Contents

Software Requirements and Installation

Software	**Version**	**Used for**	**Reference**
Bash[1]	4.4.19(1)	Command prompt and pipeline scripts.	
ESO Reflex[2] *FORS Pipeline*	2.9.1 5.3.32 & 5.4.3	Initial reduction of FORS2 images.	(1)
ESOrex[3]	3.13.1	Standard-star image reduction.	
HOTPANTS[4]	5.1.11	Image subtraction.	(2)
Montage[5]	6.0	Coaddition of dithered FORS2 images.	(3)
PSFEx[6]	3.17.1	Point spread function extraction for PSF-fitting with SExtractor.	(4)
Python	3.6.7 2.7.15	Analysis and pipeline code. Running Astrometry.net client code.	
PyCharm	PC-191.7141.48	Writing bash and Python code. Non-essential, but an excellent, fully-featured IDE.	
SExtractor[7]	2.19.5	Extraction of source magnitudes, both for calibration and science.	(5)

[1]https://www.gnu.org/software/bash/
[2]https://www.eso.org/sci/software/esoreflex/
[3]https://www.eso.org/sci/software/cpl/esorex.html
[4]https://github.com/acbecker/hotpants
[5]http://montage.ipac.caltech.edu/
[6]https://www.astromatic.net/software/psfex
[7]https://www.astromatic.net/software/sextractor

0.1 Bash

0.1.1 Required Packages

- `jq` (for reading .json files)
- `gcc` (for installations)

0.2 Python

The companion package `craftutils (https://github.com/Lachimax/craftutils)` will need to be installed and accessible by your Python environment.

Other required packages:

- `astropy`
- `matplotlib`
- `numpy`
- `pandas`
- `photutils` (Install using instructions at https://photutils.readthedocs.io/en/stable/install.html)
- `reproject` (Install using instructions at https://reproject.readthedocs.io/en/stable/#introduction)
- `ruamel.yaml` (or equivalent YAML module)

For reasons unknown, the Astrometry.net client only works well with Python 2, submitting blank fits files to the service if used with Python 3. This necessitates both versions being installed. As Python 2 comes packages with Ubuntu by default, this shouldn't present a major obstacle.

0.3 ESO Reflex

ESO Reflex is used for initial image reduction. The steps I took in order to successfully install it are provided here, as it can be tricky.

1. Install prerequisites[8]:

   ```
   $ sudo apt-get install g++ zlib1g-dev make gzip tar perl gawk sed
   grep coreutils pkg-config
   ```

2. Install Python prerequisites.

   ```
   $ sudo apt-get install python-matplotlib python-wxgtk3.0 python-astropy
   python-numpy libffi-dev
   ```

   NOTE: This does not appear to work with Python 3.8, which comes packaged with Ubuntu 20.04. In order to properly install ESOReflex on Ubuntu 20.04, I created a conda virtual environment with Python version 3.5, and manually installed `matplotlib`, `astropy`, `numpy` and `wxPython`. In this case, the installation script and ESOReflex will both need to be run from within this environment.

   (a) `$ conda create --name py35 python=3.5`

   (b) `$ conda install -n py35 wxPython`

   (c) `$ conda install -n py35 matplotlib`

   (d) `$ conda install -n py35 numpy`

   (e) `$ conda install -n py35 astropy`

   (f) `$ conda activate py35`

3. Install correct version of Java. I have experienced particular difficulties with this step, so it is described in detail, with special thanks to John Pritchard of ESO User Support with his help in overcoming them:

   (a) Install Java 1.8:

   ```
   $ sudo apt-get install openjdk-8-jdk
   ```

   (b) You must also be sure that esoreflex will not use a more recent version of Java, if one is installed:

   ```
   $ apt list --installed | grep openjdk
   ```

   If there are only `openjdk-8` packages installed, then no problem. Otherwise, eg if openjdk-11 packages are present, either:

[8] http://www.eso.org/sci/software/pipelines/installation/software_prerequisites.html

i. Remove the openjdk-11 packages:

```
$ sudo apt-get remove openjdk-11
```

ii. OR: Set the default version of Java to java-8:

```
$ sudo update-alternatives --config java
```

4. Download the `install_esoreflex` script from ftp://ftp.eso.org/pub/dfs/reflex/install_esoreflex to the preferred installation location (it does not create its own subfolder, so it's probably a good idea to put it in an 'ESOReflex' subdirectory).

5. Make executable and execute:

```
$ chmod u+x install_esoreflex
```

```
$ ./install_esoreflex
```

6. Follow the instructions given by the script. Install at least the FORS components.

7. Add the following alias to your .bashrc file, replacing <install_directory> with the location of your installation:

```
alias esoreflex=<install_directory>/install/bin/esoreflex
```

8. It is advisable to test the pipeline using the Demo Data. Instructions for doing so are here found under the individual instrument categories in http://www.eso.org/sci/software/pipelines/

0.4 Montage

1. Montage can be installed using the instructions here:

   http://montage.ipac.caltech.edu/docs/build.html

2. It can also be installed by cloning the repository at:

   https://github.com/Caltech-IPAC/Montage

   (a) You should then follow steps 4. and 5. of the instructions; for this pipeline to work, you MUST add

   ```
   export PATH=$PATH:<Install directory>/Montage/bin/
   ```

   to your `.bashrc` or `.bash-profile`.

0.5 SExtractor

SExtractor can (usually) simply be installed with:

```
$ sudo apt install sextractor
```

However, I have found this to fail with the latest version of SExtractor (2.25; or, possibly, with Ubuntu 20.04), and have had to install manually using the instructions here:
https://sextractor.readthedocs.io/en/latest/Installing.html
This also required installing `autoconf`:

```
$ sudo apt install autoconf
```

As well as `FFTW` and `ATLAS`, which are a bit more involved:

0.5.1 FFTW

1. Download the latest stable release here:
   http://www.fftw.org/download.html

2. Move the zipped file to your desired installation location, and open a terminal there.

3. Unzip:

   ```
   $ tar -xvf fftw-X.X.X.tar.gz
   ```

4. ```
   $ cd fftw-X.X.X
   ```

5. Make sure to configure with single-precision and threading enabled:

   ```
   $ ./configure --enable-single --enable-threads
   $ sudo make
   $ sudo make install
   ```

0.5.2 ATLAS

1. Download the latest stable release:
   https://sourceforge.net/projects/math-atlas/files/Stable/

2. Move the zipped file to your desired installation location, and open a terminal there.
   ```

3. Unzip:

```
$ tar -xvf atlasX.X.X.tar.bz2
```

4. Follow the instructions in the file <install_path>/ATLAS/doc/atlas_install.pdf to install.

5. In the step **'Turn off CPU throttling when installing ATLAS'**, I needed to both edit the grub file as described AND, after restarting, use the /usr/bin/cpufreq-set commands before running configure.

6. It is also possible that you will need to fully install LAPACK, in which case see Section 8 of the instructions mentioned above.

0.5.3 SExtractor

With the above prerequisites installed, you should now be able to install SExtractor following the instructions at https://sextractor.readthedocs.io/en/latest/Installing.html.

1. You may need to specify the locations of the ATLAS lib and include directories when you configure SExtractor:

```
$ ./configure --with-atlas-libdir=<ATLAS-build-directory>/lib \
          --with-atlas-incdir=<ATLAS-source-directory>/include
```

0.6 Hotpants

Hotpants is used for producing the difference images, and is necessary for running the scripts in /scripts/subtraction.

0.6.1 CFITSIO

HOTPANTS requires CFITSIO (https://heasarc.gsfc.nasa.gov/docs/software/fitsio/) to work, which I had some trouble installing - the steps I took to make it work are provided here.

1. Download the CFITSIO tar from http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/c/cfitsio-3.47.tar.gz for the specific version I used, or http://heasarc.gsfc.nasa.gov/FTP/software/fitsio/c/cfitsio_latest.tar.gz for the latest version.

2. Extract the tar file to your desired install directory. You will have to rename the extracted folder to `cfitsio`, instead of `cfitsio-x.xx`; otherwise HOTPANTS will throw an error when you try to `make` it.

3. Enter the extracted directory in the terminal and:

```
$ ./configure
$ make
$ sudo make install
```

0.6.2 Hotpants

4. Clone HOTPANTS from https://github.com/acbecker/hotpants to your desired install directory.

5. In the HOTPANTS Makefile, change the variable `CFITSIOINCDIR` to `<your-cfitiso-directory>/include` and `LIBDIR` to `<your-cfitiso-directory>/lib`, where `<your-cfitiso-directory>` is the directory to which you unpacked the CFITSIO archive.

6. From inside `<your-hotpants-directory>`:

```
$ make
```

7. Add the HOTPANTS directory to the PATH variable, preferably in your `.bashrc` file:

```
PATH=$PATH:<your-hotpants-directory>
```

8. If, upon running `hotpants`, you receive an error like the following:

```
hotpants: error while loading shared libraries: libcfitsio.so.8:
cannot open shared object file: No such file or directory
```

Then try this command:

```
$ export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/<your-cfitsio-path>/lib
```

And maybe add this command to your `.bashrc` for good measure.

Instructions for Use

0.7 General Use

Each of the pipelines must be run from the bash command line. For example, to run the FORS2 pipeline on the second FRB181112 epoch, execute:

```
$ ./scripts/pipeline_fors2/0-pipeline.sh FRB181112_1
```

0.8 Parameter files

0.8.1 Config file

Within the `/param/` directory is `config_template.yaml`. You should make a copy of this, change the title to `config.yaml`, and set it up for your workspace. Use `config_doc` for guidance.

0.8.2 FRB Files

These contain global information about the burst itself; not necessary for processing images, but should be present for host galaxy analysis etc.

0.8.3 Epoch files

Inside the `/param` directory, within subdirectories labelled `epochs_INSTRUMENT`, is a set of parameter files for each captured epoch. These, particularly the directories, must be set for each epoch you wish to process. Use the `epoch_template.yaml` and `epoch_template_doc` files in each directory for guidance.

0.9 FORS2 Image Processing

0.9.1 Downloading data

The raw data must be downloaded from the ESO archive:

1. To (kind of) automate this, go to the ESO Raw Observational Data Portal (https://archive.eso.org/eso/eso_archive_main.html).

2. Find the target by searching on this screen.

3. On the results screen, check the boxes for the desired target and epoch (this pipeline here is designed for use with one target and epoch at a time). Be careful to exclude any images with Category 'ACQUISITION', as this may later cause problems.

4. Click 'Request Marked Datasets'.

5. After logging in, select the 'Instant Download' and 'Selected files + associated raw calibrations' (this is important, as the pipeline requires the calibrations to run properly), then click submit.

6. Wait for all calibration slides to appear in the file list (the page should auto-refresh when they are available), and download the provided script to the directory you wish to use for the data processing. I suggest using the format: `data/FRBXXXXXX/FORS2/MJDXXXXX/`

7. The FORS2 pipeline will use this file to download the data.

0.9.2 Using the pipeline

1. First, ensure that there is a .yaml parameter file in the `/param` directory of the project corresponding to the epoch you are processing, as described in § 0.8. Don't worry about the .json file; the pipeline will generate one automatically. The most important parameters for the initial processing are `object`, `data_dir` & `data_title`; the rest can be ignored for now.

2. You should also ensure that your config param file is set up; `top_data_dir`, `esoreflex_input_dir`, `esoreflex_output_dir`, `proj_dir` and

3. To begin the pipeline, run:

   ```
   $ ./scripts/pipeline_fors2/0-pipeline.sh FRBXXXXXX_X <project>
   ```

   Where `FRBXXXXXX_X` is the title of the epoch parameter file (without the .yaml) and <project> is the title of the project parameter file to use.

4. Follow the instructions given by the pipeline to run the desired scripts. See below for further information on each step.

0.9.3 Initialisation and download

`1-initial` will first download the data, and then organise it within the directory. It will also copy the data to the project's ESO Reflex directory.

1. To skip the download portion of the script, change the `skip_download` argument in the epoch's parameter file to 'true'. This is useful if you already have the data downloaded, eg if you are re-running the script.

2. The ESO download script that you saved in § 0.9.1 will run, and ask you for your password. If you do not wish to type in your ESO password every time you run this script, you can add the following line to your .netrc file (found at `$HOME/.netrc`), and the ESO download script will read it automatically: `machine dataportal.eso.org login _ESOusername_ password _yourpassword_`

3. The requested files will all download automatically.

4. Once the files are downloaded, the pipeline will decompress them and organise them.

5. It will also copy the files to the project's ESO Reflex directory for further processing. The copy step can also be skipped, by setting the parameter file's `skip_copy` argument to true; this is useful if you are running the script for a second time, as the copying can be time-consuming.

6. At this point, you will need to use ESO Reflex to process the images before proceeding (unless you've done this previously). You may leave the script hanging while you do this. See the following section.

0.9.4 ESO Reflex

1. Open ESO Reflex:

   ```
   $ esoreflex
   ```

2. Click `File`, `Open`. Open the folder `fors-<version>` and the file `fors_imaging.xml`. The workflow canvas will appear in a new window.

3. Under `Setup Directories` on the main canvas, check the parameters `ROOT_DATA_DIR` and `RAW_DATA_DIR`. Within your project `.yaml` file inside `PyCRAFT/param`, `esoreflex_input_dir` will need to be somewhere within `<RAW_DATA_DIR>/reflex_input/fors/`, and `esoreflex_output_dir` should be `<ROOT_DATA_DIR>/reflex_end_products/`. If this is not the case, you may need to change them and rerun `1-initial` (with `skip_download` enabled, for speed's sake). I've encountered problems with freezing when changing these parameters within ESO Reflex, which seems to be a tad temperamental about where it looks for data - so it's just easier to change our configuration file to match.

4. A helpful option, for those not content to trust that the pipeline is running without visual feedback (a group that includes myself), is under `Tools`, `Animate at Runtime`. 1000 ms is a good setting for this.

5. Click the green triangle 'Play' button in the toolbar, and the reduction pipeline will begin executing.

6. You will be asked to select the files for reduction. Select all of the files corresponding to the object/epoch you are currently reducing, or, if this is the first time through, those that have not been reduced before will already be selected. Be careful to only reduce the data for one observation at a time; otherwise my pipeline will have trouble. Click `Continue`.

7. ESO Reflex will hopefully do its thing. This may take some time, but it will give you feedback.

8. When the reduction is complete, a Product Explorer window will pop up allowing you to explore the reduction results. You can close this and find the data directly in the directory.

0.9.5 Further processing

You can now run scripts `3`, `4`, `6` and `7`.

0.9.6 Co-addition with Montage

0.9.7 Astrometry

1. In order to perform the astrometry steps, you will need to sign up for an account on Astrometry.net and request an API key, which can be retrieved from http://nova.astrometry.net/api_help.

2. Copy this to the `astrometry` field, in the file `keys.json`, within the path `/<project-path>/para`

3. The scripts should now work.

Bibliography

[1] J. C. Jacob, D. S. Katz, G. B. Berriman, J. Good, A. C. Laity, E. Deelman, C. Kesselman, G. Singh, M.-H. Su, T. A. Prince, and R. Williams. *Montage: An Astronomical Image Mosaicking Toolkit* (2010). URL http://ascl.net/1010.036.

[2] A. Becker and Andrew. *HOTPANTS: High Order Transform of PSF ANd Template Subtraction* (2015). URL http://ascl.net/1504.004.

[3] W. Freudling, M. Romaniello, D. M. Bramich, P. Ballester, V. Forchi, C. E. Garcia-Dablo, S. Moehler, and M. J. Neeser. *Automated data reduction workflows for astronomy*. Astronomy & Astrophysics **559**, 96 (2013). URL https://www.aanda.org/articles/aa/abs/2013/11/aa22494-13/aa22494-13.html.

[4] E. Bertin. *Automated Morphometry with SExtractor and PSFEx*. Astronomical Data Analysis Software and Systems XX **442**, 435 (2011). URL http://aspbooks.org/custom/publications/paper/442-0435.html.

[5] E. Bertin and S. Arnouts. *SExtractor: Software for source extraction*. Astronomy and Astrophysics Supplement Series **117**(2), 393 (1996). URL http://aas.aanda.org/10.1051/aas:1996164.