MUSE Data Reduction Workshop- Exercises

Evelyn Johnston

`ejohnston@astro.puc.cl`

August 6, 2019

Introduction

These exercises will guide you through an example reduction of a MUSE datacube with the ESOrex package. Throughout this workshop, you will learn how to create the set-up files and call esorex to run each recipe in the data reduction. Along the way, you will encounter some common issues that you will have to fix, which will give you a deeper understanding of how the pipeline works and how you can modify each step to improve the quality of the final product.

We will carry out the exercises together, but due to computing limitations you will be working in groups of 2-3 for each datacube. In this way everyone should get a chance to run several steps of the data reduction, and we can discuss as a group what each step does to make sure everyone understands.

To get started, please log into Horus with the log in details below. The setup information for X2go is given in Figure 1.

```
open X2go
connect to PUC network via vpn
login 1:  voxel
password 1:  XXXX
login 2:  spaxel
password 2:  XXXX
login 3:  datacube
password 3:  XXXX


data directories:
Jupiter:  /data2/MUSE_DR_Workshop1/
SNe remnant:  /data2/MUSE_DR_Workshop2/
Spiral galaxy:  /data2/MUSE_DR_Workshop3/
```

1 Let's look at the data

Directories:

- **raw**: Contains the raw data downloaded from the ESO Archive

- **sof**: Contains the setup files for each recipe

- **log**: Contains the log files from each recipe

- **products**: Contains the output files from each recipe

Raw data:

- **MUSE*.fits.fz**: Raw data, including science data and calibration files

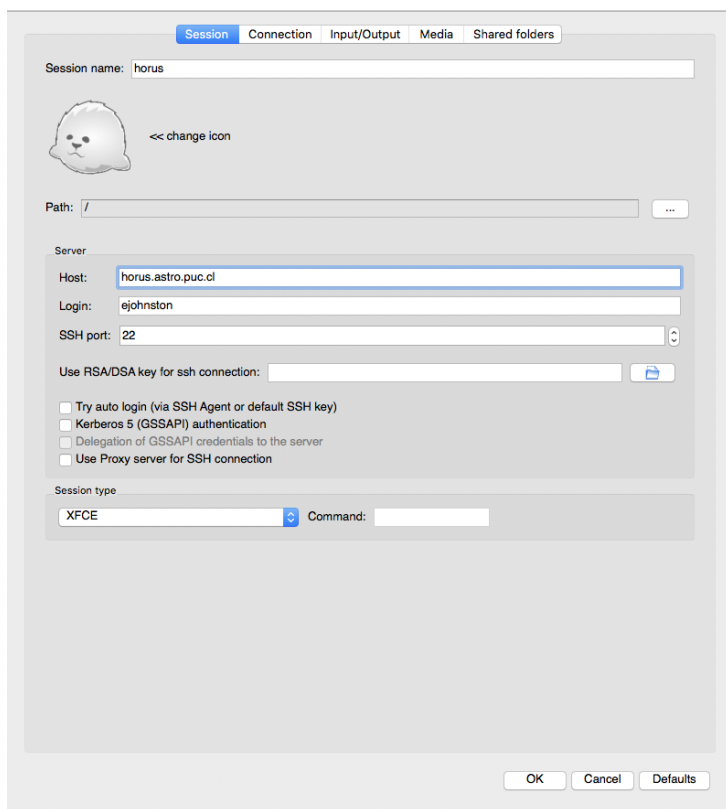- **M.MUSE*.fits**: Static calibrations from ESO

Figure 1: Set up in X2Go to connect to Horus

- **MUSE.*.NL.txt**: Night Log for each science exposure

- **MUSE.*.xml**: ESO Reflex script giving details of associated calibrations for that science exposure

Example of how to list the raw data files and to search for information in the header:
```
[ejohnston@horus MUSE_DR_workshop3] dfits raw/MUSE*.fz | fitsort DPR.TYPE INS.MODE INS.TEMP4.VAL
```

```
FILE                                     DPR.TYPE          INS.MODE      INS.TEMP4.VAL
raw/MUSE.2016-03-10T05:44:52.073.fits.fz FLAT,LAMP,ILLUM   WFM-NOAO-N    12.82
raw/MUSE.2016-03-10T05:49:26.495.fits.fz OBJECT            WFM-NOAO-N    12.91
raw/MUSE.2016-03-10T06:06:11.445.fits.fz SKY               WFM-NOAO-N    12.91
raw/MUSE.2016-03-10T06:09:56.425.fits.fz OBJECT            WFM-NOAO-N    12.89
raw/MUSE.2016-03-10T06:26:41.138.fits.fz SKY               WFM-NOAO-N    12.88
raw/MUSE.2016-03-10T06:31:46.087.fits.fz FLAT,LAMP,ILLUM   WFM-NOAO-N    12.98

raw/MUSE.2016-03-10T09:25:04.745.fits.fz STD               WFM-NOAO-N    13.13
raw/MUSE.2016-03-10T09:39:39.635.fits.fz FLAT,LAMP,ILLUM   WFM-NOAO-N    13.45

raw/MUSE.2016-03-10T10:00:35.995.fits.fz BIAS              WFM-NOAO-N    13.23
raw/MUSE.2016-03-10T10:01:45.130.fits.fz BIAS              WFM-NOAO-N    13.24
raw/MUSE.2016-03-10T10:02:53.933.fits.fz BIAS              WFM-NOAO-N    13.23
raw/MUSE.2016-03-10T10:17:29.018.fits.fz FLAT,LAMP         WFM-NOAO-N    13.34
raw/MUSE.2016-03-10T10:18:48.993.fits.fz FLAT,LAMP         WFM-NOAO-N    13.34
raw/MUSE.2016-03-10T10:20:08.995.fits.fz FLAT,LAMP         WFM-NOAO-N    13.34
raw/MUSE.2016-03-10T10:35:18.981.fits.fz WAVE              WFM-NOAO-N    13.44
raw/MUSE.2016-03-10T10:36:38.795.fits.fz WAVE              WFM-NOAO-N    13.45
raw/MUSE.2016-03-10T10:37:58.770.fits.fz WAVE              WFM-NOAO-N    13.46

raw/MUSE.2016-03-11T22:33:00.026.fits.fz FLAT,LAMP,ILLUM   WFM-NOAO-N    14.20
```

```
raw/MUSE.2016-03-11T22:36:07.264.fits.fz    FLAT,SKY        WFM-NOAO-N   14.19
raw/MUSE.2016-03-11T22:37:27.536.fits.fz    FLAT,SKY        WFM-NOAO-N   14.19
raw/MUSE.2016-03-11T22:38:47.081.fits.fz    FLAT,SKY        WFM-NOAO-N   14.20
raw/MUSE.2016-03-11T23:30:52.571.fits.fz    WAVE            WFM-NOAO-N   13.22
raw/MUSE.2016-03-11T23:32:12.721.fits.fz    WAVE            WFM-NOAO-N   13.21
raw/MUSE.2016-03-11T23:33:32.693.fits.fz    WAVE            WFM-NOAO-N   13.14
raw/MUSE.2016-03-12T09:46:14.772.fits.fz    BIAS            WFM-NOAO-N   10.98
raw/MUSE.2016-03-12T09:47:23.745.fits.fz    BIAS            WFM-NOAO-N   10.98
raw/MUSE.2016-03-12T09:48:32.585.fits.fz    BIAS            WFM-NOAO-N   11.00
raw/MUSE.2016-03-12T10:03:06.308.fits.fz    FLAT,LAMP       WFM-NOAO-N   11.29
raw/MUSE.2016-03-12T10:04:25.974.fits.fz    FLAT,LAMP       WFM-NOAO-N   11.30
raw/MUSE.2016-03-12T10:05:46.034.fits.fz    FLAT,LAMP       WFM-NOAO-N   11.31
```

You can also use `| grep` to search for specific files

Exercise 1

Use the command above to list the files in your raw directory, and then to list only the bias frames

---

2 Pre-processing

In this section, we will create the master calibration frames for each CCD, and apply the basic reduction to the science data from the 24 IFUs.

2.1 Create Master Bias Frame

Exercise 2

Run the command to create a master bias frame (should be one continuous line):

```
OMP_NUM_THREADS=24 nohup taskset -c 0-23 /raid/ejohnston/Esorex/bin/esorex
--log-file=log/bias.log --output-dir=products/ muse_bias --merge --nifu=-1
sof/bias.sof
```

The command can be found in the commands.txt file

While this command is running, I will explain what each part means:

- **OMP_NUM_THREADS=24**: Number of cores to use

- **nohup**: redirect the output to a file called nohup

- **taskset -c 0-23**: use specific cores, in this case cores 0-23

- **/raid/ejohnston/Esorex/bin/esorex**: call esorex. This path leads to the version I have installed for my own use

- **--log-file=log/bias.log**: define the location and name of the log file

- **--output-dir=products/**: define the location for the output files

- **muse_bias**: call a specific recipe. For -- options, see the pipeline manual

- **sof/bias.sof**: define the location and name of the sof file (set of files)

2.2   Create the Master Flat Field

> **Exercise 3**
>
> Run the command for the muse_flat recipe in commands.txt. It should crash.
>
> Have a look in the log file (/log/flat.log) to see if you can identify what went wrong and how to fix it.

Hint: a calibration file is missing from the sof file.

2.3   Create the Wavelength Calibration

> **Exercise 4**
>
> Run the command for the muse_wavecal recipe in commands.txt. It should crash.
>
> Have a look in the log file to see if you can identify what went wrong and how to fix it.

Hint: You might need to use the xml file.

2.4   Create the Master Sky Flat

While the bias, flat field and arc calibrations are taken in the morning following the observations, the sky flats are generally taken within a week of the observations as they require clear conditions in the evening twilight. Therefore, for best results, it's best to create the master bias, flat and wavelength calibrations for the skyflat exposures using the calibrations taken the following morning. However, to save time in this workshop, we will simply use the calibrations for the science data to reduce the skyflats.

> **Exercise 5**
>
> There are some important calibrations missing in the skyflat.sof file. Try to use the dfits command to find the correct calibrations, enter them into the sof file and run the muse_wavecal command from commands.txt.
>
> Remember to check the category names in the pipeline manual (see page 82?)

2.5   Run the muse_scibasic recipe

This step applies the basic data reduction steps to each CCD (bisa subtraction, flat field correction, wavelength calibration etc). Since the recipe can only handle 1 illumination file at a time, you may need to run this recipe multiple times for different data sets with significant time gaps and temperature variations. This step is used to reduce the science, sky and standard star exposures, though in the exercise below you will only run it for one science exposure to save time.

> **Exercise 6**
>
> The illumination correction is missing from the scibasic.sof file. Use the dfits command to identify the best illumination file for the exposure in the sof file and run the command.
>
> Remember to check the category names in the pipeline manual
>
> Only run sci_basic on the science frame, don't worry about the standard star frame. This step can take a long time, so for the purpose of this workshop I have already run sci_basic on the remaining science exposures and the standard star frames.

3 Post-processing

In this section, we will look at the post-processing steps, which convert the 24 reduced CCDs into an IFU datacube

3.1 Calculate the flux calibration

In this step, the pipeline will extract the spectrum of the standard star, and compare it to a known spectrum to calculate the flux conversion.

> **Exercise 7**
>
> Like before, the illumination calibration is missing. Find it and update the sof file to run the recipe.
>
> The command still returns an error message... See if you can use the trouble shooting skills in the previous steps to identify what it is.

3.2 Measure the sky background from a separate image

The best sky subtraction is carried out in the next step using the sky regions on the science datacube. However, in the case of a very extended object, there may be insufficient sky spaxels for a proper sky subtraction. If that's the case, a separate sky exposure must be taken, and the sky background is measured from that exposure with this recipe.

> **Exercise 8**
>
> You will find that the sof file for this recipe is empty. Use the manual to identify the required files and fill in the sof file. In this case, ignore the optional files.

3.3 Run the post-processing on the first exposure

In this step, you will finally convert the pixel tables into a datacube for each exposure, giving you a first look at the data.

> **Exercise 9**
>
> You will find that the sof file for this recipe is empty. Use the manual to identify the required files and fill in the sof file. In this case, include the LSF profile, the astrometry calibrations, and the sky lines file from your own sky background calibration.
>
> Remember to run the commands to rename the output files when this recipe has completed

At this point you will have one datacube per object. While you can combine the datacubes in this step, it's not advisable. This recipe assumes that all exposures were taken with the same guide star and at the same position angle. While some PIs specify their guide star, satisfying the first of these conditions, all exposures show a 90° rotation, thus failing the second requirement. For this reason, it's advisable to run the recipe individually for each exposure, and to rename the output files before they are overwritten.

3.4 Measure the offsets between each exposure

> **Exercise 10**
>
> You should now have a series of images called IMG_FOV_0001_expX.fits, where X is the number of the exposure in the sequence. We must now measure the offsets between each exposure to refine their relative positions when we combine them.
>
> Fill in exp_align.sof with the names of the white light images of the datacubes you wish to combine.
>
> Those reducing the datacube of Jupiter should skip this step

3.5 Combine the pixeltables to create the final datacube

> **Exercise 11**
>
> We now have pixel tables for each exposure along with the necessary offsets from the WCS coordinates. Fill in the exp_combine.sof file, and remember to include the FILTER_LIST file. Now run the command for this step, which creates a datacube with a narrow wavelength range defined by the --lambdamin and --lambdamax keywords, which is a useful trick to quickly check the alignment of each exposure in the final datacube.
>
> If you're reducing the Jupiter data, you will notice that the final datacube is blurry compared to the individual exposures. Why is that?
>
> Once you are happy with the alignment, modify the final command to create a datacube for the full wavelength range and to create g, r, i and z-band images from the datacube.

4 Visualizing the Datacube

If we have time after reducing a datacube, you can take the opportunity to work with it. Suggestions include opening the datacube in ds9 or QFitsView, or using the g, r, i, z-band images to create an RGB image of the target.