Simulating images from Gencat mock catalogs
E. Merlin

To generate images starting from Gencat catalogs we used E. Bertin's software SkyMaker, which produces realistic simulated images including sources and a (non-correlated) noise map.

We wrapped SkyMaker into an ad-hoc user friendly Python script, to control all parameters with ease. In summary, the procedure is as follows:
- the catalogs list positions and structural parameters of sources in a given band
- SkyMaker reads such list and produces an image based on it, requiring the following additional parameters (to be given in a config file):
    * image size in pixels
    * gain
    * saturation level
    * exposure time
    * magnitude zero point
    * pixel size
    * seeing FWHM
    * simulated optical instrument features (mirrors diameter, etc.)
    * background magnitude per arcsecond
    * limit of allowed magnitudes

Some of these parameters are kept fixed: e.g, all images have dimensions of 16500x16400 pixels, with allowed magnitudes 16 to 31, the gain is assumed to be 1, all images have pixels size $ps=0.06''$, and the saturation level is 6553500 counts/s. The photometric zero point is fixed to 23.9. Fixing the exposure time to 10000 sec for all images, the only free parameter is the background magnitude. However, it is easier to input a magnitude limit e.g. at 1σ, which can be taken as equal to the ones obtained for the CANDELS observations, and hence compute the background magnitude for the simulated images. To do so, we start from the formula

$$\frac{S}{N}=\frac{(F_1 t_{\mathrm{exp}})}{\sqrt{\frac{F_1 t_{\mathrm{exp}}}{g}+\frac{B_1 t_{\mathrm{exp}} A}{g}+(R.O.N./g)^2}}$$

where F_1 is source flux in counts/s, B_1 is background flux in counts/s, A is pixel area of the source, and g is gain, which is fixed to 1. Assuming the R.O.N. to be zero, a source of magnitude equal to mag_{limit} has $S/N=1$ yielding $1=\frac{F_1 t_{\mathrm{exp}}}{\sqrt{(F_1+B_1 A)t_{\mathrm{exp}}}}$ and therefore $B_1=\frac{F_1^2 t_{\mathrm{exp}}-F_1}{A}$ where $F_1=10^{-0.4(mag_{limit}-ZP)}$, from which $mag_{background}=-2.5\log\frac{B_1}{ps^2}+ZP$.

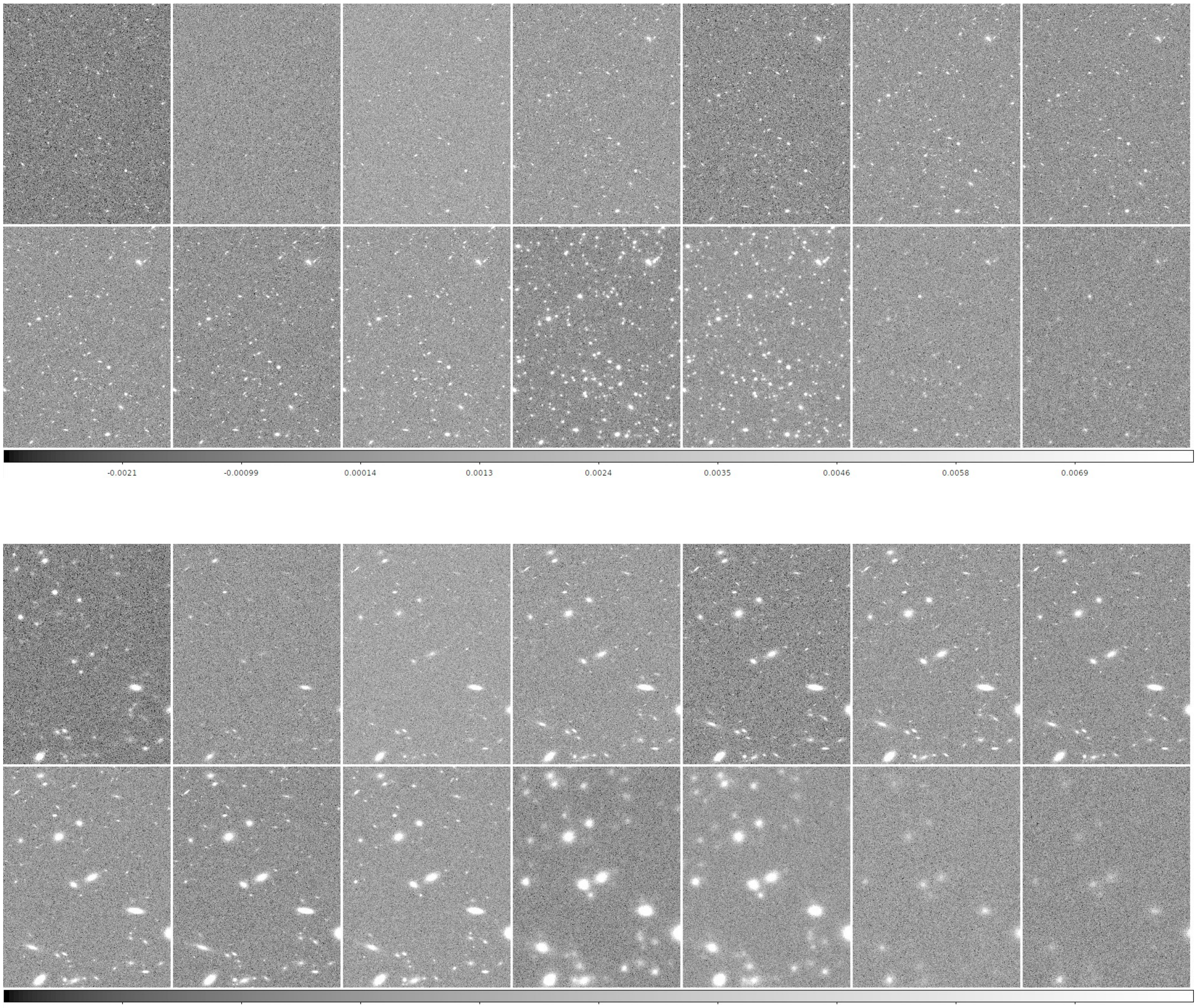
The limit magnitudes in 1 FWHM for all bands where computed from the nominal ones given in Guo+2014, as $m_{limit}=m_{1\sigma}=m_{5\sigma,CANDELS}+2.5\log 5$, with minor adjustments for "cosmetics":

VIMOS_U=29.5 (true: 29.72), ACS_f435=30.5 (true: 30.70), ACS_f606=30.5 (true: 31.10), ACS_f775=30.5 (true: 30.30), ACS_f814=30.5 (true: 30.60), ACS_f850=30.5 (true: 30.30), WFC3_f105=29.25 (true: 29.20), WFC3_f140=29.25 (true: 29.40), WFC3_f160=29.25 (true: 29.11), K_HawkI=28.2 (true: 28.20), IRAC1=27.15 (true: 27.15), IRAC2=27.15 (true: 27.15), IRAC3=25.5 (true: 25.50), IRAC4=25.5 (true: 25.47).

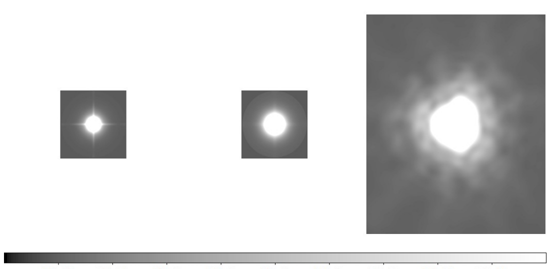
FWHM were taken as follows:
VIMOS_U=0.8 ”, ACS_f435=0.1 ”, ACS_f606=0.1 ”, ACS_f775=0.1 ”, ACS_f814=0.1 ”, ACS_f850=0.1 ”, WFC3_f105=0.2 ”, WFC3_f140=0.2 ”, WFC3_f160=0.2 ”, K_HawkI=0.4”, IRAC1=1.66 ”, IRAC2=1.7 ”, IRAC13=1.9”, IRAC4=2.0”.

The Python script we developed to produce the images calls SkyMaker iteratively to produce a set of ancillary files, i.e. a normalized image with pixels values in counts/s, a normalized and background subtracted image (obtained using the “true” background value computed as explained above), an RMS image obtained computing the standard deviation of the pixel values in the noise map image (no photon noise is therefore included).

To produce the images, SkyMaker needs a PSF. It can be generated internally, or it can be fed to the code as an external file. Internally generated PSFs have perfect circular simmetry. We used internally generated PSFs for all bands but for the four IRAC ones, because of the intrinsic asimmetry of the real IRAC PSFs. For IRAC images we therefore fed SkyMaker with “realistic” synthetic IRAC PSFs generated using a Python script by the H. Ferguson and S. Lee (CANDELS).



Regions within simulated images at different magnifications. Both panels, left to right, top to bottom: VIMOS_U, ACS_f435, ACS_f606, ACS_f775, ACS_f814, ACS_f850, WFC3_f105, WFC3_f140, WFC3_f160, K_HawkI, IRAC1, IRAC2, IRAC3, IRAC4.



Three of the adopted PSFs. Left to right: WF3_f160, K_HawkI, IRAC1.