Running Sample IMACS data through COSMOS

Edward Villanueva

August 28, 2014

Contents

1 Introduction

This documentation includes all the steps you need to perform in order to reduce the sample IMACS f/2 data with COSMOS. For your convenience, I've also included brief instructions for setting up your environment, for those of you who are new to COSMOS. These steps mirror what is outlined in the Cookbook (http://code.obs.carnegiescience.edu/cosmos/Cookbook.html), but offer more details for reducing this sample data set.

2 Setting up COSMOS and Sample Data

2.1 Download and Install COSMOS

If you haven't done so already, you can download the latest COSMOS source and binary packages from the following page:

```
http://code.obs.carnegiescience.edu/cosmos/Downloads.html
```

On this page are instructions for installing the binaries, as well as, a link to the "Installing and Running COSMOS" page, which contains details for building COSMOS from source.

2.2 Download sample IMACS f/2 data

The sample IMACS f/2 data can be downloaded from the "Downloads" page (see URL in previous section) and is in a compressed tar file. You can unpack the tar file using the following command:

```
tar -zxvf IMACS_f2_sample.tgz
```

You can also use the **wget** or **curl** command to download the sample data from our ftp server:

```
wget ftp://ftp.obs.carnegiescience.edu/pub/cosmos/IMACS_f2_sample.tgz
curl -O ftp://ftp.obs.carnegiescience.edu/pub/cosmos/IMACS_f2_sample.tgz
```

Included in the the tar file is a directory called **IMACS_f2_sample**, which contains the following data frames:

filename	exposure type
ift0009	science frame
ift0010	science frame
ift0011	spectroscopic flat frame
ift0012	comparison lamp frame

The other files that are included are **CSIfld5.SMF**, which is the Slit Mask File containing the slit definitions, and **CONTENTS**, which contains a description of the IMACS sample data frames.

2.3 Setting up your environment

As of COSMOS 2.20, you only need to source your **cosmos2-setup** file in order to set up COSMOS. First, you should edit your **cosmos2-setup** file so that the **COSMOS_HOME** environment variable points to the location of your build. For example, if you installed COSMOS in */Users/vulcan/Cosmos/Cosmos2-20*, then you would change the first line in **cosmos2-setup** to have the following:

```
setenv COSMOS_HOME /Users/vulcan/Cosmos/Cosmos2-20
```

After you are satisfied with your **cosmos2-setup** file, you can source it on the command line. You may also want to add the command to your startup file:

```
source /Users/vulcan/Cosmos/Cosmos2-20/cosmos2-setup
```

There are several ways in which you can set up your reductions, but here I opt to run the COSMOS routines and edit the parameter files in the same directory as the data. So I leave both **COSMOS_IMAGE_DIR** and **COSMOS_PAR_DIR** environment variables as "." in the **cosmos2-setup** file.

Say you unpack **IMACS_f2_sample.tgz** in */Users/vulcan/Reductions*. Then you will want to run the following commands to copy over your parameter files to the same directory as the data:

```
cd /Users/vulcan/Reductions/IMACS_f2_sample
cp $COSMOS_HOME/Cospar/*.par .
```

These data were taken with the short camera (f/2) using the 200 line grism, and are in Normal orientation, where the spectra are dispersed vertically (as opposed to "Nod & Shuffle" orientation, where the spectra are dispersed horizontally). So we can use the **SC200g_N.dewoff** file as a starting point. Most example dewoff files were made using older data, but some files have recently been updated using newer data and are stored in *$COSMOS_HOME/examples/dewoff/Aug2014/*. One of those updated files is **SC200g_N.dewoff**, so let's copy that over to our working directory:

```
cp $COSMOS_HOME/examples/dewoff/Aug2014/SC200g_N.dewoff .
```

You can also copy over example HeNeAr line lists to use for wavelength calibration, if you don't already have one to use:

```
cp $COSMOS_HOME/examples/linelists/HeNeAr*.dat .
```

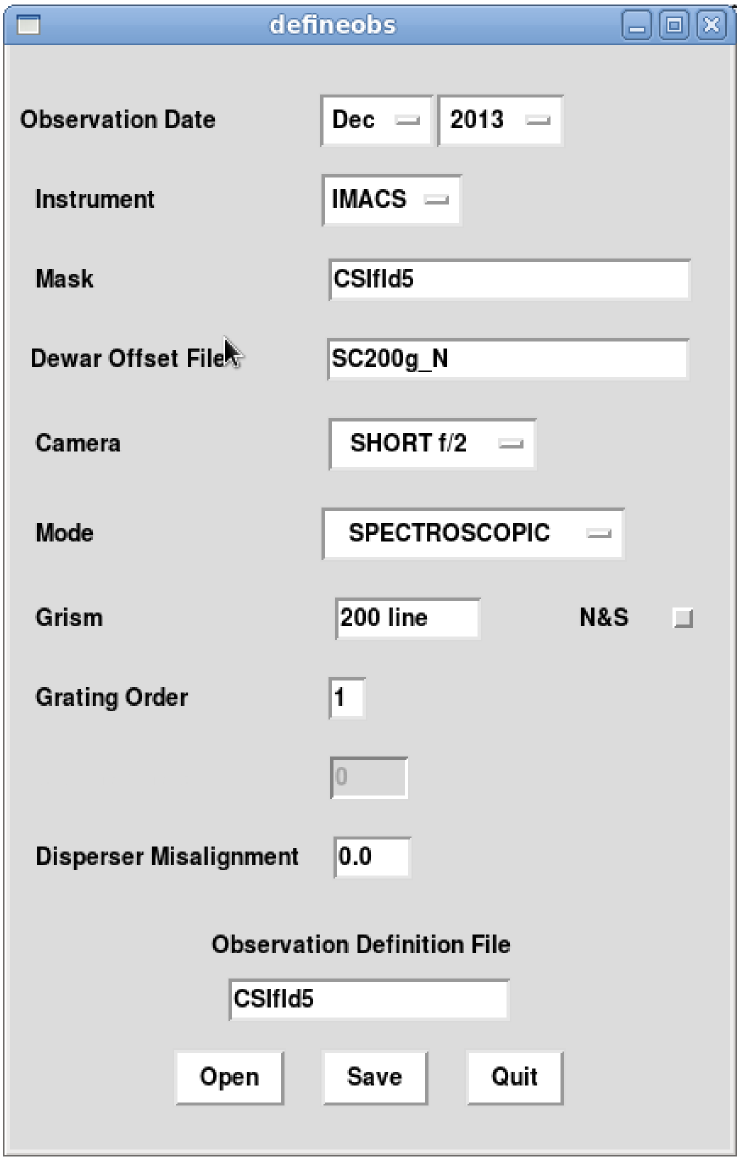
3 Setting up observation definition files

The **defineobs** routine is used to set up the observation definition (.obsdef) file. You can use the traditional Perl GUI version of **defineobs** (you need to have the appropriate Perl modules installed), the simple command-line version called **defineobs.py**, or for Mac users, the Mac OS X defineobs app.

Figure 1 shows what you would input for the sample data in the traditional Perl **defineobs** GUI, which you invoke simply by typing "defineobs" on the command line. Notice that no suffixes are included in the names for the Mask, Dewar Offset File, or Observation Definition File. The programs will automatically add these suffixes when looking for these files. Since the mask is named **CSIfld5**, I decide to give the Observation Definition File the same name. Clicking the "Save" button will create **CSIfld5.obsdef**. After saving, click on the "Quit" button to exit the GUI.

If you find you need to edit **CSIfld5.obsdef**, just run **defineobs** as follows to load your existing obsdef file:

```
defineobs CSIfld5
```



Observation Date	Dec 2013		
Instrument	IMACS		
Mask	CSIfld5		
Dewar Offset File	SC200g_N		
Camera	SHORT f/2		
Mode	SPECTROSCOPIC		
Grism	200 line	N&S	□
Grating Order	1		
	0		
Disperser Misalignment	0.0		

Observation Definition File

CSIfld5

Figure 1: defineobs input for sample IMACS f/2 data.

4 Spot-check the features

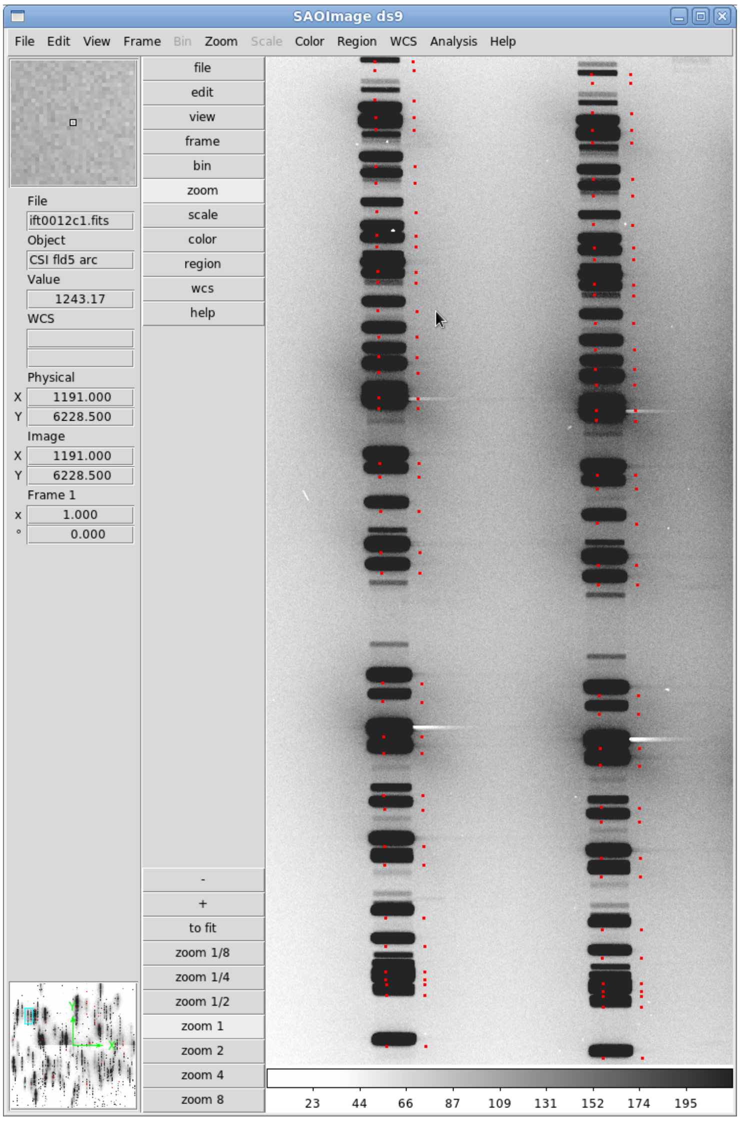
Before doing any wavelength calibration, it is beneficial to check how good your initial dewar offset file is, since the initial offset may be too large for align-mask to detect. To do this, we can run the **spectral-lines** program with the full HeNeAr.dat line list to produce a new file with the X and Y coordinates of the comparison lines for each slit:

```
spectral-lines -o CSIfld5 -l HeNeAr.dat -b 1 -e
```

Then we use **display8** and **tvmark** in IRAF to display a mosaic of the lamp frames and overplot the positions from the resulting **CSIfld5.xy** file:

```
ecl> display8 ift0012 1
ecl> tvmark 1 CSIfld5.xy color=204
```

As you can see in Figure 2, the initial positions aren't too far off, so we can proceed to the next step of aligning the masks. If they don't seem close at all, you will need to use the **adjust-offset** routine to adjust the dewoff file. You may also need to alter the *Disperser Misalignment* with **defineobs** if you see a rotation offset in the positions.



File
ift0012c1.fits
Object
CSI fld5 arc
Value
1243.17
WCS
Physical
X 1191.000
Y 6228.500
Image
X 1191.000
Y 6228.500
Frame 1
x 1.000
° 0.000

Figure 2: Zoomed-in section of lamp mosaic displayed in ds9, with initial positions overplotted using the IRAF 'tvmark' routine.

5 Aligning the masks

The first script we need to run in order to make a spectral map is **align-mask**, which uses the comparison lines to calculate the large-scale offsets and adjust the dewoff file accordingly. To edit the parameters for **align-mask**, we use the **editpar** routine:

```
editpar align-mask

type q to end

SEARCHBOX               10      half width of search range
MAGFACTOR              100      magnification factor for offset vectors
LAMFILE         somelines.dat      line list file
NAVER                    1      number of neighboring points to average
THRESHOLD         0.100000      minimum flux ratio spot to background
SIGLIMIT         10.000000      line rejection threshold
MAXFLUX         65000.000000      maximum flux
WINDOWSIZE        7.000000      width of PGPLOT window in inches


Change parameter:
```

For these data, I put the lines at 4921.93, 5015.68, and 5400.56 Angstroms in **somelines.dat**, because those lines are relatively isolated and not saturated. Avoid using more than 3 lines, as this may cause confusion in **align-mask**. Then, we run the command:

```
align-mask -o CSIfld5 -f ift0012
parameter file ./align-mask.par found
Processing "CSIfld5.SMF".
** Unknown keyword "DREF" ignored.
** Unknown keyword "DATE" ignored.
** Unknown keyword "XFWH9HRN1EPQX4OR4HP4KGD_QYMVUM
FXKNVSS7QHPASN9ETBCADLCWZA_W5NU23IMLDY6Z" ignored.
Using E2V1 distortion corrections
Delta=(-10.01 5.48) Sigma=(3.35 5.08)
Clipping out (1532.45,-343.53)
Clipping out (1534.07,-546.96)
Delta=(-10.02 5.48) Sigma=(3.33 5.06)
Clipping out (-2270.69,1252.93)
Clipping out (425.03,2647.53)
Clipping out (425.70,2435.20)
Delta=(-10.03 5.48) Sigma=(3.31 5.06)
Clipping out (-2518.94,3031.25)
Clipping out (-1780.39,-1325.13)
Clipping out (-3346.16,-229.54)
Clipping out (-3346.76,-485.34)
Clipping out (2674.58,-582.58)
Clipping out (1204.60,-1409.91)
```

```
297 matches found. sigma=  12.74 pixels  Continue? y

Average scale =          0.99869    Reset scale? y

Average rotation angle =  -0.013    Reset angle? y

Average X,Y Offsets =    -9.9    6.2 pixels. Apply shifts? y
Creating new dewar offset file ift0012.dewoff
```

A PGPLOT window will pop up with a vector plot of the calculated offsets. As you can see in Figure 3, **align-mask** detects the large-scale shifts and applies these corrections to the dewoff file. If you do not see a large-scale shift such as this for other data, then most likely you need to either increase your *SEARCHBOX* or adjust the dewoff file manually using the **adjust-offset** routine. Since we see a noticeable bulk shift, however, we can proceed to adjust the scale and angle, and apply the shifts to the dewoff file.

The output of **align-mask** in this case is **ift0012.dewoff**. Edit **CSIfld5.obsdef** to use **ift0012.dewoff** and then rerun **align-mask**. After a few iterations, you'll notice that the offset "sigma" has gone down to less than 2 pixels (see Figure 4), which means we now have a good enough dewoff to move on.

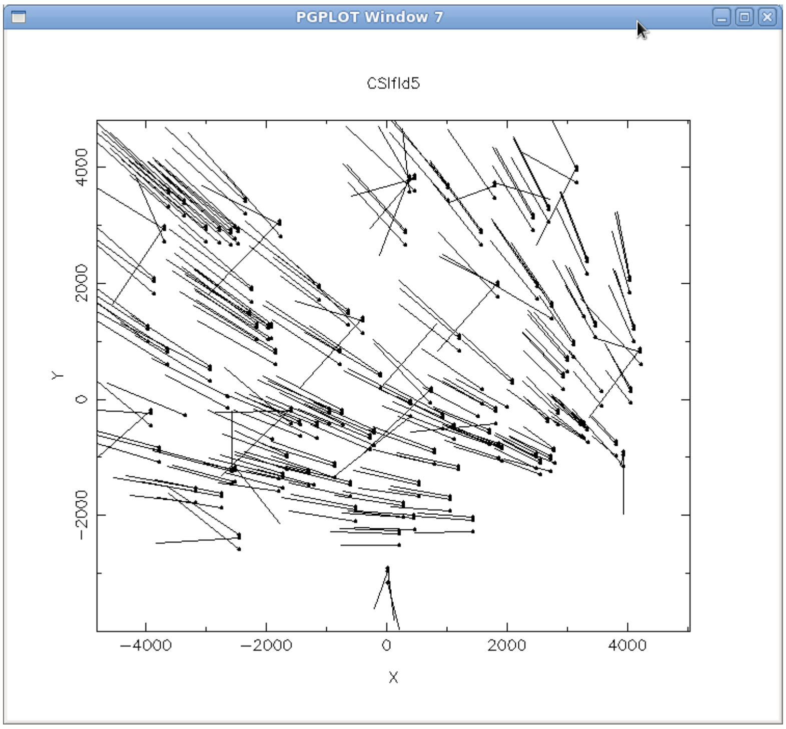


Figure 3: Vector plot of offsets as calculated by first run through align-mask.

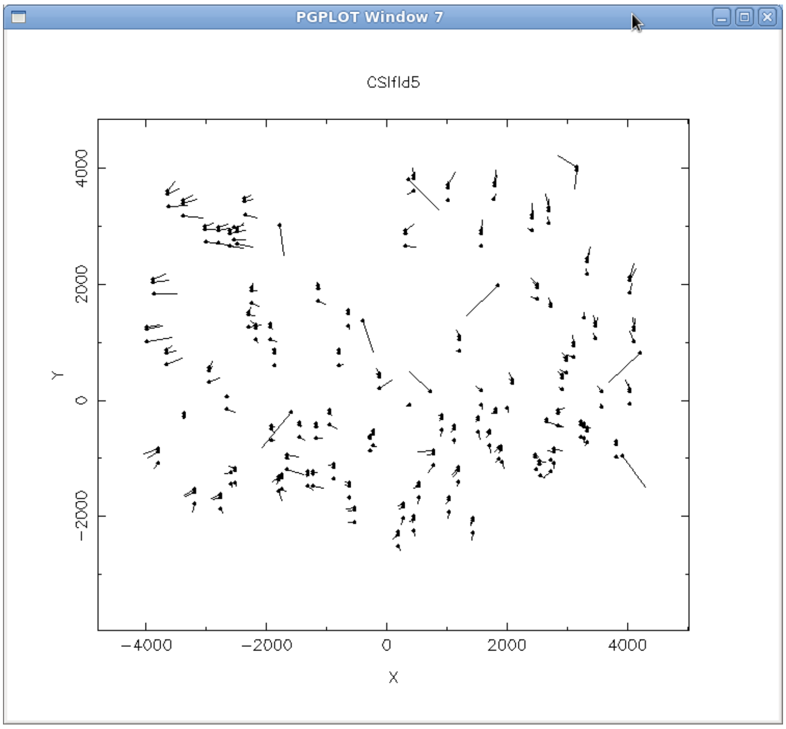


Figure 4: Vector plot of offsets after several iterations of align-mask.

6 Constructing the Spectral Map

6.1 Generating an Initial Map

Now that we have a good dewoff file, we can run **map-spectra** to make an initial spectral map. These data were taken with the *WB4800-7800* filter, so we set the wavelength range from 4800 Angstroms to 7800 Angstroms in map-spectra.par so that only this range is mapped:

```
editpar map-spectra
./map-spectra.par

type q to end

minlambda           4800.000000       minimum wavelength
minpixl                     100      minimum # pixels/spectrum
maxlambda           7800.000000       maximum wavelength
use_holes                     y      map holes as well as slits?
chipnum                       8      number of chips being used
```

Then we run the **map-spectra** command with our obsdef file as input:

```
map-spectra CSIfld5
parameter file ./map-spectra.par found
```

```
Processing "CSIfld5.SMF".
** Unknown keyword "DREF" ignored.
** Unknown keyword "DATE" ignored.
** Unknown keyword "XFWH9HRN1EPQX4OR4HP4KGD_QYMVUM
FXKNVSS7QHPASN9ETBCADLCWZA_W5NU23IMLDY6Z" ignored.
Using E2V1 distortion corrections
Processing slit 117
The following slits straddled a chip boundary, and were not mapped:

Slit  20   20040
Slit  29   20130
Slit  92   21297
Slit  95   21332
Slit  98   21388
Slit 102   21509
```

The result is a map file called **CSIfld5.map**, which contains the coefficients of the polynomials describing the dispersion, the slit's position on the image, and the line curvature. You can disregard messages saying "Unknown keyword ... ignored" because they just indicate that certain keywords in the SMF file are not read in. Notice that several slits were not mapped because they straddled a chip boundary. None of these slits will be included in any file output by **spectral-lines**, **spectral-map**, etc. because these slits will likely be difficult to map or reduce.

6.2 Fine-Tuning the Spectral Map

Now that we have an initial map file, we can run **adjust-map** to fine-tune our solutions using a larger set of lamp features. First, we edit the parameter file:

```
editpar adjust-map
./adjust-map.par

type q to end

SEARCH_HEIGHT           10       half height of search range
SEARCH_WIDTH            10       half-width of search range
SLIT_WIDTH               5       slit width
ORD_DISP                 3       order of dispersion error fit
ORD_SAG                  1       order of y-distortion error fit
ORD_TILT                 1       order of tilt error fit
FIT_TILT                 y       fit slit tilt?
MINLINES                 5       minimum # of lines for tweak
LINELIST        HeNeAr_short.dat       line list file
EDGE                     2       slit edge to exclude
SIGLIMIT          8.000000       line rejection threshold in dispersion
ITERATIONS               5       number of iterations
OUTLIER-WT               y       underweight outlying point?
NCLUMP                   6       number of points in clumps for underweighting
HISTOGRAM                y       use histogram to reject lines outside
 search region?
```

Notice that I used the *HeNeAr_short.dat* line list so that fewer lines are included in the error fits to prevent confusion. Another option would be to use the full *HeNeAr.dat* line list to attempt to get more accurate wavelength solutions, but setting SIGLIMIT to something lower may be wise if you choose to do so.

The **adjust-map** program outputs three files:

file	description
ift0012.map	new, adjusted map file
ift0012.rms	standard deviations in each error computation
ift0012.ps	plots of computed errors and fits

The diagnostic plots in *ift0012.ps* are very useful for checking the spread of the errors and the quality of the fits. You can use these plots to see whether you need to use a different order for fitting the errors (see Figure 5). The blue dots represent the errors that were factored in the fits, whereas the red dots indicate which errors were excluded, based on the value of SIGLIMIT.

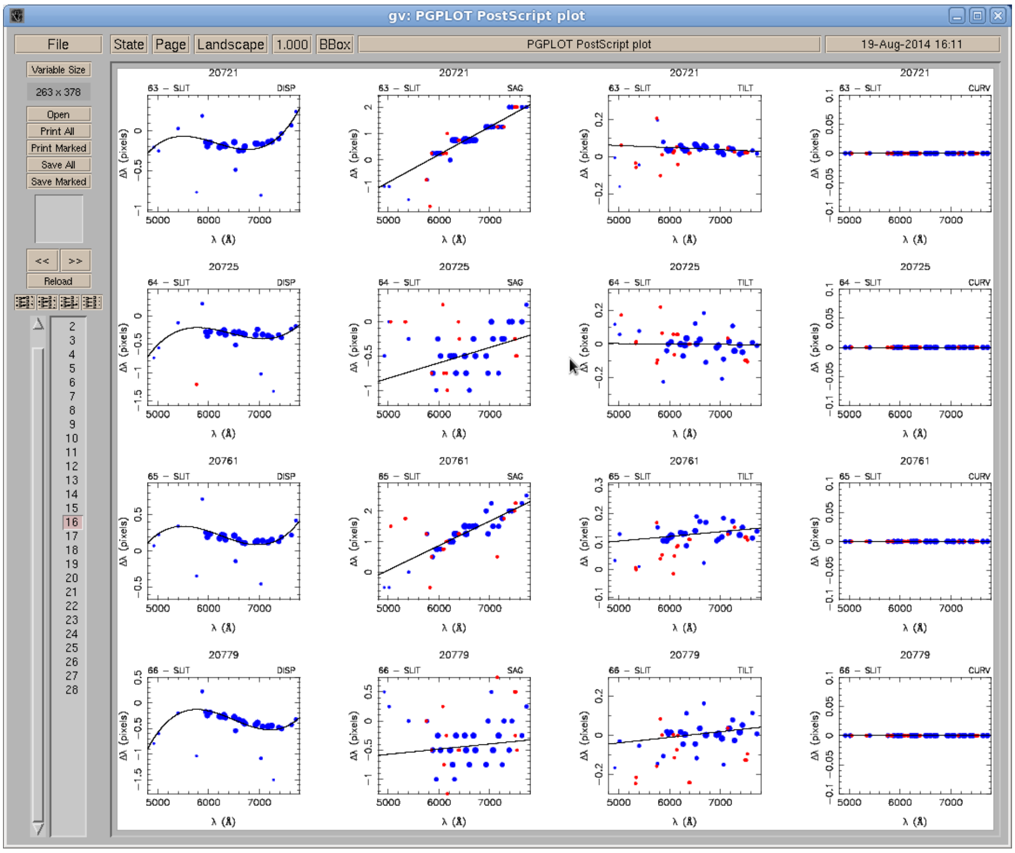
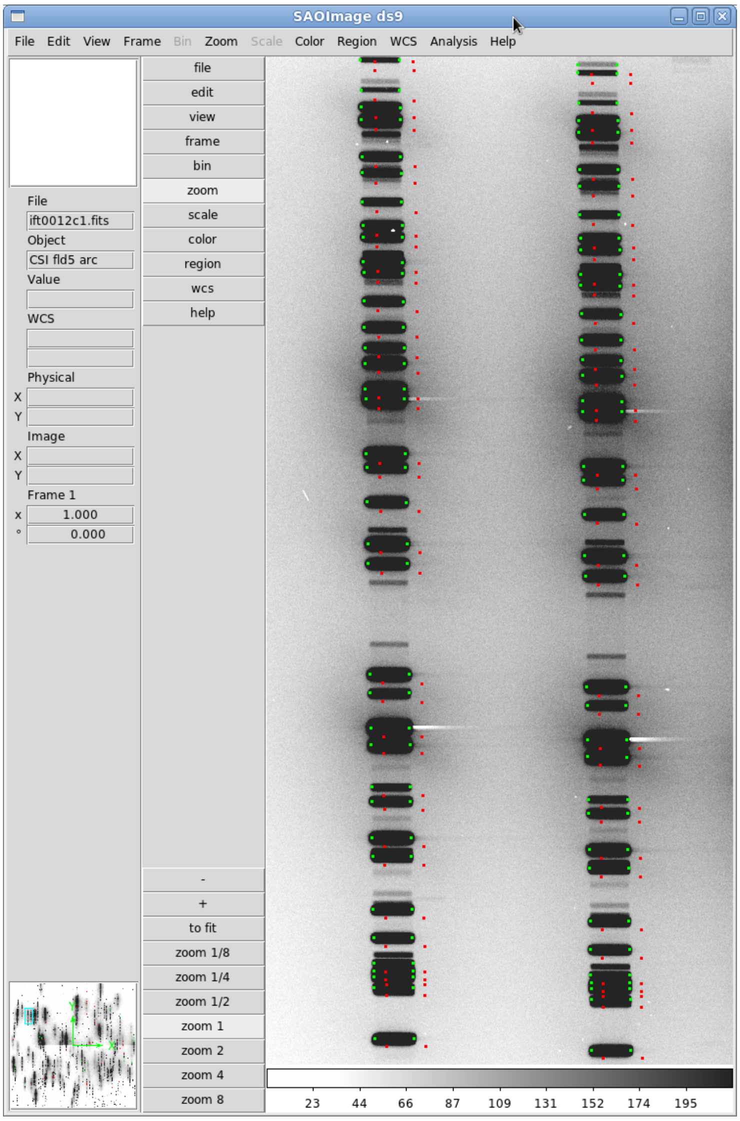


Figure 5: Sample page of plots in the ift0012.ps file output by adjust-map.

You can also check the quality of the map by generating a file with the positions of the lamp features, this time using **spectral-map**. For example, we can run the following command to use the map file we just produced:

```
spectral-map -m ift0012 -l HeNeAr.dat -b 1 -e
```

Then, we could overplot the coordinates in the resulting *ift0012.xy* file onto the lamp mosaic in ds9 using the **display8** and **tvmark** routines in IRAF. Your spectral map at this point should be quite good. Figure 6 shows how well the positions in *ift0012.xy* line up with the actual features.



File: ift0012c1.fits
Object: CSI fld5 arc
Frame 1
x 1.000
° 0.000

Figure 6: Same section of lamp mosaic displayed in ds9 as in Figure 2, but with newly mapped positions marked with green points.

7 Reducing the Spectra

The main steps for reducing the spectra, along with the COSMOS routines used, are as follows:

1. Create a spectroscopic flat field (Sflats)
2. Flat-field and bias-subtract science exposures (biasflat)
3. Subtract the sky from science exposures (subsky)
4. Extract the spectra (extract-2dspec)

7.1 Creating Spectroscopic Flat Fields

First, we edit the **Sflats** parameter file to make sure that we use the proper values for the wavelength range, noise, and gain.

```
editpar Sflats
./Sflats.par

type q to end

minlambda           4800.000000       min wavelength of extraction
maxlambda           7800.000000       max wavelength of extraction
telscale              2.899000       telescope scale sec per mm
bias_file                 none       bias level file
siglimit              4.000000       CR rejection threshold in sigmas
noise                 4.100000       CCD read noise
gain                  0.470000       CCD gain
order                        5       order of flat intensity fit
fit_mode                median       method of spectrum fitting
med_box                    100       size of sky fit median box
shuffled                     0       number of pixels to shuffle data
```

Then we can run the **Sflats** command below to generate a normalized flat-field frame, called *ift0011_flat_cX.fits*, where X goes from 1 through 8. Figure 7 shows a section of the resulting processed flat-field frame.

```
Sflats -m ift0012 -f ift0011
```

7.2 Flat-field the Science Exposures

Before running **biasflat**, make sure the proper parameters are set in the parameter file:

```
editpar biasflat
./biasflat.par

type q to end
```

```
dewar                          E2V       dewar definition file
speed                         slow       readout speed
flat_file                     none       flat field frame
bias_file                     none       bias frame
```

After the parameter file is set, we can run the **biasflat** command to flat-field and subtract the bias from the science exposures:

```
biasflat -f ift0011_flat ift0009
biasflat -f ift0011_flat ift0010
```

These commands generate *ift0009_f_cX.fits* and *ift0010_f_cX.fits*, respectively. See Figure 8 for a section of one of the flat-fielded and bias-subtracted science exposures. Now we can move on to sky-subtraction.

7.3 Run Sky-Subtraction on the Science Exposures

As usual, we take a look at the parameter file first:

```
editpar subsky
./subsky.par

type q to end

minlambda           4800.000000       min wavelength of extraction
maxlambda           7800.000000       max wavelength of extraction
exclude                       5      half width of exclusion region
deltaknot              1.000000      dispersion knot interval, in pixels
splineorder                   3      order of dispersion spline fit to sky
2d_spline                     n      Perform 2-d spline fit to spectra?
medbox                       15      boxsize for running median CR rejection
siglimit               8.000000      CR rejection threshold in sigmas
diag_0              5520.000000       lower limit of diagnostic range
diag_1              5620.000000       upper limit of diagnostic range
edge                          1      excluded edge of spectrum
edge2                         1      excluded edge of other side
objshift               0.000000      number of pixels to shift object position
noise                  4.100000      CCD read noise
gain                   0.470000      CCD gain
```

Since these are relatively short slits, we do not need to fit a 2-D spline to the spectra. The parameters *deltaknot*, *medbox*, *exclude*, and *edge/edge2* are important for getting an optimal sky fit. Once these look fine, we can run **subsky** on the science frames.

```
subsky -m ift0012 -f ift0009_f
subsky -m ift0012 -f ift0010_f
```

These commands generate *ift0009_s_cX.fits* and *ift0010_s_cX.fits*, respectively. Figure 9 shows a section of one of the sky-subtracted frames.

7.4 Extracting 2-D spectra

Before running **extract-2dspec**, we make sure the parameters are set properly:

```
editpar extract-2dspec
./extract-2dspec.par

type q to end

deltalam               2.037000       wavelength step
deltaslit              0.200000       slit postion interval in arcseconds
telscale               2.899000       telescope scale sec per mm
search                        0       half width of search region
trace                      none       spectrum trace method
trace_order                   3       order of fit to spectrum trace
align                      none       alignment method
obj_frac               0.300000       fraction of galaxies used for alignment
init_off               0.000000       initial object offset guess, in pixels
minlambda            4800.000000       min wavelength of extraction
maxlambda            7800.000000       max wavelength of extraction
sub_ns                        n       subtract shuffled region of n&s data?
sampling                 linear       linear or log sampling
edge                          0       excluded edge of spectrum
use_holes                     y       extract alignment stars?
```

Then we run the commands to generate the *ift0009_2spec.fits* and *ift0010_2spec.fits*, respectively.

```
extract-2dspec -m ift0012 -f ift0009_s
extract-2dspec -m ift0012 -f ift0010_s
```

Finally, we can use **sumspec** to stack the spectra and do some cosmic ray cleaning. The following command will generate *CSIfld5_2spec.fits*:

```
sumspec -o CSIfld5_2spec ift0009 ift0010
```

8 Viewing the Spectra

Both **extract-2dspec** and **sumspec** output the spectra in extensions format. Each extension contains the extracted spectra in the first frame and calculated errors in the second frame.

You can view and analyze the spectra in IRAF or other software, but if you have the Perl modules installed and would like to use **viewspectra**, first look at the parameter file and edit the parameters how you like:

```
editpar viewspectra
./viewspectra.par

type q to end

smooth                       10       size of smoothing box in angstroms
```

```
linelist              linelist       galaxy line list
nod                          0       telescope nod distance, in pixels
shuffle                      n       shuffled spectra?
hlfwdth                      2       halfheight of spectrum extraction box
lamint              100.000000       halfwidth of line exam window
lambda0             4800.000000       minimum wavelength to display
lambda1             7800.000000       maximum wavelength to display
```

And then simply type **viewspectra** to open the window. In the window, click on **File** and then **Open** to load a *2spec* file. To load *CSIfld5_2spec.fits*, we type the following at the end of the path in the dialog box:

```
CSIfld5_2spec
```

Figure 10 shows the spectra for one of the slits, as viewed in **viewspectra**.

9 Epilogue

In this run through the main COSMOS programs, I mostly use the default values for many of the parameter files to perform a simple reduction and demonstrate the workflow and usage of the programs. Different parameters could be used during the calibration steps to further improve the map, and fine-tuning the parameters in **subsky** could improve the sky-subtraction.

For more information on using COSMOS to reduce other types of IMACS or LDSS3 data, please visit the online pages at:

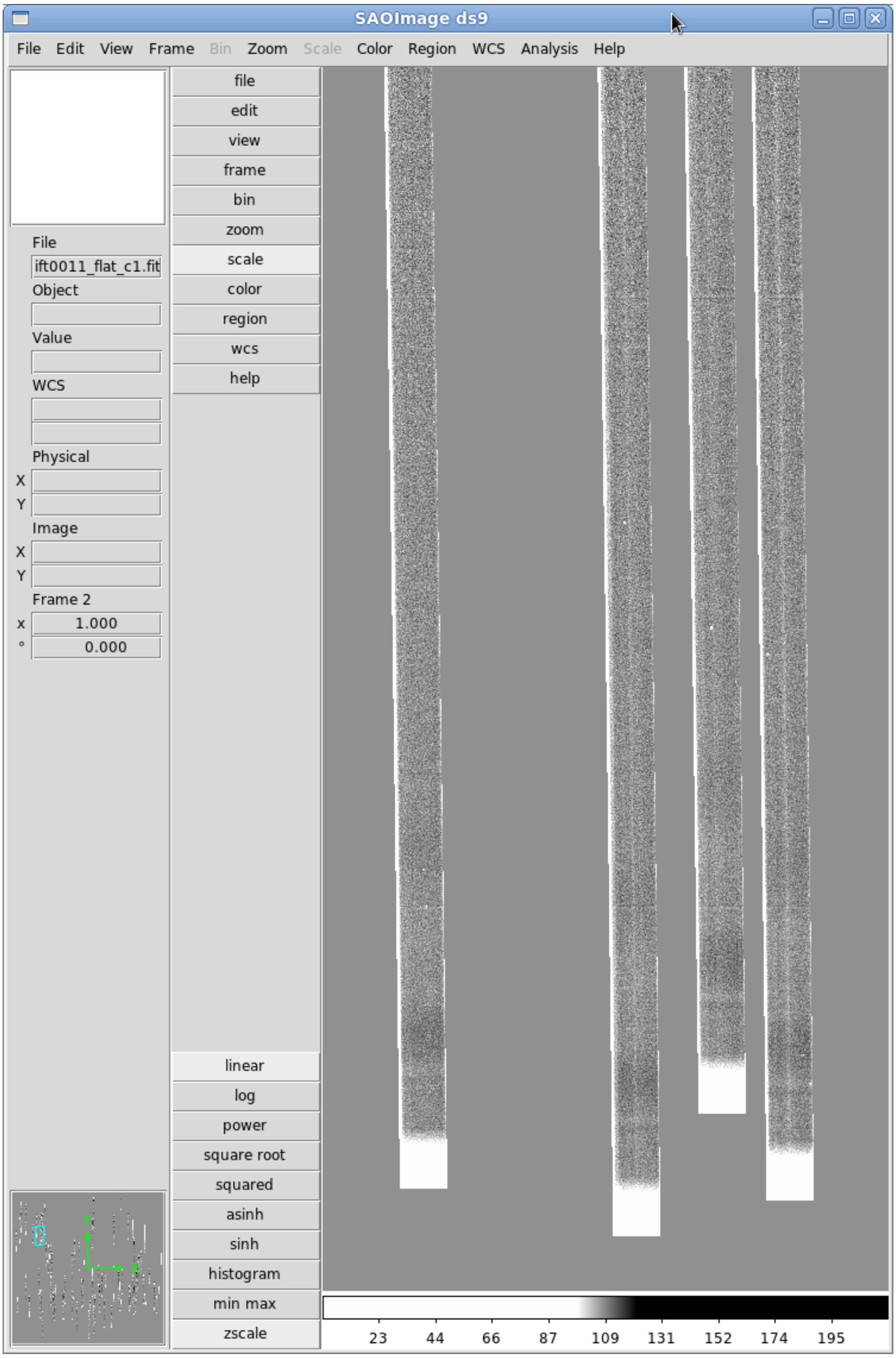
```
http://code.obs.carnegiescience.edu/cosmos
```

The documentation is also available locally at:

```
$COSMOS_HOME/docs/index.html
```

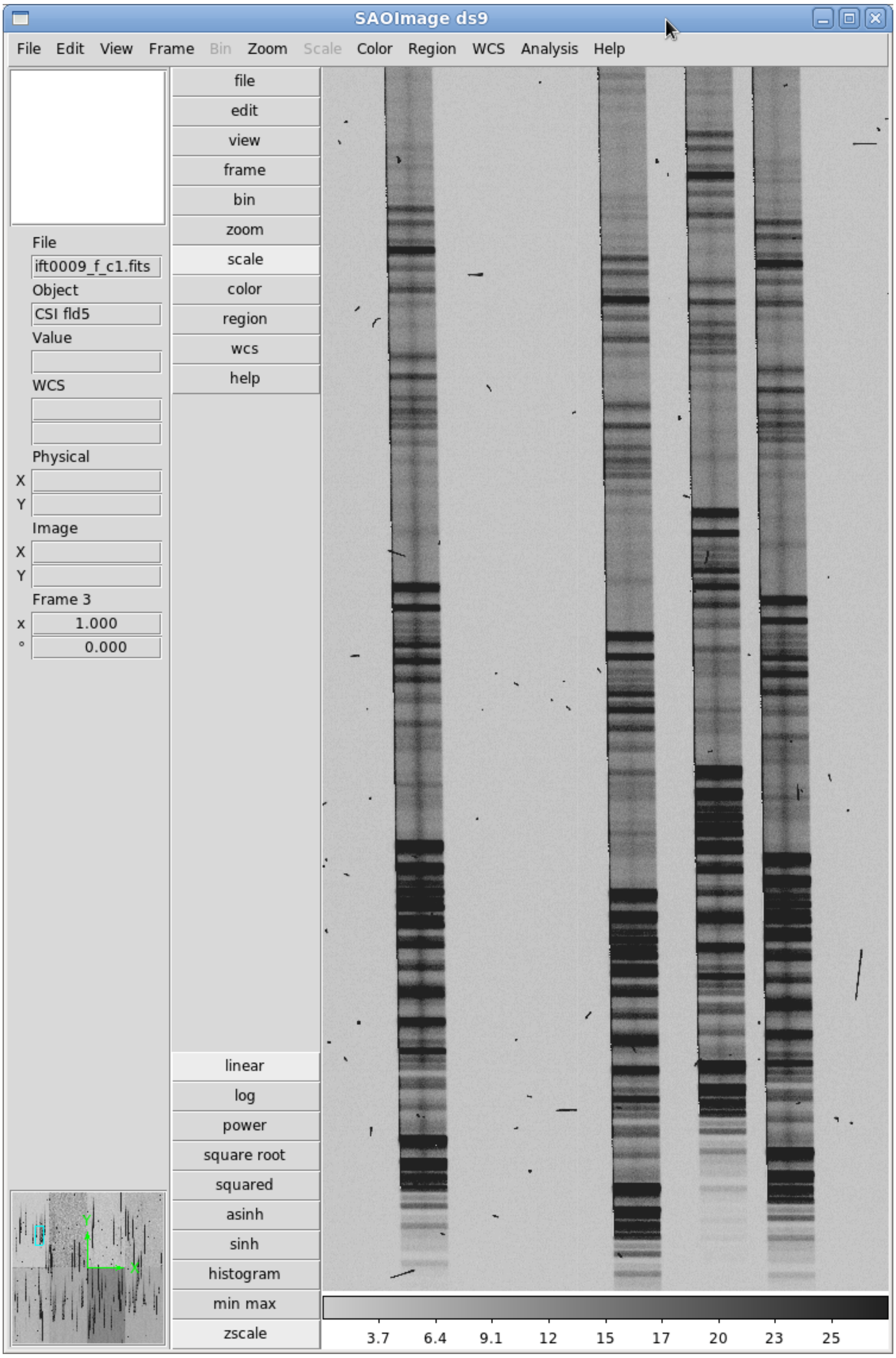
If you have any questions regarding COSMOS that the documentation does not address, feel free to e-mail the COSMOS team at:

```
cosmos@obs.carnegiescience.edu
```



File
ift0011_flat_c1.fit
Object
Value
WCS
Physical
X
Y
Image
X
Y
Frame 2
x 1.000
° 0.000

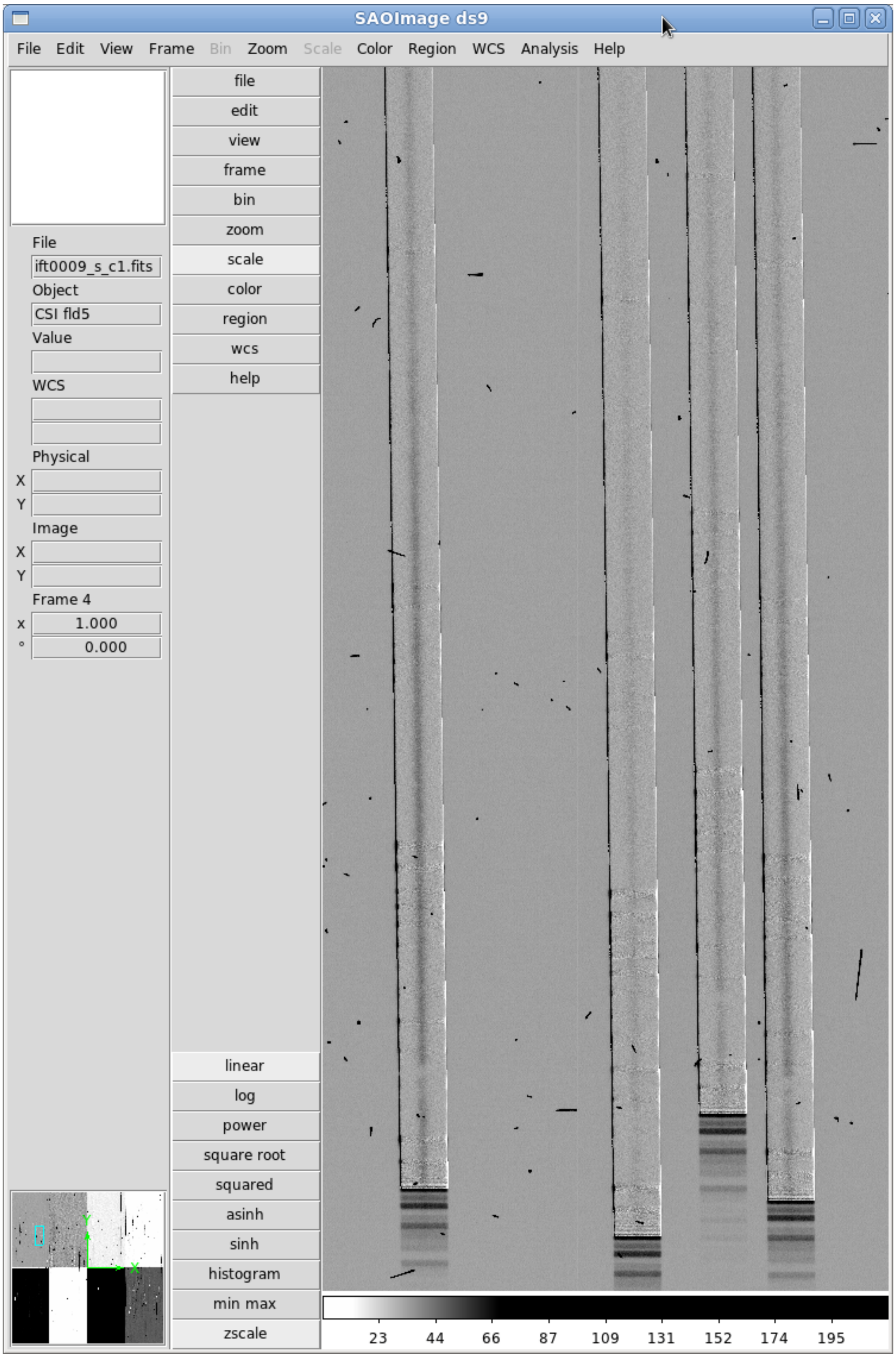
Figure 7: Section of spectroscopic flat-field frame generated by Sflats.



SAOImage ds9

File
ift0009_f_c1.fits
Object
CSI fld5
Value
WCS
Physical
X
Y
Image
X
Y
Frame 3
x 1.000
° 0.000

Figure 8: Section of flat-fielded science exposure generated by biasflat.



File
ift0009_s_c1.fits
Object
CSI fld5
Value
WCS
Physical
X
Y
Image
X
Y
Frame 4
x 1.000
° 0.000

Figure 9: Section of sky-subtracted science exposure generated by subsky.

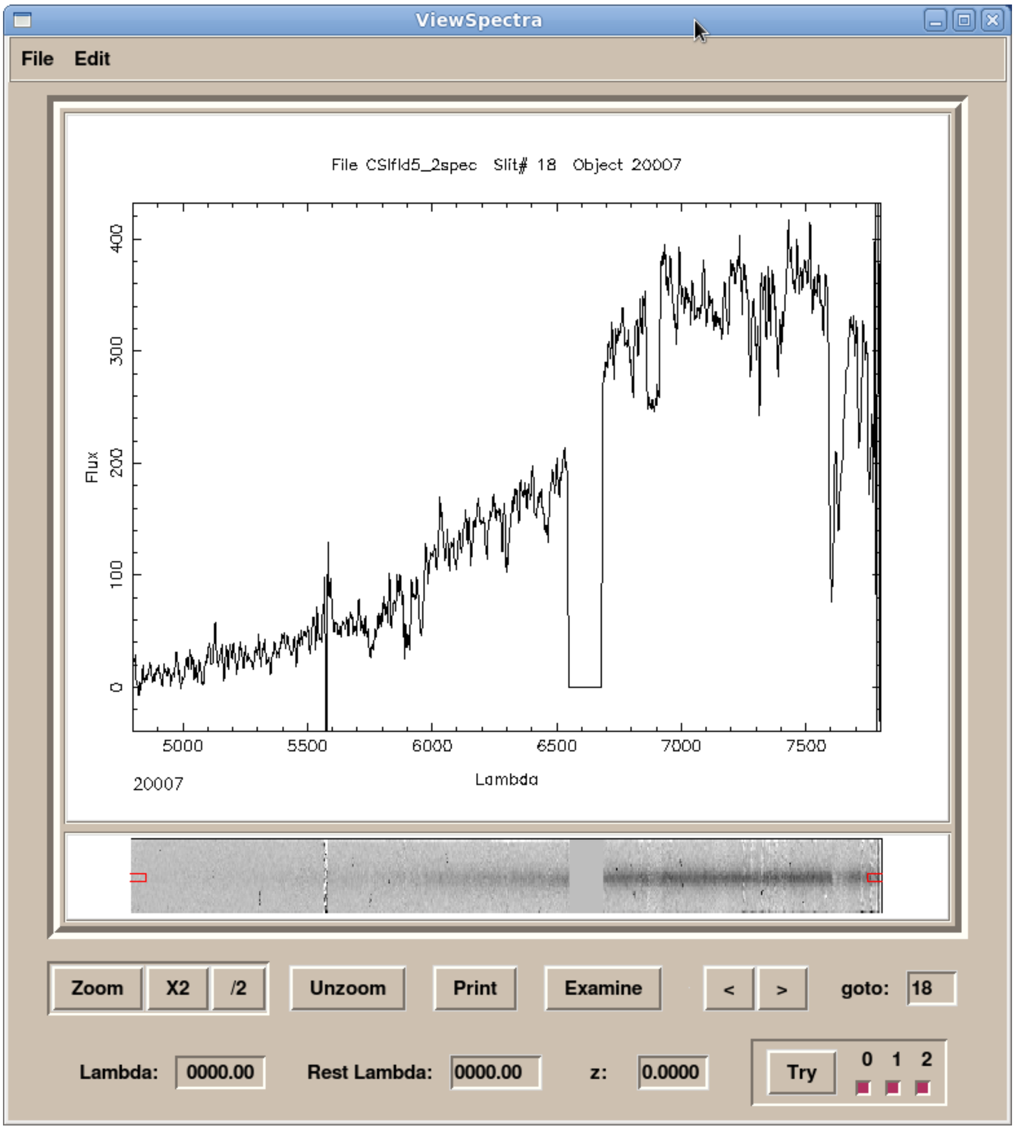


Figure 10: 1-D and 2-D spectrum for one slit in CSIfld5_2spec.fits, as seen in viewspectra.