To Start

- Start by looking at setup_running_nifspipeline_v4.pdf, which provides instructions for installing and setting up the pipeline, an overview of the directory structure, information about running the pipeline, and notes for the few steps that require user interaction.

- This document provides a general overview of the data reduction procedure, and notes for pipeline steps that the user can chose to do interactively.

General Overview of the Pipeline

Sort files / Make lists

1. Using header information, sort raw data files located in the ‘datadir’ subdirectories into the appropriate ‘reducedir’ subdirectories. Create lists that will be fed into IRAF (e.g., flatlist contains the names of all the flats for a given night and observational setup). The user is asked to check these lists and can make modifications if needed.

General Overview of the Pipeline

Reduce Baseline Calibration Files

- Work with the flats for a given night and observational setup —

1. Using one of the flats, locate the spectra by matching to the mask definition file. Save the offset needed to align to the mask definition file.
2. Update all flats and flat darks with the offset found above. Create variance and data quality extensions.
3. Combine flats using an average, including an avsigclip rejection method. Do the same for the flat darks.
4. Extract slices from each flat, apply approximate wavelength calibration to headers, update data quality extensions. Do the same for the flat darks.
5. Subtract the flat dark from the flat, and generate a normalized flat. Create a bad pixel mask from the dark. Update the data quality extensions.
6. Correct for variation between slices since the previous step normalizes each slice separately to unity.

- Work with the object darks (e.g., the 600 s darks) for a given night and observational setup —

7. Update the object darks with the offset from step #1. Create variance and data quality extensions.
8. Combine the object darks (if more than 1) using an average, including an avsigclip rejection method.

General Overview of the Pipeline

Reduce Baseline Calibration Files (continued)

- Work with the arcs for a given night and observational setup —

9. Update the arcs and the arc darks with the offset from step #1 on page 2. Create variance and data quality extensions.
10. Combine the arc darks using an average, including an avsigclip rejection method.
11. Combine arcs (if there is more than 1 one and the user wants to combine) using an average, including an avsigclip rejection method.
12. Extract slices from the combined (or individual) arc(s), apply approximate wavelength calibration to headers, subtract combined arc dark, apply combined normalized flat.
13. Identify arc lines for groups of 10 rows in each slice of each arc exposure, fit the dispersion function (mapping pixel number to wavelength), and store information in a database/ subdirectory. This information will be used later to wavelength calibrate science observations.

- Work with the ronchi masks for a given night and observational setup —

14. Update the ronchi masks and ronchi darks with the offset from step #1 on page 2. Create variance and data quality extensions.
15. Combine ronchi masks (if more than 1) using an average, including an avsigclip rejection method. Do the same for the ronchi darks.
16. Extract slices from the ronchi mask, apply approximate wavelength calibration to headers, subtract combined ronchi dark, apply combined normalized flat.
17. Identify peaks for groups of 10 columns in each slice of the ronchi mask, fit a function mapping pixel number to peak, and store information in the database/ subdirectory. This information will be used later to spatially rectify science observations.
18. As a check of the above step, the distortion correction is applied to the ronchi mask itself and a cube is generated. The user can view the cube to make sure the rows look straight.

General Overview of the Pipeline

Reduce the (A0 V) Telluric Stars

- For a given night and observational setup —

    1. Update the star exposures and the sky exposures with the offset from step #1 on page 2. Create variance and data quality extensions.
    2. Subtract each star exposure by the neighboring sky exposure.
    3. Extract slices from each star exposure, and apply combined, normalized flat. Update data quality extensions.
    4. Replace bad pixels with values interpolated from fitting neighbors.
    5. Derive the 2D spectral/spatial transformation for each slice and then apply the transformation. The result are slices that are sampled in constant steps of wavelength and arcseconds. Basically a data cube (but not actually assembled into a data cube yet) — will call these products “almost data cubes” below.
    6. Extract a 1D spectrum from each of the star “almost data cubes”.
    7. For flux calibration later on, assemble one of the telluric star “almost data cubes” into an actual data cube.
    8. Median combine the 1D telluric star spectra, update the data quality extension, and save to an output file.
    9. Create a 1D telluric correction spectrum. The star’s absorption line and blackbody shape are determined by fitting Kurucz models of Vega and A-stars to the combined 1D telluric star spectrum over the Brɣ region with pPXF. The 1D combined telluric spectrum is divided by the best-fit pPXF model to derive the final 1D telluric correction spectrum.

General Overview of the Pipeline

Reduce the Galaxy and Sky Exposures

- For a given night and observational setup —

    1. Update the galaxy exposures and the sky exposures with the offset from step #1 on page 2. Create variance and data quality extensions.
    2. Subtract each galaxy exposure by the neighboring sky exposure.
    3. Subtract combined dark from each sky exposure (since we want to also produce sky cubes).
    4. Extract slices from each galaxy exposure, apply approximate wavelength calibration to headers, apply combined normalized flat, and update data quality extensions. Repeat for sky exposures.
    5. For each galaxy exposure, replace bad pixels with values interpolated from fitting neighbors. Repeat for sky exposures.
    6. Derive the 2D spectral/spatial transformation for each slice and then apply the transformation. The result are slices that are sampled in constant steps of wavelength and arcseconds. Basically a data cube (but not actually assembled into a data cube yet) — will call these products “almost data cubes” below. Do this for both the galaxy and sky exposures.
    7. Correct galaxy and sky “almost data cubes” for telluric features.
    8. Create a temporary cube from the galaxy “almost data cubes”. Collapse (sum along the wavelength axis) and copy the resulting image to the ../merged/ directory. Delete the temporary cube.
    9. Copy the galaxy “almost data cubes” to the ../merged/ directory. Do the same for the sky “almost data cubes”, but generate an additional copy. Later on, need the same number of sky “almost data cubes” as galaxy “almost data cubes” to produce a final sky cube just like the final galaxy cube. This assumes a galaxy-sky-galaxy observing sequence.

General Overview of the Pipeline

Merge the Galaxy and Sky Exposures

- For a given observational setup —

    1. Create the wavelength axis for the final merged cube. This is done using the starting/ending wavelength and the change in wavelength per pixel from the first galaxy “almost data cube”. Apply a barycentric or heliocentric correction if the user wants. Even if no correction is applied, all spectra are sampled to the same starting/ending wavelength and $\Delta\lambda$ when creating a single merged cube in step #4 below.
    2. Calculate x and y spatial offsets between the galaxy “almost data cubes” using the images from step #8 on page 5. The spatial offsets between the images (relative to the first image) are determined three ways: (1) finding the location of the pixel with the largest flux, (2) fitting a 2D Gaussian to the image, and (3) cross-correlating the images. The offsets from each method are written to text files.
    3. The default method for determining spatial offsets is method #3. The pipeline checks that method #3 returns similar offsets as methods #1 and #2. If the offsets from method #3 are not similar, the user is notified and asked to select the method to adopt. The user can also provide their own text file with the spatial offsets instead. Fractional pixel offsets are accepted.
    4. Using all of the galaxy “almost data cubes”, produce a single cube with science, variance, and data quality extensions with the wavelength axis from step #1. The headers of the galaxy “almost data cubes” are adjusted, tricking a modified version of IRAF’s nifcube (a wrapper for gemcube) into accounting for the spatial offsets. Input masks for the slices can be provided so that bad edges can be excluded when creating the final data cube.
    5. Repeat step #4 above for the sky “almost data cubes”, using the spatial offsets adopted for the galaxy “almost data cubes” and the wavelength axis from step #1 above.

General Overview of the Pipeline

Reduce the PSF Stars

- For a given night and observational setup —

1. Update the PSF star exposures and the sky exposures with the offset from step #1 on page 2. Create variance and data quality extensions.
2. Subtract each PSF star exposure by the neighboring sky exposure.
3. Extract slices from each PSF star exposure, apply approximate wavelength calibration to headers, apply combined normalized flat, and update data quality extensions.
4. For each PSF star exposure, replace bad pixels with values interpolated from fitting neighbors.
5. Derive the 2D spectral/spatial transformation for each slice and then apply the transformation. The result are slices that are sampled in constant steps of wavelength and arcseconds. Basically a data cube (but not actually assembled into a data cube yet) — will call these products “almost data cubes” below.
6. Correct PSF star “almost data cubes” for telluric features.
7. Create a temporary cube from the PSF star “almost data cubes”. Collapse (sum along the wavelength axis) and copy the resulting image to the ../../merged/ directory. Delete the temporary cube.
8. Copy the PSF star “almost data cubes” to the ../../merged/ directory.

General Overview of the Pipeline

Merge the PSF Star Exposures

- PSF star “almost data cubes” are combined together to form a single data cube for (1) each night and observational setup, and (2) over all nights for a given observational setup.
- The same steps used to produce a merged galaxy cube are repeated. Briefly, the steps are:
    1. Create the wavelength axis for the final cube using the first PSF star “almost data cube” as reference. Apply a barycentric or heliocentric correction if the user wants.
    2. Calculate x and y spatial offsets between the PSF star “almost data cubes” using the images from step #7 on page 7.
    3. Get the offset method to adopt; the default method is cross-correlation.
    4. Using all the PSF star “almost data cubes”, produce a single at a cube with science, variance, and data quality extensions with the wavelength axis from step #1.

Flux Calibrate the Merged Galaxy Cube and Sky Cube

- For a given galaxy and observational setup —
    1. Take the data cube of an A0 V telluric star (generated previously in step #7 of page 4) and apply a telluric correction, leaving just the spectra of the A0 V star.
    2. Determine the multiplicative conservation factor to go from counts/s to the known flux density (ergs/s/cm^2/Å) at the isophotal wavelength for the A0 V star.
    3. Apply this conversion factor to the spectrum from each spaxel of the merged galaxy cube and the merged sky cube.

General Overview of the Pipeline

Measure the Line Spread Function (LSF)

- For a given galaxy and observational setup —
  1. Go through each spaxel of the merged flux calibrated sky cube, and fit Gaussians to 13 OH lines between 2.0 and 2.25 microns. Calculate the median FWHM (in Å) of the lines at each spatial location, thus providing the LSF over the NIFS field-of-view. This LSF map is saved as a fits file.

(Optional) Interactive Components of the Pipeline

- The pipeline requires a little bit (or a lot, if the user wants) of interaction. The parts that require interaction are summarized in setup_running_nifspipeline_v4.pdf. What follows is an overview of the optional interactive steps.

- At the top of nifs_main_LP.py, the user has the option to run various steps of the pipeline interactively, by setting 'flinter_arc', 'flinter_sdist', 'flinter_nsfitcoords', 'flinter_extract', and 'flinter_telluric' to 'yes' (to do so interactively) or to 'no'.
    - 'flinter_arc' corresponds to step #13 of "Reduce Baseline Calibration Files" (on page 3) — the identification of arc lines.
    - 'flinter_sdist' corresponds to step #17 of "Reduce Baseline Calibration Files" (on page 3) — the identification of peaks in the ronchi mask.
    - 'flinter_nsfitcoords' corresponds to step #5 of "Reduce the (A0 V) Telluric Stars" (on page 4), step # 6 of "Reduce the Galaxy and Sky Exposures" (on page 5), and step #5 of "Reduce the PSF Stars" (on page 7) — the determination of the 2D spectral/spatial transformation.
    - 'flinter_extract' corresponds to step #6 of "Reduce the (A0 V) Telluric Stars" (on page 4) — the extraction of a 1D spectrum from a temporary cube of the telluric star.
    - 'flinter_telluric' corresponds to step #7 of "Reduce the Galaxy and Sky Exposures" (on page 5) and step #6 of "Reduce the PSF Stars" (on page 7) — the application of the telluric correction.

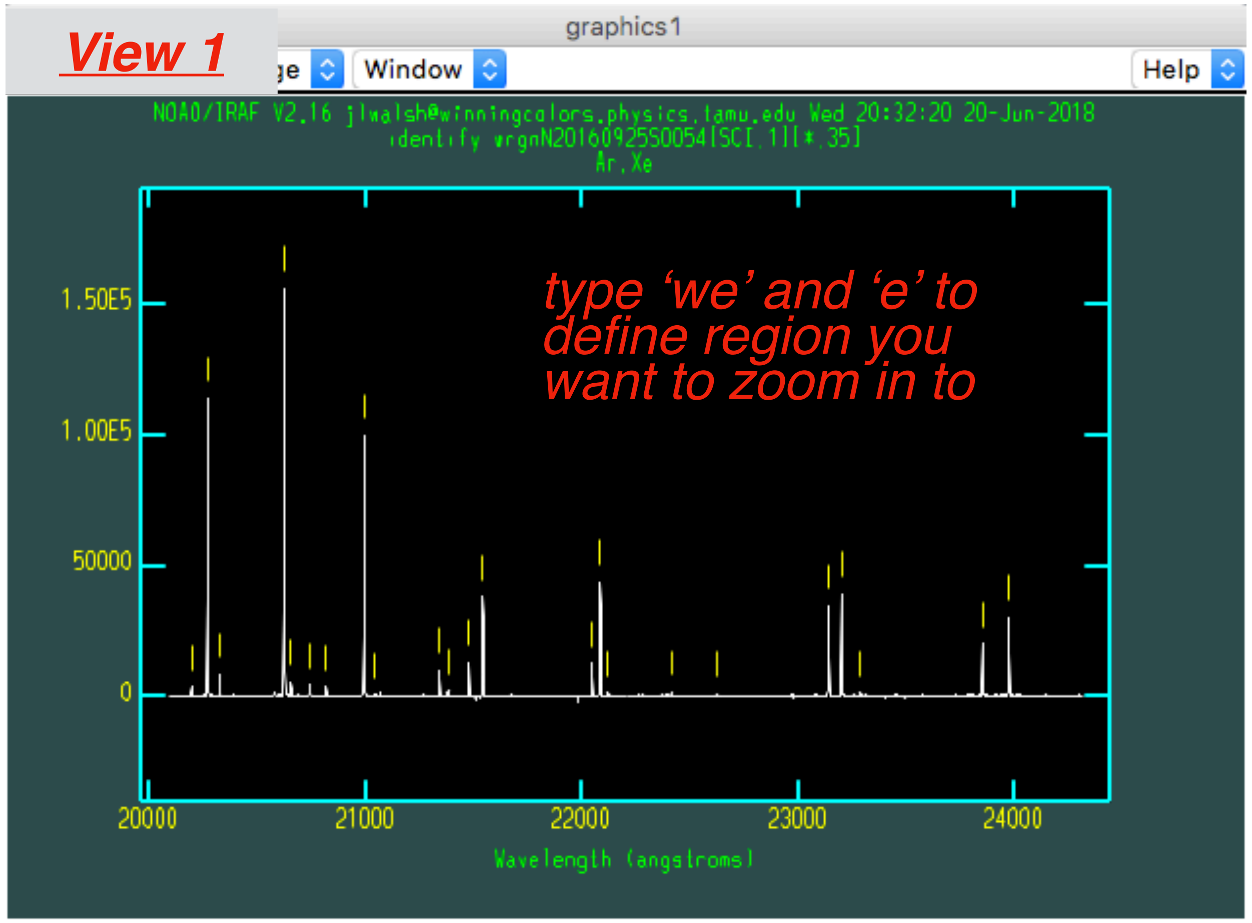
- In the pages that follow, the terminal window in which the pipeline was started will be referred to as the IRAF window. Additional windows that the pipeline automatically opens will be referred to as interactive windows.

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’

- The IRAF window will ask if you want to examine the identifications interactively. In the IRAF window, type ‘yes’, or hit return if this is the default response shown in parentheses.

- An interactive window will appear (see view #1). The plot shown is the sum of 10 rows (spectra) at the middle spatial location of one slice of one arc exposure. The arc lines are automatically identified (indicated by yellow marks above the lines) by matching to an input line list, using IRAF’s AUTOIDENTIFY task.

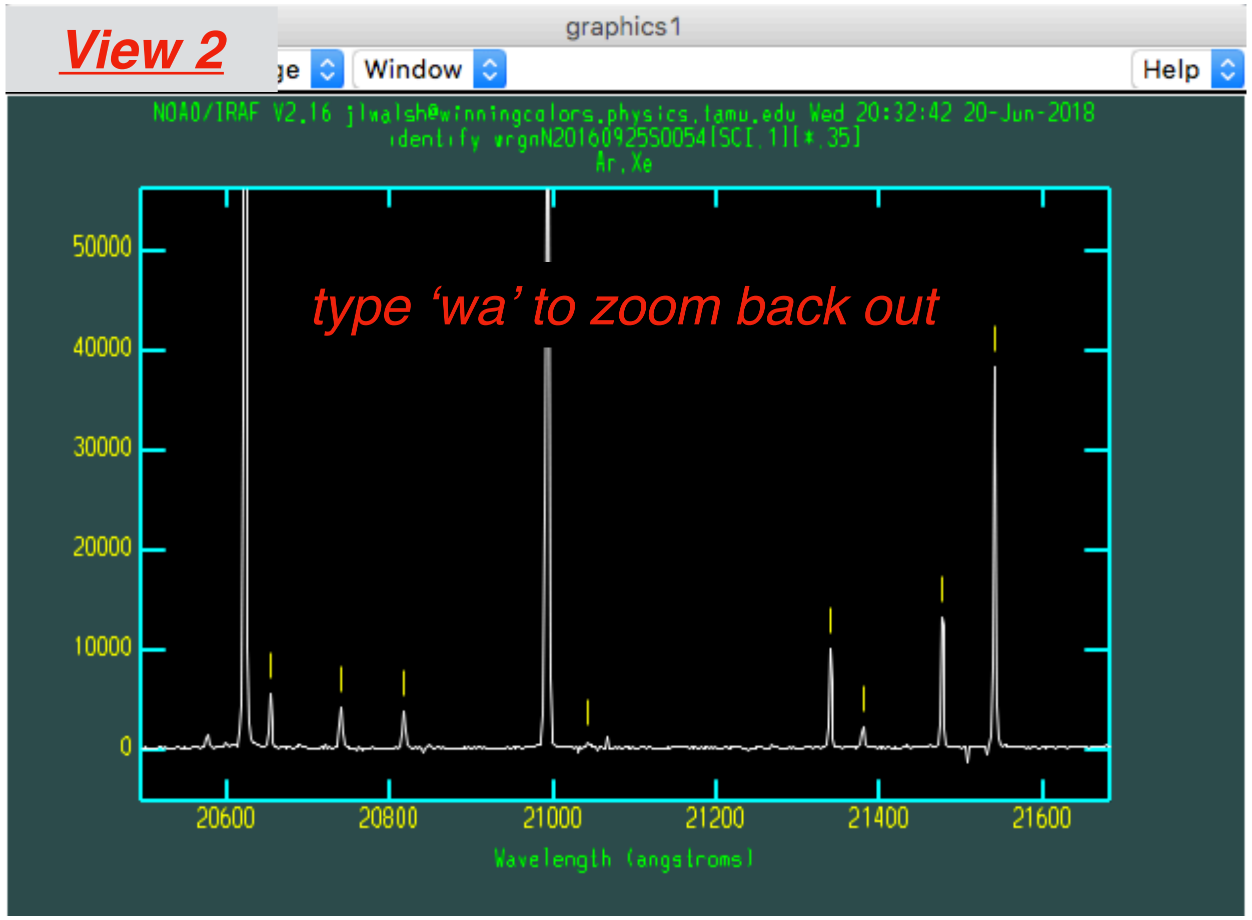
- Click on the interactive window to make active. Can zoom into a region by placing cursor on lower, left of a rectangular region and typing ‘we’ and then your cursor on the top, right of a rectangular region and typing ‘e’. You’ll see a new plot, like view #2. You can zoom back out by typing ‘wa’ in the interactive window to go back to view #1.



View 1

```
NOAO/IRAF V2.16 jlwalsh@winningcalors.physics.tamu.edu Wed 20:32:20 20-Jun-2018
identify vrgnN20160925S0054[SCI,1][*,35]
Ar,Xe
```

type ‘we’ and ‘e’ to define region you want to zoom in to



View 2

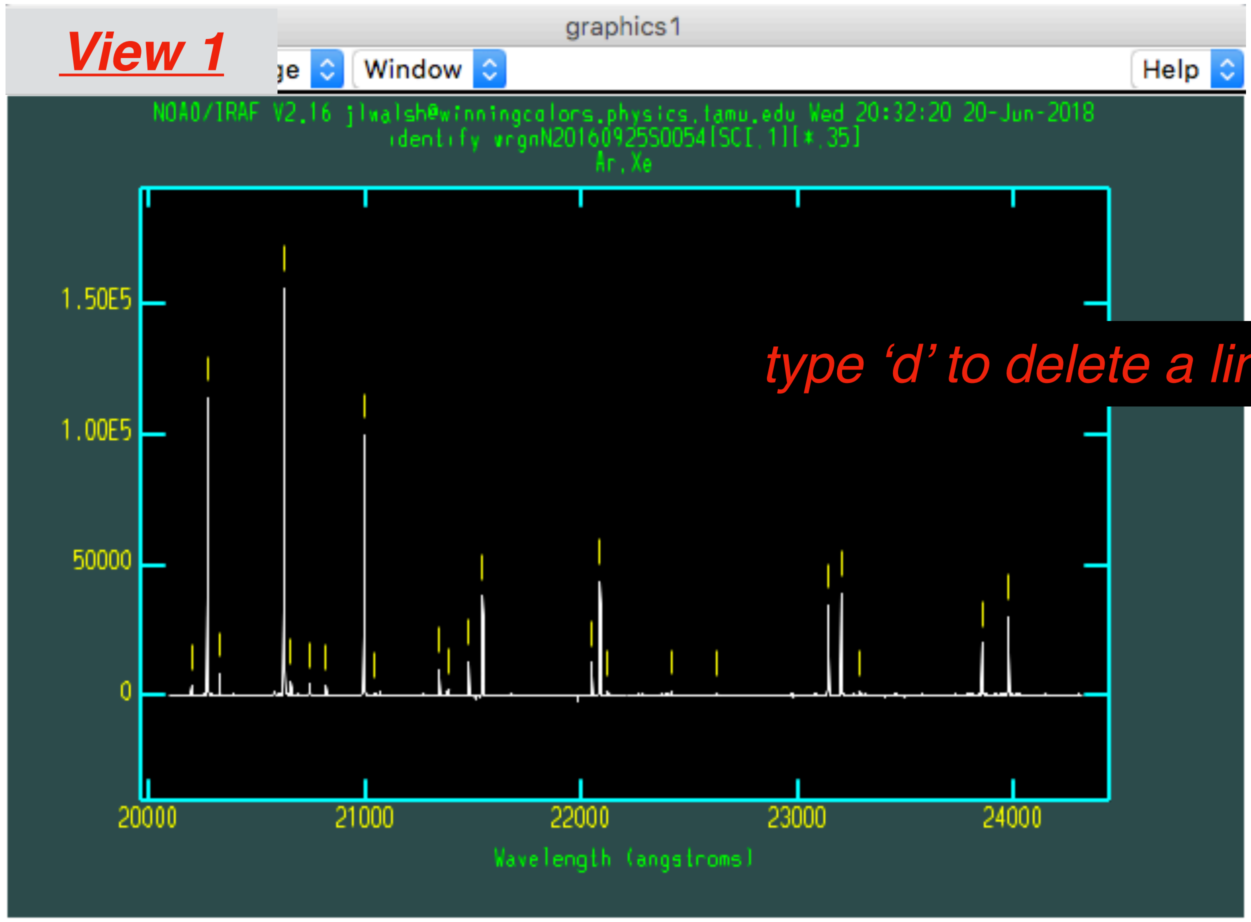
```
NOAO/IRAF V2.16 jlwalsh@winningcalors.physics.tamu.edu Wed 20:32:42 20-Jun-2018
identify vrgnN20160925S0054[SCI,1][*,35]
Ar,Xe
```

type ‘wa’ to zoom back out

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’

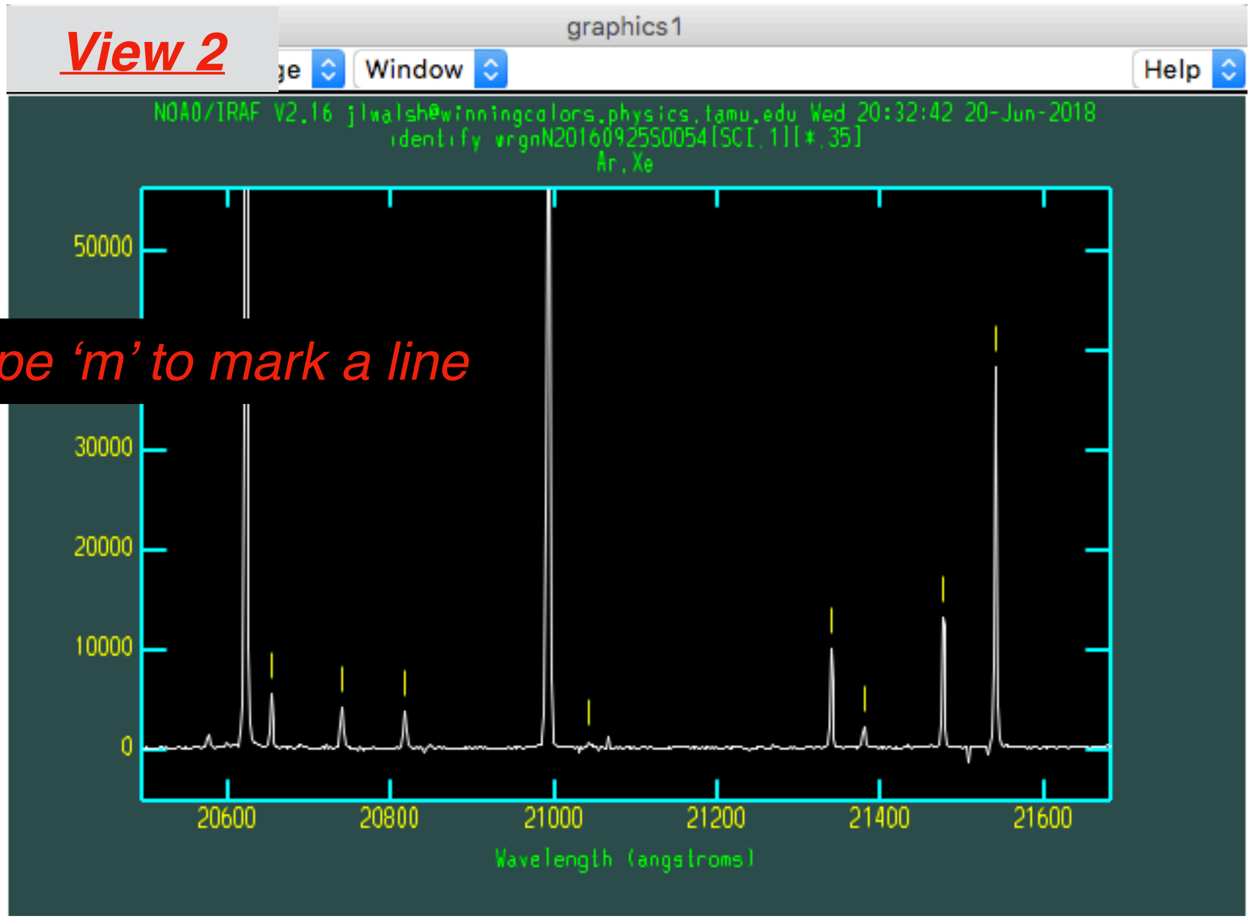
- If you want to remove an identified line (e.g., if line is weak or double peaked), in the interactive window place cursor at the top of the line, and type ‘d’. The yellow mark above the line will disappear. If you want to mark a line, place cursor at the top of the line and type ‘m’. At the bottom of the interactive window in parentheses will be the line’s wavelength based on comparing to the line list. By clicking at the bottom of the interactive window, you can hit return to accept, or you can manually type in the wavelength (in Å). A yellow mark will appear above the line.

- The identified lines are used to fit a dispersion function, mapping pixel number to wavelength. Type ‘f’ in the interactive window to see the fit.



View 1

```
NOAO/IRAF V2.16 jlwalsh@winningcalors.physics.tamu.edu Wed 20:32:20 20-Jun-2018
identify wrgnN20160925S0054[SCI,1][*,35]
Ar,Xe
```

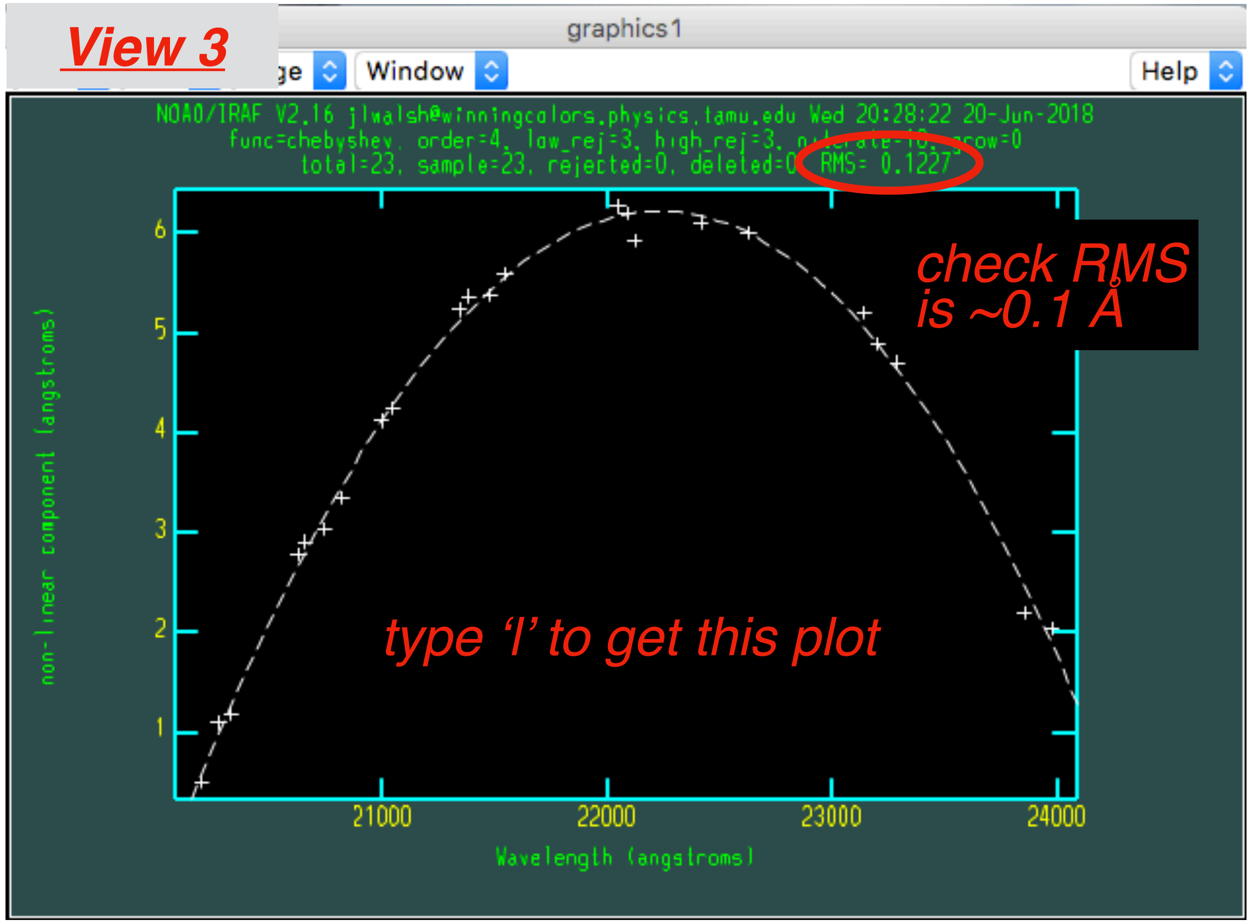


View 2

```
NOAO/IRAF V2.16 jlwalsh@winningcalors.physics.tamu.edu Wed 20:32:42 20-Jun-2018
identify wrgnN20160925S0054[SCI,1][*,35]
Ar,Xe
```

type ‘d’ to delete a line, type ‘m’ to mark a line

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’



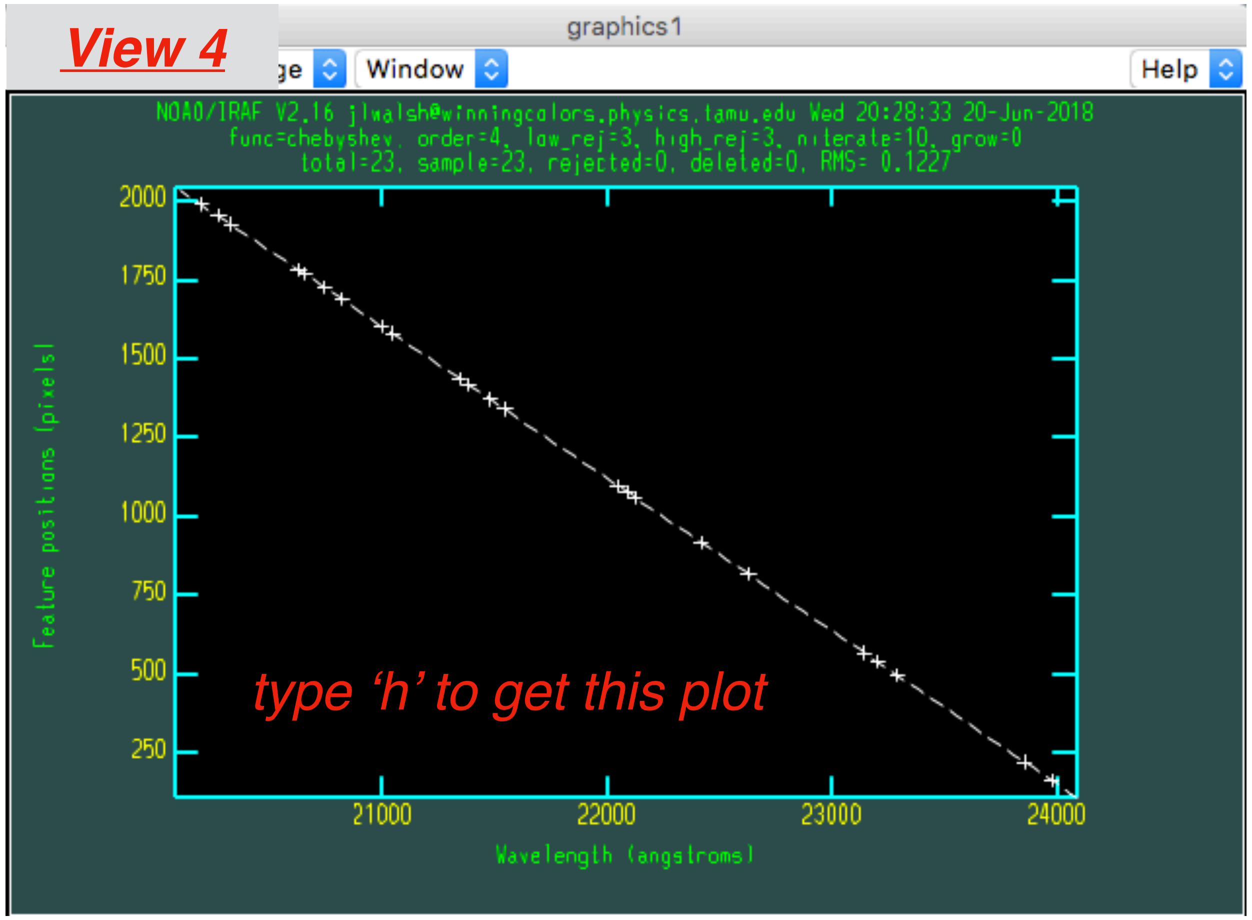
View 3

```
NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 20:28:22 20-Jun-2018
func=chebyshev, order=4, low_rej=3, high_rej=3, niterate=10, grow=0
total=23, sample=23, rejected=0, deleted=0, RMS= 0.1227
```

check RMS is ~0.1 Å

type ‘l’ to get this plot

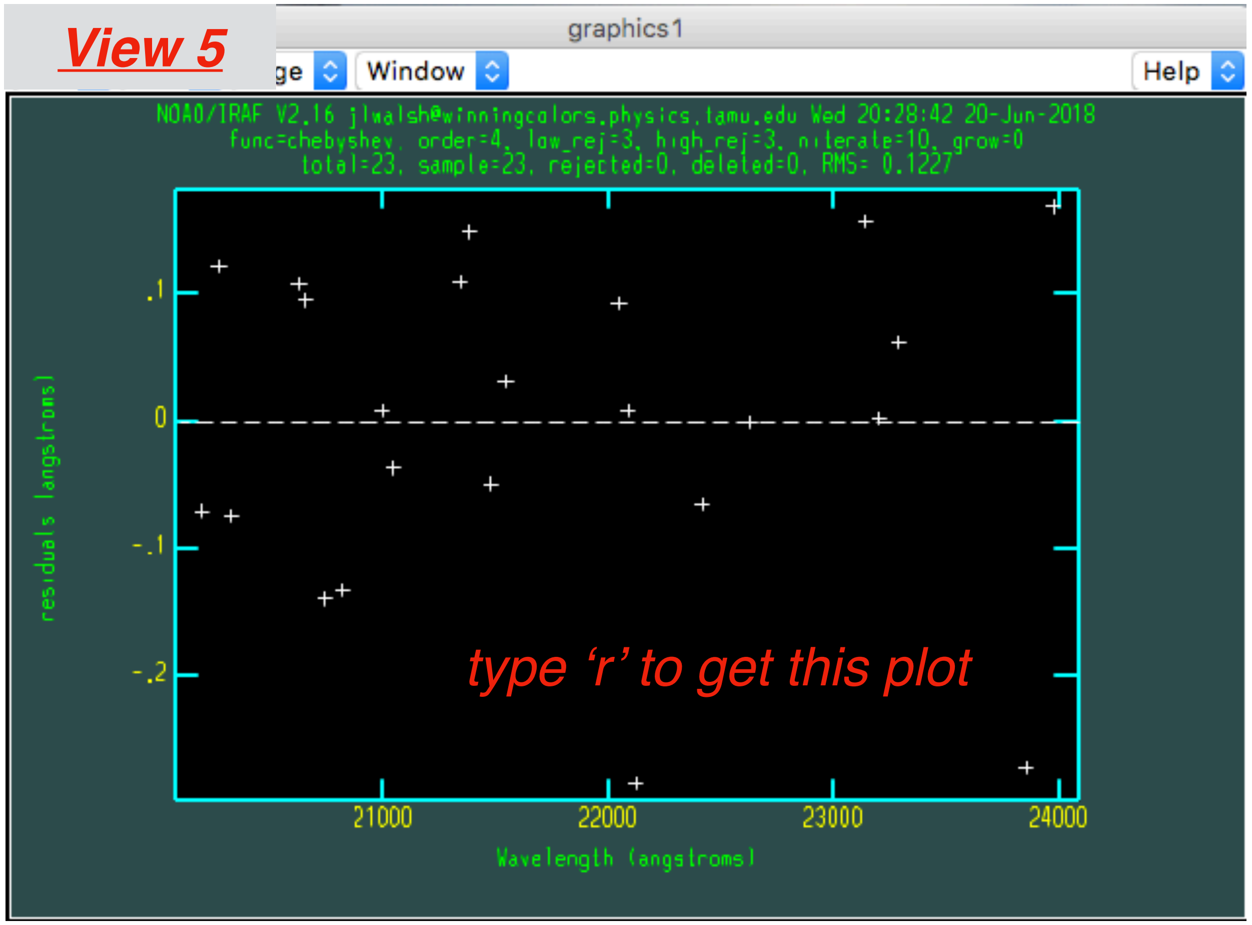
- The fit can be displayed a number of ways. View #3 is what will appear after typing ‘f’ in the interactive window while view #1 or #2 (page 12) is seen. View #3 shows the non-linear component. Typing ‘h’ in the interactive window will bring up view #4, which shows the pixel number versus wavelength. Typing ‘j’ in the interactive window will bring up view #5, which shows the residuals versus wavelength. Typing ‘l’ will bring up view #3 again. Check that the fit looks reasonable, and that the RMS is ~0.1 Å.



View 4

```
NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 20:28:33 20-Jun-2018
func=chebyshev, order=4, low_rej=3, high_rej=3, niterate=10, grow=0
total=23, sample=23, rejected=0, deleted=0, RMS= 0.1227
```

type ‘h’ to get this plot



View 5

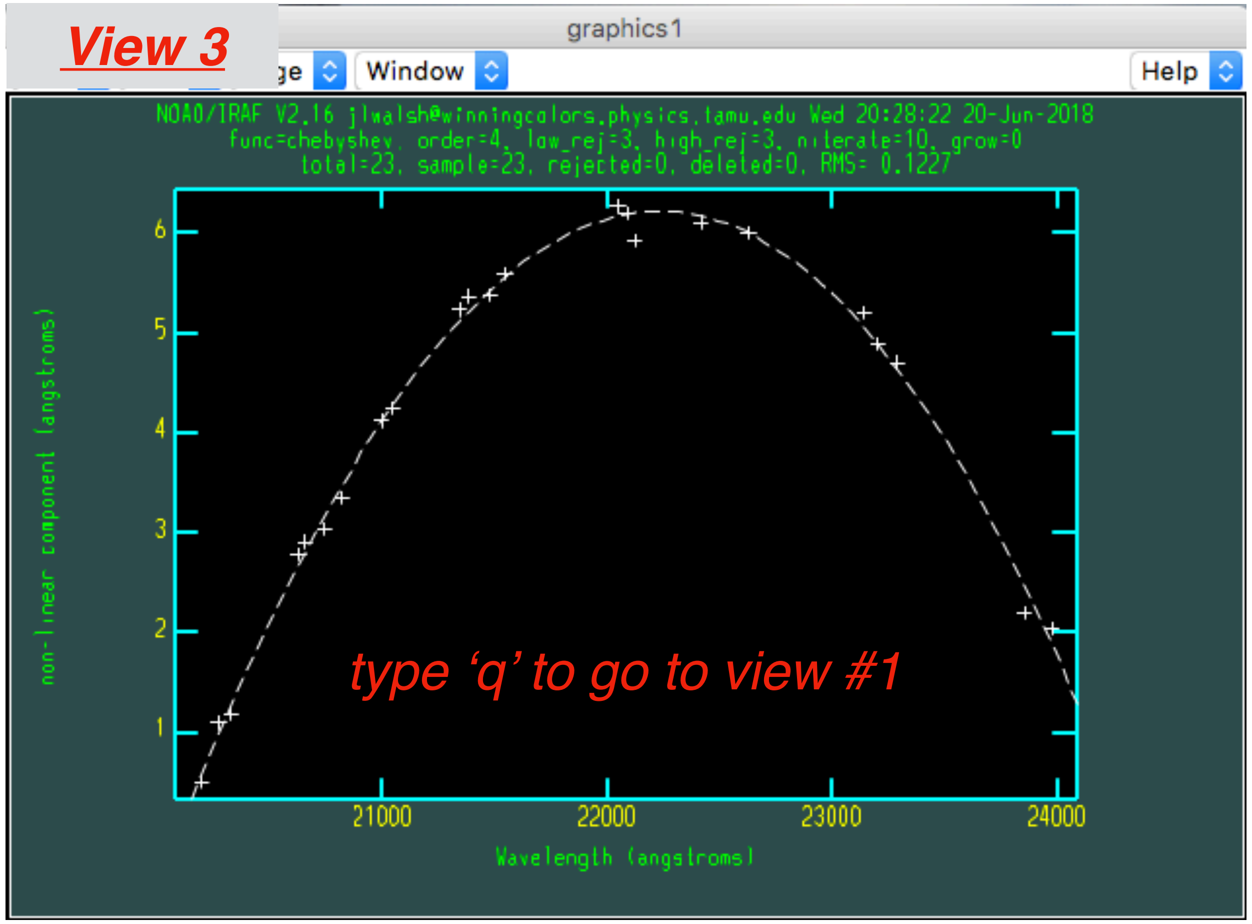
```
NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 20:28:42 20-Jun-2018
func=chebyshev, order=4, low_rej=3, high_rej=3, niterate=10, grow=0
total=23, sample=23, rejected=0, deleted=0, RMS= 0.1227
```

type ‘r’ to get this plot

(Optional) Interactive Components of the Pipeline: flinter_arc = 'yes'

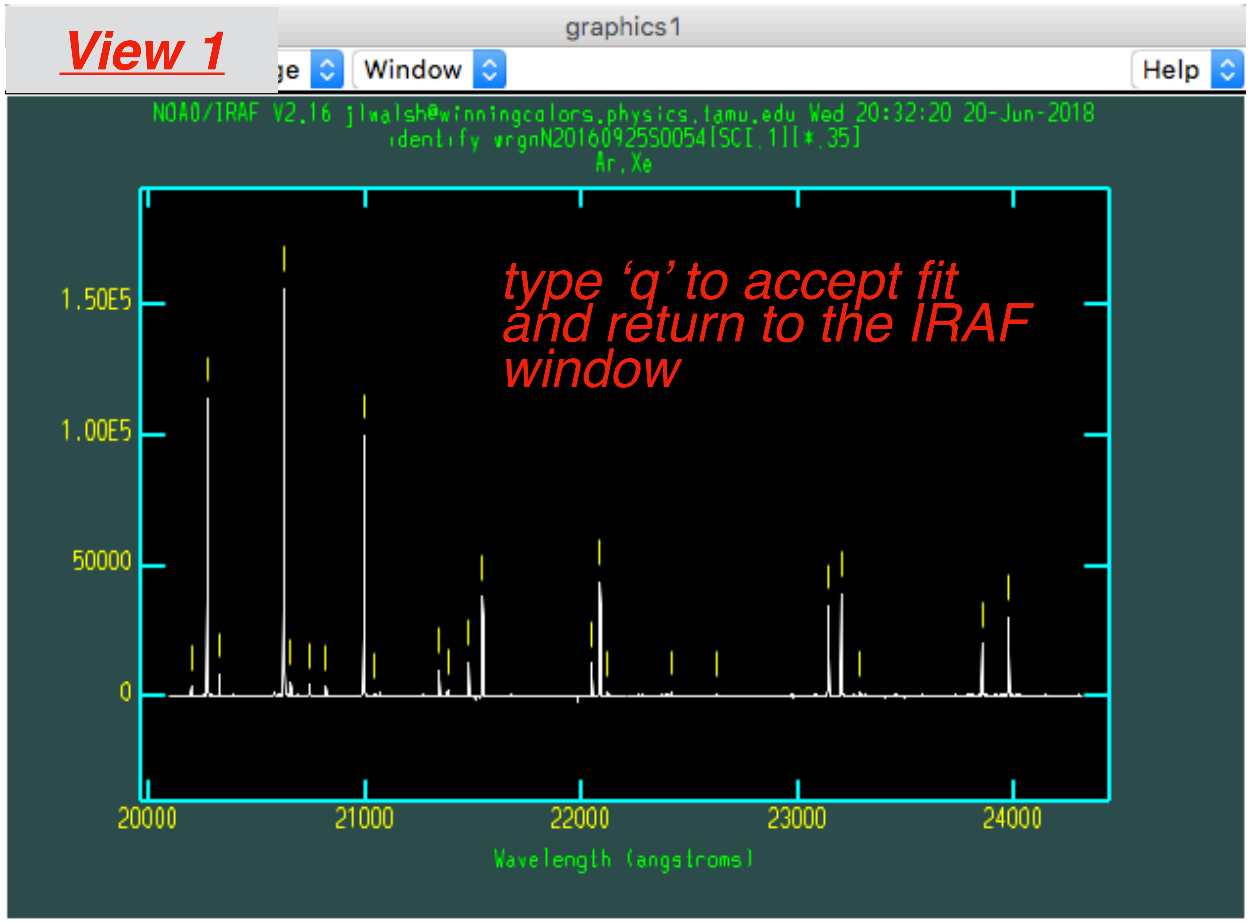
- With any of the views #3-5 up (page 13), type 'q' in the interactive window to go back to view #1 (page 11). You can delete lines in view #1, or you can delete points in views #2-5 by placing your cursor over the point and typing 'd'. Sometimes going to view #1, deleting a line, then marking the same line again produces a better RMS.

- Once satisfied, type 'q' in the interactive window with view #1 or 2 displayed to return to the IRAF window. (This means typing 'q' once with views #3-5 up, and typing 'q' a second time once view #1 is up.)



```
NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 20:28:22 20-Jun-2018
func=chebyshev, order=4, low_rej=3, high_rej=3, niterate=10, grow=0
total=23, sample=23, rejected=0, deleted=0, RMS= 0.1227
```

type 'q' to go to view #1



```
NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 20:32:20 20-Jun-2018
identify vrgnN20160925S0054[SCI,1][*,35]
Ar,Xe
```

type 'q' to accept fit and return to the IRAF window

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’

- After examining the middle spatial location of the slice, the pipeline will go through groups of 10 rows (spectra). For example, if row 35 was examined previously (pages 11-14), row 25 will examined, followed by row 15, 10, 5, then 45, 55, and 65. For each group of rows, the arc lines will be automatically identified using the dispersion function found for the middle spatial location of the slice, and the dispersion function for the current group of rows being examined will be fit using the identified lines. Before starting a new group of rows, the IRAF window will ask if you want to examine the dispersion function fit interactively.

    - Typing ‘YES’ or ‘yes’ in the IRAF window, will take you to view #1 (page 11), and you can repeat the steps on pages 11-14 for the new group of rows.

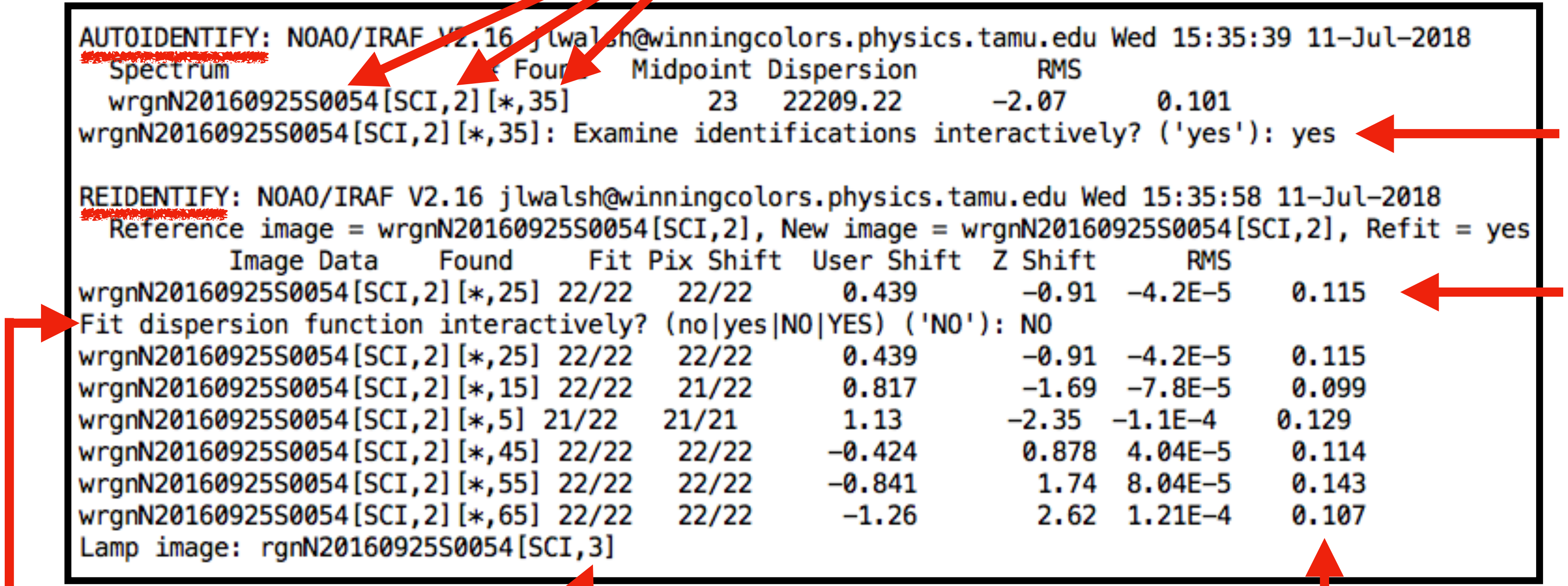
    - Alternatively, typing ‘NO’ or ‘no’ in the IRAF window, means that the dispersion function found for the middle spatial location of the slice will be used to identify lines for the remaining 6 groups of 10 rows for a single slice for one arc exposure. Those identified lines will be used to fit the dispersion function for that spatial location of the slice. There is no user interaction, and neither the line identification nor fit can be modified.

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’

Examine a group of 10 rows centered on row #35 of slice #2 of arc exposure 201609250054.fits

The AUTOIDENTIFY task automatically identifies arc lines by comparing to a user supplied line list. Typing ‘yes’ means you are able to examine the identified lines and see the dispersion function fit, as well as make adjustments (see pages 11-14).

IRAF Window



```
AUTOIDENTIFY: NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 15:35:39 11-Jul-2018
  Spectrum                         # Found   Midpoint Dispersion        RMS
  wrgnN20160925S0054[SCI,2][*,35]       23   22209.22       -2.07       0.101
wrgnN20160925S0054[SCI,2][*,35]: Examine identifications interactively? ('yes'): yes

REIDENTIFY: NOAO/IRAF V2.16 jlwalsh@winningcolors.physics.tamu.edu Wed 15:35:58 11-Jul-2018
  Reference image = wrgnN20160925S0054[SCI,2], New image = wrgnN20160925S0054[SCI,2], Refit = yes
          Image Data     Found      Fit Pix Shift  User Shift  Z Shift      RMS
wrgnN20160925S0054[SCI,2][*,25] 22/22    22/22       0.439       -0.91  -4.2E-5     0.115
Fit dispersion function interactively? (no|yes|NO|YES) ('NO'): NO
wrgnN20160925S0054[SCI,2][*,25] 22/22    22/22       0.439       -0.91  -4.2E-5     0.115
wrgnN20160925S0054[SCI,2][*,15] 22/22    21/22       0.817       -1.69  -7.8E-5     0.099
wrgnN20160925S0054[SCI,2][*,5] 21/22    21/21        1.13       -2.35  -1.1E-4     0.129
wrgnN20160925S0054[SCI,2][*,45] 22/22    22/22      -0.424       0.878  4.04E-5     0.114
wrgnN20160925S0054[SCI,2][*,55] 22/22    22/22      -0.841        1.74  8.04E-5     0.143
wrgnN20160925S0054[SCI,2][*,65] 22/22    22/22       -1.26        2.62  1.21E-4     0.107
Lamp image: rgnN20160925S0054[SCI,3]
```

The dispersion function found above (through the AUTOIDENTIFY task), for the middle spatial location of the slice, is used to identify lines in the remaining groups of rows of the slice (via the REIDENTIFY task). Typing ‘NO’ or ‘no’ means you accept the line identifications and the dispersion function, and don’t want to examine interactively. Typing ‘YES’ or ‘yes’ means you want to examine the identifications and the fit interactively (repeating steps on pages 11-14 for the next group of 10 rows of this slice).

A new slice of the current arc exposure will be examined next.

RMS values from the dispersion function fit.

Typing ‘NO’/‘YES’ means no/yes is your response for all groups of 10 rows of this slice (and you won’t be asked again while on this slice). Typing ‘no’/‘yes’ means no/yes is your response for this group of 10 rows of this slice (and you will be asked again for the next group of 10 rows of this slice).

(Optional) Interactive Components of the Pipeline: flinter_arc = 'yes'

- When AUTOIDENTIFY has been run on the middle spatial location of the slice (e.g., row #35), and you are asked if you asked to examine the identifications interactively:

    - For quick reductions, it is okay to run this part non-interactively. But for more careful/final reductions, this part should be run interactively. Perhaps by playing around more with IRAF's nswavelength parameters in nifs_basecalib_LP.py, combined with using the modified line list, this task can always be run non-interactively in the future.

    - Don't need to spend much time examining view #1 (page 11) because a modified line list with only the strong lines is being used. However, it is a good idea to check the residuals (view #5, page 13). Again, RMS should be ~0.1 Å and the residuals should be randomly scattered.

    - IRAF will automatically reject identified lines that are very large outliers from the fit, but sometimes there is a marginal outlier. By deleting that line from the fit, or deleting then marking the line again, the RMS will go down (perhaps closer to ~0.1 Å).

    - Should try to keep as many lines as possible, particularly those at the ends of the wavelength range.

(Optional) Interactive Components of the Pipeline: flinter_arc = ‘yes’

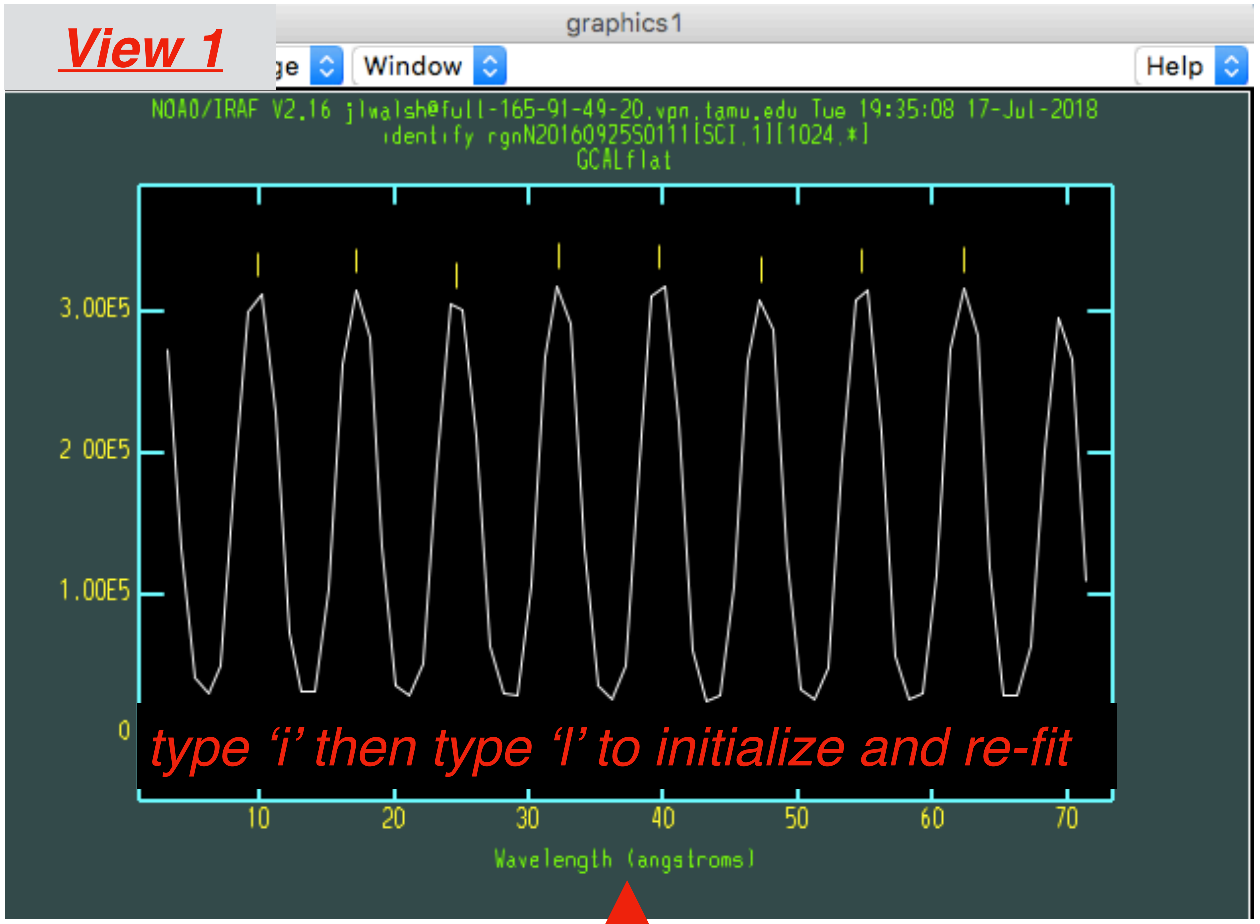
- When REIDENTIFY is being run on the other spatial locations of the slice (e.g., row #25, 15, 5, etc.), and you are asked to fit the dispersion function interactively:
    - It is okay for both quick and final reductions to run non-interactively (typing ‘NO’ in the IRAF window). It’s good to check though that the RMS is ~0.1 Å; the RMS is shown in the last column of numbers for a given spatial location in the IRAF window.
    - There is no harm in running interactively. However, this is tedious because there are 7 groups of 10 rows for a single slice, and there are 29 slices for one arc exposure. Interactively examining all groups of rows means looking at 7*29*number of arc exposures.

(Optional) Interactive Components of the Pipeline: flinter_sdist = ‘yes’

- An interactive window will appear (see view #1). The plot shown is the sum of 10 columns (spatial profile) at the middle wavelength of one slice of the ronchi mask exposure. Features are automatically identified (indicated by yellow marks above the peaks) by matching to an input coordinate list using AUTOIDENTIFY. The coordinate list is displayed in the IRAF window, and provides the pixel number at which each feature should be located. This step is analogous to identifying arc lines from an arc exposure for wavelength calibration, but it is for spatial rectification.

- Click on the interactive window to make active. Type ‘i’ to initialize, then type ‘l’ [lower case L] to re-fit. This should mark all the peaks.

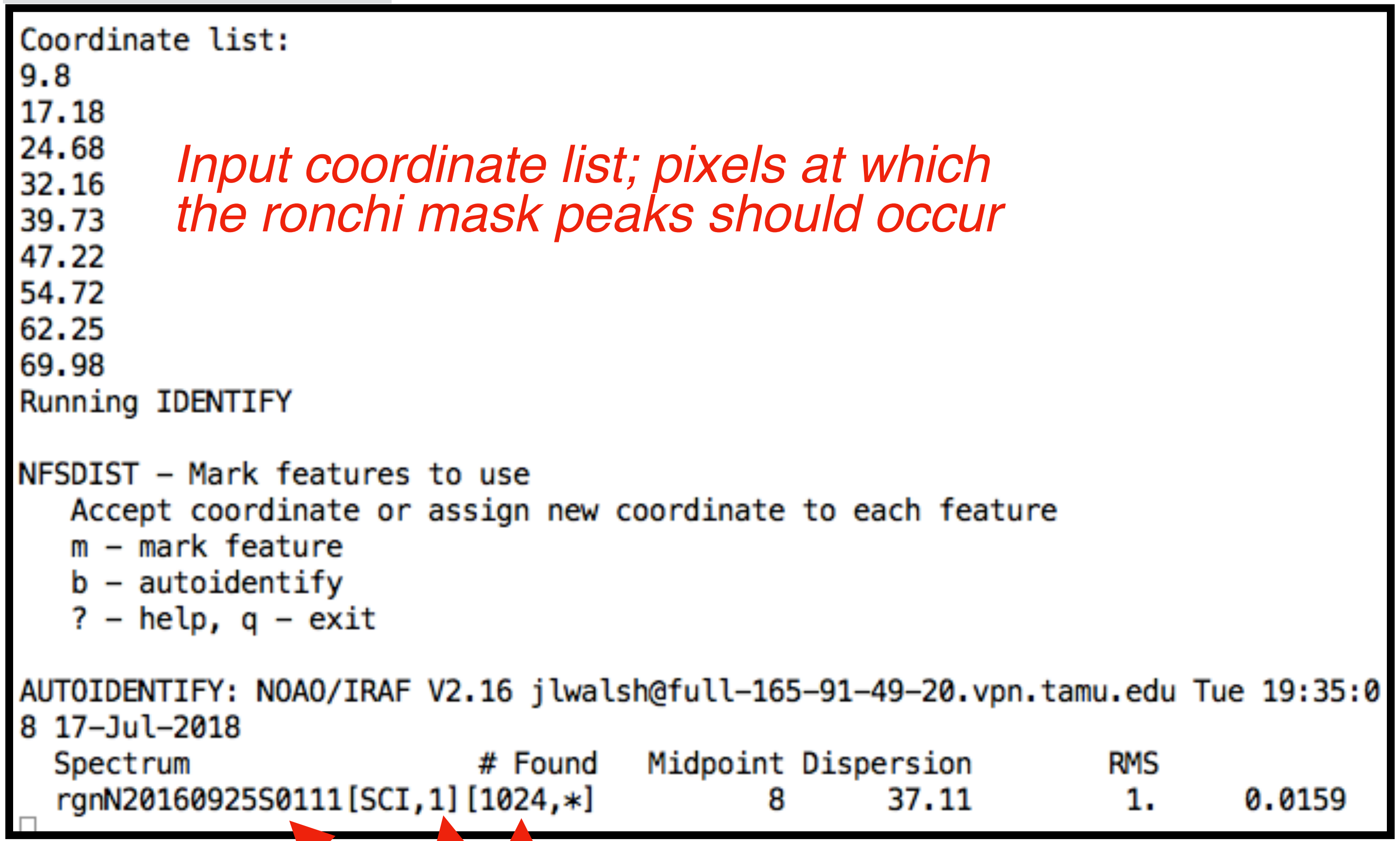
View 1


graphics1
ge
Window
Help
NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:35:08 17-Jul-2018
identify rgnN20160925S0111[SCI,1][1024,*]
GCALflat
3.00E5
2.00E5
1.00E5
0
10
20
30
40
50
60
70
Wavelength (angstroms)

type ‘i’ then type ‘l’ to initialize and re-fit

The x-axis is mislabeled. This is actually pixel number.

IRAF Window



```
Coordinate list:
9.8
17.18
24.68
32.16
39.73
47.22
54.72
62.25
69.98
Running IDENTIFY

NFSDIST - Mark features to use
   Accept coordinate or assign new coordinate to each feature
   m - mark feature
   b - autoidentify
   ? - help, q - exit

AUTOIDENTIFY: NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:35:0
8 17-Jul-2018
  Spectrum                        # Found  Midpoint Dispersion        RMS
  rgnN20160925S0111[SCI,1][1024,*]        8      37.11          1.      0.0159
```

Input coordinate list; pixels at which the ronchi mask peaks should occur

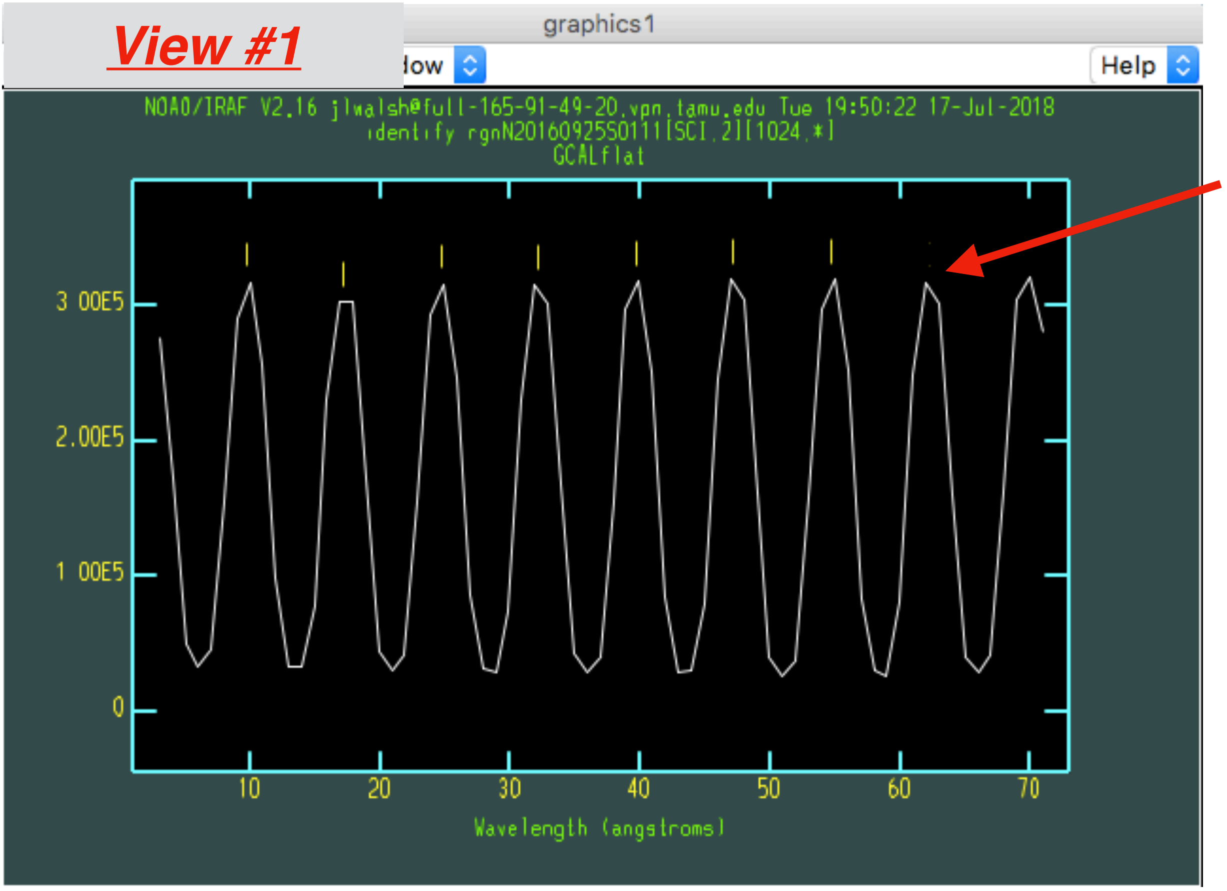
Examine a group of 10 columns centered on column #1024 of slice #1 of ronchi exposure 201609250111.fits

(Optional) Interactive Components of the Pipeline: flinter_sdist = ‘yes’

- If some good peaks are still not marked, put your cursor over the peak and type ‘m’. At the bottom of the interactive window in parentheses will be the pixel number at which the peak should be located, selected from the coordinate list. By clicking at the bottom of the interactive window, you can hit return to accept, or you can manually type in the pixel location. A yellow mark will appear above the peak. If a bad peak is marked, put your cursor over the peak and type ‘d’ to delete.

- Just like with the wavelength calibration, while view #1 is seen can type ‘f’ to look at the fit. The first plot that comes up in this new view shows the non-linear component (equivalent to view #3 on page 13). Typing ’h’ in the new view will show the peak location versus pixel number (equivalent to view #4 on page 13), and typing ‘j’ in the new view will show the residuals (equivalent to view #5 on page 13). If you want to see the non-linear component again, type ‘l’ [lower case L] in the new view. Type ‘q’ to get back to view #1.

- With view #1 up, type ‘q’ to return to the IRAF window.

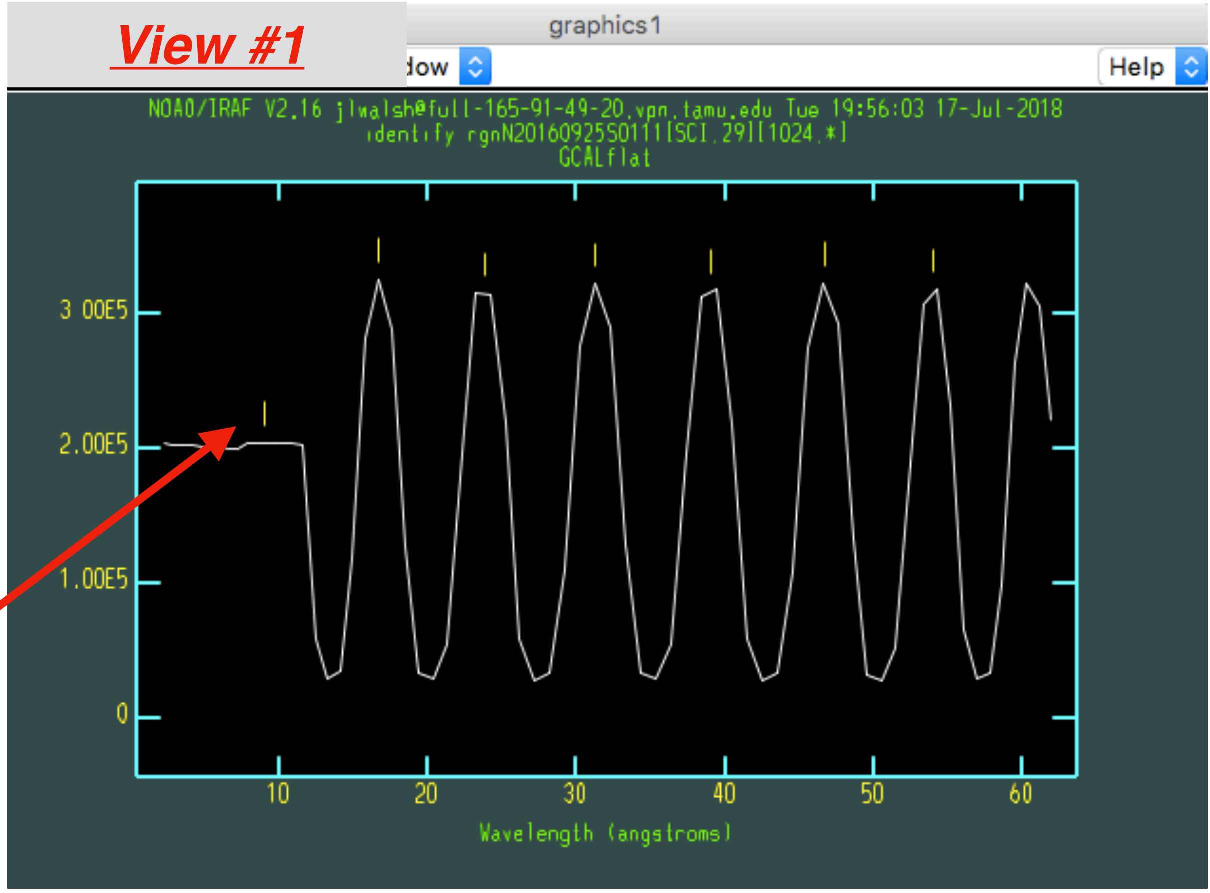


View #1

```
NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:50:22 17-Jul-2018
identify rgnN20160925S0111[SCI,2][1024,*]
GCALflat
```

a good peak is not marked. put cursor over peak and type ‘m’ to mark.

example of a bad peak. place cursor over the peak and type ‘d’ to remove.



View #1

```
NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:56:03 17-Jul-2018
identify rgnN20160925S0111[SCI,29][1024,*]
GCALflat
```

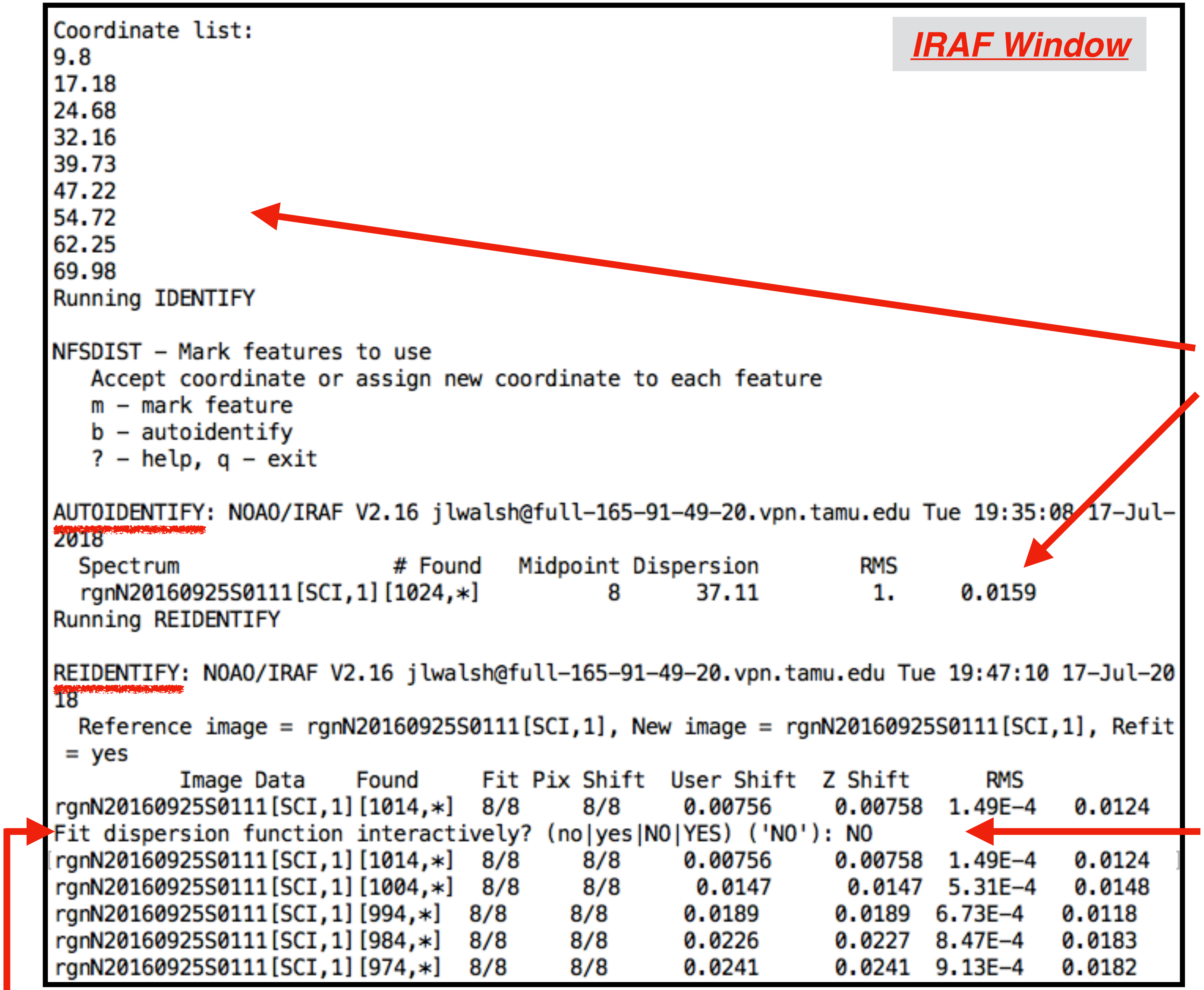
(Optional) Interactive Components of the Pipeline: flinter_sdist = ‘yes’

- After examining the middle wavelength of the slice, the pipeline will go through groups of 10 columns (spatial profiles). For example, if column 1024 was examined previously (pages 19-20), column 1014 will examined, followed by column 1004, 994, etc. down to 4, then 1034, 1044, etc. up to 2034. For each group of columns, the ronchi mask peaks will be automatically identified using the function found for the middle wavelength of the slice, and the function for the current group of columns being examined will be fit using the identified peaks. Before starting a new group of columns, the IRAF window will ask if you want to examine the function fit interactively.

    - Typing ‘YES’ or ‘yes’ in the IRAF window, will take you to view #1 (page 19), and you can repeat the steps on pages 19-20 for the new group of columns.

    - Alternatively, typing ‘NO’ or ‘no’ in the IRAF window, means that the function found for the middle wavelength of the slice will be used to identify peaks for the remaining 203 groups of 10 columns for a single slice for one ronchi mask exposure. Those identified peaks will be used to fit the function for that wavelength of the slice. There is no user interaction, and neither the peak identification nor fit can be modified.

(Optional) Interactive Components of the Pipeline: flinter_sdist = ‘yes’



```
Coordinate list:
9.8
17.18
24.68
32.16
39.73
47.22
54.72
62.25
69.98
Running IDENTIFY

NFSDIST - Mark features to use
   Accept coordinate or assign new coordinate to each feature
   m - mark feature
   b - autoidentify
   ? - help, q - exit

AUTOIDENTIFY: NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:35:08 17-Jul-
2018
  Spectrum                  # Found   Midpoint Dispersion       RMS
  rgnN20160925S0111[SCI,1][1024,*]         8       37.11         1.      0.0159
Running REIDENTIFY

REIDENTIFY: NOAO/IRAF V2.16 jlwalsh@full-165-91-49-20.vpn.tamu.edu Tue 19:47:10 17-Jul-20
18
  Reference image = rgnN20160925S0111[SCI,1], New image = rgnN20160925S0111[SCI,1], Refit
 = yes
          Image Data     Found      Fit Pix Shift  User Shift  Z Shift       RMS
rgnN20160925S0111[SCI,1][1014,*]  8/8      8/8     0.00756     0.00758  1.49E-4    0.0124
Fit dispersion function interactively? (no|yes|NO|YES) ('NO'): NO
[rgnN20160925S0111[SCI,1][1014,*]  8/8      8/8     0.00756     0.00758  1.49E-4    0.0124 ]
rgnN20160925S0111[SCI,1][1004,*]  8/8      8/8      0.0147      0.0147  5.31E-4    0.0148
rgnN20160925S0111[SCI,1][994,*]  8/8      8/8     0.0189      0.0189  6.73E-4    0.0118
rgnN20160925S0111[SCI,1][984,*]  8/8      8/8     0.0226      0.0227  8.47E-4    0.0183
rgnN20160925S0111[SCI,1][974,*]  8/8      8/8     0.0241      0.0241  9.13E-4    0.0182
```

The AUTOIDENTIFY task automatically identifies features in the ronchi mask by comparing to the coordinate list. With flinter_sdist set to ‘yes’ (at the top of nifs_main_LP.py), you’ll examine the identified peaks and can see the fitted function, as well as make adjustments (see pages 19-20).

The function found above (through the AUTOIDENTIFY task), for the middle wavelength of the slice, is used to identify peaks in the remaining groups of columns of the slice (via the REIDENTIFY task). Typing ‘NO’ or ‘no’ means you accept the feature identifications and the fitted function, and don’t want to examine interactively. Typing ‘YES’ or ‘yes’ means you want to examine the identifications and the fit interactively (repeating steps on pages 19-20 for the next group of 10 columns of this slice).

Typing ‘NO’/‘YES’ means no/yes is your response for all groups of 10 columns of this slice (and you won’t be asked again while on this slice). Typing ‘no’/‘yes’ means no/yes is your response for this group of 10 columns of this slice (and you will be asked again for the next group of 10 rows of this slice).

(Optional) Interactive Components of the Pipeline: flinter_sdist = 'yes'

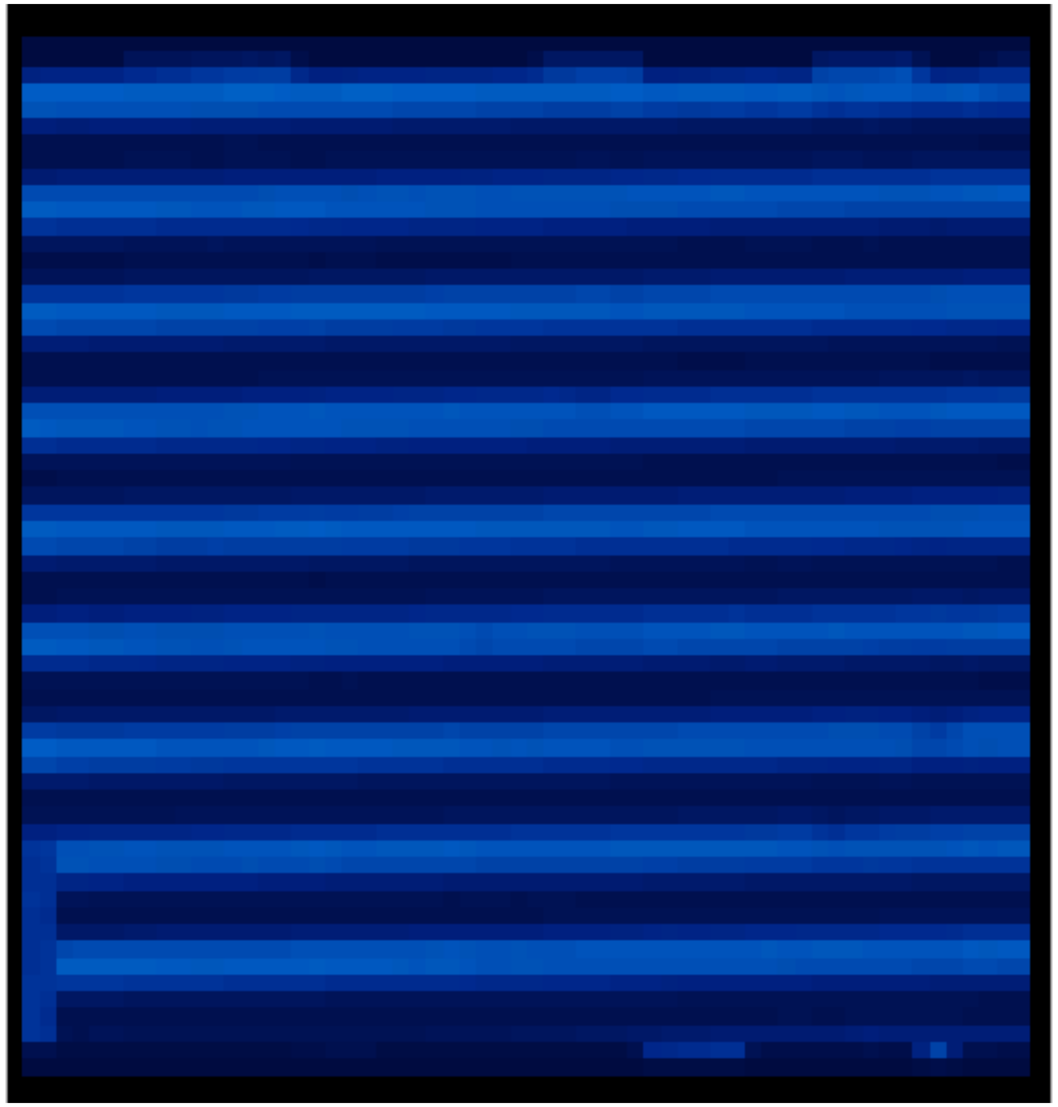
- When AUTOIDENTIFY has been run on the middle wavelength of the slice (e.g., column #1024), and you are asked if you asked to examine the identifications interactively:
  - For quick reductions, it is okay to run this part non-interactively. But for more careful/final reductions, this part should be run interactively. Perhaps by playing around more with IRAF's nfsdist_jonelle parameters in nifs_basecalib_LP.py, this task can always be run non-interactively in the future.
- When REIDENTIFY is being run on the other wavelengths of the slice (e.g., columns #1014, 1004, 994, etc.), and you are asked to fit the dispersion function interactively:
  - It is recommended for both quick and final reductions to run non-interactively (typing 'NO' in the IRAF window). Running interactively is extremely tedious because there are 203 groups of 10 columns for a single slice, and there are 29 slices. There is only one ronchi mask to examine because multiple ronchi mask exposures were averaged together before calling nfsdist_jonelle.

(Optional) Interactive Components of the Pipeline: flinter_sdist = ‘yes’

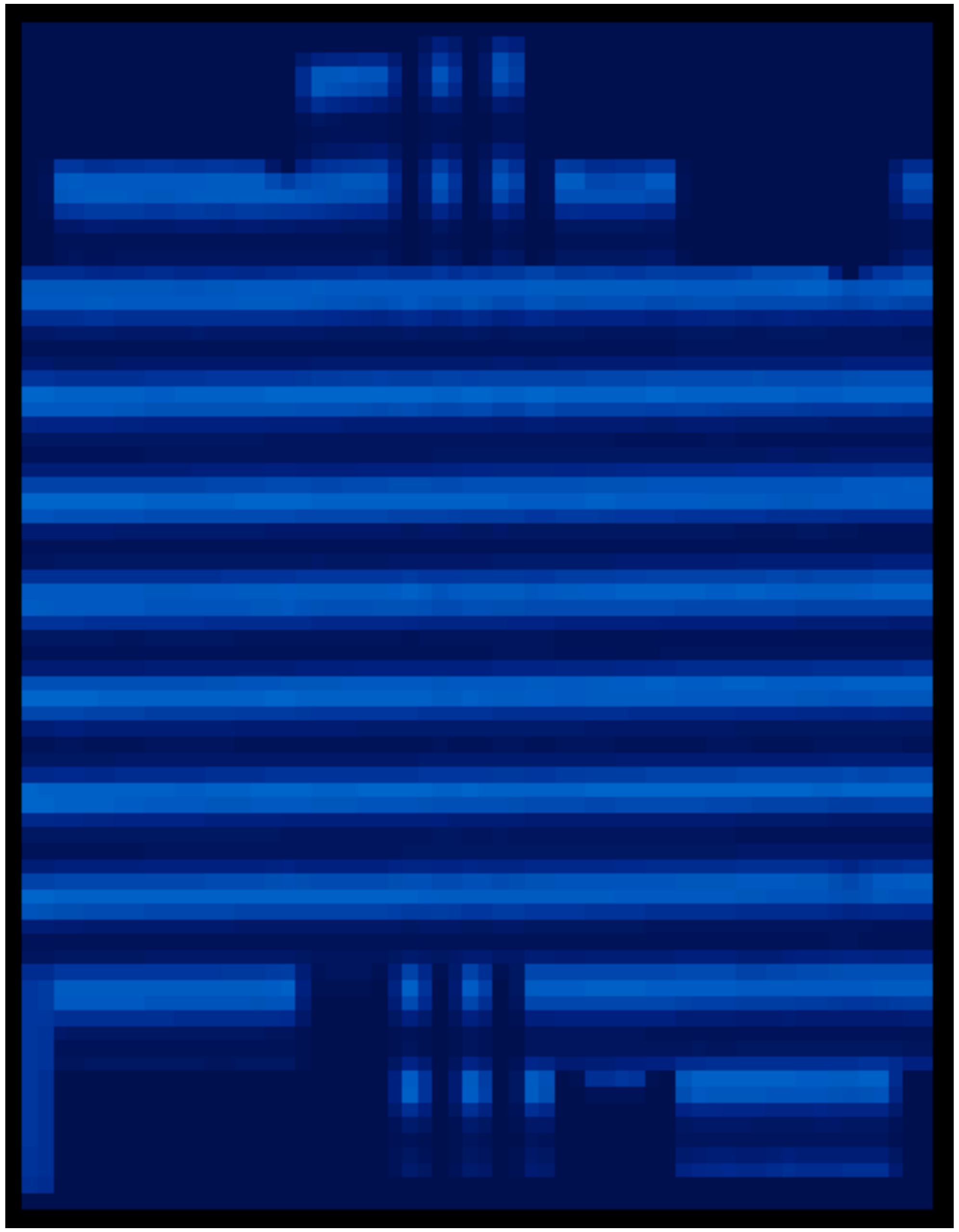
- As a check, the distortion correction determined from nfsdist_jonelle.py is applied to the ronchi mask itself (using the nsfitcoords [run non-interactively] and nstransform tasks), and a cube is generated (using the nifcube task). You should check the output cube, called “cube_tfrgn*.fits”.

The spatially (and spectrally) rectified ronchi mask cube has been collapsed to produce the images below.

Good



Bad



(Optional) Interactive Components of the Pipeline: flinter_nsfitcoords = ‘yes’

(Optional) Interactive Components of the Pipeline: flinter_extract = ‘yes’

(Optional) Interactive Components of the Pipeline: flinter_telluric = ‘yes’

Still need to document.