Making the ELAIS-N1 DR2 PyBDSF catalogue (v1.0)

This document briefly summarises the steps taken to produce the ELAIS-N1 Deep Field final radio catalogue with PyBDSF, and in particular highlights steps that are different from those in the first LoTSS Deep Fields data release (and wider LoTSS survey).

1 Motivation

The motivation for adjusting the methodology by which PyBDSF is used is the following:

- PyBDSF generates the RMS map prior to removal of any sources
- This means that when there are lots of sources, such as in the final Deep Fields maps, the RMS is elevated
- This affects detection of sources, their SNR (= peak flux density / rms noise) and leads to more sources only being detected in wavelet modes
- If we want to use sources for studies with an SNR cut then you can lose a lot of sources without access to the wavelet RMS maps and these are not stored in the PyBDSF catalogue
- An examination of the residual images from the LoTSS Deep Fields first data release (and for LoTSS) reveals that a small number of very large bright sources can be missed
- The deep multi-wavelength data in the Deep Fields, which provide a host identification for almost all radio sources, allow a direct test of whether a lower significance threshold could be set with PyBDSF to obtain more sources without a significant fraction of false detections.

2 A multi-pass PyBDSF approach to optimise RMS noise estimates

To make the primary PyBDSF catalogue we take the general approach of first removing the sources, then calculating the RMS map from this residual image (in order to get the true values), and then supplying PyBDSF with this RMS map as an input (i.e. double-pass detection). However, there is the subtlety in that we need to retain the bright ($>150\sigma$) sources in the image, so that PyBDSF correctly adjusts the RMS box around bright sources. Specifically, the process was therefore:

1. Run PyBDSF in the standard configuration (see below). From this save the residual maps and the source / Gaussian catalogues (hereafter known as Resid_Default, Src_Default and Gaul_Default)
2. Using the Src_Default and Gaul_Default catalogues, find the 150σ sources and find the associated Gaussians for each of the 150σ sources.
3. Use an elliptical cut out with radius = 1.5 x Major/Minor axis of the sources to find the model of the Gaussians which make up the 150σ sources. Inject the models for the bright sources back into the Resid_Default image to create Resid_DefaultWithBright image.
4. Run PyBDSF using the same parameters as in (1) on Resid_DefaultWithBright in order to determine the RMS map in a manner unaffected by the high density of faint sources in the image - this is known as RMS_Deep
5. Run PyBDSF over the original image supplying it mean maps = 0 and rms maps of RMS_Deep and a version of RMS_Deep which is primary beam uncorrected. This creates catalogues Srl_Deep and Gaul4 as well as RMS map RMS_Final (==RMS_Deep) and Residual map Resid_Deep

Figure 1 compares the outcome of this approach, for different thresholds in SNR (see Sec. 4). A comparison of the blue and brown curves (default PyBDSF *vs.* this double-pass detection approach for the standard 5σ threshold), shows how the double-pass detection leads to an the increase in the number of sources with SNR>5, as sources are picked up in the map using the deeper rms map and no longer rely upon being detected in the wavelet scale fitting. This further produces as a suppression of wavelet-mode detected sources with SNR<5.

Standard PyBDSF detection mode:

```
bdsf.process_image(imf, detection_image=appf, thresh_isl=3.0, thresh_pix=4.0,
rms_box=(150,15), rms_map=True, mean_map='zero', ini_method='intensity',
adaptive_rms_box=True, adaptive_thresh=150, rms_box_bright=(60,15), group_by_isl=False,
group_tol=10.0, output_opts=True, output_all=False, atrous_do=True, atrous_jmax=4,
flagging_opts=True, flag_maxsize_fwhm=0.5, advanced_opts=True, blank_limit=None,
frequency=restfrq)
```

PyBDSF detection mode when supplying maps:

```
bdsf.process_image(imf, detection_image=appf, thresh_isl=3.0, thresh_pix=4.0,
rms_box=(150,15), rms_map=True, mean_map='zero', ini_method='intensity',
adaptive_rms_box=True, adaptive_thresh=150, rms_box_bright=(60,15), group_by_isl=False,
group_tol=10.0, output_opts=True, output_all=False, atrous_do=True, atrous_jmax=4,
flagging_opts=True, flag_maxsize_fwhm=0.5, advanced_opts=True, blank_limit=None,
frequency=restfrq, rmsmean_map_filename=[meanf, rms_image_withBright_f],
rmsmean_map_filename_det=[meanf, rms_image_withBright_appf])
```

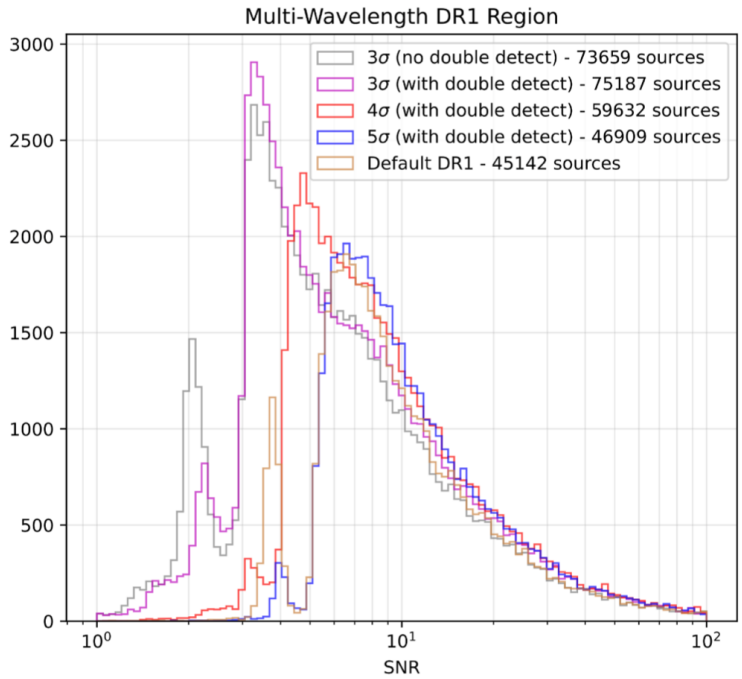

Figure 1: SNR comparisons of surveys using previous default mode (brown), and to using this "double" detection mode PyBDSF with either a 5σ (blue), 4σ (red) or 3σ (magenta) detection threshold. A 3σ run as in default mode, but using a 2/3σ limit for the island boundary and peak limits is also shown (grey).

3 Missing large and bright components

Once we have Resid_Deep, we want to detect (and add to the catalogue) the large, bright sources which remain in the image. These were found to be able to be included in the catalogue through changing the flag_maxsize_bm parameter to a larger value; however, changing that parameter on the original PyBDSF run leads to poorer performance on faint sources (where large halos can be erroneously detected). To add these missing bright sources back in, we take the following approach:

1. Run PyBDSF on Resid_Deep, with the change of setting flag_maxsize_bm=100, when supplying the RMS map (see below) and removing the atrous fitting mode.
2. Create Resid_large, Model_large, Srl_large and Gaul_large
3. Using a 10 sigma threshold on the peak flux density, include such sources (and their Gaussians) in the Srl_Deep/Gaul_Deep catalogues to create Srl_Final/Gaul_Final catalogues. Visual inspection indicated that sources with less than 10 sigma tended to mostly be false detections in the residual image.
4. Create a final residual map by subtracting the Model_large from Resid_Deep to create Resid_Final

PyBDSF detection mode for large sources:

```
img_supply_large = bdsf.process_image(int_residf,detection_image=app_residf,thresh_isl=3.0,
thresh_pix=10.0, rms_box=(150,15), rms_map=True, mean_map='zero', ini_method='intensity',
adaptive_rm s_box=True, adaptive_thresh=150, rms_box_bright=(60,15), group_by_isl=False,
group_tol=10.0, output_opts=True, output_all=False, flagging_opts=True,
flag_maxsize_fwhm=0.5, advanced_opts=True, blank_limit=None, frequency=restfrq,
rmsmean_map_filename=[meanf, rms_image_withBright_f], rmsmean_map_filename_det=[meanf,
rms_image_withBright_appf], flag_maxsize_bm=100)
```

4 Optimising the detection threshold

The 5σ peak SNR threshold that has been adopted as standard in LoTSS catalogues is conservative. PyBDSF is also significantly incomplete down to 5σ.

Using the very high host galaxy identification fraction in the Deep Fields (>97%), we test the reliability of the sample exploring to lower SNR thresholds. To do this, we ran the double-pass detection method outlined in Sec. 2 up to and including stage 4, using a 5σ detection threshold. We then ran PyBDSF supplying the 5σ derived RMS map but then extracting sources down to 3σ (setting thresh_isl=2.0 and thresh_pix=3.0). We then considered those sources within the region of the multi-wavelength imaging which have a major axis size below 10 arcsec, as these are predominantly suitable for likelihood ratio (LR) cross-matching. Figure 2 shows the fraction of sources that have a host galaxy match above the LR threshold, for different bins in source SNR, in this 3σ catalogue.

The blue horizontal line in Figure 2 shows the average fraction of host galaxy IDs for the full sample. Note that this is a little below 100%, because some of these sources may be components of multi-component sources and thus not have an ID, while for some others the host galaxy may be too faint and undetected in the optical/IR data. As can be seen in Figure 2, the host galaxy ID fraction remains around the average value down to SNR=5, then decreases slightly over $4 < \mathrm{SNR} < 5$, and then begins to fall quickly away. The drop in host galaxy ID fraction is caused by some of the PyBDSF detections not being genuine sources at lower values of SNR. The analysis shows the between 4 <SNR< 5, only 2-3% of PyBDSF sources will be false detections, while below SNR=4 that fraction begins to increase rapidly.

Figure 1 compares (in the blue, red and purple lines) the SNR distribution of sources down using double-pass detection when run to final 3σ, 4σ and 5σ thresholds, as described above, and quantifies the total number of

sources detected in the multi-wavelength regions. The total number of sources detected in the 4σ double-pass detection catalogue is $\approx$ 30% higher than the number detected in the default 5σ (or double-pass 5σ) catalogue. This is a significant addition. We note that these sources do not all correspond to 4<SNR$\leq$5 sources; the fraction of sources in the 4σ threshold catalogue with 4 <SNR< 5 is about 20%. The other additional sources arise both due to a significantly increased source completeness over the range 5 <SNR< 6 (see Fig. 1), and because the RMS map produced using a 4σ threshold will be different (and deeper) compared to that produced when a 5σ limit is used. This will affect source numbers detected above a given SNR threshold.

Our judgement is that there are significant benefits of this method: additional sources; improved completeness; and less sources detected in the wavelet mode which can appear to be below the SNR limit of the catalogue. These reasons merit selecting down to the 4σ level, even when noting that 2-3% of the additional sources will be false detections. Many of these false detections may in any case be flagged and removed during the optical cross-matching phase. Extending the selection further down to a threshold of 3σ would produce an even larger catalogue, but with around 10% of these additional sources being false; this threshold can also be seen in Figure 1 to affect the source distribution at larger SNR as well. Hence, despite many of the additional sources being real, this threshold was deemed too deep. Users interested in accessing these fainter flux densities for specific galaxies should instead make use of forced photometry measurements from the provided radio maps.

In summary, the final PyBSDF catalogue is selected down to thresh_isl=3.0 and thresh_pix=4.0, as summarised in the parameter list in Section 2.

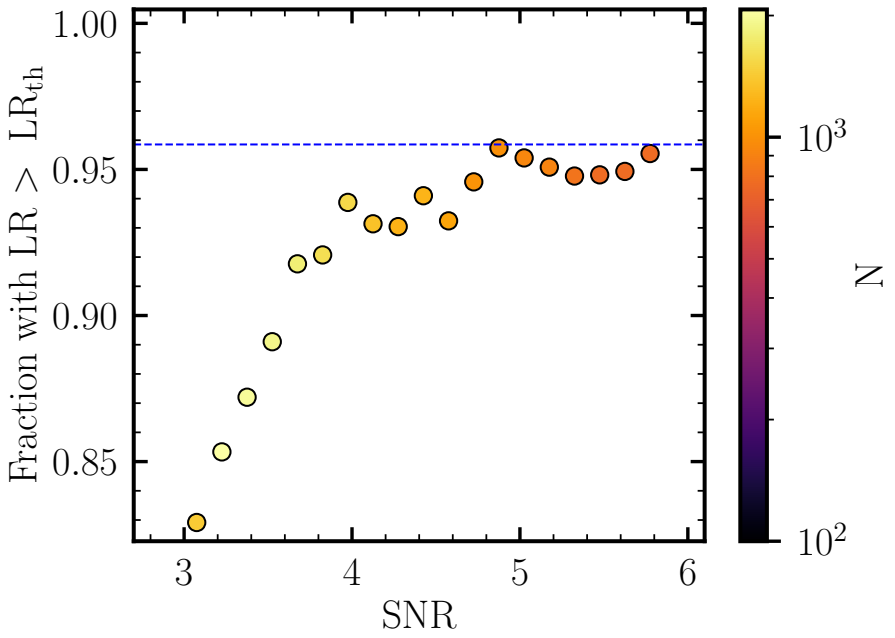

Figure 2: The fraction of compact (`Maj` < 10 arcsec) radio sources with a LR match as a function of the the SNR. The color of the points corresponds to the number of sources in each SNR bin. The blue horizontal line corresponds to the overall LR identification rate obtained.