Generating mock catalogs with *Ifni*

1 Creating the mock catalog

The main idea behind the generation process of this mock catalog is that everything can be statistically inferred from the redshift, the stellar mass and the "star-forming" flag of each galaxy. The procedure is therefore composed of two main steps: first, generate a realistic distribution of masses at different redshifts both for active and passive galaxies using observed mass-functions; second, estimate all the other physical properties using statistical recipes: morphology, SFR, attenuation, optical colors, and sky-projected position. I now describe each calculation in detail.

1.1 Generating redshifts and masses

The purpose of the mock catalog is the simulate a field similar to the GOODS–South CANDELS field. Therefore, in order to most closely mimic the properties of this field, I have estimated conditional mass functions at different redshifts, as described in my paper (Schreiber et al. 2015). Briefly, the whole GOODS–South catalog is cut at $H < 26$ to ensure high completeness, split in two population of "active" and "passive" galaxies according to the UVJ color-color selection, and further split in multiple redshift bins from $z = 0.3$ to $z = 4.5$. I have used the masses and redshifts computed by Maurilio Pannella using the CANDELS photometry, but I can easily switch to the official CANDELS mass and redshift catalog if needed. Then, I simply computed the mass distribution of each of these sub-samples, performing 1st order completeness corrections, and fit a double Schechter law. Using these fits, I can generate mass functions down to arbitrarily low stellar masses. To reach higher redshifts, I have used the mass functions calculated by Grazian et al. (2015) for $z < 7.5$. The $z = 0$ mass functions is adapted from Baldry et al. (2012) (converted from Chabrier to Salpeter), but it should not matter much since we are aiming for pencil-beam surveys which contain very few local galaxies.

Once this is done, I define a fine grid of redshifts, say from $z = 0.01$ to $z = 6$ with $\Delta z \simeq 0.1 \times (1 + z)$ (imposing a minimum $\Delta z > 0.1$), and I choose the sky area of the mock catalog (here I took an area similar to the catalog you first produced with Stuff, i.e. 17×17 arcmin). In each redshift bin, I interpolate the above mass-functions to a redshift equal to the center of the current bin, multiply it by the volume of Universe probed by the survey in this bin, and generate masses following the distributions of both passive and active galaxies (the two populations are identified by a flag in the catalog), from $M_* = M_{\rm min}$ to $M_* = M_{\rm max}$. I chose $M_{\rm max} = 10^{12}\,{\rm M}_\odot$, and $M_{\rm min}$ can be chosen either to be constant (e.g. $10^7\,{\rm M}_\odot$) or to vary with redshift so as to reach a given magnitude limit in the selection band (so you can chose to generate a catalog which is complete down to $H < 27$, for example). This later step uses the library optical SEDs described below to estimate roughly the mass completeness.

Now, the mock catalog has exactly the same mass and redshift distribution as the CANDELS catalog in GOODS–South. This is nice, but one has to keep in mind that, by construction, this also means that we have imposed the same cosmic variance than in GOODS–South.

1.2 Generating morphology

The Stuff program was not only generating photometry, but also detailed morphology and bulge-to-disk decomposition in each band. In order to be able to plug this new mock catalog in Skymaker directly, I also generate these informations.

The first important quantity is the bulge-to-total ratio B/T, which tells how much of the *mass* of the galaxy goes into the bulge, as opposed to the disk. I generate this quantity using the relations between B/T and M_* published by Lang et al. (2014). These relations are conveniently provided both for active and passive galaxies, at different redshifts. They report no strong redshift evolution between $z = 1$ and

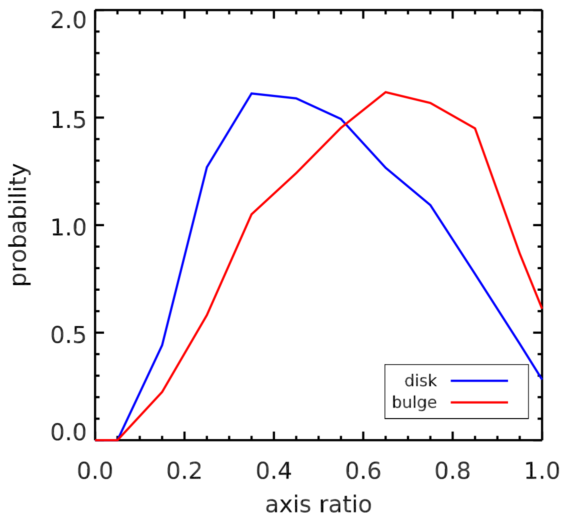

Figure 1: Observed axis ratio distribution of disk-dominated ($n < 1.5$) and bulge-dominated ($n > 2.5$) galaxies. Sérsic fits were taken from the CANDELS wiki, and were produced by Arjen van der Wel. Note that I also added a cut in stellar mass, in order not to be polluted by low mass faint galaxies ($M_* > 10^9\,\mathrm{M}_\odot$ for disks, $M_* > 3\times10^{10}\,\mathrm{M}_\odot$ for bulges).

$z = 2$, so I chose to make the B/T simply depend on mass following

$$(B/T)_{\mathrm{active}} = 0.2 \times \left(\frac{M_*}{10^{10}}\right)^{0.27} \times 10^{G(0.2)} \text{ and} \tag{1}$$

$$(B/T)_{\mathrm{passive}} = 0.5 \times \left(\frac{M_*}{10^{10}}\right)^{0.1} \times 10^{G(0.2)}\,, \tag{2}$$

where $G(\sigma)$ is a zero-mean Gaussian noise of amplitude σ. The B/T is then clamped to $0 \leq B/T \leq 1$. This quantity will be used later to define the colors of the galaxies.

The other set of morphological properties we need to generate are the axis ratio, position angle and size of both the disk and the bulge component of each galaxy. I chose to give the same position angle to both components, which is chosen randomly with uniform probability between -90 deg and $+90$ deg. The axis ratio is generated following the distribution observed in the real catalogs. For the disk (resp. bulge), I took a sample of galaxies with Sérsic index $n < 1.5$ (resp. $n > 2.5$) and computed their axis ratio distribution. The result is shown in Fig. 1. I used these distributions to generate the axis ratios of both disks and bulges in the mock catalog.

To estimate the sizes, I used the same sub-samples and looked at the relation between the observed H-band size, the mass, and the redshift. I ended up with the following relations

$$R_{\mathrm{disk}} = \begin{cases} (1+z)^{-1.25} \times \left(\frac{M_*}{10^{10}}\right)^{0.17} \times 10^{G(0.2)} & \text{for } z < 1.5, \\ 0.4 \times (1+z)^{-0.25} \times \left(\frac{M_*}{10^{10}}\right)^{0.17} \times 10^{G(0.2)} & \text{for } z > 1.5, \text{ and} \end{cases} \tag{3}$$

$$R_{\mathrm{bulge}} = (1+z)^{-2.5} \times \left(\frac{M_*}{10^{10}}\right)^{0.7} \times 10^{G(0.2)}\,, \tag{4}$$

1.3 Generating star formation rate

This step is simple. I used the Main Sequence approach, which attributes a "main sequence" SFR to every galaxy, knowing its redshift and its stellar mass. I used the $\mathrm{SFR}_{\mathrm{MS}}$ published in my paper (Schreiber et al. 2015, Eq. 9). On top of this, a random lognormal scatter of 0.3 dex is added, and a small fraction (3.3%) of the sample is randomly put in the "starburst" mode, following the 2SFM model (Sargent et al. 2012) and using the best-fit parameters obtained in Schreibet et al. (2015). In the end:

$$R_{\mathrm{SB}} = \begin{cases} 10^{G(0.3)} & \text{for Main Sequence galaxies} \\ 5.2 \times 10^{G(0.3)} & \text{for Staburst galaxies} \end{cases} \tag{5}$$

$$\mathrm{SFR} = \mathrm{SFR}_{\mathrm{MS}} \times R_{\mathrm{SB}}\,. \tag{6}$$

This quantity, R_{SB}, the "starburstiness", will be used later to generate the IR photometry.

Then, I split this SFR between obscured and non-obscured components. The obscured component generates the IR fluxes, while the non-obscured component emerges naturally in the UV. To do so, I use the evolution of $\mathrm{IRX} \equiv \mathrm{L_{IR}/L_{UV}}$ I saw when doing stacking for my latest paper, which gives

$$\mathrm{IRX} = \begin{cases} 15.8 \times \left(\frac{M_*}{3\times10^{10}}\right)^{0.45\,z+0.35} & \text{for } z < 3 \\ 15.8 \times \left(\frac{M_*}{3\times10^{10}}\right)^{1.7} & \text{for } z > 3. \end{cases} \tag{7}$$

From there is it then simple to recover L_{IR} and L_{UV}, and therefore the obscured and non-obscured part of the SFR. Note finally that passive galaxies are given zero SFR.

1.4 Generating optical colors

To pick an optical SED for each galaxy, I choose to work with the UVJ color-color diagram. In there, passive galaxies occupy a well defined region (red cloud), while star-forming galaxies form a sort of "sequence", which is actually generated by the attenuation vector (assuming the Calzetti law, see e.g. Williams et al. 2009, Fig. 8). This is useful, because it is known (e.g. Pannella et al. 2014) that the attenuation correlates strongly with the stellar mass. I used this fact to associate colors to active and passive galaxies, again starting only from their redshift and masses. You can also take a look at Fig. 1 in my paper (Schreiber et al. 2015) to see the trends.

Passive galaxies are well condensed in a fixed region, close to $V - J = 1.25$ and $U - V = 1.85$, so they are simple to generate. There is a small trend with stellar mass as well. It is not very important, but I will simulate it for completeness. The principle is to put all passive galaxies at the position given just above, shift them according to the attenuation vector with a stellar mass trend, and add some Gaussian noise to the colors. The final colors are chosen following

$$A = 0.1 \times (\log_{10}(M_*/\mathrm{M_\odot}) - 11) + G(0.1)\,, \tag{8}$$

$$(V - J)_{\mathrm{passive}} = 1.25 + A + G(0.1)\,, \tag{9}$$

$$(U - V)_{\mathrm{passive}} = 1.85 + 0.88 \times A + G(0.1)\,. \tag{10}$$

Note that the "shift" A is clamped to the range $[-0.1, 0.2]$ so that galaxies do not leave the red cloud.

For star-forming galaxies, one needs to be a bit more subtle because their colors vary a lot more. As can be seen in Fig. 1 from Schreiber et al. (2015), star-forming galaxies populate different regions of the UVJ diagram depending on the stellar mass and redshift: massive galaxies are preferentially located on the top-right corner (red $U - V$ and $V - J$ colors), while low-mass galaxies are at the bottom-left (blue in $U - V$ and $V - J$), and they are shifted to bluer colors at higher redshift. This can be due either to a difference of attenuation, or younger ages. In any case, we can parametrize this evolution.

To do so, I took a sample of UVJ star-forming galaxies in GOODS–South, and split it in mass bins. I further decompose each of these bins by slicing in redshift, and compute the median $U - V$ and $V - J$ colors. This gives me a set of tracks in the UVJ diagram, which are reproduced in Fig. 2 (left). It turns out that these tracks fall roughly on a fixed line of slope 0.65. So I computed the projection of the tracks on that line, and parametrized its evolution as

$$A_0 = 0.58 \times \mathrm{erf}(\log_{10}(M_*/\mathrm{M_\odot}) - 10) + 1.39\,, \tag{11}$$

$$A_s = \begin{cases} -0.34 + 0.3 \times \log_{10}\left(\frac{M_*}{2.2\times10^{10}\,\mathrm{M_\odot}}\right) & \text{for } M_* > 2.2 \times 10^{10}\,\mathrm{M_\odot}, \\ -0.34 & \text{for } M_* < 2.2 \times 10^{10}\,\mathrm{M_\odot}, \end{cases} \tag{12}$$

$$A_1 = A_0 + A_s \times z\,, \tag{13}$$

$$A = A_1 + G(0.1)\,, \tag{14}$$

$$(V - J)_{\mathrm{active}} = 0.0 + A \times \cos(\theta) + G(0.12)\,, \tag{15}$$

$$(U - V)_{\mathrm{active}} = 0.45 + A \times \sin(\theta) + G(0.12)\,. \tag{16}$$

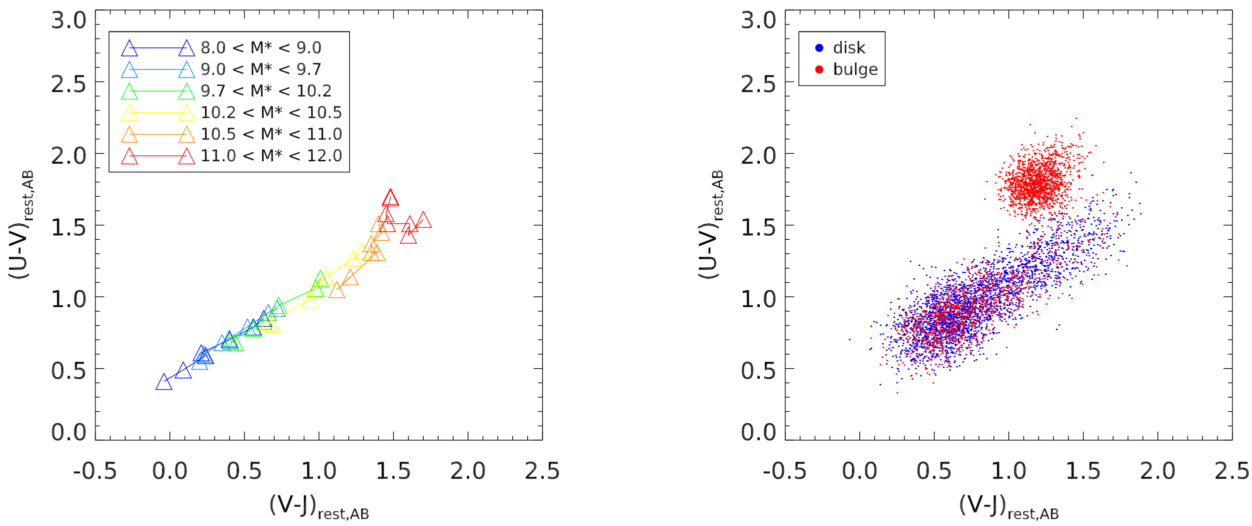

Figure 2: **Left:** Observed median colors of galaxies of different masses, for different redshift (from $z = 0.3$ to $z = 3.0$). The trend is that galaxies move diagonally toward the bottom-left corner when going to higher redshifts. **Right:** Generated UVJ colors of disk (blue) and bulge (red) components of galaxies with $M_* > 10^9$ M$_\odot$ and $0.8 < z < 1.2$.

with A_1 being limited to at most 2, and $\theta = \arctan(0.65)$.

This parametrization will generate a UVJ diagram very similar to the observed one, with the same redshift and mass trends. However, the observed UVJ diagram is made out of the *total* light of the galaxy. Here we need to decompose the galaxy into a bulge and a disk component, and both have usually different colors. The way I chose to handle this is to always use the "active" UVJ colors for disk components, always use the "passive" UVJ colors for bulges of bulge-dominated galaxies ($B/T > 0.6$), and randomly use either the "passive" or the "active" UVJ colors for the bulges of intermediate galaxies ($B/T < 0.6$) with 50% probability each.

The resulting UVJ colors are shown in Fig. 2 (right).

1.5 Choosing an optical SED

To go beyond the colors and associate a full optical SED to the galaxies (and their bulge/disk components), the idea is to consider that there is an average SED for each position on the UVJ diagram, and that one can attribute this average SED to the galaxies that are at this position.

Therefore I have binned the UVJ plane into small buckets of about 0.1 mag, and computed the observed average rest-frame SED of all the galaxies that fall inside each bucket. I also tried to split the sample in redshift bins, to further refine the SEDs, but this turned out to have negligible impact, so I assume no redshift dependence. These rest-frame SEDs are actually generated by FAST, which is the code that Maurilio Pannella used to estimate stellar masses in the first place, and are created from the Bruzual & Charlot (2003) stellar population models, assuming a delayed exponentially declining star formation history. These SEDs are output by FAST in observed-frame, so I first de-redshift them, normalize them by the total stellar mass of the galaxy, interpolate them on a common rest-frame wavelength grid, and compute the average for each wavelength. The result is a wide library of about 850 reference SEDs, all normalized per unit stellar mass.

Then the procedure is simply to pick one of these SEDs depending on the position of the galaxy in the UVJ diagram. If the average SED is made out of less than 10 observed galaxies, the bin is considered as unreliable and the average SED of the closest reliable bin on the UVJ plane is chosen instead.

I run this procedure for both disk and bulge components, multiply the chosen SEDs by the respective stellar mass of each component, redshift them to the redshift of the galaxy, and finally integrate the resulting SED over the chosen UV-NIR passbands.

1.6 Choosing an IR SED

The generation of the IR fluxes is the same as the one we used before, either for my paper, or to generate the *Herschel* images with the previous Astrodeep mock catalog. Basically, we use the Chary & Elbaz (2001) library of FIR SEDs, normalize them to unit $L_{\rm IR}$, and attribute one of these SEDs to every galaxy, from its redshift and "starburtiness" (see subsection 1.3). At higher redshifts, galaxies have warmer dust temperatures (Magdis et al. 2012), and the dust temperature also correlates with the offset of a galaxy from the Main Sequence (Magnelli et al. 2014). We use here the redshift evolution that was seen when I stacked *Herschel* images for my paper, and the starburstiness trend is put by hand (I'm not sure it has an important effect though).

Then, as for the optical flux computation, the chosen SED is multiplied by the $L_{\rm IR}$ of the galaxy, redshifted, and integrated over the chosen IR passbands to produce the final flux. Note that I chose to attribute all of the FIR flux to the "disk" component. This should not matter, since at these wavelengths we usually do not have the resolution to disentangle between bulge and disk.

1.7 Generating sky positions

The final step is to generate a position on the sky for each galaxy. The procedure I used here is the same as the one I described earlier. To start with, I make very simplistic assumptions. First, I consider that the angular clustering does not evolve with redshift. This is probably wrong, since we know that galaxies are more clustered today than in the past. On the other hand, I work here in *angular* correlation, not proper distance (say, in kpc). I assume the same angular correlation at $z = 1$ and $z = 0.5$ (e.g.), which means that galaxies will be clustered on the same angular scale. This angular scale will correspond to a smaller proper distance at $z = 0.5$ than at $z = 1$, so it will somehow mimic the increase of proper distance clustering with time. I don't know how far that is from the real observed trends though. Second, I consider that there is no sub-population of galaxies that are more clustered than the rest. E.g., massive early-type galaxies are treated the same way as, say, dwarf star-forming galaxies. This is probably wrong. For the FIR images however I did not care much, because both these populations will not contribute at all to the FIR flux, and we are mostly dominated by star-forming galaxies close to $M_* = 3 \times 10^{10} - 10^{11}\,{\rm M}_\odot$ at all redshifts.

With that in mind, I first take the observed GOODS–South catalog from CANDELS, select all galaxies between $z = 0.9$ and $z = 1.1$ ($z_{\rm phot}$) and $M_* > 10^9\,{\rm M}_\odot$. I compute the two-point correlation function of this sample using the Landy-Szalay estimator (thanks to Catherine White!). I know that this correlation function is not "pure", in the sense that 1) the redshift range considered ($0.9 < z < 1.1$) is large (~ 1 Gpc); and 2) even if it weren't, our photometric redshifts are uncertain. There is not much I can do about point 1), since we have limited volume in CANDELS, and I need large statistics to build robust correlation functions. For point 2) however, the redshift uncertainties can be simulated, and I have done so.

My objective is then to simulate a similar correlation function from scratch. To do so, I first take a redshift slice of the mock catalog. I generate clustered positions using the Soneira & Peebles algorithm (again, thanks to Catherine White!), with a fixed set of parameters (power law index equal to 0.4, number of levels $N_{\rm level} = 4$, and I generate twice more sources than needed, then pick at random the ones I need among these). These parameters were fine tuned to reproduce the observed correlation functions (see below). It turns out that, doing so, one gets the right two-point correlation slope, but not the right amplitude: the correlation is too strong at all scale. To fix this, one has to say that there is a fraction (60%) of the sample which is not clustered, and has uniformly random sky positions.

To make sure that these parameters are matching the "real" correlation function, I simulated redshift uncertainties to try to mimic the measurement that I did in GOODS–South. I took the "true" redshifts from the mock catalog, and added a Gaussian error proportional to $1+z$, with an error of 2%. I also added 20% of "outliers" with 7% relative error. These numbers were chosen to reproduce the distribution of redshift differences found when comparing our CANDELS redshifts against the 3DSHT redshifts. Re-slicing the catalog with these "uncertain" redshifts and computing the correlation function, one recovers exactly the observed correlation.

Also, as a double check, I computed the angular correlation function of the whole catalog above $M_* > 10^{10}\,\mathrm{M}_\odot$, mixing all redshifts all together. This way, I get rid of the issue of the redshift uncertainty, and the agreement is also very good. I have also tried to look at different sub-samples, like *Herschel* detections only, massive star-forming galaxies, massive passive galaxies, and it turned out that the correlation function was more or less always the same.

2 Results

I will use two diagnostics to assess the quality of this mock catalog. The first one is the flux distribution of the whole catalog, and the second is the pixel distribution of simulated images (for confused FIR images where blending is important).

In what follows, I use a mock catalog generated with 90% completeness in H-band down to $H = 29$, from $z = 0.01$ to $z = 6$. Over 17×17 arcmin, this represents 104 000 galaxies. The minimum stellar mass goes as low as $5 \times 10^4\,\mathrm{M}_\odot$ at $z = 0.01$, and rises with redshift to reach $7 \times 10^6\,\mathrm{M}_\odot$ at $z = 1$, and $10^8\,\mathrm{M}_\odot$ at $z = 4$.

2.1 Optical magnitudes

Fig. 3 is showing the agreement of the total magnitude distribution, in multiple bands. This agreement is very good in the NIR. Since these wavelengths are most closely correlated to the stellar mass of the galaxies, and since the mock catalog was built to reproduce exactly the stellar mass function in GOODS–South, this should not come as a surprise. Still, this shows that the procedure works well. The situation in the UV-optical (F435W and F606W) is not perfect though, as you can see there is a bending at faint magnitudes close to 25 magnitude. I'm not sure where this comes from though, but it seems to originate from low-redshift low-mass galaxies, which are probably not blue enough in the mock catalog. Maybe that could be fixed by tweaking the redshift splitting of the SED library.

2.2 FIR fluxes

Fig. 4 shows basically the same plots, but with the FIR fluxes. Here the agreement is very good, there is really nothing to say. Same goes for Fig. 5, where I compare the pixel histogram distribution of the simulated maps against the observed maps. This second test is important because of the blending, which sometimes pollutes the measured flux catalogs (two sources are combined into a single one), which tends to produce more bright fluxes than there actually is in the real Universe. By analyzing the map statistics directly, one gets rid of this issue of the counter part identification.

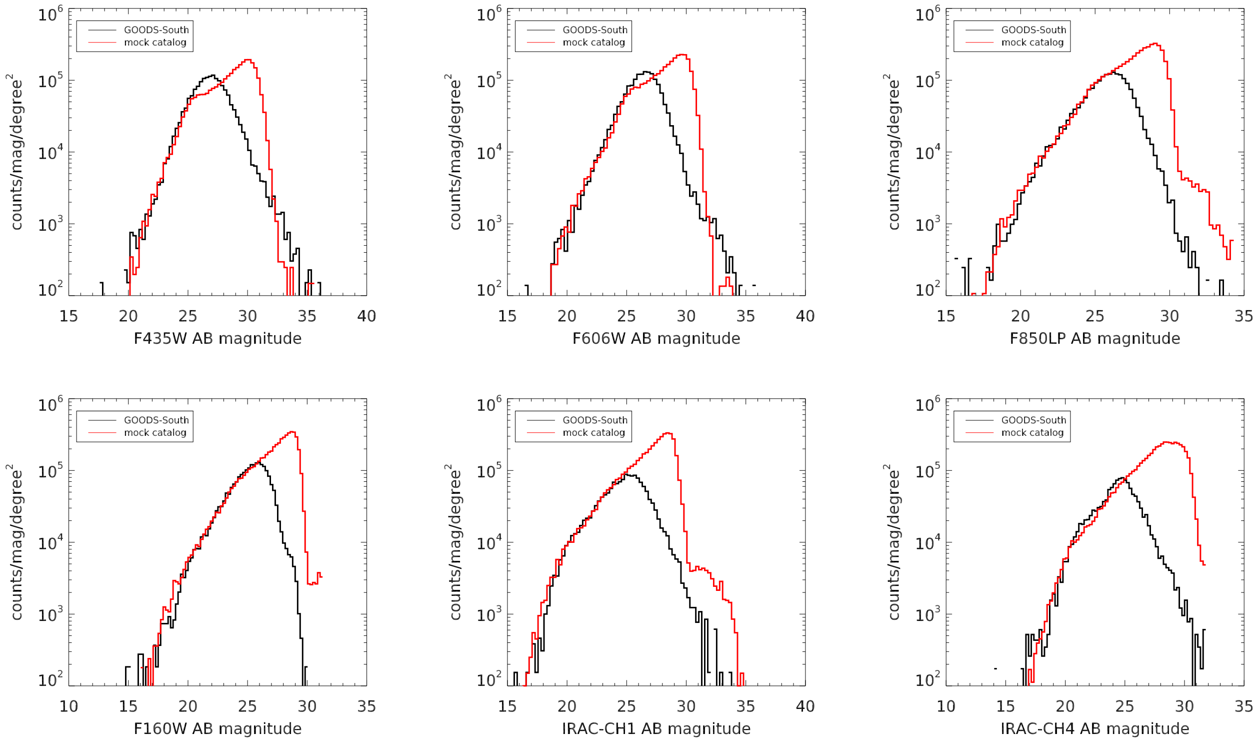

Figure 3: Total magnitude distribution of the real GOODS–South catalog (black) and the mock catalog (red), in different *HST* bands and *Spitzer* IRAC.

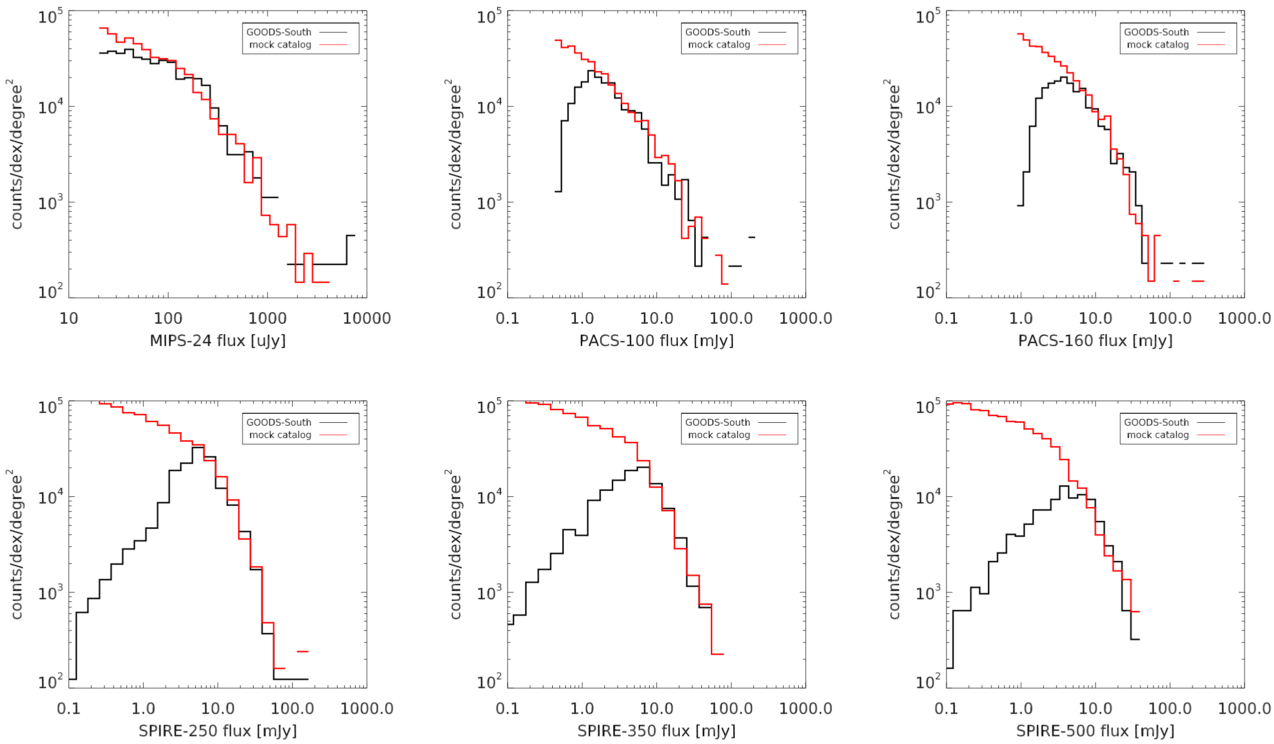

Figure 4: Total flux distribution in the MIR to FIR of the real GOODS–South catalog (black) and the mock catalog (red).

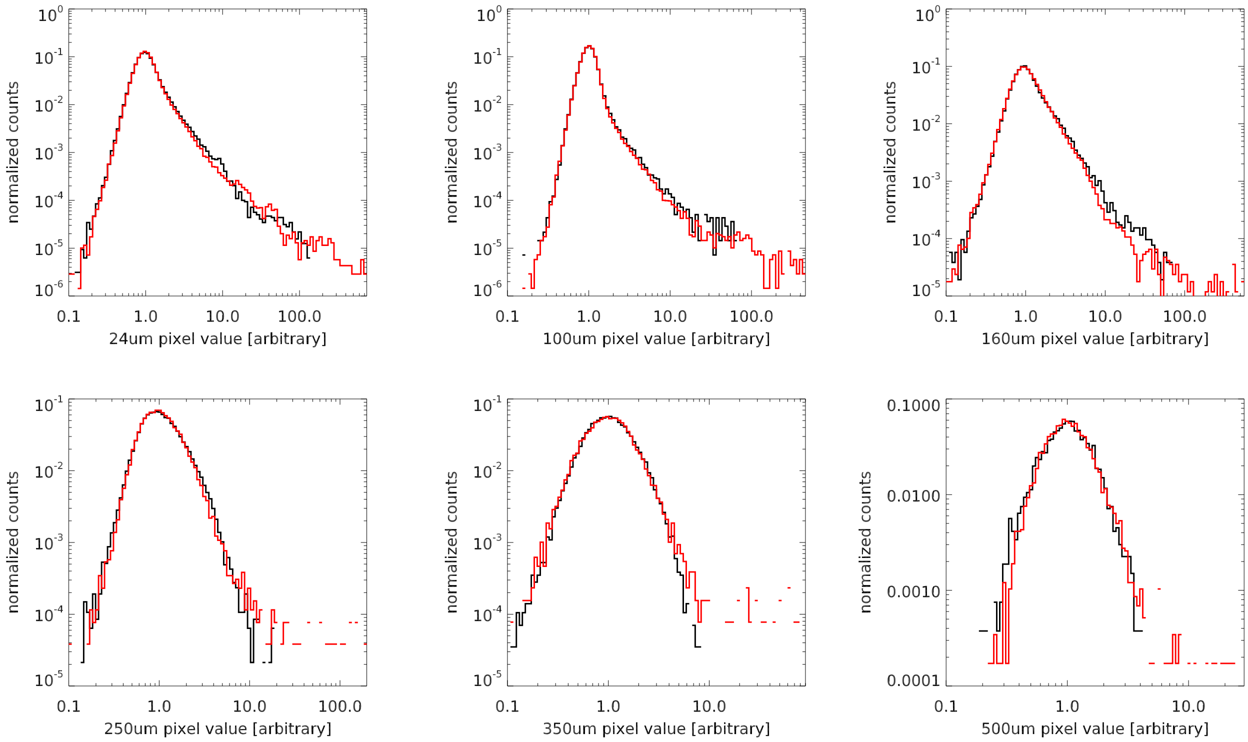

Figure 5: Pixel histogram distribution of the simulated FIR images versus real images in GOODS–South.

3 Installing *Ifni*

3.1 Forewords

Ifni 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).

3.2 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: Sorry, not supported.

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)

3.3 Install *phy*$_{++}$ and *Ifni*

The rest is a little bit harder, but not that much. Thanks to CMake, the installing process is the same on all computers. In the following, I will assume that you have a directory somewhere on your computer where you keep all your programming related stuff (e.g., the source code of TPHOT if you have tried to compile it).

1. Download the following archives and extract them inside this directory:

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

   In the end, this should create three directories:

   ```
   filter-db-master
   ifni-master
   phypp-master
   ```

2. 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. 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 directory in which you want to install the library. For example, if you want to install it in `/opt/local`, just replace "..." by `"/opt/local"`. This will install the library headers in `/opt/local/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 *Ifni*. 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 have root access
source ~/.phypprc
```

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

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

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"
```

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

```
make install
```

The `ifni` binary will be created in `../bin`. See, not that hard!

3.4 Making sure everything works

Navigate into the ifni-master/bin directory and call:

```
./ifni verbose maglim=27 filter_db=../../filter-db-master/db.dat
```

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

```
note: initializing filters...
note: 15 optical bands and 8 IR bands
note: initializing redshift bins...
note: min dz: 0.05, max dz: 0.773951
note: 36 redshift slices
note: estimating redshift-dependend mass limit...
note: will generate masses from as low as 4.56944, up to 12
note: reading mass functions...
note: found 10 redshift bins and 181 mass bins
note: generating redshifts...
note: generated 50142 galaxies
note: generating masses...
note: generating morphology...
note: generating SFR...
note: assigning optical SEDs...
[---------------------------------------] 900 100%, 285ms elapsed, 0ns left, 285ms total
[---------------------------------------] 900 100%, 304ms elapsed, 0ns left, 304ms total
note: assigning IR SED...
note: computing fluxes...
note: computing optical fluxes...
[---------------------------------------] 50142 100%, 8s elapsed, 0ns left, 8s total
[---------------------------------------] 49883 100%, 8s elapsed, 0ns left, 8s total
note: computing IR fluxes...
[---------------------------------------] 50142 100%, 5s elapsed, 0ns left, 5s total
note: generating sky positions...
[---------------------------------------] 36 100%, 1s elapsed, 0ns left, 1s total
note: saving catalog...
```

Also, a file called ifni-2015xxxx.fits (e.g., for me it was ifni-20150323.fits) weighting about 20MB 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('ifni-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 tranlsate this catalog into a Skymaker-compatible catalog, one per band. Try:

```
./ifni-2skymaker ifni-2015xxxx.fits band=f160w out=sky-f160w.cat
```

This should produce no output in the terminal, but create two files in the same directory, sky-f160w.cat, which is the Skymaker catalog, and sky-f160w-hdr.txt, which is the WCS header to feed to Skymaker.