Please watch the video at https://youtu.be/4o3l9kb3ASI (Bullet Cluster) and https://youtu.be/MHznxXPHMMk (AbellS1063) for detailed instructions on how to use the data analysis scripts. Hopefully, it will not put you to sleep! More detailed written instructions may follow.

external python packages needed

numpy, matplotlib, scikit-image, astropy, scipy, shapely , json, OpenCV, cosmocalc (at https://cxc.harvard.edu/contrib/cosmocalc/)

Installation: Put the python code in some directory that's in your PYTHONPATH, make the scripts executable, and hopefully things will work for you. The code has been tested with python 3.8 on Ubuntu 20.04

There are two main scripts in the package - get_morphology_images.py and get_galaxy_parameters.py.

Get_morphology_images uses morphological erosion and dilation to remove background sources from a radio astronomy image. It extends the technique described in Rudnick, 2002 https://iopscience.iop.org/article/10.1086/342499/pdf. See https://en.wikipedia.org/wiki/Mathematical_morphology for a general description of mathematical morphology.

The process can be described through the following equations:

o = original image

d - output from erosion/dilation

t = white TopHat, which should show only 'compact' structures

t = o - d

m = mask derived from a comparison where t > some signal m * t = m * (o - d)

o_d = output diffuse image

=o - m * t

=o - (m * o - m * d)

=o - m * o + (m * d)

m*d would add the masked dilated image to the 'diffuse' image and we do not want to do that so we ignore it to get

o_d = o - m * o and

o_c = image of compact objects = m * o

so the original image equates to o_d + o_c

We may want to judicious add selected components of o_c to o_d to get a final o* We select the components of o_c we wish to add by masking their defining polygons to get a mask m_c

o* = o_d + m_c * o_c

Now, o_d or o* will still contain areas with value 0 where we subtracted off o_c. We can fill in the values that we would `expect' to find in these areas by inpainting values from the surrounding diffuse emission. We can do this by either using Navier-Stokes inpainting or Fast Marching Method inpainting. (See e.g http://www.ifp.illinois.edu/~yuhuang/inpainting.html) This will generate a an image o_inp which we can add to o_d or o* to obtain a 'filled in' diffuse image, and subtract from o_c to get the 'actual' point source signal.

Get_galaxy_parameters integrates the signal contained within specified polygon areas of a radio astronomy image to derive integrated flux densities and other parameters of a radio source.