Appendix: 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

List of Figures

List of Tables

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 RESOLVE[1] 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 ($\sim 0.5-2''$ 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 setup is best for which galaxy, 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 always a

[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	Hβ, Mgb, Fe5270, Fe5335	Hα, [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.

SOAR (or even an SDSS) spectrum available, we often rely on 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 Calculators[2] (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 Hα 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 ~ 5.3 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. See Table 1.2 for the basic order of the observations. 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 wavelength dither (denoted as λ_{cen} in Table 1.2) . 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. For the 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

λ_{cen}	blue setup		red setup	λ_{cen}
–	acquisition images		acquisition images	–
4850 Å	galaxy flat arc	time ⟶	flat arc galaxy	6300 Å
4900 Å	arc flat galaxy		galaxy arc flat	6400 Å

Table 1.2: Comparisons of the order in which the data for each galaxy is observed in each setup.

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 Sheila[3], 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 in § 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'll next want a bias image in order to perform the bias subtraction. The process for making the bias frame is explained in § 4.1.

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 a 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. See § 4.2 for how to create the response function and flux calibration files.

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`

[3] `sheila[at]physics[dot]unc[dot]edu`

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 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 (good fibers will have BEAM=1 while bad will have BEAM=-1). See Figure 3.3 for an example of what good/bad fibers look like and see § 4.3.1 for an example of why you may need to use/change this information.

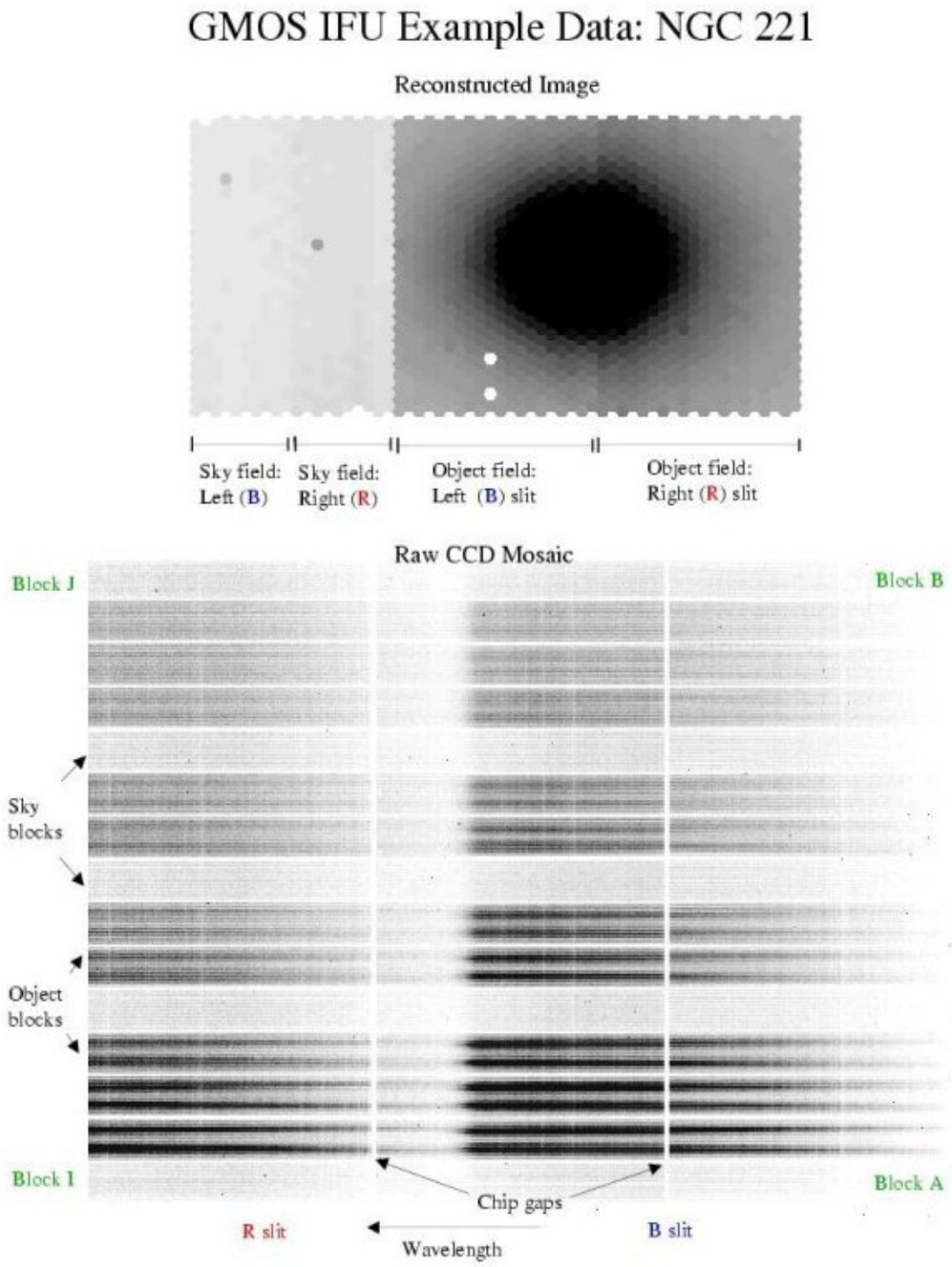

Figure 1.1: The top part of this image shows the IFU field of view over NGC221 (not a part of RESOLVE). The left portion of the image shows the sky spaxels, and the right portion shows the light of the galaxy. This is a 2D representation of the data we are getting – the third dimension is in the wavelength 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

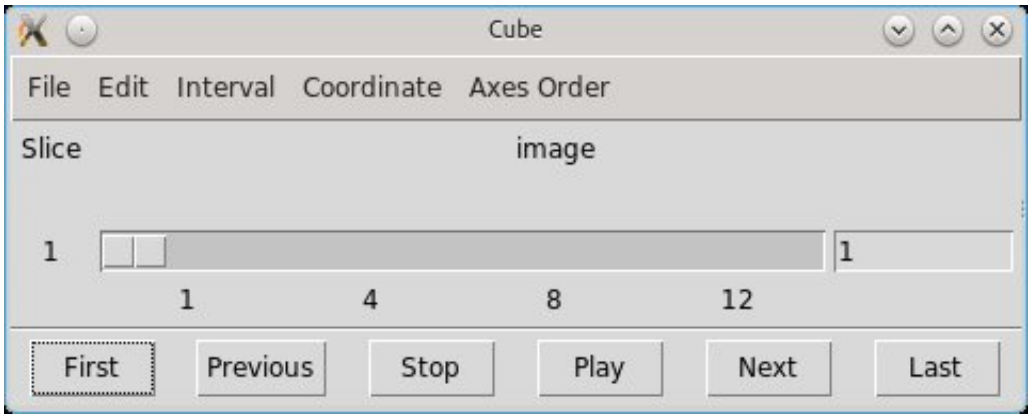

Figure 1.2: This is an example of the cube box that will open when you load a raw image in DS9.

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.

Chapter 2

Setting Up Your Workspace

There are some things you'll need to install (or update) before using this pipeline. The author highly recommends working in the Ureka environment, which includes easy-to-install versions of IRAF, Python, and PyRAF. Since this code is written for PyRAF, Ureka is indeed quite handy. Instructions on how to download Ureka can be found here: http://ssb.stsci.edu/ureka/. This pipeline was written and test using Ureka version 1.5.1 on cielo. Follow the instructions for making IRAF, and check your login screen to make sure it says you're using IRAF 2.16:

```
> pyraf

NOAO/IRAF PC-IRAF Revision 2.16 EXPORT Thu May 24 15:41:17 MST 2012
  This is the EXPORT version of IRAF V2.16 supporting PC systems.
```

There is also an updated IRAF Gemini package you'll want to install. This version is not included in the Ureka download, but includes some updated tasks you will be using. Here's the link to this newest version as of February 2016: https://www.gemini.edu/?q=node/11823. You'll also want to check this when you're done by loading the 'gemini' package in IRAF:

```
--> gemini

     +------------------- Gemini IRAF Package -------------------+
     |              Version 1.13.1, December 7, 2015              |
     |              Requires IRAF v2.14.1 or greater              |
     |               Tested with Ureka IRAF v2.16                 |
     |              Gemini Observatory, Hilo, Hawaii              |
     |    Please use the help desk for submission of questions    |
     |  http://www.gemini.edu/sciops/helpdesk/helpdeskIndex.html  |
     +------------------------------------------------------------+
```

This pipeline also makes use of a PyRAF script called PyFU which was written by James Turner, a support scientist at Gemini. The code will align and sum the data cubes we make during the reduction process. The package is available at the Gemini Data Reduction Forum (http://drforum.gemini.edu/topic/pyfu-datacube-mosaicking-package/), and comes with instructions on how to install and point to the package from your login.cl file.

You may also need to download L.A.Cosmic, an algorithm that finds and removes cosmic rays. You can read more about it at http://www.astro.yale.edu/dokkum/lacosmic/. You can download the IRAF spectroscopic version from that website, and will need to add a line in your login.cl: "task lacos_spec = /path/to/where/you/put/it/lacos_spec.cl". The pipeline will call this task during the cosmic ray removal step.

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.

You will now 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.

Next, you'll want data to actually reduce. You should have flats, arcs, science frames, and a bias image (see § 4.1) before getting started. You should also have response flats from the baseline twilights and a flux calibration from the standard star data (§ 4.2 leads you there creation of these if you don't already have them). In § 1.4 there is 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 – your data package from Gemini should come with one, and for RESOLVE, they are called obslog.txt or just log.txt. The author has also created an alternative line list file called "smalllinelist.dat" that will be helpful during the wavelength calibration steps. This can be found in the `github` folder as the pipeline.

The current Gemini data are located on cielo at `/srv/two/sheila/emsnyder/gemini/data/`. From there, the folders are divided into different semesters: either 2013B, 2014B, or 2015B. In each of these folders, the galaxy data is further divided into individual folders titled the galaxy's name. Each of these folders should contain an observation log, raw flats, arcs, and science frames. There should also be folders for the baseline standards – usually these will be named LTT#### or H###, after the stars we observed for our standard. These baselines may or may not be reduced already, and so if they are not, see § 4.2. If they are, just copy the needed files (usually named `*_resp.fits` for the response files and `sfunction_*.fits` for the flux calibration files) into your galaxy folder before starting. There should be two – one for both the red and blue setups. You can use the same baselines for each galaxy in that semester. There may also be a bias frame already in your galaxy folder, but if not, see § 4.1 for how to create one before getting started.

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, although the tasks are typically "under the hood" of the pipeline. 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 galaxy 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 (see § 4.3.1 for why this may be needed). If you are, be sure to give the full path to that file, and if not, just type `default`.

Next, the pipeline will ask if you are using a special MDF. If not, you can just enter `default` to use the standard file. The pipeline will then ask you if you have a bad pixel map already. If so, enter the path to the file, and if not, just enter `0` to create one later in § 3.7.

Lastly, 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/bias and which central wavelength they were observed at. It is good at this very first step to open up your raw images in 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 Bias + Overscan Subtraction and Trimming of the Flats and Fiber Identification

Gemini IRAF task used: `gfreduce`

This very first step can often (in the author's experience) be the most time-consuming 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 2D 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, image trimming, and fiber ID so that we may extract all the spectra. We will later re-extract the flats after performing a quantum efficiency correction, but we need to identify the fibers and ensure our MDF is correct before attaching it to other files.

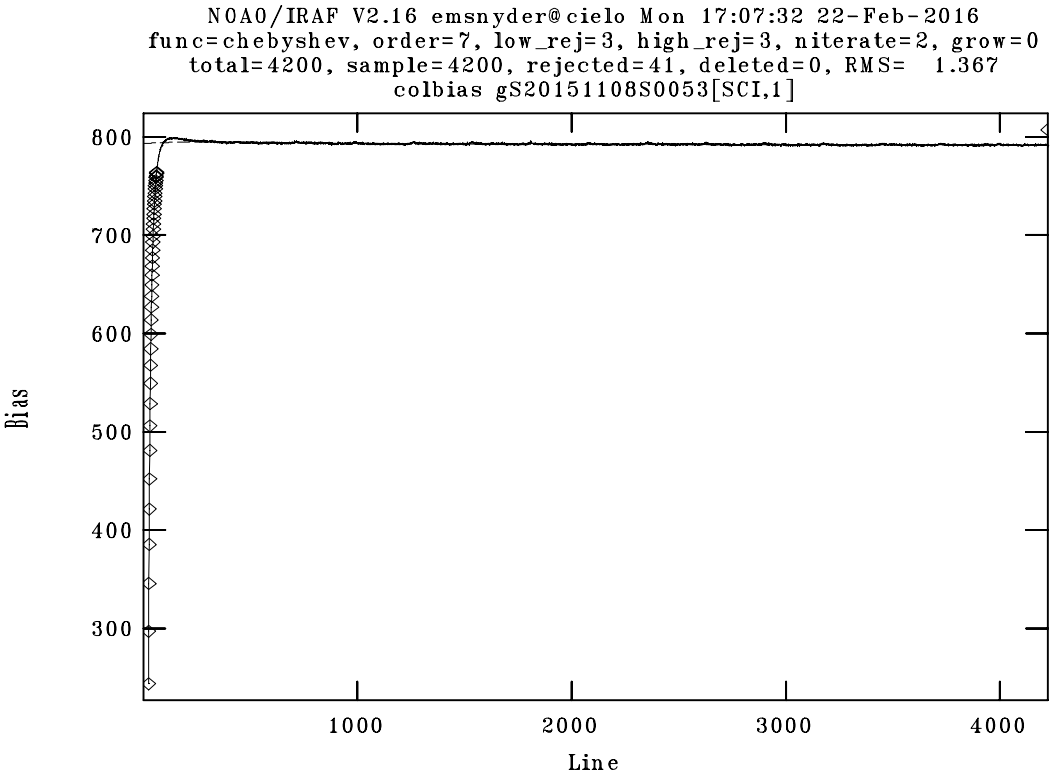

Figure 3.1: An example of the overscan subtraction window that PyRAF will open during the reduction of the flats step. The solid line represents the data, while the dashed line shows the fit to that data. The diamonds are data points that are being rejected from our fit.

1. The MDF attachment is done without user input needed. Corresponding to each input flat file, a new file is created in your working directory with the prefix 'g'. A useful way to remember this could be "g" is for "Get the MDF"
2. The bias and overscan subtraction and imaging trimming is mostly done without user input needed. The overscan regions are the regions of the image with no data, i.e., between the chips and amplifiers. We want to measure the count level in these regions and then trim them from our images. Note that we also trim the outer edges of the image where there is no data in this step too, but we are not measuring the counts there for the overscan subtraction. 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 and the "r" is added for the "subtRaction/tRimming" (this one is a stretch – sorry).

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. The left side of 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 the right side of 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 fiber numbers are skipped (not used) in the numbering. The fibers should be numbered from 1-750 or 1500, from left to right. 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 including the low-level fiber. This is where it can get time-consuming.

   Figure 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, for example, fiber 138 was still "on", you would want to edit the MDF to turn it "off" (i.e., set BEAM=-1 for that fiber number). See § 4.3.1 for how to do this, and for more info on how this step can go wrong. For the 2014-2015B slit-1 data, you will end with either fiber 742 or 743, depending on how the telescope was rotated at the time of the observation.

   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?
   ```

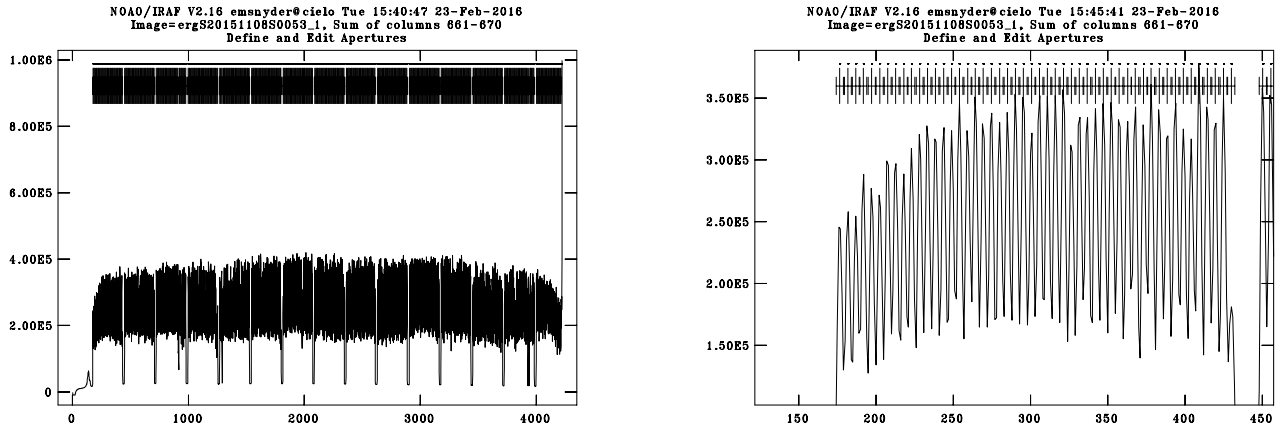

Figure 3.2: **(a).** A zoomed-out example of the fiber ID window that PyRAF will open during the reduction of the flats step. The numbers above each fiber are blurred together here. **(b).** An example of the fiber ID window that is zoomed-in on one set of fiber bundles. The numbers for each fiber aren't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

```
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" (for "Extracted"), so in full you should have an 'ergS....fits' file in your working directory. 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. 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 "erg" flat files to identify the fibers for the rest of the files (including the arcs and science).

3.5 Overscan Subtraction and Trimming of the Arcs

Gemini IRAF task used: `gfreduce`

Just as for the flats, we now attach the MDF file, do an overscan subtraction, and trim the arcs. 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 and we don't worry about any bias structure, which should be irrelevant if the arc lines are strong. Therefore, there is nothing interactive for this step. New files will be created with the prefix "rg" in your working directory (they are not yet extracted).

3.6 Bias + Overscan Subtraction and Trimming of Science Data

Gemini IRAF task used: `gfreduce`

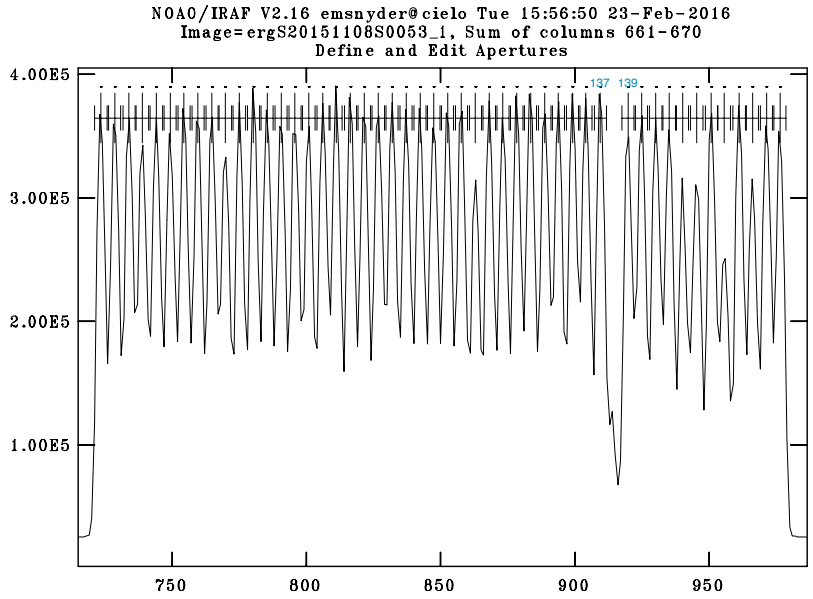

Figure 3.3: For this image, fiber 138 is bad. However, the MDF shows that BEAM = -1 for this fiber number, and thus that number is skipped during the identification routine.

Next, we perform the bias and overscan subtraction and trimming of the science frames. Nothing interactive is needed here, but again files will be made with the prefix "rg".

3.7 Identification and Removal of Bad Pixels from the Science, Flats, and Arcs

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

The pipeline prompted you at startup to enter the name of a bad pixel map (BPM) file. If you didn't give a file at the prompt, read on and you will be lead through what a BPM is and how the pipeline will help you make one.

Before we get too far into the reduction, we must look at one of the bias- and overscane-subtracted and trimmed science images (aka, the "rg" file made in § 3.6) 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 with different exposure times since bleeding can occur, so we'll create a BPM for the science frames only since they have the longest exposure time. We also look at the bias frame itself, since any hot pixels should stand out better in it than in the science frames since cosmic rays will be everywhere in the science. A BPM is a fits image that has the same dimensions and number of extensions as the 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, if you've already created a BPM for a galaxy in the same semester, you can enter its name at the pipeline prompt to use it. If this is your first galaxy of the semester, the pipeline will now lead you through how to make one.

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 science and bias images, 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 science frames (the "rg" file made in § 3.6) and the bias frame. The pipeline 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

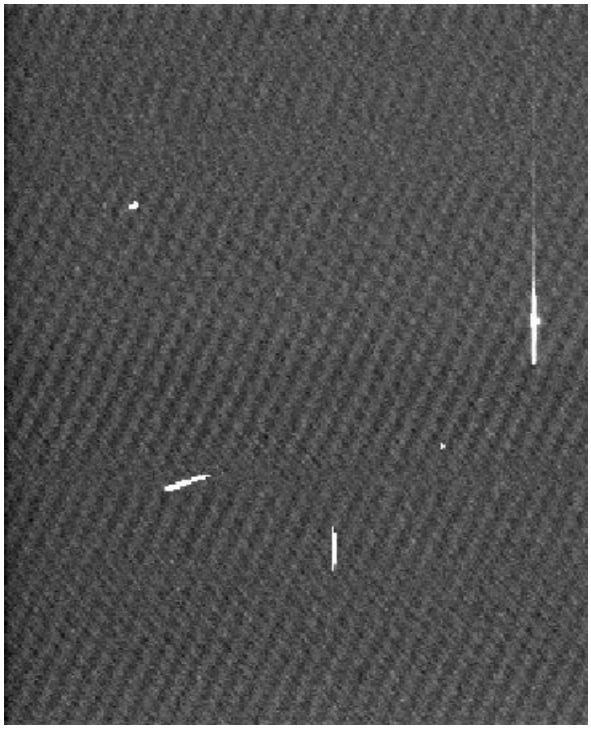

Figure 3.4: This shows you both cosmic rays and bad pixels in one of the science frames.. The long line on the right side is a bad column, whereas the other four bright spots are probably cosmic rays. These are not "bad pixels" because they change in every image.

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.

It is important to discern between cosmic rays and bad pixels/columns too. The author usually finds that cosmic rays will be curved and erratic looking, while actual bad columns will be in straight lines and may be much larger, although the extent varies so single hot pixels are also possible. See § 3.7 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.

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. The pipeline will then use this BPM to correct the science, arc, and flat frames.

Note that in this step we are applying this correction to the images with prefixes "rg". This step adds the letter "p" for "fixing Pixels".

3.8 Extraction of the Arcs

Gemini IRAF task used: `gfextract`

We now extract the spectra from the BPM-corrected arc images using the fiber IDs that were created from the flats. There is nothing interactive to be done, but file will be made with an "e" in front for the extracted arc spectra (the full prefix is now "eprg").

3.9 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, and we'll use it to assign a wavelength to each pixel for our science data (this is called a "wavelength solution"). You should see that some of the emission lines are being automatically marked with a | above them.

A quick aside about the lines being automatically ID'd: if you aren't using the "smalllinelist.dat" that was included in the download of the pipeline or if it isn't in your working directory, the pipeline will use the default line list from Gemini, which is called `GCALcuar.dat` and is located in the `gmos$data` directory (something similar to `~/iraf/gemini/gmos/data/` outside the PyRAF environment). The difference between these two lists is that the author has taken the strongest lines from the default list and placed them into "smallinelist.dat". Using this list with the strongest lines only should save you time as you perform each fit since these weak lines would most likely need to be deleted anyways. When using "smalllinelist.dat" you should see only $\sim$ 10 lines identified automatically.

Press `f` to see the fit residuals, which should look something like Figure 3.6. The number to pay attention to here is the RMS. I try to keep this under $\sim$ 0.09Å, which, after some experimenting, I've found produces a good wavelength solution. To delete bad points, type `d` and then press `f` to refit. The RMS should update after pressing `f`. You can also change the order of the fit using `order: #` and then `f`. The 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 (see Figure 3.5). Refitting with `f` should then yield a better RMS value.

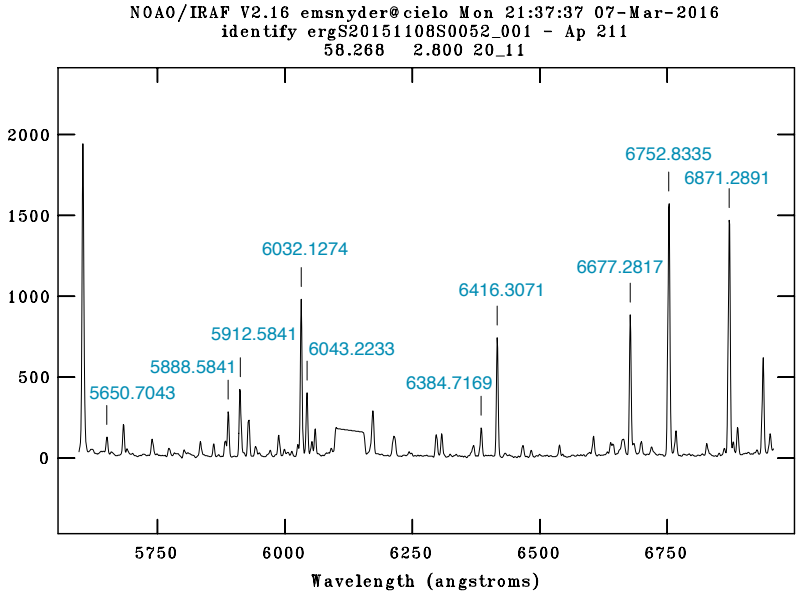

Figure 3.5: This image shows an arc spectrum. The arc lamps have certain unchanging emission lines, by knowing the wavelengths of those lines we can create a function that assigns a wavelength to each of our image pixels.

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

```
Image Data    Found          Fit   Pix Shift    User Shift Z Shift      RMS
```

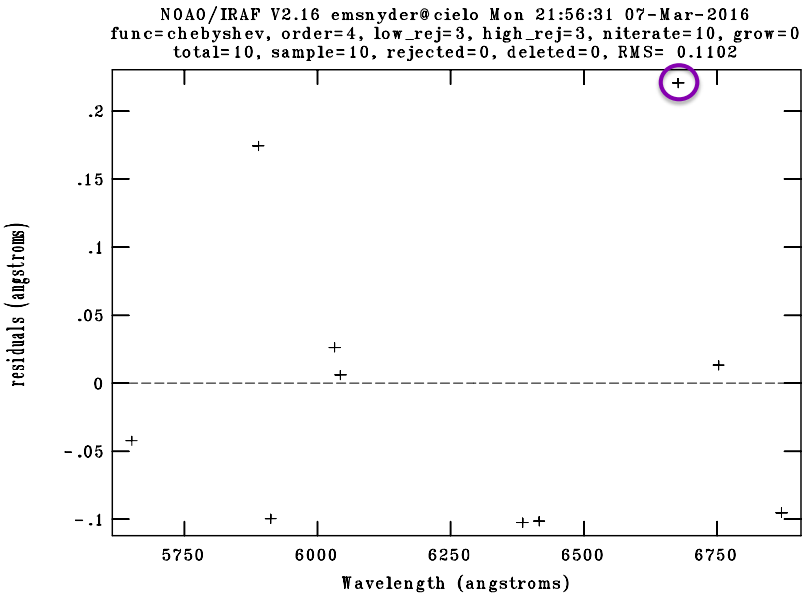

Figure 3.6: This image shows the residuals of the wavelength function fit. To improve this fit, I would delete, using `d`, the circled point. You can also change the order of the fit if the default (4) doesn't fit well enough by typing `order #` and then pressing `f` again.

```
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
```

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 in each arc (usually just two per galaxy). The outputs for this step are files in the database named "idergS...._001" and "_002" (if in 2-slit mode).

3.10 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 "t" for "Transformed") are straight, as opposed to curved. See Figure 3.7 for a before and after. The pipeline will open a DS9 window with the transformed arcs so that you can check for any

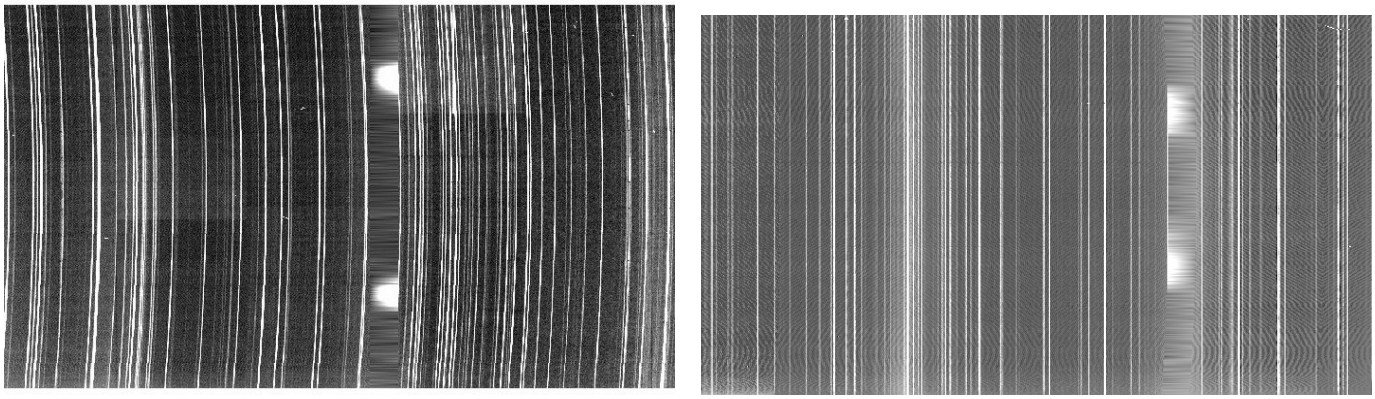

Figure 3.7: A DS9 view of the arc **(a.)** before and **(b.)** after the wavelength transformation has been applied.

errors. If everything looks good, we can move on. See § 4.3.2 for an example of a bad wavelength transformation and for information on how to correct issues or start over if needed.

3.11 Quantum Efficiency Correction of the 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 file 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" for "Quantum efficiency corrected".

3.12 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 BPM- and QE-corrected spectra. Again, nothing interactive must be done. After this step, our flats will have the prefix "eqprg". The flat images with prefixes "erg" will no longer be used from now on.

3.13 Creation of the Response Functions

Gemini IRAF task used: `gfresponse`

Next, we use the twilight flats from our baseline standard data and our newly extracted flats from § 3.12 to create a response function to use for flat fielding the arcs and science data. This response function takes into account three difference effects: pixel-to-pixel variations in the CCD itself, the wavelength-dependent pixel efficiency (i.e., pixels being more responsive to blue light than red light), and illumination variations (corrected using the twilight flats). There is nothing interactive for this step, and the output will be named "eqprgS..._resp.fits". See § 4.2 if you need to make your twilight flats.

3.14 Quantum Efficiency Correction of the Arcs

Gemini IRAF task used: `gqecorr`

We now correct for the quantum efficiency differences in the arc images, as we did for the flats in § 3.11. Again, nothing interactive is required.

3.15 Flat Fielding and Extraction of the Arcs

Gemini IRAF task used: `gfreduce`

And now we use the response functions made from the twilights to flat field the arcs, and then re-extract them. Nothing interactive is needed for this step, and a file with the prefix "e" is made.

3.16 Re-creation of the Wavelength Solution for the QE-Corrected & Flat Fielded Arcs

Since the QE correction and flat fielding can change the pixel values in the arcs, we now need to redo our wavelength solution for the arcs. The steps will be the same as in § 3.9. The reason we must do this twice is because we need a wavelength solution in order to perform the QE correction. Future work may include finding a way to automate these wavelength solution steps.

3.17 Application of the Calibration to the QE-corrected & Flat Fielded Arcs

Gemini IRAF task used: `gftransform`

Again, we must apply the new wavelength solutions to the QE-corrected and flat fielded arcs. The pipeline will open the transformed arcs so that you may check for a good solution.

3.18 Cosmic Ray Rejection in the Science Data

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 for "eXpunge cosmic rays".

3.19 QE Correction of Science

Gemini IRAF task used: `gqecorr`

We now QE correct the science frames, as we did for the flats and arcs in § 3.11 and § 3.14. Again, nothing interactive is needed here, and you should have files made with the prefixes "qxprg".

3.20 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.

3.21 Wavelength Calibration of Science

Gemini IRAF task used: `gftransform`

In this step, we apply the wavelength solution we found in § 3.16 to the science frames. There is nothing interactive for this step, and file is created with the prefix "t".

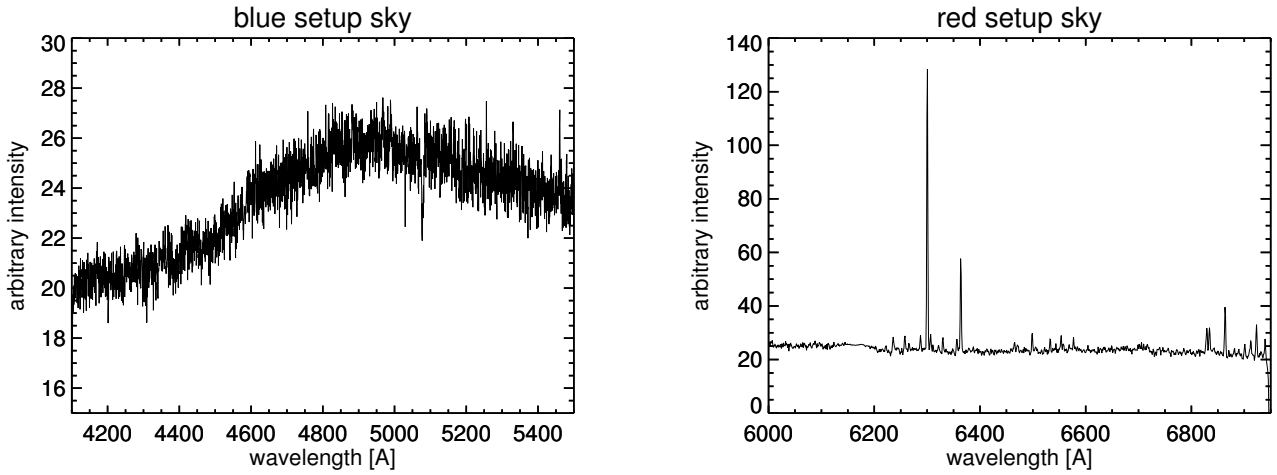

Figure 3.8: An example of the sky spectrum for **(a)** the blue setup data and **(b)** the red setup data. The blue setup data has no large sky lines, but this could vary depending on moon phase and cloud coverage during your observations. RESOLVE typically will not observe blue setup data when the moon is bright or weather is poor, so sky lines in the blue setup are rare.

3.22 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 and sky lines from the science spectra. This step is done non-interactively, but the basic process is that the task will take the median of all the sky spectra and subtract that from our science spectra. Doing a median of the spectra will ensure no spurious peaks from poorly removed cosmic rays will affect the subtraction. A file with the prefix "s" is made here for "Sky subtracted". The sky spectrum is appended to this newly made file under extension 2 or 3[1]. You can plot this spectrum in IRAF with `splot` outside of the pipeline, or load it into python. You should see spectra similar to those in Figure 3.8 for the blue setup (on the right) and the red setup (on the left). Also note that the intensity of sky lines may vary due to moon phase or cloud cover, so yours may look slightly different.

3.23 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" for "Calibrate flux" is created. If you do not have flux responses, see § 4.2 for how to make them.

3.24 Data Cube Creation

Gemini IRAF task used: `gfcube`

Now that we're nearing the end of the reduction process, we resample our spectra into data cubes. The pipeline will sample the data into 0.2″ pixels, since that's the size of the fiber and will make the

[1]Extension 0 is the MDF, 1 is the first slit, 2 is the second slit if in red setup, or the sky if in blue setup, and 3 will be the sky for the red setup

next step (applying RA/Dec coordinates to each pixel) easier. There will be as many data cubes as there are science images, meaning the spectra taken at different wavelength dithers or at different spatial positions are not yet merged. We use the task `gfcube` map the 2D images into 3D data cubes using the information from the MDF and our fiber IDs. See § 4.3.1 if you get an error here. Files with the prefix "d" are created here for "Data cube". You can open these outside the pipeline in DS9 and will get a 2D image in x and y with a dialog box that lets you "scroll" through the wavelength axis. You should be able to see in the blue setup data some continuum light, but most likely not much for the red setup, until you get to the wavelength of Hα, where you should see tons of emission!

3.25 Creation of a WCS Solution

Next the pipeline will assign a coordinate system to the individual cubes. This RA/Dec coordinate system is called a world coordinate system (WCS) solution. Data cubes taken in the same position but with different wavelength dithers will have the same WCS solution, but data cubes taken at different spatial offsets will have different ones.

To do this, we start by getting the information in the headers of the last acquisition image, which will provide us with starting WCS. 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. The pipeline then calculates the central x and y pixels of that image (i.e., the size of the image in the x and y directions, divided by two.) Now we need to account for the offset from the telescope pointing position to the IFU field of view, since the FOV isn't centered. Thanks to the Gemini-South data reduction gurus James Turner and German Gimeno, we have these offsets and the pipeline calculates the center pixels of the IFU by subtracting these offsets from the central x and y pixels of the acquisition image. We can then use the starting WCS to convert these pixels values into RA and Dec coordinates.

Now that we know the RA and Dec of the center of the IFU FOV, we can assign those coordinates to the center x, y pixel of the data cube. Knowing that each pixel is $0.2''$ in diameter and the PA at which the observation was taken, the pipeline calculates the RA and Dec for every other x,y pixel in the data cube. It lastly puts this information in the image header, so that the next step can find it.

3.26 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. Each individual cube may have slightly different spectral resolution (Å per pixel) since we created the wavelength solution for them separately, so PyFU will rebin the spectra to the same resolution and then sum each pixel in the spectra in the different cubes according to their spectral dithers. See **??** for a sketch of this process. Once the dithers are combined, PyFU will combine the spatial offsets using the WCS solutions from the image headers. The product is one final data cube, named yourgalaxyname_final.fits. The data reduction is complete!

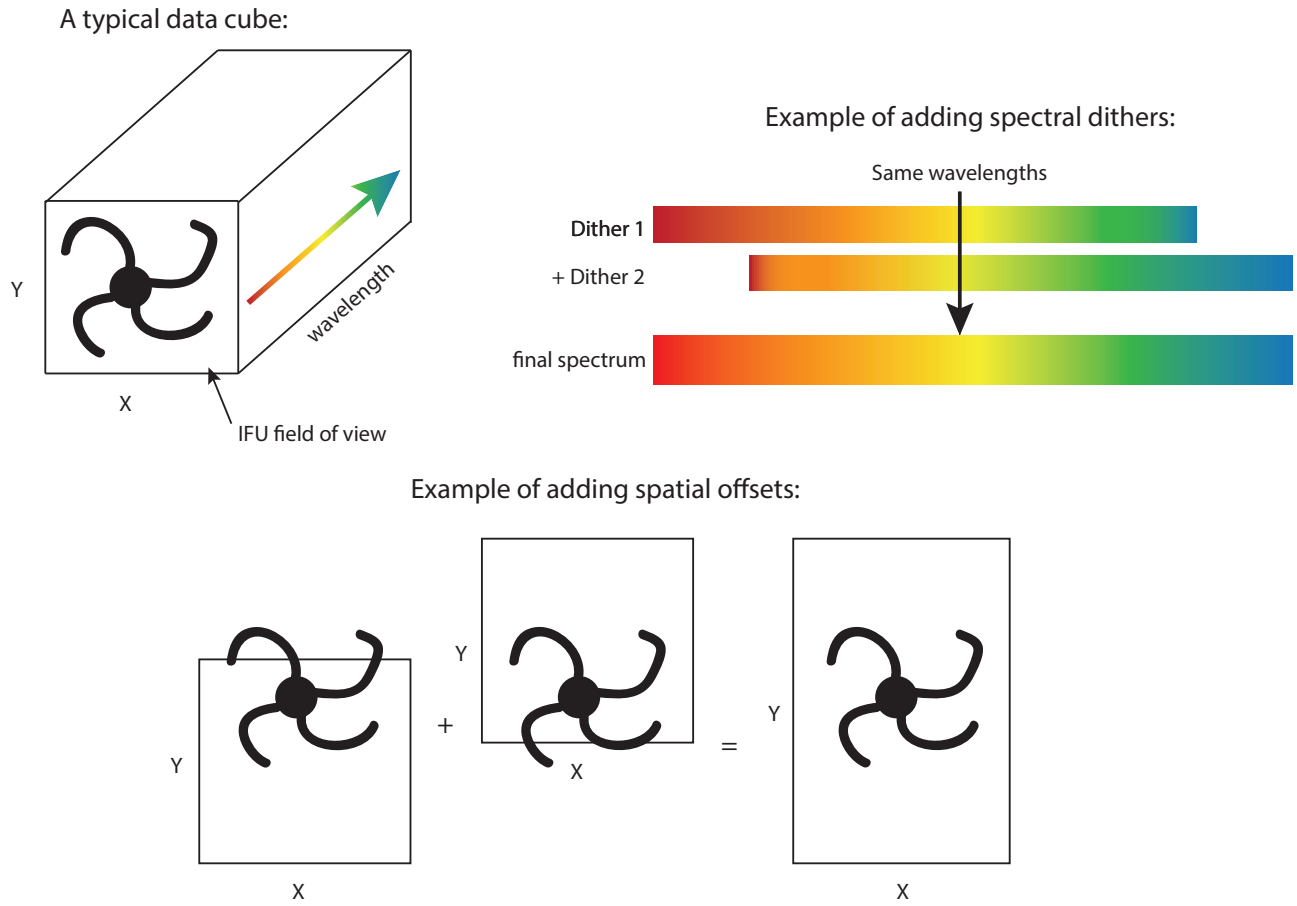

Figure 3.9: In this image, we show an example schematic of a data cube, and then examples of how the spectral dithers and spatial offsets are combined.

Chapter 4

Other Things You'll Want to Know

4.1 Creating the Bias Frame

Before starting the reduction, you'll want to make sure you have bias frame made, or individual biases that we can combine to make the bias frame. 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 Gemini Science Archive online database, but you can go to the "View Associated Calibrations" tab to find all of the biases taken for your observing program. 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`.

Outside of PyRAF, first download the frames from the database. These will come in tar file, and you can use `tar -xvf gemini_data.tar` to untar the files. The individual files are "bzipped" with the extension .bz2. Type `bunzip2 *.bz2` to unzip the files.

Now, enter PyRAF and type `gemini` and press enter, and then type `gmos` and press enter again. This loads the packages needed to access the task `gbias`. Now type `epar gbias` to open the PyRAF parameter editor. You should see a separate window open with `Task = GBIAS` at the top. Fill in the following boxes/checkmarks:

```
inimages = bias1.fits, bias2.fits, bias3.fits, bias4.fits, bias5.fits
outbias = <date>bias.fits
fl_over = yes
fl_trim = yes
```

Note that for `inimages` you can use wildcards, so for the biases taken on night Nov 11, 2015, you could also put `S20151111*.fits`, assuming you are working in a separate folder with no other files from that night in there. You can also name the output file (`outbias`) whatever you like, as long as it has the word "bias" in it since this is what the pipeline searches for. The author usually names hers with the observation date followed by "bias".

Then press `Execute` at the top of the dialog box to run the task. If everything runs smoothly, a bias frame will be output. Open this in DS9 to make sure it looks good. See Figure 4.1 for an example.

The author also notes that the since the flats, arcs, and science are all overscan subtracted and trimmed via the pipeline, the bias frame should be as well. You will get an error if your bias doesn't match the format of the flats/arcs/science! If for some reason you want to forego overscan subtracting and trimming, you will need to manually edit the pipeline to turn those features off.

4.2 Reducing the Twilight and Standard Star Data

The reduction process for the baseline standards is extremely similar to the process described above for the galaxy data, with a few added steps at the end. Here is a step-by-step primer of the changes:

1. Start up will be the same as in § 3.2, except that you will instead load the file "gemstandard-pipeline.py".

Figure 4.1: Example of the bias frame taken on the night of November 10, 2015. The near the edges of each amplifier in the CCD, the bias level rises.

2. The pipeline will ask you for your working directory, the standard star's name, it's shortened name, the MDF file, and BPM. The star name will probably be either LTT#### or H###, and this can be found in your log. The shortened name is for the LTT stars only, since the pipeline will search for its calibration data using the name L####. Everything is the same as in § 3.3. It will again print out the files it finds in your working folder, and you should be arcs, flats, science (this time a star instead of a galaxy), and also twilight flats.
3. Then the pipeline will take you through the bias/overscan subtraction, trimming, and fiber ID for the flats, just like in § 3.4.
4. It will then overscan subtract and trim the arcs, and bias/overscan subtract and trim both the science frames and twilight flats. These are detailed in § 3.5 and § 3.6. The twilights will be treated much like regular flats throughout this reduction.
5. Next the pipeline will either perform the BPM correction if you already have a BPM file made, or will lead you through how to make one. Details on this are in § 3.7.
6. Then the arcs are extracted (§ 3.8), and you will create the wavelength solution (§ 3.9), and apply the transformation (§ 3.10).
7. The pipeline then will quantum efficiency correct (§ 3.11) and re-extract (§ 3.12) the flats.
8. The pipeline now repeats the above step for the twilight flats. **These are the two files (one for both wavelength dithers) you will need to copy over to your galaxy directory to make the response functions for flat fielding.** They will be have the prefix "eqprg", but you may need to look in your log to find the correct file name base (for example, S2014....fits).
9. Next the response function is create for later flat fielding our standard star data. This process is detailed in § 3.13.
10. Now the pipeline quantum efficiency corrects (§ 3.14 and flat fields (§ 3.15) the arcs and leads you through remaking the wavelength solution (§ 3.16) and transformation (§ 3.17).
11. We now move on to the standard star science data. The cosmic ray rejection is first (detailed in § 3.18) and then quantum efficiency correction occurs (§ 3.19).
12. Then the pipeline flat fields and extracts the science spectra using the response function from step

9. This is detailed more in § 3.20.

13. Next the science frames are wavelength transformed (§ 3.21) using the output of step 10, and then sky subtraction occurs (§ 3.22).
14. The pipeline now sums the light in all the fibers to create a 1D spectrum for the standard star. Then, it creates the sensitivity curve files using the task `gsstandard` using the ID spectrum just made as the input file. The task will output two different files: "sfile_steqxprgS..." which is a text file that contains the output fluxes from the star and "sfunction_steqxprgS....fits" which is a fits file that contains the sensitivity function. The routine compares the known fluxes of the standard star you observed to the observed values to create the sensitivity function. **The two "sfunction_....fits" files will need to be copied over to your galaxy directory for the flux calibration step.**
15. We can check our work by flux calibrating the standard star itself. This process is explained in § 3.23.
16. Lastly, data cubes are made for the standard star, and the reduction is complete! Be sure to check the data cubes to make sure everything looks normal. You can use the same twilight files and sensitivity functions for each galaxy observed the same semester as the standard.

4.3 When Things Go Wrong...

4.3.1 A Bad Fiber Identification

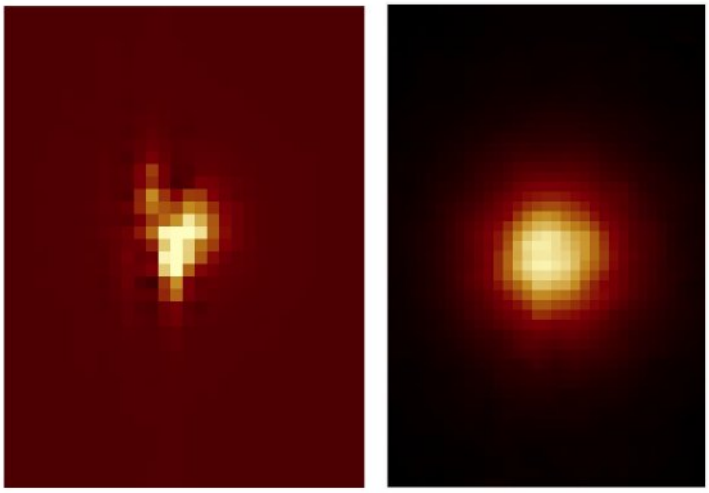

Figure 4.2: On the left is an image of a standard star that has been incorrectly reconstructed in DS9 due to the fibers being improperly identified. On the right is an image of how the star will look with the correct IDs applied. Image is Figure A.2 from http://arxiv.org/abs/1409.8264.

There are two ways (that the author is aware of) to discover that the fiber ID step has gone wrong. First, the shape of your object will look just plain wrong! After creating data cubes of your science data, you can open them in DS9 and check to make sure the object looks like it should. Figure 4.2 shows this for standard star data. Second, you may not be able to make a data cube at all. If you have ID'd more fibers than are turned on in the MDF, you will get an error from the task `gfcube` saying "number of apertures identified does not equal the number of apertures defined in the MDF". Sadly, you won't be able to test your IDs until making the data cubes, and so you may need to start from scratch.

To correct the first issue, start by removing the IDs that are located in your database directory. The author will usually make a subdirectory in the galaxy's folder called "old" or "try1" and move all the files there for safe keeping, expect for the raw data. Then restart the pipeline as normal. It will lead you to the fiber ID step again. This time, enter `no` when it asks to find apertures automatically. Now, you will be able to manually mark the apertures, using `m` to mark and `d` to delete if necessary. It's important to make sure no apertures are marked twice, and that they go in order from 1 to 750 or 1500 from left to right.

For the second issue, if you know which fiber you need to turn off in the MDF, you can edit it manually and then restart the pipeline as detailed in the previous paragraph. 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 edited MDF file to the pipeline code as one of your first steps of the reduction process.

4.3.2 A Bad Wavelength Solution

If something looks wrong, an easy way to fix this is to delete the ID files outside of the pipeline 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.

4.3.3 Correcting the 2014B Data with Bad Amplifier Effects

The blue setup RESOLVE data from the 2014B semester were heavily impacted by a bad column on the new CCDs installed on the telescope that summer. As detailed at www.gemini.edu/node/10626

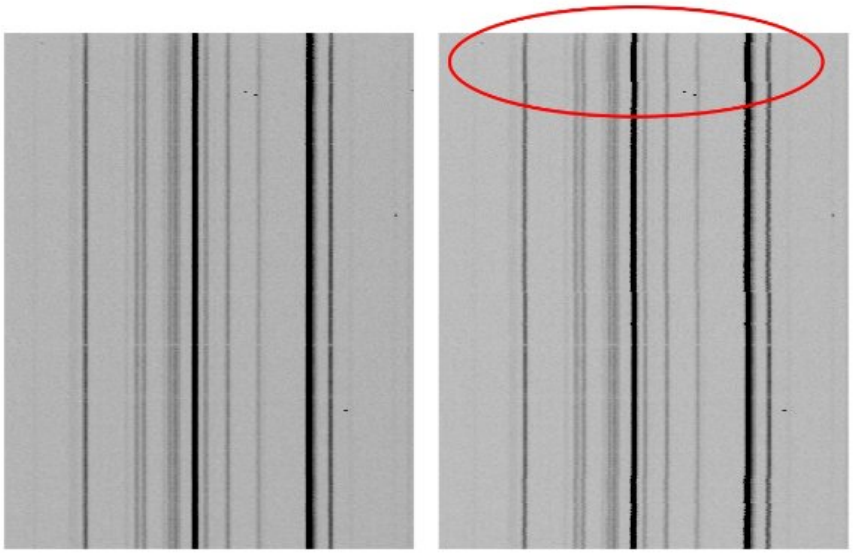

Figure 4.3: On the left is an image of a correctly transformed arc. On the right is an arc with a bad wavelength transformation. In particular, the circled red region shows how the emission lines are jagged in places. Image is Figure A.7 from http://arxiv.org/abs/1409.8264.

on February 26, 2015, the problem started as a bad pixel and has gradually gotten over the course of the semester. In Figure 4.4, we show data taken taken early in the semester, when the hot pixel had spread only to a small part of the CCD read out through amplifier 5. In contrast, we show on the left side of Figure 4.5 data taken late in the semester, where the issue is affecting the entire amplifier section. This only affected the blue setup seriously, since the exposure times in the blue setup are twice as long as those for the red setup (1200s vs. 600s each). This also means that the arcs and flats were not affected either.

Originally, the data was thought to be lost, until German Gimeno at the Gemini Observatory created a flat for us from data taken in 2013B. This flat is shown in Figure 4.6. It is the same size as the effected amplifier region in Figure 4.5, and when divided into the bad region, the saturation effects are removed.

The script to perform the correction and the flat file are both on cielo in the folder /srv/two/emsnyder/gemini/data/2014B/scriptsfromgerman/. The script is named amp5improv_20160107.cl and the flat is called gout_5B.fits. For data like in Figure 4.5, where the bad regions span the full y direction, you can use this flat in full. For smaller regions of bad data, as in Figure 4.4, you will need to copy the flat to a new file name and change the value of the regions not affected to one, so that dividing by the flat will leave those regions unchanged.

This cl script will not run properly in PyRAF, but works fine inside IRAF only. So, start IRAF, and move to the folder where your scripts are located. Then, copy over the science files you want to correct to this folder. I like working in a different folder from the rest of the data so that there's a copy of the original science files.

To perform correctly, the files need to have the MDF attached to them, i.e., have the prefix "g" before the file name. If you already have these created, copy them to this folder instead. If not, you can make them now. Inside IRAF, load the gemini and gmos packages by typing `gemini` and then `gmos`. Then use the task `gprepare` to add the MDF by typing: `gprepare filename fl_addmdf=yes`. If you have a special MDF, you can add `mdffile=filename` to the previous command. You should now have files with the prefix "g" to correct.

Next, make a txt file that just has the names of the files you want to correct listed, as so:

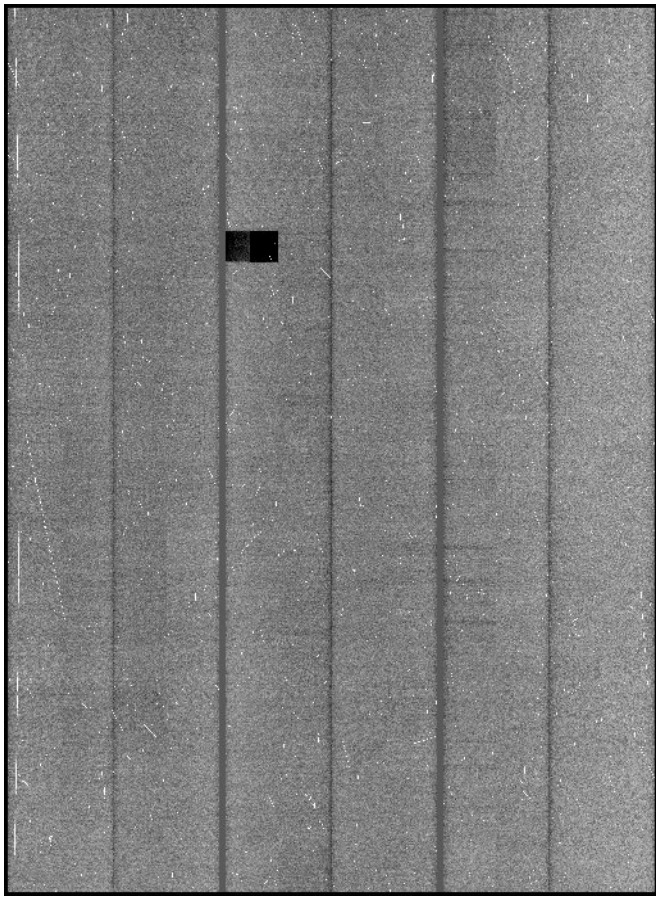

Figure 4.4

```
gS20141228S0030.fits
gS20141228S0035.fits
```

The script looks in this text file to know which files to correct. Inside the script, you can also change the name of the flat file, if you created a new flat to span a smaller region. To run the script, inside IRAF type `cl < amp5improv_20160107.cl`. The output files have the prefix "cg".

Now to get back into the pipeline to complete the reduction, copy the "cg" files back to your galaxy folder. The pipeline will crash if there's a file with the prefix "g" already, so you'll need to complete the next step of the reduction before restarting the pipeline. This is the bias/overscan subtraction and image trimming step. Use the following command in IRAF to do this:

```
gfreduce filename outimag='rgS....' slits='red' or 'blue' fl_inter- fl_over+ fl_trim+ \
fl_bias+ fl_flux- fl_gscrrej- fl_extrac- fl_gsappwave- fl_wavtran- fl_skysub- \
weights- bias='biasfilename' mdffile='yourmdffile'
```

where filename will be your "cg" file (without .fits at the end!), outimage should be just "rg" so that the pipeline will know to use it, slits should be either red or blue depending on your setup, bias is the name of your bias frame, and mdffile is the MDF file (put default if you don't have a special one made). This will make the "rg" file that will go on to be BPM corrected when you restart the pipeline.

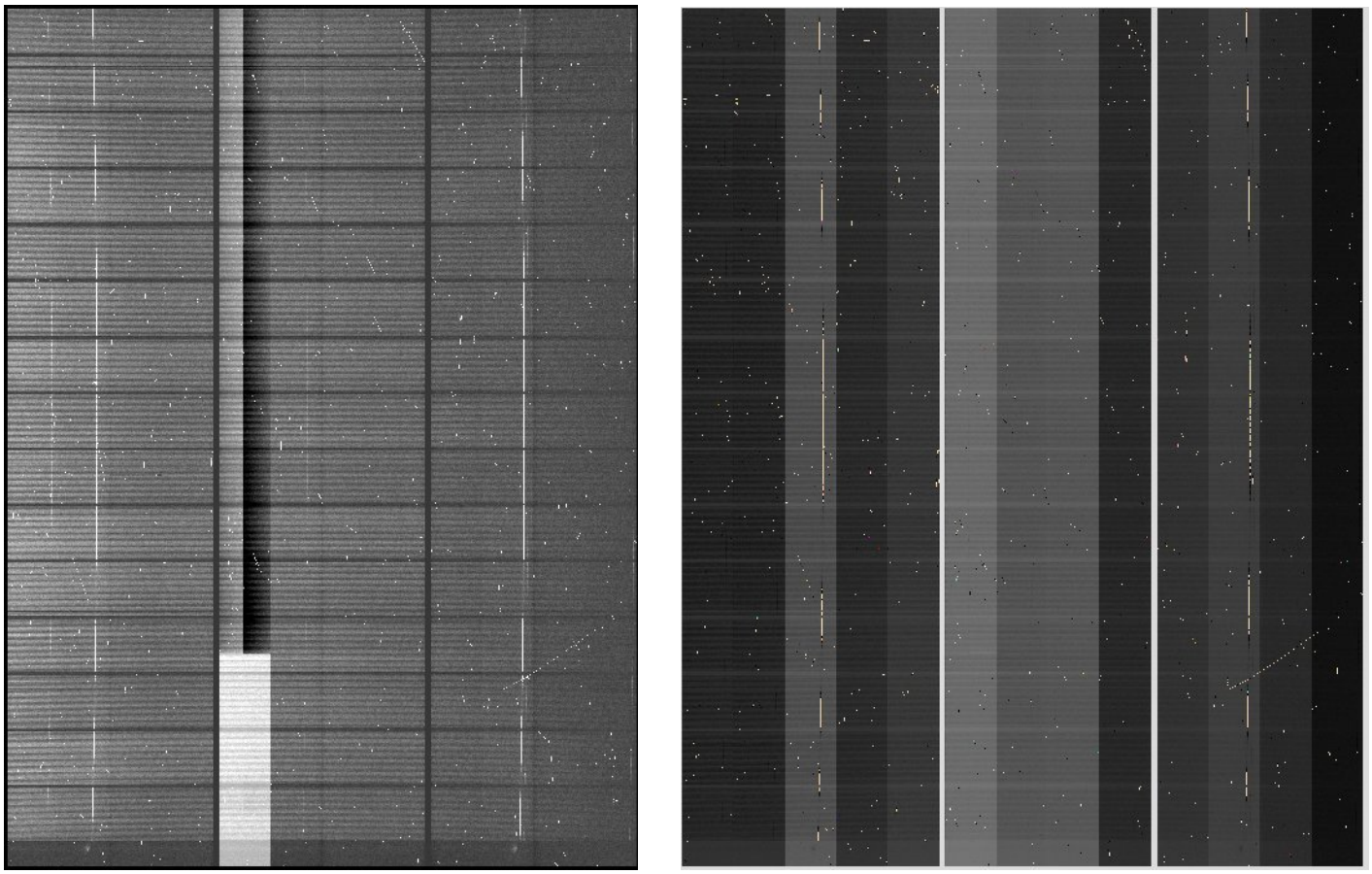

Figure 4.5: **(a).** Before and **(b).** after the amplifier 5 region in the data being corrected using the flat made by German Gimeno.

Figure 4.6: The is the flat created and given to us from German Gimeno. It is the width of one amplifier region, and spans the full y direction of the CCD.