

Gemini-South IFU Data Reduction User Manual

Elaine M. Snyder
Dept. of Physics & Astronomy
University of North Carolina at Chapel Hill
`emsnyder@live.unc.edu`

February 22, 2016

Contents

1	Introduction	3
1.1	What is an IFU and why would we want to use one?	3
1.2	Observing setups	3
1.3	Observing patterns	4
1.4	Files you'll need for reduction	5
1.5	How the data is stored	5
1.5.1	The raw fits files	5
1.5.2	What is an MDF?	5
2	Setting Up Your Workspace	9
3	Data Reduction Guide	11
3.1	How to Use This Guide	11
3.2	Entering and Exiting the Pipeline	11
3.3	Organizing and Identifying Your Data	11
3.4	Reduction of the Flats	11
3.5	Overscan Subtraction of the Arcs	14
3.6	Identification and Removal of Bad Pixels from the Flats and Arcs	15
3.7	Extraction of the Arcs	16
3.8	Creation of the Wavelength Solution	16
3.9	Application of the Calibration to the Arcs	18
3.10	Quantum Efficiency Correction of Flats	18
3.11	Re-extraction of the Flats	18
3.12	Bias and Overscan Subtraction of Science Data	18
3.13	Identification and Removal of Bad Pixels from the Science Data	19
3.14	Cosmic Ray Rejection	19
3.15	QE Correction of Science	19
3.16	Flat Fielding and Extraction of Science	19
3.17	Wavelength Calibration of Science	19
3.18	Sky Subtraction of Science	19
3.19	Flux Calibration of Science	19
3.20	Data Cube Creation	19
3.21	Creation of a WCS Solution	20
3.22	Summation of the Data Cubes	20
4	Derivation of Kinematics	21
4.1	Deriving Velocity Dispersions	21
4.2	Deriving Rotation Curves	21

List of Figures

1.1	IFU Fibers and Data Format	7
1.2	Example of DS9 cube dialog box	8
1.3	Example of DS9 cube dialog box	8
3.1	Overscan Subtraction Example	12
3.2	Examples of the Fiber ID Step	14
3.3	Example Image of a Bad Fiber	14
3.4	Cosmic Rays vs. Bad Pixels	15
3.5	Example of the wavelength solution	17
3.6	Example of the residuals of the wavelength fit	17
3.7	Before and After Wavelength Calibration	18
3.8	Examples of sky subtraction	20

List of Tables

1.1	RESOLVE's Observation Setups	4
-----	--	---

Chapter 1

Introduction

Greetings, Gemini Multi-Object Spectrograph (GMOS) integral field unit (IFU) data reduction pipeline user! Before beginning your data reduction journey, it will be important to understand the structure of an IFU and the way the Gemini IFU data is stored in the fits files you'll be reducing. If you are familiar with this already, feel free to skip ahead. Most of this information can be found online on <https://www.gemini.edu>, but I've compiled it all here for easy reference and so that I may reference it in the context of the RESOLVE¹ survey's instrument setups.

1.1 What is an IFU and why would we want to use one?

The basic idea of any spectrograph is that incoming light from an object goes through a slit, is dispersed by a grating/prism/grism, and then focused onto the CCD detector to create an image of the spectrum. Most of the time the slit is long and narrow ($\sim 0.5\text{--}2''$ typically), and most astronomers would align it along a galaxy's major axis. This image will thus have spectral information in the x direction and spatial information in the y direction. The GMOS IFU is different in that instead of having one slit aligned along a galaxy's major axis, you will have multiple fibers that cover the entire galaxy. These fibers act as "mini slits", and each one will be dispersed and refocused onto the detector. This means you receive spectral information from all parts of the galaxy, not just where a single slit falls, and therefore are getting x, y, and wavelength at once. Thus, the main benefit of using an IFU is that it enables the 3D spatial analysis of the kinematics, metallicities, or whatever else you may want to derive from your spectra.

In the case of the GMOS IFU setups for RESOLVE observations, you will have either 750 or 1500 hexagonal fibers (depending on which setup was used) that cover an on-sky area of $5'' \times 3.5''$ or $7'' \times 5''$. The top-right portion of [Figure 1.1](#) gives an example of the positions of the fibers over a galaxy. As the top left portion of [Figure 1.1](#) shows, the GMOS IFU is designed in such a way that not all of the fibers are centered on the object – either 250 or 500 of the fibers are positioned on a region of sky away from the target (but not too far away). This lets us observe separate spectra of the scattered light from our atmosphere, which can later be subtracted from the science spectra during data reduction. This "sky level" can be affected by moon illumination, clouds, etc. so it is important to be able to subtract any extra light there may be. Note that sometimes in online documents, the fibers may be called "spaxels", which is short for "spectral pixels" – don't be confused by this hip IFU lingo.

1.2 Observing setups

As alluded to before, there are two main setups for RESOLVE's IFU observations. Similarly to our SOAR observations, there is a blue setup for absorption line galaxies and red setup for emission line galaxies. [Table 1.1](#) highlights the main differences between the two setups. To determine which galaxy is best for which setup, we ideally like to use the SOAR broad spectrum as a guide. Seeing strong emission in the broad spectrum will point to the emission line setup on Gemini being best, while no emission will point to the absorption line setup being preferred. However, since there is not

¹<http://www.resolve.astro.unc.edu>

quantity	blue setup	red setup
wavelength range	4200 – 5600 Å	5500 – 6900 Å
used for	absorption line galaxies	emission line galaxies
central wavelengths	4850, 4900 Å	6300, 6400 Å
features of interest	H β , Mgb, Fe5270, Fe5335	H α , [NII], NaD
grating	B1200	B600
filter	none (open)	r-G0326
slits	1 slit	2 slits
field of view	3.5'' \times 5''	5'' \times 7''

Table 1.1: Comparison of RESOLVE’s red and blue observation setups for the Gemini-South IFU.

always a SOAR (or even an SDSS) spectrum available, we often rely on their photometric colors as an indicator of whether emission or absorption is more likely.

Using the GMOS IFU is ideal for RESOLVE’s purposes for two different reasons. In the red setup, our goal is to obtain velocity fields for RESOLVE’s smallest emission line galaxies. Thus, the spatial information we receive from the IFU spectra is ideal for this purpose. On the other hand, for the blue setup, our goal is to derive velocity dispersions. The small, absorption line galaxies we target with the IFU in the blue setup are often quite faint, so instead of focusing on the spatial information the IFU gives, we sum all the spaxels out to the effective radius of the galaxy, thereby summing the light enough to derive accurate dispersion measurements.

We estimate exposure times for both setups using the online Integration Time Calculators² (ITC) supplied by Gemini. We use an elliptical galaxy SED template for time estimates in the blue setup and adjust our exposure time calculations so that we may achieve a signal-to-noise ratio of 25 per Å when we bin all the fibers out to the effective radius of the galaxy. We use a model H α line with estimates of the line flux and continuum flux density for time estimates in the red setup, and adjust our exposure times to achieve a centroiding accuracy of $\sim 5.3 \text{ km s}^{-1}$. For both setups, we bin our spectra by 2 in the spectral direction to increase the signal-to-noise as well.

We also use the ITC to select central wavelength positions that ensure our features of interest do not fall in a chip gap (see bottom of [Figure 1.1](#)). We choose two different central wavelengths so that we can fill in the chip gaps as well when we sum the individual exposures – this is sometimes referred to as “spectral dithering”. The central wavelength positions selected for our setups are listed in [Table 1.1](#).

1.3 Observing patterns

In 2-slit (aka red setup) mode, we oftentimes will tile the IFU field of view in order to cover the entire extent of the galaxy. The 5'' \times 7'' field of view can either be tiled to 10'' \times 7'' for larger, rounder galaxies or to 5'' \times 14'' for longer, skinnier galaxies. This doesn’t make a large difference to the reduction routine, but there will be more science frames to reduce and more data cubes to stack at the end.

The basic observation sequence is different for each mode. Both will start with acquisition images to ensure the IFU field of view is centered on the galaxy. Then flats, arcs, and science frames are taken in each dither. For example, a flat, arc, science, and arc frame will be taken at one central wavelength, and then an arc, science, arc, flat frame will be taken at the other central wavelength, in that order. For reduction, we will want to flat field and wavelength transform the science images using the arcs and flats taken in the same dither. Once the data cubes are made, we can then sum the data cubes, both spatially and spectrally.

²<http://www.gemini.edu/node/10479>

1.4 Files you’ll need for reduction

The fits files you’ll want for your data reduction can all be found online in the Gemini Science Archive (<http://archive.gemini.edu>). You will have to create an account with your unique observing program ID and program key (usually sent in an email for each new semester). RESOLVE’s IDs are GS-2013B-Q-51 for the fall 2013 semester, GS-2014B-Q-13 and GS-2014B-Q-52 for the fall 2014 semester, and GS-2015B-C-1 (and carry-over GS-2014B-Q-13 band 1 time) for the fall 2015 semester (email Elaine or Sheila³, if you need the program keys for any of these programs). Once your account is created, you can look at each observing program and download the data you want.

As alluded to § 1.3, you’ll have flats, arcs, and science data for each galaxy in each wavelength dither. Normally, there are 2 flats, 2 or 4 arcs, and 2 or 4 science frames, depending on the estimated exposure times and/or the spatial offsets needed.

We also obtain a suite of baseline standards each semester – one standard for both the red and blue setups that will be used for each galaxy observation taken that semester. This includes a science observation of standard star along with flats, arcs, and twilight flats in both wavelength dithers. We use the twilight flats to create response functions for each fiber in the IFU, and use the standard star science observations to correctly flux calibrate our galaxy data.

Lastly, we’ll want a bias image in order to perform the bias subtraction. Each night after observing is complete, a standard set of bias frames are observed by the Gemini South operators. They are a little tricky to find in the online database, but you can go to the “View Associated Calibrations” tab to find all of the biases taken. Usually, searching on the observation date, the correct binning (CCDSUM 2 1), and the correct filter (see Table 1.1) will yield 5 bias frames to download and process with the Gemini IRAF routine, `gbias`. I have downloaded these already for the RESOLVE data, and have trimmed, overscan subtracted, and stacked them.

1.5 How the data is stored

1.5.1 The raw fits files

Simply opening one of the S....fits files with DS9 will yield a strange result. You’ll see a very long and skinny image appear. What you are actually seeing is 1/12th of the full image. This is because there are 3 CCDs and each one has 4 individual amplifiers that are read out into different extensions in your fits file. To see all the extensions, type `ds9 -mecube S....fits`, which will open a cube dialog box that lets you scroll through each extension in the file. See Figure 1.2 and Figure 1.3 as an example of what you’ll see when you open a raw image. A part of the data reduction is to mosaic all of these extensions onto one single large image.

1.5.2 What is an MDF?

The mask definition file (MDF) stores information about the IFU and is “attached” to the raw images as one of the first steps in the data reduction process. The actual file is in binary fits format and comes with the Gemini IRAF package. The files are called either “`gsifu_slitr_mdf.fits`” or “`gsifu_slits_mdf.fits`” for the 1- and 2-slit data, respectively, and can be found in your IRAF/Gemini folder (probably a directory path similar to `~/iraf/gemini/gmos/data/`) or by typing `cd gmos$data` in the IRAF/PyRAF environment. Reading the file into Python (example below) as a fits table reveals all the table columns: the fiber number (column name “NO”), x and y spatial position of each fiber in both arcseconds (“XINST”/“YINST”) and pixels (“XLDIS”/“YLDIS”), which block (bundles of 50 fibers) that the fiber is in (“BLOCK”), and whether the fiber is good or bad (“BEAM”).

Figure 1.1 shows the spatial positions of the fibers over an example galaxy on the top, while the bottom image shows the 2D information that is recorded on the CCD (wavelength on the x axis and fiber number on the y axis). This illustrates why we need the MDF to map the 2D image we get

³[sheila\[at\]physics\[dot\]unc\[dot\]edu](mailto:sheila[at]physics[dot]unc[dot]edu)

from the CCD to a 3D data cube of x and y position and wavelength later in the reduction.

The “BEAM” column tells us which fibers are good and bad. See [Figure 3.3](#) for an example of what good/bad fibers look like and see [§ 3.4](#) for an example of why you may need to use/change this information. Good fibers will have BEAM=1 while bad will have BEAM=-1. To mark a fiber as bad, you can read in the appropriate fits file in IDL or Python, and change the BEAM entry for the fiber(s) of your choice. In Python, this process will probably look something like this (for the example of marking fiber 745 as bad):

```
>>> from astropy.io import fits
>>> import numpy as np
>>> hdulist = fits.open('gsifu_slits_mdf.fits')
>>> table = hdulist[1].data
>>> table
FITS_rec(
[(1, 0.34640503, 4.8000002, 'I_1', 1, 49, 57),
 (2, 0.34640503, 4.5999999, 'I_2', 1, 47, 57),
 (3, 0.34640503, 4.4000001, 'I_3', 1, 45, 57), ...,
 (1498, 3.117645, 4.4000001, 'A_3', 1, 45, 41),
 (1499, 3.117645, 4.5999999, 'A_2', 1, 47, 41),
 (1500, 3.117645, 4.8000002, 'A_1', -1, 49, 41)],
 dtype=[('NO', '>i4'), ('XINST', '>f4'), ('YINST', '>f4'), ('BLOCK', 'S5'),
        ('BEAM', '>i4'), ('XLDIS', '>i4'), ('YLDIS', '>i4')])
>>> sel = np.where(table['NO'] == 745)[0][0]
>>> sel
744
>>> table['BEAM'][sel] = -1
>>> hdulist.writeto('newMDF.fits')
```

Once created, you can feed the file to the pipeline code as one of your first steps of the reduction process.

GMOS IFU Example Data: NGC 221

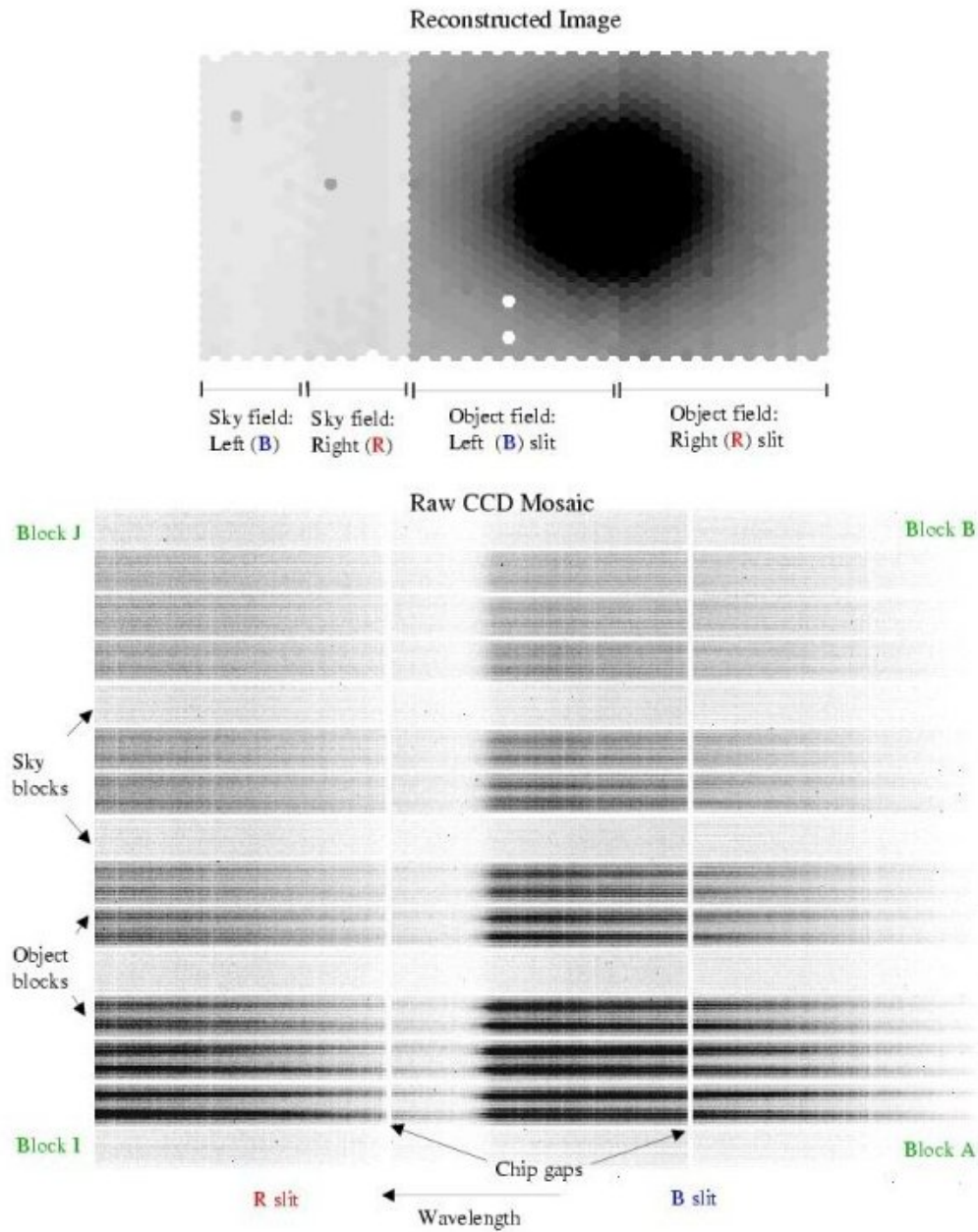


Figure 1.1: The top part of this image shows the IFU field of view over NGC221 (not a part of RESOLVE). The left portion of the image shows the sky spaxels, and the right portion shows the light of the galaxy. This is a 2D representation of the data we are getting – the third dimension is in the wavelength direction, and the darkness over where the galaxy is from summing the light inside each spaxel there. The bottom portion of this image shows the raw fits data from the above galaxies. Image from <https://www.gemini.edu/sciops/instruments/gmos/spectroscopy/integral-field-spectroscopy/field-slit-mapping>

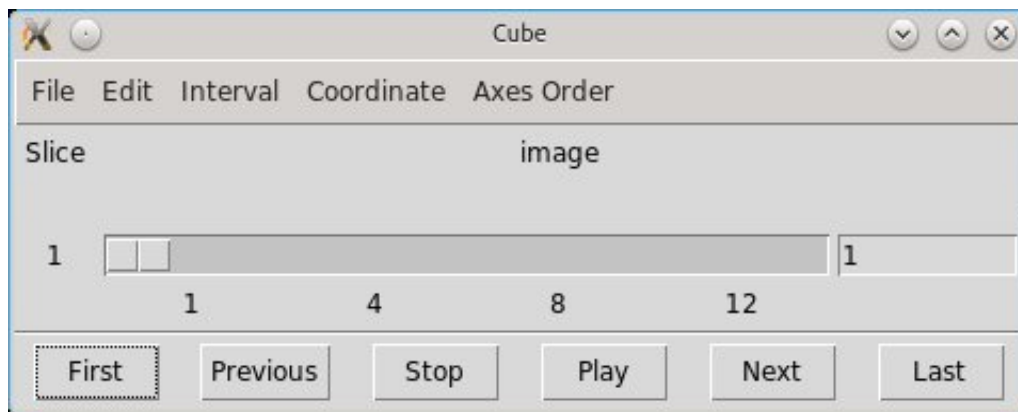


Figure 1.2: This is an example of the cube box that will open when you load a raw image in DS9.



Figure 1.3: This is an example raw image. Specifically this is a flat image, but you can also open the raw science and arcs. Each row you see is a fiber, and in the x direction is wavelength.

Setting Up Your Workspace

```
> pyraf
```

This is the EXPORT version of IRAF V2.16 supporting PC systems.

```
--> gemini
```

If you are new to data reduction in general, you will want to download DS9, an astronomical imaging application that you'll use to view our data as you progress through the pipeline. Download DS9 from this site: <http://ds9.si.edu/site/Home.html>.

Next, you'll want data to actually reduce! You should have flats, arcs, science frames, and a bias image before getting started. You should also have response flats from the baseline twilights and a flux calibration from the standard star data. See § 1.4 for more info on how to get your data and on each specific file you'll need. There should also be an observing log to inspect (RESOLVE data has `obslog.txt` or just `log.txt`) as well as an alternative line list file “`smalllinelist.dat`” that will be helpful during the wavelength calibration steps. Place all of these files in the same folder, since that's where the pipeline looks for them.

Lastly, you'll want to download the actual pipeline. You can access the code and this handy user guide at <https://github.com/emsnyder/geminiDRpipeline>. Always be on the lookout for updates in the future! You can place the code anywhere you like, but a nice folder structure would be to have individual folders for each galaxy with the pipeline code a level above. Now you should be all set, congrats!

Chapter 3

Data Reduction Guide

3.1 How to Use This Guide

The author's goals for this user guide are threefold: to make it clear what you are doing in each step, to explain why each step is important and necessary, and to demonstrate how to successfully complete each step when user interaction is needed. Also included with each step is the name of the Gemini IRAF tasks being used. For a more in depth look at what is going on, typing `help <name of task>` inside of PyRAF will open a detailed document with information about the task and what parameters you can change. This should make editing the task parameters in the pipeline easier, if it is ever needed.

3.2 Entering and Exiting the Pipeline

Here are the basic start up commands:

1. cd to the directory where you put your data
2. enter the Ureka environment by typing `ur_setup` in the command line
3. enter PyRAF by typing `pyraf` in the command line
4. enter the pipeline by typing `execfile('/path/to/your/code/gemreductionpipeline.py')`

To end your session:

1. type `CTRL-C` to exit the pipeline, if not out already
2. enter `.exit` to exit PyRAF
3. use `ur_forget` to exit the Ureka environment

3.3 Organizing and Identifying Your Data

The very first thing the pipeline does is ask you for the folder you are working in. Be sure to include the full path to your folder (including a trailing forward slash '/'), which can be found by typing `pwd` in your PyRAF window but outside of the pipeline. The pipeline checks that you are actually working inside this folder next, and will direct you to fix this if you are not. Next, it will ask if you are using a special MDF file. If you are, be sure to give the full path to that file, and if not, just type `default`.

Next, the pipeline will look in your working folder and identify which files are there. It will print out the files it finds and whether they are arcs/flats/science/etc. and which central wavelength they were observed at. It is good at this very first step to open up your raw images in a ds9 to visually inspect your data files for any strange data. Also, be sure check the identifications against your observing log! Every file is identified by looking in the image headers, so problems here most likely will be missing images. If these all look correct, type `1` to go on, if not, press `0` which will exit the pipeline and let you figure out what's wrong/missing.

3.4 Reduction of the Flats

Gemini IRAF task used: `gfreduce`

This very first step is often (in the author’s experience) the most time-consuming and frustrating step of the entire reduction. We use the flats not only for flat fielding the science data, but also to correctly order the fibers in the raw data. See bottom of [Figure 1.1](#) – each row is an individual fiber spectrum that must be correctly numbered, which ensures that we can successfully map our raw data to a 3D (x, y, wavelength) data cube. The IRAF task `gfreduce` has many different options, but for this very first step, we are just going to attach the MDF file, do a bias subtraction, overscan subtraction, and fiber ID so that we may extract all the spectra.

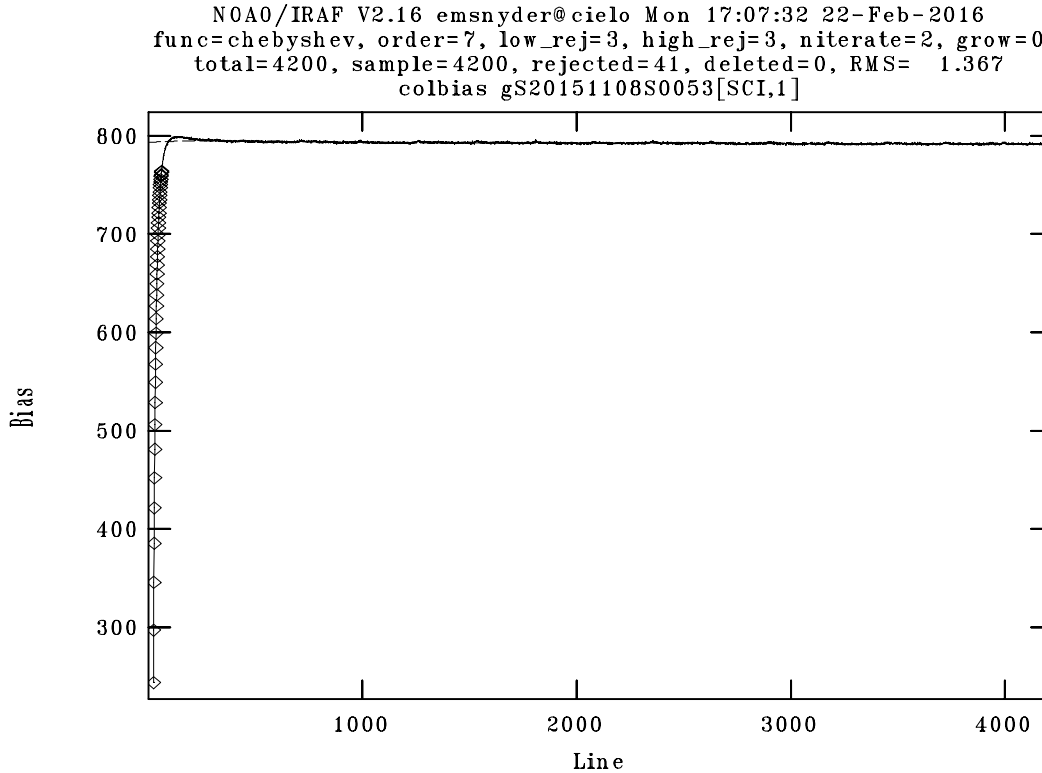


Figure 3.1: An example of the overscan subtraction window that PyRAF will open during the reduction of the flats step. The solid line represents the data, while the dashed line shows the fit to that data. The diamonds are data points that are being rejected from our fit.

1. The MDF attachment is done without user input needed. A file is produced in your working directory with the prefix ‘g’.
2. The bias and overscan subtraction is mostly done without user input needed. The overscan is the regions of the image with no data on it, i.e., between the chips and amplifiers as well as around the edges of the image. We want to measure this overscan level and also trim these regions from our images. To do this, a PyRAF window will open with the overscan level being plotted. Also plotted is a best fit line to the overscan level, and this normally looks fine as is. Press `q` inside the PyRAF window to move on. There is a plot for each amplifier in the CCDs, so you will see 12 individual plots. If the fit does not look fine, you can type `:order #` and then `f` inside the PyRAF window to change the order of the fit to a different number (with that number instead of `#` in the previous command). Higher orders typically do better at fitting any strange features (spikes/dips). Unless something is very wrong, you probably won’t need to worry about this. See [Figure 3.1](#) for an example of what you should be seeing. This step produces a file with the prefix ‘rg’, where the ‘g’ comes from the previous step, i.e. the ‘rg’ is both overscan and bias subtracted

and has the MDF attached.

3. Lastly, we must tackle the fiber identification. Your command window is going to ask you a few questions:

```
Extracting slit 1
```

```
Find apertures for ergS20151108S0053_1? ('yes'):
```

Here I recommend saying ‘yes’, as usually IRAF will be able to identify the fibers correctly for you.

```
Edit apertures for ergS20151108S0053_1? ('yes'):
```

Say ‘yes’ to this one as well, so we can inspect the fiber IDs and make sure everything looks correct. [Figure 3.2](#) shows what this will look like – a complete mess! But we can zoom in by typing **w** in the PyRAF window, and then typing **e** at the bottom-left of the first ‘fiber bundle’ and again at its top-right. [Note: use **w** and **a** to un-zoom the window.] You should now have a screen that looks like [Figure 3.2](#). I find that making this plot full-screen helps immensely. Each ‘bundle’ contains a set of 50 fibers, which you will see numbered at the top of your screen (these are hard to see in [Figure 3.2](#)). Your job is to make sure each fiber is identified correctly – this means no fiber is identified twice, and any bad fibers are skipped in the numbering. Often I find that the low level of fiber 50 makes it get skipped in IRAF’s auto-finding scheme, which makes the first fiber of the second ‘bundle’ start at 50 instead of 51. If this happens, you can use **d** in the plot window to delete the IDs for fibers 50–750, and then use **m** to re-mark the fibers. This is where it can get frustrating!

[Figure 3.3](#) shows an example of a bad fiber not being marked in the identification process. This is exactly what we want to happen in this case. If you find another fiber that looks bad, you will have to edit the MDF file to “turn off” that fiber number. For the 2014-2015B slit-1 data, you will end with either fiber 742 or 743 (the rest up to 750 are bad).

If you are pleased with your IDs, type **q**. More questions will appear in the terminal, but it will want your answer inside the PyRAF graphics window.

```
Trace apertures for ergS20151108S0053_1?
```

```
yes
```

```
Fit traced positions for ergS20151108S0053_1 interactively?
```

```
NO (the capitals will tell IRAF no to all)
```

```
Write apertures for ergS20151108S0053_1 to database
```

```
yes
```

```
Extract aperture spectra for ergS20151108S0053_1?
```

```
yes
```

```
Review extracted spectra from ergS20151108S0053_1?
```

```
NO
```

If you’re in the blue setup (1-slit mode), this is the end of IDing for you! If you’re in the red setup (2-slit mode), you’ll have to repeat this process for fibers 751–1500 (usually fibers 1493 and above are missing for this group). Once complete, IRAF will extract all the spectra and create a file with the prefix ‘e’, so in full you should have an ‘ergS....fits’ file in your working directory. Since we have two different wavelength dithers, you’ll have to repeat this process for the second flat, but after that, we’ll use these files to identify the fibers for the rest of the bunch.

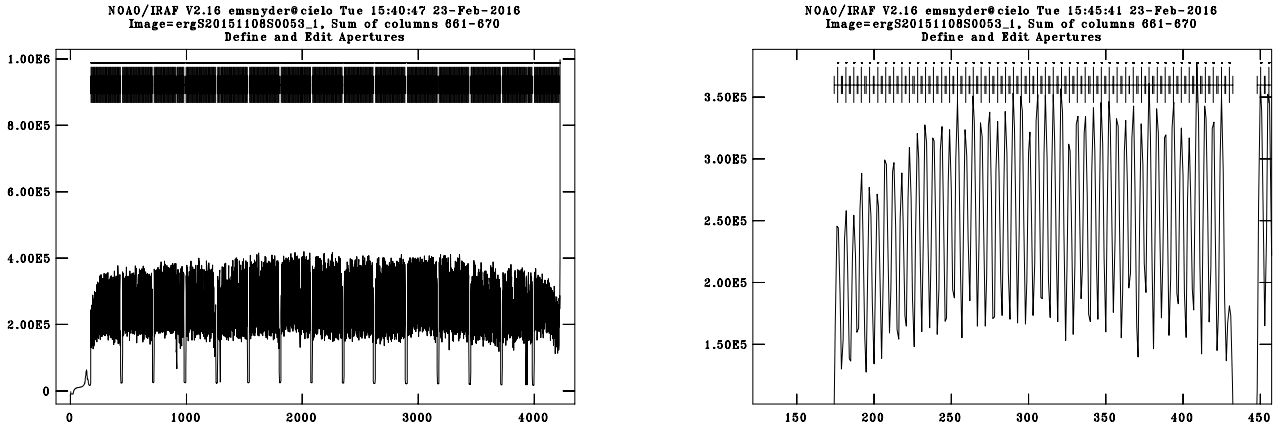


Figure 3.2: **(a)**. A zoomed-out example of the fiber ID window that PyRAF will open during the reduction of the flats step. The numbers above each fiber are blurred together here. **(b)**. An example of the fiber ID window that is zoomed-in on one set of fiber bundles. The numbers for each fiber aren't visible in this example but will be above the aperture marks at the top.

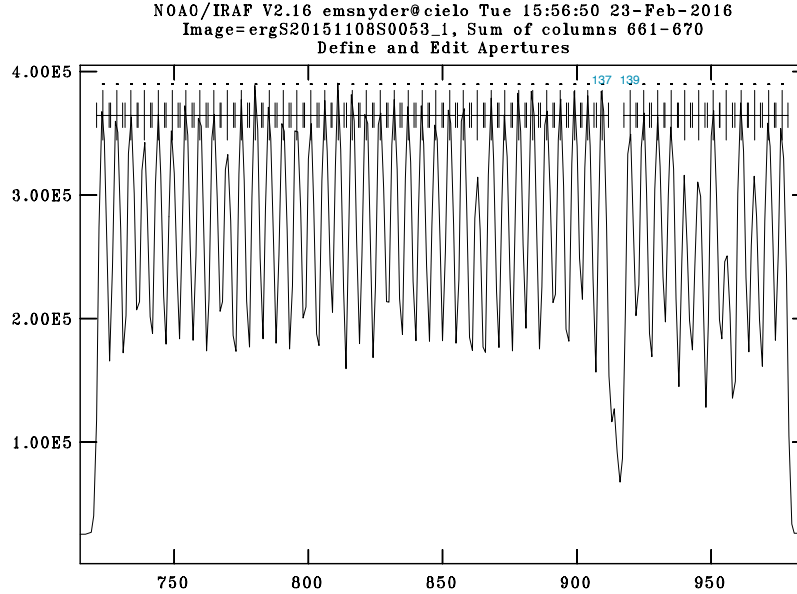


Figure 3.3: For this image, fiber 138 is bad. However, the MDF shows that $\text{BEAM} = -1$ for this fiber number, and thus that number is skipped during the identification routine.

3.5 Overscan Subtraction of the Arcs

Gemini IRAF task used: `gfreduce`

Similarly for the flats, we are using this package to attach the MDF file and do an overscan subtraction. We do not perform a bias subtraction on the arcs because they are read out from the CCD faster than our other data, meaning the read noise for them will be different than what's in our bias frame. We still do the overscan subtraction as a rough estimate of the bias level to subtract. There is nothing interactive for this step.

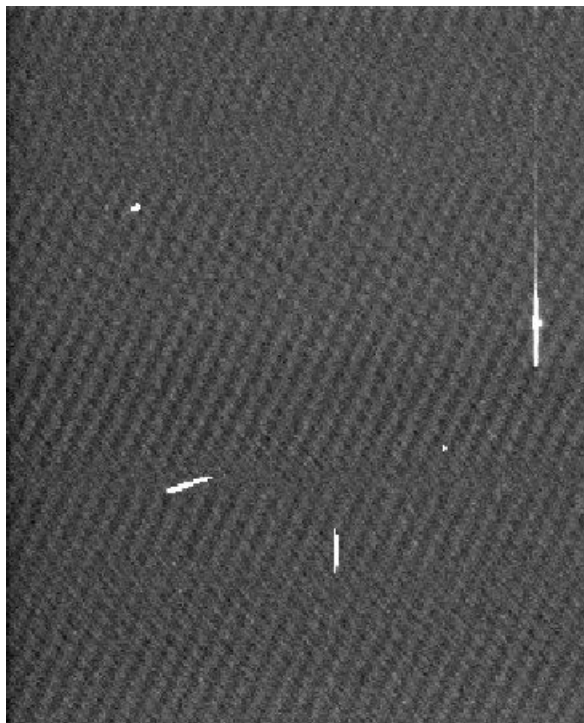


Figure 3.4: This image shows you both cosmic rays and bad pixels. The long line on the right side is a bad column, whereas the other three bright spots are probably cosmic rays.

3.6 Identification and Removal of Bad Pixels from the Flats and Arcs

Gemini IRAF tasks used: `addbpm` and `gemfix`

Before we get too into the reduction, we must look at our bias-subtracted (or overscan-subtracted in the case of the arcs) images and find any hot pixels or bad columns on the CCD itself that could later affect the reduction. The bad pixels are usually different for images taken at different exposure times since bleeding can occur, so we'll need to create a bad pixel map (BPM) for the flat, arc, and science frames separately. A BPM is a fits image that has the same dimensions and number of extensions as the arc/flat/science image, but the pixel values are only zeros and ones. The zeros denote where the pixels are good, and the ones flag where the pixels are bad. A useful caveat to bad pixels or columns on a CCD is that they will normally not change quickly, so it is okay to flag the same pixels for data sets taken in the same semester. So, it may save you some time to copy down the coordinates you find in this step to reuse for the next galaxy.

The pipeline starts by creating an image of the correct size that's entirely made of zeros. Now, we must identify the bad pixels in our data, making note of their x and y positions and which extension they are on. The pipeline will open a DS9 window with one of the flats first. It will then prompt you to look through all twelve extensions and identify any bad pixels or columns you see. If you don't see any bad pixels, you can enter `0` and move on to the arcs. If you do see some bad areas, enter `1`, which will bring up another prompt for you to enter the extension number, and x and y values of the bad region. The extension number can be 1-12, and x and y can extend to the full size of your image. To enter a bad rectangle, you can say `x = 132:140` and `y = 444:450`. For a full column, you can enter `*` for y.

After feeding the pipeline the coordinates of the bad pixels, it will change the BPM values to 1 from 0 at those locations. We then use the task `addbpm` to attach the BPM to our data file, so that `gemfix` can find it and interpolate over the bad pixels.

It is important to discern between cosmic rays and bad pixels/columns too. I usually find that cosmic rays will be curved and erratic looking, while actual bad columns will be in straight lines and will be much larger. See § 3.6 for an example. Generally, you’ll have the same bad areas in all of your images, but the extent of the damage due to these pixels can vary due to exposure time. So, if you’re unsure of bad pixels vs. a cosmic ray, look in the other images to see if there are bad pixels in the same spot. We will clean the cosmic rays from the science data in a separate step. You will repeat this for an arc image, and later in the reduction will do this for the science frames as well.

3.7 Extraction of the Arcs

Gemini IRAF task used: `gfextract`

This step will use the fiber IDs you created from the flat images to extract the spectra from the arcs. Extracting takes out the sky spectra, reorders the science spectra, and arranges them by slit. This means our output files will go from having 12 extensions to having only one or two, depending on whether you’re working with 1- or 2-slit data. There is nothing interactive for this step, and all extracted spectra will have a file name with ‘e’ in front.

3.8 Creation of the Wavelength Solution

Gemini IRAF task used: `gswavelength`

Next, the pipeline calls `gswavelength` and a PyRAF window will open with an image that looks like Figure 3.5. This is a 1D spectrum of the arc lamp, which we’ll use to assign a wavelength to each pixel for our science data (this is called a “wavelength solution”). With the smaller line list that was included in your downloads, only ~ 15 of the emission lines will be chosen automatically. Press `f` to see the fit residuals, which should look something like Figure 3.6. The number we want to pay attention to here is the RMS. I try to keep this under $\sim 0.09\text{\AA}$, which, after some experimenting, I’ve found produces a good wavelength solution. You can also change the order of the fit using `order: #`. This default order is 4, which seems to work well most of the time.

At times, especially when going from slit 1 to slit 2, the first line on the left will be misidentified. You can delete lines with `d` and then re-identify them by hovering your mouse above the correct line, typing `m`, and then entering the correct wavelength. Refitting with `f` should then yield a better RMS value.

You will see on your PyRAF screen some text that looks like this:

Image Data	Found	Fit	Pix Shift	User Shift	Z Shift	RMS
ergS20151108S0106_001 - Ap 440	10/10	10/10	-0.00557	0.00522	1.82E-6	0.0749
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 441	10/10	10/10	-0.0793	0.081	1.36E-5	0.0731
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 442	10/10	10/10	-0.0694	0.0711	1.12E-5	0.034
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 443	10/10	10/10	-0.00833	0.00873	9.11E-7	0.0476
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 444	10/10	10/10	0.0222	-0.0225	-4.0E-6	0.07
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 445	10/10	10/10	0.0256	-0.0259	-4.7E-6	0.0896
Fit dispersion function interactively? (no yes NO YES) ('no'):						
ergS20151108S0106_001 - Ap 446	10/10	10/10	0.084	-0.0855	-1.4E-5	0.126
Fit dispersion function interactively? (no yes NO YES) ('no'): yes						

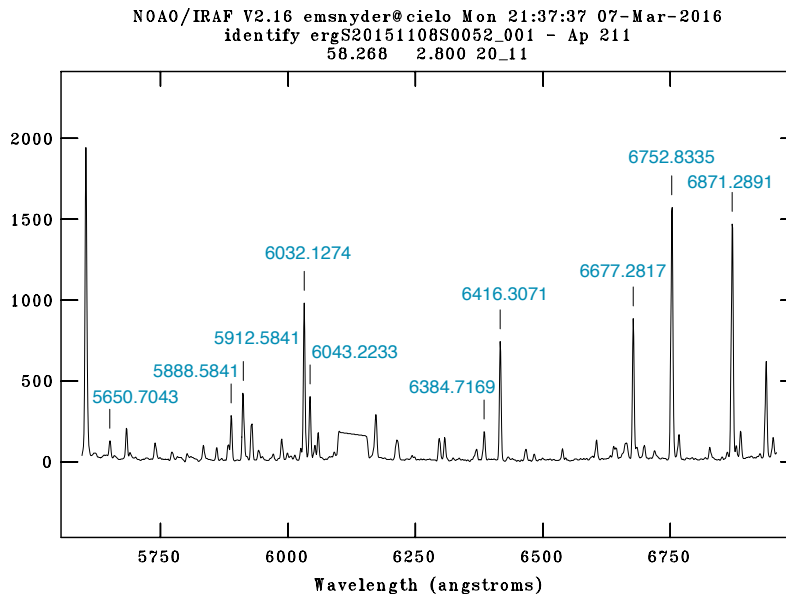


Figure 3.5: This image shows an arc spectrum. The arc lamps have certain unchanging emission lines, by knowing the wavelengths of those lines we can create a function that assigns a wavelength to all of our image pixels.

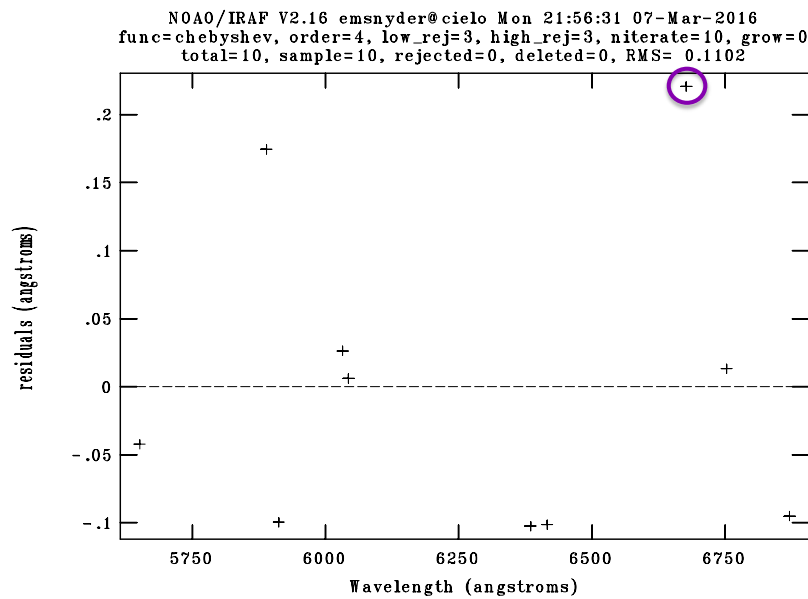


Figure 3.6: This image shows the residuals of the wavelength function fit. To improve this fit, I would delete, using `d`, the circled point. You can also change the order of the fit if the default (4) doesn't fit well enough.

My way of conquering this task is to answer `no` to each prompt until I see an RMS value that needs attention. For example, the last row in the above text is where I said yes to fit the wavelength function interactively when the RMS was 0.126. If the RMS is okay, I won't fit it interactively. You will need to do this for each fiber. The new output for this step are files in the database named "idergS....001" and ".002" (if in 2-slit mode).

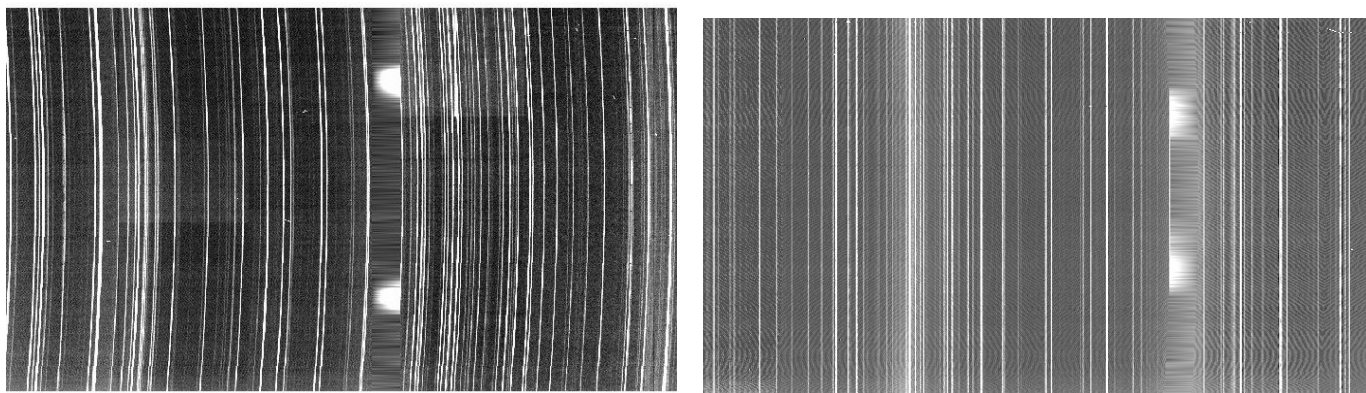


Figure 3.7: (a). A DS9 view of the arc before the wavelength transformation has been applied. (b). The same, but after the wavelength transformation has been applied.

3.9 Application of the Calibration to the Arcs

Gemini IRAF task used: `gftransform`

This step simply applies the wavelength solution you just found to the arc lamps itself. A good way to check our work here is to make sure the arc lines in the output image (which will start with the prefix “terg”) are straight, as opposed to curved. See [Figure 3.7](#) for a before and after. If everything looks good, we can use this same solution for our science data as well. If something looks wrong, an easy way to fix this is to delete the ID files created in the previous step along with the “terg” files, and start again. If you’re in 2-slit mode and see that only one of the slits is bad, you can delete only the bad one of these files (either “_001” for the first slit and “_002” for the second slit), and move the other one to a temporary file name. For example, if you find slit 2 is bad for one of the arcs, you can `rm` `IDergS...arc1.002`, and `mv` `IDergS...arc2.001` to a temporary file name. Now you can restart the pipeline, go to the wavelength calibration step, answer **NO** for slit 1 and redo the fits for slit 2. There will now be two new ID files, and you can replace `IDergS...arc1.001` with your temp file.

3.10 Quantum Efficiency Correction of Flats

Gemini IRAF task used: `gqecorr`

Some CCDs will have different quantum efficiency levels between CCD sections or amplifiers, so this step corrects for this effect. The files used for these corrections is in `gmos$data` (like the MDF) and is called `gmosQEfactors.dat`. Nothing interactive is required. This step creates files with the prefix ‘q’.

3.11 Re-extraction of the Flats

Gemini IRAF task used: `gfextract`

Now the flat spectra will be re-extracted, using the same fiber IDs as before, but this time pulling out the QE-corrected spectra. Again, nothing interactive must be done. After this step, our flats will have the prefix ‘eqprg’.

3.12 Bias and Overscan Subtraction of Science Data

Gemini IRAF task used: `gfreduce`

Next, we perform the bias and overscan subtraction of the science frames. Nothing interactive is needed here.

3.13 Identification and Removal of Bad Pixels from the Science Data

Gemini IRAF tasks used: `addbpm` and `gemfix`

We now make the BPM for the science data. See § 3.6 for more information on this step. A file with the prefix “p” is made here.

3.14 Cosmic Ray Rejection

Gemini IRAF task used: `gemcrspec`

Next, we use L.A. Cosmic to find and remove cosmic rays from our science spectra. The code will iterate many times, and may take a few minutes to complete. Nothing interactive is required, and a file with the prefix ‘x’ is created.

3.15 QE Correction of Science

Gemini IRAF task used: `gqecorr`

We now QE correct the science frames, much like we did for the flats in § 3.10. Again, nothing interactive is needed here, and you should have a file made with the prefixes “qxprg”.

3.16 Flat Fielding and Extraction of Science

Gemini IRAF task used: `gfreduce`

Next, we use the response functions made from the twilights to flat field our science data, and then extract it. Nothing interactive is needed here, and a file with the prefix “e” is made.

3.17 Wavelength Calibration of Science

Gemini IRAF task used: `gftransform`

In this step, we apply the wavelength solution we found in § 3.8 to the science frames. There is nothing interactive for this step, and file is created with the prefix ‘t’.

3.18 Sky Subtraction of Science

Gemini IRAF task used: `gfskysub`

Now, we can pull out the spectra of the sky fibers and use them to subtract the scattered sky light from the science spectra. A PyRAF window will open that looks like the left image in Figure 3.8. The peaks that we see are probably cosmic rays that were not well subtracted, so if you hover your mouse near the peak and type `d`, we can remove these artificial peaks. You can redraw the spectra after deleting using `r`. When done, you should see an image like the right frame of Figure 3.8. A file with the prefix “s” is made here.

3.19 Flux Calibration of Science

Gemini IRAF task used: `gscalibrate`

The second to last step of the reduction uses the flux responses from the standard star data to flux calibrate our science data. Nothing interactive is needed here, and a file with the prefix “c” is created.

3.20 Data Cube Creation

Gemini IRAF task used: `gfcube`

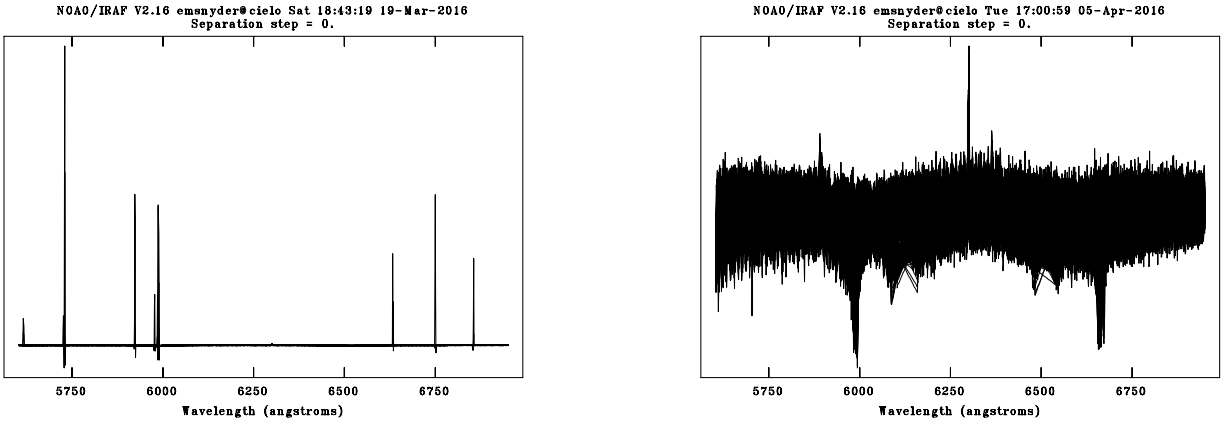


Figure 3.8: **(a).** An example of the sky subtraction PyRAF window you’ll see initially. **(b).** An example of the remaining spectra after the peaks have been deleted.

Now that we’re nearing the end of the reduction process, we resample our spectra into data cubes. I choose to sample the data into $0.2''$ pixels, since that’s the size of the fiber, and it will make the next step (making the WCS easier). There will be as many data cubes as there are science images. We use the task `gfcube` map the 2D images into 3D data cubes that we can later sum in the spectral direction to fill the chip gaps and increase the signal to noise ratio. If there are spatial offsets, we can use the offsets in noted in the image headers to align and sum these as well (detailed in the next steps). Files with the prefix “d” are created here.

3.21 Creation of a WCS Solution

Before we can sum the data cubes, we need to assign a coordinate system to the individual cubes. To do this, we first use the information in the headers of the first extracted science file, which will have the final telescope pointing coordinate system information. In the header, there will be CRPIX1 and CRPIX2 that give the reference x and y pixels, along with CRVAL1 and CRVAL2 that are the RA and Dec values at the reference coordinates. We look inside the extracted science file for this information because we want to be able to assign an RA/Dec value to a certain fiber number. From there, we can use the spatial position of that fiber (given in our MDF) along with position angle of the galaxy observation to assign an RA/Dec to every other fiber.

A complication to this is that there an offset from the telescope pointing position to the IFU field of view. Thus, we need to correct the CRPIX1 and CRPIX2 values to reflect this small offset before assigning our coordinates. Thanks to the Gemini-South data reduction gurus James Turner and German Gimeno, we have these offsets and the pipeline can correct for this automatically.

3.22 Summation of the Data Cubes

Now that the data cubes have image headers with accurate WCS coordinates, we can use the tasks included in the PyFU package to mosaic the cubes together. The Python script `pyfmosaic` looks into the headers of the supplied input data cubes and pulls out both the WCS information and the spectral offsets. It will then rebin the spectra, preserving the flux in each pixel, into one final data cube, named `galaxy_final.fits`.

Chapter 4

Derivation of Kinematics

4.1 Deriving Velocity Dispersions

4.2 Deriving Rotation Curves