars for a specific mission or flight series, after the images have been corrected for the atmospheric transmission relative to that for a reference altitude and zenith angle[2]. The pipeline performs aperture photometry on the reduced Level 2 images of the standard stars after the registration stage using a photometric aperture radius of 12 pixels (about 9.2" for FORCAST). The telluric-corrected photometry of the standard star is related to the measured photometry of the star via

$$N_e^{std,corr} = N_e^{std} \frac{R_\lambda^{ref}}{R_\lambda^{std}}$$

where the ratio $R_\lambda^{ref}/R_\lambda^{std}$ accounts for differences in system response (atmospheric transmission) between the actual observations and those for the reference altitude of 41000 feet and a telescope elevation of 45°. Similarly, for the science target, we have

$$N_e^{obj,corr} = N_e^{obj} \frac{R_\lambda^{ref}}{R_\lambda^{obj}}$$

Calibration factors (in Me/s/Jy) for each filter are then derived from the measured photometry (in Me/s) and the known fluxes of the standards (in Jy) in each filter. These predicted fluxes were computed by multiplying a model stellar spectrum by the overall filter + instrument + telescope + atmosphere (at the reference altitude and zenith angle) response curve and integrating over the filter passband to compute the mean flux in the band. The adopted filter throughput curves are those provided by the vendor or measured by the FORCAST team, modified to remove regions (around 6-7 microns and 15 microns) where the values were contaminated by noise. The instrument throughput is calculated by multiplying the transmission curves of the entrance window, dichroic, internal blockers, and mirrors, and the detector quantum efficiency. The telescope throughput value is assumed to be constant (85%) across the entire FORCAST wavelength range.

For most of the standard stars, the adopted stellar models were obtained from the *Herschel* calibration group and consist of high-resolution theoretical spectra, generated from the MARCS models (Gustafsson et al. 1975, Plez et al. 1992), scaled to match absolutely calibrated observational fluxes (Dehaes et al. 2011). For β UMi, the model was scaled by a factor of 1.18 in agreement with the results of the *Herschel* calibration group (J. Blommaert, private communication; the newer version of the model from the *Herschel* group has incorporated this factor).

The calibration factor, *C*, is computed from

$$C = \frac{N_e^{std,corr}}{F_\nu^{nom,std}(\lambda_{ref})} = \frac{N_e^{std,corr}}{\langle F_\nu^{std} \rangle} \frac{\lambda_{piv}^2}{\langle \lambda \rangle \lambda_{ref}}$$

with an uncertainty given by

$$\left(\frac{\sigma_C}{C}\right)^2 = \left(\frac{\sigma_{N_e^{std}}}{N_e^{std}}\right)^2 + \left(\frac{\sigma_{\langle F_\nu^{std} \rangle}}{\langle F_\nu^{std} \rangle}\right)^2.$$

Here, λ_{piv} is the pivot wavelength of the filter, and $\langle \lambda \rangle$ is the mean wavelength of the filter. The calibration factor refers to a nominal flat spectrum source at the reference wavelength λ_{ref}.

The calibration factors derived from each standard for each filter are then averaged. The pipeline inserts this value and its associated uncertainty into the headers of the Level 2 data files for the flux standards, and uses the value to produce calibrated flux standards. The final step involves examining the calibration values and ensuring that the values are consistent. Outlier values may come from bad observations of a standard star; these values are removed to produce a

[2] The atmospheric transmission in each filter has been computed using the ATRAN code (Lord 1992) for a range of observatory altitudes (corresponding to a range of overhead precipitable water vapor values) and telescope elevations. The ratio of the transmission at each altitude and zenith angle relative to that at the reference altitude (41000 feet) and zenith angle (45 degrees) has been calculated for each filter and fit with a low order polynomial. The ratio appropriate for the altitude and zenith angle of each observation is calculated and applied to each image.

robust average of the calibration factor across the flight series. The resulting average values are then used to calibrate the observations of the science targets.

Using the telluric-corrected photometry of the standard, $N_e^{std,corr}$ (in Me/s), and the predicted mean fluxes of the standards in each filter, $\langle F_\nu^{std}$ (in Jy), the flux of a target object is given by

$$F_\nu^{nom,obj}(\lambda_{ref}) = \frac{N_e^{obj,corr}}{C}$$

where $N_e^{obj,corr}$ is the telluric-corrected count rate in Me/s detected from the source, C is the calibration factor (Me/s/Jy), and $F_\nu^{nom,obj}(\lambda_{ref})$ is the flux in Jy of a nominal, flat spectrum source (for which $F_\nu \sim \nu^{-1}$) at a reference wavelength λ_{ref}.

The values of C, σ_C, and λ_{ref} are written into the headers of the calibrated (PROCSTAT=LEVEL_3) data as the keywords CALFCTR, ERRCALF, and LAMREF, respectively. The reference wavelength λ_{ref} for these observations was taken to be the mean wavelengths of the filters, $\langle\lambda\rangle$. Note that σ_C includes uncertainty on the stellar models and the values of $\langle F_\nu^{std}\rangle$, assumed to be on the order of 5-10% (Dehaes et al. 2011). Based on the variations seen in the calibration factors across multiple flights we estimate the overall statistical uncertainty in our flux calibrations is about 6% (see Herter et al. 2013).

7.2 Color corrections

An observer often wishes to determine the true flux of an object at the reference wavelength, $F_\nu^{obj}(\lambda_{ref})$, rather than the flux of an equivalent nominal, flat spectrum source. To do this, we define a color correction *K* such that

$$K = \frac{F_\nu^{nom,obj}(\lambda_{ref})}{F_\nu^{obj}(\lambda_{ref})}$$

where $F_\nu^{nom,obj}(\lambda_{ref})$ is the flux density obtained by measurement on a data product. Divide the measured values by *K* to obtain the "true" flux density. In terms of the wavelengths defined above,

$$K = \frac{\langle\lambda\rangle\lambda_{ref}}{\lambda_{piv}^2}\frac{\langle F_\nu^{obj}\rangle}{F_\nu^{obj}(\lambda_{ref})}.$$

For most filters and spectral shapes, the color corrections are small (<10%). Tables listing *K* values and filter wavelengths are available from the SOFIA website.

8 Spectrophotometric Flux Calibration

The common approach to characterizing atmospheric transmission for ground-based infrared spectroscopy is to obtain, for every science target, similar observations of a spectroscopic standard source with as close a match as possible in both airmass and time. Such an approach is not practical for airborne observations, as it imposes too heavy a burden on flight planning and lowers the efficiency of science observations. Therefore, we employ a calibration plan that incorporates a few observations of a calibration star per flight and a model of the atmospheric absorption for the approximate altitude and airmass (and precipitable water vapor, if known) at which the science objects were observed.

Instrumental response curves are generated from the extracted spectra of calibrator targets. For the G063 and G111 grisms, the calibrator targets comprise the set of standard stars and the associated stellar models provided by the *Herschel* Calibration program and used for the FORCAST photometric calibration. For the G227 and G329 grisms, the calibrator targets consist of bright asteroids. Blackbodies are fit to the calibrated broadband photometric observations of the asteroids and these serve as models of the intrinsic asteroid spectra. In either case, the extracted spectra are corrected for telluric absorption using the ATRAN models corresponding to the altitude and zenith angle of the calibrator observations, smoothed to the nominal resolution for the grism/slit combination, and sampled at the observed spectral binning. The telluric-corrected spectra are then divided by the appropriate models to generate response curves

(with units of Me/s/Jy at each wavelength) for the various grism+slit+channel combinations. The response curves derived from the various calibrators for each instrumental combination are then combined to generate a set of master instrumental response curves. The statistical uncertainties on these response curves are on the order of 5-10%.

Spectra of science targets are first divided by the appropriate instrumental response curve, a process that yields spectra in physical units of Jy at each wavelength.

Telluric correction of FORCAST grism data for a science target is currently carried out in a multi-step process:

1. Telluric absorption models have been computed, using ATRAN, for the entire set of FORCAST grism passbands for every 1000 feet of altitude between 35K and 45K feet, for every 5 degrees of zenith angle between 30 and 70 degrees, and for a set of precipitable water vapor (PWV) values between 1 and 50 microns. These values span the allowed ranges of zenith angle, typical range of observing altitudes, and the expected range of PWV values for SOFIA observations. The spectra have been smoothed to the nominal resolution for the grism and slit combination and are resampled to the observed spectral binning.
2. If the spectrum of the science target has a signal-to-noise ratio greater than 10, the best estimate of the telluric absorption spectrum is derived in the following manner: under the assumption that the intrinsic low-resolution MIR spectrum of most targets can be well-represented by a smooth, low-order polynomial, the telluric spectrum that minimizes χ^2 defined as

$$\chi_j^2 = \sum_i^n \Big(F_i^{obs} - P_i T_i \big(PWV_j\big) \Big)^2 / \sigma_i^2$$

   is determined. Here F_i^{obs} is the response-corrected spectrum at each of the *n* wavelength points *i*, σ_i is the uncertainty at each point, P_i is the polynomial at each point, and T_i is the telluric spectrum corresponding to the precipitable water vapor value PWV_j. The telluric spectra used in the calculations are chosen from the pre-computed library generated with ATRAN. Only the subset of ATRAN model spectra corresponding, as close as possible, to the observing altitude and zenith angle, are considered in the calculation. The free parameters determined in this step are the coefficients of the polynomial and the PWV value, which then yields the best telluric correction spectrum. The uncertainty on the PWV value is estimated to be about 1-2 microns.
3. If the spectrum of the science target has a S/N less than 10, the closest telluric spectrum (in terms of altitude and airmass of the target observations) with the default PWV value from the ATRAN model is selected from the pre-computed library.
4. In order to account for any wavelength shifts between the models and the observations, an optimal shift is estimated from the peak of the cross-correlation of the response-corrected spectrum and the telluric correction curves.
5. The observed spectrum is then divided by the smoothed, shifted, and re-sampled telluric model. This then yields a telluric-corrected and flux calibrated spectrum.

Analysis of the calibrated spectra of standard stars observed during Cycle 2 indicates that the average RMS deviation over the G063, G227, and G329 grism passbands between the calibrated spectra and the models is on the order of about 5%. For the G111 grism, the average RMS deviation is found to be on the order of about 10%; the larger deviation for this grism is due primarily to the highly variable ozone feature at 9.6 microns, which the ATRAN models are not able to reproduce accurately. The Level 3 data product for any grism includes the calibrated spectrum and an error spectrum that incorporates these RMS values. The adopted telluric absorption model and the instrumental response functions are also provided.

As for any slit spectrograph, highly accurate absolute flux levels from FORCAST grism observations (for absolute spectrophotometry, for example) require additional photometric observations to correct the calibrated spectra for slit losses that can be variable (due to varying image quality) between the spectroscopic observations of the science target and the calibration standard.

Part V

Data products

9 Filenames

Output files from Redux are named according to the convention:

FILENAME = F[flight]_FO_IMA|GRI_AOR-ID_SPECTEL1|SPECTEL2_Type_FN1[-FN2],

where flight is the SOFIA flight number, FO is the instrument identifier, IMA or GRI specifies that it is an imaging or grism file, AOR-ID is the 8 digit AOR identifier for the observation, SPECTEL1|SPECTEL2 is the keyword specifying the filter or grism used, Type is three letters identifying the product type (listed in Table 1 and Table 2:, below), FN1 is the file number corresponding to the input file. FN1-FN2 is used if there are multiple input files for a single output file, where FN1 is the file number of the first input file and FN2 is the file number of the last input file.

10 Pipeline Products

The following tables list all intermediate products generated by Redux for imaging and grism modes, in the order in which they are produced. By default, for imaging, the undistorted, merged, telluric_corrected, coadded, and calibrated products are saved; for grism, the stacked, mrgspec, rectimg, calspec, calrectimg, and combspec products are saved.

Table 1: Intermediate data products for imaging reductions

Step	Description	PRODTYPE	PROC-STAT	Code	Saved
Clean Images	Bad pixels cleaned	cleaned	LEVEL_2	CLN	N
Correct Droop	Corrected for droop effect	drooped	LEVEL_2	DRP	N
Correct Nonlinearity	Corrected for detector nonlinearity	linearized	LEVEL_2	LNZ	N
Stack Chops/Nods	Chop/Nod stacked	stacked	LEVEL_2	STK	N
Undistort	Corrected for optical distortion	undistorted	LEVEL_2	UND	Y
Merge	Chop/nod images merged into single source	merged	LEVEL_2	MRG	Y
Register	Multiple observations registered to a reference	registered	LEVEL_2	REG	N
Telluric Correct	Corrected for atmospheric transmission	telluric_corrected	LEVEL_2	TEL	Y
Coadd	Multiple observations combined	coadded	LEVEL_2	COA	Y
Flux Calibrate	Flux calibration factor applied to image	calibrated	LEVEL_3	CAL	Y
Mosaic	Calibrated images mosaicked together	mosaic	LEVEL_4	MOS	Y

Table 2: Intermediate data products for spectroscopy reduction

Step	Description	PRODTYPE	PROC-STAT	Code	Saved
Clean Images	Bad pixels cleaned	cleaned	LEVEL_2	CLN	N
Correct Droop	Corrected for droop effect	drooped	LEVEL_2	DRP	N
Correct Nonlinearity	Corrected for detector nonlinearity	linearized	LEVEL_2	LNZ	N
Stack Chops/Nods	Chop/Nod stacked	stacked	LEVEL_2	STK	Y
Stack Chops/Nods	Common dither positions combined	stackeddithers	LEVEL_2	SKD	Y
Extract Spectra	Rectified image, produced during extraction	rectified	LEVEL_2	RIM	Y
Extract Spectra	Raw extracted spectra	spec	LEVEL_2	SPC	N
Merge Spectra	Merged spectra	mrgspec	LEVEL_2	MRG	Y
Flux Calibrate	Flux calibrated spectra and rectified image	calspec, calrectimg	LEVEL_3	CAL, CRM	Y
Combine Spectra	Combined spectra	combspec	LEVEL_3	CMB	Y
Make Response	Instrumental response spectra	rspspec	LEVEL_2	RSP	Y
Make Spectral Cube	Spatial/spectral map for slit-scan mode	speccube	LEVEL_4	SCB	Y

11 Data Format

All files produced by the pipeline are FITS single-extension image files (except for the 'speccube' product – see below).

All imaging products are 3D arrays of data, where the first plane is the image and the second plane is the variance associated with each pixel in the image. Take the square root of the variance plane to get the uncertainty estimate associated with each pixel in the image. The third plane in the merged, telluric_corrected, coadded, and calibrated imaging products is an exposure map, indicating the on-source integration time in seconds at each pixel. The maximum value in this map is recorded in the header as EXPTIME.

The stacked and rectified grism products, like the imaging products, are 3D arrays of data, where the first plane is the image and the second is the variance. The third plane in the rectified image is a bad pixel mask. The rectified image also contains the wavelength calibration, encoded in the WCS in the header of the FITS file. The merged grism products are one-dimensional spectra, stored in three 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 the estimated fractional atmospheric transmission spectrum. If there were multiple orders in the spectrum (e.g. the G2xG1 cross-dispersed mode), then the spectrum for each order is stacked into a different plane. The length of the row varies depending on the data, but is typically around 240-250 pixels. For the calibrated and combined grism products, one additional row is added, for reference: the fifth row is the instrumental response curve used in flux calibration, in Me/s/Jy. For the response products, generated from telluric standard observations, the first row is the wavelengths, the second is the response spectrum (Me/s/Jy), the third is the error on the response, the fourth is the atmospheric transmission spectrum, and the fifth is the standard model used to derive the response.

The spectral cube product is also a 3D array, but each plane in the cube represents the spatial information at a wavelength slice. This flux cube is stored as an image in the primary FITS extension (EXTNAME=FLUX). The associated variance is stored in an additional extension with EXTNAME=VARAINCE, and an exposure map is stored in a third extension, with EXTNAME=EXPOSURE. The variance and exposure extensions are also 3D cubes. The spatial and spectral coordinates are stored in the WCS keywords in each extension.

The final uncertainties in the calibrated image and/or spectrum contain only the estimated statistical uncertainties

due to the noise in the image or the extracted spectrum. The systematic uncertainties due to the calibration process are recorded in header keywords. For imaging data, the error on the calibration factor is recorded in the keyword ERRCALF. For grism data, the estimated overall fractional error on the flux is recorded in the keyword CALERR.[3]

Part VI

Grouping LEVEL_1 data for processing

In order for a group of imaging data to be reduced together usefully, all images must have the same target object and be taken in the same chop/nod mode. They must also have the same detector, filter, and dichroic setting. In order to be combined together, they must also be taken on the same mission. Optionally, it may also be useful to separate out data files taken from different observation plans.

For spectroscopy, all the same rules hold, with the replacement of grism element for filter, and with the additional requirement that the same slit be used for all data files.

These requirements translate into a set of FITS header keywords that must match in order for a set of data to be grouped together. These keyword requirements are summarized in the tables below.

Table 3: Grouping Criteria: Imaging

Keyword	Data Type	Match Criterion
OBSTYPE	STR	Exact
OBJECT	STR	Exact
INSTCFG	STR	Exact
DETCHAN	STR	Exact
SPECTEL1 / SPECTEL2*	STR	Exact
BORESITE	STR	Exact
DICHROIC	STR	Exact
MISSN-ID	STR	Exact
PLANID	STR	Exact
AOR_ID (optional)	STR	Exact

Table 4: Grouping Criteria: Spectroscopy

Keyword	Data Type	Match Criterion
OBSTYPE	STR	Exact
OBJECT	STR	Exact
INSTCFG	STR	Exact
DETCHAN	STR	Exact
SPECTEL1 / SPECTEL2*	STR	Exact
BORESITE	STR	Exact
DICHROIC	STR	Exact
MISSN-ID	STR	Exact
PLANID	STR	Exact
SLIT**	STR	Exact
AOR_ID (optional)	STR	Exact

* SPECTEL1 is used if the detector is the SWC (DETCHAN=SW); SPECTEL2 is used for LWC (DETCHAN=LW)

[3] Earlier versions of this pipeline (before 1.2.0) may have stored the systematic calibration error in the error spectrum or variance image, added in quadrature with the statistical error. Check PIPEVERS and compare the error estimates for the calibrated products to earlier products to ensure correct interpretation of the error estimates.

** If SLIT is in use (value != “NONE” or “UNKNOWN”), **always** include group in the grism plan, regardless of INSTCFG. This ensures that slit images get reduced with the spectroscopic data and placed in the same preview.

Part VII

Configuration and execution

12 Installation

Running FORCAST Redux requires IDL 8.1 or later, as well as the latest version of the IDL Astronomy User’s Library, the Coyote graphics library, the DRIP package, the FSpextool package, the PipeCal package, and the Redux code. DRIP, FSpextool, PipeCal and Redux are under SOFIA DPS revision control and can be obtained directly from git repositories there. The IDL Astronomy User’s Library (astrolib) is publicly available, and can be downloaded from the website at the following URL:

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

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

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

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

When you have the gzipped tar file of the Redux, DRIP, PipeCal, and FSpextool codes, unpack them, as, for example:

```
tar xvzf redux.tar.gz
tar xvzf drip.tar.gz
tar xvzf pipecal.tar.gz
tar xvzf fspextool.tar.gz
```

This will create directories called redux, drip, pipecal, and fspextool. Each of these package directories should be added to the IDL_PATH as well.

13 Configuration

For DRIP algorithms, default options are specified in the dripconf.txt configuration file, located in the drip package directory, which is read when the pipeline is initiated. This file contains a list of keywords and their values that are used in the data reduction process. See Appendix B for a sample of this configuration file.

For FSpextool algorithms, default options are specified in a FORCAST.dat configuration file, located in the fspextool package directory. This file also 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 B for a sample of this configuration file as well.

Additional default parameter values may be defined by the Redux interface object. These values, as well as the values defined in the DRIP and FSpextool configuration files, may be overridden manually for each reduction step, while running in interactive mode. They may also be overridden by an input parameter file, in JSON format, in either interactive or automatic mode. See Appendix B for an example of an input Redux parameter file; if specified, values in this file override those derived from all other configuration files.

14 Input data

Redux takes as input raw FORCAST FITS data files, which contain image cubes composed of 256x256 pixel image arrays. The number of frames per raw data cube depends on the chop/nod mode used to acquire the data (see Table 5). 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 FITS headers in order to perform a successful grouping, reduction, and ingestion into the SOFIA archive. See Appendix A for a description of these keywords.

Table 5: Contents of FORCAST raw data files by observing mode

Chop/Nod Mode	Number of frames	Comments
C2N, NMC	4	Two-Position Chop with Nod Matched in throw and parallel to the chop direction 2 chop positions in each of 2 nod positions
C2N, NPC	4	Two-Position Chop with Nod perpendicular to the chop direction 2 chop positions in each of 2 nod positions
C2NC2	4	Extreme asymmetric chop and telescope move to blank sky: two chop positions per sky position. Typically 5 input files corresponding to ABAABAAB pattern
N	2	Two-position Nod only, may be used for grism spectroscopy
SLITSCAN	4	Spectral map of an extended source, most likely using C2NC2 but could use C2N

It is assumed that the input data have been successfully grouped before beginning reduction: Redux considers all input files in a reduction to be science files that are part of a single homogeneous reduction group, to be reduced together with the same parameters. As such, when the pipeline reads a raw FORCAST data file, it uses the first input file to identify the observing mode used. Given this information, it identifies a set of auxiliary and calibration data files to be used in the reduction (Table 6). The default files to be used are defined in a lookup table that reads the DATE-OBS keyword from the raw file, and then chooses the appropriate calibrations for that date.

Table 6: Auxiliary files

Auxiliary data file	Data type	Comments
Configuration file (e.g. dripconf.txt)	ASCII	Contains initial configuration of pipeline parameters and non-linearity coefficients
Keyword definition file (e.g. OC3_keywords.txt)	ASCII	Contains the definition of required keywords, with allowed value ranges
Bad pixel mask (e.g. swc_badpix.fits)	FITS	Single 2D image containing locations of bad pixels in short wave or long wave camera
pinhole_locs.txt	ASCII	Pinhole locations file for distortion correction (imaging only)
Spectral order definition file (e.g. G1_narrow_flat.fits)	FITS	Image file containing header keywords that define the edges of all orders in a 2D spectral image
Wavelength calibration map (e.g. G1_wavecal.fits)	FITS	Two frame image associating a wavelength value and a spatial distance across the slit with each pixel (grism only)
Atmospheric transmission curve (e.g. atran_41K_45deg_4-50mum.sav)	IDL SAV	An IDL array with wavelength and transmission values for a particular altitude and zenith angle (grism only)
Instrumental response curve (e.g. G063_LS24_Cy2_response.fits)	ASCII	Table containing the response (Me/s/Jy) at each wavelength for a particular grism/slit mode
Slit function image (e.g. g063_ls24_slitfn_x2_pix.fits)	FITS	Image file containing the slit response, in rectified spectral co-ordinates.

15 Automatic mode execution

To run the pipeline from the IDL command line as a fully automatic black box pipeline, we 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 pipeline 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 one input parameter on the command line. The param_file parameter allows the user to provide a parameter file that specifies any desired reduction parameters:

```
IDL> redux_pipe, 'infiles.txt', param_file='param.json'
```

where param.json is a file that lists parameter keywords and values in JSON format for any step in the pipeline reduction (see Appendix B). The default value will be used for any parameter not present in the file. Parameter files can be generated interactively with the GUI (see below), then edited, saved, and fed to the automatic pipeline for batch reduction of a large number of files.

16 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.

16.1 Basic workflow

To start an interactive reduction, open a set of FORCAST files, using the File menu (**File->Open New Reduction**). This will bring up a file dialog window (Fig. 15). All files selected will be reduced together as a single reduction set.

Redux will decide the appropriate reduction steps from the input files, and load them into the GUI. The steps for imaging inputs will differ from the steps for grism inputs (Fig. 16, Fig. 17), following the flowchart in Fig. 6.

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. 18). Within the parameter editor, all values may be edited; clicking **Done** will save the edited values and close the window. Clicking **Reset** will restore any edited values to their defaults; clicking **Cancel** will discard all changes to the parameters and close the editor window. The current set of parameters can be displayed, saved to a file, or reset all at once using the **Parameters** menu. A previously saved set of parameters can also be restored for use with the current reduction (**Parameters->Load Parameters**).

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**.

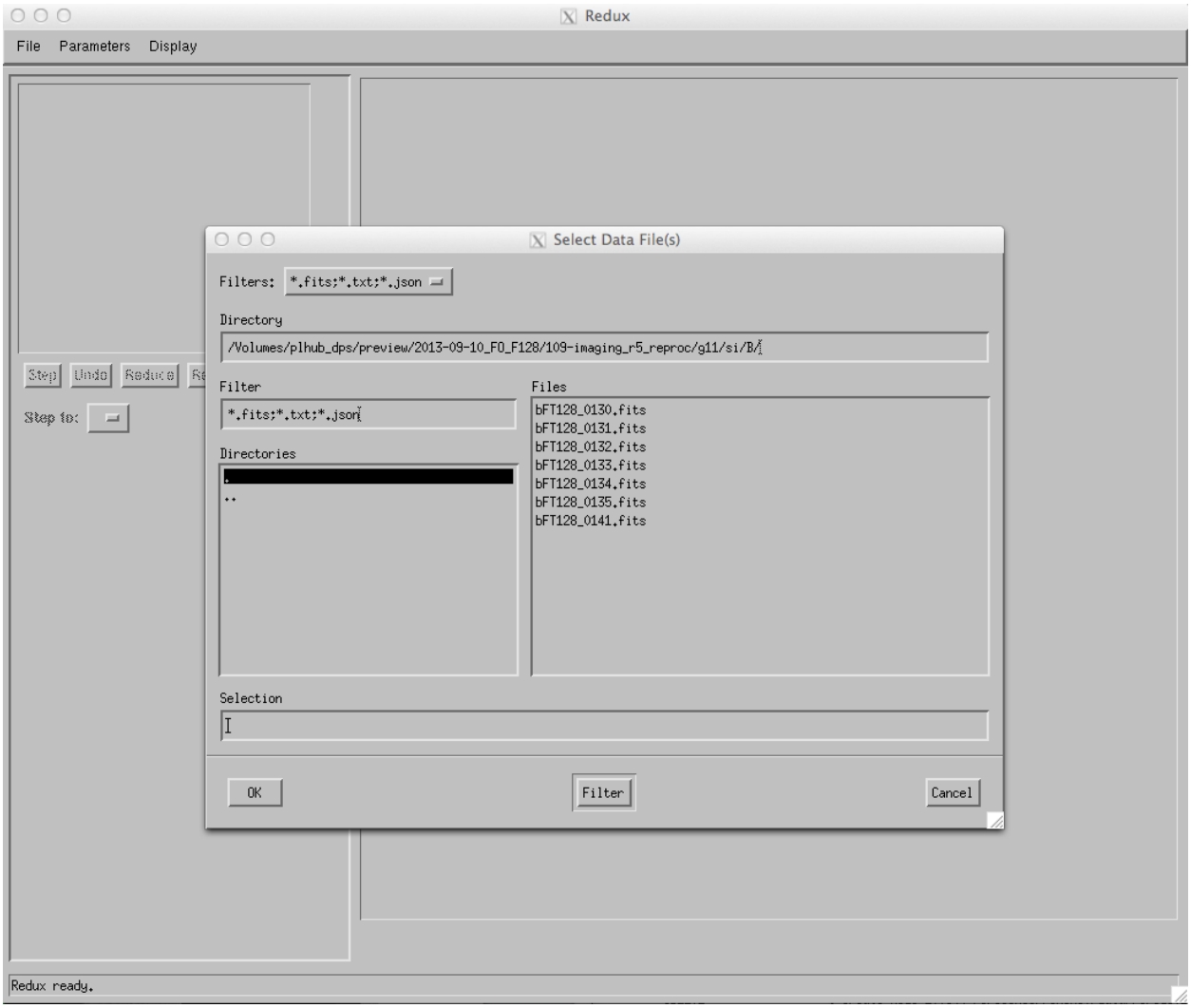


Fig. 15: Open New Reduction

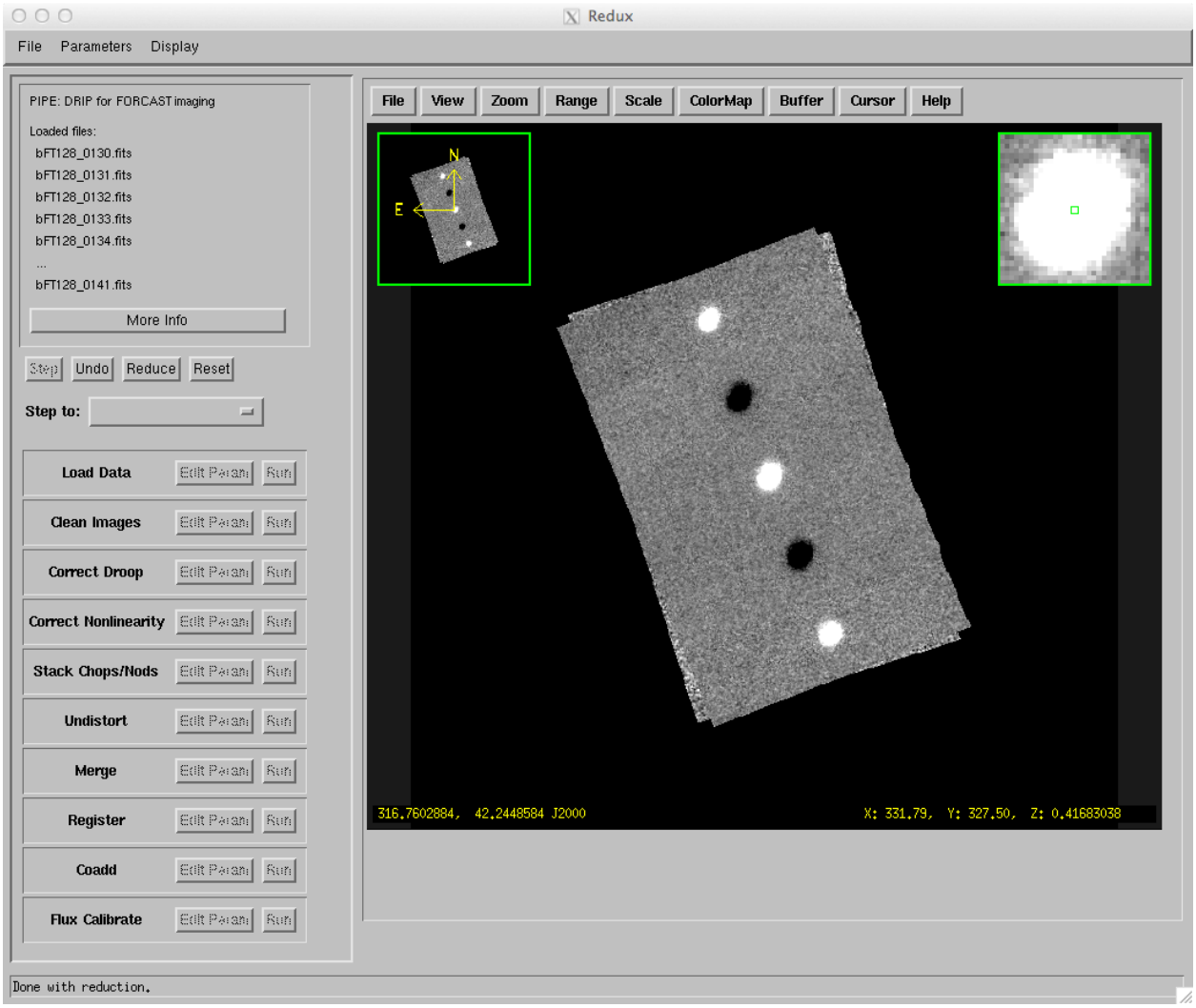


Fig. 16: Imaging reduction

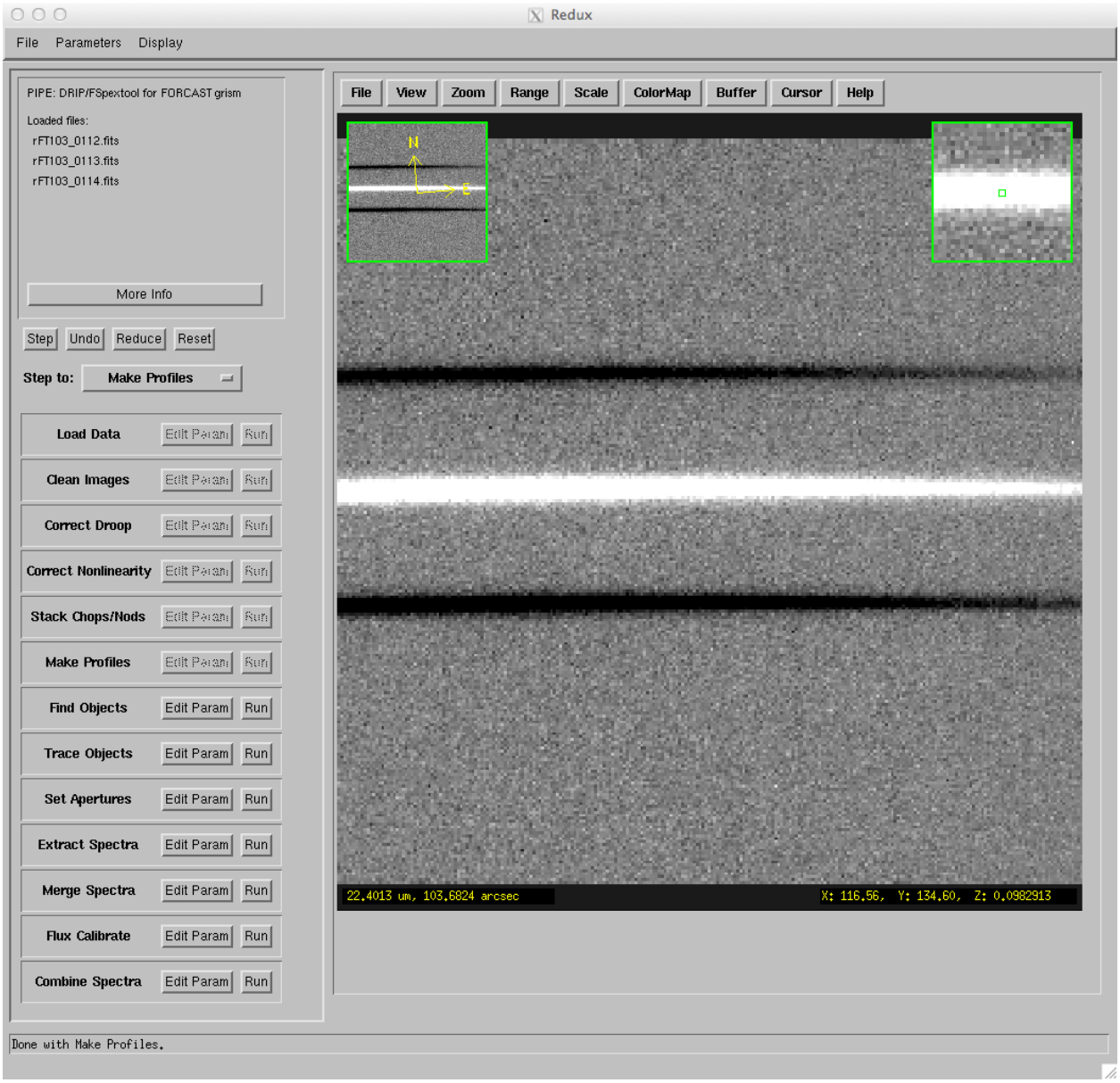


Fig. 17: Grism reduction

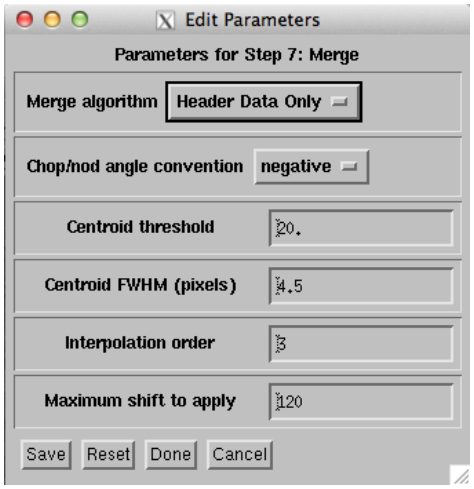
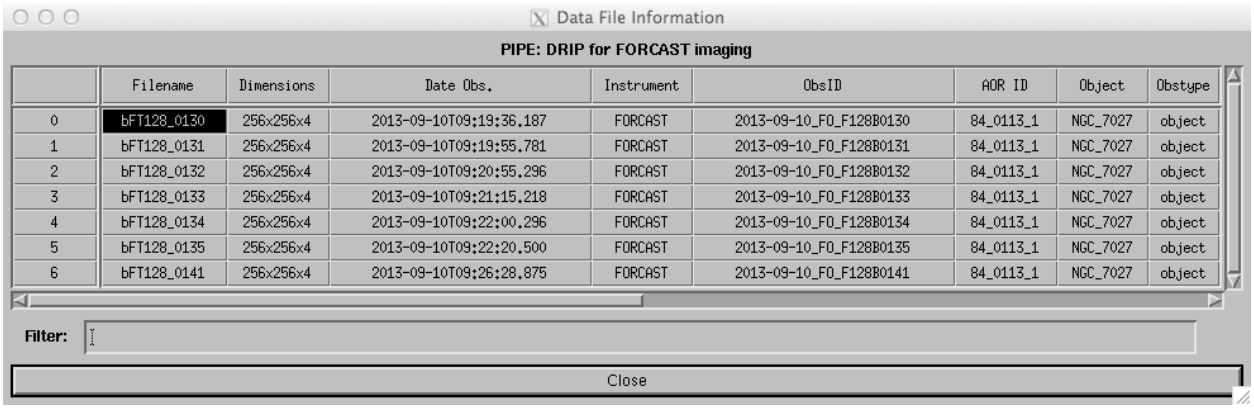


Fig. 18: Sample parameter editor (for imaging Merge step)

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



	Filename	Dimensions	Date Obs.	Instrument	ObsID	AOR ID	Object	Obstype
0	bFT128_0130	256x256x4	2013-09-10T09:19:36.187	FORCAST	2013-09-10_FO_F128B0130	84_0113_1	NGC_7027	object
1	bFT128_0131	256x256x4	2013-09-10T09:19:55.781	FORCAST	2013-09-10_FO_F128B0131	84_0113_1	NGC_7027	object
2	bFT128_0132	256x256x4	2013-09-10T09:20:55.296	FORCAST	2013-09-10_FO_F128B0132	84_0113_1	NGC_7027	object
3	bFT128_0133	256x256x4	2013-09-10T09:21:15.218	FORCAST	2013-09-10_FO_F128B0133	84_0113_1	NGC_7027	object
4	bFT128_0134	256x256x4	2013-09-10T09:22:00.296	FORCAST	2013-09-10_FO_F128B0134	84_0113_1	NGC_7027	object
5	bFT128_0135	256x256x4	2013-09-10T09:22:20.500	FORCAST	2013-09-10_FO_F128B0135	84_0113_1	NGC_7027	object
6	bFT128_0141	256x256x4	2013-09-10T09:26:28.875	FORCAST	2013-09-10_FO_F128B0141	84_0113_1	NGC_7027	object

Fig. 19: File information table

16.2 Display features

Redux displays images using ximgtool, a full-featured display tool distributed with FSpextool. For more information, see the ximgtool help file, available from Redux via the **Help** button just above the display. See Table 7 for a quick listing of the most useful ximgtool features.

Table 7: Useful ximgtool features

Feature	Menu button	Keyboard shortcut
Load new file	**File->Load FITS**	–
Load file into new frame	**File->New Frame**	–
View FITS header	**File->View Header**	–
Zoom	**Zoom->Zoom In**, **Zoom Out**, **Zoom To Fit**	Press **z** to enter zoom mode, then **i** to zoom in, **o** to zoom out, or **t** to zoom to fit
Color stretch	**Cursor->Stretch**	Press **s** to enter stretch mode, click and drag to change brightness and contrast
Set display range	**Cursor->Range**	Press **r** to enter range mode, click and drag to select the box that sets the display range
Display distance	–	Press **d** to enter distance mode, then click and drag to identify start and end points
Line cut	–	Press **l** to enter line cut mode, then click and drag to identify start and end points
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 frame, **p** to move to the previous frame.

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

16.3 Imaging Reduction

FORCAST imaging reduction with Redux is straightforward. Each processing step in the GUI corresponds to a single step in the flowchart of Figure 6 (with the exception of the Jailbar Correct step: this is applied as part of the Stack Chop/Nods processing step in the Redux GUI).

Some intermediate imaging products may be loaded into Redux, and the reduction picked up where it left off. The products that may be directly loaded are those with FITS keyword PRODTYPE set to undistorted, merged, telluric_corrected, or coadded. There are also two types of raw data that have slightly different reduction recipes. Slit images, for which a slit was present in the field of view, are processed only through the Merge step. Acquisition images, for which the boresight was the long-slit or short-slit position, but the slit was not present, are processed as normal, except that the coadd is skipped by default.

Useful Parameters

Some key parameters to note are listed below.

- **Load Data**
  - *Check headers*: By default, Redux will abort the 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 the files produced after each reduction step.
  - *Apply flux calibration factor*: Uncheck this box to skip applying flux calibration factors to the headers of the merged, registered, and coadded data products, and to skip generating the calibrated data product.
  - *Output path*: Set this parameter to a valid directory path name, in order to save the output data products to a specific location. The default output location is the current working directory.
  - *Excess noise factor (beta_g)*: Increase this value to increase the error estimate calculated by the pipeline. Currently used for testing only.
- **Clean Images**
  - *Automatically detect readout shift*: If selected, the pipeline will attempt to automatically detect and correct the readout shift described below.
  - *If not auto, correct readout shift in image number*: This option allows the user to fix an occasional issue where the data array is shifted to the right by 16 pixels. If multiple images are loaded, but only some are affected, the images to shift may be specified by their index in the list of input files, separated by semicolons. For example, to shift the first and third file, enter '1;3'. If all input files should be shifted, enter 'all'. Leave blank to leave all data unshifted (the default behavior).
  - *Bad pixel map*: If 'default', the default bad pixel mask on disk will be used. If blank, no bad pixel map will be used. Set to a valid FITS file path to override the default bad pixel mask with a new one.
  - *Jailbar cleaning method*: The default, and most reliable, method is 'Median.' 'FFT' may be used for testing. 'None' turns off jailbar cleaning. When 'FFT' is selected, the correction is applied in the Clean Images step; when 'Median' is selected, the correction is deferred until the Stack step.
- **Correct Droop**
  - *Droop fraction*: Lower value to decrease droop correction; set to zero to turn off droop correction altogether. Default value is currently 0.0035 for all data.
- **Stack Chops/Nods**
  - *Scale frames to common level*: If selected, images will be corrected for a multiplicative offset in the background level. This is not commonly used.
  - *Subtract residual background*: If selected, images will be corrected for an additive offset in the background level. This is the default, and recommended setting, but background differences may occasionally be overcorrected, in which case, this option should be deselected, or the background section modified.
  - *Combine common dither positions*: If selected, will stack images taken at the same dither position to produce a single file, before continuing the reduction. This may be used for grism reductions of faint sources, but is not commonly used for imaging reductions.
  - *TSA chop convention*: After SOFIA flight series OC1B, the sign convention of the chops changed from +1 to -1; modify this parameter to select a different value from the default convention appropriate for the date of observation. If set correctly, this parameter makes the central source positive; if incorrect, the central source appears negative. This parameter is not commonly modified.
- **Undistort**

  - *Border pixels to add*: This parameter sets the width of the padding around the undistorted image. A value of 200 is generally wide enough for any observation; it may be decreased for observations that are not shifted-and-added at the merge stage.

- **Merge**
  - *Merge algorithm*: The default for flux standard observations (OBSTYPE=STANDARD_FLUX) is to shift-and-add the data, using a centroiding algorithm. If the centroiding algorithm fails, header data will be used instead. The default for science data is not to shift-and-add ('No shift').
  - *Interpolation order*: Set to 0 to do integer pixel shifts (no interpolation), 1 for bilinear interpolation, or 3 for a cubic interpolation (default).
  - *Strip border from image*: Select to strip any remaining NaN border from the edge of the image. Deselect to continue to pad the image with a NaN border. Must be deselected to use the 'Header shifts' method for Register.
- **Register**
  - *Registration algorithm*: As for merging, the default for flux standards is to use a centroiding algorithm. All other data defaults to using WCS data for registration. Centroiding is primarily useful for bright, compact objects; cross-correlation may be useful for bright, diffuse fields. Registration via the 'Header shifts' method may be useful for older data, for which the relative WCS is not very accurate. The 'Interactive' registration algorithm allows the user to select a common point to (optionally) centroid on in each image. Zooming and scaling is not available while running the interactive registration: zoom in and set the image scale as needed before clicking 'Run' for this step.
  - *Interpolation order*: Set to 0 to do integer pixel shifts (no interpolation), 1 for bilinear interpolation, or 3 for a cubic interpolation (default).
  - *Override offsets*: If registration offsets are known a priori, they may be directly entered here. Enter offsets as comma-separated x,y values, with values for each file separated by semi-colons. For example, 0.0,0.0;-1.0,1.0 will leave the first file unshifted and shift the second file to the left 1 pixel and up 1 pixel.
- **Coadd**
  - *Skip coadd*: If selected, each input registered file will be saved as a separate file of type 'coadded' rather than combined together into a single output file.
  - *Combination statistic*: Median is the default; Robust Weighted Mean may also be useful. Be aware that different combination statistics have different methods of calculating output error values.
  - *Combination threshold*: The rejection threshold for robust combination methods, in units of sigma (standard deviation).
- **Flux Calibrate**
  - *Recalculate photometry*: If selected, and observation is a flux standard, photometric fits and aperture measurements on the brightest source will be recalculated, using the input parameters below. Recalculated values will overwrite those in the header of the coadded file.
  - *Source position*: Enter the approximate position (x,y) of the source to measure. If not specified, the SRCPOSX/SRCPOSY keywords in the FITS header will be used as the first estimate of the source position.
  - *Subimage size for profile fit*: Smaller subimages may sometimes be necessary for faint sources and/or variable background.
  - *Starting FWHM for profile fit*: Specify in pixels. This parameter should be modified only if the PSF of the source is significantly larger or smaller than usual.
  - *Profile type*: Moffat fits are the default, as they generally give a more reliable FWHM value. However, Gaussian fits may sometimes be more stable, and therefore preferable if the Moffat fit fails. Lorentzian fits are rarely used.

16.4 Grism Reduction

FORCAST grism reduction with Redux is slightly more complicated than for imaging. The GUI breaks down the spectral extraction algorithms into six separate reduction steps to give more control over the extraction process. These steps are:

- Make Profiles: Generate a smoothed model of the relative distribution of the flux across the slit (the spatial profile). After this step is run, a separate display window showing a plot of the spatial profile appears (see Fig. 20).
- Find Objects: Use the spatial profile to identify spectra to extract. By default, Redux attempts to automatically identify sources, but they can also be manually identified by entering a guess position to fit near, or a fixed position, in the Find Objects parameters.
- Trace Objects: Identify the location of the spectrum across the array, by either fitting the continuum or fixing the location to the aperture center.
- Set Apertures: Identify the data to extract from the spatial profile (Fig. 20). This is done automatically by default, but all aperture parameters can be overridden manually in the parameters for this step.
- Extract Spectra: Extract the one-dimensional spectrum from the identified apertures. By default, Redux will perform standard extraction observations that are marked as extended sources (SRCTYPE=EXTENDED_SOURCE) and will attempt optimal extraction for any other value. The method can be overridden in the parameters for this step.
- Merge Spectra: All apertures are merged into a single spectrum, and normalized according to the Chop/Nod mode.

Extracted spectra are displayed using xvspec (Fig. 21), another display tool packaged with FSpextool. Like ximgtool, this tool also has a number of useful features – see Table 8, below.

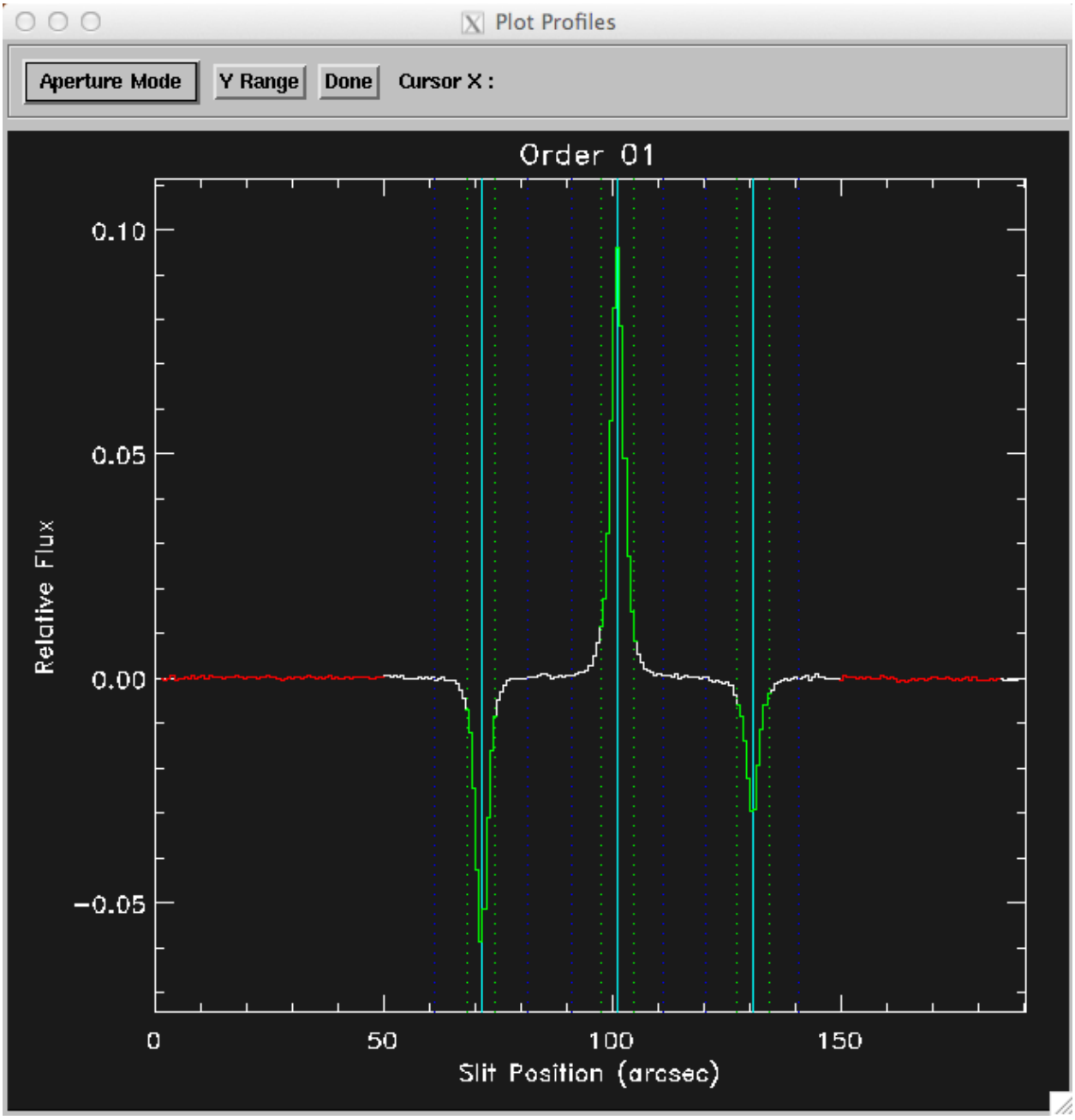


Fig. 20: Aperture locations automatically identified and overplotted on the spatial profile. The three positions identified (light blue lines) correspond to the positive and negative spectra in Fig. 17. Green lines indicated the extraction aperture, dark blue lines indicate the PSF radius (the point at which the flux goes to zero), and red lines indicate background regions.

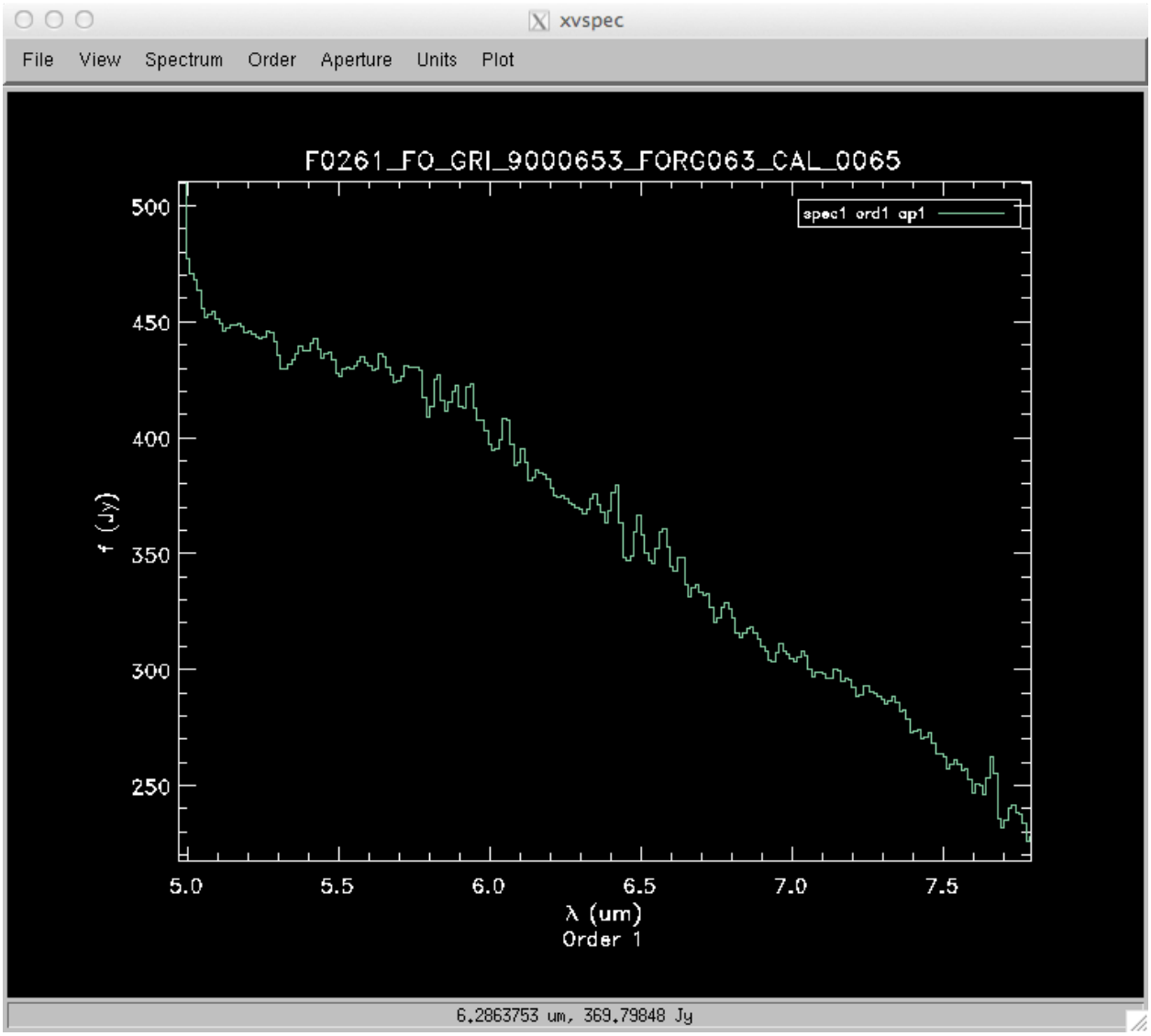


Fig. 21: Final extracted spectrum, displayed in xvspec

Table 8: 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 reference wavelengths	**Plot->Overplot->Line List**	–
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**.

Useful Parameters

Below are listed some key parameters for the grism processing steps. Note that the Load Data through Stack Chops/Nods steps are identical to those used for the imaging data: their parameters are listed above.

- **Make Profiles**
  - *Row fit degree*: Typically a third-order polynomial fit is used to calculate the smooth spatial profile. Occasionally, a higher or lower order fit may give better results.
  - *Wavecal file*: If 'default', the default wavelength calibration file on disk will be used. If blank, no wavelength calibration will be applied. Set to a valid FITS file path to override the default wavelength calibration map with a new one.
  - *Subtract median background*: If checked, and the SRCTYPE is not EXTENDED_SOURCE and the data is not cross-dispersed, then the median level of the smoothed spatial profile will be subtracted out to remove residual background from the total extracted flux. This option is appropriate as long as the slit is dominated by background, rather than source flux. If the spatial profile dips below zero at any point (other than for a negative spectrum), this option should be deselected.
- **Find Objects**
  - *Number of auto apertures*: If blank, the number of expected apertures will be determined from the instrument mode of the observation. For extended sources, a single aperture at the center of the slit is defined,

for full-slit extraction. Otherwise, for NMC mode, with on-slit chopping, 3 apertures are expected. For NPC modes, 2 apertures are expected. For all other modes, 1 aperture is expected. To override any of these defaults, set this parameter to 1 to auto-find the single brightest source, or 2 to find the two brightest sources, etc. Sources may be positive or negative.

- *Guess position*: Enter a guess value for the aperture to use as a starting point. Values are in arc seconds 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 two apertures in each of two files, 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.
- *Fix position*: Enter a value to use as the aperture center. The format of this parameter is the same as for the *Guess position* parameter, but no fit will be done to attempt to refine the position: it will be used as entered.

- **Trace Objects**
  - *Fit trace*: If 'Fit to continuum' is selected, points along the continuum will be fit with a Gaussian to determine the trace center at each location, and then the positions will be fit with a low-order polynomial. If 'Fix to aperture position' is selected, no fit will be attempted, and the default slit curvature defined by the edge definition file will be used as the aperture location. If 'Auto' is selected, the trace will be fixed for extended sources, but a fit will be attempted for all other data types. If the trace fit fails for any reason, it will fall back to a fixed aperture location.
- **Set Apertures**
  - *Auto*: If checked, the pipeline will use the header parameters to determine how to set the apertures automatically. If SRCTYPE=EXTENDED_SOURCE, the aperture will be set for full-slit, standard extraction. 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 arc seconds 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 PSF radius*: Enter a value to use as the PSF radius (the value at which the total flux from the source goes to zero). 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 SRCTYPE=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 FORCAST.
- *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. The default is 10 for FORCAST.
- *Slit function response image*: If 'default', the default slit response correction file on disk will be used. If blank, no slit correction will be applied. Set to a valid FITS file path to override the default slit correction file with a new one.

- **Flux Calibrate**
  - *Skip calibration*: If set, no telluric correction or flux calibration will be applied.
  - *Optimize ATRAN correction*: If set, the pipeline will use a library of ATRAN files to attempt to automatically select the best telluric correction. This option requires that the external library location be identified in the parameter below. The procedure may take some time to complete, and may not complete successfully for faint spectra, or spectra with significant emission features. Optimization will not be attempted for the FOR_G111 grism, or for spectra with mean S/N < 10.
  - *Calibration directory for optimization*: If the Optimize ATRAN parameter is set, this parameter specifies the location of the library of ATRAN FITS files to use.
  - *ATRAN file if not optimizing*: If the Optimize ATRAN parameter is not set, this parameter will be used to determine the ATRAN file to use for telluric correction. If 'default', the default ATRAN file on disk will be used. If blank, no telluric correction will be applied. Set to a valid FITS or .sav file path to override the default ATRAN file with a new one.
  - *Response file*: If 'default', the default instrumental response file on disk will be used. If blank, no response correction will be applied, but transmission correction will still occur. Set to a valid FITS file path to override the default response file with a new one.
  - *Override resolution (l/dl)*: Set to override the default assumed resolution of the grism.
  - *Auto shift telluric spectrum*: If checked, the data will be cross-correlated with the transmission spectrum and shifted to match it. This may help alleviate minor shifts in wavelength. However, if the correlation between the transmission spectrum and the source spectrum is not strong, this option may hurt more than help. In this case, uncheck this option. Auto shift will not be attempted for the FOR_G111 grism.
  - *Override shift (pixels)*: Set to specify a manual shift in pixels along the wavelength axis to apply to the science spectrum.

- **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. This option is not commonly used.

  - *Combination statistic*: Select the combination method. The default is median.

- **Make Response**
  - *Skip response generation, flux calibrate instead*: If checked, input standard spectra will be calibrated using the Flux Calibrate step, instead of used to make response spectra. This may be a useful check on the calibration procedure, since calibrated standards can be directly compared to their models. All other parameters for this step are the same as for the Flux Calibrate step.
- **Make Spectral Cube**
  - *Run in parallel*: By default, Redux will attempt to use all CPUs available (up to 9) to process data in parallel. Uncheck this box to use serial processing on a single CPU instead. Serial reduction is useful for debugging pipeline errors, and may be necessary for some architectures and/or IDL versions.
  - *Local polynomial surface fit order*: This parameter controls the order of the surface fit to the data at each grid point. Higher orders give more fine-scale detail, but are more likely to be unstable. Set to zero to do a weighted mean of the nearby data.
  - *Fit window*: This parameter controls how much data to use in the fit at each grid point. It is given as a radius, in pixels. Higher values will lead to more smoothing in the output data; too-low values may result in missing data (holes) in the output map.
  - *Smoothing radius*: This parameter controls the width of the distance-weighting Gaussian. It is given as a radius, in pixels. Lowering this value results in finer detail for the same input fit window. Too low values may result in noisy output data; too high values effectively negate the distance weights.
  - *Fit rejection threshold*: If the fit value is more than this number times the standard deviation away from the weighted mean, the weighted mean is used instead of the fit value. This parameter is used to reject bad fit values.
  - *Positive outlier threshold*: Sets the rejection threshold for the input data, in sigma.
  - *Negative outlier threshold*: If non-zero, sets a separate rejection threshold in sigma for negative fluxes, to be used in a first-pass rejection.
  - *Weighting scheme*: Selects the weighting scheme for the fits. Options are errors and distance (default), errors only, distance only, or no weighting.
  - *Reduce edge effects*: If selected, the order of the fit will be lowered near the edges of the image, to reduce bad pixels due to unstable fits near the edges.
  - *Display plots*: Select to display a plot of the actual and resampled flux surface for each wavelength plane. If selected, processing will be performed serially for this step.

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 calibrated (*CAL*) file generated for each observation, in both imaging and grism modes. Check the data product table (Tables 1 and 2, above) for other expected data products for each mode.

- For imaging:
  - If shifting-and-adding was performed at the merge step, display all undistorted (*UND*) and merged (*MRG*) files. Check that the pattern of positive and negative sources looks right for the observation mode. Also check that the FWHM of the source is not worse in the merged files than it was in the undistorted files. If the pattern does not look right, or the FWHM is too large, the merge may have failed.
  - Display all registered and telluric-corrected (*TEL*) files at once and check that any visible sources appear at the same pixel locations in all files.
  - Display the final coadded (*COA*) or calibrated file and check that the FWHM is not worse than it is in the registered files, which might indicate poor registration. Check for any unusual artifacts, such as variable background regions, or detector pattern noise.
  - Compare the calculated reference calibration factors for all flux standards to the last known series average. Major deviations may indicate that the photometry failed for that observation.
- For grism:
  - Display the stacked (*STK*) and stacked-dithers (*SKD*) files. Verify that the expected number of spectral traces appears in the image.
  - Display the spatial profile with apertures overlaid. Verify that apertures look well placed and the spatial profile does not dip below zero (except for negative spectral traces).
  - Display the rectified image, and overlay the locations of the extracted apertures. Verify that the apertures lie on top of any visible spectral traces.
  - Display the extracted spectra (*MRG*) along with the calibrated spectrum (*CAL* or *CMB*) and/or overplot the expected atmospheric transmission. Check that the calibrated spectrum does not include residual artifacts from the telluric absorption features. If it does, the assumed resolution for the grism, or the wavelength calibration of the observation, may need updating.
  - Overlay a model spectrum on the calibrated spectra of flux standards. Verify that the observed spectrum matches the theoretical spectrum, within the error bars of the observation. If it does not, the instrumental response file may need updating.

Part IX

Appendix A: Required input keywords

This table describes the type and expected value for all FITS keywords used by the DRIP/FSpextool pipelines.

Table 9: Required input keywords

Keyword	Type	Expected value
ALTI_STA	float	0-60000.
ALTI_END	float	0-60000.
AOR_ID	string	
DATASRC	string	ASTRO, CALIBRATION, LAB, TEST, OTHER, FIRSTPOINT
DATE-OBS	string	yyyy-mm-ddThh:mm:ss[.sss]
DETCHAN	string	SW, LW
DETECTOR	string	As-010, Sb-083
DETITIME	float	> 0
EPERADU	float	> 1

Continued on next page

Table 9 – continued from previous page

FRMRATE	float	> 0
ILOWCAP	bool	
INSTCFG	string	IMAGING_SWC, IMAGING_LWC, IMAGING_DUAL, GRISM_XD, GRISM_SWC, GRISM_LWC, GRISM_DUAL, GRISM_XD-LSV, GRISM-SSV, GRISM-LSV
INSTMODE	string	C2, C2N, C2NC2, N, SLITSCAN, NXCAC
INSTRUME	string	FORCAST
INTTIME	float	
MISSN_ID	string	
NAXIS1	int	256
NAXIS2	int	256
OBJECT	string	
OBS_ID	string	
OBSTYPE	string	OBJECT, STANDARD_FLUX, STANDARD_TELLURIC, LAMP, FLAT, DARK, BIAS, SKY, BB, GASCELL, LASER, FOCUS_LOOP
OTMODE	string	AD, SUR
OTSTACKS	int	>0
SPECTEL1	string	NONE, FOR_F054, FOR_F064, FOR_F066, FOR_F077, FOR_F111, FOR_F113, FOR_F197, FOR_F253, FOR_XG063, FOR_XG111, FOR_G063, FOR_G111
SPECTEL2	string	NONE, FOR_F086, FOR_F113, FOR_F118, FOR_F254, FOR_F315, FOR_F336, FOR_F348, FOR_F371, FOR_F242, FOR_G227, FOR_G329
TELESCOP	string	SOFIA
TIME-OBS	string	
UTCSTART	string	
WAVELNTH	float	0-40.
ZA_START	float	0-90.
ZA_END	float	0-90.
DITHER	bool	
DTHCRSYS	string	SIRF, ERF
DTHINDEX	int	> 0
DITHERX	float	
DITHERY	float	
CHOPPING	bool	
CHPCRSYS	string	SIRF, ERF
CHPAMP1	float	geq 0
CHPANGLR	float	
CHPANGLE	float	
CHPNPOS	int	> 0
NODDING	bool	
NODCRSYS	string	SIRF, ERF
NODAMP	float	geq 0
NODANGLR	float	
NODANGLE	float	
NODBEAM	string	A, B
SKY_ANGL	float	
SKYMODE	string	C2NC2, NMC, NPC, NPCNAS, NPCCAS, SLITSCAN, NOD, NXCAC
SRCTYPE	string	POINT_SOURCE, EXTENDED_SOURCE, OTHER, UNKNOWN
SLIT	string	FOR_SS24, FOR_LS24, FOR_LS47, NONE
RN_HIGH	float	

Continued on next page

Table 9 – continued from previous page

RN_LOW	float	
BETA_G	float	
BORDER	int	
BGSUB	int	0, 1
ANGLCONV	string	'positive', 'negative'
CHOPTSACONV	int	-1, 1
XYSHIFT	float	
SHIFTORD	int	0, 1, 3
CORMERGE	string	COR, CENT, N, NOSHIFT
CORCOADD	string	COR, CENT, N, NOSHIFT, WCS
MTHRESH	float	
CTHRESH	float	
JBCLEAN	string	MEDIAN, FFT, N
FRACDROOP	float	
MINDROOP	float	
MAXDROOP	float	
NRODROOP	int	
ORDER	int	
PIN_NPTS	int array	should have 2 elements
PIN_SPX	int array	should have 8 elements
PIN_SPY	int array	should have 8 elements
NLRSWCHI	float	
NLSSWCHI	float	
NLCSWCHI	float array	should have >1 element
LIMSWCHI	float array	should have 2 elements
NLRSWCLO	float	
NLSSWCLO	float	
NLCSWCLO	float array	should have >1 element
LIMSWCLO	float array	should have 2 elements
NLRLWCHI	float	
NLSLWCHI	float	
NLCLWCHI	float array	should have >1 element
LIMLWCHI	float array	should have 2 elements
NLRLWCLO	float	
NLSLWCLO	float	
NLCLWCLO	float array	should have >1 element
LIMLWCLO	float array	should have 2 elements
NLINSECTION	int array	should have 4 elements

Part X

Appendix B: Sample configuration files

This is a sample of the dripconf.txt configuration file, used to set reduction parameters for DRIP algorithms and override FITS keyword values as necessary.

```
; Set to 1 if you want to check the header keywords of the input data
DOINHDCH=1
```

(continues on next page)

(continued from previous page)

```
; Parameters for variance calculation
rn_high=2400.
rn_low=244.8
beta_g=1.0

; Parameters for resizing imaging
border=200

; Parameters for stacking
; if bgsub=1, residual background will be subtracted after stacking
bgsub=1

; Parameters for merging and coadding
anglconv='negative'
choptsaconv=-1
xyshift=15.
shiftord=3
; cormerge='XCOR' triggers drip_merge to use cross correlation
; cormerge='CENTROID' triggers drip_merge to use centroid
; cormerge='HEADER' triggers drip_merge to use nominal chop/nod positions
; cormerge='NOSHIFT' triggers drip_merge to not merge negative images
CORMERGE = 'NOSHIFT'
CORCOADD = 'HEADER'
cthresh=20.
mthresh=20.
maxregsh=2000

; Jailbar cleaning
; jbclean = 'FFT' triggers cleaning jailbar pattern with fft
; jbclean='MEDIAN' trigger cleaning jailbar pattern with median filter
; jbclean='N' no jailbar cleaning
jbclean = 'MEDIAN'

; distortion correction
order=3
fpinhole='pinhole_locs.txt'
pin_npts=[12,12]
pin_spx=[3,3,3,2,5,5,6,6]
pin_spy=[1,2,3,3,5,6,5,6]

; droop correction
mindroop    = 0.0
maxdroop    = 65535.0
nrodroop = 16
fracdroop = 0.0035

; Global image correction
nlinsection=[128,128,190,190]

; SWC high cap
nlrswchi = 7000.
nlsswchi = 7000.
nlcswchi = [0.99662136,0.41674931,-0.037328637,-0.073270770,-0.10386704,0.015290507]
limswchi  = [1613.2, 11349.0]

; SWC low cap
```

(continues on next page)

(continued from previous page)

```
nlrswclo = 6000.
nlsswclo = 6000.
nlcswclo = [0.99762921,0.34382635,-0.019972477,-0.040238521,-0.16961677,0.080713727]
limswclo  = [1926.0, 11905.0]

; LWC high cap
nlrlwchi = 7000.
nlslwchi = 7000.
nlclwchi = [0.9990756,0.3431573,-0.0614661,-0.0603023,-0.2518475]
limlwchi  = [2500.0, 11000.0]

; LWC low cap
nlrlwclo = 7000.
nlslwclo = 7000.
nlclwclo = [0.9986659,0.3415591,-0.1474720,-0.0784636]
limlwclo  = [2500.0, 11000.0]
```

This is a sample of the FSpextool configuration file for FORCAST, called FORCAST.dat. The first sections list parameters used by the pipeline. Also included is a list of all keywords that will be propagated from the input files to the output files.

```
#
# Generated by M. Clarke, 22 February 2013
#
# This is the calibration file for the FORCAST spectrograph on SOFIA.
# Note the values must be in the correct order, but can have any number of
# spaces/comments between them.
#
#================================================================================
#
 INSTRUMENT=FORCAST
 NCOLS=256
 NROWS=256
 STDIMAGE=256
 PLOTWINSIZE=700 512
 FILENAME=FILENAME
 EXPTIME=EXPTIME
 TIME=TIME_OBS
 POSANGLE=None
 HA=None
 AIRMASS=None
 NINT=4
 BADPIXMASK=None
%
%CAL BASE
%
 CALMODULE=mc_forcastcals1d
%
%FILE READ MODE
%
 FILEREADMODE=Filename
 IPREFIX=F
 OPREFIX=r
 SUFFIX=*.fits*
 FITSREADPROGRAM=mc_readforcastfits
 HEADCOMBPROGRAM=mc_forcastdcshdr
```

(continues on next page)

(continued from previous page)

```
 YUNITS=Me/s
 YTITLE=f (!5Me s!u-1!n)
 XUNITS=um
 XTITLE=!7k!5 (!7l!5m)
%
% Reduction Mode
%
 REDUCTIONMODE=A
%
% Combine Base Information
%
 COMBMODE=A
 COMBSTAT=Median (Median Error)
 COMBTHRESH=8.0
 COMBODIR=proc/
%
%  Sky Base Information
%
 SKYSTAT=Robust Weighted Mean
 SKYTHRESH=8.0
%
% Profile Parameters
%
 YBUFFER=2
 OVERSAMP=1
 ATMOSTHRESH=0
%
% Point Source Base
%
PSNAPS=1
PSPSFRAD=22.0
PSAPRAD=7.0
PSBGSUB=1
PSBGSTART=24.0
PSBGWIDTH=30
PSBGDEG=0
PSBGMULT=3.0
%
% Extended Source Base
%
XSBGSUB=1
XSBG=0-9,24-36,54-60
XSBGDEG=0
%
% Additional processing base
%
 ADDLMODULE=mc_forcastaddlproc
%
%Other Base Parameters
%
 TRACEDEG=2
 TRACESTEP=3
 TRACESUMAP=3
 TRACESIGTHRESH=1
 TRACEWINTHRESH=5
 BADPIXELTHRESH=10
 PLOTSATURATEDPIXELS=0
```

(continues on next page)

(continued from previous page)

```
 SATURATION=3000
 CHECKSEEING=0
 SEEINGTHRESH=3
 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=ALTI_STA
KEYWORD=ALTI_END
KEYWORD=AOR_ID
KEYWORD=ASSC_AOR
KEYWORD=ASSC_MSN
KEYWORD=ASSC_OBS
KEYWORD=ATRNFILE
KEYWORD=CALERR
KEYWORD=C2NC2
KEYWORD=CHOPPING
KEYWORD=CHPAMP1
KEYWORD=CHPANGLE
KEYWORD=CHPANGLR
KEYWORD=CHPCOORD
KEYWORD=CHPNPOS
KEYWORD=CHPSETL
KEYWORD=CDELT1
KEYWORD=CDELT2
KEYWORD=CROTA2
KEYWORD=CRPIX1
KEYWORD=CRPIX2
KEYWORD=CRVAL1
KEYWORD=CRVAL2
KEYWORD=DATASRC
KEYWORD=DATE-OBS
KEYWORD=DETBIAS
KEYWORD=DETCHAN
KEYWORD=DETECTOR
KEYWORD=DETITIME
KEYWORD=DICHROIC
KEYWORD=DITHER
KEYWORD=DITHERCS
KEYWORD=DITHERX
KEYWORD=DITHERY
KEYWORD=DTHINDEX
KEYWORD=DTHNPOS
KEYWORD=EPERADU
KEYWORD=FILENUM
KEYWORD=FILTER
KEYWORD=FILT1_S
KEYWORD=FILT2_S
KEYWORD=FILT3_S
```

(continues on next page)

(continued from previous page)

```
KEYWORD=FILT4_S
KEYWORD=FITPWV
KEYWORD=FITPWVER
KEYWORD=FRMRATE
KEYWORD=ICONFIG
KEYWORD=ILOWCAP
KEYWORD=INSTCFG
KEYWORD=INSTMODE
KEYWORD=INSTRUME
KEYWORD=INTTIME
KEYWORD=MISSN-ID
KEYWORD=NODAMP
KEYWORD=NODANGLE
KEYWORD=NODANGLR
KEYWORD=NODBEAM
KEYWORD=NODCOORD
KEYWORD=NODDING
KEYWORD=OBJECT
KEYWORD=OBS_ID
KEYWORD=OBSTYPE
KEYWORD=OTMODE
KEYWORD=OTNBUFS
KEYWORD=OTSTACKS
KEYWORD=RSPNFILE
KEYWORD=SKY_ANGL
KEYWORD=SKYMODE
KEYWORD=SLIT
KEYWORD=SPECTEL1
KEYWORD=SPECTEL2
KEYWORD=SRCTYPE
KEYWORD=TELESCOP
KEYWORD=TOTINT
KEYWORD=UTCSTART
KEYWORD=WAVECENT
KEYWORD=WAVELNTH
KEYWORD=ZA_START
KEYWORD=ZA_END
%
% Keywords required by SOFIA DCS keyword dictionary
%
KEYWORD=KWDICT
KEYWORD=IMAGEID
KEYWORD=AOT_ID
KEYWORD=PROCSTAT
KEYWORD=HEADSTAT
KEYWORD=FILEREV
KEYWORD=PLANID
KEYWORD=DEPLOY
KEYWORD=FLIGHTLG
KEYWORD=ORIGIN
KEYWORD=OBSERVER
KEYWORD=CREATOR
KEYWORD=OPERATOR
KEYWORD=FILENAME
KEYWORD=DATE
KEYWORD=UTCEND
KEYWORD=WVZ_STA
```

(continues on next page)

(continued from previous page)

```
KEYWORD=WVZ_END
KEYWORD=TEMP_OUT
KEYWORD=TEMPPRI1
KEYWORD=TEMPPRI2
KEYWORD=TEMPPRI3
KEYWORD=TEMPSEC1
KEYWORD=AIRSPEED
KEYWORD=GRDSPEED
KEYWORD=LAT_STA
KEYWORD=LON_STA
KEYWORD=LAT_END
KEYWORD=LON_END
KEYWORD=HEADING
KEYWORD=TRACKANG
KEYWORD=TELCONF
KEYWORD=TELRA
KEYWORD=TELDEC
KEYWORD=TELVPA
KEYWORD=TELEQUI
KEYWORD=LASTREW
KEYWORD=FOCUS_ST
KEYWORD=FOCUS_EN
KEYWORD=TELEL
KEYWORD=TELXEL
KEYWORD=TELLOS
KEYWORD=TSC-STAT
KEYWORD=FBC-STAT
KEYWORD=OBSRA
KEYWORD=OBSDEC
KEYWORD=EQUINOX
KEYWORD=TRACMODE
KEYWORD=TRACERR
KEYWORD=MAPPING
KEYWORD=SCANNING
KEYWORD=DATATYPE
KEYWORD=MCCSMODE
KEYWORD=EXPTIME
KEYWORD=RESOLUN
KEYWORD=DETSIZE
KEYWORD=PIXSCAL
KEYWORD=SIBS_X
KEYWORD=SIBS_Y
KEYWORD=CHPFREQ
KEYWORD=CHPPROF
KEYWORD=CHPSYM
KEYWORD=CHPAMP2
KEYWORD=CHPCRSYS
KEYWORD=CHPTIP
KEYWORD=CHPTILT
KEYWORD=CHPPHASE
KEYWORD=NODTIME
KEYWORD=NODN
KEYWORD=NODSETL
KEYWORD=NODPATT
KEYWORD=NODSTYLE
KEYWORD=NODCRSYS
KEYWORD=DTHPATT
```

(continues on next page)

(continued from previous page)

```
KEYWORD=DTHOFFS
KEYWORD=MAPCRSYS
KEYWORD=MAPNXPOS
KEYWORD=MAPNYPOS
KEYWORD=MAPINTX
KEYWORD=MAPINTY
KEYWORD=SCNRA0
KEYWORD=SCNDEC0
KEYWORD=SCNRAF
KEYWORD=SCNDECF
KEYWORD=SCNRATE
KEYWORD=SCNDIR
%
% Pipeline keywords
%
KEYWORD=PIPELINE
KEYWORD=PIPEVERS
KEYWORD=PRODTYPE
KEYWORD=BAD_OID
KEYWORD=LINC_OID
KEYWORD=DARK_OID
KEYWORD=FLAT_OID
KEYWORD=PARENT*
KEYWORD=NEXP
KEYWORD=WAVSHIFT
KEYWORD=HISTORY
```

The following are sample FORCAST Redux parameter override files for imaging and grism modes in JSON format, with all values stored as strings. If present, the parameter value overrides the default defined by the FORCAST reduction object, or the DRIP and FSpextool configuration files. If not present, the default value will be used. The parameters displayed here are the current default values.

```
{
 "FORCAST_IMAGING_REDUCTION":
  {
   "Load Data":
    {
     "checkhead": "1",
     "saveimf": "0",
     "applycal": "1",
     "pathsave": ".",
     "betag": "1.0"
    },
   "Clean Images":
    {
     "autoshift": "1",
     "shiftfile": "",
     "badfile": "default",
     "jbmethod": "Median"
    },
   "Correct Droop":
    {
     "fracdroop": "0.0035",
     "mindroop": "0.0",
     "maxdroop": "65535.0",
     "nrodroop": "16"
```

(continues on next page)

(continued from previous page)


```
 },
"Correct Nonlinearity":
 {
  "secctr": "128,128",
  "secsize": "190,190"
 },
"Stack Chops/Nods":
 {
  "bgscale": "0",
  "bgsub": "1",
  "secctr": "128,128",
  "secsize": "240,240",
  "stkdthr": "0",
  "statistic": "Median (MAD)",
  "thresh": "8.0",
  "choptsa": "-1"
 },
"Undistort":
 {
  "order": "3",
  "border": "200"
 },
"Merge":
 {
  "cormerge": "No shift",
  "anglconv": "negative",
  "mthresh": "20.",
  "mfwhm": "4.5",
  "shiftord": "3",
  "maxshift": "120",
  "stripborder": "1"
 },
"Register":
 {
  "corcoadd": "Compare WCS",
  "xyshift": "15.",
  "cthresh": "20.",
  "mfwhm": "4.5",
  "shiftord": "3",
  "usecent": "1",
  "offsets": ""
 },
"Telluric Correct":
 {
 },
"Coadd":
 {
  "skipcoadd": "0",
  "statistic": "Median (MAD)",
  "thresh": "8.0"
 },
"Flux Calibrate":
 {
  "recalc": "0",
  "srcpos": "",
  "fitsize": "138",
  "fwhm": "5.0",
```


(continues on next page)

(continued from previous page)

```
     "profile": "Moffat"
    }
  }
}
```

```
{
 "FORCAST_GRISM_REDUCTION":
  {
   "Load Data":
    {
     "checkhead": "1",
     "saveimf": "0",
     "pathsave": ".",
     "betag": "1.0"
    },
   "Clean Images":
    {
     "autoshift": "1",
     "shiftfile": "",
     "badfile": "default",
     "jbmethod": "Median"
    },
   "Correct Droop":
    {
     "fracdroop": "0.0035",
     "mindroop": "0.0",
     "maxdroop": "65535.0",
     "nrodroop": "16"
    },
   "Correct Nonlinearity":
    {
     "secctr": "128,128",
     "secsize": "190,190"
    },
   "Stack Chops/Nods":
    {
     "bgscale": "0",
     "bgsub": "0",
     "secctr": "128,128",
     "secsize": "240,240",
     "stkdthr": "0",
     "statistic": "Median (MAD)",
     "thresh": "8.0",
     "choptsa": "-1"
    },
   "Make Profiles":
    {
     "ndeg": "3",
     "wavecal": "default",
     "bgsub": "1"
    },
   "Find Objects":
    {
     "numaps": "",
     "guess": "",
     "fix": "",
```

(continues on next page)

(continued from previous page)


```
     "exclorders": ""
    },
   "Trace Objects":
    {
     "trace": "Auto (decide fix vs. fit from source type)"
    },
   "Set Apertures":
    {
     "auto": "1",
     "bgsub": "0",
     "optimal": "0",
     "aprad": "",
     "psfrad": "",
     "bginfo": "",
     "apsign": ""
    },
   "Extract Spectra":
    {
     "algorithm": "Auto (decide algorithm from source type)",
     "medprof": "0",
     "bgorder": "",
     "fixpix": "1",
     "bdpxthresh": "",
     "slitfn": "default",
     "debug": "None"
    },
   "Merge Spectra":
    {
     "debug": "0"
    },
   "Refine Wavecal":
    {
     "interactive": "0",
     "shift": ""
    },
   "Flux Calibrate":
    {
     "skipcal": "0",
     "optimize": "0",
     "calibdir": "$DPS_SHARE/calibrations/ATRAN/fits",
     "snthresh": "10",
     "atranfile": "default",
     "respfile": "default",
     "resolution": "",
     "autoshift": "1",
     "shiftval": "",
     "debug": "0"
    },
   "Combine Spectra":
    {
     "combineaps": "1",
     "scale": "1",
     "correctshape": "0",
     "statistic": "Median (MAD)"
    }
  }
}
```