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)

It is worth mentioning that, since many of the pipeline scripts require an internet connection, that this is practically a necessity for running them properly.

0.1 Bash

0.1.1 Required Packages

- `gcc` (for installations)

0.2 Python

Other required packages:

- `argparse`
- `astropy`

[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

- `matplotlib`
- `numpy`
- `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)
- `PyYAML`

Optional packages:

- `SciScript-Python` (https://github.com/sciserver/SciScript-Python)

  Used for automated retrieval of SDSS data for calibration, but this can be done manually.

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:

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

(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:

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 https://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. Most easily, Montage can be installed simply using `$ sudo apt install sextractor`

2. Alternately, Montage can be installed using the instructions here:

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

3. 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
```

It bears mentioning that the latest version uses `source-extractor` as its bash call when installed via `apt`, and the old calls of `sex` or `sextractor` no longer work; it may thus seem like it has not installed properly when trying to use it the old way. This was, in fact, what led me down the path below.

In case automatic installation fails, it can be installed 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.

0.6.3 GALFIT

https://github.com/vvinuv/pymorph/blob/master/galfit/README.INSTALL

```
$ cp galfit /bin/
$ sudo apt install libncurses5
```

0.6.4 Keys file

Some features of these pipelines require API keys or external accounts; the usernames and passwords of these should be placed in the file `keys.json` in your parameter directory. These are mostly optional, but certain features will not work without them.

Astrometry.net: Used for astrometric calibration of reduced images. An account can be made at http://nova.astrometry.net/

Dark Energy Survey: Data used for zeropoint calibration. An account can be made at https://des.ncsa.illinois.edu/desaccess/

SciServer: Hosts SDSS data, which is used for zeropoint calibration. An account can be created at https://apps.sciserver.org/login-portal/

0.7 FORS2 Image Processing

0.7.1 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_d` 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.

Parameter and Output files

In the following tables, <f> indicates the name of a filter. In the **Written by** column, script titles in brackets are scripts that call the script ahead of the brackets.

0.8 FRB parameter files (`FRBXXXXXX`)

Key	Description	Read by
_fwhm_arcsec	The FWHM of the PSF model produced by PSFEx, in arcseconds.	

0.9 Observation parameter files (`FRBXXXXXX_X`)

Key	Type	Units	Description	Read by
photometry_apertures	List of floats	Pixels	Diameters to use for photometry; passed to SExtractor as `PHOT_APERTURES`.	9-zeropoint. object_prop

0.10 `output_paths.yaml`

All of the following parameters are strings.

Key	Description	Written by	Read by
<f>_cat_path	Path to the final SExtractor catalogue of the processed images.	9-zeropoint.sh	zeropoint.py; object_proper

0.11 PSF <>_output_values.yaml

This file is written by `9-psf.py`, which is called by `4-divide_by_exp_time.sh` and `9-zeropoint.sh`

Key	**Description**	**Read by**
_fwhm_arcsec	The FWHM of the PSF model produced by PSFEx, in arcseconds.	
_fwhm_deg	The FWHM of the PSF model produced by PSFEx, in degrees.	
_fwhm_pix	The FWHM of the PSF model produced by PSFEx, in pixels.	
_pixel_scale	The pixel scale of the image on which PSFEx was run.	

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.