# Source Finder - Report

C. FERRARI

*First test: removal of artifacts*

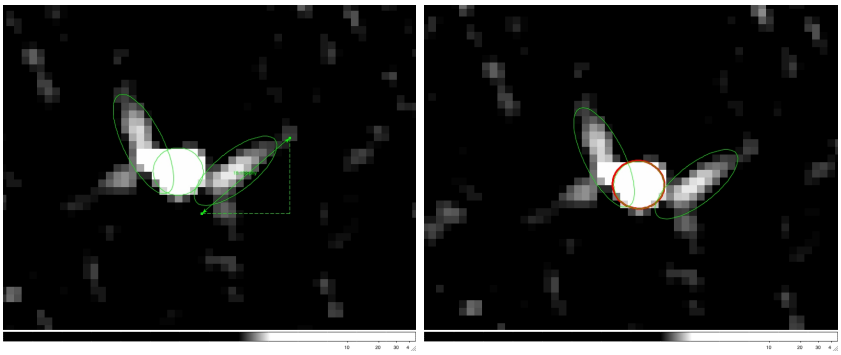

Figure 1: Left: artifacts around a bright source. PyBDSM was run with *Set #1* parameters (see text). Right: PyBDSM results with *Set #2* (red) and *Set #1* (green).

I firstly run PyBDSM on the image ~ferrari/Source_Extraction/VLSS.1.fits with the following parameters (*Set #1*):

filename = 'VLSS.1.fits'
advanced_opts = False
atrous_do = False
beam = None
flagging_opts = False
frequency = 74e+6
interactive = True
mean_map = 'default'
multichan_opts = False
output_opts False
polarisation_do = False
psf_vary_do = False
rms_box = (100.0,25.0)
rms_map = True
shapelet_do = False
spectralindex_do = False
thresh = 'hard'
thresh_isl = 3.0
thresh_pix = 5.0

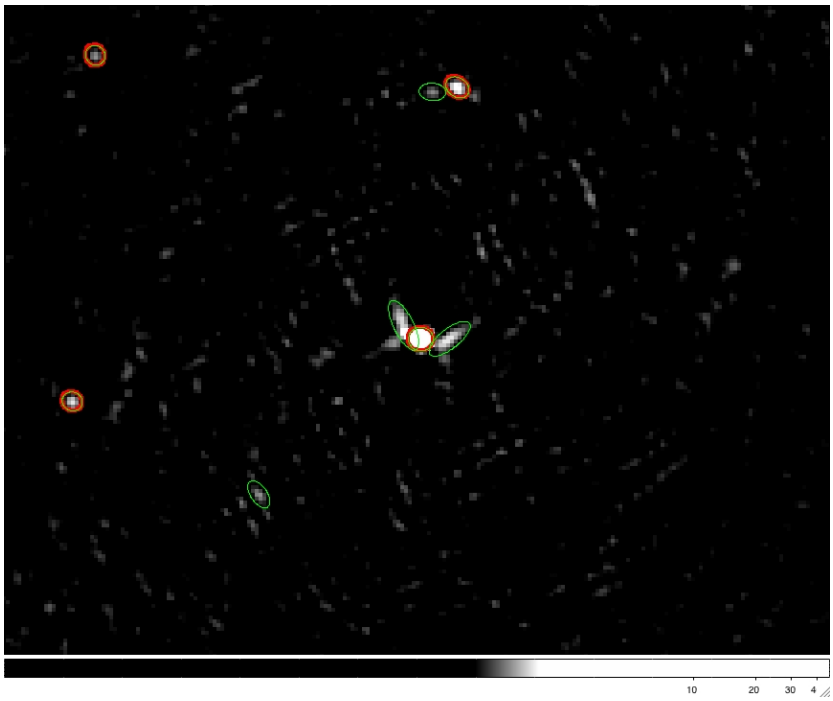


Figure 2: PyBDSM results with *Set #2* (red) and *Set #1* (green).

Results are promising for most of the sources, with some artifacts left around a bright source (see left panel of Fig. 1)[1]. I follow the approach suggested in Sect. 9.5.2 of Cookbook v.9.0, i.e. I measure the size of the artifact (in pixels) and I use this information in the "rms_box" parameter (*Set #2*):

filename = 'VLSS.1.fits'
advanced_opts = False
atrous_do = False
beam = None
flagging_opts = False
frequency = 74e+6
interactive = True
mean_map = 'default'
multichan_opts = False
output_opts = False
polarisation_do = False
psf_vary_do = False
**rms_box = (15.0,8.0)**
rms_map = True
shapelet_do = False
spectralindex_do = False
thresh = 'hard'
thresh_isl = 3.0

[1]Coordinates of the source: RA = 13:00:32 – Dec = +40:09:09.74.

thresh_pix = 5.0

Right panel of Fig. 1 shows, in red, the result around the bright source which gave artifacts with *Set #1*. The artifact problem is well solved in this case, but, by zooming out (see Fig. 2), I see that some sources are lost. I have therefore tried to increase the "rms_box" parameter, by putting an intermediate value between 100 (*Set #1*) and 15 (*Set #2*). *Set #3*:

filename = 'VLSS.1.fits'
advanced_opts = False
atrous_do = False
beam = None
flagging_opts = False
frequency = 74e+6
interactive = True
mean_map = 'default'
multichan_opts = False
output_opts = False
polarisation_do = False
psf_vary_do = False
**rms_box = (30.0,15.0)**
rms_map = True
shapelet_do = False
spectralindex_do = False
thresh = 'hard'
thresh_isl = 3.0
thresh_pix = 5.0

This is the worst result, since I keep the artifacts around bright sources but I also loose detections (see top panel in Fig. 3). I also run PYSE on this image (*Set #1 - PYSE*):

~swinbank/sourcefinder/tkp/bin/pyse VLSS.1.fits –detection=5 –analysis=3 –regions –regionext=reg

as well as Duchamp + buildsky (*Set #1 - Duchamp*):

imageFile VLSS.1.fits
logFile logfile.txt
outFile results.txt
spectraFile spectra.ps
minPix 5
flagATrous 0
snrRecon 10.
snrCut 5.
minChannels 3
flagBaseline 0
flagKarma 1
karmaFile duchamp.ann
flagnegative 0

flagMaps 0
flagOutputMask 1
flagMaskWithObjectNum 1
flagXOutput 0

PYSE is performing better both in terms of non-detection of artifacts and completeness of the output catalogue (see bottom panel of Fig. 3).

---

**Bug:** in PyBDSM, by typing "inp write_catalog", you get:

bbs_patches....":

For BBS format, type of patch to use:
None => no patches.
'single' => all Gaussians in one patch.
'gaussian' => each Gaussian gets its own patch.
'source' => all Gaussians belonging to a single source are grouped into one patch.

But if you want each Gaussian you have to type *bbs_patches = 'Gaussian'*. Either remove the capital letter from one or insert it in the other.

---

**Question:** Is it possible to get region files both with and without text written close to objects?

---

**Updates for the cookbook:** "write_catalog" instead of "write_gaul" update its parameters

---

By running PyBDSM on ~ferrari/Source_Extraction/VLSS.2.fits, I noticed another problem in the extraction of bright point sources, both with *Set #1* and with *Set #2*. In the case shown in Fig. 4 (top panel) three components were associated to one source[2] (see also Fig. 5. To this regard: **Why the three ellipses do not correspond to those stored in the ds9 region file?**). I do not clearly see why. Just one peak is present in the source, as the two profiles of Fig. 4 show. I did not have the same problem with PYSE (Fig. 4, bottom panel), that I run using the following parameters (*Set #2 - PYSE*):

~swinbank/sourcefinder/tkp/bin/pyse VLSS.2.fits –detection=5 –analysis=3 –regions –regionext=reg –bmaj=0.022 –bmin=0.022 –bpa=0

I finally run Duchamp and buildsky. Similarly to PYSE, these tools allow not to detect the artifacts around the bright source and to have just one component (and not three ellipses, as with PyBDSM; top panel of Fig. 6). I get the following flux values for the bright source:

- casaviewer: $\approx$ 46 Jy
- PyBDSM: $\approx$ 45 Jy + 4 Jy + 2 Jy

[2] RA = 12:56:57– Dec = +47:20:24

- PYSE: $\approx$ 55 Jy
- buildsky: $\approx$ 46 Jy

On the other side, PyBDSM is the most successful in separating a complex tailed galaxy in its different components (bottom panel of Fig. 6).

---

**Note:** John Swinbank has just implemented the possibility to get fluxes in output for PYSE. We may derive a skymodel for BBS and make further tests on performances in calibration.

---

To conclude, concerning point sources, there are no substantial differences between the available tools, even if, in order to optimize source extraction, some settings need to be done by hand, in particular in the case of PyBDSM. I will provide the different catalogs to Giulia Macario so that she can test how they perform in terms of calibration on the A1682 dataset (VLSS.2.fits is centered on this cluster). I will further test these tools on more complex images. At this stage I think that it is important to discuss what we will use for calibration in MSSS.

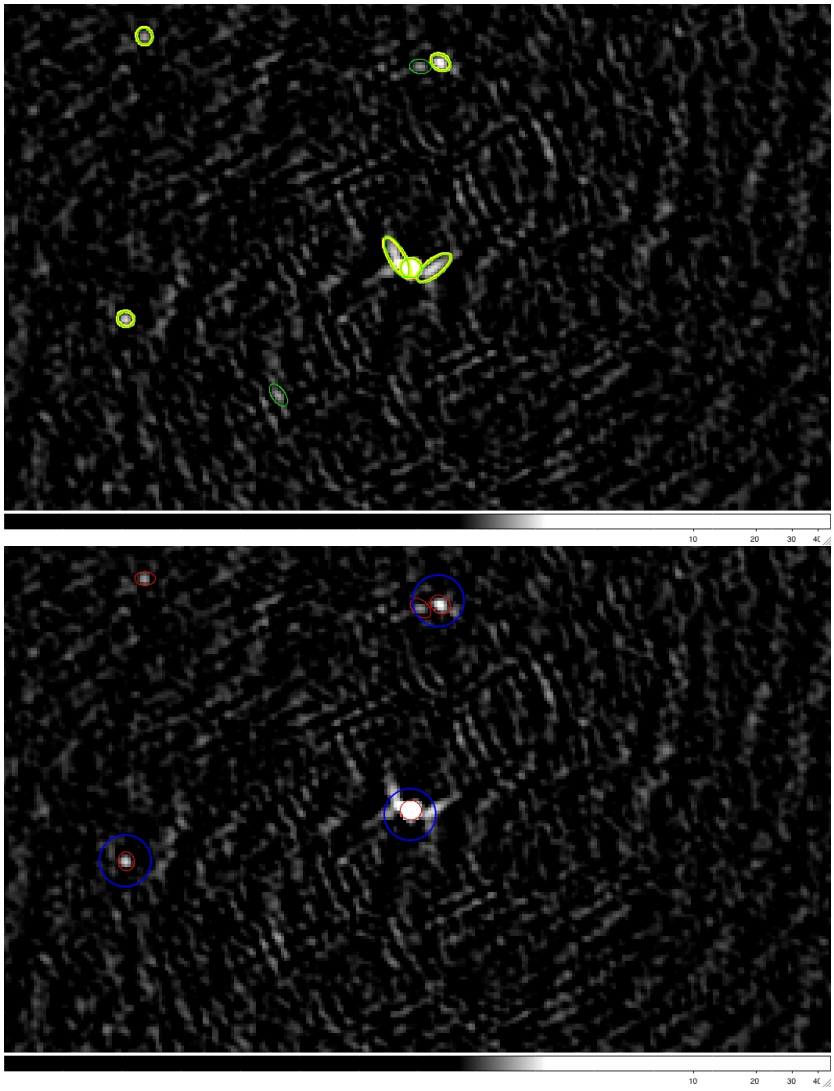

Figure 3: Top: PyBDSM results with *Set #1* (green) and *Set #3* (yellow). Bottom: PYSE (red) and Duchamp + buildsky results (blue) using *Set #1 - PYSE* and *Set #1 - Duchamp*, respectively.

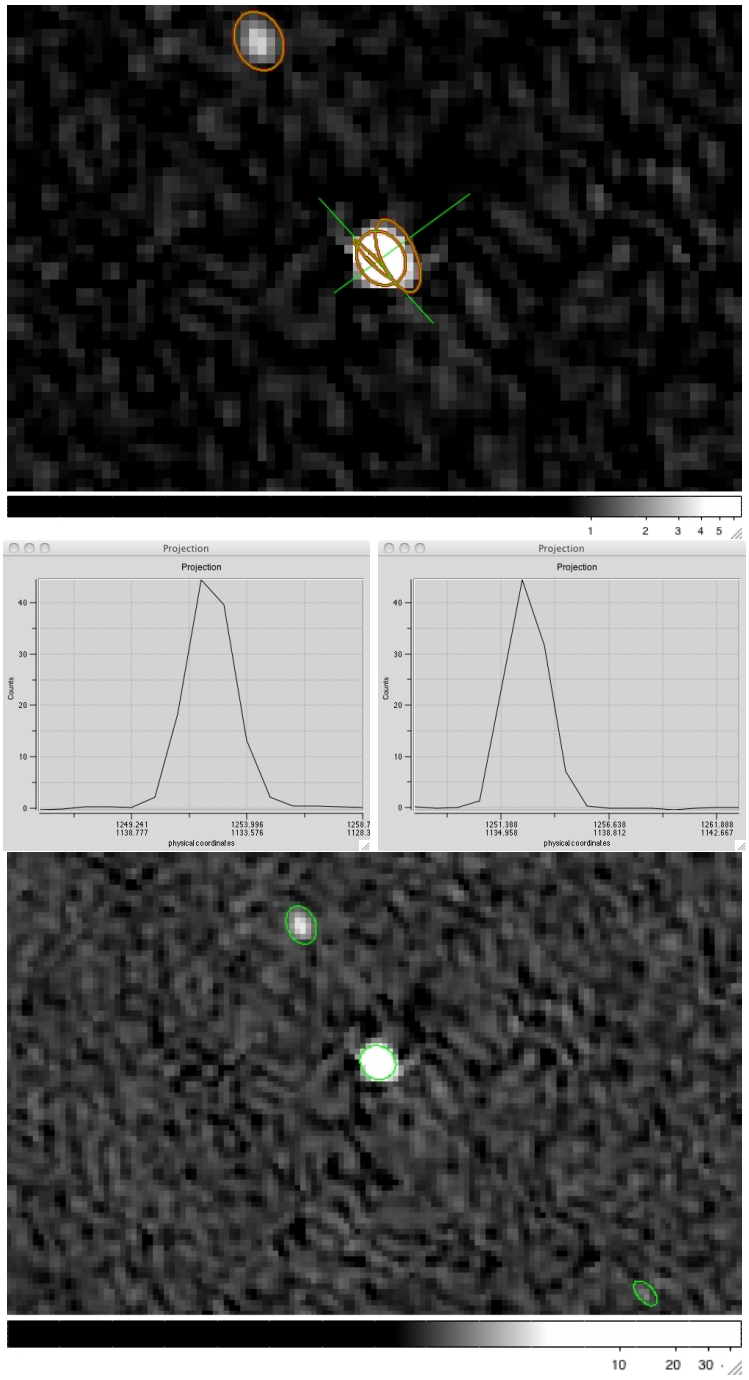


Figure 4: Top: PyBDSM results with *Set #2* (red) and *Set #1* (green) on a bright source. Middle: the green lines indicate the directions along which profiles have been extracted. Bottom: PYSE results with *Set #2* - *PYSE* on the same source shown in the top panel.

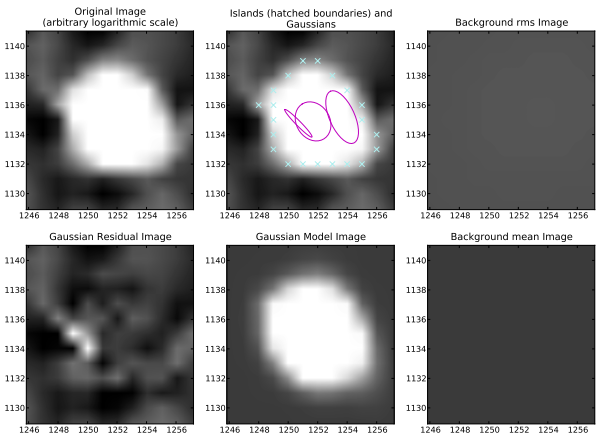


Figure 5: Islands found by PyBDSM using *Set #1*.

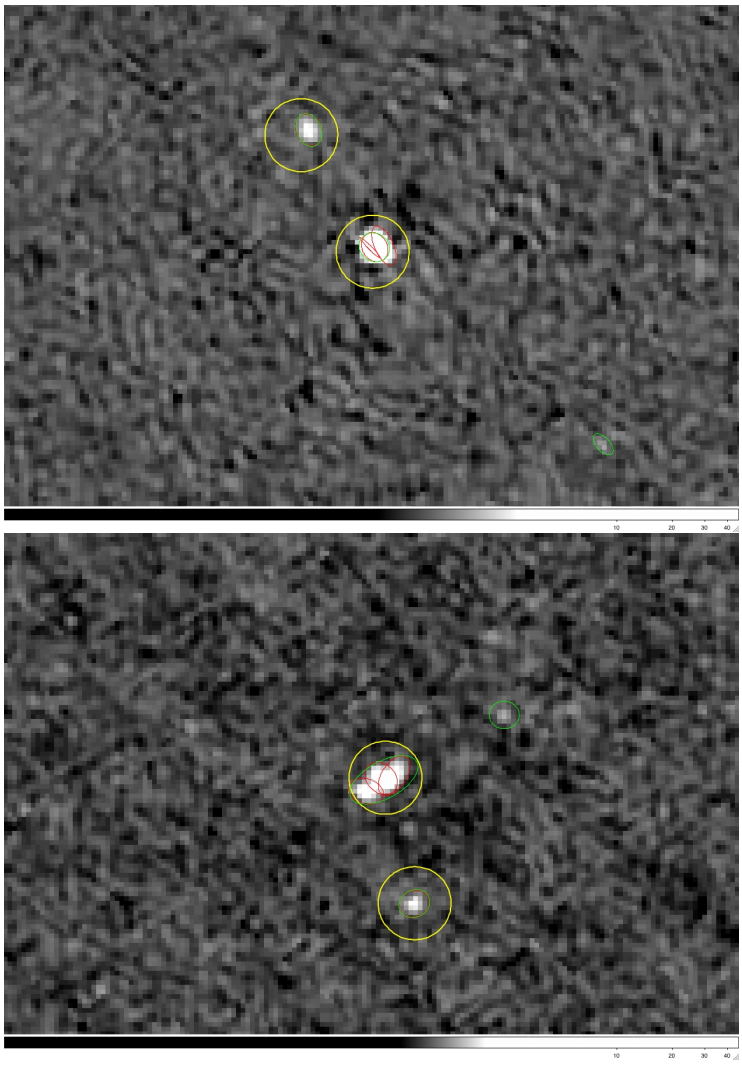

Figure 6: Yellow, green and red represent, respectively, buildsky, PYSE and PyBDSM results.