Generating mock catalogs with EGG

1 Installing EGG

1.1 Forewords

EGG is written in C++ and has a few dependencies. I have tried to keep the number of these dependencies as low as possible, and in fact for the moment there are four:

- *phy*$_{++}$, a library for numerical analysis that I have developed during my PhD,
- cfitsio, for handling FITS files,
- WCSlib, for handling sky-to-pixel conversions,
- and CMake, for managing the building process (dependencies, and platform specific stuff).

1.2 Install dependencies

If your operating system comes with a package manager, this should be very easy. Apart from *phy*$_{++}$ that we will address in the next section, these dependencies are standard libraries and tools that should be available in all the package managers.

- Mac users:

  ```
  sudo port install cfitsio wcslib cmake
  ```

  or

  ```
  sudo brew install cfitsio wcslib cmake
  ```

- Linux/Ubuntu users:

  ```
  sudo apt-get install libcfitsio3-dev wcslib-dev cmake
  ```

- Other Linux distributions: You get the point. Use `yum`, `apt`, `pacman`, or whatever package manager is supported by your distribution.
- Windows users: no package manager (see below).

If you don't have a package manager, then you have to compile these tools and libraries yourself... I hope it doesn't come to that, because you may loose a lot of time figuring this out. But in the eventuality, here are the links to places where you can download the source code. Follow the build instructions given on their respective each web page.

- cfitsio: `http://heasarc.gsfc.nasa.gov/fitsio/fitsio.html`
- WCSlib: `http://www.atnf.csiro.au/people/mcalabre/WCS/`
- CMake: `http://www.cmake.org/download/` (they also offer binaries, check this out first)

1.3 Install *phy*$_{++}$ and EGG: the short way

Once the dependencies are properly installed, you can download, build and install the *phy*$_{++}$ library and EGG. Thanks to CMake, the installing process is the same on all computers, and is rather straightforward. The next section describes each steps in detail, in case you are not familiar with CMake, and is the recommended way to go.

However, if you don't have the patience to spend a few minutes on this, or if you are completely lost, you can use the `install.sh` script provided in the `doc/script/` directory. The script does exactly what is written in the next section, and only requires you to specify three parameters ("Configurable options", at the beginning of the script):

- `INSTALL_ROOT_DIR`: This is the directory in which *phy*$_{++}$ and EGG will be installed. If you leave it blank, the script will use the default value adequate for your system (e.g., `/usr/local`). This directory is a "root" directory, in the sense that C++ headers will be installed in `$INSTALL_ROOT_DIR/include`, while executables will be installed in `$INSTALL_ROOT_DIR/bin`, etc.

- `CFITSIO_ROOT_DIR` and `WCSLIB_ROOT_DIR`: These directories tell the script where to find the cfitsio and WCSlib libraries. As above, this is a "root" directory: it must contain the header files in the `include` subdirectory, and the library files in `lib`. Leave it blank if you have installed these libraries through your package manager, or if you installed them manually in the default system folders.

Once you have modified these parameters (if need be), just make sure the script is executable and run it (you can run it from anywhere, the current directory does not matter):

```
chmod +x install.sh
./install.sh
# give your 'sudo' password, if asked
```

In the end, the script will inform you that EGG was successfully installed. If not, please fall back to the manual installation described in the next section.

1.4 Install *phy*$_{++}$ and EGG: the long way

1. Make yourself a temporary directory and open a terminal there.

2. Download the following archives and extract them in this directory:

   - https://github.com/cschreib/phypp/archive/master.tar.gz
   - https://github.com/cschreib/egg/archive/master.tar.gz

   This bash script will do that for you:

   ```
   wget https://github.com/cschreib/phypp/archive/master.tar.gz
   tar -xvzf master.tar.gz && rm master.tar.gz
   wget https://github.com/cschreib/egg/archive/master.tar.gz
   tar -xvzf master.tar.gz && rm master.tar.gz
   ```

   In the end, this should create two directories:

   ```
   egg-master
   phypp-master
   ```

3. Open a terminal and navigate to the `phypp-master` directory. Then, if you are using cfitsio and WCSlib from your package manager, run the following commands:

```
mkdir build && cd build
cmake ../
```

If instead you have installed one of these two libraries by hand, in a non-standard directory, you have to provide this directory to the CMake script. This is done this way:

```
mkdir build && cd build
cmake ../ -DWCSLIB_ROOT_DIR=... -DCFITSIO_ROOT_DIR=...
```

The "..." have to be replaced by the actual directory in which each library was installed. For example, if you have installed cfitsio in the `/opt/local/share/cfitsio` directory, then the "..." after `-DCFITSIO_ROOT_DIR` in the above command has to be replaced by `"/opt/local/share/cfitsio"`.

Another thing to consider is that, by default, CMake will try to install the library inside the system default directories (e.g., `/usr/local/include`). If you have root access to your computer, this is the recommended way to go as it will make things simpler. If you do not have root access, or if for some reason you prefer to install the library somewhere else than the default location, you can specify manually the install directory using `-DCMAKE_INSTALL_PREFIX`:

```
mkdir build && cd build
cmake ../ -DCMAKE_INSTALL_PREFIX=...
```

Here, the "..." have to be replaced by the root directory in which you want to install the library. For example, if you want to install it in `/opt/local/include`, just replace "..." by `"/opt/local"` (yes, omit `include`). This can of course be combined with the manual installation directories for WCSlib and cfitsio.

If all goes well, this will configure the *phy*$_{++}$ library and prepare it for installation. The script will most likely warn you about missing dependencies, but this is OK since none of these are needed for EGG. Just make sure that cfitsio and WCSlib are found correctly, then install the library with the following command:

```
sudo make install
# or just 'make install' if you do not need root access
```

4. Using the terminal, navigate now inside the `egg-master` directory. Similarly, run the following commands:

```
mkdir build && cd build
cmake ../
```

As for *phy*$_{++}$, the default behavior is to install EGG in the system directory (e.g., `/usr/local/bin`). If you want to change this, use instead:

```
mkdir build && cd build
cmake ../ -DCMAKE_INSTALL_PREFIX=...
```

Again, the "..." have to be replaced by the directory in which you want to install the programs. For example, if you want to install it in `/opt/local/bin`, just replace "..." by `"/opt/local"` (yes, omit `bin`).

Note that if you have installed the *phy*$_{++}$ library in a non-standard folder, for example in the `/opt/local` directory (as above), you will have to manually specify this location to CMake and use instead:

```
mkdir build && cd build
cmake ../ -DPHYPP_ROOT_DIR="/opt/local"
```

The same is true for WCSlib and cfitsio: if you had to specify their location manually when installing *phy*$_{++}$, you will have to repeat this here.

CMake will generate an error if, somehow, there was an issue in the installation of the *phy*$_{++}$ library. Else, it will configure EGG and make it ready to be built. Finally, run the last command:

```
sudo make install
# or just 'make install' if you do not need root access
```

At the end of the process, CMake will remind you in which directory the EGG executables are installed, e.g.:

```
-- Installed: /usr/local/bin/egg-gencat
-- Installed: /usr/local/bin/egg-genmap
-- Installed: /usr/local/bin/egg-gennoise
-- Installed: /usr/local/bin/egg-buildmf
-- Installed: /usr/local/bin/egg-2skymaker
```

If you chose a non-standard install directory, make sure this directory is in your PATH, and you are done. See, not that hard!

1.5 Making sure everything works

Navigate into a directory of your choosing and call:

```
egg-gencat verbose maglim=27 selection_band=hst-f160w area=0.08
```

This will take a few seconds to run. In the end, you should get something like:

```
note: initializing filters...
note: initializing SED libraries...
note: initializing redshift bins...
note: min dz: 0.1, max dz: 1.00632
note: 33 redshift slices
note: estimating redshift-dependend mass limit...
note: will generate masses from as low as 5.50406, up to 12
note: reading mass functions...
note: found 19 redshift bins and 181 mass bins
note: generating redshifts...
note: generated 66229 galaxies
note: generating masses...
note: generating sky positions...
[----------------------------------------] 33 100%, 57ms elapsed, 0ns left, 57ms total
note: generating morphology...
note: generating SFR...
note: assigning optical SEDs...
[----------------------------------------] 900 100%, 169ms elapsed, 0ns left, 169ms total
[----------------------------------------] 900 100%, 168ms elapsed, 0ns left, 168ms total
note: generate IR properties...
note: assigning IR SED...
note: computing fluxes ...
[----------------------------------------] 66229 100%, 10s elapsed, 0ns left, 10s total
[----------------------------------------] 66229 100%, 14s elapsed, 0ns left, 14s total
note: saving catalog...
```

Also, a file called egg-2015xxxx.fits (e.g., for me it was egg-20151127.fits) weighting about 27MB will be created in the same directory. This is the output catalog, in FITS format. You can open it in IDL to check its content with the following IDL command:

```
; Load the catalog
cat = mrdfits('egg-2015xxxx.fits', 1)
; Look at its content
help, cat, /str
; Then do some plots
plot, cat.z, cat.m, psym=3, xtit='redshift', ytit='stellar mass'
```

Then it remains to test the program that will translate this catalog into a *SkyMaker*-compatible catalog, one per band. Try:

```
egg-2skymaker egg-2015xxxx.fits verbose band=hst-f160w template=goodss-hst-f160w.conf
```

This will use one of the pre-defined *SkyMaker* template to configure a *Hubble* image at the same quality as in the GOODS–*South* field. This should be very fast and print a few lines in the terminal:

```
note: reading catalog...
note: found 66229 galaxies
note: convert parameters for ingestion by SkyMaker...
note: define image size and pixel coordinates...
note: image dimensions: 17951,17967 (1.2015 GB)
note: write catalog...
note: done.
```

In addition, three files should have been created in the same directory, `egg-2015xxxx-hst-f160w.cat` (the *SkyMaker* input catalog), `egg-2015xxxx-hst-f160w-hdr.txt` (the WCS header to feed to *SkyMaker*), and `egg-2015xxxx-sky.conf` (the *SkyMaker* configuration file).

If you have *SkyMaker* installed on your computer, you can generate the corresponding image by simply running:

```
sky egg-2015xxxx-hst-f160w.cat -c egg-2015xxxx-hst-f160w-sky.conf
```

This should take a couple of minutes (77 seconds on my computer), and finally produce a *Hubble* image in `egg-2015xxxx-hst-f160w-sci.fits` (about 1.3GB in size). You can open it with DS9 and admire the Universe you just created. Congratulations!